Liet. mat. rink. LMD darbai, **48/49**, 2008, 191–195

Kernel regression on matrix patterns

Povilas DANIUŠIS, Pranas VAITKUS (VU)
e-mail: povilas.daniusis@mif.vu.lt, vaitkuspranas@gmail.com

Abstract. In this paper we propose a kernel-based regression model for matrix patterns (KRMP). The training algorithm is derived. The proposed model was empirically compared with traditional models.

Keywords: matrix inputs, kernel methods, regression, classification.

1. Introduction

In most supervised or unsupervised machine learning models the inputs are described by vectors. However, there are some important applications where the inputs are sets of vectors, or matrices (for example, images, graphs, multidimensional time series, and others). The standard approach is to decompose the input matrix into the vector, but such decomposition can delete important information about an inner structure of the input matrix. Cai *et al.* [1] experimentally demonstrated that even in vector cases it can be useful to reshape the input vector into a matrix. In recent years an interest in this problem has arisen ([1,2,5,6,3]). Most publications on this topic analyze the linear methods (for example, see [1,5,6,3]). In this article we introduce a new nonlinear kernel regression model – KRMP (kernel regression on matrix patterns). In the kernel methods the initial data vectors $\mathbf{x}_i$ are mapped to high dimensional features $\boldsymbol{\phi}(\boldsymbol{x_i})$. By Mercer's theorem, which states that any continuous, symmetric, nonnegative definite function $k(\cdot,\cdot)$ can be expressed as an inner product (i.e., $k(\mathbf{x}_i,\mathbf{x}_j)=\boldsymbol{\phi}(\boldsymbol{x_i})^T\boldsymbol{\phi}(\boldsymbol{x_j})$) [4], computation of the inner products in the feature space is replaced by computation of the values of the kernel function $k(\cdot,\cdot)$. This idea is known as kernel trick.

2. Kernel least squares regression

In this section we briefly describe the traditional kernel least squares regression model (KR). Let $\mathbf{y}\in\mathbb{R}^N$ be a vector and $\mathbf{X}=[\mathbf{x}_1,\mathbf{x}_2,\dots,\mathbf{x}_N]^T$ (where $\mathbf{x}_i\in\mathbb{R}^m$) be an observation matrix. In the linear regression we seek a vector $\boldsymbol{\alpha}$ which minimizes the norm $\parallel \mathbf{y}-\mathbf{X}\boldsymbol{\alpha}\parallel^2$. The solution to this problem is defined by $\boldsymbol{\alpha}=(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$.The linear regression can be extended to nonlinear by mapping the original data into a feature space. Let $k(\mathbf{x}_i,\mathbf{x}_j)=\boldsymbol{\phi}(\boldsymbol{x_i})^T\boldsymbol{\phi}(\boldsymbol{x_j})$ be a Mercer kernel. In the kernel regression the original data are mapped into a feature space (i.e., each observation $\mathbf{x}_i$ is mapped to $\boldsymbol{\phi_i}=\boldsymbol{\phi}(\mathbf{x_i})$). Denote $\widetilde{\mathbf{X}}=[\boldsymbol{\phi_1},\boldsymbol{\phi_2},\dots,\boldsymbol{\phi_N}]$. We seek the solution $\mathbf{a}$ which minimizes

$$J=\parallel \mathbf{y}-\widetilde{\mathbf{X}}^T\mathbf{a}\parallel^2 \tag{1}$$

and is defined in the basis of the columns of $\widetilde{\mathbf{X}}$ (i.e., $\mathbf{a} = \widetilde{\mathbf{X}}\boldsymbol{\alpha}$). The least squares solution of (1) is defined by $\boldsymbol{\alpha} = \mathbf{K}^{-1}\mathbf{y}$, where $\mathbf{K} = \widetilde{\mathbf{X}}^T \cdot \widetilde{\mathbf{X}}$ is a kernel matrix. To avoid an overfitting, the regularization often is used. In that case, the norm of the solution $\boldsymbol{\alpha}$ is penalized and $J' = \| \mathbf{y} - \widetilde{\mathbf{X}}^T\mathbf{a} \|^2 + \lambda \| \mathbf{a} \|^2 = \| \mathbf{y} - \mathbf{K}\boldsymbol{\alpha} \|^2 + \lambda\boldsymbol{\alpha}^T\mathbf{K}\boldsymbol{\alpha}$ is minimized. The solution to this problem is defined by $\boldsymbol{\alpha} = (\mathbf{K}+\lambda\mathbf{I})^{-1}\mathbf{y}$, where $\lambda \geqslant 0$ is a regularization constant.

3. KRMP model

Denote the training set by $T = (\mathbf{X}_i, y_i)_{i=1}^N$, where $\mathbf{X}_i$ – $m \times n$ matrices (inputs) and y_i – scalars (outputs) and $\mathbf{y} = [y_1, y_2, \ldots, y_N]^T$. In an article [1] Cai *et al.* proposed a linear model

$$\widehat{y}(\mathbf{X}) = \mathbf{u}^T\mathbf{X}\mathbf{v}, \tag{2}$$

where $\mathbf{u} \in \mathbb{R}^m$ and $\mathbf{v} \in \mathbb{R}^n$. When the inputs are matrices (especially when they have large dimensions) Cai's model has an advantage over standard linear regression because it has fewer parameters and exploits an inner structure of the input matrix. In the following a nonlinear version of Cai's model is proposed.

Let $\mathbf{X}_i = [\mathbf{x}_1^i, \mathbf{x}_2^i, \ldots, \mathbf{x}_n^i]$, $\mathbf{x}_j^i \in \mathbb{R}^m$, $i = 1, \ldots, N$ and $j = 1, \ldots, n$. Fix a Mercer kernel $k(\mathbf{x}_i, \mathbf{x}_j) = \boldsymbol{\phi}(\boldsymbol{x}_i)^T\boldsymbol{\phi}(\boldsymbol{x}_j)$ and define $\widetilde{\mathbf{X}}_i = [\boldsymbol{\phi}_1^i, \boldsymbol{\phi}_2^i, \ldots, \boldsymbol{\phi}_n^i]$, where $\boldsymbol{\phi}_j^i = \boldsymbol{\phi}(\mathbf{x}_j^i) \in \mathbb{R}^{m'}$. We will analyze the following model:

$$\widehat{y}(\mathbf{X}) = \mathbf{u}^T\big(\widetilde{\mathbf{X}}^T\mathbf{A}\big)\mathbf{v} = \mathbf{u}^T\Big(\sum_{i=1}^N \alpha_i\widetilde{\mathbf{X}}^T\widetilde{\mathbf{X}}_i\Big)\mathbf{v} = \mathbf{u}^T\Big(\sum_{i=1}^N \alpha_i\mathbf{K}\big(\mathbf{X}, \mathbf{X}_i\big)\Big)\mathbf{v}, \tag{3}$$

where $\mathbf{u}, \mathbf{v} \in \mathbb{R}^n$, $\mathbf{A} = \sum_{i=1}^N \alpha_i\widetilde{\mathbf{X}}_i$, and kernel matrix $\mathbf{K}(\mathbf{X}, \mathbf{X}_i) = \widetilde{\mathbf{X}}^T \cdot \widetilde{\mathbf{X}}_i$. By kernel trick one can calculate $\widehat{y}(\mathbf{X})$ knowing only a kernel $k(\cdot\,,\cdot)$ and without knowing actual mapping $\phi(.)$. The (2) or (3) models can be applied on regression or classification problems.

4. Parameter estimation

In this section an algorithm for regularized sum squared error (RSSE) minimization is formulated. RSSE is defined by

$$RSSE(..) = \frac{1}{2}\sum_{(\mathbf{X},y)\in T}\Big(y - \mathbf{u}^T(\widetilde{\mathbf{X}}^T\mathbf{A})\mathbf{v}\Big)^2 + \frac{1}{2}\Big(\lambda_1\boldsymbol{\alpha}^T\boldsymbol{\alpha} + \lambda_2\mathbf{u}^T\mathbf{u} + \lambda_3\mathbf{v}^T\mathbf{v}\Big), \tag{4}$$

where $\lambda_1, \lambda_2, \lambda_3 \geqslant 0$ are regularization constants. Our aim is to estimate the parameters $\mathbf{u}$, $\mathbf{v}$ and $\boldsymbol{\alpha}$, which minimizes (4).

For the sake of convenience, define $N \times N$ matrix $\mathbf{M} = (m_{i,j})$, $m_{ij} = \mathbf{u}^T\mathbf{K}(\mathbf{X}_i, \mathbf{X}_j)\mathbf{v}$, and $n \times n$ matrix $\mathbf{Y}_i = \widetilde{\mathbf{X}}_i^T \cdot \mathbf{A}$.

Differentiate (4) with respect to $\mathbf{u}$ and $\mathbf{v}$ and set the derivatives to $\mathbf{0}$:

$$\nabla_{\mathbf{u}} RSSE(..) = \Big(\sum_{i=1}^{N} \mathbf{Y}_i \mathbf{v}\mathbf{v}^T \mathbf{Y}_i^T\Big)\mathbf{u} - \Big(\sum_{j=1}^{N} y_j \mathbf{Y}_j\Big)\mathbf{v} + \lambda_2 \mathbf{u} = \mathbf{0}, \tag{5}$$

$$\nabla_{\mathbf{v}} RSSE(..) = \Big(\sum_{i=1}^{N} \mathbf{Y}_i^T \mathbf{u}\mathbf{u}^T \mathbf{Y}_i\Big)\mathbf{v} - \Big(\sum_{j=1}^{N} y_j \mathbf{Y}_j^T\Big)\mathbf{u} + \lambda_3 \mathbf{v} = \mathbf{0}. \tag{6}$$

For fixed $\mathbf{u}$ and $\mathbf{v}$, optimal $\boldsymbol{\alpha}$ can be found by the well-known least squares formula. From equations (5), (6) we see that the optimal parameters depend on each other, thus cannot be computed explicitly. For the parameter optimization the following algorithm can be applied:

Algorithm 1 KRMP

```
1. Fix arbitrary $\mathbf{u}, \mathbf{v} \in \mathbb{R}^n$, $\boldsymbol{\alpha} \in \mathbb{R}^N$, regularization parameters $\lambda_1, \lambda_2, \lambda_3 \geqslant 0$,
   $t_0 \in \mathbb{N}$, $\epsilon > 0$ and set $t = 1$.
2. Calculate $\boldsymbol{\alpha} = (\mathbf{M} + \lambda_1 \mathbf{I})^{-1}\mathbf{y}$.
3. Calculate $\mathbf{v} = (\sum_{i=1}^{N} \mathbf{Y}_i^T \mathbf{u}\mathbf{u}^T \mathbf{Y}_i + \lambda_3 \mathbf{I})^{-1} (\sum_{j=1}^{N} y_j \mathbf{Y}_j^T)\mathbf{u}$.
4. Calculate $\mathbf{u} = (\sum_{i=1}^{N} \mathbf{Y}_i \mathbf{v}\mathbf{v}^T \mathbf{Y}_i^T + \lambda_2 \mathbf{I})^{-1} (\sum_{j=1}^{N} y_j \mathbf{Y}_j)\mathbf{v}$.
5. Set $t := t + 1$.
6. Repeat Step 2 until $RSSE < \epsilon$ or $t > t_0$.
```

Since with respect to $\boldsymbol{\alpha}$, $\mathbf{u}$, and $\mathbf{v}$ (4) is a convex function, an iterative sequence of its values, defined by the KRMP algorithm, converges because it monotonically non-increases and is bounded by 0.

5. Numerical simulations

In this section the (3) model is empirically compared with two supervised machine learning algorithms: the kernel regression, which is the analogue of (3) when the inputs are vectors, and support vector machines (SVM, [4]). The results of [1] suggest, that matrix-based models are efficient with small training samples. However, Cai *et al.* worked with linear models. In our experiments we will also use a small part of the data for the training of the models and check this assumption for nonlinear ones.

The benchmark data sets are three binary classification data sets from UCI machine learning repository[1]. In the experiments we used a Gaussian kernel $k(\mathbf{x}_i, \mathbf{x}_j) = \exp\big(-\frac{||\mathbf{x}_i - \mathbf{x}_j||^2}{2\sigma^2}\big)$. The measure of performance of the models was the correct classification probability over the testing set. In each experiment the training set was

[1] http://archive.ics.uci.edu/ml/

selected randomly, all experiments were performed 100 times, and the results were averaged. The meta-parameters (a bandwidth σ, regularization constants, etc.) were selected using cross validation.

5.1. Data sets

The *Ionosphere* data set consists of 351 observations, which have 34 features. The variance of the 2nd feature is zero, so this feature is removed. For training of the models 20 random examples are selected; the others are left for testing. For the KRMP model the input vectors are preprocessed into 3×11 matrices.

SPECTF data set consists of 80 observations, which have 44 features. For the KRMP model the input vectors are preprocessed into 4×11 matrices. In this case 10 observations are randomly selected for the training of the models.

In *Australian credit approval* data set the input data consists of 690 14-dimensional input vectors. When KRMP is used, the initial data vectors are preprocessed into the 2×7 matrices. For training of the models 10 training examples are selected, others are left for testing.

5.2. Empirical results

Sign ">" means that p-value in the signed rank test for zero median between differences of the performances of the models was < 0.01, "$\sim$" means the opposite case. From the Table 1 we see that the KRMP model was more efficient than the traditional kernel regression (KR) model and performed similarly or better than SVM. In our opinion the KRMP was more effective than the KR because of the model structure.

Table 1. Correct classification probabilities

Correct classification probabilities					
Dataset	KRMP	KR	SVM	KRMP vs KR	KRMP vs SVM
Ionosphere	0.86	0.80	0.83	>	$\sim$
SPECTF	0.68	0.65	0.65	>	>
Australian	0.73	0.71	0.69	>	>

References

1. D. Cai, X. He, J. Han, *Learning with Tensor Representation*, Preprint (2006).
2. P. Daniušis, P. Vaitkus, Neural network with matrix inputs, *Informatica* (to appear) (2007).
3. S. Hochreiter, K. Obermayer, *Classification, Regression, and Feature Selection on Matrix Data*, Technical report (2004).
4. V.N. Vapnik, *The Nature of Statistical Learning Theory*, Springer-Verlag New York, Inc., New York (1995).

5. Z. Wang, S. Chen, J. Liu, D. Zhang, Pattern representation in feature extraction and classifier design: matrix versus vector, *IEEE Transaction on Neural Networks*, **19**(5), 758–769.
6. Z. Wang, S. Chen, New least squares support vector machines based on matrix patterns, *Neural Processing Letters*, **26**(1), 41–56(16) (2007).

REZIUMĖ

P. Daniušis, P. Vaitkus. Branduolinė regresija matricoms

Šiame straipsnyje pasiūlytas branduolinės regresijos modelis, kai modelio įėjimai yra matricos, pateiktas jo apmokymo algoritmas, pasiūlytas modelis empiriškai palygintas su tradiciniais modeliais.

Raktiniai žodžiai: matriciniai įėjimai, branduoliniai metodai, regresija, klasifikavimas.