Исследование способов согласования моделей с помощью снижения размерности пространства

Ф. Р. Яушев, Р. В. Исаченко, В. В. Стрижов
yaushev@phystech.edu; roman.isachenko@phystech.edu; strijov@ccas.ru

В работе исследуется задача прогнозирования сложной целевой переменной. Под сложностью подразумевается наличие зависимостей (линейных или нелинейных). Предполагается, что исходные данные гетерогенны. Это значит, что пространства независимой и целевой переменных имеют разную природу. Предлагается построить предсказательную модель, которая учитывает зависимость в исходном пространстве независимой переменной, а также в пространстве целевой переменной. Согласование моделей предлагается производить в низкоразмерном пространстве. В качестве базового алгоритма используется метод проекции в скрытое пространство (PLS). В работе проводится сравнение линейного PLS и предложенных нелинейных моделей. Сравнение производится на гетерогенных данных в пространствах высокой размерности.

Ключевые слова:

1 Введение

В данной работе решается задача прогнозирования целевой переменной с наличием зависимостей. Трудность задачи в том, что исходные данные имеют высокую размерности и в пространствах целевой и независимой переменных есть скрытые зависимости. Чрезмерно высокая размерность пространств и наблюдаемая множественная корреляция приводят к неустойчивости модели. Для решения предлагается построить модель, которая бы учитывала обе эти зависимости. Она переводит данные в низкоразмерные пространства и согласование данных происходит в полученном скрытом пространстве.

Метод проекции в скрытое пространство (Projection to Latent Space, PLS) [1,2] восстанавливает зависимости между двумя наборами данных. Он применяется в биоинформатике, медицине, социальных науках [3–6]. Алгоритм PLS строит матрицу совместного описания признаков и целевой переменной. Полученное пространство является низкоразмерным. Это позволяет получить простую, точную и устойчивую прогностическую модель. Наряду с PLS используется алгоритм CCA [7]. CCA применяется для поиска зависимостей между двумя наборами данных и получения их низкоразмерного представления [8,9]. CCA максимизирует корреляции, а PLS — ковариации. Обзор и сравнение CCA и PLS приводится в [1]. PLS и CCA — линейные модели, которые игнорируют сложные нелинейные зависимости.

Задачи, в которых между данными существует нелинейная зависимость описаны в работе [**?**]. Аппроксимация этой зависимость линейной PLS моделью приводит к неудовлетворительным результатам. Разработано нелинейные модификации PLS [10–12] и CCA [13,14]. Например, Deep CCA [13] преобразует исходные данные с помощью нейронной сети таким образом, что результирующее представление становится согласованным. Deep CCA используется для генерации текстового описания по изображениям в работе [15].

В работе проведено два эксперимента. Первый эксперимент направлен на сравнение эффективности Deep CCA и CCA на задаче классификации зашумленных цифровых изображений MNIST [16]. Во втором эксперименте используется набор данных, полученный делением каждого изображения из MNIST на левую и правую части. На задаче регрессии правой части изображения по левой проводится сравнение нелинейных моделей с применением автоэнкодеров, моделей без преобразования данных и линейного PLS. На основании полученных результатов сделан вывод о точности и сложности нелинейных алгоритмов и о целесообразности использования той или иной модели.

2 Постановка задачи

Пусть дана выборка $(\mathbf{X}, \mathbf{Y})$, где $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_n]^\mathsf{T} \in \mathbb{R}^{n\times m}$ — матрица независимых переменных, $\mathbf{Y} = [\mathbf{y}_1, \ldots, \mathbf{y}_n]^\mathsf{T} \in \mathbb{R}^{n\times k}$ — матрица целевых переменных.
Предполагается, что между $\mathbf{X}$ и $\mathbf{Y}$ существует зависимость

$$\mathbf{Y} = f(\mathbf{X}) + \boldsymbol{\varepsilon}, \tag{1}$$

где $f : \mathbb{R}^{n\times m} \to \mathbb{R}^{n\times k}$ — функция регрессионной зависимости, $\boldsymbol{\varepsilon}$ — матрица регрессионных ошибок.

Необходимо восстановить зависимость f по заданной выборке.

2.1 Линейная регрессия

Предположим, что зависимость (1) линейна. Требуется найти эту зависимость:

$$\mathbf{Y} = f(\mathbf{X}) + \boldsymbol{\varepsilon} = \mathbf{X}\mathbf{W}^\mathsf{T} + \boldsymbol{\varepsilon}, \tag{2}$$

где $\mathbf{W} \in \mathbb{R}^{k\times m}$ – матрица параметров модели.
Оптимальные параметры определяются минимизацией функции потерь. Используется квадратичная функция потерь:

$$\mathcal{L}(\mathbf{W}|\mathbf{X}, \mathbf{Y}) = \left\| \underset{n\times k}{\mathbf{Y}} - \underset{n\times m}{\mathbf{X}} \cdot \underset{m\times k}{\mathbf{W}}{}^\mathsf{T} \right\|_2^2 \to \min_{\mathbf{W}}. \tag{3}$$

Решение (3) имеет следующий вид:

$$\mathbf{W} = \mathbf{Y}^\mathsf{T}\mathbf{X}(\mathbf{X}^\mathsf{T}\mathbf{X})^{-1}.$$

Линейная зависимость столбцов матрицы $\mathbf{X}$ приводит к неустойчивости решения задачи минимизации (3), так как в этом случае матрица $\mathbf{X}^\mathsf{T}\mathbf{X}$ является плохо обусловленной. Для борьбы с линейной зависимостью используются методы снижения размерности, путем перехода в низкоразмерное латентное пространство.

Определение 1. *Параметрическая функция $\varphi_1 : \mathbb{R}^{n\times m} \to \mathbb{R}^{n\times p}$, переводящая исходных данных в латентное пространство, называется* ***функцией кодирования****.*

Определение 2. *Функция $\varphi_2 : \mathbb{R}^{n\times k} \to \mathbb{R}^{n\times p}$, переводящая данные из латентного пространства в исходное, называется* ***функцией восстановления****.*

Определение 3. *Функция $g : \mathbb{R}^{n\times p} \times \mathbb{R}^{n\times p} \to \mathbb{R}$, связывающая закономерности в низкоразмерных латентных представления, называется* ***функцией согласования****.*

Определение 4. *Согласование — процесс максимизации функции согласования.*

2.2 Снижение размерности

Общая схема модели выглядит следующим образом:

$$\begin{tikzcd}
\underset{n\times m}{\mathbf{X}} \arrow[r, "f"] \arrow[d, bend right, "\varphi_1"'] & \underset{n\times k}{\mathbf{Y}} \arrow[d, bend left, "\psi_1"] \\
\underset{n\times p}{\mathbf{T}} \arrow[u, "\varphi_2"'] \arrow[r, leftrightarrow, "g"] & \underset{n\times p}{\mathbf{U}} \arrow[u, "\psi_2"]
\end{tikzcd} \tag{4}$$

где $\varphi_1 : \mathbb{R}^{n\times m} \to \mathbb{R}^{n\times p}$ — функция кодирования независимых переменных; $\psi_1 : \mathbb{R}^{n\times k} \to \mathbb{R}^{n\times p}$ — функция кодирования целевых переменных; $\varphi_2 : \mathbb{R}^{n\times p} \to \mathbb{R}^{n\times m}$ — функция восстановления независимых переменных; $\psi_2 : \mathbb{R}^{n\times p} \to \mathbb{R}^{n\times k}$ - функция восстановления целевых переменных; $g : \mathbb{R}^{n\times p} \times \mathbb{R}^{n\times p} \to \mathbb{R}$ — функция согласования.
$\mathbf{T} = \varphi_1(\mathbf{X}) \in \mathbb{R}^{n\times p}$ и $\mathbf{U} = \psi_1(\mathbf{Y}) \in \mathbb{R}^{n\times p}$ — матрицы представлений данных в латентном пространстве низкой размерности.

Оптимальные параметры $\theta^*_{\varphi_1}, \theta^*_{\psi_1}$ для функций кодирования φ_1 и ψ_1 находятся из следующей задачи параметрической оптимизации:

$$(\theta^*_{\varphi_1}, \theta^*_{\psi_1}) = \mathop{\arg\max}_{(\theta_{\varphi_1}, \theta_{\psi_1})} [g\big(\varphi_1(\mathbf{X}; \theta_{\varphi_1}), \psi_1(\mathbf{Y}; \theta_{\psi_1})\big)]. \tag{5}$$

Так как параметры функции кодирования подбираются из условия максимизации функции согласования (5), то после перехода в латентное пространство между $\mathbf{T}$ и $\mathbf{U}$ существует зависимость

$$\mathbf{U} = h(\mathbf{T}) + \boldsymbol{\eta}, \tag{6}$$

где $h : \mathbb{R}^{n\times p} \to \mathbb{R}^{n\times p}$ — функция регрессионной зависимости, $\boldsymbol{\eta}$ — матрица регрессивных ошибок.
Оптимальная h выбирается минимизацией функции ошибки. Используем квадратичную функцию ошибки потерь $\mathcal{L}$ на $\mathbf{T}$ и $\mathbf{U}$:

$$\mathcal{L}(h|\mathbf{T}, \mathbf{U}) = \Big\| \underset{n\times p}{\mathbf{U}} - h(\underset{m\times p}{\mathbf{T}}) \Big\|_2^2 \to \min_h . \tag{7}$$

Финальная прогностическая модель имеет вид: $\widehat{\mathbf{y}} = \psi_2\big(h(\varphi_1(\mathbf{x}))\big)$, то есть

$$f = \psi_2 \circ h \circ \varphi_1. \tag{8}$$

2.3 Метод Главных Компонент (PCA)

PCA — способ снижения размерности данных, сохраняющий максимальную дисперсию. PCA представляет собой ортогональное линейное преобразование исходного признакового пространства в новое пространство меньшей размерности. Первый базисные векторы строятся так, чтобы выборочная дисперсия данных вдоль них была максимальной:

$$\mathbf{p} = \mathop{\arg\max}_{\|\mathbf{p}\|_2 = 1} [\mathbf{var}(\mathbf{Xp})], \tag{9}$$

где $\mathbf{var}(\mathbf{Xp}) = \frac{1}{n}(\mathbf{Xp})^\mathsf{T}\mathbf{Xp}$ обозначает выборочную дисперсию.

Функция кодирования $\varphi_1 : \mathbb{R}^{n\times m} \to \mathbb{R}^{n\times p}$ имеет вид:

$$\varphi_1(\mathbf{X}) = \underset{n\times m}{\mathbf{X}} \underset{m\times p}{\mathbf{P}}^\mathsf{T}, \tag{10}$$

где $\mathbf{P} = [\mathbf{p}_1, \dots, \mathbf{p}_p]$.
PCA не согласует независимые переменные и целевые переменные. Из-за этого зависимости в обоих пространствах не учитываются.

2.4 PLS

PLS — алгоритм для восстановления связи между двумя наборами данных $\mathbf{X}$ и $\mathbf{Y}$. Алгоритм проецирует $\mathbf{X}$ и $\mathbf{Y}$ на латентное пространство $\mathbb{R}^p$ меньшей размерности. PLS находит матрицы исходных данных $\mathbf{X}$ и $\mathbf{Y}$ в латентном пространстве $\mathbf{T}$ и $\mathbf{U}$ соответственно. Матрица объектов $\mathbf{X}$ и целевая матрица $\mathbf{Y}$ проецируются на латентное пространство следующим образом:

$$\underset{n\times m}{\mathbf{X}} = \underset{n\times p}{\mathbf{T}} \cdot \underset{p\times m}{\mathbf{P}}^{\mathsf{T}} + \underset{n\times m}{\mathbf{F}}, \tag{11}$$

$$\underset{n\times k}{\mathbf{Y}} = \underset{n\times p}{\mathbf{U}} \cdot \underset{p\times k}{\mathbf{Q}}^{\mathsf{T}} + \underset{n\times k}{\mathbf{E}}, \tag{12}$$

где $\mathbf{T}$ и $\mathbf{U}$ — матрицы описания объектов и исходов в латентном пространстве; $\mathbf{P}$ и $\mathbf{Q}$ — матрицы перехода из латентного пространства в исходное; $\mathbf{F}$, $\mathbf{E}$ — матрицы остатков.

Для PLS функции кодирования имеют вид:

$$\varphi_1(\mathbf{X}) = \mathbf{X}\mathbf{W}_\mathbf{x}, \quad \psi_1(\mathbf{Y}) = \mathbf{Y}\mathbf{W}_\mathbf{y}, \tag{13}$$

где матрицы весов $\mathbf{W}_\mathbf{x} \in \mathbb{R}^{m\times p}, \mathbf{W}_\mathbf{y} \in \mathbb{R}^{k\times p}$ находятся путем максимизации функции согласования $g(\mathbf{X}\mathbf{W}_\mathbf{x}, \mathbf{Y}\mathbf{W}_\mathbf{y}) = \mathbf{Cov}(\mathbf{X}\mathbf{W}_\mathbf{x}, \mathbf{Y}\mathbf{W}_\mathbf{y})^2$:

$$(\mathbf{W}_\mathbf{x}, \mathbf{W}_\mathbf{y}) = \underset{\mathbf{W}_\mathbf{y}, \mathbf{W}_\mathbf{y}}{\arg\max}[\mathbf{Cov}(\mathbf{X}\mathbf{W}_\mathbf{x}, \mathbf{Y}\mathbf{W}_\mathbf{y})^2], \tag{14}$$

где $\mathbf{Cov}(\mathbf{X}\mathbf{W}_\mathbf{x}, \mathbf{Y}\mathbf{W}_\mathbf{y})$ — выборочная ковариация.
Функции восстановления принимают вид:

$$\varphi_2(\mathbf{T}) = \mathbf{T}\mathbf{P}^{\mathsf{T}}, \quad \psi_2(\mathbf{U}) = \mathbf{U}\mathbf{Q}^{\mathsf{T}}. \tag{15}$$

2.5 CCA

Канонический анализ корреляций (CCA) находит два набора базисных векторов $\{\mathbf{w}_{\mathbf{x}i}\}_{i=1}^p$, $\mathbf{w}_\mathbf{x} \in \mathbb{R}^m$ и $\{\mathbf{w}_{\mathbf{y}i}\}_{i=1}^p$, $\mathbf{w}_\mathbf{y} \in \mathbb{R}^k$, один для $\mathbf{X}$ и другой для $\mathbf{Y}$, так что коэффициент корреляция между проекциями переменных на эти базисные векторы была максимальной. Функция согласования для CCA

$$g(\mathbf{X}\mathbf{W}_\mathbf{x}, \mathbf{Y}\mathbf{W}_\mathbf{y}) = \mathbf{corr}(\mathbf{X}\mathbf{W}_\mathbf{x}, \mathbf{Y}\mathbf{W}_\mathbf{y}), \tag{16}$$

где $\mathbf{corr}(\mathbf{X}\mathbf{w}_\mathbf{x}, \mathbf{Y}\mathbf{w}_\mathbf{y})$ – коэффициент корреляции между векторами.
Таким образом, функции кодирования

$$\varphi_1(\mathbf{X}) = \mathbf{X}\mathbf{W}_\mathbf{x}, \quad \psi_1(\mathbf{Y}) = \mathbf{Y}\mathbf{W}_\mathbf{y}, \tag{17}$$

где первые столбцы матриц весов находится, как вектора максимизирующие функцию согласования g. Далее ищутся вектора, максимизирующие g, но с ограничением, что они не коррелируют с первой парой векторов. Процедура продолжается до тех пор, пока количество векторов не станет равным p.

2.6 Deep CCA

Deep CCA — нелинейная модификация CCA. Deep CCA преобразует исходные данные с помощью нейронной сети таким образом, что результирующее представление становится согласованным. Предполагается, что есть d слоев нейронной сети.

Обозначим $\theta_\mathbf{x}$, $\theta_\mathbf{y}$ – параметры для функций кодирования, то есть матрицы весов и векторы смещений. Оптимальные параметры $\theta_\mathbf{x}^*$, $\theta_\mathbf{y}^*$ находятся из задачи оптимизации:

$$(\theta_\mathbf{x}^*, \theta_\mathbf{y}^*) = \underset{(\theta_\mathbf{x}, \theta_\mathbf{y})}{\arg\max}[g(\varphi_1(\mathbf{X}; \theta_\mathbf{x}), \psi_1(\mathbf{Y}; \theta_\mathbf{y}))] = \underset{(\theta_\mathbf{x}, \theta_\mathbf{y})}{\arg\max}[\mathbf{corr}(\varphi_1(\mathbf{X}; \theta_\mathbf{x}), \psi_1(\mathbf{Y}; \theta_\mathbf{y}))]. \tag{18}$$

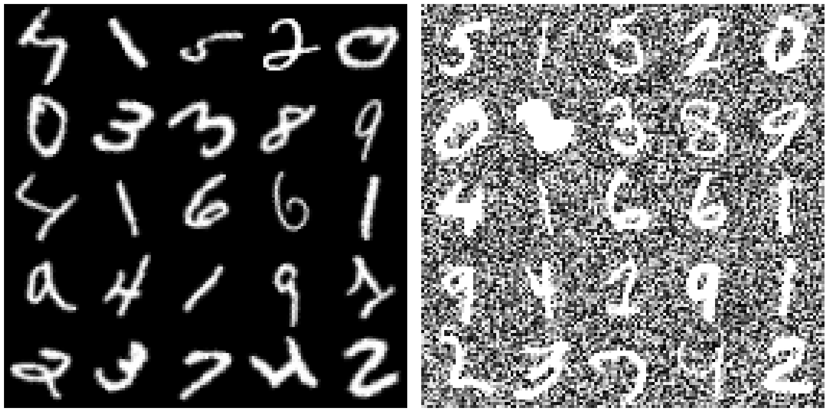

Рис. 1 Зашумленные изображений из набора данных MNIST

Таблица 1 Получение нового признакового пространство размерности 15 с использованием DCCA и CCA. Показателем эффективности будет точность классификации линейного SVM (ACC).

	DeepCCA(L=3)	CCA
Validation data	92.74%	76.21%
Test data	92.14%	76.07%

2.7 Эксперимент №1

Проведем сравнение качества DeepCCA и CCA на задаче классификации зашумленных цифровых изображений (1). Для этого используется набор данных MNIST [16], который состоит из 70000 цифровых изображений 28×28 образцов рукописного написания цифр. Предлагается получить два новых набора данных $\mathbf{X}$ и $\mathbf{Y}$ следующим образом. Первый набор получим поворотом исходных изображений на угол в диапазоне $[\frac{-\pi}{4}, \frac{\pi}{4}]$. Для получения второго набора данных для каждой картинки из первого набора данных ставится в соответствие случайным образом картинка с той же цифрой, но с добавлением независимого случайного шума, распределенного равномерно на отрезке $[0, 1]$.

Применив к двум новым наборам данных DeepCCA или CCA, мы получаем новое низкоразмерное признаковое пространство, которое игнорирует шумы в исходных данных. Таким образом, получаем функции кодирования φ_1 и ψ_1 для исходных наборов данных. На новых признаках, полученных разными моделями (DeepCCA и CCA), для первого набора данных, то есть на данных после применения функции кодирования φ_1 к первому набору исходных данных, обучим линейный SVM классификатор. Показателем эффективности будет доля правильных ответов SVM на новых данных — accuracy (ACC). Результаты эксперимента приведены в таблице (1).

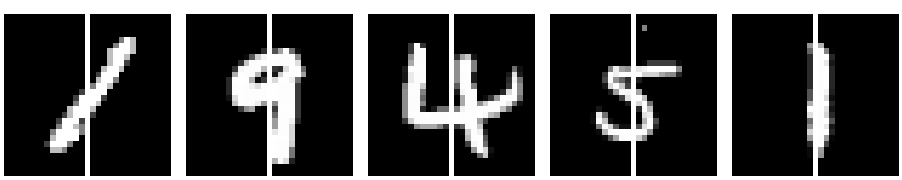

Рис. 2 Набор данных MNIST, каждое изображение в котором разделили пополам.

Таблица 2 Восстановление правой части изображения по левой с использованием различных моделей. Для измерения качества моделей считается среднеквадратическое отклонения от оригинального изображения.

	EncNet1	LinNet1	EncNet2	LinNet2	DumbNet	PLS
Кол-во весов	283k	239k	283k	239k	283k	-
MSE loss on test data	0.147	0.235	0.149	0.236	0.128	0.188

2.8 Эксперимент №2

Проведем на задаче регрессии сравнение нескольких моделей, которые используют автоэнкодеры для снижения размерности пространства, моделей без преобразования исходных данных и линейный PLS. Для этого используется набор данных MNIST. Каждое изображение делится на левую и правую части (2). Модели по левому изображению предсказывают правое (3).

Модель EncNet1 — нейронная сеть с нелинейными функциями активации, которая обучается на данных после преобразования их автоэнкодером. Модель LinNet1 — нейронная сеть с одним линейным слоем, которая также обучается на преобразованных данных. Для EncNet1 и LinNet1 автоэнкодеры для объектов и ответов используют совместную функцию потерь, которая связывает выходы енкодеров. Модели EncNet2 и LinNet2 устроены аналогично EncNet1 и LinNet1 соответственно, но в автоэнкодерах нет совместной функции потерь. Модель DumbNet — нейронная сеть, которая обучается на исходных данных и имеет такую же структуру, что и EncNet, то есть имеет такое же количество слоев и в каждом слое такое же количество нейронном, что и у EncNet.

Для измерения качества моделей будет считать среднеквадратическое отклонение. Результаты работы алгоритмов показаны на изображении (3). Качество работы моделей, а также их сложность представлены в таблице (2).

Литература

[1] Roman Rosipal and Nicole Kramer. Overview and recent advances in partial least squares. *C. Saunders et al. (Eds.): SLSFS 2005*, (LNCS 3940):34–51, 2006.

[2] Roman Rosipal. Nonlinear partial least squares: An overview. *Chemoinformatics and Advanced Machine Learning Perspectives*, (Complex Computational Methods and Collaborative Techniques):169–189, 2011.

[3] D.V. Nguyen and D.M. Rocke. Tumor classification by partial least squares using microarray gene expression data. *Bioinformatics*, (18):39–50, 2002.

[4] K.J. Worsley. An overview and some new developments in the statistical analysis of pet and fmri data. *Human Brain Mapping*, (5):254–258, 1997.

[5] J.S. Hulland. Use of partial least squares (pls) in strategic management research: A review of four recent studies. *Strategic Management Journal*, (20):195–204, 1999.

[6] P.E. Shalamu Abudu J. Phillip King and Thomas C. Pagano. Application of partial least-squares regression in seasonalstreamflow forecasting. *Journal of Hydrologic Engineering*, (15(8)), 2010.

[7] S. R. Szedmak D. R. Hardoon and J. R. Shawe-taylor. Canonical correlation analysis: An overview with application to learning methods. *Neural Comput*, (vol. 16, no. 12):2639–2664, 2004.

[8] Y. Y. Schechner E. Kidron and M. Elad. Pixels that sound. *IEEE Computer Society*, page 88–95, 2005.

[9] S. Ji L. Sun and J. Ye. A least squares formulation for canonical correlation analysis. *ICML*, page 1024–1031, 2008.

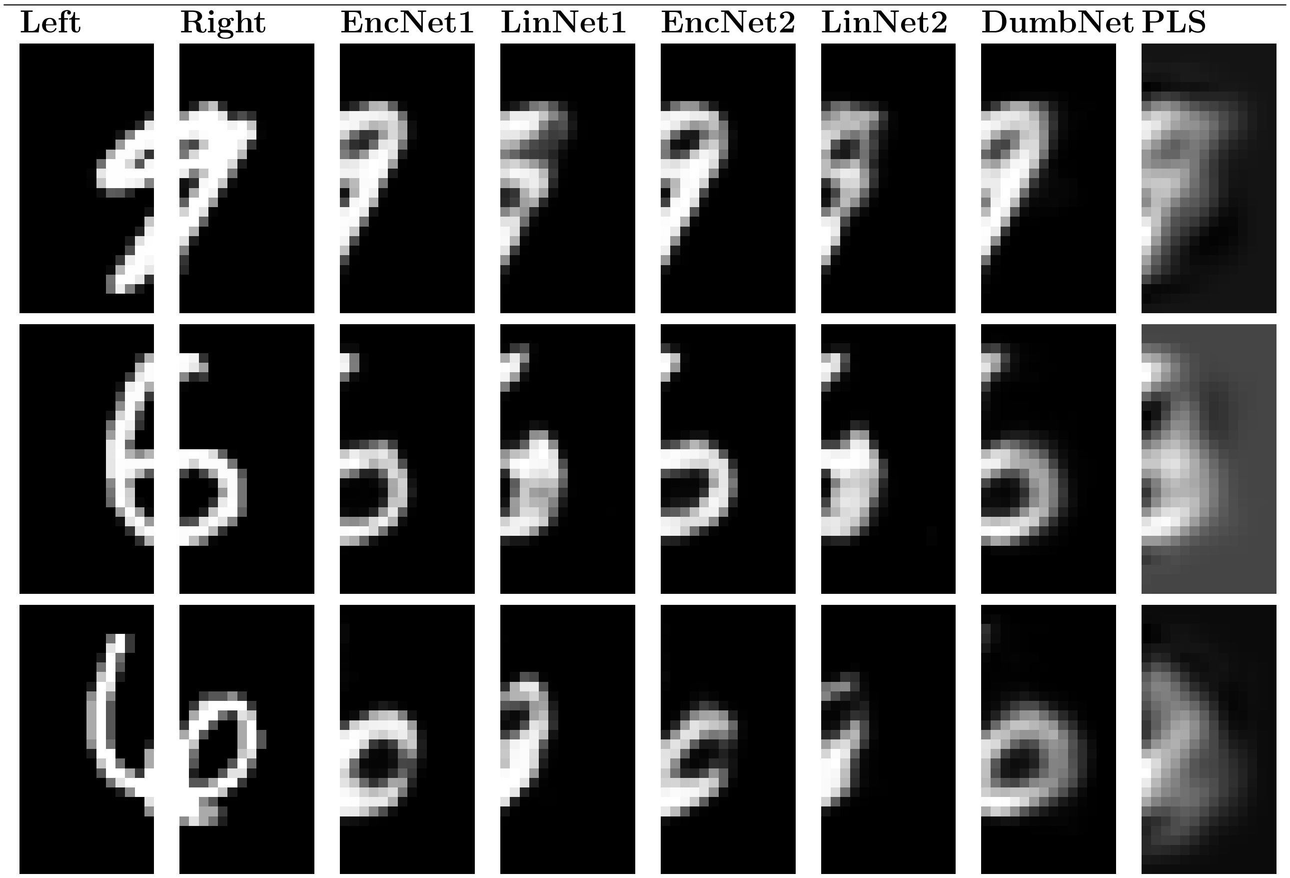

Рис. 3 Восстановление правой части изображения по левой.

[10] S Joe Qin and Thomas J McAvoy. Nonlinear pls modeling using neural networks. *Computers Chemical Engineering*, (16(4)):379–391, 1992.

[11] Dezhao Z. Chen Xuefeng F. Yan and Shangxu X. Hu. Chaos-genetic algorithms for optimizing the operating conditions based on rbf-pls model. *Computers and Chemical Engineering*, (27(10)):1393–1404, 2003.

[12] Mark Willis Hugo Hiden, Ben McKay and Gary Montague. Non-linear partial least squares using genetic. *Computers and Chemical Engineering In Genetic Programming 1998*, (Proceedings of the Third):128–133, 1998.

[13] J. A. Bilmes G. Andrew, R. Arora and K. Livescu. Deep canonical correlation analysis. *ICML*, page 1247–1255, 2013.

[14] P. L. Lai and C. Fyfe. Kernel and nonlinear canonical correlation analysis. *IJCNN*, (4):614, 2000.

[15] F. Yan and K. Mikolajczyk. Deep correlation for matching images and text. *CVPR*, (4):3441–3450, 2015.

[16] Y. LeCun and C. Cortes. The mnist database of handwritten digits. 1998.

Поступила в редакцию