



Московский государственный университет имени М. В. Ломоносова

Факультет вычислительной математики и кибернетики

Кафедра математических методов прогнозирования

Коваленко Павел Антонович

Использование нейροкомпьютерного интерфейса для управления роботизированной рукой

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

Научный руководитель:

к.ф-м.н., доцент

Гуров Сергей Исаевич

Москва, 2017

Содержание

1	Введение	2
2	Обзор предметной области	2
2.1	Обзор используемого оборудования	4
2.2	Формальная постановка задачи	5
3	Описание экспериментов	6
3.1	Запись обучающей выборки	7
3.2	Предложенные способы обработки сигнала	8
3.3	Использованные алгоритмы машинного обучения	13
4	Вычислительные эксперименты	15
4.1	Результаты экспериментов	19
4.2	Управление роботизированной рукой	20
5	Заключение	20
	Список литературы	22

1 Введение

Нейрокомпьютерные интерфейсы (англ. Brain-Computer Interfaces, BCI) — это семейство устройств, созданных для передачи информации от мозга непосредственно в компьютер, без использования привычных средств ввода, таких как мышь или клавиатура. В последние два десятилетия эти устройства вышли за пределы медицинских лабораторий и стали доступны для работы рядовым пользователям.

Основной задачей данной работы является создание программного комплекса для управления механическим манипулятором (роботизированной рукой) при помощи ментальных команд, считываемых нейрокомпьютерным интерфейсом. Распознавание команд основано на классификации при помощи градиентного бустинга над решающими деревьями. Был испытан ряд способов обработки сигнала, описанных в разделе 3.2. В качестве управляющих сигналов были использованы некоторые мимические движения как самые легко воспроизводимые и простые в распознавании.

Полученные в работе результаты приведены в разделах 4.1 и 4.2. Там же приведена ссылка на видео с примером использования разработанного программного комплекса. Максимальная полученная точность классификации — 87%.

По результатам проделанной работы был сделан доклад на международной научной конференции студентов, аспирантов и молодых учёных «Ломоносов-2017».

2 Обзор предметной области

Первые исследования области нейрокомпьютерных интерфейсов начались в 1970-х годах с экспериментов над животными. Их устройство было достаточно простым: датчики вживлялись непосредственно в мозг животного и измеряли активность нейронов. Такой подход к получению информации из мозга называется инвазивным. Он позволил лучше понять принципы работы мозга и отдельных его частей. В частности, в 1999 году команде исследователей из Университета Калифорнии, Беркли, под руководством Ян Дань удалось вживить 117 датчиков в таламус¹

¹Таламус — область головного мозга, отвечающая за перераспределение информации, поступающей от органов чувств, к другим отделам головного мозга

кошки, что позволило восстановить изображение, которое видит кошка, по данным с этих датчиков [5].

В 2000 году Мигель Николесис из Университета Дьюка, США, вживил около 100 датчиков в мозг макаки-резуса, чтобы макака смогла управлять механической рукой [16]. Однако сама макака не видела результата своих действий. Этот эксперимент в 2006 году был дополнен группой исследователей под руководством Эндрю Шварца из Университета Питтсбурга. Теперь механическая рука находилась прямо перед макакой, и макака видела результат своих действий. Таким образом макака смогла сама себя кормить фруктами, управляя механической рукой при помощи ментальных сигналов [18].

В последние два десятилетия нейрокомпьютерные интерфейсы стремительно развивались. Исследователи научились восстанавливать зрение слепым людям, подключая камеру напрямую к мозгу, а также заменять утраченные конечности механическими протезами [12, 15]. Однако у инвазивного подхода есть несколько явных недостатков. Эксперименты такого рода требуют операции на мозге, а это, во-первых, является сложной и рискованной операцией, а во-вторых, требует больших затрат и высокого профессионализма исследователей. Такие ограничения сильно уменьшают круг людей, готовых работать с нейрокомпьютерными интерфейсами. Поэтому стали стремительно развиваться неинвазивные нейрокомпьютерные интерфейсы. Принцип работы большей части из них основан на считывании электроэнцефалограммы (ЭЭГ, англ. Electroencephalogram, EEG) — потенциала с разных точек на поверхности головы. К голове прикрепляется несколько датчиков, среди которых выделяется базовый (референт), сигнал которого принимается за нулевой уровень потенциала. Из показаний остальных датчиков вычитается показание референта, чтобы уменьшить влияние внешних полей. Пример медицинского электроэнцефалографа представлен на рис. 1. Существует общепринятая система размещения датчиков на голове, называемая “Система 10-10” (рис. 2).

Достоинства этого метода очевидны: использование такого устройства требует минимальных настроек, электроэнцефалографы могут быть портативными и простыми в использовании. С 2000 по 2010 год появился ряд коммерческих энцефалографов, доступных для использования не только исследовательским лабораториям,



Рис. 1: Медицинский электроэнцефалограф

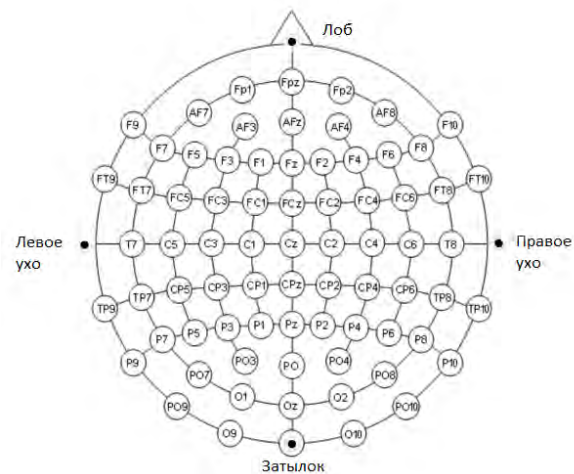


Рис. 2: Система 10-10

но и рядовым пользователям. Однако в большинстве случаев электроэнцефалографы дают менее точный и более зашумленный сигнал, чем инвазивные нейрокомпьютерные интерфейсы. Кроме того, в силу особенностей строения черепа разных людей, положения электродов могут отклоняться от стандартной схемы.

2.1 Обзор используемого оборудования

Emotiv EPOC

Emotiv EPOC (рис. 3) является коммерческим электроэнцефалографом, выпущенным компанией Emotiv Systems в 2009 году. Его сравнительно невысокая стоимость и простота и удобство в использовании делают его частым объектом различного рода исследований. Он имеет 14 сенсоров — в системе 10–10 это сенсоры AF3, AF4, F7, F8, F3, F4, FC5, FC6, T7, T8, P7, P8, O1 и O2, а также базовые сенсоры P3 и P4 (рис. 4). Его внутренняя частота дискретизации — 2048 Гц, однако встроенное оборудование по подавлению шума понижает частоту до 128 Гц. Одним из значимых достоинств данного устройства является большой пакет программ, идущих с ним в комплекте и позволяющих использовать многие из его возможностей, не написав ни строчки кода. В частности, в комплекте идет программа, распознающая одно из четырех эмоциональных состояний — радость, скука, усталость и медитация, а



Рис. 3: Emotiv EPOC

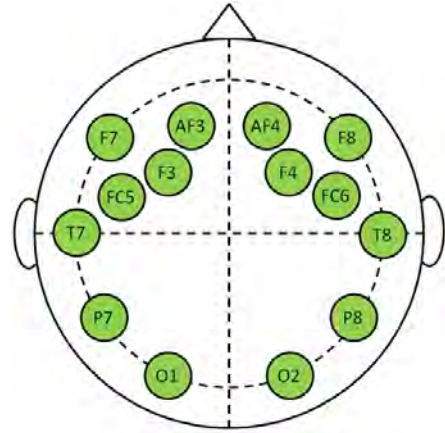


Рис. 4: Датчики Emotiv EPOC в системе 10-10

также программа, распознающая некоторую мимику лица. Также в комплекте идет утилита, способная научиться распознавать конкретные ментальные команды по показаниям ЭЭГ и привязать их к определенным действиям на компьютере, к примеру, движению мыши или нажатию определенных клавиш на клавиатуре.

Katana

Katana Robotic Arm — механический манипулятор, выпущенный компанией Neuronics AG в 2008 году (представлен на рис. 5). Katana является механической рукой с шестью степенями свободы — три пространственных, наклон гриппера (захвата), поворот гриппера и состояние гриппера (“схватить” — “отпустить”). В комплекте с роботом идут библиотеки для работы с ним для операционных систем Windows и Linux и языков C++ и Python.

2.2 Формальная постановка задачи

Рассматривается задача классификации. Дано пространство объектов X и конечное множество меток Y , а также целевая функция $y^* : X \rightarrow Y$, значение которой известно только на конечной выборке объектов $X^N = \{x_1, x_2, \dots, x_N\}$. Требуется построить функцию (алгоритм) $a : X \rightarrow Y$, предсказывающую ответы для любых объектов из X .



Рис. 5: Katana Robotic Arm

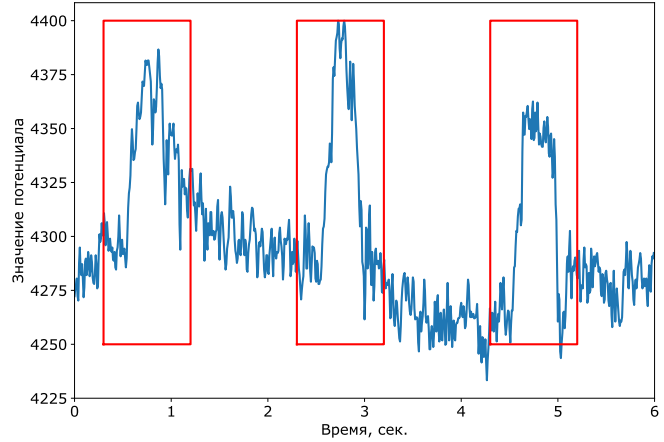


Рис. 6: Запись 6 секунд сигнала с датчика F8. Прямоугольниками выделены моргания

Признаком объекта называется отображение $f : X \rightarrow D_f$, где D_f — множество всех возможных значений признака. Если заданы несколько признаков f_1, f_2, \dots, f_k , то вектор $\mathbf{x} = (f_1(x), f_2(x), \dots, f_k(x))$ называется признаковым описанием x , а множество $D = D_{f_1} \times D_{f_2} \times \dots \times D_{f_k}$ — признаковым пространством.

Для корректной постановки задачи необходимо ввести функционал качества. Пусть для выборки известны истинные ответы на ней и ответы, полученные алгоритмом a . Тогда качеством алгоритма назовем долю правильно предсказанных ответов

$$accuracy(X^N) = \frac{1}{N} \sum_{i=1}^N [y^*(x_i) = a(x_i)],$$

где $[условие] = 1$, если условие верно, и 0 в противном случае. Поскольку большинство алгоритмов машинного обучения настраиваются на получение наиболее точных ответов на обучающей выборке, качество нужно измерять на отложенной выборке, которая не подавалась алгоритму для обучения.

3 Описание экспериментов

При помощи Emotiv Eroc, описанного в 2.1, можно снимать ЭЭГ с 14 датчиков с частотой дискретизации 128 Гц. Пример записи 6 сек. ЭЭГ с электрода F8 представлен на рис. 6. Требуется найти удобные управляющие сигналы и научиться выделять

их из записи ЭЭГ с достаточно высокой точностью. Тогда в дальнейшем эти сигналы можно будет использовать для управления механическим устройством.

В данной работе роботизированная рука перемещается в двумерной плоскости, то есть задействует две степени свободы. Для этого достаточно следующих четырех команд: “*вперед*”, “*назад*”, “*влево*” и “*вправо*”. Предлагается привязать к командам следующие управляющие сигналы:

- *Вперед* — поднять брови вверх;
- *Назад* — опустить брови вниз (нахмуриться);
- *Влево* — напрячь левый уголок губ;
- *Вправо* — напрячь правый уголок губ.

Данные сигналы были выбраны, поскольку они легко воспроизводимы и эксперименты показали, что их можно выделить с достаточно высокой точностью. Исследование по выбору наиболее удобных для распознавания сигналов было проведено в ходе курсовой работы.

В итоге задача распознавания сводится к классификации сигналов на пять классов: “*вперед*”, “*назад*”, “*влево*”, “*вправо*” и *нейтральный*, не требующий никаких действий. Далее требуется выбрать удобное признаковое описание данных и алгоритм распознавания.

3.1 Запись обучающей выборки

В данной задаче, как и в большинстве задач машинного обучения, главной проблемой, стоящей перед исследователем, является сбор и подготовка данных.

Как указано выше, было рассмотрено 5 классов управляющих сигналов. Для классов “*вперед*”, “*назад*”, “*влево*” и “*вправо*” было записано 10 сессий (выборок) по 8 сек., то есть всего по 80 сек. (около 10000 записей) на класс. В течение этих 8 секунд требовалось непрерывно выполнять одно и то же действие — напрягать брови или уголки губ соответственно. Для *нейтрального* класса было записано 10 сессий (выборок) по 30 сек. каждая. В течение их записи требовалось находиться в расслабленном состоянии и минимально использовать мышцы лица.

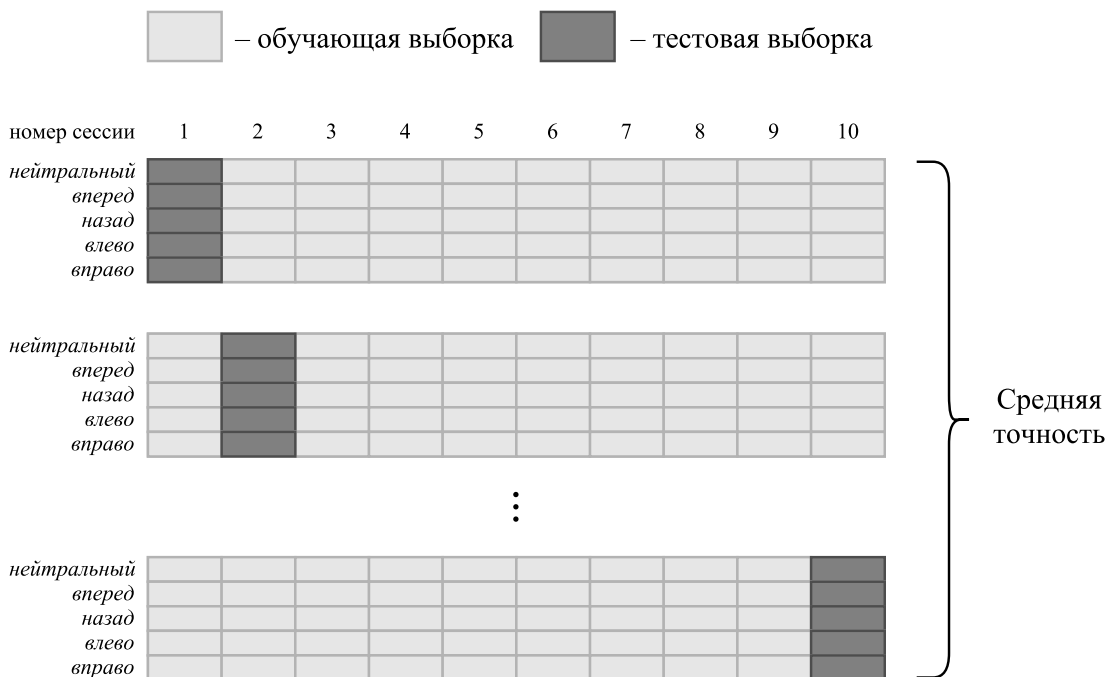


Рис. 7: Иллюстрация алгоритма кросс-валидации

В следующих разделах качество алгоритмов измеряется следующим образом:

1. Сначала все выборки группируются по номеру сессии, получаем 10 групп, в каждой из которых содержится по одной выборке для каждого класса.
2. Далее в цикле по номеру группы k от 1 до 10 алгоритм обучается на всех группах, кроме k -й, и вычисляется его точность на группе k .
3. Итоговая точность алгоритма получается усреднением точности алгоритма на каждой группе.

Иллюстрация данного алгоритма представлена на рис. 7.

3.2 Предложенные способы обработки сигнала

В данной работе в качестве единицы распознавания используется запись показаний 14 сенсоров продолжительностью L секунд. Длительность одной записи L подбиралась по кросс-валидации отдельно для каждого метода обработки сигнала. Из исходных данных выборка формировалась следующим образом: вся выборка разбивалась на группы длины L с шагом $L/2$, так что соседние группы перекрываются

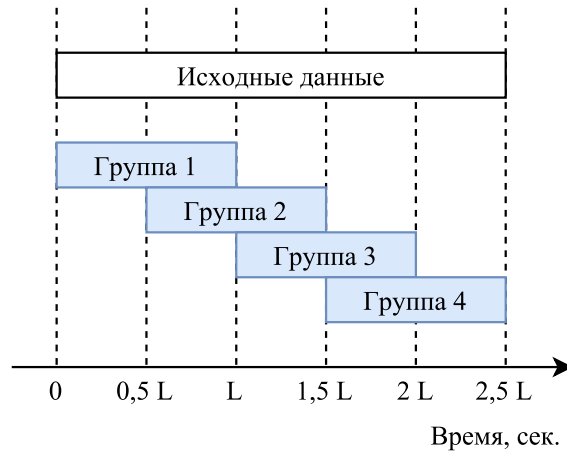


Рис. 8: Иллюстрация разделения данных на группы

на $L/2$ секунд, а группы, находящиеся через одну, не пересекаются. Иллюстрация данного алгоритма разбиения выборки представлена на рис. 8. Данный метод был выбран по следующим причинам. С одной стороны, если взять меньший шаг между соседними группами, то выборка получится слишком зависимой и повышается риск переобучения. С другой стороны, если соседние записи не будут пересекаться, придется записать в два раза больше “сырых” данных, чтобы получить тот же размер выборки.

Для удобства дальнейших обозначений введем ширину окна $W = L \cdot 128$ — количество записей в одной группе. Здесь 128 — частота дискретизации Emotiv EPOC, то есть количество записей в секунду.

Предложенные ниже преобразования были применены отдельно к каждой группе. Внутри группы преобразования проводились независимо по каждому каналу. Далее обозначим x_1, \dots, x_N — последовательные значения некоторого признака внутри группы — единица данных, подвергающихся обработке, y_1, \dots, y_M — их новое (преобразованное) представление. Все предложенные ниже преобразования будут описаны в этих терминах.

Данным образом каждый объект обучающей выборки представляется в виде матрицы вещественных чисел размера $M \times 14$. Для дальнейшей работы эта матрица преобразуется в вещественный вектор длины $14M$.

Исходные данные

В данном случае исходные (записанные с сенсоров) данные не были дополнительно обработаны или отфильтрованы. С данным методом ниже будут сравниваться другие методы обработки данных, чтобы определить, насколько обработка позволяет повысить точность алгоритма.

Дискретное преобразование Фурье

Преобразование Фурье является традиционной техникой при работе с временными рядами. Оно позволяет перейти от временного представления сигнала к его амплитудному представлению. В данном случае сигнал преобразуется в набор комплексных амплитуд X_k , $k = 0, \dots, N - 1$, каждая из которых является характеристикой исходного сигнала на частоте $\frac{k}{N}$. Преобразование выполняется по следующей формуле:

$$X_k = \frac{1}{N} \sum_{n=1}^N x_n e^{-\frac{2\pi i}{N} k(n-1)},$$

где i — мнимая единица.

Далее было рассмотрено два способа вещественного представления этих комплексных чисел.

1. Положим $M = 2N$ и $y_k = |X_{n-1}|$, $k = 1, \dots, N$ — амплитуда сигнала на соответствующей частоте, $y_{k+N} = \arg(X_{n-1})$, $k = 1, \dots, N$ — аргумент комплексной амплитуды, фаза сигнала на соответствующей частоте.

2. Положим $M = 2N$, $y_k = \operatorname{Re}(X_{n-1})$, $k = 1, \dots, N$ — вещественная часть амплитуд, $y_{k+N} = \operatorname{Im}(X_{n-1})$, $k = 1, \dots, N$ — мнимая часть амплитуд.

Дискретное вейвлет-преобразование

Дискретное вейвлет-преобразование задается парой дискретных функций:

$g = (g_0, \dots, g_K)$ — вейвлет-функция,

$h = (h_0, \dots, h_K)$ — функция растяжения.

Существуют семейства часто используемых функций (вейвлетов), среди которых были испытаны вейвлеты Добеши, биортогональные вейвлеты, симмлеты и Койфле-

ты. Подробное описание дискретного вейвлет-преобразования и распространенных классов вейвлетов, а также формулы для функций g и h можно найти в [7].

Добавим фиктивные значения ряда $x_{2-i} = x_i$, $x_{N+i} = x_{N-i}$, $i = 1, \dots, K/2$. Формально мы получили, что значения ряда теперь симметрично отражаются относительно его первого и последнего элементов. В предположении, что N и K четные, преобразование задается следующими формулами:

$$a_k = \sum_{i=-K/2}^{K-K/2} h_{i+K/2} x_{2k+i}, \quad k = 1, 3, 5, \dots, N-1$$

— детализирующие коэффициенты

$$c_k = \sum_{i=-K/2}^{K-K/2} g_{i+K/2} x_{2k+i}, \quad k = 1, 3, 5, \dots, N-1$$

— коэффициенты аппроксимации

Далее возьмем $M = N$ и $y_k = a_{2k-1}$, $k = 1, \dots, N/2$, $y_{k+N/2} = c_{2k-1}$, $k = 1, \dots, N/2$.

Цифровой фильтр Баттерворта

Цифровые фильтры — семейство методов обработки дискретного сигнала с целью выделения или подавления определенных частотных составляющих сигнала. Выделяют фильтры высоких частот (оставляющие в сигнале высокие частоты и подавляющие низкие), фильтры низких частот, полосовые фильтры (оставляющие частоты в определенной полосе и подавляющие все остальные) и полосно-заграждающие фильтры (подавляющие частоты в заданной полосе).

Распространенным классом цифровых фильтров являются линейные цифровые фильтры, задаваемые своим обобщенным разностным уравнением:

$$y_k = \sum_{i \in I} b_i x_{k-i} + \sum_{j \in J} a_j y_{k-j},$$

где $I, J \subset \mathbb{Z}$ — интервалы целочисленной прямой, содержащие 0.

Если $a_0 = 0$, то фильтр называется нерекурсивным, иначе рекурсивным.

Если $I = \{0, 1, \dots, n\}$, $J = \{0, 1, \dots, m\}$, то фильтр называется однопроходным и применить его можно за один проход по исходным данным. В этом случае порядком фильтра называется $\max\{n, m\}$.

Для $i \leq 0$ обычно полагают $x_i = 0$.

Фильтры Баттерворта являются семейством нерекурсивных однопроходных фильтров, отличительной особенностью которых является гладкость их амплитудно-частотной характеристики в частотах, где сигнал пропускается (не подавляется). Подробнее о цифровом фильтре Баттерворта можно прочитать в [6]. В данной работе был использован цифровой фильтр Баттерворта 5-го порядка в варианте полосового фильтра и фильтра высоких частот. Соответственно, $M = N$, y_1, \dots, y_N — отфильтрованное представление x_1, \dots, x_N .

Дифференцированный сигнал

Дифференцирование сигнала — распространенная техника обработки временных рядов, позволяющая избавиться от низкочастотных колебаний. При данном входном сигнале x_1, \dots, x_N дифференцированный сигнал вычисляется по формуле $y_k^{(1)} = x_{k+1} - x_k$, $k = 1, \dots, N - 1$. Производные более высоких порядков вводятся рекуррентно: $y_k^{(s)} = y_{k+1}^{(s-1)} - y_k^{(s-1)}$.

В данной работе были использованы производные первого и второго порядка. Для производной первого порядка: $M = N - 1$, $y_k = y_k^{(1)} = x_{k+1} - x_k$, $k = 1, \dots, N - 1$. Для производной второго порядка: $M = N - 2$, $y_k = y_k^{(2)} = x_{k+2} - 2x_{k+1} + x_k$, $k = 1, \dots, N - 2$.

Фильтры Савицкого-Голея

Идея данного семейства фильтров заключается в полиномиальном сглаживании сигнала, то есть избавлении от высокочастотного шума. Фильтр задается нечетной шириной окна m — числом соседних значений, используемых для сглаживания, и порядком полинома p . Тогда сглаженные значения вычисляются по следующей формуле:

$$y_k = \sum_{i=-\frac{m-1}{2}}^{\frac{m-1}{2}} C_i x_{k+i},$$

где коэффициенты C_i однозначно задаются шириной окна и порядком полинома. Как и в случае дискретного вейвлет-преобразования, вводятся фиктивные переменные, симметрично отражающие данные относительно первого и последнего элементов, чтобы алгоритм корректно работал на границах ряда.

Для удаления низкочастотного шума перед применением фильтра Савицкого-Голея можно продифференцировать сигнал описанным выше способом. Фактически дифференцирование сводится к модификации коэффициентов C_i . Подробнее о данном семействе фильтров можно прочитать в статье [17].

В введенных ранее обозначениях $M = N$, формулы для y_k выписаны выше, $k = 1, \dots, N$.

3.3 Используемые алгоритмы машинного обучения

В работе был испытан ряд алгоритмов машинного обучения (SVM, логистическая регрессия, дискриминантный анализ...), среди них два — Random Forest и градиентный бустинг над решающими деревьями — показали значительно более высокую точность, чем остальные, поэтому далее будут использоваться только эти два алгоритма. Ниже идет краткое описание принципов их работы в контексте задачи многоклассовой классификации.

В основе обоих алгоритмов лежат решающие деревья, поэтому сначала рассмотрим алгоритм построения решающих деревьев.

Решающие деревья

Решающее дерево — алгоритм машинного обучения, используемый как для задачи классификации (предсказания ответа из конечного множества меток), так и для задачи регрессии (предсказания вещественного значения).

Решающее дерево строится рекурсивно. На одном шаге построения имеется множество объектов R , заданное своим признаковым описанием. Требуется разбить его на два множества A и B условием вида “ $x \in A$, если $x_i < threshold$, иначе $x \in B$ ”. При этом номер признака i и порог $threshold$ нужно выбрать таким образом, чтобы разброс значений значений среди объектов внутри каждого из множеств A и B был минимален. Далее каждое из множеств A и B рекурсивно разбивается по тому же алгоритму.

Построение дерева начинается с разбиения всего множества объектов обучающей выборки X^N по описанному выше алгоритму. Заканчивается построение дерева,

если на очередной итерации в разбиваемом множестве осталось слишком мало объектов (меньше заранее заданного значения), разброс ответов для объектов в этом множестве достаточно мал, достигнута максимальная глубина дерева или выполнено другое заранее заданное условие останова. В таком случае для этого множества записывается ответ, равный среднему (в случае регрессии) или моде (в случае классификации) ответов для объектов из этого множества.

Полученный набор условий разбивает все признаковое пространство на набор непересекающихся многомерных параллелепипедов, для каждого из которых записан константный ответ. Теперь чтобы предсказать ответ для нового объекта, нужно посмотреть, в какой из параллелепипедов он попадает, и вернуть ответ для этого параллелепипеда. Подробнее об алгоритме можно прочитать в [3].

Random Forest

Random Forest является ансамблевым алгоритмом, построенным на идее бэггинга. Классификатор представляет собой множество отдельных решающих деревьев. Каждое из деревьев строится по выборке того же размера, что и исходная выборка, полученной из исходной путем случайного выбора с возвращением (бутстрепа). Чтобы предсказать ответ для нового объекта, вычисляются предсказания для него каждого из решающих деревьев, и ответом считается мода ответов решающих деревьев. Подробнее об алгоритме можно прочитать в [3, 4].

Градиентный бустинг

Градиентный бустинг обычно применяется для задачи регрессии. Его идея заключается в представлении ответа как линейной комбинации ответов базовых алгоритмов (регрессоров), каждый из которых исправляет ошибки предыдущих. В данном случае в качестве базовых алгоритмов использовались решающие деревья.

Каждый базовый алгоритм обучается на выборке того же размера, что и исходная, построенной из исходной бутстрепом, то есть случайным выбором с возвращением. Также иногда базовый алгоритм обучается не на всех признаках, а на их случайном подмножестве. Построение композиции ведется итеративно.

Пусть y — правильные ответы на обучающей выборке, $\hat{y}^{(k)}$ — ответ, полученный линейной комбинацией ответов первых k базовых регрессоров, $L(y, \hat{y})$ — функция потерь. Тогда $(k + 1)$ -й базовый алгоритм обучается так, чтобы его ответы были как можно ближе к $-\frac{\partial}{\partial \hat{y}} L(y, \hat{y})|_{\hat{y}=\hat{y}^{(k)}}$. Пусть $z^{(k+1)}$ — ответы $(k + 1)$ -го алгоритма на обучающей выборке. Далее ищется

$$\alpha^* = \arg \min_{\alpha} L(y, \hat{y}^{(k)} + \alpha z^{(k+1)})$$

и полагается $\hat{y}^{(k+1)} = \hat{y}^{(k)} + \alpha^* z^{(k+1)}$. На практике обычно делают шаг не на $\alpha z^{(k)}$, а на $\eta \alpha z^{(k)}$, где $\eta \in [0, 1]$ — скорость обучения (learning rate). Данный алгоритм по сути является минимизацией функции потерь градиентным спуском в пространстве ответов на обучающей выборке с определенными ограничениями, задаваемыми структурой базовых алгоритмов.

В случае классификации на K классов фактически решается K связанных задач регрессии. Строится K ансамблей базовых алгоритмов, k -й из которых предсказывает вероятность отнести объект к классу k . В качестве функции потерь выступает кросс-энтропия:

$$L(z, y) = - \sum_{i=1}^N \sum_{k=1}^K z_k(x_i) \log p_k(x_i),$$

где $z_k(x) = 1$, если объект x принадлежит классу k , и 0 иначе,

$$p_k(x) = \frac{1}{1 + e^{-\hat{y}_k(x)}}$$

— предсказанная вероятность отнести объект x к k -му классу, $\hat{y}_k(x)$ — ответ k -го ансамбля, внешняя сумма ведется по объектам обучающей выборки. Подробнее об алгоритме можно прочитать в [3, 9].

4 Вычислительные эксперименты

В данном разделе представлена точность алгоритмов на предложенных данных при каждом из описанных выше способов обработки сигналов. Решается задача классификации на 5 классов, при этом около 45% записей относятся к нейтральному классу. Параметры алгоритмов, ширина окна W и другие параметры подбирались по кросс-валидации, если не указано иное.

Исходный сигнал

Оптимальная ширина окна $W = 10$.

Алгоритм	Точность	Оптимальные параметры
Random Forest	0.582	Число деревьев: 100
Градиентный бустинг	0.618	Число деревьев: 100, скорость обучения: 0.02

Таблица 1: Точность алгоритмов на не обработанных данных

Дискретное преобразование Фурье. Амплитудное и фазовое представление

Оптимальная ширина окна $W = 30$.

Алгоритм	Точность	Оптимальные параметры
Random Forest	0.840	Число деревьев: 100
Градиентный бустинг	0.865	Число деревьев: 100, скорость обучения: 0.3

Таблица 2: Точность алгоритмов после применения преобразования Фурье в виде амплитуды и фазы

Дискретное преобразование Фурье. Вещественное и мнимое представление

Оптимальная ширина окна $W = 30$.

Алгоритм	Точность	Оптимальные параметры
Random Forest	0.808	Число деревьев: 100
Градиентный бустинг	0.857	Число деревьев: 100, скорость обучения: 0.3

Таблица 3: Точность алгоритмов после применения преобразования Фурье в виде вещественной и мнимой части

Дискретное вейвлет-преобразование

Наибольшей точности удалось добиться с вейвлетом Добеши db5.

Оптимальная ширина окна $W = 20$.

Алгоритм	Точность	Оптимальные параметры
Random Forest	0.750	Число деревьев: 100
Градиентный бустинг	0.861	Число деревьев: 100, скорость обучения: 0.3

Таблица 4: Точность алгоритмов после применения вейвлет-преобразования

Полосовой частотный фильтр

Исследования показывают, что интересующие нас волны, испускаемые мозгом, лежат в диапазоне от 1 до 30 Гц [13]. Поэтому был использован фильтр частот между 1 и 30 Гц.

Оптимальная ширина окна $W = 10$.

Алгоритм	Точность	Оптимальные параметры
Random Forest	0.633	Число деревьев: 100
Градиентный бустинг	0.673	Число деревьев: 100, скорость обучения: 0.3

Таблица 5: Точность алгоритмов после применения полосового фильтра частот

Фильтр высоких частот

Эксперименты показали, что подавление высоких частот понижает точность, поэтому в данном эксперименте был использован только фильтр частот выше 1 Гц.

Оптимальная ширина окна $W = 10$.

Алгоритм	Точность	Оптимальные параметры
Random Forest	0.678	Число деревьев: 100
Градиентный бустинг	0.741	Число деревьев: 100, скорость обучения: 0.3

Таблица 6: Точность алгоритмов после применения фильтра высоких частот

Однократное дифференцирование

Оптимальная ширина окна $W = 10$.

Алгоритм	Точность	Оптимальные параметры
Random Forest	0.816	Число деревьев: 100
Градиентный бустинг	0.853	Число деревьев: 100, скорость обучения: 0.3

Таблица 7: Точность алгоритмов на дифференцированных данных

Двукратное дифференцирование

Оптимальная ширина окна $W = 10$.

Использование производных более высокого порядка не повысило точность распознавания.

Алгоритм	Точность	Оптимальные параметры
Random Forest	0.859	Число деревьев: 100
Градиентный бустинг	0.868	Число деревьев: 100, скорость обучения: 0.3

Таблица 8: Точность алгоритмов на дважды дифференцированных данных

Фильтр Савицкого-Голея

Лучшей точности удалось добиться при использовании однократного дифференцирования и сглаживания полиномом порядка $p = 5$ с шириной окна сглаживания $m = 9$.

Оптимальная ширина окна $W = 10$.

Алгоритм	Точность	Оптимальные параметры
Random Forest	0.750	Число деревьев: 100
Градиентный бустинг	0.762	Число деревьев: 100, скорость обучения: 0.2

Таблица 9: Точность алгоритмов после применения фильтра Савицкого-Голея

4.1 Результаты экспериментов

	Random Forest	Градиентный бустинг
Исходные данные	0.582	0.618
Преобразование Фурье. Амплитуда и фаза	0.840	0.865
Преобразование Фурье. Вещественная и мнимая часть	0.808	0.857
Вейвлет-преобразование	0.750	0.861
Полосовой фильтр	0.633	0.673
Фильтр высоких частот	0.678	0.741
Однократное дифференцирование	0.816	0.853
Двукратное дифференцирование	0.859	0.868
Фильтр Савицкого-Голея	0.750	0.762

Таблица 10: Точность алгоритмов при разных способах обработки данных

Как видно из результатов, выбор информативного признакового описания данных позволяет повысить точность на 25%.

Однако при применении фильтрации сигнала (как частотной, так и фильтра Савицкого-Голея) пропадает часть информации и точность оказывается ниже, чем для не отфильтрованных, но преобразованных данных.

Одновременно три метода обработки сигнала показали близкий результат, наилучшей точности удалось добиться, используя двукратное дифференцирование.

На рис. 9 представлена матрица ошибок лучшего классификатора — градиентного бустинга, обученного на дважды дифференцированных данных. Судя по матрице, все пять классов распознаются с достаточно высокой точностью, ни один класс заметно не пересекается с другим. Это позволяет использовать данный классификатор для управления роботизированной рукой.

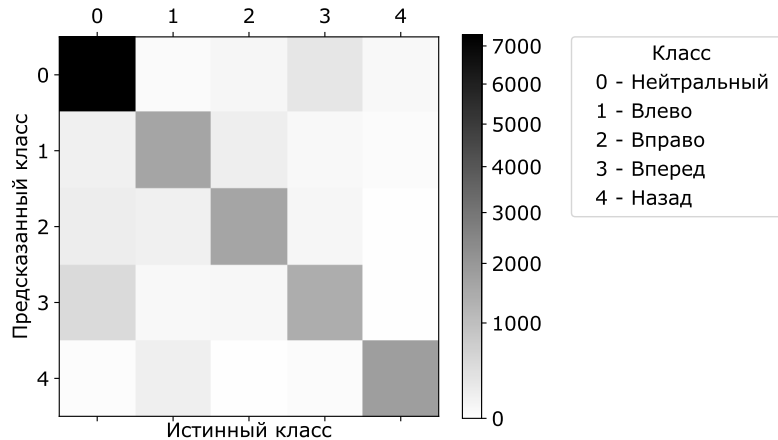


Рис. 9: Матрица ошибок лучшего алгоритма классификации

4.2 Управление роботизированной рукой

Была написана программа на языке Python 2.7 для считывания показаний с Emotiv EPOC, их распознавания в реальном времени, выделения управляющих команд и их передачи на роботизированную руку Katana. В качестве управляющих команд были использованы пять команд, описанных в разделе 3.

Команды распознаются с достаточно высокой точностью (около 87%), доля ложных срабатываний сравнительно невелика. Однако для использования в реальных условиях эта небольшая доля ложных срабатываний критична, поэтому систему нужно развивать дальше. Видео с примером управления роботизированной рукой можно найти в [2].

5 Заключение

В данной работе было проведено исследование по распознаванию некоторых мимических движений на показаниях, полученных с электроэнцефалографа. Был рассмотрен ряд способов обработки и фильтрации сигнала, для них были подобраны параметры, позволяющие добиться наиболее высокой точности. Наилучший результат показал подход с двойным дифференцированием входного сигнала.

Для решения задачи распознавания команд был рассмотрен ряд алгоритмов машинного обучения, среди них наибольшей точности удалось добиться с использо-

ванием градиентного бустинга. Лучшая точность, полученная в работе, — 87% на несбалансированной пятиклассовой классификации.

Результатом работы стало создание программы, связывающей нейроинтерфейс Emotiv EPOC с роботизированной рукой Katana. Предложенная система команд позволяет перемещать роботизированную руку в плоскости, при этом достигается достаточно высокая точность, что позволяет говорить о решении поставленной задачи.

В ходе выполнения выпускной квалификационной работы:

1. Исследован ряд способов обработки данных, заданных в виде временного ряда;
2. Рассмотрены некоторые алгоритмы машинного обучения применительно к задаче распознавания команд на электроэнцефалограмме;
3. Создан программный комплекс по управлению роботизированной рукой при помощи нейрокомпьютерного интерфейса;
4. По результатам работы сделан доклад на международной научной конференции студентов, аспирантов и молодых учёных «Ломоносов-2017» [1].

Список литературы

1. *Коваленко П.* Управление механическим манипулятором посредством нейрокомпьютерного интерфейса // Сборник тезисов XXIV Международной научной конференции студентов, аспирантов и молодых ученых «Ломоносов-2017». — Макс-Пресс, 2017.
2. Пример использования разработанной системы. — <https://yadi.sk/i/5swHHkXh3HrF6w>.
3. *Bishop C. M.* Pattern Recognition and Machine Learning. — Springer, 2006.
4. *Breiman L.* Random forests // *Machine Learning*. — 2001. — Vol. 45.
5. *Dan Y. et al.* Reconstruction of natural scenes from ensemble responses in the lateral geniculate nucleus // *Journal of Neuroscience*. — 1999. — Vol. 19.
6. *Daniels R. W.* Approximation Methods for Electronic Filter Design. — New York: McGraw-Hill, 1974.
7. *Daubechies I.* Ten Lectures on Wavelets. — SIAM, 1992.
8. *Ekanayake H.* P300 and Emotiv EPOC: Does Emotiv EPOC capture real EEG?: Tech. rep.: 2010. — <http://neurofeedback.visaduma.info/emotivresearch.htm>.
9. *Friedman J. H.* Stochastic gradient boosting: Tech. rep.: Department of Statistics, Stanford University, 1999.
10. *Harrison T.* The Emotiv mind: Investigating the accuracy of the Emotiv EPOC in identifying emotions and its use in an Intelligent Tutoring System: Tech. rep.: Department of Computer Science and Software Engineering, University of Canterbury, 2013.
11. *Heger D., Putze F., Schultz T.* Online recognition of facial actions for natural EEG-based BCI applications: Tech. rep.: Cognitive Systems Lab, Karlsruhe Institute of Technology, Germany, 2011.

12. *Kennedy P., Bakay R.* Restoration of neural output from a paralyzed patient by a direct brain connection // *NeuroReport*. — 1998. — Vol. 9.
13. *Kuzovkin I.* Pattern recognition for non-invasive EEG-based BCI: Tech. rep.: University of Tartu, Faculty of Mathematics and Computer Science, Institute of Computer Science, 2011.
14. *Maskeliunas R. et al.* Consumer-grade EEG devices: are they usable for control tasks? // *PeerJ*. — 2016. — Vol. 4.
15. *Naumann J.* Search for Paradise: A Patient's Account of the Artificial Vision Experiment. — Xlibris Corporation, 2012.
16. *Nicolelis M. et al.* Real-time prediction of hand trajectory by ensembles of cortical neurons in primates // *Nature*. — 2000. — Vol. 408.
17. *Savitzky A., Golay M. J. E.* Smoothing and differentiation of data by simplified least squares procedures // *Analytical Chemistry*. — 1964. — Vol. 36.
18. *Schwartz A. et al.* Cortical control of a prosthetic arm for self-feeding // *Nature*. — 2008. — Vol. 453.