Fitting Piecewise Linear Continuous Functions

Alejandro Toriello*
Daniel J. Epstein Department of Industrial and Systems Engineering
University of Southern California
3715 McClintock Avenue, GER 240
Los Angeles, California 90089
toriello *at* usc *dot* edu

Juan Pablo Vielma
Department of Industrial Engineering
1048 Benedum Hall
University of Pittsburgh
3700 O'Hara Street
Pittsburgh, Pennsylvania, 15261
jvielma *at* pitt *dot* edu

August 25, 2011

Abstract

We consider the problem of fitting a continuous piecewise linear function to a finite set of data points, modeled as a mathematical program with convex objective. We review some fitting problems that can be modeled as convex programs, and then introduce mixed-binary generalizations that allow variability in the regions defining the best-fit function's domain. We also study the additional constraints required to impose convexity on the best-fit function.

Keywords: integer programming, quadratic programming, data fitting/regression, piecewise linear function

*Corresponding author. Office: +1(213)740-4893. Fax: +1(213)740-1120.

1 Introduction

The problem of fitting a function of some prescribed form to a finite set of data points is fundamental and has been studied for hundreds of years. In one form or another, fitting has applications in areas as varied as statistics, econometrics, forecasting, computer graphics and electrical engineering.

Within optimization, fitting problems are cast as convex norm minimization models whose properties are well known (Boyd and Vandenberghe (2004); see also Bot and Lorenz (2011); Williams (2007)). For example, the classical least-squares linear fitting problem is an unconstrained quadratic program with a closed-form solution obtained by setting the gradient of the objective equal to zero. Similarly, more complex piecewise linear and piecewise polynomial fitting models can be formulated as constrained convex programs.

Continuous piecewise linear functions and their discontinuous extensions are also extensively studied within discrete optimization and *mixed-integer programming* (MIP), e.g. de Farias et al. (2008); Vielma et al. (2008, 2010); Wilson (1998). However, most related work in this field concentrates on the *modeling* of a given piecewise linear function and of the subsequent incorporation of the function into a MIP or more general mathematical program. Nonetheless, recent work in various areas of discrete optimization motivates the issue of efficiently fitting a continuous piecewise linear function to a set of points.

In *approximate dynamic programming* (ADP) (Bertsekas and Tsitsiklis, 1996; Novoa and Storer, 2009; Papadaki and Powell, 2002; Powell, 2007), solutions to a dynamic or periodic model are generated by approximating the value function of a state variable, and then optimizing single-period subproblems with respect to this approximation. When the state space is continuous, an approximate value function can be constructed by sequentially sampling from the state space, observing the samples' values, fitting a function to the observations, and then repeating the process with the updated value function. In Toriello et al. (2010), for example, a separable, piecewise linear concave value function is constructed using an algorithm of this type.

In *mixed-integer nonlinear programming* (MINLP), recent algorithmic and software developments combine branch-and-bound frameworks common in MIP with nonlinear and global optimization methodology, e.g. Abhishek et al. (2010); Belotti et al. (2009); Bonami et al. (2008); Geißler et al. (2011); Leyffer et al. (2008). The algorithms construct and refine polyhedral approximations of nonlinear, possibly non-convex constraints. Efficient fitting models could be used inside this framework to expedite the construction of the approximations.

Our main contribution is the introduction of mixed-binary models that solve continuous piecewise linear fitting problems under various conditions. We also devote attention to the additional constraints required to make the best-fit function convex. The continuous models we present encompass simpler cases where the function domain's partition is predetermined, and are extensions of known models (Boyd and Vandenberghe, 2004). Conversely, the mixed-binary models we introduce allow for more general fitting problems where the domain partition may be partially or wholly determined along with the best-fit function. We believe the use of integer programming for this type of problem is new, and the only similar work we are aware of is Bertsimas and Shioda (2007), where the authors apply MIP modeling techniques to classification problems.

The paper is organized as follows. Section 2 introduces our general fitting problem, reviews related work and covers continuous models. Section 3 introduces mixed-binary models for more complex fitting problems, where the additional difficulty comes from adding variability to the regions that define the best-fit function. Section 4 has computational examples that highlight the benefit of considering variable regions. Section 5 gives conclusions and indicates some directions for further research.

We follow standard MIP notation and terminology as much as possible; see, e.g. Nemhauser and Wolsey (1999). Our disjunctive programming notation follows Balas (1998).

2 Problem Definition

Suppose we are given a finite set of data points $(x^i, y_i) \in \mathbb{R}^n \times \mathbb{R}, i = 1, \ldots, m \in \mathbb{N}$. We are interested in the problem

$$\min \; \|w - y\|_q \tag{2.1a}$$

$$\text{s.t. } w_i = f(x^i), \forall\, i = 1, \ldots, m \tag{2.1b}$$

$$f \in F, \tag{2.1c}$$

where F defines a set of continuous piecewise linear functions over a common domain that contains all points x^i, and $\|\cdot\|_q$ is the ℓ_q-norm in $\mathbb{R}^m$. In other words, we would like the function $f^* \in F$ that best fits the data set according to the measure $\|\cdot\|_q$.

For our purposes, a *piecewise linear function* is a continuous function f with domain $\bigcup_{P \in \mathcal{P}} P$, where $\mathcal{P}$ is finite, each $P \in \mathcal{P}$ is a full-dimensional polytope, the interiors of any two $P, Q \in \mathcal{P}$ are disjoint, and f is affine when restricted to any $P \in \mathcal{P}$. Although not strictly necessary, we also make the common sense assumption that for distinct $P, Q \in \mathcal{P}$, $P \cap Q$ is a face (possibly $\varnothing$) of both P and Q. Finally, when fitting convex functions we assume that $\bigcup_{P \in \mathcal{P}} P$ is convex.

Although fitting models have been studied for general values of q (Gonin and Money, 1989), we focus on the cases $q \in \{1, 2\}$. For $q < \infty$, (2.1) is equivalent to

$$\min \sum_{i=1}^{m} |f(x^i) - y_i|^q \tag{2.2a}$$

$$\text{s.t. } f \in F. \tag{2.2b}$$

In the sequel, we treat (2.2) as our generic model. However, the extension to the ℓ_∞-norm case is achieved by simply replacing the summation with a maximum over all absolute differences.

From a fitting perspective, the models we study are either parametric or non-parametric. *Parametric* models construct a function by estimating the parameters (i.e. slopes and intercepts) that define it. *Non-parametric* models define function values at a predetermined set of points. The function's value for a general point is then calculated as a linear interpolation or extrapolation of the function values for a subset of the predetermined set. In MIP modeling terms, parametric fitting problems result in functions easily modeled in a *multiple choice* model, while non-parametric fitting problems yield functions that fit in a *convex combination* or *disagreggated convex combination* model (cf. Vielma et al. (2010)). In either case, however, if the function is convex or concave, it may be possible to model it in a purely linear model.

2.1 Literature Review

Fitting problems of various kinds arise in many different areas, and even when restricting to the piecewise linear case, we cannot hope to provide an exhaustive list of references. We instead give examples of papers that study (2.1) and related problems from different points of view. A thorough treatment of convex optimization models used in fitting, approximation and interpolation is Chapter 6 of Boyd and Vandenberghe (2004). A recent text on statistical techniques used for problems related to (2.1) is Ruppert et al. (2003).

The one-dimensional case ($n = 1$) of (2.1) has been extensively studied, e.g. in econometrics and forecasting. For instance, Strikholm (2006) attempts to determine the number of breaks in a one-dimensional piecewise linear function using statistical inference methods. The computer graphics and visualization community has concentrated on the two-dimensional case. Pottmann et al. (2000) address the issue of choosing an optimal partition of the domain of a quadratic function so the resulting implied piecewise linear interpolation's error is below an accepted tolerance. For general n, an important case of (2.1) is when the union of polyhedra $\mathcal{P}$ is not predetermined and must be chosen as part of the fitting problem. Some variants of this problem are studied as extensions of *classification* and *clustering* problems, e.g. Bertsimas and Shioda (2007); Lau et al. (1999); Pardalos and Kundakcioglu (2009). Ferrari-Trecate et al. (2001) use neural network methodology for a problem of this kind, where the best-fit piecewise linear function is allowed to be discontinuous. Holmes and Mallick (1999) enforce continuity and allow $|\mathcal{P}|$ to vary by assuming a probability distribution exists for all unknowns (including $|\mathcal{P}|$,) and updating the distribution based on observations; this approach is known as *Bayesian regression*. Magnani and Boyd (2009) introduce the convex piecewise linear fitting problem with undetermined $\mathcal{P}$ that we study in Section 3.2, and use a Gauss-Newton heuristic related to the k-means clustering algorithm. Bertsimas and Shioda (2007) use integer programming models for classification and discontinuous piecewise linear fitting. This paper is the most closely related to our work.

2.2 Known Continuous Models

We first consider the case when F is the set of piecewise linear functions defined over a predetermined union of polytopes $\mathcal{P}$. Let $V(P)$ be the set of vertices of $P \in \mathcal{P}$ and let $\mathcal{V}(\mathcal{P}) = \bigcup_{P \in \mathcal{P}} V(P)$. Also, for each $v \in \mathcal{V}(\mathcal{P})$ let $\mathcal{P}_v$ be the set of polytopes that contain v, and for each i let $P_i \in \mathcal{P}$ be a polytope that contains x^i. See Figure 1 for a two-dimensional example.

Since we explicitly consider each polytope $P \in \mathcal{P}$, practical fitting problems must restrict themselves to a fairly low dimension n, especially because $|\mathcal{P}|$ tends to depend exponentially on n.

In a parametric model, any $f \in F$ can be given by

$$f(x) = c^P x + d_P, \text{ for } x \in P, \tag{2.3}$$

where $(c^P, d_P) \in \mathbb{R}^{n+1}, \forall\ P \in \mathcal{P}$, $c^P x$ denotes the inner product between $c^P, x \in \mathbb{R}^n$, and the requirement that f is continuous implies

$$c^P x + d_P = c^Q x + d_Q, \forall\ x \in P \cap Q. \tag{2.4}$$

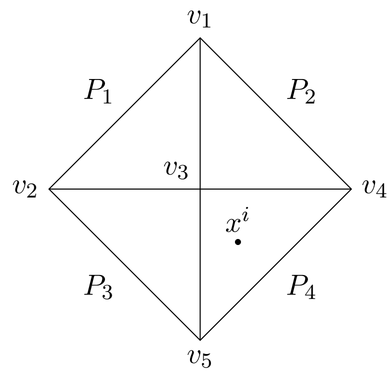


Figure 1: Domain example with $n = 2$. In this example, $V(P_2) = \{v_1, v_3, v_4\}$, $\mathcal{P}_{v_2} = \{P_1, P_3\}$ and $P_i = P_4$.

Then (2.2) becomes

$$\min \sum_{i=1}^{m} |c^{P_i} x^i + d_{P_i} - y_i|^q \tag{2.5a}$$

$$\text{s.t. } c^P v + d_P = c^Q v + d_Q, \forall\ P, Q \in \mathcal{P}_v, \forall\ v \in \mathcal{V}(\mathcal{P}) \tag{2.5b}$$

$$c^P \in \mathbb{R}^n, d_P \in \mathbb{R}, \forall\ P \in \mathcal{P}. \tag{2.5c}$$

Note that constraints (2.5b) enforce the continuity equation (2.4) by requiring that $f(x)$ be continuous at every vertex $v \in \mathcal{V}(\mathcal{P})$.

To restrict F to convex functions, we make use of the following result.

Proposition 2.1. *(Carnicer and Floater, 1996, Proposition 2.4) Any $f \in F$ is convex if and only if its restriction to any two polytopes from $\mathcal{P}$ that share a facet is convex.*

Let $W(P)$ be the set of facets of P that are also facets of another polytope in $\mathcal{P}$, and let $\mathcal{W}(\mathcal{P}) = \bigcup_{P \in \mathcal{P}} W(P)$. For any facet $\omega \in \mathcal{W}(\mathcal{P})$, let $P_\omega, Q_\omega \in \mathcal{P}$ be the unique pair of polytopes that satisfies $P_\omega \cap Q_\omega = \omega$. Choose $r^\omega \in \omega$ and $\delta^\omega \in \mathbb{R}^n \setminus \{0\}$ to satisfy $r^\omega + \delta^\omega \in P_\omega \setminus \omega$ and $r^\omega - \delta^\omega \in Q_\omega \setminus \omega$. To restrict problem (2.5) to convex functions, we add the constraints

$$\begin{aligned} &\frac{1}{2}(c^{P_\omega}(r^\omega + \delta^\omega) + d_{P_\omega} + c^{Q_\omega}(r^\omega - \delta^\omega) + d_{Q_\omega}) \\ &\qquad \ge c^{P_\omega} r^\omega + d_{P_\omega}, \forall\ \omega \in \mathcal{W}(\mathcal{P}). \end{aligned} \tag{2.6}$$

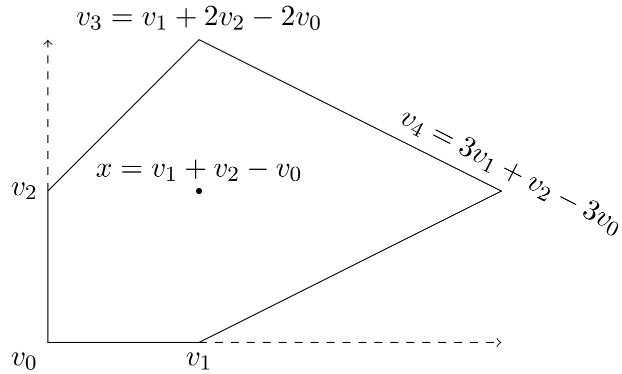


Figure 2: Non-parametric example in two dimensions with $V^+(P) = \{v_0, v_1, v_2\}$. Even though P has five vertices, only three are necessary to express any point as an affine combination. For this example, $\lambda_x^{P,v_1} = \lambda_x^{P,v_2} = 1$ and $\lambda_x^{P,v_0} = -1$.

Observe that r^ω is the midpoint of $r^\omega + \delta^\omega$ and $r^\omega - \delta^\omega$, so constraint (2.6) simply requires the midpoint convexity of f along the segment $[r^\omega - \delta^\omega, r^\omega + \delta^\omega]$, implying convexity because the defining functions are affine.

For the non-parametric case, let $V^+(P)$ for each $P \in \mathcal{P}$ be a set of $n+1$ affinely independent vertices of P. For any $x \in P$, let $\lambda_x^{P,v}, v \in V^+(P)$ be the set of *barycentric coordinates* (Rockafellar, 1970) of x with respect to $V^+(P)$; that is, the unique set of affine multipliers that expresses x as an affine combination of the elements of $V^+(P)$. (Figure 2 shows an example for $n = 2$.) Then any $f \in F$ can be expressed as

$$f(x) = \sum_{v \in V^+(P)} \lambda_x^{P,v} f_v, \text{ for } x \in P, \tag{2.7}$$

where $f_v = f(v), \forall\ v \in \mathcal{V}(\mathcal{P})$.

For each x^i, define $\lambda_i^{P_i,v}, v \in V^+(P_i)$ analogously to $\lambda_x^{P,v}$. The model then becomes

$$\min \sum_{i=1}^{m} \left| \sum_{v \in V^+(P_i)} \lambda_i^{P_i,v} f_v - y_i \right|^q \tag{2.8a}$$

$$\text{s.t. } f_v = \sum_{u \in V^+(P)} \lambda_v^{P,u} f_u, \forall\ P \in \mathcal{P}_v, \forall\ v \in \mathcal{V}(\mathcal{P}) \tag{2.8b}$$

$$f_v \in \mathbb{R}, \forall\ v \in \mathcal{V}(\mathcal{P}). \tag{2.8c}$$

Constraints (2.8b) are the non-parametric analogues of constraints (2.5b) from the previous model. They are necessary because function values within each P are expressed as affine combinations of the function values from $V^+(P)$, and vertex function values must match from one polytope to each adjacent one. However, if $\mathcal{P}$ is a triangulation or its higher-dimensional generalization, then $V^+(P) = V(P), \forall\ P \in \mathcal{P}$. In this case, constraints (2.8b) are implicitly satisfied, which means they can be removed and (2.8) goes back to a classical unconstrained linear fitting problem.

To enforce convexity of the best-fit function, we again use Proposition 2.1. For every $\omega \in \mathcal{W}(\mathcal{P})$, let $\lambda_{\omega+}^{P_\omega,v}$, $\lambda_{\omega}^{P_\omega,v}, \forall\ v \in V^+(P_\omega)$ and $\lambda_{\omega-}^{Q_\omega,v}, \forall\ v \in V^+(Q_\omega)$ respectively define the sets of affine multipliers for $r^\omega + \delta^\omega$, r^ω and $r^\omega - \delta^\omega$. The convexity enforcing constraints are

$$\begin{aligned} \frac{1}{2}\Bigg(\sum_{v \in V^+(P_\omega)} \lambda_{\omega+}^{P_\omega,v} f_v + \sum_{v \in V^+(Q_\omega)} \lambda_{\omega-}^{Q_\omega,v} f_v \Bigg) \\ \geq \sum_{v \in V^+(P_\omega)} \lambda_{\omega}^{P_\omega,v} f_v, \forall\ \omega \in \mathcal{W}(\mathcal{P}). \end{aligned} \tag{2.9}$$

As in the parametric case, these constraints simply enforce midpoint convexity along the segment between the points $r^\omega \pm \delta^\omega$.

Note that even though both the parametric and non-parametric models have equally many decision variables and constraints, the non-parametric model requires substantially more pre-processing, because the affine multipliers λ must be calculated for every vertex $v \in \mathcal{V}(\mathcal{P})$ and every x^i. For both models, the number of decision variables is $\Theta(n|\mathcal{P}|)$, and the number of constraints is $\Theta(|\mathcal{V}(\mathcal{P})| \max_v |\mathcal{P}_v|)$.

3 Mixed-Binary Models for Fitting over Variable Regions

3.1 Adding Variability to the Regions in Two Dimensions

We next generalize Section 2.2's non-parametric model to include some variability in $\mathcal{P}$. We concentrate on the two-dimensional case, which is already of significant interest, although the model may in theory be extended to any dimension. A similar but more general problem was studied by Pottmann et al. (2000) to approximate two-dimensional quadratic functions.

Following our previously introduced notation, let $(x^i, y_i) \in \mathbb{R}^2 \times \mathbb{R}, \forall\ i = 1, \ldots, m$, and assume in addition that $y_i \in [0, U], \forall\ i$. This assumption can be made without loss of generality by adding an appropriate positive constant to every y_i. Let $0 < b_j^1 < \cdots < b_j^p$ for $p \in \mathbb{N}$, $j = 1, 2$ with $b_j^p > x_j^i, \forall\ i, j$ be a partition of the function domain into p^2 rectangles. (The case when the number of breakpoints varies by dimension is a simple extension.)

Let $R = [b_1^{k_1-1}, b_1^{k_1}] \times [b_2^{k_2-1}, b_2^{k_2}]$ be any rectangle in the grid, and let v_0, v_1, v_2 and v_3 be its four vertices (see Figure 3.) R may be *triangulated* in one of two ways: If we choose to divide R along segment v_1v_2, we obtain triangles $v_0v_1v_2$ and $v_1v_2v_3$, whereas if we choose segment v_0v_3, we obtain triangles $v_0v_1v_3$ and $v_0v_2v_3$. Instead of fixing the triangulation beforehand (as we would do in Section 2.2), our next model adds the choice of two possible triangulations for each rectangle. The partition $\mathcal{P}$ is then one of the 2^{p^2} possible triangulations of the domain.

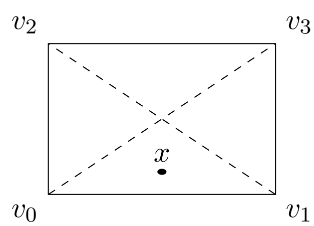


Figure 3: Two possible triangulations of rectangle R.

Let $x \in R$ and assume x lies both in $v_0v_1v_2$ and in $v_0v_1v_3$. It may therefore be expressed as a convex combination of the vertices of either triangle. Let $\lambda_x^{0,v}, v \in \{v_0, \ldots, v_3\}$ be the convex multipliers for x with respect to $v_0v_1v_2$ (with $\lambda_x^{0,v_3} = 0$,) and define $\lambda_x^{1,v}$ analogously with respect to $v_0v_1v_3$. Letting $f_v = f(v)$, we have the natural disjunction

$$f(x) = \lambda_x^{0,v_0} f_{v_0} + \lambda_x^{0,v_1} f_{v_1} + \lambda_x^{0,v_2} f_{v_2} \quad \vee \quad f(x) = \lambda_x^{1,v_0} f_{v_0} + \lambda_x^{1,v_1} f_{v_1} + \lambda_x^{1,v_3} f_{v_3},$$

expressed more generically as

$$f(x) = \sum_{v \in \{v_0, \ldots, v_3\}} \lambda_x^{0,v} f_v \quad \vee \quad f(x) = \sum_{v \in \{v_0, \ldots, v_3\}} \lambda_x^{1,v} f_v. \tag{3.1}$$

Similarly, for every $x^i \in R$, let $\lambda_i^{0,v}, \forall\ v \in \{v_0, \ldots, v_3\}$ be the convex multipliers of x^i when we triangulate R along v_1v_2, and define $\lambda_i^{1,v}$ similarly for the triangulation along v_0v_3. Letting $f_i = f(x^i)$, we have the aggregate disjunction

$$\left(f_i = \sum_{v\in\{v_0,\ldots,v_3\}} \lambda_i^{0,v} f_v \quad \vee \quad f_i = \sum_{v\in\{v_0,\ldots,v_3\}} \lambda_i^{1,v} f_v\right), \forall\; x^i \in R, \tag{3.2}$$

where the f_i and f_v's are variables and the λ's are constants. Unfortunately, a disjunction of this type between two affine subspaces cannot be modeled with standard MIP disjunction modeling techniques unless we can bound the variables (Balas, 1998; Jeroslow and Lowe, 1984). Therefore, we make the additional assumption that $f_v \in [0, U]$, for every vertex $v = (b_1^{k_1}, b_2^{k_2})$ in the grid. In principle, this assumption can be made without loss of generality, although care must be taken in choosing the bounds (that is, choosing the positive constant to add to the y_i's and then choosing the subsequent upper bound $U > 0$.) Note that even if $y_i \geq 0, \forall\; i$, it is quite possible that the optimal value of some f_v could be negative. Similarly, it is possible for the optimal value of some f_v to be greater than $\max_i\{y_i\}$.

Let $\mathcal{R}$ be the set of rectangles defined by the grid breakpoints b. For each $R \in \mathcal{R}$, let $V(R)$ be the four vertices that define it, and let $\mathcal{V}(\mathcal{R}) = \bigcup_{R\in\mathcal{R}} V(R)$. For each $v \in \mathcal{V}(\mathcal{R})$, let $\mathcal{R}_v \subseteq \mathcal{R}$ be the set of rectangles that contain v. Finally, for each i let R_i be a rectangle that contains x^i. Then (2.2) is given by

$$\min \sum_{i=1}^{m} |f_i - y_i|^q \tag{3.3a}$$

$$\text{s.t. } f_i = \sum_{v\in V(R_i)} (\lambda_i^{0,v} f_v^{0,R_i} + \lambda_i^{1,v} f_v^{1,R_i}), \forall\; i = 1,\ldots,m \tag{3.3b}$$

$$f_v^{0,R} \leq U(1 - z_R), \forall\; v \in V(R), \forall\; R \in \mathcal{R} \tag{3.3c}$$

$$f_v^{1,R} \leq U z_R, \forall\; v \in V(R), \forall\; R \in \mathcal{R} \tag{3.3d}$$

$$f_v^{0,R} + f_v^{1,R} = f_v^{0,S} + f_v^{1,S}, \forall\; R, S \in \mathcal{R}_v, \forall\; v \in \mathcal{V}(\mathcal{R}) \tag{3.3e}$$

$$z_R \in \{0, 1\}, \forall\; R \in \mathcal{R} \tag{3.3f}$$

$$f_v^{0,R}, f_v^{1,R} \in [0, U], \forall\; R \in \mathcal{R}_v, \forall\; v \in \mathcal{V}(\mathcal{R}) \tag{3.3g}$$

$$f_i \in [0, U], \forall\; i = 1,\ldots,m. \tag{3.3h}$$

In this model, the binary variable z_R represents the triangulation choice for rectangle R, and the continuous variable $f_v^{k,R}, k = 0, 1$ represents the value $f(v)$ under triangulation k in rectangle R. The variable upper bound constraints (3.3c) and (3.3d) allow exactly one value of $f_v^{k,R}$ per rectangle to be positive, and the constraints (3.3e) make sure values of $f(v)$ match

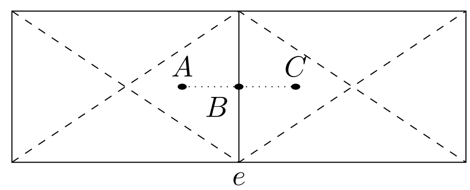


Figure 4: Enforcing convexity across shared edges. The midpoint convexity constraint for points A, B and C concerns the egde e.

from one rectangle to each intersecting one. Constraints (3.3b) along with variable upper bounds (3.3c) and (3.3d) establish disjunction (3.2).

To enforce convexity in this model, we use a generalization of constraints (2.9) and exploit the grid's added structure. We must consider two classes of constraints. The first concerns the edges of the grid $\mathcal{R}$. Let $\mathcal{E}(\mathcal{R})$ be the set of edges shared by two rectangles from $\mathcal{R}$, and for each $e \in \mathcal{E}(\mathcal{R})$ let $R_e, S_e \in \mathcal{R}$ be the unique pair of rectangles that satisfy $R_e \cap S_e = e$. Choose $r^e \in e$ and $\delta^e \in \mathbb{R}^2 \setminus \{0\}$ so that $r^e + \delta^e \in R_e$ lies in the intersection of the two triangles in R_e that contain e, and $r^e - \delta^e \in S_e$ satisfies the analogous condition in S_e. As an example, in Figure 4 we have $B = r^e$, $A = r^e + \delta^e$, and $C = r^e - \delta^e$. Extending the previous section's notation, let $\lambda_e^{k,R_e,v}, \forall\ v \in V(R_e), k = 0, 1$ be the set of convex multipliers for r^e in R_e under triangulation k, and define $\lambda_{e+}^{k,R_e,v}$ and $\lambda_{e-}^{k,S_e,v}$ analogously for $r^e \pm \delta^e$. The constraints are then

$$\frac{1}{2}\Bigg(\sum_{v \in V(R_e)} (\lambda_{e+}^{0,R_e,v} f_v^{0,R_e} + \lambda_{e+}^{1,R_e,v} f_v^{1,R_e}) + \sum_{v \in V(S_e)} (\lambda_{e-}^{0,S_e,v} f_v^{0,S_e} + \lambda_{e-}^{1,S_e,v} f_v^{1,S_e}) \Bigg)$$
$$\geq \sum_{v \in V(R_e)} (\lambda_e^{0,R_e,v} f_v^{0,R_e} + \lambda_e^{1,R_e,v} f_v^{1,R_e}), \forall\ e \in \mathcal{E}(\mathcal{R}). \quad (3.4)$$

The second type of convexity enforcing constraint concerns each individual rectangle $R \in \mathcal{R}$, and in this case convexity can be enforced directly on the f_v variables, without multipliers. Referring again to Figure 3, a triangulation along $v_1 v_2$ means that convexity is enforced if $f_{v_0} + f_{v_3} \geq f_{v_1} + f_{v_2}$, since the function value of the rectangle's center would be the average of f_{v_1} and f_{v_2}. Similarly, the opposite triangulation yields the constraint

$f_{v_1} + f_{v_2} \geq f_{v_0} + f_{v_3}$ (see Figure 5.) Generalizing to (3.3), we get

$$\begin{aligned} f^{0,R}_{v_0^R} + f^{0,R}_{v_3^R} &\geq f^{0,R}_{v_1^R} + f^{0,R}_{v_2^R} \\ f^{1,R}_{v_0^R} + f^{1,R}_{v_3^R} &\leq f^{1,R}_{v_1^R} + f^{1,R}_{v_2^R}, \forall\ R \in \mathcal{R}, \end{aligned} \tag{3.5}$$

where we use the notation $v_0^R, v_1^R, v_2^R, v_3^R$ to identify the four vertices of rectangle R according to Figure 3.

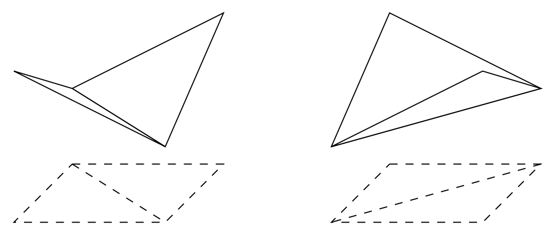


Figure 5: Enforcing convexity under two different triangulations. In the left-hand side triangulation, the sum of the top-right and bottom-left function values must be at least the sum of the top-left and bottom-right function values. The converse holds in the right-hand side triangulation.

Proposition 3.1. *Constraints* (3.4) *and* (3.5) *restrict the feasible region of* (3.3) *to convex functions.*

Proof. Constraints (3.4) enforce midpoint convexity under any of the four triangulation possibilities of R_e and S_e. Note that $\lambda_e^{0,R_e,v} = \lambda_e^{1,R_e,v}, \forall\ v \in V(R_e)$, because r^e only has positive convex multipliers for e's two endpoints, and these multipliers are equal under either triangulation.

For (3.5), only one of the two constraints applies to every R, but the variable upper bounds (3.3c) and (3.3d) force the inactive triangulation's variables to zero, thus satisfying that constraint trivially. □

Because we have fixed $n = 2$, the model (3.3) has $\Theta(|\mathcal{R}|) = \Theta(p^2)$ continuous variables, as many binary variables, and $\Theta(m+|\mathcal{R}|) = \Theta(m+p^2)$ constraints. However, if we were to generalize the model for any n, the constants that multiply all of these quantities would grow exponentially with n.

3.1.1 A Swapping Heuristic

Algorithm 1 presents a *swapping* local search heuristic to find a solution to (3.3). The heuristic is based on a simple swapping idea used, for example, in calculating Delaunay triangulations or convex interpolants (cf. Carnicer and Floater (1996).) In the heuristic, a *swap* of $R \in \mathcal{R}$ simply means replacing

Algorithm 1 Swapping heuristic for (3.3)

```
randomly choose a triangulation of 𝓡
solve (2.8) over this fixed triangulation
while swapping some R ∈ 𝓡 and re-solving improves the objective in
(2.8) do
  update the triangulation by swapping R
end while
return  the last triangulation and the corresponding solution to (2.8)
```

the existing triangulation of R by the opposite one. The triangulation swap changes the convex multipliers that interpolate each data point $x^i \in R$ (i.e. the λ_i values in (3.3),) and thus also changes the objective (3.3a). Since the swaps always improve the objective value, the heuristic is guaranteed to terminate in finitely many steps. Although we do not consider it here, the heuristic can also be generalized to a k-opt local search by simultaneously considering k rectangles for swapping.

3.2 Convex Fitting over Variable Regions

In this section we examine the parametric model that occurs when F is the set of convex piecewise linear functions defined as the maximum of $p \in \mathbb{N}$ affine functions; this problem was first studied by Magnani and Boyd (2009). Each function $f \in F$ has the form

$$f(x) = \max_{k=1,\dots,p} \{c^k x + d_k\}, \tag{3.6}$$

where $(c^k, d_k) \in \mathbb{R}^{n+1}, \forall\, k = 1, \dots, p$. (Figure 6 has an example with $n = 2$.) The partition $\mathcal{P}$ is implicitly defined by f, as each member polyhedron is given by

$$P_k = \{x \in \mathbb{R}^n : c^k x + d_k \geq c^\ell x + d_\ell, \forall\, \ell \neq k\}, \forall\, k = 1, \dots, p. \tag{3.7}$$

Note that not all polyhedra P_k thus defined are bounded, but we can always add bounds if necessary. The fitting problem (2.2) over functions (3.6) yields

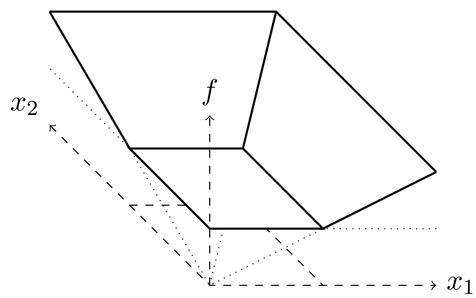


Figure 6: Bivariate convex piecewise linear function $f(x) = \max\{1, x_1, x_2\}$.

the non-convex optimization model

$$\begin{aligned} \min \ & \sum_{i=1}^{m} \Big| \max_{k=1,\ldots,p} \{c^k x^i + d_k\} - y_i \Big|^q \\ \text{s.t. } & c \in \mathbb{R}^{n \times p}, d \in \mathbb{R}^p. \end{aligned} \tag{3.8}$$

Let $x \in \mathbb{R}^n_+$; for given $c^k \in \mathbb{R}^n, d_k \in \mathbb{R}$, we cannot model the non-convex relation (3.6) directly in a MIP. However, we can model $f(x)$ disjunctively as

$$f(x) \geq c^k x + d_k, \forall\ k = 1, \ldots, p \tag{3.9}$$

$$\bigvee_{k=1}^{p} \{f(x) \leq c^k x + d_k\}. \tag{3.10}$$

As in the previous section, the disjunction (3.10) cannot be modeled as stated with MIP disjunctive techniques, because the polyhedra have different recession cones (Balas, 1998; Jeroslow and Lowe, 1984). However, if we let $M > 0$ be an appropriately large number, we can use a big-M approach (see also Bertsimas and Shioda (2007)). Let $z^k \in \{0, 1\}, \forall\ k$ be a binary variable that indicates which affine function is maximal at x. Then (3.10) can be expressed as

$$f(x) \leq c^k x + d_k + M(1 - z^k), \forall\ k = 1, \ldots, p \tag{3.11a}$$

$$\sum_{k=1}^{p} z^k = 1. \tag{3.11b}$$

Let $f_i = f(x^i)$, and define $z_i^k \in \{0,1\}$ analogously to z^k with respect to x^i. Then (2.2) becomes

$$\min \sum_{i=1}^{m} |f_i - y_i|^q \tag{3.12a}$$

$$\text{s.t. } f_i \geq c^k x^i + d_k, \forall\ k = 1, \ldots p, \forall\ i = 1, \ldots, m \tag{3.12b}$$

$$f_i \leq c^k x^i + d_k + M(1 - z_i^k), \forall\ k = 1, \ldots p, \forall\ i = 1, \ldots, m \tag{3.12c}$$

$$\sum_{k=1}^{p} z_i^k = 1, \forall\ i = 1, \ldots, m \tag{3.12d}$$

$$z \in \{0,1\}^{m \times p} \tag{3.12e}$$

$$c \in \mathbb{R}^{n \times p}, \quad d \in \mathbb{R}^p \tag{3.12f}$$

$$f_i \in \mathbb{R}, \forall\ i = 1, \ldots, m. \tag{3.12g}$$

Any permutation of the k-indices of a feasible solution (c, d) to (3.8) would yield another feasible solution of equal objective, because the maximum operator is invariant under a permutation. This leads to substantial symmetry in the formulation (3.12) and potential increases in computation time. The following result addresses this issue.

Proposition 3.2. *There is an optimal solution of* (3.8) *that satisfies the constraints*

$$c_1^1 \leq \cdots \leq c_1^p. \tag{3.13}$$

Proof. Let $(\tilde{c}, \tilde{d})$ be any optimal solution of (3.8). Let $\pi : \{1, \ldots, p\} \to \{1, \ldots, p\}$ be a permutation satisfying $\tilde{c}_1^{\pi(1)} \leq \cdots \leq \tilde{c}_1^{\pi(p)}$. Define

$$c_j^{k*} = \tilde{c}_j^{\pi(k)}, \forall\ j = 1, \ldots, n, \forall\ k = 1, \ldots, p$$
$$d_k^* = \tilde{d}_{\pi(k)}, \forall\ k = 1, \ldots, p.$$

In words, (c^*, d^*) permutes the k-indices of $(\tilde{c}, \tilde{d})$ to order the resulting solution by the first coordinate of the c variables. Since the maximum operator is invariant under permutations, (c^*, d^*) is also optimal for (3.8). □

By adding constraints (3.13) to (3.12), we impose an ordering on the feasible region and remove solution symmetry.

The LP relaxations of mixed-integer models with big-M constraints are notoriously weak. The root cause of the problem is the definition of the maximum operator and the use of constraints (3.11). However, as we have pointed out, the big-M technique is in some way inescapable if we wish to use MIP methodology. We attempt to mitigate the weakness of the LP relaxation with the next set of constraints, for which we first establish a technical result.

Lemma 3.3. *For each $k = 1, \ldots, p$, define the set*

$$S_k = \{f \in \mathbb{R}, g \in \mathbb{R}^p_+ : f = g_k; f \geq g_\ell, \forall\ \ell \neq k\},$$

and define also the set

$$S = \left\{ f \in \mathbb{R}, g \in \mathbb{R}^p_+ : f \leq \sum_{k=1}^{p} g_k; f \geq g_k, \forall\ k = 1, \ldots, p \right\}.$$

Then $S = \operatorname{conv}\big(\bigcup_k S_k\big)$.

Proof. Both S and $S_k, \forall\ k$ are pointed polyhedral cones contained in the positive orthant. Let $e_k \in \mathbb{R}^p$ be the k-th unit vector. For each S_k, a complete set of extreme rays is

$$(f, g) = \left(1, e_k + \sum_{\ell \in T} e_\ell\right), \forall\ T \subseteq \{1, \ldots, p\} \setminus \{k\}.$$

This can be verified by noting that at any extreme ray, exactly one of the pair of constraints $f \geq g_\ell$ and $g_\ell \geq 0$ can be binding for each $\ell \neq k$. (Both are binding only at the origin.)

Similarly, a complete set of extreme rays of S is

$$(f, g) = \left(1, \sum_{k \in T} e_k\right), \forall\ T \subseteq \{1, \ldots, p\}, T \neq \varnothing.$$

As in the previous case, at any extreme ray exactly one of $f \geq g_k$ and $g_k \geq 0$ can be binding for each k. The constraint $f \leq \sum_k g_k$ ensures that at least one of the former is always binding.

The union over all k of the sets of extreme rays of S_k gives the set of rays for S, which proves the result. $\square$

The lemma gives a polyhedral description of the convex hull of a union of polyhedra with differing recession cones (see also Theorem 1 in Queyranne and Wang (1992)). Using $g_k = c^k x^i + d_k$, the next result applies our lemma to (3.12).

Proposition 3.4. *Suppose some optimal solution of* (3.8) *satisfies*

$$c^k x^i + d_k \geq 0, \forall\ k = 1, \dots, p, \forall\ i = 1, \dots, m. \tag{3.14}$$

Then the following constraints are valid for (3.12)*:*

$$f_i \leq \sum_{k=1}^{p} c^k x^i + d_k, \forall\ i = 1, \dots, m \tag{3.15}$$

Constraints (3.14) and (3.15) may be added to (3.12) without loss of generality if we add a large positive constant to each y_i, thus guaranteeing that (3.14) holds for some optimal solution.

Model (3.12) has $\Theta(mp)$ binary variables, $\Theta(m + np)$ continuous variables, and $\Theta(mp)$ constraints. Because of the relatively large number of binary variables, the model may only be computationally tractable for small-to-medium data sets. This computational difficulty would be especially apparent when $q = 2$ and (3.12) becomes a MIP with convex quadratic objective.

3.2.1 Concave Data and Heuristic Performance

Magnani and Boyd (2009) developed a Gauss-Newton heuristic for (3.8). For convenience, we reproduce it here in Algorithm 2 with our notation. The heuristic is fast and often finds a high-quality solution if repeated from several different random initial partitions; the interested reader may consult the same article for details.

Algorithm 2 Clustering Gauss-Newton heuristic for (3.8)

randomly partition $\{x^i\}_{i=1}^m$ into p non-empty sets with pairwise non-intersecting convex hulls
repeat
  for $k = 1, \dots, p$ **do**
    solve linear fitting problem for points in set k to obtain best-fit affine function (c^k, d_k)
  end for
  set $f(x) \leftarrow \max_k\{c^k x + d_k\}$
  partition $\{x^i\}_{i=1}^m$ into p sets according to (3.7), breaking ties arbitrarily
until partition is unchanged
return f

However, as the authors point out, the heuristic may cycle indefinitely even with a tiny data set. In addition, if the heuristic terminates or is cut off after a maximum number of iterations, the resulting function may be far from an optimal fit. Consider the following example: Fix $m \in \mathbb{N}$, and for $i = \pm 1, \dots, \pm m$, define

$$(x^i, y_i) = \begin{cases} (i, i), & i < 0 \\ (i, -i), & i \geq 0. \end{cases}$$

Claim. Let $m \geq 2$, $p = 2$. For any $j \in \{-m+2, \dots, m-1\}$, if we initialize Algorithm 2 by partitioning the points into $\{x^i : i < j\}$ and $\{x^i : i \geq j\}$, it returns $f(x) = \max\{x, -x\}$.

Proof of claim. The proof is by induction on $|j|$; by symmetry we may assume that $j > 0$. If $j = 1$, Algorithm 2 terminates immediately, returning $\max\{-x, x\}$.

Now suppose $j \geq 2$. The best-fit affine function for $\{(x^i, y_i)\}_{i \geq j}$ is $-x$. For $\{(x^i, y_i)\}_{i<j}$, the graph $y = \tilde{c}x + \tilde{d}$ of the best-fit affine function $(\tilde{c}, \tilde{d})$ must intersect the convex hull of the points $\{(x^i, y_i)\}_{i<j}$ as a subset of $\mathbb{R}^2$. By LP duality, this is equivalent to $(\tilde{c}, \tilde{d})$ being a convex combination of the lines defined by facets of $\operatorname{conv}\{(x^i, y_i)\}_{i<j}$: The inequality $-\tilde{c}x + y \leq \max\big\{-\tilde{c}x + y : (x, y) \in \operatorname{conv}\{(x^i, y_i)\}_{i<j}\big\}$ is a convex combination of certain facets, as is the reverse inequality with "min" replacing "max".

For the remainder of the proof, assume $j \geq 3$; the case $j = 2$ is similar. The lines defined by facets of $\operatorname{conv}\{(x^i, y_i)\}_{i<j}$ are

$$(-1, 0), (1, 0), \left(\frac{m - j + 1}{m + j - 1}, -\frac{2m(j-1)}{m + j - 1}\right), (0, -1),$$

with the first coordinate representing slope and the second representing the intercept. It is not hard to show that $(0, -1)$ is in the convex hull of the first three lines (when considered as elements of $\mathbb{R}^2$), so we may omit it from consideration. Let us call the third line $(\hat{c}, \hat{d})$ for brevity.

The best-fit affine function $(\tilde{c}, \tilde{d})$ is a convex combination of $(-1, 0)$, $(1, 0)$ and $(\hat{c}, \hat{d})$; using first-order arguments it can be shown that it must be a *strict* convex combination. Therefore its value at 0 is less than 0 and its value at $(j-1)$ is greater than $(-j+1)$. In particular, the re-partition implied by it and $-x$ decreases j to some value greater than or equal to 1. Thus $|j|$ decreases and the induction hypothesis gives us the result. $\square$

The previous example shows that for any arbitrarily large number of points m and from any non-trivial initial partition, Algorithm 2 converges to a max-affine function that overestimates the value at every single data point in the set, while the best-fit max-affine function is (by inspection) an affine function. We can further generalize the example with the following result.

Proposition 3.5. *Let $n = 1$, and suppose the data set (x^i, y_i) is concave; i.e. there is a concave function $g : \mathbb{R} \to \mathbb{R}$ with $g(x^i) = y_i$. Then the best fit over all convex functions is given by an affine function. In particular, the optimal solution of* (3.8) *is affine, regardless of p.*

Proof. Assume without loss of generality that $x^1 < \cdots < x^m$, and let $\hat{f} : \mathbb{R} \to \mathbb{R}$ be a best-fit convex function for the set. Let $\hat{g} : [x^1, x^m] \to \mathbb{R}$ be the piecewise linear concave function defined by the data set; i.e.

$$\hat{g}(x) = \lambda y_{i-1} + (1 - \lambda) y_i, \quad x^{i-1} \leq x \leq x^i, \quad x = \lambda x^{i-1} + (1 - \lambda) x^i.$$

If $\hat{g}$ is affine, then $\hat{f}$ is affine and there is nothing to prove. Otherwise, we may assume that there is a subinterval $[x', x'']$ of $[x^1, x^m]$ with $x^1 \leq x' < x'' \leq x^m$ such that $\hat{f}(x) \leq \hat{g}(x)$ for $x \in [x', x'']$ and $\hat{f}(x) > \hat{g}(x)$ otherwise. Then we can replace $\hat{f}$ with

$$f^*(x) = \frac{(\hat{f}(x'') - \hat{f}(x'))}{(x'' - x')}(x - x') + \hat{f}(x')$$

without worsening the fit: If $x \in [x', x'']$, then $\hat{f}(x) \leq f^*(x) \leq \hat{g}(x)$. Similarly, if $x \notin [x', x'']$, then $\hat{f}(x) \geq f^*(x) \geq \hat{g}(x)$. □

Proposition 3.5 fails for $n \geq 2$. If each x^i is an extreme point of the set $\{x^i\}_{i=1}^m$, then for any values of y_i (and any m) the data set can be interpolated with either a piecewise linear convex or concave function.

In situations such as those exemplified by the previous proposition, the heuristic may converge to a fitting function that is far from optimal, even if started from many random initial partitions. One distinct advantage of the MIP approach is the guarantee of an optimal fitting function under any circumstance. Of course, if the data is concave, the best-fit max-affine function is likely to be affine (and a poor fit.) In this case, the MIP solution is useful more as a qualitative indicator that an underlying assumption about the data points is incorrect.

3.2.2 Modifying the Model for Separable Fitting

There may be situations in which we desire the best-fit function to be separable. If we divide each dimension's axis into p intervals, the resulting set of

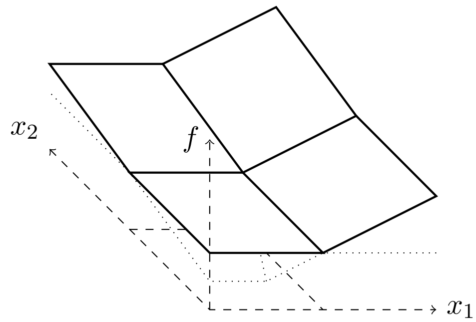


Figure 7: Separable bivariate convex piecewise linear function f defined as the sum of the two univariate max-affine functions $\max\{1, x_1\}$ and $\max\{0, \frac{1}{2}x_2 - \frac{1}{2}\}$. The domain grid implied by the function is $\{[0,1],[1,\infty)\} \times \{[0,1],[1,\infty)\}$.

polytopes $\mathcal{P}$ has $|\mathcal{P}| = p^n$, a number we could never hope to model explicitly in higher dimensions. We next present a variation of (3.12) that fits data with a separable convex function.

Each function $f \in F$ now has the form

$$f(x) = \sum_{j=1}^{n} \max_{k=1,\ldots,p} \{c_j^k x_j + d_j^k\}, \tag{3.16}$$

where $(c^k, d^k) \in \mathbb{R}^{n+n}, \forall\ k = 1, \ldots, p$. See Figure 7 for a two-dimensional example. The generic model (2.2) becomes

$$\begin{aligned} \min \quad & \sum_{i=1}^{m} \left| \sum_{j=1}^{n} \max_{k=1,\ldots,p} \{c_j^k x_j^i + d_j^k\} - y_i \right|^q \\ \text{s.t.} \quad & c \in \mathbb{R}^{n \times p}, \quad d \in \mathbb{R}^{n \times p}. \end{aligned} \tag{3.17}$$

The solution to (3.17) finds the optimal grid over which to define the separable best-fit convex function, in addition to finding the function itself.

Applying the same modeling techniques used in the non-separable model, the problem can be written as

$$\min \quad \sum_{i=1}^{m} \left| \sum_{j=1}^{n} f_i^j - y_i \right|^q \tag{3.18a}$$

$$\text{s.t.} \ f_i^j \geq c_j^k x_j^i + d_j^k, \forall\ k = 1, \ldots, p, \forall\ j = 1, \ldots, n, \forall\ i = 1, \ldots, m \tag{3.18b}$$

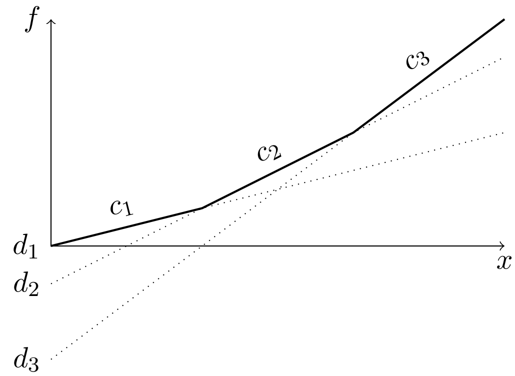


Figure 8: An ordering of the univariate affine functions. Starting from the origin, the first function has the smallest slope and largest intercept, with subsequent functions following the orderings (3.18e) and (3.18f).

$$f_i^j \le c_j^k x_j^k + d_j^k + M(1 - z_{ij}^k), \forall\ k = 1, \dots, p, \quad \forall\ j = 1, \dots, n, \forall\ i = 1, \dots, m \tag{3.18c}$$

$$\sum_{k=1}^{p} z_{ij}^k = 1, \forall\ j = 1, \dots, n, \forall\ i = 1, \dots, m \tag{3.18d}$$

$$c_j^1 \le \dots \le c_j^p, \forall\ j = 1, \dots, n \tag{3.18e}$$

$$d_j^1 \ge \dots \ge d_j^p, \forall\ j = 1, \dots, n \tag{3.18f}$$

$$z \in \{0, 1\}^{m \times n \times p} \tag{3.18g}$$

$$c \in \mathbb{R}^{n \times p}, \quad d \in \mathbb{R}^{n \times p} \tag{3.18h}$$

$$f_i^j \in \mathbb{R}, \forall\ j = 1, \dots, n, \forall\ i = 1, \dots, m. \tag{3.18i}$$

Here, $f_i^j = f_j(x_j^i) = \max_k\{c_j^k x_j^i + d_j^k\}$ and $z_{ij}^k \in \{0, 1\}$ is defined with respect to f_i^j in an analogous fashion to z_i^k in (3.12).

We can use reasoning analogous to Proposition 3.2 to show that constraints (3.18e) are valid for at least one optimal solution. In addition, because the functions are univariate, those constraints also allow us to add constraints (3.18f) (see Figure 8.)

The following result shows how to further restrict the feasible region.

Proposition 3.6. *There is an optimal solution of* (3.17) *that satisfies*

$$d_j^1 = 0, \forall\ j \ge 2. \tag{3.19}$$

Proof. Let $(\tilde{c}, \tilde{d})$ be an optimal solution to (3.17). Define

$$c_j^{k*} = \tilde{c}_j^k, \forall\ k = 1, \ldots, p, \forall\ j = 1, \ldots, n$$
$$d_j^{k*} = \begin{cases} \tilde{d}_j^k + \sum_{j'=2}^n \tilde{d}_{j'}^1, & j = 1 \\ \tilde{d}_j^k - \tilde{d}_j^1, & j \geq 2 \end{cases}, \forall\ k = 1, \ldots, p.$$

The solution (c^*, d^*) shifts the affine functions in the first dimension up by $\sum_{j' \geq 2} \tilde{d}_{j'}^1$, and shifts each affine function in dimension $j \geq 2$ down by $\tilde{d}_j^1$, satisfying (3.19). Let $x \in \mathbb{R}^n$; then

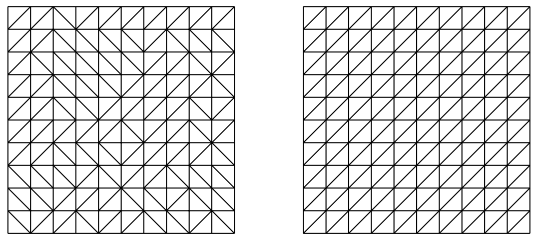
$$\sum_{j=1}^n \max_{k=1,\ldots,p} \{c_j^{k*} x_j + d_j^{k*}\} =$$
$$\max_{k=1,\ldots,p} \left\{ \tilde{c}_1^k x_1 + \tilde{d}_1^k + \sum_{j=2}^n \tilde{d}_j^1 \right\} + \sum_{j=2}^n \max_{k=1,\ldots,p} \{\tilde{c}_j^k x_j + \tilde{d}_j^k - \tilde{d}_j^1\} =$$
$$\max_{k=1,\ldots,p} \{\tilde{c}_1^k x_1 + \tilde{d}_1^k\} + \sum_{j=2}^n \tilde{d}_j^1 + \sum_{j=2}^n \big(\max_{k=1,\ldots,p} \{\tilde{c}_j^k x_j + \tilde{d}_j^k\} - \tilde{d}_j^1 \big) =$$
$$\sum_{j=1}^n \max_{k=1,\ldots,p} \{\tilde{c}_j^k x_j + \tilde{d}_j^k\},$$

and therefore (c^*, d^*) is also optimal. □

The separable model increases the number of binary variables by an order of magnitude to $\Theta(mnp)$ and the number of continuous variables by a comparable amount to $\Theta(mn+np)$. The number of constraints also increases to $\Theta(mnp)$. This substantial increase in the model's size may again restrict the realistic size of data sets we can hope to optimally fit with current optimization technology. An option available exclusively in the separable case is to partially solve (3.18) to obtain a feasible solution, and then fix the grid and solve the resulting continuous model.

4 Computational Examples

We next present computational examples to highlight the benefit of considering the fitting models over variable regions introduced in Section 3. All fittings were performed with a least-squares objective ($q = 2$ in (2.2).) The fitting models were optimized using CPLEX 11.1 on a Xeon 2.66 GHz workstation with 8 Gb of RAM.



(a) (b)

Figure 9: Random and optimal domain triangulations for cubic function grid fitting experiment.

4.1 Grid Fitting with Variable Triangulations

4.1.1 Cubic Example

To test our grid fitting model, we generated a sample of of the function $f(x) = (x_1 - \frac{1}{2}x_2)^3$ over the domain $[0, 10]^2$. We defined the grid points as $b_j^k = k, \forall\ k = 0, \dots, 10, \forall\ j = 1, 2$, and sampled ten points from each grid square's uniform distribution. For each point x^i, we set $y_i = (x_1 - \frac{1}{2}x_2)^3 + 125$, and defined the upper bound $U = \max\{f(x) + 125 : x \in [0, 10]^2\} = 1125$. We also generated a random triangulation of the domain to fit the function initially; see Figures 9a and 10a.

Using the initial random triangulation in (2.8), we obtained a best-fit total squared error of 1538.12. We then ran the swapping heuristic outlined in Algorithm 1 starting from the randomly generated triangulation to obtain the triangulation shown in Figures 9b and 10b, yielding a total squared error of 423.57. Using (3.3) with the heuristic solution as a warm-start, we were able to prove the optimality of this triangulation in under 10 seconds.

By optimizing the fitting over all possible triangulations, we were able to exploit the underlying structure of our data without any prior knowledge. In this example, we find an optimal triangulation that clearly reflects the invariance of the underlying cubic function along the line $2x_1 = x_2$, and can therefore decrease the squared error by over 70%.

4.1.2 Sinusoidal Example

We next tested the model on a different function with more complicated sinusoidal structure. Using the same domain and grid from the previous

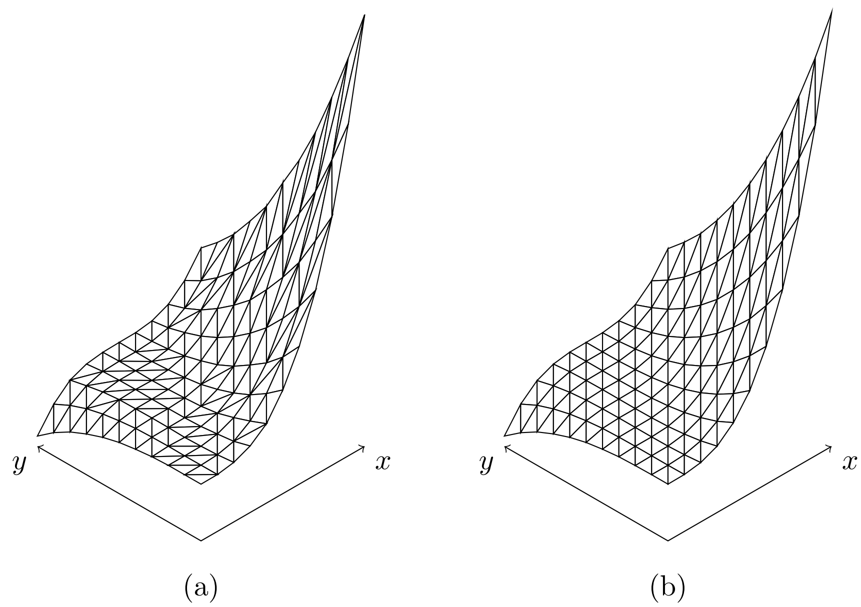


Figure 10: Best-fit function plots for random and optimal triangulations in cubic grid fitting experiment.

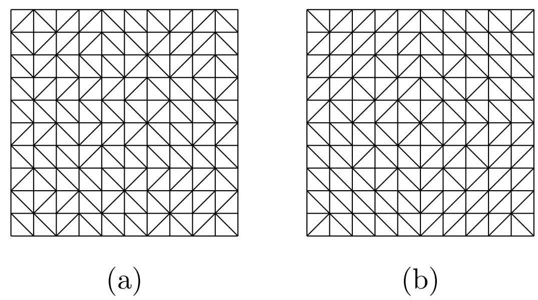


Figure 11: Random and optimal domain triangulations for sinusoidal function grid fitting experiment.

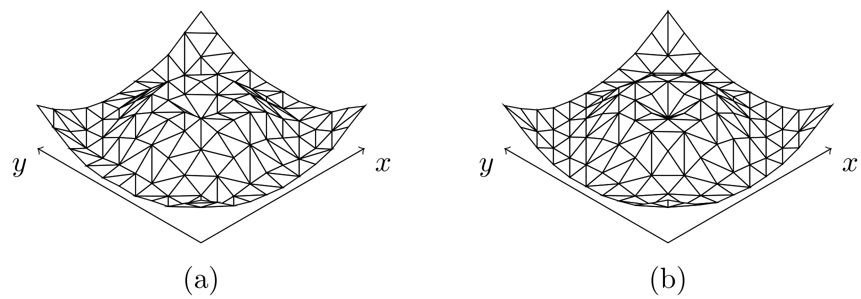


Figure 12: Best-fit function plots for random and optimal triangulations in sinusoidal grid fitting experiment.

example, we generated 1000 points for the function $f(x) = \sin(\|x-(5,5)\|_2)$. For each point x^i, we set $y_i = f(x)+1.5$ and defined the upper bound $U = 3$. We also generated another random triangulation for the initial fitting, shown in Figures 11a and 12a.

We solved model (2.8) with this triangulation to obtain a total squared error of 1.26. We then ran Algorithm 1, producing the triangulation in Figures 11b and 12b, with squared error of 0.42, after the second outer iteration. Using (3.3) with this solution as a warm-start, we proved optimality in approximately 4.5 minutes. As before, the optimization over all triangulations allowed us to discover the radially symmetric structure of the data around the point $(5, 5)$ without requiring prior knowledge, and enabled us to decrease the initial squared error by two thirds.

4.2 Convex Piecewise Linear Fitting

We next present an example for the convex non-separable model (3.12). To test this model, we generated a set of 300 data points on the domain $[0, 10]^2$ as in the previous section, by generating three points in each unit square from the square's uniform distribution. In this case, we used the convex function $f(x) = \ln(e^{x_1} + e^{2x_2})$, setting $y_i = f(x^i), \forall\ i$. As in the previous experiment, we generated a feasible solution to the model using Algorithm 2's heuristic and then validated the solution's quality with our MIP model.

We first solved the model with $p = 3$. From the first random partition of the data points, the heuristic found the best-fit function given by

$$
\begin{aligned}
(c^1, d_1) &= (0, 1.99, 0.05) \\
(c^2, d_2) &= (0.37, 1.25, 0.73) \\
(c^3, d_3) &= (0.97, 0.04, 0.18),
\end{aligned}
$$

with total squared error 0.26. We were then able to prove optimality with (3.12) in under two minutes. Figure 13 illustrates how the best-fit function's implicit partition of the domain reflects the underlying function's curvature. For comparison, the squared error when fitting the data points in the classical least-squares linear fitting model is 479.14. This drastic decrease in fitting error indicates the potential benefit of variable-region fitting when compared to a more classical approach.

However, the model's difficulty increases substantially with the problem's size. For example, we attempted to solve the same fitting problem with the same data points and $p = 4$. After the fifth iteration (starting from a random partition of the data points,) the heuristic found a best-fit solution with total squared error equal to 0.11, a marginal improvement from the best-fit function found above for $p = 3$. After one hour of computation time, CPLEX was only able to decrease the optimality gap to 84%. We conducted a similar experiment with $p = 5$, and CPLEX was not able to improve the lower bound from the trivial 0 after one hour. (The heuristic solution's squared error was 0.07.)

Part of the difficulty may come from the underlying function's structure, which lends itself to fitting with three affine functions. However, part of the difficulty must also come from model (3.12)'s own limitations. Even with the addition of constraints (3.15), we did not strengthen the LP relaxation enough. It seems that an entirely new and different modeling technique may be needed for (3.10).

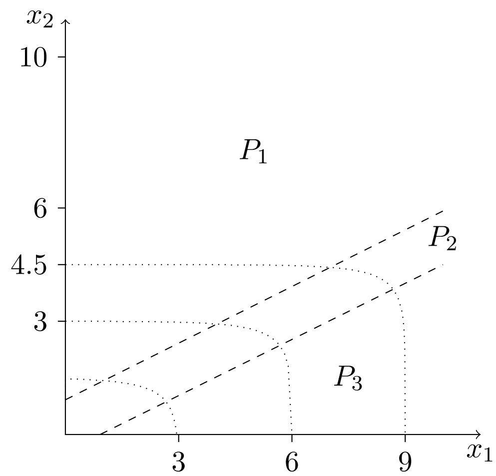


Figure 13: Level curves of original function for $f(x) = 3, 6, 9$ (dotted curves) and partition implied by best-fit max-affine function (dashed lines).

5 Conclusions

We have introduced models for continuous piecewise linear data fitting that

i) study piecewise linear data fitting from a linear and discrete optimization perspective, and

ii) introduce mixed-binary techniques to solve models with region variability.

Generally, one of the main benefits of using optimization to solve a problem is the straightforward fashion in which additional constraints can be added without a change in methodology. A specific example from our current work is the introduction of convexity-enforcing constraints in each of our models. However, many other constraints are possible and may be desirable depending on the fitting problem's application. This flexibility, epitomized by the region variability of our mixed-binary models, is a distinct advantage of the optimization approach.

Our work also introduces various questions. Chief among them is the scalability of our models for large data sets, which are important in many applications. Bertsimas and Shioda (2007) circumvent this issue by employing clustering heuristics to reduce their original data set to a manageable size. Jiang and Klabjan (2009) use a branch-and-price algorithm for the clusterwise regression problem, which is related to (2.1). These and other techniques may allow our models to accommodate large data sets and thus increase their application value.

Another important issue concerns MINLP. As we mention in Section 1, the optimization community has lately focused on MINLP methodology. In the least-squares case ($q = 2$ in (2.2)), our models have a convex-quadratic objective, and therefore the mixed-binary models encompass a novel and potentially important class of convex MINLP problems for researchers to study, both to generate good solutions and also to strengthen lower bounds (Bienstock, 2010).

Acknowledgements

The authors would like to thank George Nemhauser for many helpful comments on earlier versions of this paper. A. Toriello gratefully acknowledges support from the National Science Foundation, via a Graduate Research Fellowship, and from the ARCS Foundation.

References

Abhishek, K., Leyffer, S., and Linderoth, J. (2010). FilMINT: An Outer Approximation-Based Solver for Convex Mixed-Integer Nonlinear Programs. *INFORMS Journal on Computing*, 22:555–567.

Balas, E. (1998). Disjunctive programming: Properties of the convex hull of feasible points. *Discrete Applied Mathematics*, 89:3–44.

Belotti, P., Lee, J., Liberti, L., Margot, F., and Wächter, A. (2009). Branching and bounds tightening techniques for non-convex MINLP. *Optimization Methods and Software*, 24:597–634.

Bertsekas, D. and Tsitsiklis, J. (1996). *Neuro-Dynamic Programming.* Athena Scientific.

Bertsimas, D. and Shioda, R. (2007). Classification and Regression via Integer Optimization. *Operations Research*, 55:252–271.

Bienstock, D. (2010). Eigenvalue Techniques for Convex Objective, Nonconvex Optimization Problems. In Eisenbrand, F. and Shepherd, F., editors, *Proceedings of the 14th International Conference on Integer Programming and Combinatorial Optimization, Lausanne, Switzerland, June*

9–11, 2010, volume 6080 of *Lecture Notes in Computer Science*, pages 29–42. Springer.

Bonami, P., Biegler, L., Conn, A., Cornuéjols, G., Grossmann, I., Laird, C., Lee, J., Lodi, A., Margot, F., Sawaya, N., and Wächter, A. (2008). An Algorithmic Framework for Convex Mixed Integer Nonlinear Programs. *Discrete Optimization*, 5:186–204.

Bot, R. and Lorenz, N. (2011). Optimization problems in statistical learning: Duality and optimality conditions. *European Journal of Operational Research*, 213:395–404.

Boyd, S. and Vandenberghe, L. (2004). *Convex Optimization*. Cambridge University Press.

Carnicer, J. and Floater, M. (1996). Piecewise linear interpolants to Lagrange and Hermite convex scattered data. *Numerical Algorithms*, 13:345–364.

de Farias, Jr., I., Zhao, M., and Zhao, H. (2008). A Special Ordered Set Approach for Optimizing a Discontinuous Separable Piecewise Linear Function. *Operations Research Letters*, 36:234–238.

Ferrari-Trecate, G., Muselli, M., Liberati, D., and Morari, M. (2001). A learning algorithm for piecewise linear regression. In Marinaro, M. and Tagliaferri, R., editors, *Neural Nets: WIRN VIETRI-01, 12th Italian Workshop on Neural Nets*. Springer.

Geißler, B., Martin, A., Morsi, A., and Schewe, L. (2011). Using Piecewise Linear Functions for Solving MINLPs. To appear in the IMA Volume on MINLP.

Gonin, R. and Money, A. (1989). *Nonlinear L_p Norm Estimation*. CRC Press.

Holmes, C. and Mallick, B. (1999). Bayesian regression with multivariate linear splines. *Journal of the Royal Statistical Society, Series B*, 63:3–17.

Jeroslow, R. and Lowe, J. (1984). Modeling with integer variables. *Mathematical Programming Study*, 22:167–184.

Jiang, Y. and Klabjan, D. (2009). A Branch-and-price Algorithm for Clusterwise Linear Regression. Presentation at the *20th International Symposium on Mathematical Programming*.

Lau, K.-N., Leung, P.-L., and Tse, K.-K. (1999). A mathematical programming approach to clusterwise regression model and its extensions. *European Journal of Operational Research*, 116:640–652.

Leyffer, S., Sartenaer, A., and Wanufelle, E. (2008). Branch-and-Refine for Mixed-Integer Nonconvex Global Optimization. Technical report, Mathematics and Computer Science Division, Argonne National Laboratory. Preprint ANL/MCS-P1547-0908.

Magnani, A. and Boyd, S. (2009). Convex Piecewise-Linear Fitting. *Optimization and Engineering*, 10:1–17.

Nemhauser, G. and Wolsey, L. (1999). *Integer and Combinatorial Optimization*. John Wiley & Sons, Inc.

Novoa, C. and Storer, R. (2009). An approximate dynamic programming approach for the vehicle routing problem with stochastic demands. *European Journal of Operational Research*, 196:509–515.

Papadaki, K. and Powell, W. (2002). Exploiting structure in adaptive dynamic programming algorithms for a stochastic batch service problem. *European Journal of Operational Research*, 142:108–127.

Pardalos, P. and Kundakcioglu, O. (2009). Classification via Mathematical Programming (Survey). *Applied and Computational Mathematics*, 8:23–35.

Pottmann, H., Krasauskas, R., Hamann, B., Joy, K., and Seibold, W. (2000). On Piecewise Linear Approximation of Quadratic Functions. *Journal for Geometry and Graphics*, 4:31–53.

Powell, W. (2007). *Approximate Dynamic Programming: Solving the curses of dimensionality*. John Wiley & Sons, Inc.

Queyranne, M. and Wang, Y. (1992). On the convex hull of feasible solutions to certain combinatorial problems. *Operations Research Letters*, 11:1–11.

Rockafellar, R. (1970). *Convex Analysis*. Princeton University Press.

Ruppert, D., Wand, M., and Carroll, R. (2003). *Semiparametric Regression*. Cambridge University Press.

Strikholm, B. (2006). Determining the number of breaks in a piecewise linear regression model. Technical report, Department of Economic Statistics and Decision Support, Stockholm School of Economics. SSE/EFI Working Paper Series in Economics and Finance, No. 648.

Toriello, A., Nemhauser, G., and Savelsbergh, M. (2010). Decomposing inventory routing problems with approximate value functions. *Naval Research Logistics*, 57:718–727.

Vielma, J., Ahmed, S., and Nemhauser, G. (2010). Mixed-Integer Models for Nonseparable Piecewise Linear Optimization: Unifying Framework and Extensions. *Operations Research*, 58:303–315.

Vielma, J., Keha, A., and Nemhauser, G. (2008). Nonconvex, lower semicontinuous piecewise linear optimization. *Discrete Optimization*, 5:467–488.

Williams, H. (2007). *Model Building in Mathematical Programming*. John Wiley & Sons, Ltd., fourth edition.

Wilson, D. (1998). *Polyhedral methods for piecewise-linear functions*. PhD thesis, University of Kentucky.