IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received December 26, 2018, accepted February 26, 2019, date of publication March 4, 2019, date of current version March 20, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2902620

Piecewise Linear Regression Based on Plane Clustering

XUBING YANG[1], HONGXIN YANG[1], FUQUAN ZHANG[1], LI ZHANG[1], XIJIAN FAN[1], QIAOLIN YE[1], (Member, IEEE), AND LIYONG FU[2]

[1]College of Information Science and Technology, Nanjing Forestry University, Nanjing 210037, China
[2]Institute of Forest Resource Information Techniques, Chinese Academy of Forestry, Beijing 100091, China

Corresponding author: Liyong Fu (fuliyong840909@163.com)

This work was supported in part by the Central Public-Interest Scientific Institution Basal Research Fund under Grant CAFYBB2019QD003, in part by the Natural Science Foundation of China under Grant 31670554 and Grant 61871444, and in part by the Jiangsu Science Foundation under Grant BK20161527 and Grant BK20171453.

ABSTRACT Piecewise linear regressions have shown many successful applications in image denoising, signal process, and data mining fields. In essence, they attempt to seek multiple linear functions (piecewise/stepwise function) to fit the given scatter data points by various methodologies, typically point-centered clustering methods, such as *k*-means or fuzzy c means. Obviously, it is reasonable that plane-centered clustering is more suitable for capturing the linearities in data. In this paper, we propose an efficient piecewise linear regression method based on *k*-plane clustering, termed as PlrPC. The proposed method first partitions the data into multiple plane-centered clusters and then analytically compute corresponding piecewise linear functions. Compared with the state-to-the-art linear regressors, the advantages of the PlrPC lie in fourfold: 1) it is generated from plane clustering, which is truly coincident with geometrical intuition; 2) to fuse the linear characteristics into plane clustering, a new implicit regression method is proposed; 3) a new plane jump method is proposed to detect the number of clusters, and; 4) the leading problem can be solved by ordinary eigenvalue problems. The experimental results will show the aforesaid characters on some artificial and some benchmark datasets.

INDEX TERMS Piecewise linear regression, minimum square error, optimization, closed-form solution.

I. INTRODUCTION

Regression analysis is a powerful tool in statistical process and data mining for estimating the relationships among variables, and usually be characterized by an input-output map (also called regressor) between a dependent variable and one or more independent variables. It is useful for people to explore and understand the latent information from the map. By the generated functional descriptor, people can accurately predict the output variable from the relevant input variables without further considering complicated inner mechanism [1]. Another scenario for regression is data denoising [2], [3], which is under the assumption that the data observations sampled from different distribution satisfy with local similarity condition. Due to the various learning tasks, there are many regression methods, including MSE (Minimum Square Error) linear regression, ridge regression, support vector regression, artificial neural network, kernel-based regression, piecewise/stepwise regression, etc. Especially in recent years, regression analysis has achieved great success in many fields, such as consumption forecasting [4], chemical component investigation [5], [6], Remote-sensing data analysis [7], wastewater process [8], Hybrid Magnetic research [9], image classification [10], solder Remaining-life prediction [11], highway vehicle classification [12]. In this paper, we only focus on piecewise linear regressions.

As nonlinear approximators, piecewise functions are frequently studied in the literatures. It is well known that a nonlinear function can be approximated by a series of linear segments/planes that follow the gradient of the function. However, such approximations mainly confront two challenges: domain selection and the number of partitions. Herein domain selection means how to partition the dataset into several clusters/groups, and the number of partitions equals to the number of clusters. The previous work for this topic can go back to classic point-centered clustering

The associate editor coordinating the review of this manuscript and approving it for publication was Bora Onat.

2169-3536 © 2019 IEEE. Translations and content mining are permitted for academic research only.
Personal use is also permitted, but republication/redistribution requires IEEE permission.
See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

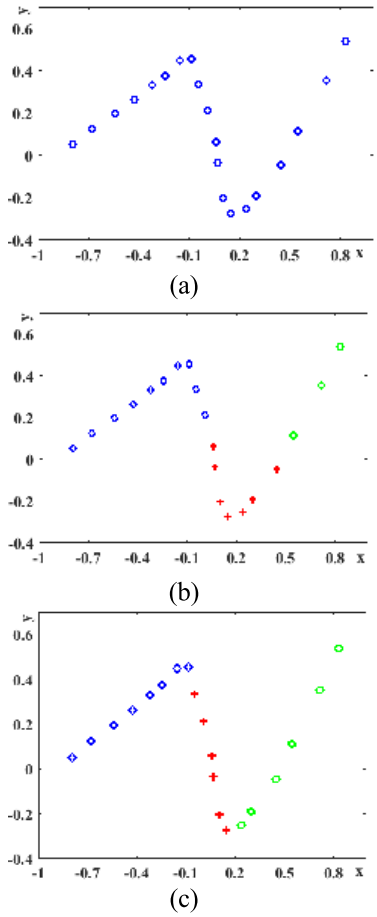


FIGURE 1. Illustration for 3 segmented clusters: *k*-means vs. *k*PC, (a) original data distribution, (b) *k*-means, and (c) *k*PC clustering.

algorithms [13], [14], such as k-means [15] and FCM (fuzzy c-means) [16], [17]. After data clustering, people continue to do domain selection or execute linear regression for each cluster. Once combined with concrete applications, there are a large number of regression technologies. Here we only named a few. Some related work will be presented in Section II A. Instead of the point-prototype of the foresaid point-centered clustering, for instance, k-means, k plane clustering (kPC) [18] takes the planes as their corresponding cluster centers. In doing so, kPC aims to capture the linearities in data. Figure 1 illustrates an example with three segmented scatter points, whose distribution like the slanted letter ''Z'' (Fig. 1a). Set the number of clusters, $k = 3$, we run k-means (Fig. 1b) and kPC (Fig. 1c) respectively and mark cluster results with colorful ''o'', ''+'' and diamond.

Fig. 1 shows that k-means easily ignores such linearity and clusters the points, especially located at corner area, to the wrong cluster, while kPC able to capture underlying local linearities in data, and its clustering results are coincident with original distribution.

Our motivation originally generates from k-plane clustering which is more powerful for capturing linearity in data. In this paper, we propose a new Piecewise linear regression under the guidance of Plane Clustering algorithm, termed as PlrPC. No free Lunch (NFL) theorem says that sufficiently making use of prior information is one of the most effective ways to promote learning machine's performance. By replacing the point prototypes with so-called plane prototypes as cluster centers, we define a new optimization objective to meet the foresaid problem.

We highlight the contributions of this paper.

1). Similar to linear regression methods, the PlrPC has clear geometrical interpretation. We will illustrate this work on several small-scale synthetic datasets.

2). The leading problem is a non-successive quadratic optimization, which can be decomposed into two sub models. One corresponds to our proposed implicit target-variable regression, which can be solved by ordinary eigenvalue problem, thus linear function can be uniquely determined by the corresponding covariance. The other, i.e. the number of piecewise linear functions, will be determined by our proposed plane jump method [19].

3). Instead of point-to-point distance in point-centered clustering, PlrPC adapts point-to-plane distance to design optimization objective, which is more suitable for capturing linearities in data.

The rest of paper is organized as follows. In Section 2, we briefly review some preliminary work. The PlrPC will be detailed in Section 3, including optimization model, geometrical interpretation, and solution. In Section 4, we provide the experimental results on some artificial and public datasets. Finally, we conclude the whole paper in Section 5.

II. PRELIMINARIES AND NOTATIONS

In this section, we first review some related work about piecewise linear regressions.

A. RELATED WORK

As aforementioned, piecewise linear regressions have been widely studied in literatures in last decade. Similar to k-means clustering based piecewise regressions, a convex piecewise-linear fitting method [20] was proposed with the fixed number of clusters. Moreover, it sometimes suffers from matrix singularity during solving the leading problem. Following the convex optimization route, CAP (convex adaptive partitioning) [21] also creates a globally convex regression model and estimates the number of partitions by cross validation. However, it is hard to determine two tuning parameters: the number of knots and the size of minimal subset, which are sensitive to regression loss and computational time. Instead of Euclidean distance, Shao *et al.* [24] introduced a new metric into super resolution and proposed a piecewise linear regression named HHCR (Half Hypersphere Confinement Regression). It is still based on the point-centered method (named anchored neighborhood regression) in the half hypersphere space which is transformed from input space by MDS (multidimensional scaling).

Another branch abandons clustering analysis, and directly estimates break-points or interval fields. Yang *et al.* [25]

investigated statistics F-test and p-value to check the partition of experimental adsorption data. The method largely depended on the graphical plot and scarcely spread to high dimension cases. Malash and El-Khaiary [26] proposed a threshold selection to determine break points by maximizing the adjusted coefficient of determination. It maybe works well for only one independent variable (one-dimension information). Ahmed and Ramadan [27] theoretically constructed an alternative sequential procedure to estimate the number of breaks. Conclusions hold under a series of statistic assumption, such as parameter consistent estimation, well-defined moment matrix, and infinite tendency. To estimate rotor position of switch reluctance machine, Strikholm [28] advised a method for selecting the optimal interval. In fact, it is a two-region data fitting method, and almost impossibly extended to multiple regions. OPLRA (Optimal Piecewise Linear Regression Analysis) [29] able to simultaneously identify the partition feature/variable and the number of regions under linear minimum optimization objective with bilinear constraints. It needs extra heuristic procedure to find the number of break points, which need to repeatedly solve optimization problems. Yang *et al.* [30] estimated breaks in linear model with band spectral regression. Owing to conclusions mainly come from statistical significance, people do not know the performance for the case of limit samples. In ICML 2016, Acharya and partners [31] proposed a piecewise linear regression named GreedyMerging which has two tuning parameters. While facing real application, the authors failed to give more advice for parameter selection.

As aforementioned, quite different from point-centered clusters, plane-centered clustering methods [18], [23]–[24], such as kPC and its variants [32]–[33], aim to seek k planes to assign data points into k clusters according to the point-to-plane distance. This will be described by mathematical programming problem in next section.

B. kPC: k-PLANE CLUSTERING

Assume a given data set $\{\boldsymbol{x}_i|\boldsymbol{x}_i \in R^d\}_{i=1}^n$ represented by a matrix $\boldsymbol{A}(\in R^{n\times d})$, where the ith line of $\boldsymbol{A}$, $\boldsymbol{A}_i$, corresponds to the point $\boldsymbol{x}_i$, and R^d denotes d-dimensional real space.

Define k planes in R^d: $\{\boldsymbol{x}|\boldsymbol{x} \in R^d, \boldsymbol{w}_l^T\boldsymbol{x} + b_l = 0\}$, $l = 1, 2, \cdots, k$, where the parameter pair $(\boldsymbol{w}_l, b_l)$ denotes the normal vector and threshold of the lth plane, respectively. The superscript "T" denotes matrix transpose operator. Training kPC is alternatively running two steps: "point assignment" and "plane update". "Point assignment" means assigning each point to the cluster corresponding to its closest plane, while "plane update" means updating k planes by minimizing the sum of the squares of distances between planes and points in their corresponding clusters.

C. DETERMINING THE NUMBER OF CLUSTERS k

To determine the "true" number of groups in a data set is one of the most difficult problems in cluster analysis. The proposals are Gaussian model-based approach and the gap statistic by comparing the change in within-cluster dispersion like variances, mean square error, etc. [34]. Theoretically and empirically, "Sharp jump method" [19], derived from Shannon mutual information theoretic ideas, would be suitable for piecewise problems. However, the jump method starts from k-means algorithm (details see Fig. 1) and its transformation power Y should be early determined though the authors gave a default value with $Y = d/2$, where d denotes samples dimension. Next, we will improve this method to suit our PlrPC, and named it as "plane jump method".

III. PIECEWISE LINEAR REGRESSION BASED ON PLANE CLUSTER

From figure 1, we know that, for k-means, the points are misled into wrong clusters, and at least the points in two clusters are not consistent with original line-shaped distribution (marked blue diamond and red "+", see Fig. 1b). While kPC is capable of capturing linearity and clusters the points into the three line-shaped clusters (Fig. 1c). The example clearly illustrates superiority of the plane clustering, especially for the data sampled from plane-shaped distribution (also named subspace distribution [35]).

To meet regression tasks, we redefine n points and their responses as $\{(\boldsymbol{x}_i, y_i) \in R^d \times R\}_{i=1}^n$, where the symbol $\boldsymbol{x}$ is a d-dimensional dependent (input) variable, and y is its corresponding target variable (response/independent variable). Define a fixed but currently unknown map: $f : R^d \to R$ from hypothesis set F of candidate functions. By merging the target variable into the independent variables, we denote it as $\{\tilde{\boldsymbol{x}}_i|\tilde{\boldsymbol{x}}_i \in R^{d+1}\}_{i=1}^n$, where $\tilde{\boldsymbol{x}}_i^T = [\boldsymbol{x}_i^T, y_i]$. Thus each $\tilde{\boldsymbol{x}}$ can be viewed as a point in the $(d+1)$ dimensional real space.

A. OPTIMIZATION PROBLEM

As foresaid in Section II B, kPC attempts to seek k planes (presented as linear equations) to partition the given n points into k clusters. By the Euclidian point-to-plane distance, we obtain the optimization problem as

$$\min J(\tilde{\boldsymbol{W}}, \boldsymbol{b}, k) = \sum_{i=1}^{k}\sum_{j=1}^{n_i} \frac{|\tilde{\boldsymbol{w}}_i^T\tilde{\boldsymbol{x}}_j^{(i)} + b_i|}{||\tilde{\boldsymbol{w}}_i||} + \lambda k \qquad (1)$$

where $\{\boldsymbol{x}|\tilde{\boldsymbol{w}}_i^T\boldsymbol{x} + b_i = 0\}$ denotes the ith plane, $\tilde{\boldsymbol{w}}_i(\in R^{d+1})$ and $b_i(\in R)$ denotes normal vector and bias, respectively. $\tilde{\boldsymbol{x}}_j^{(i)}$ denotes the jth sample in the ith cluster whose cardinality is n_i. $\lambda(> 0)$ is a regularization parameter. The first term in (1), $\sum_{j=1}^{n_i} \frac{|\tilde{\boldsymbol{w}}_i^T\tilde{\boldsymbol{x}}_j^{(i)}+b_i|}{||\tilde{\boldsymbol{w}}_i||}$, is total sum of distances between the points in the ith clusters to its corresponding fitting plane $\{\boldsymbol{x}|\tilde{\boldsymbol{w}}_i^T\boldsymbol{x} + b_i = 0\}$. Define $\tilde{\boldsymbol{W}} = [\tilde{\boldsymbol{w}}_1, \tilde{\boldsymbol{w}}_2, \cdots, \tilde{\boldsymbol{w}}_k]$ and $\boldsymbol{b} = [b_1, b_2, \cdots, b_k]^T$.

Obviously, when k increases, the value of the first term will decrease, and vice versa. The objective function is capable for minimizing the distance sum to fitting data and simultaneously controlling the number of clusters k. However, it is hard to solve because there exist nonsuccessive nondifferentiable

terms. Next we will divide and conquer it. Firstly, we consider linear regression problem in next subsection with a fixed k.

B. PIECESIZE LINEAR REGRESSION

With a fixed the number of clusters k, rewrite first term of (1) as,

$$\min\ J(\tilde{\boldsymbol{W}}, \boldsymbol{b}) = \sum_{i=1}^{k}\sum_{j=1}^{n_i}(\tilde{\boldsymbol{w}}_i^T\tilde{\boldsymbol{x}}_j^{(i)} + b_i)^2$$
$$s.t.\ ||\tilde{\boldsymbol{w}}_i||^2 = 1, \quad i = 1 \sim k \tag{2}$$

The advantages of the optimization problem of (2) lie in two-fold: 1) inheriting geometrical interpretation of the original objective function in (1); and 2) restraining singularity of the plane norm vector when the objective value decreases. We will discuss its solution described as following theorems.

Theorem 1: The problem (2) is a convex problem.

Theorem 2: The problem (2) can be individually solved by k ordinary eigenvalue problems.

For proofs, see Appendix. Theorems 1 and 2 say that fitting planes can be solved by k ordinary eigenvalue problems, thus we have the following corollaries.

Corollary 1: The optimum of problem (2) reaches at minimal sum of the k minimal eigenvalues of the k eigenvalue problems.

Corollary 2: When the objective function in (2) reaches the optimum, the optimal solution will be uniquely determined by k covariance matrices and means of the corresponding k clusters.[1]

For the proofs of Corollary 1 and 2, please see Appendix. From the proof of corollary 1 and [32, Th. 2.1], the solution of the problem (2) can be stationary at and only at the eigenvectors of the ordinary eigenvalue problems. Corollary 2 says the solution of (2) only depends on the samples of its corresponding cluster without considering other cluster samples (eq. (3-4)). That is, the problem (2) can be solved by the k individual ordinary eigenvalue problems.

$$\boldsymbol{\Phi}\tilde{\boldsymbol{w}}_i = \eta_i\tilde{\boldsymbol{w}}_i \tag{3}$$
$$b_i = -\tilde{\boldsymbol{w}}_i^T\tilde{\boldsymbol{m}}_i \tag{4}$$

C. IMPLICIT TARGET REGRESSION

In the Section III B, we have concluded a linear regression method as a byproduct, which merging target variable into dependent variables and then seeking a fitting plane to fit these variant sample points. In the merged dataset, since we need not to strictly distinguish target variable or independent variables, we name it *implicit target regression* (Hereafter, shortly ImTarReg).

Linear regression aims to find a linear function $y = f(\boldsymbol{x})$, $\boldsymbol{x} \in R^d$ (generally, convex linear function) to fit a set of scatter points in d-dimensional linear space. We rewrite it as an equation $\varphi(\boldsymbol{x}, y) = y - f(\boldsymbol{x}) = 0$. As foresaid, we define such equation as

$$\tilde{\boldsymbol{w}} \cdot \begin{bmatrix} \boldsymbol{x} \\ y \end{bmatrix} + b = 0. \tag{5}$$

where $\tilde{\boldsymbol{w}}(\in R^{d+1})$ and $b(\in R)$ denote the parameters of the plane $\{z|\tilde{\boldsymbol{w}}^T\boldsymbol{z} + b = 0, \boldsymbol{z} \in R^{d+1}\}$ in the $(d+1)$ dimensional space. We describe this process in algorithm form.

Algorithm 1 ImTarReg: Implicit target regression algorithm

Input: a set of scatter points $\{\boldsymbol{x}_j, y_j\}_{j=1}^{l}$;

Output: parameter pair $(\tilde{\boldsymbol{w}}, b)$ of the fitting plane $\tilde{\boldsymbol{w}}^T\boldsymbol{z} + b = 0$;

Step1. Form a new data set $\{\tilde{\boldsymbol{x}}_j\}_{j=1}^{l}$, where $\tilde{\boldsymbol{x}}_j^T = (\boldsymbol{x}_j^T, y_j)$;
Step2. Compute the covariance matrix $\boldsymbol{\Phi}$ and mean $\boldsymbol{m}$ of the sample set $\{\tilde{\boldsymbol{x}}_j\}_{j=1}^{l}$;
Step3. Compute fitting plane by $\boldsymbol{\Phi}\tilde{\boldsymbol{w}} = \lambda\tilde{\boldsymbol{w}}$ and $b = -\tilde{\boldsymbol{w}}^T\boldsymbol{m}$, where $\tilde{\boldsymbol{w}}$ is an eigenvector corresponding to the minimal eigenvalue of $\boldsymbol{\Phi}$.

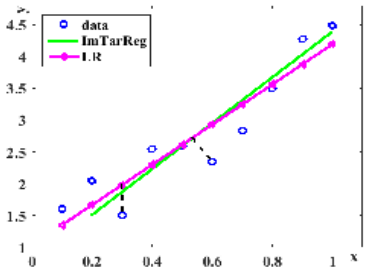


FIGURE 2. Illustration for line regression: ImTarReg vs. LR.

Fig. 2 illustrates an example for ImTarReg. The scatter points marked blue "o" are generated from the function $y = 3x + 0.5 + \varepsilon$, where ε is uniform distribution noise, and the component x of samples are sampling from the interval [0.1, 1) with step 0.1. The lines, marked green solid and magenta solid plus triangle, are regressed by ImTarReg and classic linear regression (shortly LR), respectively. Instead of minimizing square sum of distances between target and estimated value in LR, ImTarReg aims to seek a plane under minimizing square sum of distances between scatter points to the regression plane (the line marked magenta solid plus triangle). Geometrically, the LR minimizes the sum of line segments (left dash line segment of the Fig. 2) which parallelling to the y-axis, while for ImTarReg, it minimizes the sum of line segments (right dash line segment) perpendicular to the regression plane (magenta solid line plus triangle of Fig. 2).

From the viewpoint of data fitting, both LR and ImTarReg are capable of obtaining line regression functions, though they have different optimization objectives. However, similar to LR, when need predict values by ImTarReg, one can compute it by the expression (6), a variant version of (5).

$$y = -(\boldsymbol{w}^T\boldsymbol{x} + b)/\tilde{w}_{d+1}, \tag{6}$$

[1]When suffering degenerate eigenvalue problem, i.e., many eigenvectors sharing the same eigenvalue, one can refer to our work in 2010 [42].

Algorithm 2 The Plane Jump Method

Input: n scatter point pairs $\{\boldsymbol{x}_i, y_i\}_{i=1}^n$;
Output: the optimal k^*;

Step1. Compute $\hat{d}_k = \min_{(\tilde{\boldsymbol{w}}_i, b_i)} \sum_{i=1}^{k} \sum_{j=1}^{n_i} (\tilde{\boldsymbol{w}}_i^T \tilde{\boldsymbol{x}}_j^{(i)} + b_i)^2$, symbols defined as before;
Step2. Select a nonnegative transformation power p. (A typical value $p = (d+1)/2$, advised by [27]);
Step3. Compute the jumps $J_k = \hat{d}_k^{-p} - \hat{d}_{k-1}^{-p}$;
Step4. Find $k^* = \arg\max_k J_k$

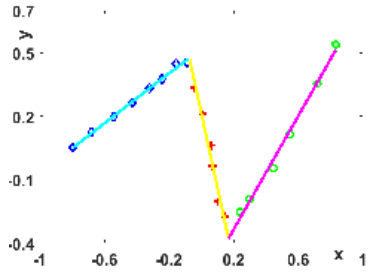


FIGURE 3. Illustration the PlrPC on 3-segmented data.

where $\boldsymbol{w} \in R^d$, and $w_j = \tilde{w}_j, j = 1, 2, \cdots, d$, w_j and $\tilde{w}_j$ denote the jth component of $\boldsymbol{w}$ and $\tilde{\boldsymbol{w}}$, respectively.

Next, we will discuss how to determine the number of plane clustering.

D. DETERMINING THE NUMBER OF PLANE CLUSTERING

In the cluster analysis fields, it is the most difficult problem for how to identify the "true" number of groups in a data set. Numerous approaches for solving this problem have been suggested over the years. Unfortunately, as said in [19] and [36], those methods are more generally applicable tend either to be model-based, which hence requiring strong parametric assumptions, or to be computation-intensive, or both. According to NFL theories, it is the fact that the optimal cluster partition algorithm does not exist unless merging knowledge in prior. Here we will merge plane linearity characteristics into "cluster jump method" [19] by replacing the point with plane as cluster center, and name it "plane jump method" (see Algorithm 2).

The jump method describes a monotone decreasing relation between d_k and k. Here d_k, a quantity from asymptotic rate distortion theory, is a measure of within cluster dispersion, and $\hat{d}_k$, its estimated value. Particularly, when set $k = 1$, the piecewise problem degenerates to linear regression, i.e., adopting one plane to fitting all training data, as described in Step1. Thus $\hat{d}_1 = \sum_{j=1}^{n} (\tilde{\boldsymbol{w}}^T \tilde{\boldsymbol{x}}_j^{(i)} + b)^2$, where $\tilde{\boldsymbol{w}}$ and b can be directly solved by (3) and (4).

In the end, we restatement our PlrPC in Algorithm 3 and interpret its geometrical meaning in Figure 3.

Fig. 3 gives an example for the PlrPC on the foresaid three segmented data. After Step 1 and 2, we sort the clusters by the first components, x-component, of their means, and then obtain the borders of the clusters neighbor to each other by their corresponding fitting planes.

Algorithm 3 The PlrPC Method

Input: n scatter point pairs $\{\boldsymbol{x}_i, y_i\}_{i=1}^n$;
Output: k sub intervals and fitting segmented planes

Step1. Run the plane jump method to determine the number of cluster k;
Step2. Alternatively run k-PC and ImTarReg to obtain k clusters and their corresponding fitting planes;
Step3. Sort the k clusters to partition the interval into k disjoint sub intervals by some certain component, saying, the first one;
Step4. List piecewise linear functions on each subinterval.

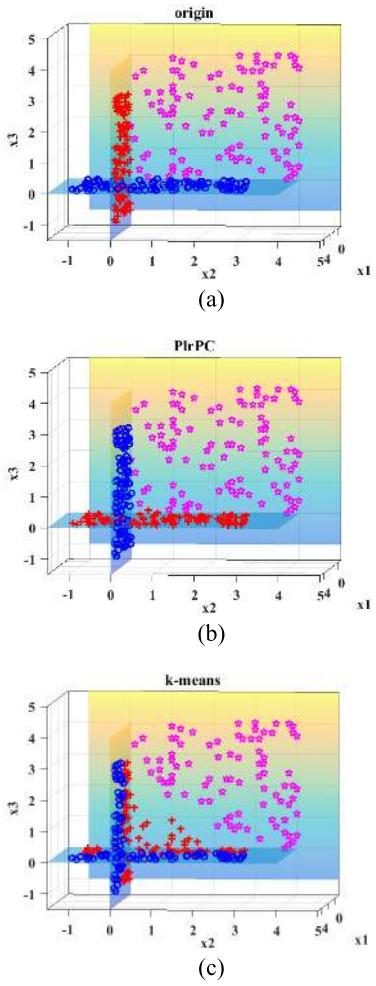


FIGURE 4. Illustration for 3-class plane-shaped data and experimental comparison between PlrPC and the k-Means based regression. (a) original distribution for 3-class synthetic data; (b-c) PlrPC based on kPC clustering and k-Means clustering.

IV. EXPERIMENTAL COMPARISONS

To verify the effectiveness of our proposed PlrPC, in additional of the foresaid theoretical analysis, we report the

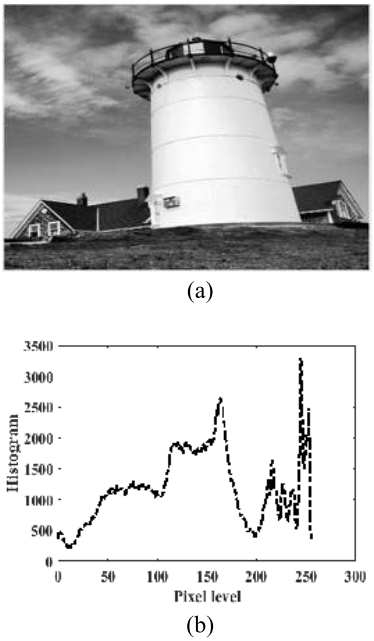


FIGURE 5. Illustration for the image "lighthouse" and its histogram. (a) 256-level gray image and (b) histogram.

empirical performance on synthetic and real data. As baseline, firstly we compare our plane-polytype piecewise regressor to the point-polytype one, typically, k-means, on some synthetic datasets. Then we further report the comparison results with the state-of-the-art regressors in two aspects: selection of the number of cluster k and regression error with MSE loss function on real data. All experiments are conducted on dell inspiron laptop computer with a 2.2GHz Intel Core i7 CPU and 8 GB of RAM running matlab 2015b.

A. SYNTHETIC DATA

For visualization, synthetic data points are composed of 300 samples from 3 different clusters. Points of each cluster are drawn from a 3-dimensional plane with 10 percentage uniform noise. Fig. 4a illustrates the 3-class points marked red "+", blue "o" and magenta "☆", respectively. To improve 3D stereoscopic vision, we also provide plane-shaped semi-transparent shadows (set transparency parameter alpha 0.5), where three planes are perpendicular to each other. Among them, points in one plane are linearly separate from the points in the other two crossed planes. Intuitionally, PlrPC able to achieve better fitting performance (Fig. 4b) than that of the k-means based regression (Fig. 4c). The regression result for PlrPC is almost coincident with original three plane distributions. Numerically, Fig. 4 reports that the cluster accuracy rate of PlrPC is 99.3% (298 out of 300), while for k-means regression, it is 65.0% (195 out of 300). Figure 4c also figures out that point-centered clustering usually failed to deal with the points located at overlapped area. While for PlrPC, due to considering data linearity (prior knowledge), it is more appropriate for piecewise linear regression tasks.

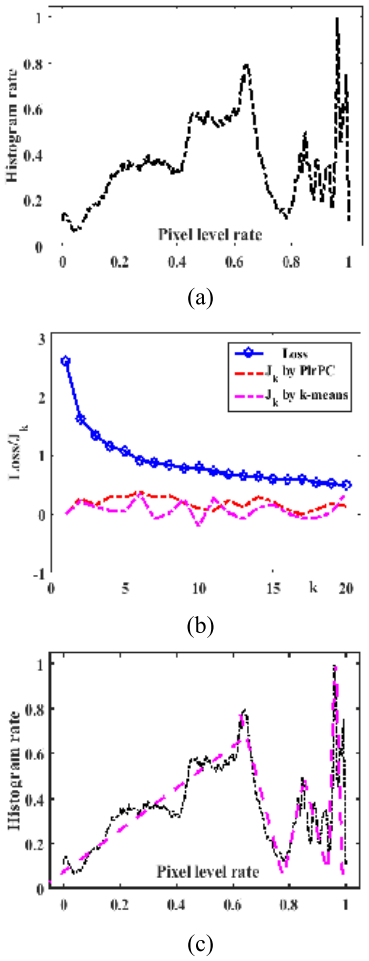


FIGURE 6. Illustration for our PlrPC on the histogram. (a) Normalized "lighthouse" histogram, (b) regression loss marked "loss" and Jk values by PlrPC and k-means, respectively, and (c) piecewise regression with 6 line segments by PlrPC.

B. REAL DATA

In this subsection, firstly, we will experimentally interpret the foresaid plane jump method (see Algorithm 2). A benchmark image, titled "lighthouse" from matlab image toolbox, will be used to validate our plane jump method. Fig. 5 illustrates the original image (Fig. 5a) and its histogram (Fig. 5b), where horizontal axis denotes image gray level, and the vertical axis, marked "histogram", denotes the number of pixels corresponding to different gray level.

This histogram will be piecewisely regressed by PlrPC, illustrated in Fig. 6. For better view sight, both pixel level and histogram are normalized to the interval [0,1], and mark them with "pixel level rate" and "histogram rate" respectively (see Fig. 6a). Fig. 6b illustrates three curves corresponding to kPC loss, Jk by PlrPC and Jk by k-means, respectively, when the number of clusters k grows from 1 to 20. The kPC loss curve (blue solid line plus "o") tends to monotonous decrease with the increase of k, and there seems to exist a knee point nearby $k = 6$ [37]. This may be an appropriate number of clusters k, as [38] advised. Meanwhile, the curve of Jk by PlrPC reaches the maximizer also at this position.

TABLE 1. UCI data information.

Datasets	Description	Total samples	Dimension
Yacht Hydrodynamics (Yacht)	Hydrodynamic performance of sailing yachts	308	7
Energy Efficiency (EEH and EEC)	Building shapes, 8 attributions and two responses: Heating and Cooling Load.	768	8
Concrete Strength (Concrete)	Concrete Compressive Strength	1030	9
Airfoil (Airfoil)	From NASA, aerodynamic and acoustic tests	1503	6
Wine Quality (Wine)	Red and white vinho verde wine	4898	12
Cycle Power Plant (Cycle)	Collected from a Combined Cycle Power Plant over 6 years (2006-2011)	9568	4
Facebook Comment Volume (Fbook)	Contain Features extracted from facebook posts	40949	54

TABLE 2. Comparisons of 8 UCI benchmark datasets.

Methods		Yacht	EEH	EEC	Concrete	Airfoil	Wine	CyclePower	Fbook
LR	loss*	7.270	2.089	2.266	8.311	0.037	0.586	3.6311	4.1472
	time#	**0.0016**	**0.0023**	**0.0036**	**0.0690**	**0.0035**	**0.1148**	**3.2480**	**37.8510**
MLP	loss	0.809	0.993	1.924	6.229	0.035	0.623	5.1939	6.5129
	time	10.3044	36.0802	40.0733	0.0989	0.5321	139.2146	152.6100	2972.1000
Kriging	loss	4.324	1.788	2.044	6.224	0.030	0.576	4.232	-$
	time	0.8520	3.2632	53.5670	45.4980	248.8750	380.2562	948.2500	-
SVR	loss	6.445	2.036	2.191	8.212	0.037	0.585	3.6237	-
	time	13.2530	38.4560	80.4320	160.7654	201.2345	380.4253	1134.5800	-
KNN	loss	5.299	1.937	2.148	7.068	0.026	0.537	2.880	-
	time	45.6715	152.4500	145.8710	215.4560	380.6547	420.5802	1223.8400	-
MARS	loss	1.011	**0.796**	1.324	4.871	0.035	0.570	2.431	6.5979
	time	0.9977	34.2314	45.2689	158.3650	320.2547	400.2587	440.8468	517.4350
Pace	loss	7.233	2.089	2.261	8.298	0.037	0.586	3.6311	24.1282
	time	2.3365	7.6580	9.3216	69.3284	120.5643	128.4365	263.0630	358.1100
ALAMO	loss	0.787	2.722	2.765	8.044	0.032	0.639	3.3374	7.3229
	time	12.3657	23.4587	40.2354	88.4569	220.6781	326.4521	289.4520	354.1200
OPLRA	loss	0.706	0.810	1.278	4.870	0.029	0.551	5.2782	-
	time	18.2541	126.5874	118.3695	198.4561	215.4682	320.4563	423.3400	-
PlrPC	loss	**0.664**	0.852	**1.128**	**2.563**	**0.023**	**0.427**	**1.2348**	**3.2324**
	time	8.3254	85.1432	88.3420	168.5474	123.5687	180.4569	223.4581	238.1520

*The best result of loss of each dataset is bold.
The results for average training time, the shortest training time is also bold.
$ The symbol "-" means that the results are unavailabe because of out of memory or overheavy training tasks.

To compare with point-centered clustering methods, we also report the Jk by k-means in Fig. 6b, which achieves the maximum J_k at $k = 20$. To further validate our proposed method, we also report piecewise fitting results (six segments, marked magenta dash line) by PlrPC, as illustrated in Fig. 6c. Note that our piecewise regressor almost able to reflect the tendency for the foresaid image histogram.

Another experiment carries on the benchmark UCI data. Eight UCI real world datasets will be used to test the prediction performance as reported in [29] and [39], including Yacht Hydrodynamics (shortly Yacht), Energy Efficiency Heating (EEH), Energy Efficiency Cooling (EEC), Concrete Strength, Airfoil, White Wine Quality (Wine), Combined Cycle Power Plant (Cycle), and Facebook Comment Volume (Fbook). Corresponding data information are listed in Tab. 1. Similar experimental setting to OPLRA [37], a 5-fold cross validation on each dataset is performed to estimate the absolute predictive accuracy (loss) of the regressors. The predictive results of 50 rounds of 5-fold cross validation are averaged as the final error. But for the latter two larger scale datasets, we average 5 rounds of 2-fold cross validation. For the purpose of comparison, we also introduce some state-of-the-art regression methods, including classic linear regression (LR), MLP (Mulilayer Perceptron), Kriging [40],

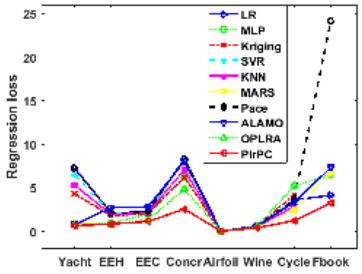


FIGURE 7. Illustration for the loss comparison of UCI datasets.

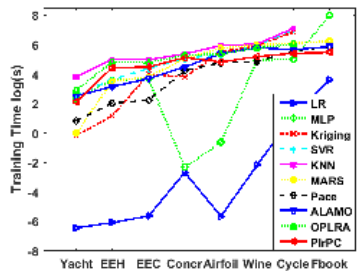


FIGURE 8. Illustration for the training time comparison.

SVR (Support Vector Regression), KNN (k Nearest Neighbor), MARS (Multi-variate Adaptive Regression Splines), Pace-Regression (Pace), ALAMO (Automated Learning of Algebraic Models for Optimization) and OPLRA. The experimental results are listed in Tab. 2 and showed in Fig. 7. To test model complexity, we also report average training time in Tab. 2 and Fig. 8.

From Tab. 2 and Fig. 7, different from the foresaid MSE, for the fair comparison, here we also adopt the absolute error (regression loss) to measure losses for all regressors. Generally, as piecewise regressors, both OPLRA and PlrPC outperform than other global regression methods. Our PlrPC achieves better regression loss on the most of datasets. For the data EEH, PlrPC is ranked as the third method, merely 0.054 far behind MARS and 0.042 behind OPLRA. Obviously, when we set the bigger number of clustering k, PlrPC able to achieve lower loss, as illustrated in Figure 6b. Note that, the results on the data Fbook (Facebook) are unavailable for the four regressors, including Kriging, SVR, KNN, and OPLRA. Kriging, SVR and KNN need to compute high order matrix operator for matrix inverse, kernel matrix and distance matrix, respectively. It is still a big challenge for time- and memory-consuming problem when facing large scale learning tasks. While for OPLRA, as foresaid, it need to consider attribution combination problem. That is, the dimension of the search space, spanned by attribution combination, will be increased exponentially, which unavoidably result in curse of dimensionality. Hence, there are no reports on the data Fbook for the four regressors in Tab. 2, Figs 7 and 8.

As far as training time is concerned, from Tab. 2 and Fig. 8, LR is the fastest method, while KNN is the slowest one among ten regressors. For the convenience of visualization, Fig. 8 reports training time in logarithmic to balance value difference from 0.0016 (LR training time on Yacht) to 2972.1 seconds (MLP on Fbook). The training time of our PlrPC is longer than that of LR, MLP, ALAMO, and Pace, but shorter than that of KNN, OPLRA. With the size of samples raising, PlrPC defeats KNN, MARS, Kriging, SVR, OPLRA, and ALAMO. PlrPC spends much training time in preparing kPC, while KNN, in finding k nearest neighbors by Euclidian distance. As for MARS, Kriging, ALAMO, they involve in matrix inversion. SVR needs to solve quadratic programming problem, besides kernel parameter selection and kernel matrix computation. For OPLRA, it costs much more time on attribution selection to determine intervals. As aforementioned, PlrPC only has one parameter, the number of clusters for kPC, which can be determined by the proposed ''Plane jump method''. Due to limit of paper volume, here we can only test main aspects, including regression loss and training time.

V. CONCLUSIONS

Following the point-centered regression, we proposed a plane-centered piecewise linear regressor PlrPC. Compare to the point-centered regressor, instead of the distance between points to cluster centers, PlrPC aims to seek k planes by minimizing the sum of distance between points to its corresponding fitting plane, and the leading problems can be analytically solved by k ordinary eigenvalue problems. Furthermore, to fuse the plane characteristics into optimization model, an implicit target regression and a plane jump method are also proposed, respectively. The former is used to transform the general regression problem to cluster one, while the latter is used to determine an appropriate number of clusters. Compared to the k-means, experiments on some artificial datasets explains that our PlrPC is more suitable for capturing linearity in data. We illustrates an example for fitting image histogram with a series of line segments. Experiments on some UCI benchmark datasets, PlrPC able to achieve lower regression loss and less training time on most cases.

APPENDIX
PROOFS OF THEOREMS

Theorem 1: The optimization problem (2) is a convex problem.

Proof: We rewrite the problem (2) as (A-1)

$$\begin{aligned} &\min\ J(\tilde{\boldsymbol{W}}, \boldsymbol{b}) = \sum_{i=1}^{k}\sum_{j=1}^{n_i} (\tilde{\boldsymbol{w}}_i^T \tilde{\boldsymbol{x}}_j^{(i)} + b_i)^2 \\ &s.t.\ ||\tilde{\boldsymbol{w}}_i||^2 = 1, \quad i = 1 \sim k \end{aligned} \tag{A-1}$$

Let $\boldsymbol{u}_i^T = [\tilde{\boldsymbol{w}}_i^T, b_i]$, $(\boldsymbol{z}_j^{(i)})^T = [(\tilde{\boldsymbol{x}}_j^{(i)})^T, 1] = [(\boldsymbol{x}_j^{(i)})^T, y_j, 1]$. We have

$$\sum_{i=1}^{k}\sum_{j=1}^{n_i} (\tilde{\boldsymbol{w}}_i^T \tilde{\boldsymbol{x}}_j^{(i)} + b_i)^2 = \sum_{i=1}^{k} \boldsymbol{u}_i^T \boldsymbol{A}_i \boldsymbol{u}_i \tag{A-2}$$

and the constraints

$$\boldsymbol{u}_i^T \boldsymbol{G} \boldsymbol{u}_i = 1, \quad i = 1 \sim k, \tag{A-3}$$

where $\boldsymbol{A}_i = \sum_{j=1}^{n_i} \boldsymbol{z}_j^{(i)} (\boldsymbol{z}_j^{(i)})^T$ is a Gram matrix, $\boldsymbol{G} = \begin{bmatrix} \boldsymbol{I}_{(d+1)\times(d+1)} & \boldsymbol{0}_{(d+1)\times 1} \\ \boldsymbol{0}_{1\times(d+1)} & 0 \end{bmatrix}$, $\boldsymbol{I}_{d\times d}$ denotes an identity matrix with the size d, and $\boldsymbol{0}_{m\times n}$ denotes $m \times n$ vector or matrix with all 0 entries.

We introduce new variables $\boldsymbol{U}^T = [\boldsymbol{u}_1^T, \boldsymbol{u}_2^T, \cdots, \boldsymbol{u}_k^T]$ and $\boldsymbol{A} = diag(\boldsymbol{A}_1, \boldsymbol{A}_2, \cdots, \boldsymbol{A}_k)$, herein $diag(.)$ means block matrix diagonalization. Thus $\sum_{i=1}^{k} \boldsymbol{u}_i^T \boldsymbol{A}_i \boldsymbol{u}_i = \boldsymbol{U}^T \boldsymbol{A} \boldsymbol{U}$, $\boldsymbol{U}$ is a column vector with the size of $k(d+2)$. The Hessian matrix of $\boldsymbol{U}^T \boldsymbol{A} \boldsymbol{U}$ over $\boldsymbol{U}$, says $2\boldsymbol{A}$, is semi-definite positive. Therefore, the objective function of (A-1) is convex, so it does the constraints of (A-3). ■

Theorem 2: The optimization problem (A-1) can be respectively solved by k ordinary eigenvalue problems.

Proof: Constructing Lagrange function and simplifying the problem of (A-2) and (A-3), we have

$$L(\boldsymbol{u}_i, \lambda_i) = \sum_{i=1}^{k} \boldsymbol{u}_i^T \boldsymbol{A}_i \boldsymbol{u}_i - \sum_{i=1}^{k} \lambda_i (\boldsymbol{u}_i^T \boldsymbol{G} \boldsymbol{u}_i - 1)$$

Let the deviation of (5) w.r.t. $\boldsymbol{u}_i$ equals to 0, the original optimization problem can be effectively solved by the following general eigenvalue problem.

$$\boldsymbol{A}_i \boldsymbol{u}_i = \lambda_i \boldsymbol{G} \boldsymbol{u}_i, \tag{A-4}$$

where $\boldsymbol{A}_i = \sum_{j=1}^{n_i} \boldsymbol{z}_j^{(i)} (\boldsymbol{z}_j^{(i)})^T$, $\boldsymbol{G} = \begin{bmatrix} \boldsymbol{I}_{(d+1)\times(d+1)} & \boldsymbol{0}_{(d+1)\times 1} \\ \boldsymbol{0}_{1\times(d+1)} & 0 \end{bmatrix}$.

It is important to note that,

$$\boldsymbol{A}_i = \sum_{j=1}^{n_i} \boldsymbol{z}_j^{(i)} (\boldsymbol{z}_j^{(i)})^T = [\boldsymbol{z}_1^{(i)}, \boldsymbol{z}_2^{(i)}, \cdots, \boldsymbol{z}_{n_i}^{(i)}] \begin{bmatrix} (\boldsymbol{z}_1^{(i)})^T \\ (\boldsymbol{z}_2^{(i)})^T \\ \vdots \\ (\boldsymbol{z}_{n_i}^{(i)})^T \end{bmatrix} \tag{A-5}$$

Substituting $(\boldsymbol{z}_j^{(i)})^T = [(\tilde{\boldsymbol{x}}_j^{(i)})^T, 1]$ into (A-5), we have

$$\boldsymbol{A}_i = \begin{bmatrix} \boldsymbol{x}_1^{(i)} & \boldsymbol{x}_2^{(i)} & \cdots & \boldsymbol{x}_{n_i}^{(i)} \\ 1 & 1 & \cdots & 1 \end{bmatrix} \begin{bmatrix} (\boldsymbol{x}_1^{(i)})^T & 1 \\ (\boldsymbol{x}_2^{(i)})^T & 1 \\ \vdots & \vdots \\ (\boldsymbol{x}_{n_i}^{(i)})^T & 1 \end{bmatrix}.$$

Let $\boldsymbol{X}_i = \begin{bmatrix} (\boldsymbol{x}_1^{(i)})^T \\ (\boldsymbol{x}_2^{(i)})^T \\ \vdots \\ (\boldsymbol{x}_{n_i}^{(i)})^T \end{bmatrix}$ and $\boldsymbol{1} = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}$, then

$$\boldsymbol{A}_i = \begin{bmatrix} \boldsymbol{X}_i^T \\ \boldsymbol{1}^T \end{bmatrix} \begin{bmatrix} \boldsymbol{X}_i & \boldsymbol{1} \end{bmatrix}. \tag{A-6}$$

The problem (A-4) is readily reformulated as

$$\begin{bmatrix} \boldsymbol{X}_i^T \boldsymbol{X}_i & \boldsymbol{X}_i^T \boldsymbol{1} \\ \boldsymbol{1}^T \boldsymbol{X}_i & \boldsymbol{1}^T \boldsymbol{1} \end{bmatrix} \begin{bmatrix} \boldsymbol{u}_i' \\ u_{i,d+2} \end{bmatrix} = \lambda_i \begin{bmatrix} \boldsymbol{u}_i' \\ 0 \end{bmatrix} \tag{A-7}$$

where $\boldsymbol{u}_i = [(\boldsymbol{u}_i')^T, u_{i,d+2}]^T$, $\boldsymbol{u}_i' = [u_{i1}, u_{i2}, \cdots, u_{i,d+1}]^T$, $u_{ij} \in R, j = 1, 2, \cdots, d+2$.

Expanding (A-7), we have

$$\boldsymbol{X}_i^T \boldsymbol{X}_i \boldsymbol{u}_i' + u_{i,d+2} \boldsymbol{X}_i^T \boldsymbol{1} = \lambda_i \boldsymbol{u}_i' \tag{A-8}$$

$$\boldsymbol{1}^T \boldsymbol{X}_i \boldsymbol{u}_i' + u_{i,d+2} n_i = 0 \tag{A-9}$$

So that for $\boldsymbol{u}_i' \in R^{d+1}$,

$$(\boldsymbol{X}_i^T \boldsymbol{X}_i - \frac{1}{n_i} \boldsymbol{X}_i^T \boldsymbol{1} \boldsymbol{1}^T \boldsymbol{X}_i) \boldsymbol{u}_i' = \lambda_i \boldsymbol{u}_i' \tag{A-10}$$

Let $\tilde{\boldsymbol{m}}_i = \frac{1}{n_i} \sum_{j=1}^{n_i} \tilde{\boldsymbol{x}}_j^{(i)}$ denotes the mean of the ith cluster. Substituting $n_i \tilde{\boldsymbol{m}}_i = \boldsymbol{X}_i^T \boldsymbol{1}$ into (A-10), we have

$$(\frac{1}{n_i} \boldsymbol{X}_i^T \boldsymbol{X}_i - \tilde{\boldsymbol{m}}_i \tilde{\boldsymbol{m}}_i^T) \boldsymbol{u}_i' = \eta_i \boldsymbol{u}_i' \tag{A-11}$$

where $\eta_i = \frac{\lambda_i}{n_i}$. ■

Corollary 1: The optimal value of problem (2) achieves at minimum sum of the k minimal eigenvalues of the k eigenvalue problems.

Proof: Substituting (A-4) into (A-2), we have $\sum_{i=1}^{k} \boldsymbol{u}_i^T \boldsymbol{A}_i \boldsymbol{u}_i = \sum_{i=1}^{k} \lambda_i \boldsymbol{u}_i^T \boldsymbol{G} \boldsymbol{u}_i = \sum_{i=1}^{k} \lambda_i$, where λ_i is the eigen-value of the ith eigenvalue equation $\boldsymbol{A}_i \boldsymbol{u}_i = \lambda_i \boldsymbol{G} \boldsymbol{u}_i$. Because of symmetric positive semi-definition of the matrices $\boldsymbol{A}_i$ and $\boldsymbol{G}$, each λ_i should be non-negative. Therefore, when the sum of k eigenvalues $\sum_{i=1}^{k} \lambda_i$ reaches the optimum, each λ_i should be minimum eigenvalue of its corresponding eigen-equation. ■

Corollary 2: When optimization objective function in (2) reaches the optimal value, the optimal solution can be uniquely determined by k covariance matrices and means of the corresponding k clusters.

Proof: $\boldsymbol{X}_i$, denoted as (A-6), is a sample matrix, where its jth row corresponds to the jth sample of the ith cluster, we can reformulate the expression $\frac{1}{n_i} \boldsymbol{X}_i^T \boldsymbol{X}_i - \tilde{\boldsymbol{m}}_i \tilde{\boldsymbol{m}}_i^T$ in (A-11) as

$$\begin{aligned} \frac{1}{n_i} \boldsymbol{X}_i^T \boldsymbol{X}_i - \tilde{\boldsymbol{m}}_i \tilde{\boldsymbol{m}}_i^T &= \frac{1}{n_i} \sum_{i=1}^{n_i} \tilde{\boldsymbol{x}}_j^{(i)} (\tilde{\boldsymbol{x}}_j^{(i)})^T - \tilde{\boldsymbol{m}}_i \tilde{\boldsymbol{m}}_i^T \\ &= \frac{1}{n_i} \sum_{i=1}^{n_i} (\tilde{\boldsymbol{x}}_j^{(i)} - \tilde{\boldsymbol{m}}_i)(\tilde{\boldsymbol{x}}_j^{(i)} - \tilde{\boldsymbol{m}}_i)^T \end{aligned} \tag{A-12}$$

Substituting (A-12) into (A-11), we have

$$\boldsymbol{\Phi} \boldsymbol{u}_i' = \eta_i \boldsymbol{u}_i' \tag{A-13}$$

where $\boldsymbol{\Phi} = \frac{1}{n_i} \sum_{i=1}^{n_i} (\tilde{\boldsymbol{x}}_j^{(i)} - \tilde{\boldsymbol{m}}_i)(\tilde{\boldsymbol{x}}_j^{(i)} - \tilde{\boldsymbol{m}}_i)^T$ is the covariance matrix of the ith cluster.

From the definition of $\boldsymbol{u}_i$ (defined in the proof of Theorem 1, $\boldsymbol{u}_i^T = [\tilde{\boldsymbol{w}}_i^T, b_i]$) and (A-7), we know that $\tilde{\boldsymbol{w}}_i = \boldsymbol{u}_i'$ and $b_i = u_{i,d+2}$, i.e.,

$$\boldsymbol{\Phi}\tilde{\boldsymbol{w}}_i = \eta_i \tilde{\boldsymbol{w}}_i \tag{A-14}$$

and

$$b_i = -\tilde{\boldsymbol{w}}_i^T \tilde{\boldsymbol{m}}_i \tag{A-15}$$

Thus the ith fitting plane is determined by the covariance matrix and mean of the ith cluster. The constraints of (A-1) are satisfied with constraint qualifications where there always exists strictly feasible solution since the constraints in (A-1) are only used to control the length of the optimal variables and avoid them shrink to zeros. So they satisfy Slater's condition [38], Section 5.2.3], and strong duality holds. Obviously, both the objective function and equation constraints are all convex (Theorem 1). Therefore, the problem has a unique solution [39]. ■

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers for their valuable comments and suggestions.

REFERENCES

[1] M. B. Douglas and G. W. Donald, *Nonlinear Regression Analysis and its Applications*. Hoboken, NJ, USA: Wiley, 2008.
[2] Y. Wang, W. Zheng, D. Zhang, and L. Zhang, "Pulsar profile denoising using kernel regression based on maximum correntropy criterion," *Optik Int. J. Light Electron Opt.*, vol. 130, pp. 757–764, Feb. 2017.
[3] Y. Zhang, S. Xu, K. Chen, Z. Liu, and C. L. P. Chen, "Fuzzy density weight-based support vector regression for image denoising," *Inf. Sci.*, vol. 339, no. 3, pp. 175–188, 2016.
[4] R. Hu, S. Wen, Z. Zeng, and T. Huang, "A short-term power load forecasting model based on the generalized regression neural network with decreasing step fruit fly optimization algorithm," *Neurocomputing*, vol. 221, pp. 24–31, Jan. 2017.
[5] Q. Di, P. Koutrakis, and J. Schwartz, "A hybrid prediction model for PM2.5 mass and components using a chemical transport model and land use regression," *Atmos. Environ.*, vol. 131, pp. 390–399, Apr. 2016.
[6] L. Wang, H. Liu, L. Liu, Q. Wang, S. Li, and Q. Li, "Prediction of peanut protein solubility based on the evaluation model established by supervised principal component regression," *Food Chem.*, vol. 218, pp. 553–560, Mar. 2017.
[7] N. Amrani, J. Serra-Sagristà, V. Laparra, M. W. Marcellin, and J. Malo, "Regression wavelet analysis for lossless coding of remote-sensing data," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 9, pp. 5616–5627, Sep. 2016.
[8] P. S. Kumar, A. Saravanan, K. A. Kumar, R. Yashwanth, and S. Visvesh, "Removal of toxic zinc from water/wastewater using eucalyptus seeds activated carbon: Non-linear regression analysis," *IET Nanobiotechnol.*, vol. 10, no. 3, pp. 244–253, 2016.
[9] S. J. Kim, C.-H. Kim, S.-Y. Jung, and Y.-J. Kim, "Shape optimization of a hybrid magnetic torque converter using the multiple linear regression analysis," *IEEE Trans. Magn.*, vol. 52, no. 3, Mar. 2016, Art. no. 8102504.
[10] G. Cheng, F. Zhu, S. Xiang, Y. Wang, and C. Pan, "Semisupervised hyperspectral image classification via discriminant analysis and robust regression," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 9, no. 2, pp. 595–608, Feb. 2016.
[11] D. Kwon, M. H. Azarian, and M. Pecht, "Remaining-life prediction of solder joints using RF impedance analysis and Gaussian process regression," *IEEE Trans. Compon., Packag., Manuf. Technol.*, vol. 5, no. 3, pp. 1602–1609, Nov. 2015.
[12] M. Liang, X. Huang, C.-H. Chen, X. Chen, and A. Tokuta, "Counting and classification of highway vehicles by regression analysis," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 5, pp. 2878–2888, Oct. 2015.
[13] A. Gersho and R. Gray, *Vector Quantization and Signal Compression*. Norwell, MA, USA: Kluwer, 1991.
[14] S. Chattopadhyay, D. K. Pratihar, and S. C. D. Sarkar, "Comparative study of fuzzy c-means algorithm and entropy-based fuzzy clustering algorithms," *Comput. Informat.*, vol. 30, no. 4, pp. 701–720, 2011.
[15] J. Macqueen, "Some methods for classification and analysis of multivariate observations," in *Proc. Berkeley Symp. Math. Statist. Probab.*, 1967, pp. 281–297.
[16] J. C. Bezdek, "Cluster validity with fuzzy set," *J. Cybern.*, vol. 3, pp. 58–72, Feb. 1974.
[17] J. Dunn, "A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters," *J. Cybern.*, vol. 3, no. 3, pp. 32–57, 1974.
[18] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proc. KDD*, vol. 96, 1996, pp. 226–231.
[19] P. S. Bradley and O. L. Mangasarian, "K-plane clustering," *J. Global Optim.*, vol. 16, no. 1, pp. 23–32, Jan. 2000.
[20] C. A. Sugar and G. M. James, "Finding the number of clusters in a dataset: An information-theoretic approach," *J. Amer. Statist. Assoc.*, vol. 98, no. 463, pp. 750–763, Sep. 2003.
[21] A. Magnani and S. P. Boyd, "Convex piecewise-linear fitting," *Optim. Eng.*, vol. 10, no. 3, pp. 1–17, 2009.
[22] L. A. Hannah and D. B. Dunson, "Multivariate convex regression with adaptive partitioning," *J. Mach. Learn. Res.*, vol. 14, no. 3, pp. 3261–3294, 2011.
[23] E. Pérez-Pellitero, J. Salvador, J. Ruiz-Hidalgo, and B. Rosenhahn, "Half hypersphere confinement for piecewise linear regression," in *Proc. IEEE Winter Conf. Appl. Comput. Vis.*, Lake Placid, NY, USA, Mar. 2016, pp. 1–9.
[24] Y.-H. Shao, L. Bai, Z. Wang, X.-Y. Hua, and N.-Y. Deng, "Proximal plane clustering via eigenvalue," *Procedia Comput. Sci.*, vol. 17, pp. 41–47, May 2013.
[25] Z.-M. Yang, Y.-R. Guo, C.-N. Li, and Y.-H. Shao, "Local k-proximal plane clustering," *Neural Comput. Appl.*, vol. 26, no. 1, pp. 199–211, 2015.
[26] G. F. Malash and M. I. El-Khaiary, "Piecewise linear regression: A statistical method for the analysis of experimental adsorption data by the intraparticle-diffusion models," *Chem. Eng. J.*, vol. 163, no. 3, pp. 256–263, 2010.
[27] A. S. Ahmed and I. M. I. Ramadan, "A generalized piecewise regression for transportation models," *Int. J. Comput. Appl.*, vol. 129, no. 1, pp. 16–22, 2015.
[28] B. Strikholm, "Determining the number of breaks in a piecewise linear regression model," Dept. Econ. Statist. Decis. Support, Stockholm School Econ., SSE/EFI Working Paper Series Econ. Finance, Tech. Rep. 648, 2006.
[29] S. Song, L. Ge, and Z. Zhang, "Accurate position estimation of SRM based on optimal interval selection and linear regression analysis," *IEEE Trans. Ind. Electron.*, vol. 63, no. 6, pp. 3467–3478, Jun. 2016.
[30] L. Yang, S. Liu, S. Tsoka, and L. G. Papageorgiou, "Mathematical programming for piecewise linear regression analysis," *Expert Syst. Appl.*, vol. 44, pp. 156–167, Feb. 2016.
[31] Y. Yamamoto and P. Perron, "Estimating and testing multiple structural changes in linear models using band spectral regressions," *Econometrics J.*, vol. 16, no. 3, pp. 400–429, 2013.
[32] J. Acharya *et al.*, "Fast algorithms for segmented regression," in *Proc. ICML*, vol. 48, New York, NY, USA, 2016, pp. 2878–2886.
[33] O. Mangasarian and E. Wild, "Multisurface proximal support vector machine classification via generalized eigenvalues," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 1, pp. 69–74, Jan. 2006.
[34] B. Mirkin, "Choosing the number of clusters," *Wiley Interdiscipl. Rev., Data Mining Knowl. Discovery*, vol. 1, no. 3, pp. 252–260, May/Jun. 2011.
[35] O. Arbelaitz, I. Gurrutxaga, J. Muguerza, J. M. Pérez, and I. Perona, "An extensive comparative study of cluster validity indices," *Pattern Recognit.*, vol. 46, no. 1, pp. 243–256, 2013.
[36] X. Yang, S. Chen, and B. Chen, "Plane-Gaussian artificial neural network," *Neural Comput. Appl.*, vol. 21, no. 2, pp. 305–317, 2012.
[37] A. Keshvari, "Segmented concave least squares: A nonparametric piecewise linear regression," *Eur. J. Oper. Res.*, vol. 266, no. 2, pp. 585–594, 2018.
[38] J. Liang, X. Zhao, D. Li, F. Cao, and C. Dang, "Determining the number of clusters using information entropy for mixed data," *Pattern Recognit.*, vol. 45, no. 6, pp. 2251–2265, 2012.

[39] L. Yang, "Optimization approaches for data mining in biological system," M.S. thesis, Univ. College London, London, U.K., 2015.
[40] G. Szatmàri *et al.*, "Optimization of second-phase sampling for multivariate soil mapping purposes: Case study from a wine region, Hungary," *Geoderma*, to be published. doi: 10.1016/j.geoderma.2018.02.030.
[41] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.
[42] X. Yang, Y. Wang, and T. Yun, "Direct multicategory proximal support vector machine classifier with degenerate eigenvalue problem," in *Proc. Chin. Conf. Pattern Recognit. (CCPR)*, Chongqing, China, Oct. 2010, pp. 1–5. doi: 10.1109/CCPR.2010.5659331.

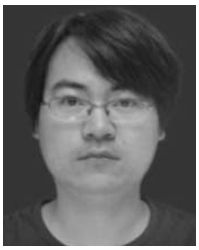
XUBING YANG received the B.S. degree in mathematics from Anhui University, in 1997, and the M.S. and Ph.D. degrees in computer applications from the Nanjing University of Aeronautics and Astronautics, in 2004 and 2008, respectively. In 2008, he joined Nanjing Forestry University, where he is currently an Associate Professor with the Computer Science and Engineering Department. His research interests include pattern recognition, machine learning, and neural computing. In these fields, he has authored or co-authored over 50 scientific journal papers.

HONGXIN YANG received the B.S. degree in computer science and technology from Nanjing Forestry University, in 2017, where she is currently pursuing the master's degree. Her research interests include pattern recognition, machine learning, and image processing.

FUQUAN ZHANG received the M.S. degree in computer science from Shen Yang Li Gong University, in 2005, and the Ph.D. degree from Hanyang University, Seoul, South Korea. His research fields include 3G/4G cellular systems and wireless mesh networks.

LI ZHANG received the B.S. degree in computer science from the Changsha University of Science and Technology, in 2007, and the M.S. and Ph.D. degrees in computer science from the Nanjing University of Aeronautics and Astronautics, in 2010 and 2015, respectively. Since 2016, he has been an Assistant Professor with the College of Information Science and Technology, Nanjing Forestry University. His research interests include machine learning, probabilistic graphical model, and bioinformatics.

XIJIAN FAN received the B.Sc. degree in information and communication technology from the Nanjing University of Posts and Telecommunications, in 2009, the M.Sc. degree in computer information and science from Hohai University, in 2012, and the Ph.D. degree from the School of Engineering, University of Warwick, U.K. His research interests include image processing, computer vision, and biomedical engineering.



QIAOLIN YE received the B.S. degree in computer science from the Nanjing Institute of Technology, Nanjing, China, in 2007, the M.S. degree in computer science and technology from Nanjing Forestry University, Nanjing, in 2009, and the Ph.D. degree in pattern recognition and intelligence system from the Nanjing University of Science and Technology, Jiangsu, China, in 2013. He is currently an Associate Professor with the Computer Science Department, Nanjing Forestry University. He has authored over 50 scientific papers. Some of them are published in the IEEE TNNLS, the IEEE TIFS, and the IEEE TCSVT. His research interests include machine learning, data mining, and pattern recognition.

LIYONG FU received the B.Sc. degree in forestry from Shanxi Agriculture University, in 2007, the M.Sc. degree in forest biometrics from Nanjing Forestry University, in 2009, and the Ph.D. degree in forest biometrics from the Chinese Academy of Forestry, China, in 2012, where he is currently a Full Professor of forest biometrics with the Department of Forest Management and Statistics. He has published over 30 scientific articles in prestigious peer-reviewed international journals, including briefings in bioinformatics and frontiers in plant science, during the recent five years.