Regression via Logic Supervised Classification

Vladimir Ryazanov

Institution of Russian Academy of Sciences, Dorodnicyn Computing Centre of RAS
Vavilov st. 40, 119333 Moscow, Russia
rvvccas@mail.ru

Abstract. An approach to the restoration of dependences (regressions) is proposed that is based on solving problems of supervised classification. The main task is finding the optimal partitioning of the range of values of dependent variable on a finite number of intervals. It is necessary to find optimal number of change-points and their positions. This task is formulated as search and application of piece-wise constant function. When restoring piecewise constant functions, the problem of local discrete optimization using a model of logic supervised classification in leave –one-out mode is solved. The value of the dependent value is calculated in two steps. At first, the problem of classification of feature vector is solved. Further, the dependent variable is calculated as half of the sum of change-points values of the corresponding class.

Keywords: regression, supervised classification, discrete optimization, approximation.

1 Introduction

Many data analysis tasks may be written in the next standard form. Let training sample $\{z_i, \mathbf{x}_i\}, i = 1,2,...,m$, is given, and $\mathbf{x}_i = (x_{i1}, x_{i2},..., x_{in})$ is a feature description of an object, $z_i \in R$, $x_{ij} \in M_j$ (M_j - a set of allowed values of feature № j). Vector $\mathbf{x}_i$ will be considered as a vector of values of independent parameters, and scalar z_i as a dependent value (it is supposed that z_i may be calculated by $\mathbf{x}_i$, i.e. $z_i = f(\mathbf{x}_i)$). It takes to calculate $z = f(\mathbf{x})$, $z \in R$ for any new $\mathbf{x} = (x_1, x_2,..., x_n)$, $x_j \in M_j$.Vector $\mathbf{x}$ is the objects (situation, phenomenon or process) description in term of features, and z is the value of some of its hidden scalar characteristic.

This problem in the statistical formulation is known as the problem of the regression reconstruction. Regression is a function of conditional expectation, assuming the existence of conditional density $p(z \mid \mathbf{x})$. In this paper, we will not use any probabilistic models.

Currently, there are different approaches to the restoration of regressions when $x_i \in R, i = 1,2,..., n$, which can be conditionally divided into two types:

J.A. Carrasco-Ochoa et al. (Eds.): MCPR 2013, LNCS 7914, pp. 242–253, 2013.
© Springer-Verlag Berlin Heidelberg 2013

parametric and nonparametric. Parametric approach [1] assumes the functional dependence of a certain type, depending on the parameters $\boldsymbol{\omega}$:

- linear regression - $f(\boldsymbol{\omega},\mathbf{x}) = \sum_{j=1}^{n} \omega_j x_j + \omega_0$;
- polynomial regression of degree γ - $f(\boldsymbol{\omega},\mathbf{x}) = \sum_{p_1=0}^{\gamma_1} \ldots \sum_{p_n=0}^{\gamma_n} \omega_{p_1 \ldots p_n} x_1^{p_1} \ldots x_n^{p_n}, \gamma = \sum_{i=1}^{n} \gamma_i$;
- curvilinear regression - $f(\boldsymbol{\omega},\mathbf{x}) = \sum_{j=1}^{k} \omega_j \phi_j(x_1,\ldots,x_n) + \omega_0$, $\phi_1,\ldots,\phi_k$ - transformations $R^n \to R$;
- logistic regression: $f(\boldsymbol{\omega},\mathbf{x}) = \dfrac{1}{1+\exp(-s)}$, $s = \sum_{j=1}^{n} \omega_j x_j + \omega_0$.

In the non-parametric approach [2] characteristic z for $\mathbf{x}$ is defined as $z = \dfrac{\sum_{i=1}^{m} \omega_i(\mathbf{x}) z_i}{\sum_{i=1}^{m} \omega_i(\mathbf{x})}$, where $\omega_i(\mathbf{x}) = K\left(\dfrac{\rho(\mathbf{x},\mathbf{x}_i)}{h}\right)$, $i = 1,2,\ldots,m$, K - kernel function, h - the width of the window. Well-known methods of support vector machine regressions [3] may be considered as curvilinear regressions.

In recent years a large number of publications devoted to regression via classification. In [4] it was proposed general fuzzy piecewise regression where change-points and their positions are obtained simultaneously as a solution of a mixed-integer programming problem. It is supposed that $\mathbf{x} \in R^n$. Change-points which are the joints of the pieces are quoted from conventional statistical piecewise regression. The proposed fuzzy piecewise regression is the direct generalization of linear piecewise regression. In [5] the Bayesian regression algorithm for piecewise constant functions with unknown segment number, location and level is proposed. It is assumed that one-dimension measurements of some function at discrete locations are given and measurements are independently distributed. The common polynomial-time dynamic-programming-type algorithm is derived.

A heuristics approach for learning regression rules by transforming the regression problem into a classification task is proposed in [6]. The discretization of the class variable is integrated into the rule learning algorithm. The key idea is the dynamical definition of a region around the target value predicted by the rule. The common approaches to construction of regression via classification are observed in [7]. The authors note that standard methods

comprise two stages: the discretization of the numeric dependent variable in order to learn a classification model, and the reverse process of transforming the class output of the model into a numeric prediction. The discretization of target variable is considered usually as some unsupervised classification task. The universal estimator for the regression problem in supervised learning is considered in [8]. The based on a least-square fitting procedure estimator does not depend on any a priori assumptions about the regression function to be estimated. It is proved that if the regression function is of a certain class, then the estimator converges to the regression function with an optimal rate of convergence in terms of the number of samples. It is assumed the existence of probability measure on R^{n+1}. There are other closes in a matter of fact approaches.

It should be noted certain limitations of these approaches. Parametric approaches require a priori knowledge of the analytical form of functions. The presence of different types of features (real, nominal, binary, ordinal, etc.) requires additional tools for describing objects in a single scale. Nonparametric methods use widely frequency estimation, distance functions, which can be very approximate and practically difficult for samples of small length, with a large number of independent parameters under various information and diverse nature. Many studies suggest a probabilistic model of the data and $\mathbf{x} \in R^n$. Construction of the functions of multiple nonlinear regressions using the analytical methods of mathematical statistics is impossible in most cases.

In this paper we propose an approach not involving probabilistic assumptions and based on supervised classification. According to the training set are the change-points that define an optimal partition of the sample into a finite number of classes. To find the optimal number of classes and positions of change-points the logic supervised classification model with leave-one-out procedure is used. All problems related to different information content of features, their type, and metrics are transferred to the level of supervised classification where the effective logic methods for supervised classification are used. The optimal number of change-points, their positions, and the classification algorithm are obtained simultaneously by solution of the discrete optimization task and use fast procedure for re-training of neighboring supervised classification tasks.

The basic idea is as follows. We assume that the range of dependent variable is the interval $[a,b]$, $a = \min\limits_{i=1,2,...,m} z_i$, $b = \max\limits_{i=1,2,...,m} z_i$. There is a partition of the segment $[a,b]$ using points $a_1 < a_2 < ... < a_{l-1}$ for l segments $I_1 = [a_0, a_1], I_2 = (a_1, a_2],..., I_l = (a_{l-1}, a_l]$, $a_0 = a$, $a_l = b$. Then the problem of approximate calculating the value of the regression $z = f(\mathbf{x})$ can be solved as follows.

1. The set $\tilde{K}_j = \{\mathbf{x}_i : z_i \in I_j, i = 1,2,...,m\}$ will be corresponding to the segment I_j, $j = 1,2,...,l$. Denote class $K_j = \{\mathbf{x} : z = f(\mathbf{x}) \in I_j\}$, $\tilde{K}_j \subseteq K_j$, $j = 1,2,...,l$.

2. For any $\mathbf{x}$, supervised classification problem is solved with respect to classes $K_\nu, \nu = 1,2,...,l$.

3. We put $z = f(\mathbf{x}) = (a_i + a_{i-1})/2$, if $\mathbf{x}$ classified as $\mathbf{x} \in K_i$.

Function $f(\mathbf{x})$ is uniquely determined by the partition and classification algorithms. So, we must enter the criterion of $f(\mathbf{x}) \equiv f(\mathbf{x}, l, a_1, a_2, ..., a_{l-1})$ quality as a function of the number of change-points l and their positions $a_1, a_2, ..., a_{l-1}$, take classification model and find the optimal solution, which is to construct function $f(\mathbf{x})$.

Note that for the implementation the item number 2, there are various models of classification by precedent. The following sections will be offered to the implementation of this general approach for the case of some logical supervised classification algorithm. Without loss of generality, we assume that the values z_i are different, and the objects of the training sample in ascending order of values z_i, i.e. $z_i < z_{i+1}, i = 1,2,...,m-1$.

The paper is organized as follows. Section 2 develops the theoretical part of paper. The statement of main optimization task is formulated in 2.1. The standard local optimization algorithm is considered in 2.2. The logical supervised classification algorithm is explained in 2.3. It's modification does not require introduction of metric for some feature. The feature may be only ordered. The neighboring classes are considered in main optimization task. It is required to construct a classification algorithm for neighbor task and efficiently to compute the optimized criterion in leave-one-out mode. The problem of recalculation of considered supervised classification algorithm for neighbor classes is considered in 2.4. Section 3 gives some illustration of proposed method and the results of experiments for one practical task. Some remarks are denoted in paper conclusion (section 4).

2 Construction of Approximate Regressions as Piecewise Constant Functions

2.1 Main Optimization Task

Let $A^j \equiv A^j(a_1, a_2, ..., a_{l-1}), j = 1,2,...,m,$ is some classification algorithm with respect to the classes defined in accordance with item 1 of introduction. A^j corresponds to some numbers $a_1 < a_2 < ... < a_{l-1}$, and to sample $\{z_i, \mathbf{x}_i\}, i = 1,2,...,m$, $i \neq j$. Let $A^j(\mathbf{x}_j) = t$ denotes that $\mathbf{x}_j$ is classified by A^j as $\mathbf{x}_j \in K_t$.

Let $l \in \{2,3,...,[m/3]\}$ is fixed. We introduce $F(y_0, y_1, ..., y_{l-1}, y_l)$ as the criterion of $f(\mathbf{x}) \equiv f(\mathbf{x}, l, a_1, a_2, ..., a_{l-1})$ quality, and consider the next discrete optimization task for

$$F(y_0, y_1, ..., y_{l-1}, y_l) = \sum_{i=1}^{m} \left| z_i - (y_{A^i(\mathbf{x}_i)} - y_{A^i(\mathbf{x}_i)-1})/2 \right| \rightarrow \min_{y_1, y_2, ..., y_{l-1}}, \quad (1)$$

$$\left| \{ z_j, j = 1,2,...,m : y_0 \le z_j \le y_1 \} \right| \ge 3, \quad (2)$$

$$\left| \{ z_j, j = 1,2,...,m : y_i < z_j \le y_{i+1} \} \right| \ge 3, \ i = 1,2,...,l-1, \quad (3)$$

$$y_0 = z_1, \ y_i < y_{i+1}, i = 0,1,...,l-1, \ y_l = z_m,$$
$$y_i \in \{z_3, z_4, ..., z_{m-3}\}, i = 1,2,...,l-1, \quad (4)$$

Restrictions (2-3) are the consequence of leave-one-out mode and classification method that were used. The optimal value l and change points $a_1 < a_2 < ... < a_{l-1}$ are calculated by solving task (1-4) for various l.

So, efficiency of task (1-4) solution depends highly from efficiency of $A^j \equiv A^j(a_1, a_2, ..., a_{l-1}), j = 1,2,...,m$ construction. Later we use the local approach for $F(y_0, y_1, y_2, ..., y_l)$ optimization where one logical classification model will be used with fast procedure of classification algorithm re-training for neighboring classification tasks.

2.2 Local Optimization

Consider the problem (1-4), where the function $f(\mathbf{x})$ (and the corresponding classification algorithm A) given by the current values $(y_0, y_1, ..., y_{l-1}, y_l)$. Since objects ordered by increasing z_i, takes place $y_t = z_{i_t}, 1 \le t \le l-1$. A scheme of standard local optimization of $F(y_0, y_1, ..., y_{l-1}, y_l)$, $y_0 = z_1, \ y_l = z_m,$ has been considered. Points $\{(y_0^*, y_1^*, ..., y_{l-1}^*, y_l^*)$, $y_j^* \in \{z_{i_j-1}, z_{i_j+1}\}$, $y_t^* = y_t$, $t \ne j$, $y_0^* = y_0, \ y_l^* = y_l\}, \ j = 1,2,...,l-1$, are called neighboring for $(y_0, y_1, ..., y_{l-1}, y_l)$ if conditions (2-4) are satisfied.

Starting with an arbitrary admissible $(y_0, y_1^{(0)}, ..., y_{l-1}^{(0)}, y_l)$ we browse all nothing more than $2(l-1)$ a neighboring admissible points and find minimum for $F(y_0, y_1, ..., y_{l-1}, y_l)$ in neighborhood of $(y_0, y_1^{(0)}, ..., y_{l-1}^{(0)}, y_l)$. Later, the procedure is repeated for point $(y_0, y_1^{(1)}, ..., y_{l-1}^{(1)}, y_l)$ that is point of F

minimum in neighbor of $(y_0, y_1^{(0)},..., y_{l-1}^{(0)}, y_l)$. Finiteness of local optimization follows from the finiteness of the set of all possible values of $F(y_0, y_1,..., y_{l-1}, y_l)$.

2.3 Supervised Classification Model

Classification algorithms have been considered, which are modifications of estimation calculation algorithms [9]. We describe the principle of the estimation calculation algorithms (ECA).

Let a set X of admissible objects $\mathbf{x} \in R^n$ has the form $X = \bigcup_{j=1}^{l} K_j, K_\nu \cap K_\mu = \emptyset, \quad \nu \neq \mu$. Given training sample $\{z_t, \mathbf{x}_t, t = 1,2,...,m\}$, where $z_t = j$ if $\mathbf{x}_t \in K_j$. Let $\mathbf{x} = (x_1, x_2,..., x_n) \in R^n$, and a training sample contains representatives of all classes. Denote $\tilde{K}_j = \{\mathbf{x}_i : \mathbf{x}_i \in K_j, \ i = 1,2,...,m\}$, $\left|\tilde{K}_j\right| \geq 1$. Let the system of supporting sets $\Omega_A = \{\Omega\}$, $\Omega \subseteq \{1,2,...,n\}$ of algorithm A is the fixed one. Some supporting set Ω defines a subset of features. The proximity of classified object $\mathbf{x}$ to some training object $\mathbf{x}_t$ by support set Ω is defined as

$$\mathrm{B}_\Omega(\mathbf{x}, \mathbf{x}_t) = \begin{cases} 1, & |x_i - x_{ti}| \leq \varepsilon_i, \forall i \in \Omega, \\ 0, & \text{otherwise.} \end{cases} \tag{5}$$

There are the numerical parameters $\varepsilon_i, i = 1,2,...,n$ in (5) set by the user or calculated as, for example, $\varepsilon_i = \frac{2}{m(m-1)} \sum_{\alpha,\beta=1,\alpha<\beta}^{m} \left|x_{\alpha i} - x_{\beta i}\right|$. The estimation $\Gamma_j(\mathbf{x})$ for class $K_j, j = 1,2,...,l$ is calculated for the object $\mathbf{x}$.

$$\Gamma_j(\mathbf{x}) = \frac{1}{\left|\tilde{K}_j\right|} \sum_{\mathbf{x}_i \in \tilde{K}_j} \sum_{\Omega \in \Omega_A} B_\Omega(\mathbf{x}, \mathbf{x}_i). \tag{6}$$

Estimation $\Gamma_j(\mathbf{x})$ characterizes the heuristic degree of proximity of the object $\mathbf{x}$ to the class K_j. Next, apply the decision rule in the space of estimates: the object $\mathbf{x}$ is classified by algorithm A as belonging to class K_j when $\Gamma_j(\mathbf{x}) > \Gamma_i(\mathbf{x}), \forall i \neq j$. Otherwise, the rejection is made. Usually, the set $\Omega_A = \{\Omega : |\Omega| = k\}$, $1 \leq k \leq n$, k - integer, or all possible subsets of $\{1,2,...,n\}$ are used as a system of supporting sets of classification algorithm.

Parameter k is the external one, usually $k = \left[\frac{n}{3}\right]$ is used. In [9] proved that $\Gamma_j(\mathbf{x}) = \frac{1}{\left|\tilde{K}_j\right|} \sum_{\mathbf{x}_i \in \tilde{K}_j} C^k_{d(\mathbf{x},\mathbf{x}_i)}$ in the first method of supporting sets choice, and $\Gamma_j(\mathbf{x}) = \frac{1}{\left|\tilde{K}_j\right|} \sum_{\mathbf{x}_i \in \tilde{K}_j} (2^{d(\mathbf{x},\mathbf{x}_i)} - 1)$ in the second case, where $d(\mathbf{x},\mathbf{x}_i) = \left|\{ j : \left|x_j - x_{ij}\right| \le \varepsilon_j,\ j = 1,2,...,n\}\right|$. In this study, we used some modification of the proximity function (5) and estimation (6).

Let $\mathbf{x}_\alpha, \mathbf{x}_\beta \in \tilde{K}_j$. Define the proximity function $\tilde{\mathrm{B}}_\Omega(\mathbf{x}_\alpha, \mathbf{x}, \mathbf{x}_\beta)$ of the object $\mathbf{x}$ to the couple $\mathbf{x}_\alpha, \mathbf{x}_\beta$, and its estimation $\tilde{\Gamma}_j(\mathbf{x})$ for the class K_j by the following expressions.

$$\tilde{\mathrm{B}}_\Omega(\mathbf{x}_\alpha, \mathbf{x}, \mathbf{x}_\beta) = \begin{cases} 1, & (x_{\alpha i} \le x_i \le x_{\beta i}) \vee (x_{\beta i} \le x_i \le x_{\alpha i}), \forall i \in \Omega, \\ 0, & \text{otherwise.} \end{cases} \tag{7}$$

$$\tilde{\Gamma}_j(\mathbf{x}) = \frac{2}{\left|\tilde{K}_j\right|\left(\left|\tilde{K}_j\right| - 1\right)} \sum_{\mathbf{x}_\alpha, \mathbf{x}_\beta \in \tilde{K}_j, \alpha < \beta} \left(\sum_{\Omega \in \Omega_A} \tilde{\mathrm{B}}_\Omega(\mathbf{x}_\alpha, \mathbf{x}, \mathbf{x}_\beta)\right).$$

It can be proved that here also we have similar effective formulas for calculating $\tilde{\Gamma}_j(\mathbf{x})$: $\tilde{\Gamma}_j(\mathbf{x}) = \frac{2}{\left|\tilde{K}_j\right|\left(\left|\tilde{K}_j\right| - 1\right)} \sum_{\mathbf{x}_\alpha, \mathbf{x}_\beta \in \tilde{K}_j, \alpha < \beta} C^k_{d(\mathbf{x}_\alpha, \mathbf{x}, \mathbf{x}_\beta)}$, and

$\tilde{\Gamma}_j(\mathbf{x}) = \frac{2}{\left|\tilde{K}_j\right|\left(\left|\tilde{K}_j\right| - 1\right)} \sum_{\mathbf{x}_\alpha, \mathbf{x}_\beta \in \tilde{K}_j, \alpha < \beta} (2^{d(\mathbf{x}_\alpha, \mathbf{x}, \mathbf{x}_\beta)} - 1)$, where

$d(\mathbf{x}_\alpha, \mathbf{x}, \mathbf{x}_\beta) = \left|\{t : (x_{\alpha t} \le x_t \le x_{\beta t}) \vee (x_{\beta t} \le x_t \le x_{\alpha t}), t = 1,2,...,n\}\right|$. Denote that here $\left|\tilde{K}_j\right| \ge 2$.

After calculating the estimations $\tilde{\Gamma}_j(\mathbf{x}), j = 1,2,...,l,$ the previously given decision rule is used. Note that any metric in feature space and parameters $\varepsilon_i, i = 1,2,...,n$ are not used in this case. Here $x_j \in M_j$ (M_j - linearly ordered set). The features may be ordinal. This algorithm does not contain parameters that require adjustment during training.

2.4 Recalculation of Classification Algorithm for Neighbor Samples

Denote $m_j = \left|\tilde{K}_j\right|, j = 1,2,...,l$. Consider the optimization task (1-4), when the classification of objects is carried out in a leave-one-out mode. The choice of change-points determines a partition of the training set for classification problems. Neighboring training samples (and the classification problems) can be obtained from

the original by deleting an object, or moving an object from one class to another. Efficiency of task (1-4) solution depends directly on quick recalculation of classification algorithm after transition to the neighbor sample.

In solving the problem (1-4) estimations $\widetilde{\Gamma}_j(\mathbf{x})$ are easily converted into a leave-one-out mode. Really, we calculate the matrices $\mathbf{D}^1 = \left\| D^1_{\alpha\gamma\beta} \right\|_{m\times m\times m}$, $D^1_{\alpha\gamma\beta} = \mathrm{C}^k_{d(\mathbf{x}_\alpha,\mathbf{x}_\gamma,\mathbf{x}_\beta)}$, $\mathbf{D}^2 = \left\| D^2_{\alpha\gamma\beta} \right\|_{m\times m\times m}$, $D^2_{\alpha\gamma\beta} = 2^{d(\mathbf{x}_\alpha,\mathbf{x}_\gamma,\mathbf{x}_\beta)} - 1$.

Let $\mathbf{x}_t$ is any object in the training sample, there is a current partition into classes $K_1, K_2, ..., K_l$, and $\mathbf{x}_t \in K_i$. Then in a leave-one-out mode $\widetilde{\Gamma}_i(\mathbf{x}_t) = \dfrac{2}{(m_i-1)(m_i-2)} \sum\limits_{\substack{\alpha<\beta:\alpha,\beta\neq t \\ \mathbf{x}_\alpha,\mathbf{x}_\beta\in K_i,}} D^h_{\alpha t\beta}$, $\widetilde{\Gamma}_j(\mathbf{x}_t) = \dfrac{2}{m_j(m_j-1)} \sum\limits_{\substack{\alpha<\beta: \\ \mathbf{x}_\alpha,\mathbf{x}_\beta\in K_j,}} D^h_{\alpha t\beta}, j \neq i \cdot$

Here $h \in \{1,2\}$. For simplicity, we omit here and further h in notations of $\widetilde{\Gamma}_j(\mathbf{x}_t)$. Consider the recalculation of estimations $\widetilde{\Gamma}_j(\mathbf{x}_t), j = 1,2,...,l$, during function (1) recalculation in arbitrary neighbor point on some step of optimization. In this case, the boundary between a pair of classes is changing as a result of the transfer of an object $\mathbf{x}_\tau$ from one class to the neighboring class. Denote the "new" classes as $K^*_1, K^*_2, ..., K^*_l$, and estimations for $\mathbf{x}_t$ through $\widetilde{\Gamma}^*_j(\mathbf{x}_t), j = 1,2,...,l$.

We have the following four possible variants:

1. $K^*_i = K_i \cup \{\mathbf{x}_\tau\}$, $\mathbf{x}_\tau \in K_u, u \neq i$,
   $K^*_u = K_u \setminus \{\mathbf{x}_\tau\}$, $K^*_j = K_j, j \neq i, u$.
2. $K^*_u = K_u \setminus \{\mathbf{x}_\tau\}$, $K^*_v = K_v \cup \{\mathbf{x}_\tau\}$, $K^*_j = K_j, j \neq u, v$, $u, v \neq i$.
3. $K^*_i = K_i \setminus \{\mathbf{x}_\tau\}, \tau \neq t$, $K^*_u = K_u \cup \{\mathbf{x}_\tau\}$, $K^*_j = K_j, j \neq i, u$.
4. $K^*_i = K_i \setminus \{\mathbf{x}_t\}$, $K^*_u = K_u \cup \{\mathbf{x}_t\}$, $K^*_j = K_j, j \neq i, u$.

Then estimations $\widetilde{\Gamma}^*_j(\mathbf{x}_t), j = 1,2,...,l$ are recalculated as follows:

1. $\widetilde{\Gamma}^*_i(\mathbf{x}_t) = \dfrac{2}{m_i(m_i-1)} \left(\dfrac{(m_i-1)(m_i-2)}{2} \widetilde{\Gamma}_i(\mathbf{x}_t) + \sum\limits_{\substack{\alpha:\alpha\neq t \\ \mathbf{x}_\alpha\in K_i, \\ \mathbf{x}_\tau\in K_u, u\neq i}} D^h_{\alpha t\tau} \right)$,

$\widetilde{\Gamma}^*_u(\mathbf{x}_t) = \dfrac{2}{(m_u-1)(m_u-2)} \left(\dfrac{m_u(m_u-1)}{2} \widetilde{\Gamma}_u(\mathbf{x}_t) - \sum\limits_{\substack{\alpha:\mathbf{x}_\alpha\in K_u, \\ \mathbf{x}_\tau\in K_u, u\neq i}} D^h_{\alpha t\tau} \right)$,

$$\tilde{\Gamma}_j^*(\mathbf{x}_t) = \tilde{\Gamma}_j(\mathbf{x}_t), j \neq i, u \quad .$$

$$2.\ \tilde{\Gamma}_u^*(\mathbf{x}_t) = \frac{2}{(m_u - 1)(m_u - 2)} \left(\frac{m_u(m_u - 1)}{2} \tilde{\Gamma}_u(\mathbf{x}_t) - \sum_{\substack{\alpha : \mathbf{x}_\alpha \in K_u, \\ \mathbf{x}_\tau \in K_u, u \neq i}} D^h_{\alpha t \tau} \right),$$

$$\tilde{\Gamma}_v^*(\mathbf{x}_t) = \frac{2}{(m_v + 1)m_v} \left(\frac{m_v(m_v - 1)}{2} \tilde{\Gamma}_v(\mathbf{x}_t) + \sum_{\substack{\alpha : \mathbf{x}_\alpha \in K_v, \\ \mathbf{x}_\tau \in K_u, u \neq i}} D^h_{\alpha t \tau} \right), u, v \neq i, \ \tilde{\Gamma}_j^*(\mathbf{x}_t) = \tilde{\Gamma}_j(\mathbf{x}_t), j \neq u, v \cdot$$

$$3.\ \tilde{\Gamma}_i^*(\mathbf{x}_t) = \frac{2}{(m_i - 2)(m_i - 3)} \left(\frac{(m_i - 1)(m_i - 2)}{2} \tilde{\Gamma}_i(\mathbf{x}_t) - \sum_{\substack{\alpha : \mathbf{x}_\alpha \in K_i, \\ \mathbf{x}_\tau \in K_i, \tau \neq t}} D^h_{\alpha t \tau} \right),$$

$$\tilde{\Gamma}_u^*(\mathbf{x}_t) = \frac{2}{(m_u + 1)m_u} \left(\frac{m_u(m_u - 1)}{2} \tilde{\Gamma}_u(\mathbf{x}_t) + \sum_{\substack{\alpha : \mathbf{x}_\alpha \in K_u, \\ \mathbf{x}_\tau \in K_i, \tau \neq t}} D^h_{\alpha t \tau} \right),$$

$$\tilde{\Gamma}_j^*(\mathbf{x}_t) = \tilde{\Gamma}_j(\mathbf{x}_t), j \neq i, u \cdot$$

$$4.\ \tilde{\Gamma}_i^*(\mathbf{x}_t) = \tilde{\Gamma}_i(\mathbf{x}_t), i = 1,2,...,l.$$

Thus, the calculation of the function (1) in the next point of general optimization algorithm is carried out effectively. The complexity of one optimization step will be $O(m^2)$.

3 Illustrations and Experiments

To date, the set of initial experiments have been carried out that have confirmed practical use of suggested algorithm. In this model, we assumed at first that the number l of change-points (number of classes in classification) is fixed. For function (1) calculation, the leave-one-out mode is used. The optimal number of change-points and their positions are computed by solving the problem (1-4) with various values of l. Finally, the efficiency of built piecewise constant dependence was estimated in leave-one-out mode.

As an illustrative example, consider the function $y(x) = \sin(x) + x/3$ on the interval $[1:25]$. To create a training set has been used 100 points $x_i = 1 + i/4, i = 0,1,...,99$. Figure 1 shows the function $z(x)$ and optimal piecewise constant functions obtained with the ECA. Corresponding to the ECA values of the functions (1) are denoted by the symbols «−». The values of the mean modulus error were found to be 0.63 for the linear regression, and 0.19 for ECA with $l = 32$.

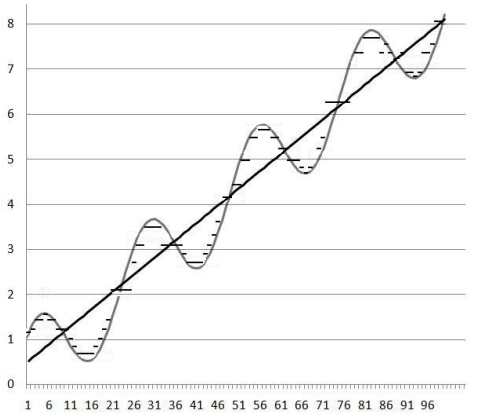

Fig. 1. Piece-wise constant function approximating $z(x) = \sin(x) + x/3$

Below is an example of constructing a piecewise constant function, according to data [10]. Considered the problem of automatic evaluation of the cost of housing in the suburbs of Boston. The training set consisted of 366 objects described in terms of 13 features (12 real and one binary). Figure 2 shows the distribution of housing costs in the axes of "number of the object - the cost of housing in thousands of dollars". Objects pre-ordered by increasing cost of housing. The real cost of housing is indicated by «▲». The calculated estimates of the cost of housing using the proposed in this work method are marked by symbols " _ ". Number of piecewise components found to be 32, the value of the mean modulus of error is 1.07. The average modulus error of linear regression was equal to 1.213.

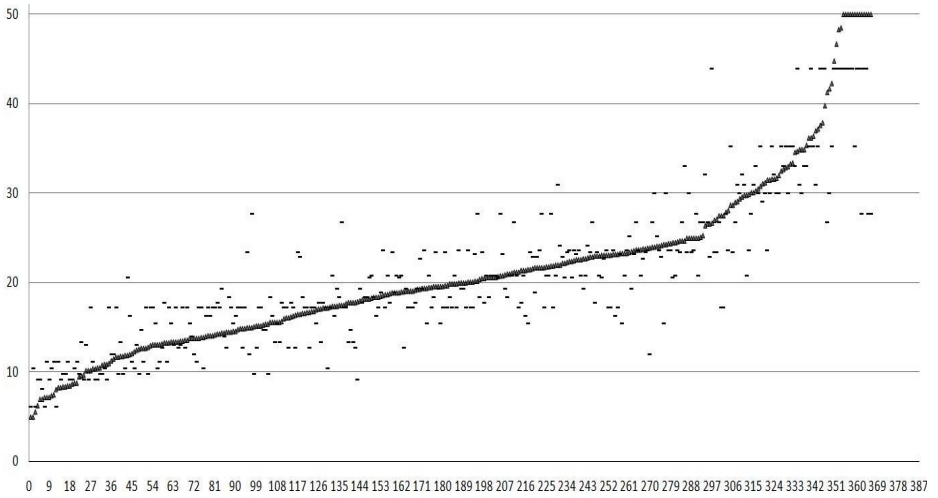

Fig. 2. Piece-wise constant function approximating the cost of housing in the suburbs of Boston

4 Conclusions

It is important to note the following details of this research.

1. Algorithms for reconstruction of the regression were considered for the case of numeric attributes $x_i \in R, i = 1,2,...,n$. It is easy to see that features may be dissimilar (numeric, binary, or ordinal) when we are constructing piecewise constant regressions by using the training samples. The proposed model of regressions restoration is based on the modification of the ECA, which does not require a metric in the feature space. The feature values are used only in (7) where the order relation for each feature is applied.
2. At the beginning of this paper, it was remarked some situations where the classical methods of regressions restoration are not applicable or restricted. It should be added the cases when the values of the dependent quantities are very unevenly shared, or dependent quantities are in fact the l-valued with large value of l. The most preferable case for supervised classification at a fixed length of the training sample is the case when number of classes is equal to 2. However, this case may not be optimal for the functional (7). Thus, it is expected that the model for constructing piecewise constant regressions will be helpful in addressing the many "bad " problems, inconvenient for standard regression approaches, and for classification tasks. In these cases, "bad" regression problem reduces to the problem of classification with an optimal choice of l.

Previously, we have considered the case $z = f(\mathbf{x}) = (a_i + a_{i-1})/2$. Of course, it can be used here other methods for $z = f(\mathbf{x})$ calculation by $a_1, a_2, ..., a_{l-1}$ (mean in I_i by training sample, median, etc.).

In future work a generalization of the approach is supposed to restore the piece-wise non-constant dependencies.

Acknowledgments. I would like to thank the postgraduate student of M.V.Lomonosov Moscow State University A.S. Schichko for experiments performed. This work was supported by RAS Presidium Program number 15, Program number 2 of Department of Mathematical Sciences of RAS, RFBR 12-01-00912, 11-01-00585, 12-01-90012-bel.

References

1. Draper, N., Smith, H.: Applied regression analysis. John Wiley & Sons, New York (1966)
2. Hardle, W.: Applied nonparametric regression. Cambridge University Press, Cambridge (1990)
3. Collobert, R., Bengio, S.: Support Vector Machines for Large-Scale Regression Problems. Journal of Machine Learning Research 1, 9/1/, 143–160 (2001)
4. Yu, J.-R., Tzeng, G.-H., Li, H.-L.: General fuzzy piecewise regression analysis with automatic change-point detection. Fuzzy Sets and Systems 119, 247–257 (2001)
5. Hutter, M.: Bayesian Regression of Piecewise Constant Functions. Technical Report IDSIA-14-05, Galleria 2, CH-6928 Manno-Lugano, Switzerland (2005)

6. Janssen, F., Fyurnkranz, J.: Heuristic Rule-Based Regression via Dynamic Reduction to Classification. In: Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence (IJCAI 2011), pp. 1330–1335 (2011)
7. Bibi, S., Tsoumakas, G., Stamelos, I., Vlahavas, I.: Regression via Classification applied on software defect estimation. Expert Systems with Applications 34, 2091–2101 (2008)
8. Binev, P., Cohen, A., Dahmen, W., DeVore, R., Temlyakov, V.: Universal Algorithms for Learning Theory. Part I: Piecewise Constant Functions. Journal of Machine Learning Research 6, 1297–1321 (2005)
9. Zhuravlev, Y.: Selected Scientific Publications, p. 420. M. Magistr Publishing (1998)
10. Harrison, D., Rubinfeld, D.L.: Hedonic prices and the demand for clean air. J. Environ. Economics & Management 5, 81–102 (1978)