APPROXIMATION OF MULTIDIMENSIONAL DEPENDENCY BASED ON AN EXPANSION IN PARAMETRIC FUNCTIONS FROM THE DICTIONARY

E.V. Burnaev, M.G. Belyaev, P.V. Prihodko
Kharkevich Institute for Information Transmission Problems RAS
Moscow, RUSSIA
e-mail: burnaev@iitp.ru, belyaevmichel@gmail.com, prikhodkop@gmail.com

Abstract

In the present work method for construction of approximation of unknown multidimensional dependency based on data sample is proposed. Approximation is constructed in the class of linear expansions in parametric functions from the dictionary. Parameters of the functions from the dictionary are estimated using gradient methods and expansion coefficients are calculated using adaptively regularized method of least squares. Regularization is used to increase the stability of iterative estimation of parameters. For additional improvement of stability/generalization ability of approximation specially developed method for boosting is used.

1 Introduction

The problem of determining the analytical description for a set of data arises in numerous sciences and applications. Examples of methods for construction of such analytical description (construction of approximating function, i.e. approximator) on basis of available data in the form of inputs and outputs are Artificial Neural Networks, Kriging etc [8].

However typical methods for construction of approximations have numerous shortcomings. For example, kriging methods are local by their nature and computationally intensive in case of high dimensional input or big size of learning sample etc.

The aim of the present work is to describe the general methodology for high dimensional function approximation (HDA) based on an expansion in the parametric functions from the dictionary.

2 Problem statement

In general the problem can be formulated as follows. Let $f(X)$ be some unknown function with input $X \in \mathbf{X} \subset \mathrm{R}^m$ and output $Y = f(X) \in \mathrm{R}^n$. Let $D_{learn} = \{(X_i, Y_i = f(X_i)), i = 1, 2, \ldots, N_{learn}\}$ be learning sample. The problem is to construct an approximation $\hat{Y} = \hat{f}(X) = \hat{f}(X|D_{learn})$ (approximator) for the initial dependence $Y = f(X)$ using the set D_{learn}.

If for all $X \in \mathbf{X}$ (not only for $X \in D_{learn}$) an approximate equality $f(X) \approx \hat{f}(X)$ takes place then the approximator $\hat{f}(X)$ is considered to be appropriate. This fact is approximately checked using independent testing sample.

Quality of the constructed approximation is estimated by the mean error of approximation $\varepsilon\left(\hat{f}|D_{test}\right) = \sqrt{\frac{1}{N_{test}}\sum_{i=1}^{N_{test}}\left\|Y_i - \hat{f}(X_i)\right\|_2^2}$ on the test set D_{test}.

3 Model of an approximator

We will model an unknown function globally using basis expansion in functions from the specified dictionary, i.e.

$$\hat{f}(X) = \sum_{j=1}^{p} \alpha_j \psi_j(X), \tag{1}$$

where $\psi_j(X)$, $j = 1, \ldots, p$ are some parametric functions. We will use three main classes of parametric functions, namely

1. Sigmoid basis function $\psi_j(X) = \sigma\left(\sum_{i=1}^{m} \beta_{j,i} x_i\right)$, where $X = (x_1, \ldots, x_m)$, $\sigma(x) = \frac{e^x - 1}{e^x + 1}$. In order to model sharp features of the function $f(x)$ different parametrization can be used, namely $\psi_j(X) = \sigma\left(\left|\sum_{i=1}^{m} \beta_{j,i} x_i\right|^{\sigma(\alpha_j)+1} \operatorname{sign}\left(\sum_{i=1}^{m} \beta_{j,i} x_i\right)\right)$, where parameter α_j is adjusted independently of the parameters $\beta_j = (\beta_{j,1}, \ldots, \beta_{j,m})$. The essence is that for big negative values of α_j the function $\psi_j(X)$ behaves like step function.

2. Radial basis functions $\psi_j(X) = \exp\left(-\|X - d_j\|_2^2 / \sigma_j^2\right)$.

3. Linear basis functions $\psi_j(X) = x_j$, $j = 1, 2, \ldots, m$, $X = (x_1, \ldots, x_m)$.

Thus, the index set $J = \{1, \ldots, p\}$ can be decomposed into tree parts $J = J_{lin} \cup J_{sigmoid} \cup J_{RBF}$, where $J_{lin} = \{1, \ldots, m\}$ corresponds to the linear part (linear basis functions), $J_{sigmoid}$ and J_{RBF} correspond to sigmoid and radial basis functions correspondingly.

4 Hybrid Learning Algorithm

Usually in order to fit the model of type (1) to the data simple gradient descent methods are used for adjustment of its coefficients. However, as it is well-known, such methods have slow convergence. In this section hybrid learning algorithm is described based on Regression Analysis and gradient optimization method so called Resilient Propagation [10] method. The use of Regression Analysis allows to considerably decrease the learning time.

Any iterative estimation algorithm (and proposed hybrid algorithm is not exception) should be initialized. Parameters of the radial basis functions are initialized in the first place. For this elastic net is used [8] which consistently selects input vectors from the train set D_{learn} which will be used as centers for radial basis functions. For initialization of parameters of sigmoid functions it is proposed to use well-known initialization method [7]. The number of sigmoid basis functions, i.e. the cardinality of the set $J_{sigmoid}$ can be estimated using statistical learning theory, see the description of algorithm in [4].

4.1 Iterative estimation algorithm

Let us now describe how Regression Analysis can be used to estimate the parameters of the model (1). Divide randomly the learning data set D_{learn} into train set D_{train} and validation set D_{val}. Obtained subsets D_{train} and D_{val} are used for parameter estimation of $\hat{f}(X)$ and control of generalization ability of $\hat{f}(X)$ correspondingly.

It is obvious that the minimization of the error $\varepsilon\left(\hat{f}|D_{train}\right)$ can be done explicitly over the parameters α_j, $j = 1, \ldots, p$ using least squares for some fixed values of the parameters of basis functions $\psi_j(X)$, $j = m+1, \ldots, p$.

Thus hybrid learning algorithm can be described as successive switching between calculation of optimal values of α_j, $j = 1, \ldots, p$ using ridge regression and adaptation of the parameters of basis functions $\psi_j(X)$, $j = m+1, \ldots, p$ using gradient descent Resilient Propagation method (see [10]). On each iteration of the algorithm the regularization parameter is increased if the condition number of the corresponding regression matrix is big. Otherwise the regularization parameter is decreased (see details of algorithm for adaptive regularization in [5]). If for the user-defined number of consecutive iterations the error on the set D_{val} does not decrease below its current minimal value the hybrid learning algorithm is stopped.

5 Ensemble of approximators

Proposed hybrid learning algorithm outperforms standard algorithms (see [8], [10]) in accuracy and time [1, 2, 3, 9, 5]. However the hybrid algorithm has typical shortcomings of standard learning algorithms, namely the division of D_{learn} into D_{train} and D_{val} is random etc. In order to smooth over the influence of these random effects we propose to construct ensemble on basis of specially elaborated algorithm of boosting [8] described as follows [6]:

1. Let the initial output of the ensemble equal to $F_0(X) = 0$.
2. For $B = 1, 2, \ldots$:
   a) Let us define a new sample $D_{learn,B} = \{y^B, X\}$, where $y^B = B \cdot y - (B-1) \cdot F_{B-1}(X)$ for $\{y, X\} \in D_{learn}$.
   b) Train the B-th approximator $f_B(X)$ of the ensemble using the sample $D_{learn,B}$ and the hybrid learning algorithm.
   c) The output of the ensemble is set to $F_B(X) = \frac{B-1}{B} \cdot F_{B-1}(X) + \frac{1}{B} \cdot f_B(X)$.
   d) The algorithm for ensemble construction is stopped if for the user-defined number of iterations the error $\varepsilon\left(\hat{f}|D_{learn}\right)$ does not decrease below its current minimal value. Otherwise continue.

6 Experimental comparison

Experimental comparison of the proposed approach with conventional methods (Artificial Neural Networks, Kriging etc.) can be found in [1, 2, 3, 9, 5, 6]. Testing of the

proposed method showed its superiority compared to methods mentioned above.

References

[1] Bernstein A.V., Burnaev E.V., Kuleshov A.P. (2008). On a methodology for constructing approximations of multidimensional dependences. *In Plenary and Selected Papers of the Fourth International Conference "Parallel Computations and Control Problems"*, October 27-29, 2008, Moscow, Russia. pp. 56-62. V.A. Trapeznikov Institute of Control Systems of RAS, Moscow.

[2] Burnaev E.V., Grihon S. (2009). Construction of the Metamodeles in Support of Stiffened Panel Optimization. *Extended abstracts of VI International Conference "Mathematical Methods in Reliability. Theory. Methods. Applications" (MMR-2009)*, June 22-29, Moscow, Russia. pp. 124-128.

[3] Burnaev E.V., Belyaev M.G., Prihodko P.V., Chernova S.S. (2009). Neural Approximation based on regression and boosting. *Extended abstracts of VI International Conference "Mathematical Methods in Reliability. Theory. Methods. Applications" (MMR-2009)*, June 22-29, Moscow, Russia. pp. 119-123.

[4] Burnaev E.V., Belyaev M.G., Lokot A.S. (2010). Model selection in function approximation problem based on statistical learning theory. *In the present book.*

[5] Burnaev E.V., Belyaev M.G. (2009). Adaptive regularization in the problem of approximation of multidimensional dependencies. *Proceedings of Conference "Information Technologies and Systems"*. Moscow, IITP, pp. 431-435. (In Russian)

[6] Burnaev E.V., Prihodko P.V. (2009). About boosting methods. *Proceedings Information technologies and systems (ITIS'09)*. Moscow, IITP. pp. 422-427. (In Russian)

[7] Drago G.P., Ridella S. (1992). Statistically controlled activation weight initialization (SCAWI). *IEEE Trans Neural Netw.* Vol. **3** (4). pp. 627-31.

[8] Hastie T., Tibshirani R., Friedman J. (2008). *The elements of statistical learning: data mining, inference, and prediction*. Springer.

[9] Kuleshov A.P., Bernstein A.V., Burnaev E.V. (2010). Invariant approximation and its application for integration of data-domain knowledge into metamodels. *In the present book.*

[10] Riedmiller M., Braun H. (1993). A direct adaptive method for faster backpropagation learning: the RPROP algorithm. *In Proceedings of the IEEE International Conference on Neural Networks (ICNN)*. Vol. **16**, pp. 586-591. Piscataway, NJ: IEEE.