Neurocomputing ∎ (∎∎∎∎) ∎∎∎–∎∎∎

Contents lists available at ScienceDirect

Neurocomputing

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

Feature vector regression with efficient hyperparameters tuning and geometric interpretation

Jie Liu [a], Enrico Zio [a,b,*]

[a] Chair on System Science and the Energetic Challenge, EDF Foundation, Laboratoire de Genie Industriel, CentraleSupélec, University of Paris-Saclay Paris, France

[b] Energy Department, Politecnico di Milano, Milano, Italy

ARTICLE INFO

Article history:
Received 16 September 2015
Received in revised form
30 March 2016
Accepted 24 August 2016
Communicated by: Sanguineti Marcello

Keywords:
Regression
Prediction
Computational complexity
Feature vector selection
Kernel method
Feature Vector Regression
Hyperparameters tuning

ABSTRACT

Machine learning methods employing positive kernels have been developed and widely used for classification, regression, prediction and unsupervised learning applications, whereby the estimate function takes the form of a weighted-sum kernel expansion. Unacceptable computational burden with large datasets and difficulty in tuning hyperparameters are usually the drawbacks of kernel methods. In order to reduce the computational burden, this paper presents a modified version of the Feature Vector Selection (FVS) method, proposing an approximation of the estimate function as a weighted sum of the predicted values of the Feature Vectors (FVs), where the weights are computed as the oblique projections of the new data points on the FVs in the feature space. Such approximation is, then, obtained by optimizing only the predicted values of the FVs. By defining a least square error optimization problem with equal constraints, analytic solutions of the predicted values of the FVs can be obtained. The proposed method is named Feature Vector Regression (FVR). The tuning of hyperparameters in FVR is also explained in the paper and shown to be less complicated than for other kernel methods. Comparisons with some other popular kernel methods for regression on several public datasets show that FVR, with a small subset of the training dataset (i.e. selected FVs), gives results comparable with those of the methods which give best results in terms of the prediction accuracy. The main contribution of this paper is the new kernel method (i.e. FVR), capable of achieving satisfactory results with reduced efforts because of the small number of hyperparameters to be tuned and the reduced training dataset size used.

1. Introduction

Because of their computational simplicity and good generalization performance, kernel methods have received much attention for regression [10,13,25,26], classification [20,31,37] and unsupervised learning [23,33,35]. Good and comprehensive reviews of these methods can be found in [16,27]. Focusing on regression and prediction, some popular kernel methods are Support Vector Machine (SVM) [1,6,34], Kernel Gaussian Process (KGP) [15,29,45], Kernel Ridged Regression (KRR) [11,12,40], Kernel Logistic Regression (KLR) [19,50], Kernel Principal Component Analysis (KPCA) [32,47].

The nonparametric and semi-parametric representer theorems given by Schölkopf et al. [36] show that for a large class of kernel algorithms, the minimum of the sum of an empirical risk term and a regularization term in a Reproducing Kernel Hilbert Space (RKHS) leads to optimal solutions for the estimate function that can be written as a kernel expansion on training data points. Specifically, in mathematical terms, the estimate function $f(\boldsymbol{x})$ of kernel methods, such as SVM, KGP, KRR, KLR and KPCA, can be formulated as

$$f(\boldsymbol{x}) = \sum_{i=1}^{N} \alpha_i k(\boldsymbol{x}_i, \boldsymbol{x}) + b, \tag{1}$$

where $k(\boldsymbol{x}_i, \boldsymbol{x}_j)$ is the inner product of the mapping of the data points $\boldsymbol{x}_i, \boldsymbol{x}_j$, $i, j = 1,2, \ldots, N$ in the high dimensional feature space, i.e. Reproduced Kernel Hilbert Space (RKHS), α_i, $i = 1,2, \ldots, N$ are the unknown weights to be optimized and b is a constant that can be zero or non-zero.

The unknowns α_i, $i = 1,2, \ldots, N$ and b in (1) have no physical meaning and their values are determined by a quadratic optimization. In the optimization, there are three types of hyperparameters: 1) the penalty factor C representing the trade-off between the empirical risk term and the regularization term, 2)

* Corresponding author at: Chair on System Science and the Energetic Challenge, EDF Foundation, Laboratoire de Genie Industriel, CentraleSupélec, University of Paris-Saclay Paris, France and Energy Departement, Politechnico di Milano, Milano, Italy.
E-mail address: enrico.zio@ecp.fr (E. Zio).

http://dx.doi.org/10.1016/j.neucom.2016.08.093
0925-2312/

hyperparameters related to the definition of the empirical risk term (e.g. the parameter ϵ in the ϵ-insensitive loss function of SVM) and 3) hyperparameters related to the kernel function itself (e.g. the parameter σ in the Gaussian Radial Basis kernel Function (RBF), $k(\boldsymbol{x}_i, \boldsymbol{x}_j)=e^{-\|\boldsymbol{x}\_i-\boldsymbol{x}\_j\|\wedge 2/(2\sigma\wedge 2)}$).

Drawbacks of previous kernel methods include the computational burden required for training on large datasets, the difficulty in tuning the hyperparameters and the difficulty of interpreting the resulting expansion model.

Various works have been proposed to address these drawbacks in the literature. Some approaches are proposed for easing the computational burden of SVM training, by reducing the number of training data points. They can be based on the characteristics of the inputs in RKHS, e.g. KPCA [51], Feature Vector Selection (FVS) [2], convex Hull vertices selection [17], Orthogonal Least Squares (OLS) regression [7], Minimum Enclosing Ball (MEB) [42], Sparse Online Gaussian Process (SOGP) [4], approximate extreme points [52], random features [53,54], or on the prediction accuracy, e.g. orthogonal least squares learning algorithm [8], Fisher Discriminant Analysis [34], significant vector learning [17], kernel F-score feature selection [28]. Methods like KPCA reduce the data size by combining explicitly the training data points, but the computation burden is not significantly decreased. All these methods use the same form of the estimate function (1) and the weights are some optimized empirical values, with no physical meaning.

In order to reduce the difficulty in tuning hyperparameters, Analytic Parameter Selection (APS) has been proposed to calculate the hyperparameters values directly from the training dataset [9]. A combination of APS and Genetic Algorithm (GA) has also been used, with superior prediction results [48]. Many optimization approaches, e.g. Particle Swarm Optimization (PSO) [22], Monte Carlo method (MC) [14], Particle Filtering (PF) [49], Competitive Agglomeration (CA) clustering [18], asymptotically optimal selection [39], have also been proposed to optimize the hyperparameters values. The computational burden is still a main obstacle for these latter approaches, whereas APS is computationally efficient but it cannot achieve satisfactory results, especially for the penalty parameter.

Although the possibility of using super-computers can alleviate the burden of tuning hyperparameters in many applications, it can still be beneficial to reduce the computational burden in practice, because super-computation may not be affordable for some applications.

In this paper, we propose an approximation of the estimate function in (1), based on a modified version of FVS. The proposed method is called FVR, whose unknown parameters are tuned with less computational burden than other methods because:

– in the optimization function, there is no hyperparameters related to the regularization term and the loss function;.

– the tuning of hyperparameters follows an iterative procedure, rather than the random process as in GA and some other methods.

Also, the proposed method reduces the size of the kernel expansion in the estimate function by selecting directly part of the training dataset, thus, reducing also the computational burden of the training and test processes, whereas KPCA, OLS and some other methods construct the kernel expansion which includes always all the training data points.

Finally, so far as the authors know, there have not been any new approaches proposed to tackle the interpretability of an SVM model. In this paper, by analyzing the distribution property of the inner product (as mentioned in relation to (1) above, the kernel function is an inner product of two vectors in RKHS) and the geometrical relation between a training data point and the FVs selected by FVS [2], the proposed method, i.e. FVR, is a geometrically interpretable kernel method, which describes the linear relation between the predicted values of FVs and that of any other data point. FVS selects the FVs which can represent the dimensions of the training dataset in RKHS, and the linear relations between the predicted value of the FVs and those of the other data points are derived from the general form (1) of the estimate function. In order to keep all the information contained in the selected FVs, an optimization problem with equal constraints (similar to a LS-SVM) is defined to find the minimal Mean Squared Error (MSE) (without regularization term) on the whole training dataset (not only on the selected FVs). Thus, in the proposed approach the unknowns in the estimate function are the predicted values of the FVs and a constant (zero or nonzero), which can be calculated analytically. The equal constraints in the optimization problem keep all the information in the FVs (i.e. no FV is ignored through the loss function, as in SVM).

Note that the Reduced Rank Kernel Ridge Regression (RRKRR) proposed in [5] already integrates FVS in a Least Square-Support Vector Machine (LS-SVM) to decrease the size of the training dataset and, thus, the computational complexity related to training. The differences between RRKRR and FVR lie in the objective function of the optimization and in the estimate function (1). The hyperparameters of FVR are less and more easy to be tuned. As a result, comparisons on several public datasets show that FVR performs better than RRKRR.

The comparisons with various popular kernel methods are also carried out. Considering prediction accuracy and computational burden show that FVR gives comparable results with the best prediction results of benchmarks. The experiment results show that minimizing the MSE on the whole training dataset of the kernel model built on the selected FVs can guarantee the generalization performance of the model, even without a regularization term. An efficient method for tuning hyperparameters is also proposed.

The structure of the paper is as follows. Section 2 gives a brief introduction to FVS and the derivation of FVR is also given in this section, with analytic solutions for the unknowns. Prediction results and comparisons with several popular kernel methods are illustrated in Section 3. Some conclusions and perspectives are drawn in Section 4.

2. Feature vector regression (FVR)

In this Section, a brief introduction of the FVS in [2] is firstly given with attention to its geometrical interpretation and FVR is, then, derived from (1). An optimization problem is defined to calculate analytically the unknown parameters in FVR. Insightful considerations on the optimization problem are provided.

2.1. Feature vector selection

FVS proposed in [2] aims at selecting a number of data points (which are called Feature Vectors (FVs)) $\boldsymbol{S}=(\boldsymbol{x}_i, y_i), i = 1,2, \ldots, M$, from the training dataset $\boldsymbol{T}=(\boldsymbol{x}_i, y_i), i = 1,2, \ldots, N$, with $M \leq N$, such that the other data points can be expressed as a linear combination of the selected FVs in RKHS. Let us denote by $\boldsymbol{\varphi}(\boldsymbol{x})$ the function which maps each training data point $\boldsymbol{x}_i$ into a high dimensional RKHS and by $k(\boldsymbol{x}_i, \boldsymbol{x}_j)$ the kernel function, defined as the inner product $\langle \boldsymbol{\varphi}(\boldsymbol{x}_i), \boldsymbol{\varphi}(\boldsymbol{x}_j) \rangle$ in RKHS. For a given $\boldsymbol{x}$, the Local Fitness (LF) with respect to the feature space $\boldsymbol{S}$ is calculated as:

Please cite this article as: J. Liu, E. Zio, Feature vector regression with efficient hyperparameters tuning and geometric interpretation, Neurocomputing (2016), http://dx.doi.org/10.1016/j.neucom.2016.08.093