Predictive analysis of two bijectively related families of functions in $L^2$, which are expressed as tuple pairs

V G Mosin$^{1,*}$ and A A Abashkin$^1$

$^1$Samara State Technical University, 244 Molodogvardeyskaya str., Samara, 443100, Russia

Abstract. This article proposes a method of function image prediction calculation by its preimage. This method is based on regression analysis of function image and preimage pairs in $L^2$. The prediction model application procedure is described. The algorithm pseudocode are given.

1. Introduction

The machine learning is impotent tool of science data research [1-3]. It lets solve many problems, such as prognoses problem [2, 4, 5]. Prognosing is making by regression model constructing [3, 5, 6, 7]. The objects number must be significantly more than the attributes number [8], otherwise re-education effect is originating [8,[9] and model prediction power goes down [2, 3, 10].

Insufficient data situation often appears in practice (if it’s getting associate with technical and another difficulties), then regression model isn’t applied. Then different sampling and composition methods are used [11-14]. One of these methods is proposed in this article.

2. Problem statement

Let

$$f_i(s) \in L^2[a, b], \quad g_i(t) \in L^2[c, d], \quad i = 0, n$$

be two function families.

Let

$$\theta: L^2[a, b] \rightarrow L^2[c, d]$$

be a continuous map, such that $\theta(f_i) = g_i$. Suppose we know values of the function $f_i(s)$ at the points $s_{ij}, j = 0, p_i$, such that:

$$s_{ij} < s_{i,j+1}, \quad i = 0, n, \quad j = 0, p_i - 1,$$

* Corresponding author: samcocaa@rambler.ru

© The Authors, published by EDP Sciences. This is an open access article distributed under the terms of the Creative Commons Attribution License 4.0 (http://creativecommons.org/licenses/by/4.0/).
\[ s_{i0} = a, \quad s_{i_{pl}} = b, \quad i = 0, n \]

and values of the function \( g_i(t) \) at the points \( t_{ik}, k = 0, q_i \), such that:

\[ t_{ik} < t_{i,k+1}, \quad i = 0, n, \quad k = 0, q_i - 1, \]

\[ t_{i0} = c, \quad t_{iq_i} = d, \quad i = 0, n. \]

Denote \( f_i(s_{ij}) = f_{ij}, g_i(t_{ik}) = g_{ik} \). We have tuple pairs \((s_{ij}, f_{ij}), (t_{ik}, g_{ik})\), and map \( \Theta \) induces bijection

\[ M_\Theta : (s_{ij}, f_{ij}) \rightarrow (t_{ik}, g_{ik}), \quad i = 0, n. \]

Suppose we have another function \( f \) and we know tuple pairs

\[ (s_{j}, f_{j}), \quad j = 0, p. \]

Our problem is to obtain predictive expression of function \( g = \Theta(f) \) as tuple pairs

\[ (t_{k}, g_{k}), \quad k = 0, q. \]

### 3. Data preparing

Important stage of data preparing is its normalization. We normalize tuple pairs

\[ s_{ij}^* = \frac{s_{ij} - a}{b - a}, \quad t_{ik}^* = \frac{t_{ik} - c}{d - c}. \]

Now we get

\[ s_{ij}^* \in [0, 1], \quad t_{ik}^* \in [0, 1]. \]

We create grids for variables \( s^* = \frac{s - a}{b - a} \) and \( t^* = \frac{t - c}{d - c} \). By \( \zeta_l, l = 0, m_\zeta \) and \( \eta_r, r = 0, m_\eta \) denote the values of variables \( s^* \) and \( t^* \) at grid nodes. Values of \( \zeta_l \) and \( \eta_r \) depend on type of the grid. For uniform grid we have

\[ \zeta_l = \frac{l}{m_\zeta}, \quad l = 0, m_\zeta, \quad \eta_r = \frac{r}{m_\eta}, \quad r = 0, m_\eta. \]

If the grid is denser at point 0, we have

\[ \zeta_l = \left( \frac{l}{m_\zeta} \right)^2, \quad l = 0, m_\zeta, \quad \eta_r = \left( \frac{r}{m_\eta} \right)^2, \quad r = 0, m_\eta. \]

If the grid is denser at point 1, we have

\[ \zeta_j = 1 - \left( \frac{l}{m_\zeta} - 1 \right)^2, \quad l = 0, m_\zeta, \quad \eta_r = 1 - \left( \frac{r}{m_\eta} - 1 \right)^2, \quad r = 0, m_\eta. \]

There are many types of grids, that are denser at some points from [0, 1]. In any case we have two grids, such that
\[ \zeta_i \in [0,1], \quad \eta_r \in [0,1]. \]

After the grids were chosen, we complete a definition of values of function \( f \) and function \( g \) at the grid nodes.

At the boundary nodes we have
\[
f_i(\zeta_0) = f_{i,0}, \quad f_i(\zeta_{m_i}) = f_{i,m_i}, \quad i = 0, n, \\
g_i(\eta_0) = g_{i,0}, \quad g_i(\eta_{m_i}) = g_{i,m_i}, \quad i = 0, n.
\]

At the inner nodes we use the following rule.

We fix number \( i \) and study all values of number \( l \). If \( \forall j, \zeta_l \neq s_{ij} \) then unique number \( j_0 \) exists, such that \( s_{ij_0} < \zeta_l < s_{ij_0+1} \). In this case we apply linear interpolation
\[
f_i(\zeta_l) = (f_{i,j_0+1} - f_{i,j_0}) \frac{\zeta_l - s_{ij_0}^*}{s_{ij_0+1}^* - s_{ij_0}^*} + f_{i,j_0}.
\]

Otherwise there is number \( j_0 \), such that \( \zeta_l = s_{ij_0}^* \), in this case we have
\[
f_i(\zeta_l) = f_{i,j_0}.
\]

Similarly, we do calculation for function \( g \). We fix number \( i \) and study all values of number \( r \). If \( \forall k, \eta_r \neq t_{ik} \) then unique number \( k_0 \) exists, such that \( t_{i,k_0} < \eta_r < t_{i,k_0+1} \). In this case we apply linear interpolation
\[
g_i(\eta_r) = (g_{i,k_0+1} - g_{i,k_0}) \frac{\eta_r - t_{i,k_0}^*}{t_{i,k_0+1}^* - t_{i,k_0}^*} + g_{i,k_0}.
\]

Otherwise there is number \( k_0 \), such that \( \eta_r = t_{i,k_0}^* \), in this case we have
\[
g_i(\eta_r) = g_{i,k_0}.
\]

We do this procedure for all \( i, \quad i = 0, n \).

Note, that we use linear interpolation to simplify narrative. We can use any interpolation \([3, 15, 16]\).

Now we have corteges put in the grids:
\[
\begin{array}{cccccccc}
f_0(\zeta_0) & f_0(\zeta_1) & \cdots & f_0(\zeta_{m_0}) & g_0(\eta_0) & g_0(\eta_1) & \cdots & g_0(\eta_{m_0}) \\
f_1(\zeta_0) & f_1(\zeta_1) & \cdots & f_1(\zeta_{m_1}) & g_1(\eta_0) & g_1(\eta_1) & \cdots & g_1(\eta_{m_1}) \\
\cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\
f_n(\zeta_0) & f_n(\zeta_1) & \cdots & f_n(\zeta_{m_n}) & g_n(\eta_0) & g_n(\eta_1) & \cdots & g_n(\eta_{m_n}) \\
\end{array}
\]

Then we do normalization of the grid nodes. For this we denote normalizing value of functions \( f_i \) at points \( \zeta_l \) by \( \varphi_{il} \). Similarly, we denote normalizing value of functions \( g_i \) at the points \( \eta_r \) by \( \psi_{ir} \). We have
\[
\varphi_{il} = \frac{f_i(\zeta_l) - \min_{i \in \mathbb{Z}[0,n]} f_i(\zeta_l)}{\max_{i \in \mathbb{Z}[0,n]} f_i(\zeta_l) - \min_{i \in \mathbb{Z}[0,n]} f_i(\zeta_l)}, \quad \psi_{ir} = \frac{g_i(\eta_r) - \min_{i \in \mathbb{Z}[0,n]} g_i(\eta_r)}{\max_{i \in \mathbb{Z}[0,n]} g_i(\eta_r) - \min_{i \in \mathbb{Z}[0,n]} g_i(\eta_r)}.
\]
Finally, we have two data sets
\[
\begin{array}{cccccc}
\varphi_{00} & \varphi_{01} & \cdots & \varphi_{0m} & \\
\varphi_{10} & \varphi_{11} & \cdots & \varphi_{1m} & \\
\vdots & \vdots & \ddots & \vdots & \\
\varphi_{n0} & \varphi_{n1} & \cdots & \varphi_{nm} & \\
\end{array}
\quad
\begin{array}{cccccc}
\psi_{00} & \psi_{01} & \cdots & \psi_{0m} & \\
\psi_{10} & \psi_{11} & \cdots & \psi_{1m} & \\
\vdots & \vdots & \ddots & \vdots & \\
\psi_{n0} & \psi_{n1} & \cdots & \psi_{nm} & \\
\end{array}
\]

such that
\[
\varphi_{il} \in [0,1], \quad \psi_{ir} \in [0,1].
\]

Further, we will deal with two matrices
\[
\Phi = (\varphi_{il})_{i=0}^{n}_{l=0}^{m}, \quad \Psi = (\psi_{ir})_{i=0}^{n}_{r=0}^{m}.
\]

Rows of matrix $\Phi$ are normalized values of functions $f_i$ at the grid nodes $\zeta_l$. Rows of matrix $\Psi$ are normalized values of functions $g_i$ at grid nodes $\eta_r$. The map $\Theta$ induces bijection between $i$-th row of $\Phi$ and $i$-th row of $\Psi$.

4. Model parameters

We insert column of ones in the end of $\Phi$ and denote obtained matrix by $\tilde{\Phi}$. Let $I$ and $J$ be multi-indexes. Let $\Phi_{IJ}$ be matrix, that consists of $I$-th rows and $J$-th columns elements of $\tilde{\Phi}$. Let $\Psi_{I}$ be matrix, that consists of $I$-th rows elements of $\Psi$. By $\Psi_{Ir}$ denote $r$-th column of $\Psi_I$.

**Definition 1.** We shall say that, **partial regression problem** is overdetermined system of equations
\[
\Phi_{IJ} \cdot \alpha = \Psi_{Ir}.
\]

Note. If that system is determined or underdetermined, then model is overtraining. In order that the above system to be overdetermined it is necessary to have inequality
\[
dim I > \dim J + 1.
\]

**Definition 2.** For partial regression problem we shall say, that **regression dimension** is $\dim J$.

Let $\varphi$ and $\varphi_i$ be two preimage functions, those are expressed as tuple pairs $\varphi_{il}$, $\zeta_l$ and $\varphi_{iL}$, $\zeta_l$. By definition put
\[
\|d(\varphi_i, \varphi_i')\| = \left(\sum_{l=1}^{m} \left(\frac{\varphi_{i,l} + \varphi_{i,l-1}}{2} - \frac{\varphi_{i',l} + \varphi_{i',l-1}}{2}\right)^2 (\zeta_l - \zeta_{l-1})\right)^{1/2}.
\]

**Definition 3.** The number $d(\varphi_i, \varphi_i')$ is called **distance between functions** $\varphi_i$ and $\varphi_i'$. For any given function $\varphi_i$ we can create nondecreasing sequence of distances
\[
d(\varphi_i, \varphi_{i1}) \leq d(\varphi_i, \varphi_{i2}) \leq \cdots \leq d(\varphi_i, \varphi_{ikNN}) \leq \cdots \leq d(\varphi_i, \varphi_{im})
\]
where $kNN$ is number of nearest to $\varphi_i$ functions.

**Definition 4.** We shall say that, **multi-index of the nearest neighbors** of $\varphi_i$ is ordered set
\[ I_{\varphi_i} = \{ i_1, i_2, ..., i_{rNN} \}. \]

For model training we will solve series of partial regression problems for multi-index of the nearest neighbors:

\[ \Phi_{I_{\varphi_i} J} \cdot \alpha = \Psi_{I_{\varphi_i} r}. \]

We number these problems by multi-index \( J \). Therefor we must choose multi-index \( J \) iteration method.

**Definition 5.** One of these methods is **complete enumeration** of all multi-index with fixed length

\[ J = \{ i_1, i_2, ..., i_{\dim J} \}. \]

The number of these multi-indexes is great. We can get it as

\[
\binom{m_\zeta}{\dim J} = \frac{m_\zeta!}{(\dim J)! (m_\zeta - \dim J)!}.
\]

**Definition 6.** Another multi-index iteration method is **enumerating in the ring** \( \mathbb{Z}_p \). For this method dimension \( \dim J \) must be divisor of \( m_\zeta \). We choose multi-indexes by following rule

\[ J = \{ i_1, i_2, ..., i_{\dim J} \} = \left\{ \begin{array}{ll}
0 & \text{mod} \left( \frac{m_\zeta}{\dim J} \right) \\
1 & \text{mod} \left( \frac{m_\zeta}{\dim J} \right) \\
\vdots & \vdots \\
\left( \frac{m_\zeta}{\dim J} - 1 \right) & \text{mod} \left( \frac{m_\zeta}{\dim J} \right)
\end{array} \right. \]

The number of these multi-indexes is \( \frac{m_\zeta}{\dim J} \) and not great.

There are other multi-index iteration methods.

Finally, we have following model parameters: 1) dimension of the partial regression problem \( \dim J \), 2) number of the preimage function nearest neighbors \( kNN \), 3) multi-index iteration method.

**5. Model training**

**Step 0.** We fix following model parameters: \( kNN \), \( \dim J \), and iteration method.

**Step 1.** We choose functions \( \varphi_i \) and \( \psi_i \) and create multi-index of the nearest neighbors

\[ I_{\varphi_i} = \{ i_1, i_2, ..., i_{rNN} \}. \]

**Step 2.** We choose column \( \Psi_I r \).

**Step 3.** We choose multi-index \( J \).

**Step 4.** We create partial regression problem with the multi-indexes \( I_{\varphi_i}, J \) and column \( \Psi_I r \):

\[ \Phi_{I_{\varphi_i} J} \cdot \alpha = \Psi_I r. \]
We solve this problem and get set of numbers \( \alpha_1, \alpha_2, \ldots, \alpha_{\text{dim} f} \) and \( \alpha_0 \). We create linear function with these numbers
\[
\omega_{ijk}(x_1, x_2, \ldots, x_{\text{dim} f}) = \alpha_1 x_1 + \alpha_2 x_2 + \cdots + \alpha_{\text{dim} f} x_{\text{dim} f} + \alpha_0.
\]
We substitute \( \varphi_i, \varphi_i l_1, \varphi_i l_2, \ldots, \varphi_i l_k \) for \( \omega_{ijk}(x_1, x_2, \ldots, x_{\text{dim} f}) \). Number \( \omega_{ijr}(\varphi_i, \varphi_i l_1, \varphi_i l_2, \ldots, \varphi_i l_k) \) is called predictive value of function \( \psi_i \) at \( r \)-th node of grid \( \eta \)
\[
\psi_{ijr} = \omega_{ijr}(\varphi_i, \varphi_i l_1, \varphi_i l_2, \ldots, \varphi_i l_k).
\]
This value depends on multi-index \( f \).

**Go to step 3.** We choose new multi-index \( f \) according to iteration method and repeat step 4. We repeat step 4 for all multi-indexes \( f \).

**Step 5.** We average of the multi-index \( f \) all predictive values \( \psi_{ijr}^{\text{pred}} \). We get predictive value of function \( \psi_i \) at \( k \)-th node of grid \( \eta \). This value doesn’t depend on multi-index \( f \).
Denote this value by \( \psi_{ir}^{\text{pred}} \). We calculate error of prediction value
\[
\varepsilon_{ir} = |\psi_{ir}^{\text{pred}} - \psi_{ir}|.
\]

**Go to step 2.** We choose column \( \Psi_{ir+1} \) and make steps from 3 to 5. We repeat step 2 for all columns of matrix \( \Psi \).

**Go to step 1.** We choose function \( \varphi_i l_1 \) and make steps from 2 to 5. We repeat step 1 for all rows of the matrix \( \Phi \).

**Step 6.** We average of \( i \) and \( r \) values of prediction errors \( \varepsilon_{ir} \),
\[
\varepsilon_i = \frac{1}{m_\eta + 1} \sum_{r=0}^{m_\eta} \varepsilon_{ir}, \quad \varepsilon = \frac{1}{n + 1} \sum_{i=0}^{n} \varepsilon_i.
\]
This value \( \varepsilon \) is averaged predictive error value of model for giving set of parameters.

**Go to step 0.** We choose new values of model parameters and repeat all procedure. We repeat step 0 for all set of model parameters.

**Final of model training.** We find set of parameters with the smallest value of error \( \varepsilon \). We call this set of parameters the optimal one. This is the finale of the model training.

### 6. Model application

Let we have a training model with the optimal parameters: 1) \( kNN \), 2) \( \text{dim} f \) and 3) multi-index iteration method. Let we have preimage function \( f \) as tuple pairs \( f_j, s_j \), where \( f_j = f(s_j) \) and
\[
s_0 = a, \quad s_p = b, \quad s_j < s_{j+1}, \quad j = 0, p.
\]

We want to get predictive expression of image function \( g = \Theta(f) \) as tuple pairs \( g_k, t_k \), where \( g_k = g(t_k) \) and
\[
t_0 = c, \quad t_q = d, \quad t_k < t_{k+1}, \quad k = 0, q.
\]

We begin as we did on data preparing stage. We normalize tuple \( s_j \), linearly interpolate the values \( f(\xi_i) \) at nodes of grid \( \xi \) and normalize values \( f(\xi_i) \) with considering all values \( f_i(\xi_i) \). As result we get tuple \( \varphi_0, \varphi_1, \ldots, \varphi_m \).

We create multi-index of nearest neighbors \( l_{qp} \), then we solve partial regression problem for all multi-indexes \( f \) and columns \( \Psi_{l_{qp} r} \). As result we calculate predictive values \( \psi_{fr}^{\text{pred}} \).
These values we average of \( J \) and get predictive value \( \psi_{r}^{\text{pred}} \) of preimage function \( \psi \) at \( r \)-th node of grid \( \eta \) for ever \( r \). We get cortege of image function predictive values \( \psi_{0}^{\text{pred}}, \psi_{1}^{\text{pred}}, \ldots, \psi_{m_{\eta}}^{\text{pred}} \) on grid \( \eta \).

After these calculations we do transformation, that is invers for normalization transformations. For values \( t_{r} \) we put

\[
t_{r} = (d - c)\eta_{r} + c, \quad r = 0, m_{\eta}.
\]

For values \( g_{r} \) we put

\[
g_{r} = \left( \max_{i \in \mathbb{Z}[0, n]} \psi_{ir} - \min_{i \in \mathbb{Z}[0, n]} \psi_{ir} \right) \psi_{r}^{\text{pred}} + \min_{i \in \mathbb{Z}[0, n]} \psi_{ir}, \quad r = 0, m_{\eta}.
\]

These tuple pairs are expression of function \( g \).

References

1. Poggio T and Shelton C R 1999 AI Magazine 20 37-55
2. Bishop C M 2006 Pattern Recognition and Machine Learning (Berlin: Springer) p 738
3. Clarke B, Fokoue E and Zhang H 2009 Principles and Theory for Data Mining and Machine Learning (Heidelberg: Springer-Verlag) p 781
4. Prakasa Rao B L S 1984 Statist. Probab. Lett. 2(3) 139-142
5. Sieders A and Dzharapradz K A 1987 Ann. Statist. 15(3) 1031-1049
6. Ivanov A V 2017 Theor. Probability and Math. Statist 95 99-108
7. Ivanov A V and Matsak I K 2019 Theor. Probability and Math. Statist 99 91-99
8. Dedieu J-P and Shub M 2000 Mathematics of Computation 69 1099-1115
9. Smale S and Zhou D 2003 Anal. Appl. 1 1-25
10. Barron A R 1994 Machine Learning 14 115-133
11. Cucker F and Smale S 2001 Bull. Amer. Math. Soc. (N.S.) 39 1-49
12. Devore R A 1998 Acta Numer 7 51-150
13. Evgeniou T, Pontil M and Poggio T 2000 Adv. Comput. Math 13 1-50
14. Poggio T and Shelton C R 1999 AI Magazine 20 37-55
15. Micchelli C A 1986 Constr. Approx 2 11-22
16. Meinguet J 1979 Appl. Math. Phys 30 292-304