Pattern Recognition Letters 69 (2016) 56–61

Contents lists available at ScienceDirect

Pattern Recognition Letters

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

Kernel subspace pursuit for sparse regression☆,☆☆

Jad Kabbara, Ioannis N. Psaromiligkos*

Department of Electrical and Computer Engineering, McGill University, 3480 University Street, Montreal, QC, H3A 0E9, Canada

ARTICLE INFO

Article history:
Received 15 December 2014
Available online 22 October 2015

Keywords:
Kernel methods
Sparse function approximation
Regression
Subspace pursuit

ABSTRACT

Recently, results from sparse approximation theory have been considered as a means to improve the generalization performance of kernel-based machine learning algorithms. In this paper, we present Kernel Subspace Pursuit (KSP), a new method for sparse non-linear regression. KSP is a low-complexity method that iteratively approximates target functions in the least-squares sense as a linear combination of a limited number of elements selected from a kernel-based dictionary. Unlike other kernel methods, by virtue of KSP's algorithmic design, the number of KSP iterations needed to reach the final solution does not depend on the number of basis functions used nor that of elements in the dictionary. We experimentally show that, in many scenarios involving learning synthetic and real data, KSP is less complex computationally and outperforms other kernel methods that solve the same problem, namely, Kernel Matching Pursuit and Kernel Basis Pursuit.

1. Introduction

For decades, non-linear regression models have been extensively studied in the area of statistics, econometrics and machine learning. Quite often these models involve a non-linear transformation of data into a high-dimensional space in which linear regression models are expected to be more accurate compared to the original space. Examples include Artificial Neural Networks [9], Decision Trees [9] and Support Vector Machines (SVMs) [3].

An important family of non-linear regression methods is kernel methods [19] that have received major attention in the past two decades, as they allowed non-linear versions of conventional linear supervised and unsupervised learning algorithms, yielding impressive regression performance. Using the kernel trick, interesting "kernelized" extensions of many well-known algorithms were presented, including kernel SVMs [19], kernel Principle Component Analysis (PCA) [18] and kernel Fisher discriminant analysis [11].

The generalization, or "out-of-sample," performance of a learning method, including kernel methods considered in this work, quantifies its ability to predict on independent never-seen-before data. From a learning-theoretic perspective, controlling the "capacity" of learning algorithms is necessary to guarantee good generalization performance [23]. This in turn is related to the issue of overfitting: the more complex a model is, the more likely it is to overfit the data. Traditional non-parametric kernel regression methods such as the Nadaraya-Watson method [12,25] affect generalization performance through tuning parameters (such as the kernel bandwidth), which could be viewed as a way of controlling model complexity. Recently, results from sparse approximation theory [7] have been considered as another means to directly control the model complexity and, consequently, limit overfitting. Sparse approximation refers to estimating a vector (or function) as a linear combination of a small number of elements selected from a larger set, called dictionary, of vectors (or functions). In our regression context, we can control model complexity by restricting the regression function to be sparse, i.e., that it is a linear combination of a fixed (but small) number of functions selected from a given dictionary.

1.1. Related work

Previous work has discussed interesting connections between non-linear kernel-based learning and pursuit algorithms [10]. Pursuit algorithms, are a family of greedy, iterative approaches to obtain sparse approximations of a function. Poggio and Girosi relate in their work [15] the basis pursuit algorithm [2] to kernel SVMs. The work of Smola and Schölkopf [20] presents ties between the Matching Pursuit (MP) algorithm [10] and kernel PCA, and shows how such ties can be used to compress the kernel matrix in SVMs to allow dealing with large datasets. Also, Smola and Bartlett present in [21] a greedy MP-like technique that approximates Maximum A Posteriori (MAP) estimates of Gaussian Processes by expressing the MAP estimate as an expansion in terms of a small subset of pre-specified kernel functions.

☆ This paper has been recommended for acceptance by G. Moser.

☆☆ This work was supported in part by the Natural Sciences and Engineering Research Council (NSERC), grant RGPIN 262017.

* Corresponding author. Tel.: +1 514 398 2465; fax: +1 514 398 4470.
E-mail addresses: jad.kabbara@mail.mcgill.ca (J. Kabbara), yannis@ece.mcgill.ca (I. N. Psaromiligkos).

http://dx.doi.org/10.1016/j.patrec.2015.09.018
0167-8655/

Following this previous work, Kernel Matching Pursuit (KMP) [24] and Kernel Basis Pursuit (KBP) [8] were presented as kernel methods that learn target functions by means of sparse approximation. KMP adopts a greedy suboptimal iterative approach to construct a sparse linear approximation of the target regression function. It starts with the approximation being initialized to zero, and then builds it by adding to it, at each iteration, a new term consisting of an appropriately weighted function from the dictionary. The function is chosen according to a correlation-based criterion and then, the corresponding weight is computed so that the approximation error at that iteration is minimized. KBP, on the other hand, solves a relaxed version of the same problem by incorporating ℓ_1-regularization on the minimization of the approximation error. In doing so, KBP controls the sparsity of the solution. The problem formulation in KBP corresponds to the well-known Least Absolute Shrinkage and Selection Operator (LASSO) formulation [22] in the feature space which combines an ℓ_2-loss function (squared error) with ℓ_1-regularization. While previous work (e.g., [2] that inspired KBP) considered finding the optimal solution to the minimization problem through costly and complex linear programming techniques, KBP uses the Least Angle Regression (LARS) technique [6] which also finds the exact solution of the LASSO but in an iterative and efficient way.

In summary, KBP and KMP attempt to solve similar problems while addressing the sparsity of the solution in different ways. Indeed, KMP guarantees that the solution is K-sparse by imposing a pre-specified finite number K of basis functions that will be used to construct the approximation function. On the other hand, KBP, in its original formulation, uses a regularization term that controls the sparsity of the solution. In addition, both KMP and KBP suffer, computationally, from the same drawback: the number of iterations that they have to run directly depends on the intended number of basis functions in the final solution.

Other work in the literature has addressed similar problems. In [16], a stochastic version of KMP is presented for large datasets. In this sub-optimal version of KMP, at a given iteration, the search space from which basis functions are selected is reduced. More specifically, a basis function is selected from a randomly chosen subset of the available basis functions. The work in [13] presents a family of greedy algorithms for building sparse kernel-based regression and classification models. An ℓ_2-loss function is iteratively minimized until a specified stopping criterion is satisfied. Different greedy criteria for basis selection from the literature are discussed and two numerical schemes are presented for updating the weights and residue (approximation error), the first based on residual minimization and the other based on QR factorization.

1.2. Main contribution

Our research effort aims at identifying an alternative framework to KMP and KBP, one that would: (1) address the need to dissociate (to the extent possible) kernel-based learning algorithms from the computational constraints from which KMP and KBP suffer (i.e., the dependency of the number of iterations on the number of basis functions), and (2) still provide us with the control of the sparsity of the solution. Thus, we apply the Subspace Pursuit (SP) algorithm of Dai et al. [4] to non-linear regression problems, and introduce the Kernel SP (KSP) algorithm. SP was first introduced in the context of compressive sensing. It was originally proposed as an iterative method for the reconstruction of an unknown sparse signal from a set of linear measurements. In our work, we capitalize on the fact that SP is, in essence, a low complexity method to obtain least-squares solutions with a pre-specified sparsity level.

As in KMP and KBP, the proposed algorithm iteratively learns a regression function with a predefined sparsity level. In contrast to both KMP and KBP that start by initializing the regression function to zero, and then iteratively expand it until it reaches the desired sparsity level, KSP always maintains an estimate of the regression function built using the pre-specified number K of dictionary elements, and refines the estimate through a usually limited number of iterations. To build the regression function, KMP and KBP (using the LARS implementation) need a number of iterations equal to the desired number of basis functions in the expansion. However, the number of KSP iterations needed to reach the final solution does not depend on the required number of basis functions and is normally smaller than in KMP and KBP. We experimentally show that in various scenarios that involve learning synthetic and real data, this required number of KSP iterations is indeed much smaller than that required for KMP and KBP, and that in many of these scenarios, KSP is less computationally intensive than KMP and KBP. We further present experimental validation that shows that, in most of these learning scenarios, KSP outperforms both KMP and KBP in the task of learning real and synthetic data.

The remainder of this paper is organized as follows: In Section 2, we formulate the problem considered in this work. In Section 3, we introduce the Kernel Subspace Pursuit algorithm. In Section 4, we compare the computational overhead for running KSP, KMP and KBP in various scenarios involving learning synthetic and real data. We also present the results of simulations showing that our algorithm outperforms both KMP and KBP in most of these learning scenarios. Finally, Section 5 concludes the paper.

2. Problem formulation

We are given L noisy observations $\{y_1, \ldots, y_L\}$ of an unknown target function $f : \mathbb{R}^d \mapsto \mathbb{R}$ at the inputs $\{\mathbf{x}_1, \ldots, \mathbf{x}_L\}$ where $\mathbf{x}_i \in \mathbb{R}^d$ (for a given d) and $y_i \in \mathbb{R}, \forall i$. Let $k : \mathbb{R}^d \times \mathbb{R}^d \mapsto \mathbb{R}$ be a positive definite kernel and let $\mathcal{H}$ be the associated kernel Hilbert space whose norm is denoted by $\|\cdot\|_{\mathcal{H}}^2$. We are interested in identifying a function $\hat{f} \in \mathcal{H}$ that is a good (in some sense) approximation of f. According to the Representer Theorem [17], given a strictly increasing function $\Omega : [0, \infty) \mapsto \mathbb{R}$ and an arbitrary cost function $c : (\mathbb{R}^d \times \mathbb{R}^2)^L \mapsto \mathbb{R} \cup \{\infty\}$, any minimizer $\hat{f} \in \mathcal{H}$ of the regularized cost $c((\mathbf{x}_1, y_1, \hat{f}(\mathbf{x}_1)), \ldots, (\mathbf{x}_L, y_L, \hat{f}(\mathbf{x}_L))) + \Omega(\|\hat{f}\|_{\mathcal{H}}^2)$ has the form

$$\hat{f}(\mathbf{x}) = \sum_{i=1}^{L} \alpha_i k(\mathbf{x}, \mathbf{x}_i), \ \ \alpha_i \in \mathbb{R}. \tag{1}$$

That is, $\hat{f}$ can be written as a linear combination of the elements of the set $\mathcal{G}$ containing L functions in $\mathcal{H}$:

$$\mathcal{G} = \{g_i = k(\cdot, \mathbf{x}_i) | i = 1, \ldots, L\} \subset \mathcal{H}. \tag{2}$$

Borrowing terminology from Sparse Approximation theory [7], we refer to $\mathcal{G}$ as the dictionary and to its elements as atoms.

In this paper, the goal is to approximate f using a given number $K < L$ of dictionary atoms, called basis functions, that is, we want our solution to admit the form in (1) but we now add the constraint that only K of the coefficients α_i are non-zero. In other words, we are interested in constructing an approximation $\hat{f}_K$ of f as a linear combination of K atoms $g_{\gamma_i}, i = 1, \ldots, K$:

$$\hat{f}_K(\mathbf{x}) = \sum_{i=1}^{K} \alpha_i g_{\gamma_i}(\mathbf{x}) = \sum_{i=1}^{K} \alpha_i k(\mathbf{x}, \mathbf{x}_{\gamma_i}) \tag{3}$$

where $\{\gamma_1, \ldots, \gamma_K\}$ are the indices of the selected atoms and $\alpha_1, \ldots, \alpha_K$ are the corresponding coefficients. Given that K is smaller than L, we talk of a sparse approximation of f.

Let $\hat{\mathbf{y}}$ be the vector consisting of the evaluation of $\hat{f}_K$ at the L input vectors, i.e., $\hat{\mathbf{y}} = [\hat{f}_K(\mathbf{x}_1), \ldots, \hat{f}_K(\mathbf{x}_L)]^T$. We also define the residue as the approximation error between the target vector $\mathbf{y} = [y_1, \ldots, y_L]^T$ and $\hat{\mathbf{y}}$, i.e.,

$$\mathbf{r} = \mathbf{y} - \hat{\mathbf{y}}. \tag{4}$$