Применение методов машинного обучения для оптимального хранения информации о вертикальном разрезе скорости звука

В. О. Захаров[1],* М. В. Лебедев[2]†
[1] *Федеральное государственное бюджетное образовательное учреждение высшего образования «Московский авиационный институт (национальный исследовательский университет)» Россия, 125993 Москва, Волоколамское шоссе, д. 4*
[2] *АО Акустический Институт им. Академика Н.Н. Андреева Россия, 117036 Москва, ул. Шверника, д. 4*
(Статья поступила 14.11.2019; подписана в печать 25.11.2019)

В настоящей работе представлены результаты исследования различных методов сжатия информации о вертикальном разрезе скорости звука (ВРСЗ), а именно: метод главных компонент, нейронные сети, K-SVD. Также приведено сравнение этих методов сжатия информации с точки зрения наилучшего восстановления информации о месяце в котором производилось измерение ВРСЗ. Для этого использовались следующие методы классификации: бустинг, логистическая регрессия, случайный лес. Все исследования проводились с базой профилей Баренцева моря.

PACS: 02.60.-x, 02.60.Ed
Ключевые слова: машинное обучение, методы снижения размерности, ВРСЗ, метод главных компонент

ВВЕДЕНИЕ

Хорошо известно, что на точность расчетов гидроакустического поля большое влияние оказывает точность с которой было произведено измерение ВРСЗ [1]. В данной работе рассматриваются оптимальные методы параметризации профиля ВРСЗ для последующего его хранения и обработки. Оптимальность понимается в смысле высокой точности восстановления информации о ВРСЗ и минимума объема информации требуемой для его хранения.

В ряде практических задач требуется восстановить профиль ВРСЗ по акустическим данным [2]. Чтобы результат был физически реалистичным, профиль ВРСЗ представляют в виде суммы эмпирических ортогональных функций [1, 3]. Но это часто приводит к низкой точности оценки ВРСЗ и большому количеству параметров участвующих в описании профилей.

В настоящей работе исследуется подход, основанный на построении базисных функций, на основе имеющейся информации о профилях с использованием методов снижения размерности [5, 6, 7]. В работе [8] подробно рассмотрена подобная задача. Отличительной особенностью настоящей работы является следующее: рассматривается распределение профилей не во времени, а в пространстве; экспериментальные данные о ВРСЗ взяты из другого региона, более актуального для России; рассмотрен ряд новых идей для улучшения оценки качества восстановления информации о ВРСЗ. По сравнению с более ранней работой [4] в предложенной статье продемонстрирована работа алгоритма DL, а также приведена модификация ранее использованного автокодировщика.

*E-mail: aladin7806@yandex.ru
†E-mail: max_lebedev@mail.ru

1. ПОСТАНОВКА ЗАДАЧИ СНИЖЕНИЯ РАЗМЕРНОСТИ

Задано множество профилей ВРСЗ $\mathcal{Y} = \{y_i \in \mathbb{R}^n, i = 1, \ldots, m\}$. Каждый профиль y_i это вектор размерности n, где n — количество горизонтов по глубине где производится измерение скорости звука (см. рис. 1). Глубина для каждого горизонта зафиксирована для всех профилей. Количество горизонтов также зафиксировано. i — определяет номер для географической координаты где было произведено измерения для данного профиля. Количество глубин зафиксировано. Таким образом, в векторе ВРСЗ y_i хранятся только значения скоростей распространения звука.

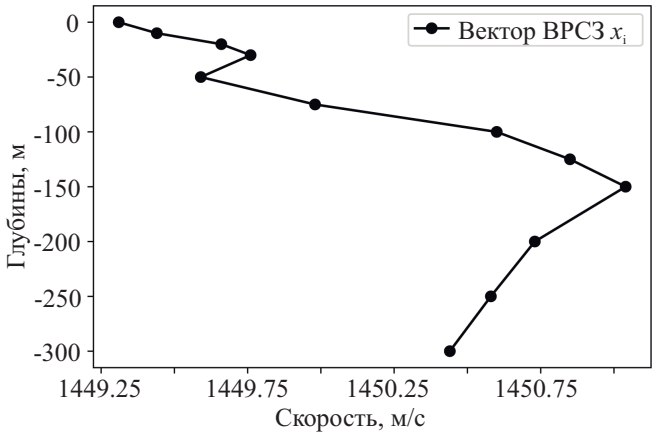

Рис. 1: Вектор одного профиля ВРСЗ y_i

В случае если измерения для ВРСЗ были произведены для разных глубин, предварительно требуется разбить все множество профилей на подмножества с одинаковым набором глубин (горизонтов), а далее применять рассмотренные далее алгоритмы для каждого из подмножеств отдельно.

Для эффективного сжатия информации о ВРСЗ требуется найти преобразование $\mathbf{F}$ которое переводит ис-

ходный профиль y_i в пространство меньшей размерности $\mathbb{R}^k$ $(k < n)$. Кроме того, должно быть задано обратное преобразование $\mathbf{F}^{-1}$ в исходное пространство $\mathbb{R}^n$. Это преобразование позволяет восстанавливать исходный профиль ВРСЗ из пространства меньшей размерности (пространства сжатых параметров). Очевидно, что при таком преобразовании может возникнуть ошибка. В общем случае требуется найти преобразования $\mathbf{F}$ и $\mathbf{F}^{-1}$ при которых достигается наименьшая ошибка при восстановлении произвольного профиля $y \in \mathcal{Y}$

$$\|\mathbf{F}^{-1}\mathbf{F}(y) - y\| \to \min_{\mathbf{F}}.$$

Для анализа точности данного преобразования будет использоваться подход, основанный на идеях машинного обучения. Он заключается в том, что все множество профилей $\mathcal{Y}$ разбивается на два не пересекающихся подмножества: обучающее $\mathcal{Y}_i^l \subset \mathcal{Y}$ и тестовое $\mathcal{Y}_i^t \subset \mathcal{Y}$, а $\mathcal{Y} = \mathcal{Y}_i^l \cup \mathcal{Y}_i^t$. Далее этот процесс повторяется $i = 1, \ldots, p$ раз. На основе обучающего множества $\mathcal{Y}_i^l \subset \mathcal{Y}$ строится преобразование $\mathbf{F}_i$. Тестовое множество служит для проверки качества (точность восстановления) построенного преобразования. Для оценки этого качества будет использоваться оценка, полученная на основе скользящего контроля:

$$\begin{aligned} RMSE &= \frac{1}{p}\sum_i \rho(\mathbf{F}_i, \mathcal{Y}_i^t), \\ \rho(\mathbf{F}, \mathcal{Y}) &= \sqrt{\frac{1}{|\mathcal{Y}|}\sum_{y\in\mathcal{Y}} \|\mathbf{F}^{-1}\mathbf{F}(y) - y\|_2^2}, \end{aligned} \tag{1}$$

где $|\mathcal{Y}|$ — количество профилей в $\mathcal{Y}$.

Алгоритмы снижения размерности можно условно разделить на три вида.

1. Восстановленный профиль $\hat{y}_i$ получается за счет аффинного преобразования в векторном пространстве:

$$\hat{y} = \mathbf{F}^{-1}x_i = Dx_i + b,$$

где $D \in \mathbb{R}^{n\times k}$ — представляет собой матрицу базовых профилей (словарь), а b — вектор смещения равный в большинстве случаев усредненному профилю ВРСЗ. $x_i \in \mathbb{R}^k$ — представление профиля в сжатом пространстве. Таким образом, вместо хранения всех векторов y_i, $i = 1, \ldots, m$ которые представляют собой таблицу размера $m \cdot n$, достаточно хранить векторы коэффициентов сжатого пространства x_i для каждого профиля y_i, т.е. таблицу размера $m \cdot k$, $k < n$.

2. $\mathbf{F}$ строится на основе нелинейного преобразования (например, нейронная сеть с весовыми коэффициентами). Для восстановления информации достаточно хранить векторы коэффициентов x_i при нелинейных преобразованиях.

3. Также сжатие информации в первых двух случаях можно достигнуть не за счет уменьшения размерности векторов x_i, а за счет их разреженной структуры (добиться, чтобы x_i содержали много нулевых элементов).

2. МЕТОД ГЛАВНЫХ КОМПОНЕНТ

Данный метод относится к первому типу, указанных выше алгоритмов. Для сжатия информации в начале составляется матрица из центрированных векторов обучающей выборки:

$$Y = [\mathring{y_1}, \ldots, \mathring{y_l}]^T, \quad \mathring{y_i} = y_i - \overline{y}, \quad \overline{y} = \frac{1}{l}\sum_{i=1}^{l} y_i, \quad Y \in \mathbb{R}^{l\times n}.$$

Далее строится SVD разложение матрицы Y: $Y = USV^T$. Тогда в качестве словаря будет являться матрица $D = V_{i,j}$, $i = 1, \ldots, n$, $j = 1, \ldots, k$, т.е. словарь составлен из первых k столбцов матрицы V. Тогда преобразования $\mathbf{F}, \mathbf{F}^{-1}$ для произвольного вектора профиля y будут иметь следующий вид:

$$x = \mathbf{F}y = D^T(y - \overline{y}), \quad \hat{y} = \mathbf{F}^{-1}x = Dx + \overline{y}, \quad x \in \mathbb{R}^k,$$

где $\hat{y}$ — восстановленный профиль для y, x — координаты профиля в сжатом пространстве. Метод главных компонент (PCA), использованный в работе, подробно описан в [5].

3. МЕТОД K-SVD

Теперь рассмотрим алгоритмы которые позволяют получить существенно разреженное представление профилей ВРСЗ в сжатом пространстве. Общая постановка такой задачи выглядит следующим образом:

$$\begin{aligned} (\hat{D}, \hat{X}) &\in \arg\min \|Y - DX\|_F^2 \\ \|x_i\|_0 &\le T, \quad i = 1, \ldots, k, \end{aligned} \tag{2}$$

где $\|x\|_0$ — количество ненулевых элементов в векторе x, T — число, ограничивающее сверху количество ненулевых элементов в векторе (оно определяет степень разреженности матрицы $\hat{X}$). Известно [6], что такая задача решается только полным перебором. Поэтому рассматриваются альтернативные постановки [6, 9, 10, 11, 12], в которых например [11] вместо нормы $\|\cdot\|_0$ используется норма $\|\cdot\|_1$. Такого рода алгоритмы в общем случае служат для приближенного решения данной задачи (2). К ним относятся MP [9], OMP [6, 10], BP [11], FOCCUS [12] и другие. Для решения задачи (2) и построения словаря $\hat{D}$ в работе будет использоваться модификация OMP алгоритма [6], которая реализована в пакете python k-svd [14].

4. МЕТОД DICTIONARY LEARNING

В пакете scikit-learN [15] реализована другая идея [7] поиска разреженного представления для матрицы данных $\hat{X}$. Коэффициенты разложения $\hat{X}$ и словарь $\hat{D}$ находятся из решения оптимизационной задачи

$$(\hat{D},\hat{X}) = \arg\min_{D,X} \frac{1}{2}\|Y - DX\|_2^2 + \lambda\|D\|_1,$$
$$\|x_i\|_2 = 1, \quad i = 1\dots k,$$

где λ — параметр регуляризации. В работе используется метод DictonaryLearning из пакета scikit-learn [16] для python. Данный алгоритм в отличие от K-SVD позволяет получать словарь D, в котором количество базисных векторов (атомов) больше размерности исходного пространства профилей n. Также будут представлены результаты для версии данного алгоритма (Mini-Batch DL), в котором оптимизация осуществляется не по всем данным, а по некоторому набору подмножеств. Данный подход дает менее точное решение, но при этом процесс расчета существенно ускоряется.

5. АВТОКОДИРОВЩИК

Следующий представленный в работе алгоритм относится ко второму типу алгоритмов, т.е. когда процедура сжатия и восстановления информации представляет собой некоторое нелинейное преобразование с неизвестными коэффициентами. В качестве такого нелинейного преобразования будет использоваться нейронная сеть со специальной структурой, а именно автокодировщик. Неизвестные весовые коэффициенты будут строиться по обучающей выборке методом обратного распространения ошибки. Нейронная сеть будет сжимать данные в пространство меньшей размерности. Выбирая количество нейронов на скрытом слое, задается размерность сжатого пространства, в которое перейдут профили ВРСЗ. Таким образом, закодированные координаты пространства x_i сжатых параметров соответствуют следующей функции:

$$x_j = relu(\sum_{i=1}^{n} y_i W^e_{i,j} + b^e_i), \quad relu(x) = \max(0, x),$$

где W^e — веса нейронной сети, соответствующие кодировщику (процесс сжатия). В качестве функции активации используется функция relu. Она позволяет получать разреженную структуру для элементов x_j, что в свою очередь уменьшает размер хранимой информации (см. табл. 1, 2). Восстановленный профиль из сжатого пространства будет иметь следующий вид:

$$\hat{y}_i = \sum_{i=1}^{k} x_i W^d_{i,j} + b^d_i.$$

Весовые матрицы находятся из решения оптимизационной задачи:

$$(W_e, W_d, b_e, b_d) \in \arg\min \sum_{y\in\mathcal{B}} \|\hat{y} - y\|^2,$$

где $\mathcal{B}$ множество минибатчей (подмножество профилей из $\mathcal{Y}^l$) по которому обучается нейронная сеть на каждом шаге. В качестве оптимизационной процедуры нейронной сети использовался алгоритм Adam [13] (адаптивная оценка момента), который является известной модификацией алгоритма стохастического градиентного спуска.

Для построения нейронных сетей использовался пакет tensorflow [17]. Обучающая выборка перед обучением нейронной сети была нормализована с помощью StandardScalar (центрирование и нормирование с помощью СКО) из пакета python [16].

6. ОБРАБОТКА ЭКСПЕРИМЕНТАЛЬНЫХ ДАННЫХ

В качестве экспериментальных данных использовались профили ВРСЗ для Баренцева моря из базы данных (БД) АКИН. Рассматривались профили с фиксированной глубиной равной 300 метров. Эта глубина была выбрана в силу того, что количество профилей с ней оказалось наибольшим в БД.

6.1. ТОЧНОСТЬ ВОССТАНОВЛЕНИЯ БД ВРСЗ

Количество профилей в БД было $m = 700$. Число горизонтов $n = 12$. Для алгоритмов сжатия K-SVD и DL предварительно над каждым профилем было произведено два вида усреднения: $Y = Y - \overline{Y}^v$ и $Y = Y - \overline{Y}^h$, где $Y \in \mathbb{R}^{n\times m}$ — матрица состоящая из m профилей размерности n. Матрица Y^v состоит из усредненного по всем профилям вектора

$$\overline{y}^v = \frac{1}{m}\sum_{i=1}^{m} y_i, \quad \overline{Y}^v = [\overline{y}^v, \dots, \overline{y}^v],$$

а матрица Y^h состоит из строк усредненных по глубинам $\overline{y}^h = \frac{1}{n}\sum_{i=1}^{n} y_i$. На рис. 2 изображено RMSE (1) полученная для всех сжатых, а затем восстановленных профилей (когда $\mathcal{Y} = \mathcal{Y}^l = \mathcal{Y}^t$) для разной размерности. На левом рисунке указана зависимость ошибки (RMSE) от размера пространства сжатых параметров. На правом рисунке изображена ошибка (RMSE) в зависимости от глубины при фиксированном размере пространства сжатых параметров.

На рис. 3 изображены те же характеристики, что и на рис. 2, но для оценки точности использовался скользящий контроль на наборе из пяти пар подмножеств. Для этого разбиения использовался алгоритм

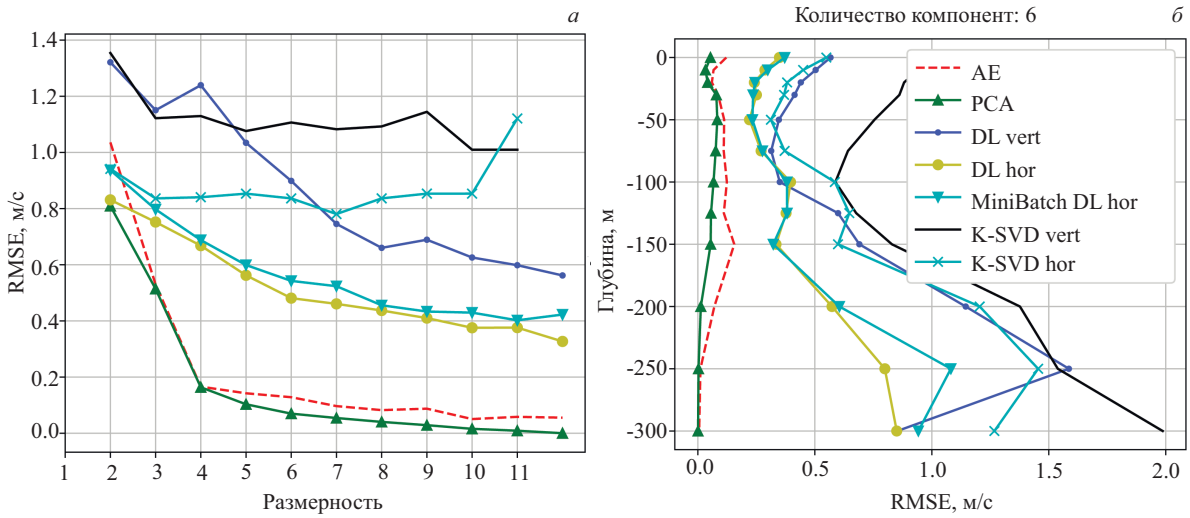

Рис. 2: Точность восстановления на всей БД ВРСЗ

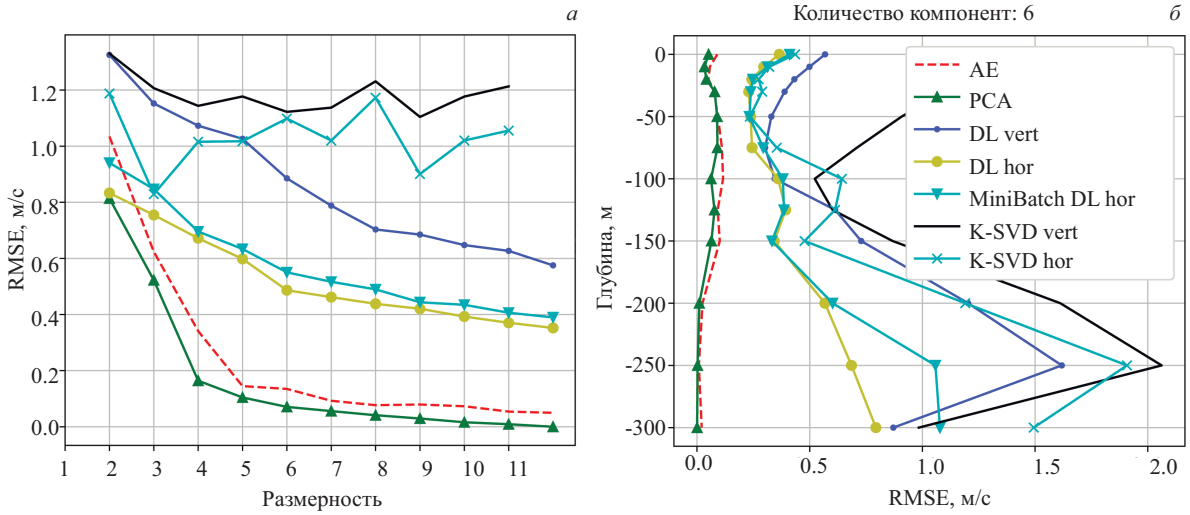

Рис. 3: Точность восстановления при кросс-валидации

K-Fold из пакета scikit-learn [16]. На обучающем множестве производилось обучение модели снижения размерности (поиск словаря или параметров), а на тестовом множестве проверялась точность восстановления для полученных параметров модели сжатия информации (AE, PCA, DL, K-SVD).

В таблицах 1, 2 для каждого алгоритма сжатия представлено количество ненулевых элементов матрицы X, которое требуется для хранения информации для восстановления профилей в зависимости от размерности пространства сжатых параметров (см. Comp.num).

6.2. ЗАДАЧА КЛАССИФИКАЦИИ

Далее протестируем представленные алгоритмы снижения размерности для задачи предсказания месяца которому соответствует профиль. Для этого множество профилей разбивается на обучающее и тестовое множества как в главе 2, но добавляется еще информация о сезоне, когда производились измерения для каждого профиля ВРСЗ. На обучающем множестве производится обучение классификатора для следующих двух случаев: используются реальные координаты ($y \in \mathbb{R}^m$); обучение происходит в сжатом пространстве для рассмотренных ранее алгоритмов ($x \in \mathbb{R}^k$). Проверка качества классификации проводилась на одном тестовом множестве. На рис. 4 представлены результаты, полученные после классификации. В качестве оценки качества классификации выступает доля правильных ответов. Для классификации использовалось три алгоритма: логистическая регрессия (из пакета scikit-learn), градиентный бустинг XGBoost [18], случайный лес (из пакета scikit-learn).

Таблица 1: Количество ненулевых элементов в матрице X после сжатия всех профилей

Comp. num.	2	3	4	5	6	7	8	9	10	11
AE	1397	2095	2793	2798	2799	3298	3493	3618	4897	4431
PCA	1400	2100	2800	3500	4200	4900	5600	6300	7000	7700
DL, K-SVD	700	700	700	700	700	700	700	700	700	700

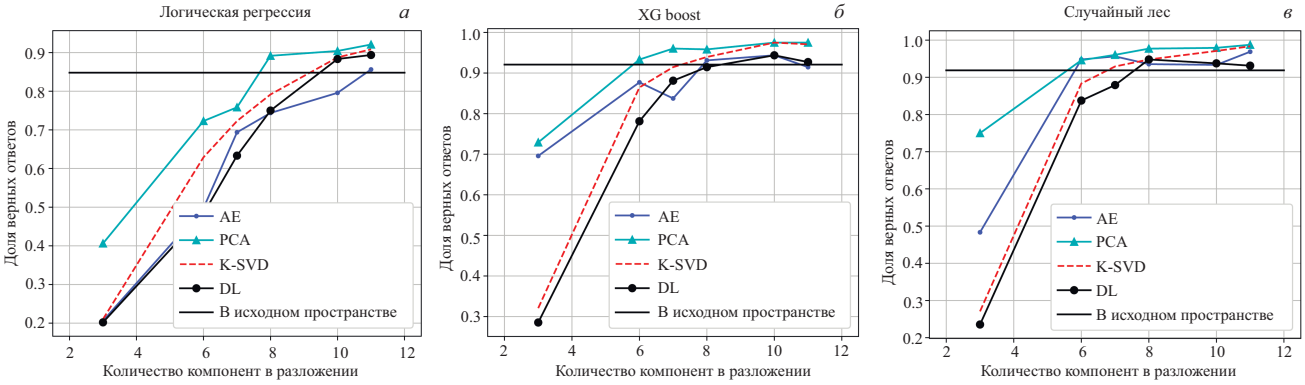

Рис. 4: Доля верных ответов для различных алгоритмов классификации

Таблица 2: Количество ненулевых элементов в матрице X после сжатия всех тестовых профилей с помощью обученной модели

Comp.num.	2	3	4	5	6	7	8	9	10	11
AE	280	280	556	559	559	784	698	701	931	1121
PCA	280	420	560	700	840	980	1120	1260	1400	1540
DL, K-SVD	140	140	140	140	140	140	140	140	140	140

ЗАКЛЮЧЕНИЕ

Из приведенных рис. 2, 3 следует, что самая высокая точность восстановления достигается с помощью метода главных компонент (PCA). На втором месте идет автокодировщик (AE). Далее идут методы DictionaryLearning (DL) из пакета scikit-learN [16].

Методы AE, DL и K-SVD позволяют сжимать профили за счет разреженной структуры матрицы коэффициентов (см. табл. 1, 2). Разреженность существенно меньше в методе AE чем в алгоритмах DL и K-SVD, но при этом точность выше. Алгоритм K-SVD дает аналогичное DL представление по степени разреженности, но в нем имеется ограничение на размер пространства сжатия $k \leq n$, а также этот алгоритм сильно проигрывает по точности DL и даже его приближенной версии MiniBatchDL. Также стоит отметить, что вычитание из данных среднего по глубине Y^h (DL hor, K-SVD hor) дает заметное улучшение в качестве восстановления исходной информации для DL и K-SVD.

Из рис. 4 с результатами классификации видно, что начиная с некоторой размерности пространства сжатых параметров их использование дает существенное улучшение в качестве классификации. Это позволяет сделать вывод, что использование рассмотренных алгоритмов снижения размерности при адекватном уровне понижения размерности дает большую возможность алгоритмам классификации извлекать информацию из данных для прогнозирования сезона, по сравнению с использованием стандартного представления профилей ВРСЗ.

[1] *Huang C. F., Gerstoft P., Hodgkiss W.S.* J. Acoust. Soc. Am. 2008. **123**, N 6. P. 162.
[2] *Gerstoft P.* J. Acoust. Soc. Am. 1994 **95**, N 2. P. 770.
[3] *Leblanc L. R., Middleton F. H.* J. Acoust. Soc. Am. 1980. **67**, N 6. P. 2055.
[4] *Захаров В. О.* ВМСППС. 2019. С. 150.
[5] *Tipping M. E., Bishop M. C.* Neural Computatin MIT Press 1999. **11**, N 2. P. 443.
[6] *Rubinstein R., Zibulevsky M., Elad M.* Eficient implementation of the K-SVD algorithmusing batch orthogonal matching pursuit. Computer Science Department, Technion Israel Institute of Technology, technical report. Haifa 32000 Israrel, 2008.
[7] *Mairal J., Bach F., Ponce J., Sapiro G.* Proceedings of

the 26 th International Conference on Machine Learning. Montreal. Canada. 2009.
[8] *Bianco M., Gerstft P.* J. Acoust. Soc. Am. March. 2017. **141**, N 3. P. 1749.
[9] *Mallat S. G., Zhang Z.* IEEE Transactions on Signal Processing. 1993. **12**, P. 3397.
[10] *Davis G., Mallat S., Avellaneda M.* Constructive Approximation. 1997. **13**, N 1. P. 57.
[11] *Chen S., Donoho D., Saunders M.* SIAM Review. 2001. **43**, N 1. P. 129.
[12] *Gorodnitsky I., Rao B.* IEEE Trans. on Signal Processing. 1997. **45**, N 3. P. 600.
[13] *Kingma D. P., Ba J.* Adam: A Method for Stochastic Optimization. ICLR. 2015.
[14] github.com/nel215/ksvd
[15] scikit-learn.org/stable/modules/decomposition.html
[16] scikit-learn.org
[17] tensorflow.org
[18] xgboost.readthedocs.io

Application of machine learning methods for sound speed profiles optimal storage

M. V. Lebedev[1,a], **V. O. Zaharov**[2,b]

[1]*Andreyev Acoustics Institute, Moscow 119991, Russia*
[2]*Moscow Aviation Institute (National Research University)*
E-mail: [a]*max_lebedev@mail.ru,* [b]*aladin7806@yandex.ru*

This paper was presented researches for dimension reduction of sound speed profiles (SSP). In addition, we shown optimal information extraction and optimal compression for the SSP profiles.

For the dimension reduction of SSP we used the principal component analysis, K-SVD, autoencoder and DictionaryLearning. For the information extraction was used following machine learning methods: boosting, logistic regression, random forest. These researches was conducted for the Barencevo sea SSP base.

PACS: 43.30.Xm, 43.30.Vh, 43.30.Wi
Keywords: machine learning, dimension reduction, sound speed profile, principal component analysis.
Received 14 November 2019.

Сведения об авторах

1. Лебедев Максим Витальевич — канд. физ.-мат. наук, ст. науч. сотрудник; тел.: (903) 005-85-75, e-mail: max_lebedev@mail.ru.
2. Захаров Вадим Олегович — тел.: (968) 692-95-00, e-mail: aladin7806@yandex.ru.