

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ
ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение
высшего образования

УРАЛЬСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ
имени первого Президента России Б.Н. Ельцина

ИНСТИТУТ ЕСТЕСТВЕННЫХ НАУК И МАТЕМАТИКИ

Департамент математики, механики и компьютерных наук

**ИСПОЛЬЗОВАНИЕ ОКУЛОГРАФИИ ДЛЯ ОЦЕНКИ
ДОСТОВЕРНОСТИ СООБЩАЕМОЙ ИНФОРМАЦИИ**

Направление подготовки 02.03.01 «Математика и компьютерные науки»

Допустить к защите:

Директор департамента:
к.ф.-м.н, доцент М.О. Асанов

Выпускная квалификационная работа
бакалавра

**Румянцева
Андрея Николаевича**

Научный руководитель:
Ассистент кафедры
высокопроизводительных компьютерных
технологий ИМКН УрФУ Блинов Павел
Анатольевич

Екатеринбург
2017

РЕФЕРАТ

Данная работа посвящена проверке гипотезы о наличии связи между движением и размером зрачка испытуемого и достоверности сообщаемой им в момент наблюдения информации.

В ходе работы был проведен эксперимент, в ходе которого были получены данные для исследования. На основе этих данных был построен ряд классификаторов, показавших следующее качество:

Качество классификатора, основанного на логистической регрессии с l2 регуляризацией: 0.870198690199 при значении с 10

Качество классификатора, основанного на логистической регрессии с l1 регуляризацией: 0.905458333333 при значении с 1

Качество классификатора, основанного на градиентном бустинге: 0.75038003663

Качество классификатора, основанного на использовании случайного леса: 0.772410353535

Качество классификатора на основе нейронной сети: 0.840473970474

Качество классификатора на основе наивного байесовского метода: 0.762193362193

Качество классификатора на основе стохастического градиентного спуска: 0.817426989677

Качество классификатора на основе метода опорных векторов: 0.840834276834

Многие модели показали высокое качество. Наилучшим качеством в данной задаче обладает классификатор, основанный на логистической регрессии с l1-регуляризацией. Его качество 0.905458333333.

Также были получены данные о влиянии характеристик движения и размера зрачка на целевую переменную. Целевой переменной является 0 в случае,

если испытуемый говорил правду и 1 в противном случае.

Веса признаков в классификаторе, основанном на логистической регрессии с l_1 регуляризацией:

Среднее положение по оси X 0.884254925874

Выборочная дисперсия по оси X 0.285071119207

Среднее положение по оси Y -1.4537381405

Выборочная дисперсия по оси Y 0.197491967092

Средний диаметр 1.02458772173

Выборочная дисперсия диаметра 1.54995051721

Средняя скорость перемещения зрачка -0.0104773458606

Чем больше вес признака по модулю, тем большее влияние он имеет.

В работе была предпринята попытка решить задачу кластеризации. То есть требовалось по признаковым описаниям объектов, характеризующих движение и размер зрачка испытуемого во время ответа, разбить эти объекта на два класса, один из которых соответствует ответам, в ходе которых испытуемый говорил правду, а другой соответствует ответам, в ходе которых испытуемый лгал.

В ходе решения задачи кластеризации были обучены модели, показавшие следующее качество:

Качество модели, основанной на методе k ближайших соседей:
0.459909655562

Качество модели, основанной на агломеративной кластеризации:
0.486666666667

Качество модели, основанной на спектральной кластеризации:
0.543295502312

Все модели в задаче кластеризации показали качество, близкое к простому угадыванию, что является низким показателем.

СОДЕРЖАНИЕ

ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ.....	5
ВВЕДЕНИЕ.....	6
ОСНОВНАЯ ЧАСТЬ.....	8
1 Обзор литературы	8
2 Постановка задачи работы	11
3 Методика эксперимента	12
4 Результаты и их обсуждение	15
4.1 Извлечение признаков	15
4.2 Подход как к задаче классификации.....	17
4.3 Подход как к задаче кластеризации	22
ЗАКЛЮЧЕНИЕ.....	25
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ И ЛИТЕРАТУРЫ.....	27
ПРИЛОЖЕНИЯ	29
ПРИЛОЖЕНИЕ А. Схема сборки окулографа	29

ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ

В работе приняты следующие обозначения:

- число в квадратных скобках обозначает ссылку на список использованных источников и литературы. Само число указывает порядковый номер ссылки в списке.
- число, написанное надстрочным шрифтом, обозначает ссылку на комментарий внизу страницы. Само число указывает порядковый номер комментария.

ВВЕДЕНИЕ

Окулография (eye tracking) – техника, основанная на наблюдении за движением зрачка или глазного яблока испытуемого. Первые исследования с использованием данной технологии осуществлялись исключительно методом зрительного наблюдения за объектом исследования. Так в 1879 году французский исследователь Луи Эмиль Жаваль в ходе своих исследований[1] с применением окулографии заметил, что глазные яблоки движутся не монотонно, а скачками. Такие скачки называются саккадами, а периоды, когда глазные яблоки неподвижны – фиксации.

Первое механическое устройство для отслеживания движения глаз было создано в 1908 году психологом Эдмундом Хью[2]. Представляло оно из себя линзу с отверстием для зрачка. Более совершенное устройство, основанное на фотографировании, разработали в 1901 году Р. Додж и Т. Клайн[3]. Впоследствии устройства, основанные на фото- и видеосъемке получили наиболее широкое применение.

Пожалуй первой фундаментальной работой в области окулографии стала книга советского ученого Альфреда Ярбуса «Роль движений глаз в процессе зрения», опубликованная в 1965 году[4]. В этой работе описаны анатомические особенности движения глаз, методы окулографии, особенности реакции зрачка на экстремально яркие объекты, быстро меняющиеся цвета, неподвижные и мелькающие объекты. Так же автором было сделано несколько фундаментальных выводов о взаимосвязи между мотивацией испытуемого и фиксациями его взгляда.

Важная гипотеза в области окулографии была сформулирована П. Карпентером и М. Джастом в 1980 году[5]. «Не существует значительного расхождения между тем, на чем фиксирует свой взгляд человек и тем, что он пытается осмыслить». Из этого следует, что когнитивный процесс относительно объекта сопоставим по времени с периодом фиксации взгляда на этом объекте. Именно на эту гипотезу ссылаются многие исследователи в

работах, связанных с психологией и исследованием когнитивного процесса. На данный момент наиболее актуальной работой в области методологии окулографии является работа А. Духовски “Методология окулографии: теория и практика”, опубликованная в 2007 году [6].

Сейчас окулография получила широкое применение. Современные технологии позволяют фиксировать движение зрачка и глазного яблока с достаточной точностью и частотой. Окулография (eye tracking) применяется для абсолютно различных целей, начиная от медицинских исследований и заканчивая изучением взаимодействия человека и машины, а также маркетинговых исследований. Данную технологию можно применять для анализа утомляемости водителя тепловоза[7], исследования больных аутизмом[8] или для усовершенствования юзабилити web ресурсов[9].

В работе рассматривается применимость окулографии (eye tracking) для оценки достоверности сообщенной информации. В области психологии высказывались гипотезы о том, что ложь имеет влияние на движение глаз. Подобные предположения делал П. Карпетнер в 1977 году в своей работе, посвященной движению глаз[10]. Также большое внимание данному вопросу было уделено в работе Милтона Эриксона “Стратегия психотерапии” 2002 года [11]. Данная работа посвящена проверке гипотезы применимости техники отслеживания взгляда для оценки сообщаемой человеком информации.

ОСНОВНАЯ ЧАСТЬ

1 Обзор литературы

На данный момент существует множество работ, посвященных окулографии (eye tracking)[4][7][12], реакции организма и в частности зрачков на ложь[5][10][11][13], методам проверки испытуемого на детекторе лжи[14][15]. Рассмотрим некоторые из этих работ.

Работа Джозефа Вонга “Расширение зрачка и окулография” 2011 года [12] описывает основы современной окулографии (eye tracking), основанной на видеосъемке. Приведены характеристики современных окулографов (eye tracker), которые позволяют фиксировать положение глаза с частотой около 2000 Гц, то есть каждые 0.5 мс, и имеют низкий процент ошибок (менее 0.5%). Полученные от окулографа данные, а именно координаты и диаметр зрачка, используются для получения других характеристик движения глаза. Например скорость, среднее положение, средний диаметр, скорость изменения диаметра, саккады, фиксации.

Также в работе кратко рассматривается реакция зрачка на эмоциональные стимулы, приведены результаты исследований на данную тему. Так, например, приведены время расширения зрачка, когда испытуемый испытывает страх (от 0.33мс до 1.25мс), средний диаметры зрачка при отрицательном и положительном эмоциональном воздействии (0,14мм и 0,2мм соответственно).

В данной работе приведено множество практических примеров использования окулографии. Описана система, которая с точностью 81% определяет предпочтения заказчика при просмотре товаров на экране [16].

В работе Гарри Бонда “Экспертиза на детекторе лжи” 2008 года [14] подробно рассматривается процедура проведения экспертизы на детекторе лжи с использованием окулографии. Описывается два эксперимент. В ходе

первого эксперимента группа людей проходила экспертизу на детекторе лжи, а в ходе второго испытуемым задавали вопросы во время процедуры окулографии.

Приведены результаты обоих экспериментов и соответствия между ними. Автор работы пришел к выводу, что между движением глаз и правдивостью сообщаемой испытуемым информацией существует связь. Также были сделаны выводы о влиянии информированности испытуемого о том, что за ним наблюдают на результаты исследования, а именно - движения глаз испытуемого становятся менее выраженными.

Гипотезу о том, что движения зрачков связаны с достоверностью сообщаемой испытуемым информацией проверяли в работе “Использование окулографии и расширения зрачка для определения достоверности информации в экономической игре” Джозеф Вонг, Михаэль Специо и Колин Крамер в 2006 году [13].

Эксперимент проводился в форме экономической игры. Приведем краткое ее описание. Первый игрок в начале раунда получает информацию о секретном числе S из отрезка $[1, 5]$ и сдвиге b из отрезка $[0, 2]$. Второй игрок получает информацию только о сдвиге. После этого первый игрок сообщает первому число M из отрезка $[1, 5]$ в формате “Секретное число в этой игре M ”. Второй игрок на основе сообщения первого игрока должен сделать предположение о секретном числе A . Итоговый выигрыш первого игрока определяется как $110 - 10 \cdot |S - A + b|$, а второго как $110 - 10 \cdot |S - A|$. Таким образом первому игроку выгодно, чтобы второй ошибся ровно на b .

После сбора необходимых данных исследователи проанализировали их и вывели математическую модель, которая на основании движения глазного яблока и размера зрачка определяет ложь с вероятностью 80%, а размер лжи ($|S - M|$) с вероятностью 66%. Средний выигрыш второго игрока при использовании данной модели составил 93,4 и 86,2 при сдвигах 1 и 2 соответственно, против предсказываемыми теорией игр [17][18] 87,4 и 71,6,

что доказывает полезность построенной модели.

На основании описанных исследований есть основания полагать, что:

- Существует связь между эмоциональным состоянием испытуемого и размерами его зрачка.
- Существует связь между движением глаз испытуемого и достоверностью сообщаемой им в этот момент информации.
- Имеет отрицательное влияние информированность испытуемого о том, что он проходит процедуру окулографии.
- Имеет влияние на движение глаз и размер зрачка испытуемого не только факт наличия лжи, но и ее размер.

Исходя из этого можно сделать вывод о наличии связи между характеристиками глаза испытуемого и достоверностью сообщаемой им информации, а также о возможности составить модель для определения достоверности информации на основе окулографии.

2. Постановка задачи работы

В ходе работы требуется проверить гипотезу о наличии связи между движением и размером зрачка испытуемого и достоверности сообщаемой им в момент наблюдения информации.

Для этого необходимо поставить эксперимент, в ходе которого нужно собрать данные для исследования. По полученным данным сформировать выборку и построить классификатор, который по признаковому описанию объекта, характеризующего движение и размер зрачка испытуемого во время ответа, классифицирует ответ испытуемого как правду или ложь.

В качестве дополнительной задачи попробовать решить задачу кластеризации. То есть требуется по признаковым описаниям объектов, характеризующих движение и размер зрачка испытуемого во время ответа, разбить эти объекты на два класса, один из которых соответствует ответам, в ходе которых испытуемый говорил правду, а другой соответствует ответам, в ходе которых испытуемый лгал. В данной задаче нет этапа обучения модели, требуется разбить объекты на классы в предположении, что у нас нет обучающей выборки.

3. Методика эксперимента

Для проведения эксперимента был собран и использован окулограф (eye tracker) на основе камеры [здесь будет название камеры].

Для сборки окулографа были использованы следующие детали:

[здесь будет список деталей окулографа]

Схема сборки представлена в приложении А.

Данный окулограф удовлетворяет требованиям, предъявляемым современным окулографам, так как камера фиксирует положение зрачка с частотой [здесь частота камеры], что больше, чем частота саккадических движений глаз, которая составляет 20-150 Гц [6].

Окулограф совместим с открытым программным обеспечением для окулографии (eye tracking) pupil labs¹, которое используется в эксперименте для фиксации положения зрачка. Данное программное обеспечение было выбрано по причине его кроссплатформенности и открытого программного кода. Pupil labs также обладает достаточной частотой фиксации положения зрачка - 120 Гц. Выбранное программное обеспечение предоставляет необходимую для исследования информацию, а именно положение зрачка и его диаметр. Задержка камеры в среднем составляет 5.7 миллисекунд, точность определения угла поворота зрачка - 0.6 градуса.

В ходе эксперимента испытуемому предлагалась игра, которая протекает по следующему сценарию:

- В начале раунда испытуемому выводится на экран число раунда от 0 до 9. Выпадение каждого числа раунда равновероятно
- Также на экран выводится действие, которое должен совершить

¹ <https://pupil-labs.com>

испытуемый: сказать правду или солгать. Данные действия равновероятны

- Испытуемый называет число раунда интервьюеру. При этом он либо лжет либо говорит правду, в зависимости от действия раунда
- Интервьюер пытается угадать, правду ли сказал испытуемый
- Схема начисления очков за раунд следующая:

Вердикт интервьюера Действие раунда	Ложь	Правда
Ложь	0 / 25	30 / 0
Правда	5 / 0	5 / 25

где первое число - количество очков испытуемого, а второе - количество очков интервьюера.

Данная модель похожа на используемую в работе “Использование окулографии и расширения зрачка для определения достоверности информации в экономической игре” Джозефа Вонга, Михаэля Специо и Колина Крамера, однако в ней для простоты не используется понятие размера лжи.

Игра была проведена в 100 раундов.

Для проведения эксперимента был написан программный код игры на языке python¹.

Такая схема начисления очков за раунд была выбрана для того, чтобы в случае автоматизации игры интервьюером, который равновероятно выбирает правду сказал испытуемый или ложь, математическое ожидание количества очков интервьюера и испытуемого было одинаковым.

¹ <https://github.com/anssar/eye-tracking>

Так математическое ожидание очков испытуемого:

$$M(X) = 100 * (5 * 1 + 30 * 0.25) = 1250$$

А математическое ожидание очков интервьюера:

$$M(X) = 100 * (25 * 0.5) = 1250$$

Вероятности действий раунда, определяющих правду должен сказать испытуемый или солгать, были выбраны равновероятными, чтобы получить достаточные наборы данных для случаев, когда испытуемый говорит правду и когда лжет. В реальных экспериментах эти вероятности могут отличаться.

В ходе эксперимента был исследован один испытуемый. Возраст испытуемого - 21 год, пол мужской.

Таким образом, в ходе эксперимента было получено 100 наборов данных для единственного испытуемого, из которых могут быть извлечены признаки для дальнейшего анализа.

4. Результаты и их обсуждение

4.1. Извлечение признаков

В ходе эксперимента были получены закодированные программным обеспечением окулографа с помощью стандартной библиотеки `pickle` встроенной в `python` третьей версии данные о координатах зрачка в различные моменты времени, а также диаметр зрачка. Время на ответ испытуемого не было ограничено, а фильтрация интервала временных меток происходила посредством синхронизации временных меток, оставляемых программным обеспечением окулографа и меток, оставляемых программным обеспечением игры. Таким образом на каждый ответ приходится около 100 измерений, что приводит к большому числу зависимых друг от друга признаков.

Для дальнейшего изучения данные были проанализированы и из них были извлечены следующие признаки:

- Выборочная дисперсия положения зрачка по оси X. Число с плавающей точкой, нормализованное от 0 до 1.
- Самое правое положение зрачка. Число с плавающей точкой, нормализованное от 0 до 1.
- Среднее положение зрачка по оси X. Число с плавающей точкой, нормализованное от 0 до 1.
- Выборочная дисперсия положения зрачка по оси Y. Число с плавающей точкой, нормализованное от 0 до 1.
- Среднее положение зрачка по оси Y. Число с плавающей точкой, нормализованное от 0 до 1.
- Выборочная дисперсия диаметра зрачка. Число с плавающей точкой, измеряется в мм.
- Средний диаметр зрачка. Число с плавающей точкой, измеряется в мм.
- Средняя скорость перемещения зрачка. Число с плавающей точкой,

измеряется в мм/с.

- Правду или ложь сообщал испытуемый. Целевая категориальная переменная, имеет 2 возможных значения - 0 (правда) и 1 (ложь).

Дисперсия положений зрачка и диаметра была выбрана в качестве признака, так как она более устойчива к выбросам, чем другие характеристики разброса.

В ходе извлечения признаков были отобраны только те измерения, уверенность программного обеспечения в которых составляет более 80%, чтобы отсеять неточные измерения, которые могут послужить причиной выбросов. Это возможно благодаря тому, что программное обеспечение возвращает степень уверенности в точности своих измерений в виде числа с плавающей точкой от 0 до 1.

Извлеченные данные записываются в файл в формате csv с разделителем ';'.

Для извлечения признаков был написан программный код на языке python¹.

¹ <https://github.com/anssar/eye-tracking>

4.2. Подход как к задаче классификации

После первичного анализа сырых данных были получены признаковые описания объектов, характеризующих движение и размер зрачка в период, когда испытуемый говорил правду или ложь. Также нам известно значение целевой переменной, которая может принимать 2 значения: 0 - испытуемый говорил правду или 1 - испытуемый лгал. Наша задача - имея обучающую выборку, научиться по признаковому описанию объекта, характеризующего движение и размер зрачка во время ответа, определять правду или ложь говорил испытуемый. Таким образом мы имеем задачу бинарной классификации.

Для решения задачи классификации в работе используется библиотека `scikit-learn` для `python3` версии не ниже 0.18.1.

Перед тем, как обучать модели, данные были дополнительно нормализованы при помощи объекта `StandardScaler` библиотеки `scikit-learn`. Это сделано для того, чтобы признаки имели одинаковый масштаб, потому что некоторые алгоритмы машинного обучения чувствительны к масштабу признаков. Поскольку объект имеет только количественные признаки, дополнительных действий для подготовки данных не требуется.

Для оценки качества обученного классификатора будем использовать метрику `roc-auc`. Она характеризует площадь под `roc` кривой и подходит для оценки качества бинарной классификации. `Roc` кривая отображает соотношение между верно классифицированными несущими признак объектами и объектами, классифицированных, как несущие признак, но на самом деле не несущие его. Таким образом, в работе принято соглашение, что ошибки первого и второго рода равноценны. То есть ложноположительное и ложноотрицательное срабатывания трактуются, как ошибки одинаковой серьезности, однако на практике это может быть не так.

Также для измерения качества обученного классификатора используется скользящий контроль (cross validation). При помощи объекта KFold выборка разбивается на 5 равных частей. Алгоритм обучается 5 раз так, что 4 из 5 частей выборки, которые в сумме составляют 80% используются, как обучающая выборка, а оставшаяся часть используется для как тестовая выборка для оценки качества обученной модели. После это качество полученной модели для всех пяти испытаний усредняется. Данная техника позволяет контролировать, чтобы модель не получилась переобученной. Переобученная модель очень хорошо подстраивается под обучающую выборку, однако на тестовых данных работает намного хуже.

В работе обучаются следующие модели:

- Логистическая регрессия
- Стохастический градиентный спуск
- Метод опорных векторов
- Наивный Байесовский классификатор
- Градиентный бустинг
- Случайный лес
- Нейронная сеть

Для классификатора основанного на логистической регрессии используется объект LogisticRegression библиотеки scikit-learn. Основа классификатора - подгонка данных к логистической кривой. Данный классификатор использует гиперпараметр, отвечающий за величину штрафа во время обучения при отдалении от логистической кривой. Переберем в качестве значения этого параметра степени числа 10 от -3 до 3 и выберем модель с лучшим качеством. Для уменьшения вероятность переобучения в логистической регрессии может использоваться регуляризация. Есть 2 типа регуляризации: l1 и l2. При этом l1 регуляризация более жесткая и позволяет отсеивать признаки, которые имеют малое влияние на целевую переменную.

Для логистической регрессии с l2-регуляризацией было получено качество модели 0.882043956044 при значении гиперпараметра, отвечающего за величину штрафа 10.

Для логистической регрессии с l1-регуляризацией было получено качество модели 0.905458333333 при значении гиперпараметра, отвечающего за величину штрафа 1.

При этом в логистической регрессии с l1-регуляризацией были получены следующие веса признаков:

Среднее положение по оси X - 0.884254925874

Выборочная дисперсия по оси X - 0.285071119207

Среднее положение по оси Y - -1.4537381405

Выборочная дисперсия по оси Y - 0.197491967092

Средний диаметр - 1.02458772173

Выборочная дисперсия диаметра - 1.54995051721

Средняя скорость перемещения зрачка - -0.0104773458606

Чем больше по модулю вес признака, тем большее влияние на целевую переменную он имеет. Таким образом видно, что наибольшее влияние имеют выборочная дисперсия диаметра зрачка и среднее положение зрачка по оси Y, а выборочные дисперсии положения зрачка по обеим осям и скорость перемещения зрачка почти не имеют влияния. Средний диаметр зрачка и его среднее положение по оси X так же имеют некоторое влияние на целевую переменную.

Для классификатора основанного на стохастическом градиентном спуске используется объект SGDClassifier библиотеки scikit-learn. Метод основан на пошаговом изменении вектора весов в сторону антиградиента функционала ошибок. Данный классификатор также имеет ряд гиперпараметров, для него может быть использована регуляризация. В ходе работы была обучена модель с параметрами по умолчанию и l2 регуляризацией. Полученное качество обученного классификатора на основе метода стохастического

градиента - 0.817426989677.

Для классификатора основанного на методе опорных векторов используется объект SVC библиотеки scikit-learn. Метод основан на переходе из признакового пространства в пространство более высокой размерности и поиске в новом пространстве разделяющей гиперплоскости, обеспечивающей максимальный зазор между объектами разных классов. Данный классификатор имеет ряд гиперпараметров, в ходе работы была обучена модель с параметрами по умолчанию. Полученное качество обученного классификатора на основе метода опорных векторов - 0.840834276834.

Для классификатора основанного на наивном методе Байеса используется объект GaussianNB библиотеки scikit-learn. Метод основан на применении теоремы Байеса в предположении, что события независимы. Данный классификатор использует Гауссовское распределение и имеет ряд гиперпараметров. В ходе работы была обучена модель с параметрами по умолчанию. Полученное качество обученного классификатора основанного на наивном методе Байеса - 0.762193362193.

Для классификатора основанного на градиентном бустинге используется объект GradientBoostingClassifier библиотеки scikit-learn. Метод основан на минимизации функционала ошибки, который в свою очередь использует композицию ответов нескольких моделей, использующих градиентный спуск. Данный классификатор имеет ряд гиперпараметров, в том числе количество моделей, используемых для композиции. В ходе работы была обучена модель с использованием 100 моделей для композиции. Полученное качество обученного классификатора на основе градиентного бустинга - 0.75038003663.

Для классификатора основанного на случайном лесе используется объект

RandomForestClassifier библиотеки scikit-learn. Метод основан на композиции ответов решающих деревьев. Данный классификатор имеет ряд гиперпараметров, в том числе количество решающих деревьев. В ходе работы была обучена модель с использованием 100 решающих деревьев. Полученное качество обученного классификатора на основе случайного леса - 0.772410353535.

Для классификатора основанного на нейронной сети используется объект MLPClassifier библиотеки scikit-learn. Метод основан на обучении многослойной нейронной сети с использованием метода обратного распространения ошибки. Данный классификатор имеет ряд гиперпараметров, в ходе работы была обучена модель с параметрами по умолчанию. Полученное качество обученного классификатора на основе нейронной сети - 0.840473970474.

Поскольку наибольшее качество было получено для линейных моделей, можно предположить, что зависимость целевой переменной от признаков линейная. Метод основанный на наивном Байесе не дал хороших результатов, потому что признаки зависимы. Нейронная сеть дала хороший результат, однако можно высказать гипотезу, что при более тонкой ее настройке можно добиться лучших результатов.

Для решения задачи классификации был написан программный код на языке python¹.

¹ <https://github.com/anssar/eye-tracking>

4.3. Подход как к задаче кластеризации

Ранее в работе предполагалось, что у нас есть некая обучающая выборка для настройки модели. Однако в реальной жизни обучающей выборки у нас может и не быть. Нам нужно уметь разделять объекты на два класса по их признаковому описанию так, чтобы объекты одного кластера отвечали за ответы, во время которых испытуемый лгал, а второго за ответы, во время которых испытуемый говорил правду. Таким образом мы имеем задачу разделения объектов на два кластера, то есть задачу кластеризации.

Для решения задачи кластеризации в работе используется библиотека `scikit-learn` для `python3` версии не ниже 0.18.1.

Аналогично задаче классификации используется нормализация данных, скользящий контроль и метрика `roc-auc`.

В работе используются следующие модели:

- Кластеризация методом `k` ближайших соседей
- Агломеративная кластеризация
- Спектральная кластеризация

Для кластеризации основанной на методе `k` ближайших соседей используется объект `KMeans` библиотеки `scikit-learn`. Метод основан на присвоении объекту определенной метки кластера на основе кластерных меток `k` ближайших к нему объектов. Данный метод имеет ряд гиперпараметров, в частности количество кластеров. В ходе работы была обучена модель с параметрами по умолчанию и количеством кластеров равным двум. Один из кластеров отвечает за ответы, в ходе которых испытуемый говорил а правду, а другой за ответы, в ходе которых испытуемый говорил ложь. Полученное качество обученной модели на основе метода `k` ближайших соседей - 0.459909655562.

Для агломеративной кластеризации используется объект `AgglomerativeClustering` библиотеки `scikit-learn`. Метод основан на пошаговом объединении объектов, находящихся ближе всего в признаковом пространстве в один кластер. Данный метод имеет ряд гиперпараметров, в частности количество кластеров. В ходе работы была обучена модель с параметрами по умолчанию и количеством кластеров равным двум. Один из кластеров отвечает за ответы, в ходе которых испытуемый говорил а правду, а другой за ответы, в ходе которых испытуемый говорил ложь. Полученное качество обученной модели на основе агломеративной кластеризации - 0.486666666667.

Для спектральной кластеризации используется объект `SpectralClustering` библиотеки `scikit-learn`. Метод основан на использовании спектра матрицы сходства объектов для понижения размерности признакового пространства и последующего использования метода k ближайших соседей. Данный метод имеет ряд гиперпараметров, в частности количество кластеров. В ходе работы была обучена модель с параметрами по умолчанию и количеством кластеров равным двум. Один из кластеров отвечает за ответы, в ходе которых испытуемый говорил а правду, а другой за ответы, в ходе которых испытуемый говорил ложь. Полученное качество обученной модели на основе спектральной кластеризации - 0.543295502312.

Все модели дали качество, близкое к 0.5, что соответствует простому угадыванию. Если посмотреть присвоенные метки кластеров, то можно видеть, что первые несколько измерений (около 40, зависит от конкретной модели) были отнесены к первому кластеру, а оставшиеся измерения ко второму. Это может соответствовать тому, что движение и размер зрачка в начале эксперимента отличалось от движения и размера в конце из-за привыкания испытуемого к обстановке эксперимента. Вероятно некоторые

признаки, которые имеют малое значение в задаче классификации, здесь имеют большое влияние, что приводит к полученным результатам.

В общем случае решение задачи кластеризации неоднозначно и существенно зависит от выбранной метрики в признаковом пространстве. Можно высказать гипотезу, что при более тонкой настройке обучения моделей и выборе другой метрики задача кластеризации была бы решена более успешно, но это требует дополнительного исследования.

Для решения задачи классификации был написан программный код на языке python¹.

¹ <https://github.com/anssar/eye-tracking>

ЗАКЛЮЧЕНИЕ

В ходе работы были получены следующие содержательные результаты:

- По обучающей выборке было обучено несколько моделей, способных с определенным качеством классифицировать объекты, характеризующие движение и размер зрачка во время ответа испытуемого на два класса. Один из классов включает в себя объекты, соответствующие ответам, во время которых испытуемый говорил правду, а второй класс включает объекты, соответствующие ответам, во время которых испытуемый лгал. Наилучшее качество получается при использовании логистической регрессии с l-1 регуляризацией. Качество данной модели составляет 0.905458333333.
- В среднем модели, которые используют линейной разделение объектов, получили более высокие оценки качества. Соответственно можно сделать вывод, что объекты разделены линейно.
- Наибольшие веса для определения лжи имеют выборочная дисперсия диаметра зрачка и среднее положение зрачка по оси Y, а выборочные дисперсии положения зрачка по обеим осям и скорость перемещения зрачка почти не имеют влияния. Средний диаметр зрачка и его среднее положение по оси X так же имеют некоторое влияние на целевую переменную.
- При решении задачи кластеризации не удалось построить модель с качеством выше 0.543295502312. Это может быть связано с тем, что была выбрана неудачная метрика в признаковом пространстве.

В дальнейшем для развития данной работы необходимо проверить следующие гипотезы:

- Более тонкая настройка нейронной сети может дать классификатор с лучшим качеством, чем было показано логистической регрессией.
- Более тонкая настройка и подбор метрики моделей кластеризации может решить задачу кластеризации с качеством близким к решению задачи классификации.

- Использование более совершенного окулографа или 3d окулографа поможет построить более точные модели классификации и кластеризации.
- Необходимо выяснить, насколько индивидуальны полученные обученные модели. Для этого нужно провести исследования для многих испытуемых и построить более общую модель.
- Обстановка эксперимента и наличие окулографа влияет на испытуемого. Для проверки этой гипотезы необходимо использовать в качестве окулографа скрытую от испытуемого камеру, которая будет достаточно точно фиксировать положение зрачка.

Так же в дальнейшем необходимо ввести или разработать метрику оценки качества моделей, которая по разному штрафует за ошибки первого и второго рода. Размер штрафов зависит от конкретного эксперимента.

Содержательные результаты данной работы могут быть использованы в области прохождения процедуры допроса на детекторе лжи в совокупности с другими методами оценки достоверности сообщаемой информации в ходе прохождения процедуры.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ И ЛИТЕРАТУРЫ

1. Un opthalmomètre pratique, Annales d'oculistique, L. E. Javal, H. Schiötz: Paris, 1881
2. The Psychology and Pedagogy of Reading, Huey Edmund, MIT Press 1968 (originally published 1908)
3. The angle velocity of eye movements. Dodge, Raymond; Cline, Thomas Sparks Psychological Review, Mar 1901
4. Роль движений глаз в процессе зрения, А. Л. Ярбус, Наука, 1965
5. A theory of reading: From eye fixations to comprehension. Just, Marcel A.; Carpenter, Patricia A. Psychological Review, Jul 1980
6. Eye Tracking Methodology: Theory and Practice, Duchowski, A. T. 2007
7. Eye tracking in human-computer interaction and usability research: Ready to deliver the promises, RJ Jacob, KS Karn - Mind, 2003
8. Eye-tracking for detection of driver fatigue, M Eriksson - System, 1997
9. The application of eye-tracking technology in the study of autism, Z Boraston, SJ Blakemore - The Journal of physiology, 2007
10. Movements of the eyes, R.H.S. Carpenter.- London: Pion, 1977
11. Strategy of the Psychotherapy, Milton H. Erickson, 2002
12. Pupil Dilation and Eye-tracking, JT Wang - A handbook of process tracing methods, 2011
13. Pinocchio's Pupil: Using Eyetracking and Pupil Dilation To Understand Truth-telling and Deception in Games, JT Wang, M Spezio, CF Camerer - The American Economic, 2010
14. Deception Detection Expertise, GD Bond - Law and Human Behavior, 2008
15. Psychology and the lie detector industry. Lykken, David T. American Psychologist, Oct 1974
16. AutoSelect: What You Want Is What You Get: Real-Time Processing of Visual Attention and Affect, Bee, N., Prendinger, H., Nakasone, A., André, E., &

Ishizuka, M., Perception and Interactive Technologies, 2006

17. Non-cooperative Games, Nash, John Forbes, 1951

18. Two-person Cooperative Games, Nash, John Forbes, 1953

ПРИЛОЖЕНИЯ

ПРИЛОЖЕНИЕ А. Схема сборки окулографа

[здесь будет схема сборки окулографа]