Automatica 73 (2016) 155–162

Contents lists available at ScienceDirect

Automatica

journal homepage: www.elsevier.com/locate/automatica

ELSEVIER

Brief paper

Piecewise affine regression via recursive multiple least squares and multicategory discrimination[☆]

Valentina Breschi, Dario Piga, Alberto Bemporad

IMT School for Advanced Studies Lucca, Piazza San Francesco 19, 55100 Lucca, Italy

ARTICLE INFO

Article history:
Received 1 July 2015
Received in revised form
25 May 2016
Accepted 20 June 2016
Available online 5 September 2016

Keywords:
PWA regression
System identification
Clustering
Recursive multiple least squares
Multicategory discrimination

ABSTRACT

In nonlinear regression choosing an adequate model structure is often a challenging problem. While simple models (such as linear functions) may not be able to capture the underlying relationship among the variables, over-parametrized models described by a large set of nonlinear basis functions tend to overfit the training data, leading to poor generalization on unseen data. Piecewise-affine (PWA) models can describe nonlinear and possible discontinuous relationships while maintaining simple local affine regressor-to-output mappings, with extreme flexibility when the polyhedral partitioning of the regressor space is learned from data rather than fixed a priori. In this paper, we propose a novel and numerically very efficient two-stage approach for PWA regression based on a combined use of (i) recursive multi-model least-squares techniques for clustering and fitting linear functions to data, and (ii) linear multi-category discrimination, either offline (batch) via a Newton-like algorithm for computing a solution of unconstrained optimization problems with objective functions having a piecewise smooth gradient, or online (recursive) via averaged stochastic gradient descent.

1. Introduction

Regression analysis is a supervised learning method which aims at reconstructing the relationship between feature vectors $x \in \mathbb{R}^{n_x}$ and continuous-valued target outputs $y \in \mathbb{R}^{n_y}$ from a set of training data. PieceWise Affine (PWA) functions provide simple yet flexible model structures for nonlinear regression, as they can describe nonlinear and possible discontinuous relationships between the regressor x and the output y. They are defined by partitioning the regressor space into a finite number of polyhedral regions with non-overlapping interiors and by considering an affine model on each polyhedron.

The *PWA regression problem* amounts to learning, from a set of training data, both the partition of the regressor space and the parameters defining each affine submodel. PWA regression is an NP-hard problem in general (see Lauer, 2015 for a detailed analysis on the complexity of PWA regression), and several algorithms to estimate PWA maps from data are available in the literature (see Garulli, Paoletti, & Vicino, 2012; Paoletti, Juloski, Ferrari-Trecate, & Vidal, 2007 for an overview). A convex relaxation, based on ℓ_1 regularization, is proposed in Ohlsson and Ljung (2013) to approximate the underlying combinatorial problem arising from PWA regression. In Roll, Bemporad, and Ljung (2004) the authors solve the PWA regression problem via mixed-integer programming. As the number of integer variables increases with the number of training samples, the approach is limited to problems with a small number of observations in which global optimality is sought. The algorithms proposed in Bemporad, Garulli, Paoletti, and Vicino (2005), Ferrari-Trecate, Muselli, Liberati, and Morari (2003), Juloski, Weiland, and Heemels (2005) and Nakada, Takaba, and Katayama (2005) first compute the parameters of the affine local models, then partition of the regressor space. The observations are clustered by assigning each datapoint to a submodel according to a certain criterion, estimating at the same time the parameters of the affine submodels. In a second stage, linear separation techniques are used to compute the polyhedral partition. These algorithms have shown good performance in practice, but can be numerically inefficient. The *greedy* algorithm of Bemporad et al. (2005) to partition infeasible sets of linear inequalities can be computationally heavy in case of large training sets. The *Expectation Maximization* (EM) algorithm

[☆] This work was partially supported by the European Commission under project H2020-SPIRE-636834 "DISIRE—Distributed In-Situ Sensors Integrated into Raw Material and Energy Feedstock" (http://spire2030.eu/disire/). The material in this paper was partially presented at the 15th European Control Conference, June 29–July 1, 2016, Aalborg, Denmark. This paper was recommended for publication in revised form by Associate Editor Erik Weyer under the direction of Editor Torsten Söderström.

E-mail addresses: valentina.breschi@imtlucca.it (V. Breschi), dario.piga@imtlucca.it (D. Piga), alberto.bemporad@imtlucca.it (A. Bemporad).

http://dx.doi.org/10.1016/j.automatica.2016.07.016
0005-1098/© 2016 Elsevier Ltd. All rights reserved.

used to numerically implement the statistical clustering method of Nakada et al. (2005) can become inefficient in case of PWA maps with many parameters. In Juloski et al. (2005), the submodel parameters are described through probability density functions, which are iteratively updated through particle filtering algorithms; however, an accurate approximation of the probability density functions might require a high number of particles. In Ferrari-Trecate et al. (2003), the regressor vectors are clustered through a K-means-like algorithm and the submodel parameters are obtained via weighted least-squares. Although Ferrari-Trecate et al. (2003) is able to handle large training sets both in the clustering and in the parameter estimation phase, poor results might be obtained when the affine local submodels are over-parametrized (i.e., the local models depend on redundant regressors), since the distances in the regressor space (i.e., the only criterion used for clustering) turns out to be corrupted by redundant, thus irrelevant, information.

Another limitation affecting the PWA regression algorithms mentioned above is that they can be executed only in a batch mode and thus they are not suitable for online applications, in which the PWA model must be updated in real-time when new data are acquired. A computationally efficient algorithm for online PWA regression was proposed in Bako, Boukharouba, Duviella, and Lecoeuche (2011), where training samples are clustered iteratively and model parameters are updated through recursive least-squares. A main limitation of the approach is that the polyhedral partition of the regressor space is given by the Voronoi diagram of the clusters' centroids, a less flexible structure than general linear separation maps that may limit regression capabilities.

This paper describes a novel approach for approximating vector-valued, and possibly discontinuous, functions in PWA form, trying to overcome the aforementioned limitations of existing methods. The proposed algorithm consists of two stages: (**S1**) simultaneous *clustering* of the regressor vectors and *estimation* of the model parameters, performed recursively by processing the training pairs $\{x(k), y(k)\}$ sequentially; (**S2**) *computation of a polyhedral partition* of the regressor space through efficient multi-class linear separation methods, either performed in a batch way via a Newton-like method, or online (recursively) via an averaged stochastic gradient descent algorithm. Overall, the PWA regression algorithm is computationally very effective for offline learning and suitable for online learning, as shown in the example. The application of the proposed PWA regression algorithm to the identification of linear parameter-varying and hybrid dynamical models is discussed in Breschi, Bemporad, and Piga (2016).

The paper is organized as follows. The PWA regression problem is described in Section 2. Section 3 describes the algorithm used to simultaneously cluster the observed regressors and update the model parameters, and the multi-category discrimination algorithms used to compute the polyhedral partition of the regressor domain. A simulation example is reported in Section 4 to show the effectiveness of the proposed approach.

1.1. Notation

Let $\mathbb{R}^n$ be the set of real vectors of dimension n. Let $I \subset \{1, 2, \dots, \}$ be a finite set of integers and denote by $|I|$ the cardinality of I. Given a vector $a \in \mathbb{R}^n$, let a_i denote the ith entry of a, a_I the subvector obtained by collecting the entries a_i for all $i \in I$, $\|a\|_2$ the Euclidean norm of a, a_+ a vector whose ith element is $\max\{a_i, 0\}$. Given two vectors $a, b \in \mathbb{R}^n$, $\max(a, b)$ is the vector whose ith component is $\max\{a_i, b_i\}$. Given a matrix $A \in \mathbb{R}^{n\times m}$, A' denotes the transpose of A, A_i the ith row of A, A_I the submatrix of A obtained by collecting the rows A_i for all $i \in I$, $A_{I,J}$ the submatrix of A obtained by collecting the rows and columns of A indexed by $i \in I$ and $j \in J$, respectively. Let I_n be the identity matrix of size n, and $\mathbf{1}_n$ and $\mathbf{0}_n$ be the n-dimensional column vector of ones and zeros, respectively. The symbol "$\propto$" denotes linear proportionality.

2. Problem statement

Consider a vector-valued PWA function $f : \mathcal{X} \to \mathbb{R}^{n_y}$ defined as

$$f(x) = \begin{cases} A_1[1 \quad x']' & \text{if } x \in \mathcal{X}_1, \\ \vdots & \\ A_s[1 \quad x']' & \text{if } x \in \mathcal{X}_s, \end{cases} \tag{1}$$

where $x \in \mathbb{R}^{n_x}$, $\mathcal{X} \subseteq \mathbb{R}^{n_x}$, $s \in \mathbb{N}$ denotes the number of affine local models defining f, $A_i \in \mathbb{R}^{n_y\times(n_x+1)}$ are parameter matrices, and the sets $\mathcal{X}_i$, $i = 1, \dots, s$ are polyhedra, that form a complete polyhedral partition[1] of the space $\mathcal{X}$. Function f is not assumed to be continuous over the boundaries of the polyhedra $\{\mathcal{X}_i\}_{i=1}^s$. Therefore, to avoid that f might take multiple values at the boundaries of $\{\mathcal{X}_i\}_{i=1}^s$, some inequalities can be replaced by strict inequalities in the definition of the sets $\mathcal{X}_i$ to avoid ambiguities when evaluating f on the boundary between neighboring polyhedra.

We address a PWA regression problem, which aims at computing a PWA map f fitting a given set of N input/output pairs $\{x(k), y(k)\}_{k=1}^N$. Computing the PWA map f requires (i) choosing the number of affine submodels s, (ii) computing the parameter matrices $\{A_i\}_{i=1}^s$ that characterize the affine local models of the PWA map f and (iii) finding the polyhedral partitioning $\{\mathcal{X}_i\}_{i=1}^s$ of the regressor space $\mathcal{X}$ where those local models are defined.

When choosing s one must take into account the tradeoff between fitting the data and avoiding model complexity and overfit, with consequent poor generalization on unseen data. This is related to one of the most crucial aspects in function learning, known as *bias–variance tradeoff* (Vapnik, 1998). In this work, we assume that s is fixed by the user. The value of s can be chosen through cross-validation, with a possible upper-bound dictated by the maximum tolerable complexity of the estimated model.

3. PWA regression algorithm

As mentioned in Section 1, we tackle the PWA regression problem in two stages: S1 (iterative clustering and parameter estimation) and S2 (polyhedral partition of the regressor space).

3.1. Recursive clustering and parameter estimation

Stage S1 is carried out as described in Algorithm 1. The algorithm is an extension to the case of multiple linear regressions and clustering of the (computationally very efficient) approach proposed in Alexander and Ghirnikar (1993) for solving recursive least squares problems using inverse QR decomposition. Algorithm 1 updates the clusters and the model parameters iteratively and thus it is also suitable for online applications, when data are acquired in real-time.

The algorithm requires an initial guess for the parameter matrices A_i and cluster centroids c_i, $i = 1, \dots, s$. Because of the greedy nature of Algorithm 1, the final estimate depends on the chosen initial conditions, and no fit criterion to minimize $\{\|y(k) - f(x(k))\|\}_{k=1}^N$ is optimized. Zero matrices A_i, randomly chosen centroids c_i, and identity covariance matrices R_i are a possible initialization. In alternative, if Algorithm 1 can be executed in a batch mode, one can initialize the parameter matrices $A_1, \dots, A_s$ all equal to the best linear model

$$A_i \equiv \arg\min_A \sum_{k=1}^N \left\| y(k) - A \begin{bmatrix} 1 \\ x(k) \end{bmatrix} \right\|_2^2, \quad \forall i = 1, \dots, s \tag{2}$$

[1] A collection $\{\mathcal{X}_i\}_{i=1}^s$ is a complete partition of the regressor domain $\mathcal{X}$ if $\bigcup_{i=1}^s \mathcal{X}_i = \mathcal{X}$ and $\mathring{\mathcal{X}}_i \cap \mathring{\mathcal{X}}_j = \emptyset$, $\forall i \neq j$, with $\mathring{\mathcal{X}}_i$ denoting the interior of $\mathcal{X}_i$.