Improving the kernel regularized least squares method for small-sample regression

Neurocomputing, Amsterdam, v. 163, p. 106-114, Sep. 2015
http://www.producao.usp.br/handle/BDPI/50885

Downloaded from: Biblioteca Digital da Produção Intelectual - BDPI, Universidade de São Paulo
Improving the kernel regularized least squares method for small-sample regression

Igor Braga, Maria Carolina Monard

Institute of Mathematics and Computer Science, University of São Paulo, Av. Trabalhador São-carlense, 400, São Carlos–SP 13566-590, Brazil

ARTICLE INFO

Article history:
Received 13 February 2014
Received in revised form 30 November 2014
Accepted 10 December 2014
Available online 4 April 2015

Keywords:
Non-linear regression
Kernel regularized least squares
Cross-validation
Kernel
Spline kernel
Parameter selection

ABSTRACT

The kernel regularized least squares (KRLS) method uses the kernel trick to perform non-linear regression estimation. Its performance depends on proper selection of both a kernel function and a regularization parameter. In practice, cross-validation along with the Gaussian RBF kernel have been widely used for carrying out model selection for KRLS. However, when training data is scarce, this combination often leads to poor regression estimation. In order to mitigate this issue, we follow two lines of investigation in this paper. First, we explore a new type of kernel function that is less susceptible to overfitting than the RBF kernel. Then, we consider alternative parameter selection methods that have been shown to perform well for other regression methods. Experiments conducted on real-world datasets show that an additive spline kernel greatly outperforms both the RBF and a previously proposed multiplicative spline kernel. We also find that the parameter selection procedure Finite Prediction Error (FPE) is a competitive alternative to cross-validation when using the additive splines kernel.

© 2015 Elsevier B.V. All rights reserved.

1. Introduction

Non-linear regression estimation is an important scientific modeling tool. Several methods have been proposed to tackle this estimation problem, with the most flexible and powerful ones falling in the category of the so-called kernel methods [1]. Among those is the kernel regularized least squares (KRLS) method [2–4], which enjoys good statistical and computational properties.

In a nutshell, the kernel regularized least squares method works as follows. Using a sequence of training data

\[(x_1, y_1), \ldots, (x_n, y_n), \quad x \in \mathbb{R}^d, y \in \mathbb{R},\]

where \(\gamma > 0\) is a real-valued regularization parameter and \(\mathcal{H}_K\) is a Reproducing Kernel Hilbert Space (RKHS) induced by a kernel \(K\). A function \(f \in \mathcal{H}_K\) with bounded \(\|f\|\) satisfies some regularity properties (e.g., smoothness), hence the use of the term “regularized” to name the method.

In order to apply KRLS successfully, that is, to use the obtained \(f_{\gamma}(x)\) to predict the output \(y\) of unseen \(x\), we must find such \(f_{\gamma}\) that (1) fits the training sequence well (i.e., minimizes the squared loss) and (2) is a reasonably smooth function (i.e., minimizes the norm \(\|f\|\)). As Statistical Learning Theory dictates [5], one can always minimize the former at the expense of the latter, and vice versa. Therefore, proper selection of both the kernel \(K\) and the regularization parameter \(\gamma\) is indispensable for the generalization performance of KRLS.

Formally, the best choice of \(K\) and \(\gamma\) is the one that yields in Expression (2) a function \(f_{\gamma}\), that minimizes the risk of prediction error as measured by the expected squared loss

\[R(f) = \int (y-f(x))^2 \, dP(x,y).\]  

The minimum of the functional \(R(f)\) is attained at the regression function [5, Chapter 1]. Thus, the closer \(R(f_{\gamma})\) is to the minimum of \(R(f)\), the closer the outputs of \(f_{\gamma}\) are to those of the real

regression function.

The choice of suitable \(K\) and \(\gamma\) belongs to the category of problems known as model selection. In contrast to the related category of model assessment, model selection does not require the estimation of the value of the prediction error \(R(f)\). It suffices to indicate the function with the smallest \(R(f)\) among a set of candidate functions \(f_1, f_2, \ldots, f_N\).

In practice, the value of \(R(f)\) cannot be calculated because \(P(x,y)\) is unknown. A widely employed workaround in this case is

E-mail addresses: igorab@icmc.usp.br, igor.braga@bigdata.inf.br (I. Braga), mcmonard@icmc.usp.br (M.C. Monard).

http://dx.doi.org/10.1016/j.neucom.2014.12.097
0925-2312/© 2015 Elsevier B.V. All rights reserved.
to use available data in a cross-validation setting, that is, to use some portion of the data to perform the minimization of Expression (2) for several candidates of $K$ and $\gamma$, and to reserve the other portion for approximating $\mathcal{R}(f)$ and selecting the best $K$ and $\gamma$. Conducting cross-validation in KRLS is relatively inexpensive compared to other learning methods, and this corresponds to the most interesting property of the method.

Given the universal approximation properties of the Gaussian RBF kernel [6]

$$k(x_i, x_j) = \exp \left( -\frac{\|x_i - x_j\|^2}{\sigma^2} \right). \tag{4}$$

it has become the kernel of choice in much of machine learning research. However, these nice theoretical properties of the RBF kernel do not extend very well to practice. When combined with cross-validation and small training sets, RBF kernels have a great potential for overfitting. Recently, there has been a renewed interest in developing kernels with less potential for overfitting while retaining a good approximation property [7].

The kernel regularized least squares method is computationally efficient in small sample situations, although it may be rendered ineffective by the issues plaguing the popular combination of cross-validation and RBF kernels. Having that in mind, in this paper we follow [7] and investigate the use of splines as a safer choice to compose a multidimensional kernel function. We go one step further in this work and propose the use of additive spline kernels instead of multiplicative ones. We have found experimental evidence that the additive version is more appropriate to regression estimation in small sample situations.

We then proceed by investigating alternative statistical and heuristic procedures for the selection of the regularization parameter $\gamma$. The procedures we consider were shown to perform well for other regression methods, and, to the best of our knowledge, have not been applied to KRLS before. Surprisingly, though, most of these procedures fail to outperform cross-validation in small sample situations. A notable exception is the Finite Prediction Error (FPE) method, which has performed as well as cross-validation when both were used in combination with the additive spline kernel.

The remainder of this paper is organized as follows. In Section 2 we show how the minimization problem in Expression (2) is solved for fixed $K$ and $\gamma$. In Section 3 we describe the issues surrounding the choice of a kernel function and present arguments in defense of the additive spline kernel. In Sections 4 and 5 we describe statistical and heuristic procedures used in this work to perform parameter selection, starting with an explanation on how to efficiently conduct leave-one-out cross-validation in KRLS. In Section 6 we report experimental evidence in favor of the additive spline kernel and also the results of the experimental evaluation of the considered parameter selection procedures. We conclude and give indications of future work in Section 7.

### 2. Solving the minimization problem of KRLS

The content in this section is informational and also introduces some notation used afterwards. To start with, note that KRLS requires the choice of a symmetric, positive definite kernel function $k : (\mathbb{R}^d \times \mathbb{R}^d) \rightarrow \mathbb{R}$ that spans the set of functions $\mathcal{R}_K$. Under consideration. An example of such function is the well-known Gaussian RBF kernel—Expression (4). In this section, we assume that a kernel function $k(x, x)$ has already been chosen, including eventual parameters.

By the representer theorem [8], the minimizer in Expression (2) has an expansion of the form

$$f(x) = \sum_{i=1}^{n} \alpha_i k(x_i, x), \quad \alpha_i \in \mathbb{R}. \tag{5}$$

Hereafter, we denote by $y$ the $n \times 1$ vector $[y_1, \ldots, y_n]^T$ and by $K$ the $n \times n$ matrix with entries $k_{ij} = k(x_i, x_j)$. We also denote by $\alpha$ the $n \times 1$ vector $[\alpha_1, \ldots, \alpha_n]^T$.

Plugging Expression (5) into Expression (2) yields the following expression for calculating the squared loss:

$$\frac{1}{n} \sum_{i=1}^{n} (y_i - f(x_i))^2 = \frac{1}{n} \alpha^T K K \alpha - 2 \frac{2}{n} \alpha^T K y + \text{const.} \tag{6}$$

Moreover, by considering the special properties of the norm in an RKHS, we have that $\|f\|^2 = \alpha^T K \alpha$. Ignoring the constant term in Expression (6), we arrive at the following quadratic minimization problem for Expression (2):

$$\alpha_\gamma = \arg \min_{\alpha \in \mathbb{R}^n} \left[ \frac{1}{n} \alpha^T K K \alpha - \frac{2}{n} \alpha^T K y + \gamma \alpha^T K \alpha \right]. \tag{7}$$

A necessary and sufficient condition for the solution of this minimization problem is obtained by taking the derivative of Expression (7) with respect to each $\alpha_i$ and equating it to zero. By doing that, we arrive at the following system of linear equations:

$$\frac{2}{n} K \alpha_\gamma - \frac{2}{n} K y + 2 \gamma \alpha_\gamma = 0. \tag{8}$$

Denoting by $I$ the $n \times n$ identity matrix, extracting 1/n from $\gamma$, and solving for $\alpha_\gamma$ in Expression (8), we arrive at the solution of the minimization problem in Expression (7):

$$\alpha_\gamma = (K + \gamma I)^{-1} y. \tag{9}$$

Plugging (9) into Expression (5) yields the closed form expression for the function minimizing Expression (2).

Most model selection procedures require the calculation of $\alpha_\gamma$ for a fair number of $\gamma$ candidates. In order to avoid solving one system of linear equations for each new $\gamma$, one can take advantage of the eigendecomposition of the kernel matrix: $K = U \Sigma U^T$, where $U$ is the $n \times n$ matrix formed by the eigenvectors of $K$ and $\Sigma$ is the $n \times n$ diagonal matrix containing the eigenvalues $\sigma_i$ of $K$. Denoting by $A_\gamma$ the $n \times n$ diagonal matrix with entries $A_{ij} = 1/((\sigma_i + \gamma)^2)$, $\alpha_\gamma$ can be calculated by performing only matrix multiplications

$$\alpha_\gamma = U A_\gamma U^T y. \tag{10}$$

Both the eigendecomposition of a matrix and a typical algorithm for solving a dense system of linear equations can be carried out in $O(n^3)$ time, with smaller constants for the latter. However, the eigendecomposition may still be preferable depending on $n$ and the number of $\gamma$ candidates considered.

### 3. Choosing a kernel function for KRLS

The choice of a kernel function for kernel regularized least squares defines the set of functions where the minimization of Expression (2) occurs. For example, if a linear kernel $k(x_i, x_j) = x_i^T x_j$ is chosen, then the function obtained by KRLS will be a hyperplane in the input space, which is enough for learning linear regressions. However, the regression function is not linear in the input space. A typical non-linear kernel is the widely used Gaussian Radial Basis Function (RBF)—Expression (4). In fact, this expression defines a family of kernel functions parameterized by $\sigma > 0$, the so-called width parameter. By controlling $\sigma$, it is possible to achieve universal
approximation of continuous functions [6]. However, at least two problems occur due to this extra parameter: (1) the need for selecting it properly for obtaining good generalization and (2) the potential for overfitting in small sample situations, as the set of induced functions is very flexible. These problems also affect polynomial kernels, since by choosing to use high-order polynomials, (by controlling the degree parameter), one can easily overfit the training data.

The issues plaguing RBF and polynomial kernels have led researchers to carry out regression estimation based on splines [2], which are piecewise polynomial functions. The underlying idea is to approximate a complicated regression locally, using a low-order polynomial in each segment of the function. By doing that, we avoid working with higher-order polynomials, decreasing the risk of overfitting, while retaining good approximation properties.

It is possible to define a kernel for generating splines in one dimension [5, Chapter 11]. Denoting $x_i \wedge x_j = \min(x_i, x_j)$,

the expression

$$k_i(x_i, x_j) = 1 + x_i x_j + \frac{1}{2} |x_i - x_j| (x_i \wedge x_j)^2 + \frac{1}{3} (x_i \wedge x_j)^3$$

defines a spline kernel for positive values of $x_i$ and $x_j$. The functions generated by this kernel in KRLS are twice differentiable at the training points.

The problem now is how to extend the one-dimensional spline to the multidimensional case. In Ref. [7], a multidimensional spline kernel was proposed as the multiplication of one-dimensional kernels (11) corresponding to each dimension of the data. The authors also propose normalizing it to mitigate (but not avoid) numerical difficulties. The expression for this multiplicative spline kernel is

$$k_{\text{mul}}^d(x_i, x_j) = \prod_{k=1}^d \frac{k_i(x_i^k, x_j^k)}{\sqrt{k_{\text{add}}(x_i^k, x_j^k) k_{\text{add}}(x_i^k, x_j^k)}}.$$

In Ref. [7], the kernel in Expression (12) was experimentally evaluated only on SVM classification. In this work, we evaluate it on KRLS regression. We also investigate the normalized spline kernel, which is a sum over the normalized one-dimensional kernels corresponding to each dimension of the data:

$$k_{\text{add}}^d(x_i, x_j) = \sum_{k=1}^d \frac{k_i(x_i^k, x_j^k)}{\sqrt{k_{\text{add}}(x_i^k, x_j^k) k_{\text{add}}(x_i^k, x_j^k)}}.$$

Theoretically, the set of functions spanned by $k_{\text{add}}^d$ is less diverse than $k_{\text{mul}}^d$. This fact makes the additive spline kernel well-suited for small sample regression. As an analogy, consider both linear and polynomial kernels. The former spans a set of functions which is less diverse than the latter. This is because the linear kernel is just the sum of coordinate-wise products, while a polynomial kernel includes products across coordinates.

Besides the aforementioned advantage, using $k_{\text{add}}^d$ instead of $k_{\text{mul}}^d$ is numerically safer. In particular, Expression (13) enables us to use a spline kernel in high dimensions, whereas Expression (12) would easily fall out of the precision range of standard floating-point arithmetic.

### 4. Statistical model selection

In this section we focus on statistical procedures of model selection. These methods can be used to bound or estimate the value of the prediction error $R(f)$ of a function $f(x)$

$$R(f) = \int (y - f(x))^2 \, dp(x, y).$$

Even though these statistical procedures can be used to perform model assessment, recall that only their model selection capabilities are of interest in this paper.

#### 4.1. Leave-one-out cross-validation

Cross-validation (CV) is one of the earliest model selection/assessment techniques that appeared in machine learning [9], and it is still the most frequently used one. The idea is to allow the same dataset to be used for learning and evaluation.

Here we focus on the special case of leave-one-out cross-validation, which in its $j$-th iteration uses all training examples except the $j$-th one to learn a function $f_j(x)$, and after that evaluates the prediction error of $f_j(x)$ on the $j$-th example. After iterating over all $n$ training examples, the following estimate of the prediction error of the learning algorithm is calculated

$$R_{\text{loo}}(f) = \frac{1}{n} \sum_{i=1}^n \left( y_i - f_j(x_i) \right)^2.$$  

We use the following proposition regarding $R_{\text{loo}}(f)$ [2,9]: under the assumption that the functions $f_j(x)$ are not very different from the function $f(x)$ learned using all training data (i.e., the learning algorithm is stable), then $R_{\text{loo}}(f)$ is an “almost” unbiased estimate of $R(f)$. This way, among a set of candidate functions, the leave-one-out procedure selects the function which minimizes $R_{\text{loo}}(f)$.

In general, the major drawback of leave-one-out CV is the requirement to call the learning algorithm as many times as the number of training examples. Fortunately, the special structure of the kernel regularized least squares method allows the calculation of $R_{\text{loo}}(f_k)$ practically for free after finding the solution $\alpha_0$ according to Expression (10) using all training data and the kernel matrix $K$.

In order to show this remarkable property, let us consider in the following that the $j$-th example is held-out. Let $f_j^* = \alpha_0 k_j$ be the function obtained by solving Expression (10) using the $n-1$ remaining examples

$$f_j^*(x) = \sum_{i=1}^{n-1} \alpha_i^* k(x_i, x).$$

The following proposition regarding $f_j^*$ is true [2,4].

**Proposition 1.** Let $f_j^*$ be the minimizer of the KRLS problem

$$\min f \in \mathcal{H} \left[ \frac{1}{n} \sum_{i=1}^n \left( y_i - f(x_i) \right)^2 + y_f f_k^2 \right].$$

Then, $f_j^*$ is also the minimizer of the KRLS problem

$$\min f \in \mathcal{H} \left[ \frac{1}{n} \sum_{i=1}^n \left( y_i - f(x_i) \right)^2 + \frac{1}{n} \left( f_j(x_j) - f(x_j) \right)^2 + y_f f_k^2 \right].$$

Denoting by $y_1$ the $n \times 1$ vector $[y_1, \ldots, y_n]^T$ in which

$$y_i = \begin{cases} y_i, & i \neq j, \\ f_j(x_i), & i = j, \end{cases}$$

...
and using Expression (17), the coefficients $\alpha_i$ in (16) would be calculated by solving

$$\alpha_i = (K + \gamma I)^{-1}y_i.$$ 

However, $\gamma$ is not fully specified because it depends on the knowledge of $f_j$ (and $\alpha_i$ thereof). Fortunately, it is possible to avoid this circular dependence. For this, let $f_j(x_i)$ be the function obtained from Expression (10) using all training examples. Denoting $G = (K + \gamma I)$, the following chain of equalities is true:

$$f_j'(x_i) - f_j(x_i) = \sum_{i=1}^{n} (K^{-1})_{ij}(y_i' - y_i)$$

$$= (K^{-1})_{ij}(f_j'(x_i) - y_i).$$

Solving for $f_j'(x_i)$ in Expression (18) and letting $h_{ij} = (K^{-1})_{ij}$, the following expression for $f_j'(x_i)$ is obtained:

$$f_j'(x_i) = \frac{f_j'(x_i) - h_{ij}y_i}{1 - h_{ij}}.$$ 

(19)

Expression (19) tells us how to obtain the leave-one-out prediction for any $x_i$, based on calculations using all training data. Computationally speaking, the most expensive calculation is that of $\gamma$, which can be carried out either directly or through eigen decomposition.

Using Expression (19), we arrive at the following for the leave-one-out estimator of KRLS

$$\mathcal{R}_{\text{loo}}(f_{K_r}) = \frac{1}{n} \sum_{i=1}^{n} (f_j'(x_i) - y_i)^2.$$ 

Note that the above expression should be minimized over different candidate pairs of kernel functions and $\gamma$ values.

### 4.2. Complexity penalization

Another direction of research in statistical model selection is based on the idea of bounding the prediction error in Expression (14). Let us suppose that a learning algorithm selects from a set of functions $\mathcal{F}$ a function $f$ which minimizes the empirical prediction error (squared error) over $\mathcal{F}$

$$\mathcal{R}_{\text{emp}}(f) = \frac{1}{n} \sum_{i=1}^{n} (y_i - f(x_i))^2.$$ 

Moreover, suppose there is a quantity $\mathcal{P}(n, p)$ which measures the degree of complexity (or capacity) of the set of functions $\mathcal{F}$. This function $\mathcal{P}(n, p)$ depends on the size of the training data $n$ and on a parameter of complexity $p$. For instance, in polynomial regression, $p$ would be set to the degree of the polynomial used to fit a set of data.

In what follows, we consider bounds of the form

$$\mathcal{R}(f) \leq \mathcal{R}_{\text{emp}}(f)\mathcal{P}(n, p).$$

This expression captures the idea that the more complex a set of functions $\mathcal{F}$ is, the less we can trust the empirical prediction error $\mathcal{R}_{\text{emp}}(f)$.

In the 1970s, several attempts were made at defining $\mathcal{P}(n, p)$, resulting in model selection procedures that were proven to work asymptotically ($n \to \infty$). In Table 1, the most common definitions are listed. The parameter $p$ present in all of the definitions is meant to quantify the degrees of freedom of the set of functions $\mathcal{F}$ from which $f$ is selected. We will come back to $p$ at the end of this section.

<table>
<thead>
<tr>
<th>Name</th>
<th>$\mathcal{P}(n, p)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Final Prediction Error (FPE) [10]</td>
<td>$1 + \frac{p}{n}$</td>
</tr>
<tr>
<td>Generalized Cross-Validation (GCV) [11]</td>
<td>$1 + \frac{p}{n}$</td>
</tr>
<tr>
<td>Schwartz Criterion (SC) [12]</td>
<td>$\frac{p}{n} \ln n$</td>
</tr>
<tr>
<td>Shibata's Model Selector (SMS) [13]</td>
<td>$1 + \frac{p}{2n}$</td>
</tr>
</tbody>
</table>

Table 1: Classical complexity penalization (as they appear in Ref. [5]).

In the 1990s, a non-asymptotic expression for $\mathcal{P}(n, p)$ was derived based on the results of the VC-Theory [5]. The general form of this expression is

$$\mathcal{P}(n, p) = \left( 1 - c \left( 1 + \frac{\ln \frac{n}{p}}{p} - \ln \eta \right) \right)^{\frac{1}{2}}.$$ 

(20)

where $p$ corresponds to the VC-dimension of the set of functions $\mathcal{F}$ and $c, a, \eta$ are constants that need to be set to reflect the particularities of the learning method being used. In Ref. [14], the assignment $c = a = 1$ and $\eta = 1/\sqrt{n}$ was shown to perform well in a polynomial regression task.

#### 4.2.1. The effective number of parameters in KRLS

One of the tenets of Statistical Learning Theory [5] is that the VC-dimension is the characterization of the complexity (capacity) of a set of functions $\mathcal{F}$. However, in the case of the sets of functions in KRLS, it is not known how to efficiently calculate their VC-dimension. When $\mathcal{F}$ is a set of functions linear in their parameters, which is the case of KRLS, the heuristic method for calculating the effective number of parameters [15] can be used as a proxy for the VC-dimension.

Let $\sigma_1 > \sigma_2 > \ldots > \sigma_n$ be the eigenvalues of the kernel matrix $K$ used for solving the KRLS problem in (10). The value

$$p_r = \sum_{i=1}^{r} \frac{\sigma_i}{\sigma_i + \gamma} < n$$

quantifies the effective degrees of freedom of the set of functions from where the solution $f_{X_r}$ is selected [5]. Using $p = p_r$ in any of the expressions for $\mathcal{P}(n, p)$ shown in this section, we arrive at a complexity penalization estimator in KRLS

$$\mathcal{R}_{\text{cp}}(f_{X_r}) = \mathcal{R}_{\text{emp}}(f_{X_r})\mathcal{P}(n, p_r).$$

#### 5. Heuristic model selection

In this section we present two metric-based heuristic procedures that have been found to perform well in certain model selection tasks. Both procedures, called TRI and ADJ, use geometric ideas and unlabeled data to perform model selection [16, 17]. In particular, the ADJ heuristic was found to be a state-of-the-art model selection procedure for selecting the degree parameter in polynomial regression [18, 17].

Since these methods were developed for estimating a continuous generalization parameter, here they apply only to the

---

2 Unlike its name suggests, the GCV criterion is not a cross-validation procedure, but actually an approximation to the leave-one-out estimator presented in Expression (15).

3 $\Theta(p) = \max(0, a)$. 

selection of the parameter \( \gamma \) of KRLS. That is, in general, these methods cannot be used for choosing a kernel function.

5.1. TRI

Suppose the real regression function \( f_{\text{reg}}(x) \) is known and belongs to a metric space. Then, given any two functions \( f_1(x) \) and \( f_2(x) \) in the same metric space and the distance (metric) \( \rho(\cdot, \cdot) \), the triangle inequality applies:

\[ \rho(f_1, f_{\text{reg}}) + \rho(f_2, f_{\text{reg}}) \geq \rho(f_1, f_2). \]

Given a training sequence \((x_1, y_1), \ldots, (x_n, y_n)\) and a large set of unlabeled data \( x_1', \ldots, x_m' \) which comes from the same distribution as the training sequence, we can approximately verify the validity of the triangle inequality using

\[ \rho_{\text{un}}(f_1, f_2) = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (f_1(x_i') - f_2(x_i'))^2}. \]

Now, let us use KRLS and a fixed kernel \( K \) to obtain two functions \( f_1(x) \) and \( f_2(x) \) such that \( \gamma_1 > \gamma_2 \). Consequently, while \( f_1(x) \) will potentially have a smaller empirical error than \( f_2(x) \), there is a risk that its empirical error is a rough estimate of \( \rho(f_1, f_{\text{reg}}) \), and it may happen that

\[ \rho_{\text{tr}}(f_1, y) + \rho_{\text{tr}}(f_2, y) < \rho_{\text{un}}(f_1, f_2). \]

In this situation, the TRI heuristic blames the function \( f_1 \) for the violation of the triangle inequality. This gives rise to the TRI procedure for parameter selection: given a sequence of functions \( f_1, f_2, \ldots \) such that \( \gamma_1 > \gamma_2 > \ldots \), choose the last function in the sequence that does not violate the triangle inequality along with any preceding function.

5.2. ADJ

This other heuristic is based on the assumption that as the training and unlabeled sets come from the same distribution, the following relation should be observed:

\[ \frac{\rho_{\text{un}}(f_1, f_2)}{\rho_{\text{un}}(f_1, f_{\text{reg}})} = \frac{\rho_{\text{tr}}(f_1, f_2)}{\rho_{\text{tr}}(f_1, f_{\text{reg}})}. \]  

(21)

If the functions \( f_1 \) and \( f_2 \) are given along with the training and unlabeled sets, the values \( \rho_{\text{tr}}(f_1, f_2) \) and \( \rho_{\text{un}}(f_1, f_2) \) can be readily calculated. The value \( \rho_{\text{tr}}(f_1, f_{\text{reg}}) \) can be estimated using \( \rho_{\text{tr}}(f_1, y) \). The only remaining unknown in Expression (21) is \( \rho_{\text{un}}(f_1, f_{\text{reg}}) \). Thus, it can be estimated using

\[ \rho_{\text{un}}(f_1, f_{\text{reg}}) = \rho_{\text{tr}}(f_1, y) \frac{\rho_{\text{tr}}(f_1, f_2)}{\rho_{\text{tr}}(f_1, f_{\text{reg}})}. \]  

(22)

This means that, if the assumption in Expression (21) is valid, the prediction error of function \( f_1 \) on unseen data, which is a good approximation to the true prediction error, can be estimated from its empirical error \( \rho_{\text{tr}}(f_1, y) = \sqrt{\mathcal{R}_{\text{emp}}(f_1)} \) and the ratio \( \rho_{\text{un}}(f_1, f_{\text{reg}})/\rho_{\text{tr}}(f_1, f_{\text{reg}}) \).

This gives rise to the ADJ procedure, which for parameter selection in KRLS reads: given a fixed kernel function \( K \) and a sequence of KRLS functions \( f_1 = f_{k_1}, f_2 = f_{k_2}, \ldots \) such that \( \gamma_1 > \gamma_2 > \ldots \), first multiply \( \rho_{\text{tr}}(f_1, y) \) by

\[ \mu(f_1) = \max_{|j' < i|} \frac{\rho_{\text{un}}(f_{j'}, f_{\text{reg}})}{\rho_{\text{tr}}(f_{j'}, f_{\text{reg}})} \]

and then choose the function \( f_{\gamma_i} \) that has the smallest value of

\[ \mathcal{R}_{\text{adj}}(f_{\gamma_i}) = \sqrt{\mathcal{R}_{\text{emp}}(f_{\gamma_i})/\mu(f_{\gamma_i})}. \]  

(23)

Note from Expression (23) that the ADJ method resembles a complexity penalization procedure.

6. Experiments

The experiments reported in this section aim at answering the following questions regarding model selection in kernel regularized least squares:

Q1: The proposed additive spline kernel—Section 3—has potential advantages over both the multiplicative spline and RBF kernels. Do these advantages show up experimentally?

Q2: The complexity penalization methods described in Section 4.2 were evaluated mainly for polynomial regression in artificial datasets, where they were found to be comparable to cross-validation [14]. Does this result hold for KRLS in real datasets?

Q3: Can the constants in Expression (20) (VC bound) be improved for model selection in KRLS?

Q4: The heuristic approach ADJ was demonstrated to give state-of-the-art results in polynomial regression, even outperforming cross-validation, while TRI has been less successful than ADJ [17,18]. Do these results hold for KRLS in real datasets?

6.1. Experimental setup

We use 10 regression datasets from two repositories: UCI\(^4\) and Keel\(^5\) [19]. A detailed description of the datasets can be found in their repository websites. A brief description is shown in Table 2.

In what follows we describe a single trial of our experiments for one dataset. Two-thirds of the training examples were randomly held-out to form a test set \((x_1, y_1), \ldots, (x_n, y_n)\). From the remaining one-third, \( n \) examples are randomly selected to compose the training set \((x_1, y_1), \ldots, (x_n, y_n)\) with only their labels as the unlabeled set \( x_1', \ldots, x_m' \) for both TRI and ADJ methods.

The training set thus created is used along with a fixed kernel \( K \) and value of \( \gamma \) to obtain a function \( f_{K,\gamma_{\text{opt}}} \) in accordance with Expression (10). The root mean squared error (RMSE) of \( f_{K,\gamma_{\text{opt}}} \) in the test set

\[ \text{rmse}(f_{\gamma_{\text{opt}}}) = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (f_{\gamma_{\text{opt}}}(x_i) - y_i)^2}. \]

is then recorded for evaluation purposes.

In order to try reasonable values of \( \gamma \), the first (larger) eigenvalue \( \sigma_1 \) of the kernel matrix \( K \) is taken as a reference. Using the 50 values \( v_1, \ldots, v_{50} \) equally spaced according to the logarithmic scale in the range \([10^{-4}, 10^1]\), we try the following 50 values:

\[ \gamma_1 = v_1 \sigma_1, \ldots, \gamma_{50} = v_{50} \sigma_1. \]

Thus, after obtaining \( f_{\gamma_1}, \ldots, f_{\gamma_{50}} \), and calculating their respective RMSE on the test set, we find the function \( f_{\gamma_{\text{opt}}} \), with the minimum value of RMSE, which is the gold standard to our model selection experiments described above.

For the RBF kernel, we used only leave-one-out cross-validation for selecting \( \gamma \) and \( \sigma \). For selecting a good range of candidates of \( \sigma \), we calculated the empirical distribution of the distance between training points and then considered 11 quantiles of this distribution as

\(^4\) http://archive.ics.uci.edu/ml/
\(^5\) http://sci2s.ugr.es/keel/datasets.php
candidates of $\sigma$: $(0.010, 0.025, 0.125, 0.250, 0.375, 0.500, 0.625, 0.750, 0.875, 0.975, 0.990)$.

For both the additive and the multiplicative spline kernels, we implemented and ran the following model selection procedures. Leave-one-out cross-validation (LOOCV); complexity penalization: Final Prediction Error (FPE), Generalized Cross-Validation (GCV), Schwarz Criterion (SC), Shibata’s Model Selector (SMS), VC bound (VC1); and metric-based: TRI and ADJ. Each model selection procedure had access only to the training set, with the exception of TRI and ADJ, which had access to the unlabeled set as well. According to each model selection procedure, one function $f_{chosen}$ is picked among $f_{r_1}, \ldots, f_{r_N}$. We evaluate the quality of each procedure by the ratio:

$$r = \frac{\text{rmse}(f_{chosen})}{\text{rmse}(f_{\gamma})}$$

(24)

That is, the closer $r$ is to 1, the better the model selection performance is.

So far, this description refers to a single trial. For a fixed size of the training set $n$, we conducted 20 such trials and report the mean and the standard deviation of each model selection procedure. This experiment was carried out for $n=20$ and $n=100$ on each dataset.

In order to address question Q2, the constants $c$ and $a$ in Expression (20) were experimentally optimized using only one of the datasets (randomly selected). The resulting constants $c=0.74$ and $a=1.35$ were used as another model selection procedure with the VC expression (VC2).

### 6.2. Results and discussion

We start the analysis of the results by comparing the three kernel functions considered in this paper: RBF, multiplicative spline, and additive spline. Table 3 reports (1) the best value of RMSE achieved on the test set by one of the $f_{r_1}, \ldots, f_{r_N}$ of a fixed kernel; (2) the value of RMSE achieved on the test set by the best LOOCV function among the $f_{r_1}, \ldots, f_{r_N}$ of a fixed kernel. For each dataset and $n$, we highlight in bold the best value of RMSE across the three kernels in the LOOCV set of columns, and do the same for the BEST set of columns.

Table 2: Datasets used in the experiments.

<table>
<thead>
<tr>
<th>#</th>
<th>Dataset</th>
<th>Instances</th>
<th>Features</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Abalone</td>
<td>4177</td>
<td>8</td>
</tr>
<tr>
<td>2</td>
<td>Concrete-str</td>
<td>1030</td>
<td>8</td>
</tr>
<tr>
<td>3</td>
<td>Friedman</td>
<td>1200</td>
<td>5</td>
</tr>
<tr>
<td>4</td>
<td>Mortgage</td>
<td>1049</td>
<td>15</td>
</tr>
<tr>
<td>5</td>
<td>Stock</td>
<td>950</td>
<td>9</td>
</tr>
<tr>
<td>6</td>
<td>Treasury</td>
<td>1049</td>
<td>15</td>
</tr>
<tr>
<td>7</td>
<td>Wankara</td>
<td>321</td>
<td>9</td>
</tr>
<tr>
<td>8</td>
<td>Wine-red</td>
<td>1599</td>
<td>11</td>
</tr>
<tr>
<td>9</td>
<td>Wine-white</td>
<td>4898</td>
<td>11</td>
</tr>
</tbody>
</table>
most of the cases considered. Moreover, VC1 shows several catastrophic performances. Thus, we can conclude that there is room for improving the constants of the VC expression for $P(n, p)$ given in Ref. [14].

Q4: Comparing the results obtained by the two heuristic metric-based approaches, TRI and ADJ, there is no strong pattern of one heuristic outperforming the other. Thus, these results do not show, as claimed in polynomial regression [18,17], that ADJ outperforms TRI. For the additive spline in particular, the reverse seems to be the case. Comparing TRI and ADJ with LOO, worse results are obtained by TRI and ADJ in most of the cases. These results show that both heuristic metric-based approaches do not outperform the leave-one-out cross-validation procedure.

7. Conclusion

In this paper we investigated the model selection problem for small sample regression in the kernel regularized least squares (KRLS) method, which narrows down to the selection of a kernel function and a regularization parameter.

As for the selection of the kernel function, we proposed an additive multidimensional spline kernel that outperformed the
traditional RBF kernel, as well as the previously proposed multiplicative spline kernel. This perceived advantage stems from the fact that additive spline kernels span a set of functions which are less diverse than multiplicative ones, making additive spline kernels well-suited to small sample regression.

As for the regularization parameter $\gamma$, alternative statistical and heuristic model selection procedures were investigated for its selection. These procedures, six of them based on complexity penalization (FPE, SC, GCV, SMS, VC1 and VC2) and two of them based on the geometry of metric spaces (TRI and ADJ), were experimentally evaluated in real datasets and compared to the traditionally used RBF kernel, as well as the previously proposed multiplicative spline kernel. This perceived advantage stems from the fact that additive spline kernels span a set of functions which are less diverse than multiplicative ones, making additive spline kernels well-suited to small sample regression.

Our results corroborated that cross-validation may perform poorly when data is scarce, mainly when used in conjunction with RBF kernels. Unfortunately, the best performing alternative method—FPE—does not provide a large improvement on the results when cross-validation fails. In future work, we plan to investigate ways of combining FPE and LOO that may lead to an improved parameter selection method.

Among the investigated procedures, the only one that lends itself to further optimization is the VC expression. In future work, we intend to investigate whether there is an assignment of constants that may render the VC expression practical.

Acknowledgments

This work is supported by Grant #2009/17773-7, São Paulo Research Foundation (FAPESP).

References

Igor Braga holds a PhD in Computer Science from the University of São Paulo, Brazil. His research interests lie in the area of machine learning and Statistical Learning Theory.

Maria Carolina Monard holds an MSc degree from Southampton University, UK, and a PhD degree from Pontifícia Universidade Católica do Rio de Janeiro, Brazil, in 1968 and 1980 respectively. Currently, she is an Emeritus Professor in Computer Science University of São Paulo, Campus São Carlos, Institute of Mathematics and Computer Science. Her research interests are in the field of Artificial Intelligence and include machine learning, data and text mining.