Expert Systems with Applications 39 (2012) 9246–9254

Contents lists available at SciVerse ScienceDirect

Expert Systems with Applications

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

Evolutionary kernel density regression

Oliver Kramer [*], Fabian Gieseke

Department for Computer Science, University of Oldenburg, 26111 Oldenburg, Germany

ARTICLE INFO

Keywords:
Kernel regression
Bandwidth optimization
Manifold learning
Unsupervised kernel regression
Evolution strategies

ABSTRACT

The Nadaraya–Watson estimator, also known as kernel regression, is a density-based regression technique. It weights output values with the relative densities in input space. The density is measured with kernel functions that depend on bandwidth parameters. In this work we present an evolutionary bandwidth optimizer for kernel regression. The approach is based on a robust loss function, leave-one-out cross-validation, and the CMSA-ES as optimization engine. A variant with local parameterized Nadaraya–Watson models enhances the approach, and allows the adaptation of the model to local data space characteristics. The unsupervised counterpart of kernel regression is an approach to learn principal manifolds. The learning problem of unsupervised kernel regression (UKR) is based on optimizing the latent variables, which is a multimodal problem with many local optima. We propose an evolutionary framework for optimization of UKR based on scaling of initial local linear embedding solutions, and minimization of the cross-validation error. Both methods are analyzed experimentally.

1. Introduction

Kernel-based machine learning methods have shown great success in the last decades. Many successful methods like support vector machines (SVM) (Schölkopf & Smola, 2001; Suykens & Vandewalle, 1999; Vapnik, 1995) are based on quadratic programming as the corresponding problem can be formulated as convex optimization problem. But there are also cases when the employment of stochastic optimization is reasonable in machine learning, e.g., in case of non-convex optimization problems induced by non-positive definite kernel functions, noisy optimization problems, e.g., from real-world data observations, non-differentiable loss functions (e.g. the L_1 loss) or large data sets that may afford parallelization. Stochastic search like evolutionary algorithms may help to overcome these problems. Evolutionary computation has grown to a rich field of powerful methods for global optimization. They are embarrassingly parallelizable, and thus fairly efficient search methodologies in distributed computing scenarios.

There are already many examples in literature that show that evolutionary methods are successful in kernel-based machine learning. Stoean, Dumitrescu, Preuss, and Stoean (2006), Stoean, Preuss, Stoean, and Dumitrescu (2007) and Stoean, Stoean, E-Darzi, and Dumitrescu (2009a, 2009b) directly solve the primal optimization problem of SVMs to find the optimal discriminant function for regression and classification tasks by means of evolution strategies. Mierswa and Morik (2008) investigated simple data sets, in which feature spaces induced by usual kernel functions fail. They propose a generic kernel learning scheme that is based on non-convex optimization. Furthermore, Mierswa (2006) explicitly optimizes the inherent tradeoff between training error and model complexity of SVMs by means of multi-objective evolutionary algorithms, i.e., NSGA-II (Deb, Pratap, Agarwal, & Meyarivan, 2002). An example for the application of evolutionary methods to combinatorial problems in kernel-based machine learning stems from Gieseke, Pahikkala, and Kramer (2009). They solve the combinatorial problem of assigning elements to proper clusters with a (1 + 1)-EA. The approach aims at finding an optimal partitioning of data into two classes, i.e., at solving a mixed integer problem. With the help of a kernel matrix approximation shortcut, computational costs can be reduced during the approximation, and the evaluation of a huge set of solutions is possible within reasonable time.

The purpose of this paper is to show that evolutionary continuous methods, in particular CMSA-ES and a Powell ES, are good in exploring the search space of kernel bandwidths of kernel regression (KR), and the search space of latent variables of the unsupervised counterpart UKR. In Section 2 we introduce the evolutionary KR variant based on the Nadaraya–Watson estimator (Nadaraya, 1964; Watson, 1964) with a parameterized kernel function, Huber's loss function (Huber, 1981), and robust leave-one-out cross-validation (LOO-CV). The covariance matrix adaptation evolution strategy CMSA-ES is used for adaptation of the kernel parameters. In Section 3 we introduce an evolutionary engine for the unsupervised counterpart of kernel regression for learning principal manifolds. Here, the CMSA-ES is used for optimization of the scaling

[*] Corresponding author.
E-mail address: oliver.kramer@uni-oldenburg.de (O. Kramer).

0957-4174/$ - see front matter © 2012 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2012.02.080

parameters of initial local linear embedding solutions, and minimization of the cross-validation error. In Section 4 we summarize the results and give an outlook to future research perspectives.

2. Evolutionary kernel regression

In this section we will introduce a variant of kernel regression conducting evolutionary search in the space of kernel bandwidths.

2.1. Regression

Regression is a field of statistical learning that comprises methods to predict output values $\mathbf{y} \in \mathbb{R}^d$ to given input values $\mathbf{x} \in R^q$ based on sets of input–output examples. The goal is to learn a function $\mathbf{f}:\mathbf{x} \to \mathbf{y}$ known as regression function. We assume that a data set consisting of observed pairs $(\mathbf{x}_i, \mathbf{y}_i) \in \mathbf{X} \times \mathbf{Y}$ is given. We assume that the ground truth $\mathbf{f}$ is unknown, and the task of our regression model is to estimate $\mathbf{f}$ by learning a "good" model $\hat{\mathbf{f}}$. In particular, the regression model should fulfill two conditions: First, it should reconstruct the observed data, and second, it should generalize and predict unknown mappings.

Simple linear regression has been successfully applied for more than 150 years (Weisberg, 1985). It is based on the assumption that the relationship between $\mathbf{y}$ and $\mathbf{x}$ is approximately linear. A kernel-based regression function estimator is support vector regression. The goal of ϵ-SVR introduced by Vapnik (1995) is to learn a function $\hat{\mathbf{f}}$ with only an ϵ-deviation of the target outputs $\mathbf{y}_i$. The idea is to fit a linear function of the form $\mathbf{f}(\mathbf{x}) = \langle \mathbf{w}, \mathbf{x} \rangle + b$ with $b \in \mathbb{R}$ and dot product $\langle \cdot, \cdot \rangle$. Smola and Schölkopf (1998) present a comprehensive tutorial to SVR and nonlinear kernel-based methods. Many other methods have been proposed in the past, e.g., locally weighted projection regression (LWPR) (Vijayakumar & Schaal, 2000), or an evolutionary approach to least trimmed squares by Morell, Bernholt, Fried, Kunert, and Nunkesser (2008). A complete depiction of regression methods goes beyond the scope of this work that has a focus on evolutionary tuning of supervised and unsupervised kernel (density) regression.

2.2. Kernel density estimation

The first important ingredient of kernel regression is kernel density estimation, which stems from statistics. Kernel density estimation is a method for estimation of distributions (Parzen, 1962). It can be seen as a smooth version of histograms that count the number of samples in consecutive intervals. The kernel density approximation of the probability density function is:

$$p(\mathbf{x}) = \frac{1}{Nh} \sum_{i}^{N} K\left(\frac{\mathbf{x} - \mathbf{x}_i}{h}\right), \tag{1}$$

with a kernel function $K : \mathbb{R}^d \to \mathbb{R}$ that measures the density at point $\mathbf{x}$ (see Section 2.4), and the so-called bandwidth parameter h. Eq. (1) is also known as *Parzen window estimator*. The bandwidth h has a similar meaning like the width of histogram bins as they define the width of the influence of kernel K. Intuitively, the kernel density estimator places small bumps[1] at each observation and the whole function $p(\mathbf{x})$ becomes the sum of bumps. Recently, Ozakin and Gray (2010) have shown that due to correlations between variables, kernel density estimation is more effective than it has previously been believed. This is because real data is often lying on a low-dimensional "sub-manifold".

2.3. Nadaraya–Watson estimator

Kernel regression makes use of kernel density estimates, and weights the output values with relative kernel densities. The idea has been introduced by Nadaraya (1964), Watson (1964), and is known as *Nadaraya–Watson estimator*. The Nadaraya–Watson estimator combines the expression of the regression problem as joint marginal distribution:

$$\mathbf{f}^*(\mathbf{x}) = \int \mathbf{y} p(\mathbf{y}|\mathbf{x})\mathbf{y} = \int \mathbf{y} \frac{p(\mathbf{x}, \mathbf{y})}{p(\mathbf{x})} \, \mathrm{d}\mathbf{y}, \tag{2}$$

with the Parzen window estimator (see Eq. (1)). In the multivariate expression the Nadaraya–Watson estimator weights the output values of the training samples with their relative kernel densities:

$$\mathbf{f}(\mathbf{x}; \mathbf{H}) = \sum_{i=1}^{N} \mathbf{y}_i \frac{K_{\mathbf{H}}(\mathbf{x} - \mathbf{x}_i)}{\sum_{j=1}^{N} K_{\mathbf{H}}(\mathbf{x} - \mathbf{x}_j)}. \tag{3}$$

Bandwidth matrix $\mathbf{H}$ contains the bandwidths. It is a diagonal matrix in the multivariate variant of a Gaussian kernel, see Section 2.4. The bandwidth is an important parameter that controls the smoothness of the predicted function. We will discuss the influence of the bandwidths in the following section.

Let N be the set of the data sample archive. For the prediction of the function value of one data sample, N kernel densities have to be computed, i.e., the prediction of $\widehat{N}$ data samples can be computed in $\mathcal{O}(N \cdot \widehat{N})$.

2.4. Kernel functions

Kernel regression is based on a density estimate of data samples with a kernel function $K : \mathbb{R}^d \to \mathbb{R}$. A typical kernel function is the Gaussian (multivariate) kernel:

$$K_G(\mathbf{z}) = \frac{1}{(2\pi)^{q/2} \det(\mathbf{H})} \exp\left(-\frac{1}{2} |\mathbf{H}^{-1}\mathbf{z}|^2\right), \tag{4}$$

with bandwidth matrix $\mathbf{H} = \mathrm{diag}(h_1, h_2, \ldots, h_q)$. Another frequent choice – due to its convenient characteristics – is the Epanechnikov kernel:

$$K_E(\mathbf{z}) = D\left(\frac{|\mathbf{z}|}{h}\right), \tag{5}$$

with

$$D_E(t) = \frac{3}{4}[1 - t^2]_+ = \begin{cases} \frac{3}{4} \cdot (1 - t^2) & |t| < 1, \\ 0 & |t| \geqslant 1. \end{cases} \tag{6}$$

The bandwidth h defines the radius of the supported region, similar to the standard deviation of the Gaussian function. The Epanechnikov kernel exhibits useful asymptotic characteristics. For $h \to 0$ it reconstructs the data points, for $h \to \infty$ it averages the over all N data points (Härdle & Simar, 2007).

An essential part of kernel regression is the bandwidth h. For small bandwidths the estimate generates small bumps at the locations of the sample points. For large bandwidths an overgeneralization hides the real structure of the sample distribution, and leads to oversmoothing. Fig. 1 shows three different bandwidth for the Epanechnikov kernel. The left plot shows a situation of an overfitted, too small bandwidth, that only reconstructs the data points and their close neighborhood. The right plot shows the opposite: an undersmoothed model with an oversized bandwidth. The bandwidth in the middle allows to estimate the structure of the underlying distribution. An experimental illustration of the influence of bandwidth choices on the regression model will be shown in the experimental part in Section 2.8.

[1] In case of a Gaussian kernel the bumps have a Gaussian shape.