**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

|  |  |
| --- | --- |
| **QY2IeRIhQNY.jpg** | **ИНСТИТУТ ИНТЕЛЛЕКТУАЛЬНЫХ КИБЕРНЕТИЧЕСКИХ СИСТЕМ**  **Кафедра**  **«Криптология и кибербезопасность»** |

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**ОТЧЕТ**

**о научно-исследовательской работе**

«Анализ методов машинного обучения, применимых к задаче аутентификации пользователей мобильных устройств»

|  |  |  |
| --- | --- | --- |
| Исполнитель:  студентка гр. С13-503 |  | Валацкайте Я.В. |
| Научный руководитель: | подпись, дата | Когос К.Г. |
| Зам. зав. каф. № 42:  к.т.н. | подпись, дата | Когос К.Г. |
|  | подпись, дата |  |

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Москва – 2017**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**РЕФЕРАТ**

Отчет 41 с., 6 рис., 2 табл., 16 ист.

МАШИННОЕ ОБУЧЕНИЕ, АУТЕНТИФИКАЦИЯ, МОБИЛЬНЫЕ УСТРОЙСТВА, ПОВЕДЕНЧЕСКАЯ БИОМЕТРИЯ, МЕТОД КЛАССИФИКАЦИИ, МЕТОД ОБНАРУЖЕНИЯ АНОМАЛИЙ.

Объектом исследования являются методы машинного обучения, применимые к задаче аутентификации на основе поведенческой биометрии.

Цель — оценить возможность применения методов машинного обучения для решения задачи аутентификации на основе поведенческой биометрии.

Для поиска подходящего к задаче аутентификации метода машинного обучения был произведен анализ существующих на данный момент методов.

В результате работы систематизированы методы машинного обучения, сделан вывод о возможности применения подобных методов для задачи аутентификации пользователя мобильного устройства на основе поведенческой биометрии.

СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 4](#_Toc472493865)

[1 Машинное обучение 5](#_Toc472493866)

[1.1 Решаемые задачи 5](#_Toc472493867)

[1.1.1 Обучение с учителем 6](#_Toc472493868)

[1.1.2. Обучение без учителя 8](#_Toc472493869)

[1.2 Признаковое пространство 8](#_Toc472493870)

[1.3 Переобучение и недообучение 9](#_Toc472493871)

[1.3.1 Процедура скользящего контроля 10](#_Toc472493872)

[1.3.2 L1- и L2-регуляризация в машинном обучении 14](#_Toc472493873)

[2 Метод обнаружения аномалий 16](#_Toc472493874)

[2.1 Метод, основанный на восстановлении плотности 18](#_Toc472493875)

[2.2 Сведение к методу классификации 21](#_Toc472493876)

[3 Применение метода обнаружения аномалий к задаче аутентификации пользователя смартфона 33](#_Toc472493877)

[3.1 Обзор проблемы 33](#_Toc472493878)

[3.2 Роль машинного обучения в задаче аутентификации 35](#_Toc472493879)

[ЗАКЛЮЧЕНИЕ 39](#_Toc472493880)

[СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ 40](#_Toc472493881)

# **ВВЕДЕНИЕ**

В настоящее время все больше конфиденциальной информации передается по телефону, все больше хранится на самом устройстве. Пользователи используют новейшие приложения, позволяющие оплачивать покупки on-line, коммунальные услуги, переводить деньги со счета на счет. Вместе с популярностью смартфона растет и количество мошенничеств, происходящих из-за слабой защиты мобильных устройств. Проблема аутентификации пользователя является актуальной.

Необходимы методы защиты данных мобильных устройств, удобство которых не будут уступать существующим. С задачей распознавания нелегитимного пользователя на основе данных о законном владельце может помочь машинное обучение.

В первой главе рассматриваются основные методы машинного обучения. Представлена систематизация по виду решаемых задач, а также рассматривается проблема переобучения и способы борьбы с ней.

Во второй главе более подробно рассматриваются методы обнаружения аномалий, такие как метод, основанный на восстановлении плотности, а также метод сведения к задаче классификации.

В третьей главе проведено исследование возможности применения методов машинного обучения к решению задачи аутентификации пользователя мобильного устройства.

# **1 Машинное обучение**

Машинное обучение — это процесс, в результате которого машина (компьютер) способна показывать поведение, которое в нее не было явно заложено (запрограммировано). Такое нестрогое определение машинного обучения дал Артур Самуэль в далеком 1959 году в своей инновационной работе по компьютерным шашкам [1].

Если говорить о строгом, формальном определении, то его хорошо дал Том Митчелл в своей книге о машинном обучении [2], ставшей классической: компьютерная программа обучается на основе опыта E по отношению к некоторому классу задач T и меры качества P, если качество решения задач из T, измеренное на основе P, улучшается с приобретением опыта E.

Таким образом, машинное обучение — это подраздел искусственного интеллекта, изучающий алгоритмы, способные к обобщению и обучению. Машинное обучение находится на стыке многих разделов множества наук. В задачах машинного обучения часто применяются методы теории вероятностей, линейной алгебры, статистики, оптимизации и многих других дисциплин.

## 1.1 Решаемые задачи

Машинное обучение классифицируется по типу решаемых задач [3] (рисунок 1.1).

Обучение с учителем, когда на подмножестве рассматриваемых объектов известны ответы для каждого из них. Учителем, обычно, называют обучающую выборку пар <объект – ответ>. Задачей такого обучение является нахождение закономерности, согласно которой можно найти ответ для любого объекта.

Частичное обучение, когда при частичном обучении большая часть ответов неизвестна.

Активное обучение очень похоже на обучение с учителем с той разницей, что ответы изначально неизвестны. Основная идея состоит в том, что алгоритм сам может обучаться на малых выборках, если он сам выбирает, какие данные ему нужны. То есть алгоритм составляет запросы, ответы на которые помогают ему обучаться.



Рисунок 1.1 — Методы машинного обучения

В стимулируемом обучении правильных ответов не существует, а алгоритм пытается найти оптимальную стратегию.

Обучение без учителя отличается тем, что алгоритм ищет не пары объект-ответ, а связи между объектами.

Рассмотрим подробнее обучение с учителем и без него.

### 1.1.1 Обучение с учителем

Задача можно поставить следующим образом: пусть X — множество объектов, Y — множество ответов и имеется некоторая зависимость (детерминированная или вероятностная), позволяющая по x∈X предсказать y∈Y. То есть, если эта зависимость детерминированная, то существует некоторая функция ƒ\*:X→Y. Причем изначально, эта зависимость известна только на объектах из обучающей выборки. Задача машинного обучения с учителем — научиться по новым объектам x∈X предсказывать y∈Y, т.е. восстановить зависимость.

В зависимости от множества Y выделяют типы задач обучения с учителем, представленные ниже [4].

Задача классификации, которая заключается в том, по x∈X требуется предсказать, к какому классу он принадлежит, при условии, что множество X разбивается на K классов. В частности, задачей классификации является задача распознавания образов. Существующие методы решения задачи классификации [5]:

* линейный классификатор, основанный на построении линейной разделяющей поверхности. В случае двух классов разделяющей поверхностью является гиперплоскость, которая делит пространство признаков на два полупространства. В случае большего числа классов разделяющая поверхность кусочно-линейна;
* байесовский классификатор, основанный на принципе максимума апостериорной вероятности. Для классифицируемого объекта вычисляются функции правдоподобия каждого из классов, по ним вычисляются апостериорные вероятности классов. Объект относится к тому классу, для которого апостериорная вероятность максимальна;
* нейронные сети, представляющие из себя математическую модель, имитирующую биологическую нейронную сеть мозга.

Задача восстановления регрессии, в которой требуется найти функцию ƒ из определенного класса, которая аппроксимирует ƒ\*.

Задача ранжирования, которая заключается в автоматическом подборе по обучающей выборке ранжирующей модели, позволяющей получать ответы и упорядочивать их.

Задача прогнозирования, особенность которой заключается в том, что объектами являются отрезки временных рядов, обрывающиеся в тот момент, когда требуется сделать прогноз на будущее.

### 1.1.2 Обучение без учителя

В случае обучения без учителя, обучающая выборка состоит только из объектов. X — множество объектов. Задача программы — определить, как элементы из X связаны между собой. В частности, решить задачу кластеризации, то есть разбить объекты на группы таким образом, чтобы в одной группе оказались объекты похожие, а в разных — существенно различные.

В обучении без учителя выделяют несколько типов задач [6].

Задача кластеризации состоит в разбиении объектов на группы (кластеры), так, чтобы в одном кластере оказались близкие друг к другу объекты, а в разных кластерах объекты были существенно различные.

Задача обнаружение аномалий состоит в идентификации точек данных, которые просто являются необычными.

Задача сокращения размерности заключается в том, чтобы по исходным признакам с помощью некоторых функций преобразования перейти к наименьшему числу новых признаков, не потеряв при этом никакой существенной информации об объектах выборки

Задача [поиска ассоциативных правил](http://www.machinelearning.ru/wiki/index.php?title=%D0%9F%D0%BE%D0%B8%D1%81%D0%BA_%D0%B0%D1%81%D1%81%D0%BE%D1%86%D0%B8%D0%B0%D1%82%D0%B8%D0%B2%D0%BD%D1%8B%D1%85_%D0%BF%D1%80%D0%B0%D0%B2%D0%B8%D0%BB&action=edit) отличается тем, что исходные данные представляются в виде признаковых описаний. Требуется найти такие наборы признаков значения этих признаков, которые неслучайно часто встречаются в признаковых описаниях объектов.

В задаче заполнения пропущенных значений при помощи прогнозных значений происходит замена недостающих значений в матрице <объекты – признаки>.

## 1.2 Признаковое пространство

Признаком называется отображение , где  — множество допустимых значений признака. Если заданы признаки , то вектор  — признаковое описание объекта . Признаковые описания допустимо отождествлять с самими объектами. При этом множество  называют признаковым пространством [6].

В зависимости от множества  признаки делятся на следующие типы:

* бинарный признак: ;
* номинальный признак:  — конечное множество;
* порядковый признак:  — конечное упорядоченное множество;
* количественный признак:  — множество действительных чисел.

## 1.3 Переобучение и недообучение

Говорят, что [алгоритм обучения](http://www.machinelearning.ru/wiki/index.php?title=%D0%90%D0%BB%D0%B3%D0%BE%D1%80%D0%B8%D1%82%D0%BC_%D0%BE%D0%B1%D1%83%D1%87%D0%B5%D0%BD%D0%B8%D1%8F) обладает способностью к обобщению, если вероятность ошибки на [тестовой выборке](http://www.machinelearning.ru/wiki/index.php?title=%D0%A2%D0%B5%D1%81%D1%82%D0%BE%D0%B2%D0%B0%D1%8F_%D0%B2%D1%8B%D0%B1%D0%BE%D1%80%D0%BA%D0%B0) достаточно мала или хотя бы предсказуема, то есть не сильно отличается от ошибки на [обучающей выборке](http://www.machinelearning.ru/wiki/index.php?title=%D0%9E%D0%B1%D1%83%D1%87%D0%B0%D1%8E%D1%89%D0%B0%D1%8F_%D0%B2%D1%8B%D0%B1%D0%BE%D1%80%D0%BA%D0%B0). Обобщающая способность тесно связана с понятиями переобучения и недообучения [7].

*Переобучение, переподгонка*(overtraining, overfitting) — нежелательное явление, возникающее при решении задач [обучения по прецедентам](http://www.machinelearning.ru/wiki/index.php?title=%D0%9E%D0%B1%D1%83%D1%87%D0%B5%D0%BD%D0%B8%D0%B5_%D0%BF%D0%BE_%D0%BF%D1%80%D0%B5%D1%86%D0%B5%D0%B4%D0%B5%D0%BD%D1%82%D0%B0%D0%BC), когда вероятность ошибки обученного алгоритма на объектах [тестовой выборки](http://www.machinelearning.ru/wiki/index.php?title=%D0%A2%D0%B5%D1%81%D1%82%D0%BE%D0%B2%D0%B0%D1%8F_%D0%B2%D1%8B%D0%B1%D0%BE%D1%80%D0%BA%D0%B0) оказывается существенно выше, чем средняя ошибка на [обучающей выборке](http://www.machinelearning.ru/wiki/index.php?title=%D0%9E%D0%B1%D1%83%D1%87%D0%B0%D1%8E%D1%89%D0%B0%D1%8F_%D0%B2%D1%8B%D0%B1%D0%BE%D1%80%D0%BA%D0%B0). Переобучение возникает при использовании избыточно сложных [моделей](http://www.machinelearning.ru/wiki/index.php?title=%D0%9C%D0%BE%D0%B4%D0%B5%D0%BB%D1%8C_%D0%B7%D0%B0%D0%B2%D0%B8%D1%81%D0%B8%D0%BC%D0%BE%D1%81%D1%82%D0%B8).

*Недообучение* — нежелательное явление, возникающее при решении задач [обучения по прецедентам](http://www.machinelearning.ru/wiki/index.php?title=%D0%9E%D0%B1%D1%83%D1%87%D0%B5%D0%BD%D0%B8%D0%B5_%D0%BF%D0%BE_%D0%BF%D1%80%D0%B5%D1%86%D0%B5%D0%B4%D0%B5%D0%BD%D1%82%D0%B0%D0%BC), когда [алгоритм обучения](http://www.machinelearning.ru/wiki/index.php?title=%D0%90%D0%BB%D0%B3%D0%BE%D1%80%D0%B8%D1%82%D0%BC_%D0%BE%D0%B1%D1%83%D1%87%D0%B5%D0%BD%D0%B8%D1%8F) не обеспечивает достаточно малой величины средней ошибки на [обучающей выборке](http://www.machinelearning.ru/wiki/index.php?title=%D0%9E%D0%B1%D1%83%D1%87%D0%B0%D1%8E%D1%89%D0%B0%D1%8F_%D0%B2%D1%8B%D0%B1%D0%BE%D1%80%D0%BA%D0%B0). Недообучение возникает, в отличии от переобучения, при использовании недостаточно сложных [моделей](http://www.machinelearning.ru/wiki/index.php?title=%D0%9C%D0%BE%D0%B4%D0%B5%D0%BB%D1%8C_%D0%B7%D0%B0%D0%B2%D0%B8%D1%81%D0%B8%D0%BC%D0%BE%D1%81%D1%82%D0%B8).

Большей проблемой на данный момент является именно переобучение, которое связано с избыточной сложностью используемой [модели](http://www.machinelearning.ru/wiki/index.php?title=%D0%9C%D0%BE%D0%B4%D0%B5%D0%BB%D1%8C_%D0%B7%D0%B0%D0%B2%D0%B8%D1%81%D0%B8%D0%BC%D0%BE%D1%81%D1%82%D0%B8).

Возможные причины переобучения:

* линейная зависимость признаков;
* слишком мало объектов;
* слишком много признаков.

Всегда существует оптимальное значение сложности модели, при котором переобучение минимально. Рассмотрим методы борьбы с этим явлением.

Скользящий контроль (или кросс-валидация, cross-validation, CV) — процедура эмпирического оценивания [обобщающей способности](http://www.machinelearning.ru/wiki/index.php?title=%D0%9E%D0%B1%D0%BE%D0%B1%D1%89%D0%B0%D1%8E%D1%89%D0%B0%D1%8F_%D1%81%D0%BF%D0%BE%D1%81%D0%BE%D0%B1%D0%BD%D0%BE%D1%81%D1%82%D1%8C) алгоритмов, [обучаемых по прецедентам](http://www.machinelearning.ru/wiki/index.php?title=%D0%9E%D0%B1%D1%83%D1%87%D0%B5%D0%BD%D0%B8%D0%B5_%D0%BF%D0%BE_%D0%BF%D1%80%D0%B5%D1%86%D0%B5%D0%B4%D0%B5%D0%BD%D1%82%D0%B0%D0%BC), один из самых популярных методов борьбы с переобучением.

Коротко говоря, метод заключается в том, что фиксируется некоторое множество разбиений исходной выборки на две подвыборки: обучающую и контрольную. Для каждого разбиения выполняется настройка [алгоритма](http://www.machinelearning.ru/wiki/index.php?title=%D0%90%D0%BB%D0%B3%D0%BE%D1%80%D0%B8%D1%82%D0%BC) по обучающей подвыборке, затем оценивается его средняя ошибка на объектах контрольной подвыборки. Оценкой скользящего контроля называется средняя по всем разбиениям величина ошибки на контрольных подвыборках.

### 1.3.1 Процедура скользящего контроля

В общем случае выборка  разбивается N различными способами на две непересекающиеся подвыборки: , где  — обучающая подвыборка длины m,  — контрольная подвыборка длины , n=1,..,N — номер разбиения.

Для каждого разбиения n строится алгоритм  и вычисляется значение функционала качества . Оценка скользящего контроля — среднее арифметическое значений  по всем разбиениям:

|  |  |  |
| --- | --- | --- |
|  | . | (1) |

Различные варианты скользящего контроля отличаются видами функционала качества и способами разбиения выборки.

Рассмотрим разновидности скользящего контроля.

#### 1.3.1.1 Полный скользящий контроль (complete CV)

Оценка скользящего контроля строится по всем  разбиениям. В зависимости от k (длины обучающей выборки) различают:

* частный случай при  — [контроль по отдельным объектам (leave-one-out CV)](http://www.machinelearning.ru/wiki/index.php?title=%D0%A1%D0%BA%D0%BE%D0%BB%D1%8C%D0%B7%D1%8F%D1%89%D0%B8%D0%B9_%D0%BA%D0%BE%D0%BD%D1%82%D1%80%D0%BE%D0%BB%D1%8C#.D0.9A.D0.BE.D0.BD.D1.82.D1.80.D0.BE.D0.BB.D1.8C_.D0.BF.D0.BE_.D0.BE.D1.82.D0.B4.D0.B5.D0.BB.D1.8C.D0.BD.D1.8B.D0.BC_.D0.BE.D0.B1.D1.8A.D0.B5.D0.BA.D1.82.D0.B0.D0.BC_.28leave-one-out_CV.29);
* общий случай при . Здесь число разбиений  становится слишком большим даже при сравнительно малых значениях k, что затрудняет практическое применение данного метода. Для этого случая полный скользящий контроль используется либо в теоретических исследованиях (Воронцов, 2004), либо в тех редких ситуациях, когда для него удаётся вывести эффективную вычислительную формулу.

#### 1.3.1.2 Контроль на отложенных данных (hold-out CV)

Оценка скользящего контроля строится по одному случайному разбиению, N=1.

Этот способ имеет существенные недостатки:

* слишком много объектов приходится оставлять в контрольной подвыборке. Уменьшение длины обучающей подвыборки приводит к смещённой (пессимистически завышенной) оценке вероятности ошибки;
* оценка существенно зависит от разбиения, тогда как желательно, чтобы она характеризовала только алгоритм обучения;
* оценка имеет высокую дисперсию, которая может быть уменьшена путём усреднения по разбиениям.

Следует различать скользящий контроль по отложенным данным и контроль по тестовой выборке. Если во втором случае оценивается вероятность ошибки для классификатора, построенного по обучающей подвыборке, то в первом случае — для классификатора, построенного по полной выборке (то есть доля ошибок вычисляется не для того классификатора, который выдается в качестве результата решения задачи).

#### 1.3.1.3 Контроль по отдельным объектам (leave-one-out CV)

Является частным случаем полного скользящего контроля при k=1, соответственно, N=L. Это, пожалуй, самый распространённый вариант скользящего контроля.

Преимущества LOO в том, что каждый объект ровно один раз участвует в контроле, а длина обучающих подвыборок лишь на единицу меньше длины полной выборки.

Недостатком LOO является большая ресурсоёмкость, так как обучаться приходится L раз. Некоторые методы обучения позволяют достаточно быстро перенастраивать внутренние параметры алгоритма при замене одного обучающего объекта другим. В этих случаях вычисление LOO удаётся заметно ускорить.

#### 1.3.1.4 Контроль по q блокам (q-fold CV)

Выборка случайным образом разбивается на q непересекающихся блоков одинаковой (или почти одинаковой) длины :

|  |  |  |
| --- | --- | --- |
|  |  | (2) |
|  |  | (3) |

Каждый блок по очереди становится контрольной подвыборкой, при этом обучение производится по остальным q-1 блокам. Критерий определяется как средняя ошибка на контрольной подвыборке:

|  |  |  |
| --- | --- | --- |
|  |  | (4) |

В таком случае, обучение производится только q раз, а не L. А так же длина обучающих подвыборок, равная  с точностью до округления, не сильно отличается от длины полной выборки L. Обычно выборку разбивают случайным образом на 10 или 20 блоков.

#### 1.3.1.5 Контроль по r×q блокам (r×q-fold CV)

Контроль по q блокам (q-fold CV) повторяется r раз. Каждый раз выборка случайным образом разбивается на q непересекающихся блоков. Этот способ наследует все преимущества q-fold CV, при этом появляется дополнительная возможность увеличивать число разбиений.

Существуют некоторые недостатки скользящего контроля.

Во-первых, задачу обучения приходится решать N раз, что ведет за собой значительные вычислительные затраты.

Во-вторых, оценка скользящего контроля предполагает, что алгоритм обучения  уже задан. Она ничего не говорит о том, какими свойствами должны обладать «хорошие» алгоритмы обучения, и как их строить. Такого рода подсказки дают, например, теоретические оценки обобщающей способности.

Также попытка использовать скользящий контроль для обучения, в роли оптимизируемого критерия, приводит к тому, что он утрачивает свойство несмещённости, и снова возникает риск [переобучения](http://www.machinelearning.ru/wiki/index.php?title=%D0%9F%D0%B5%D1%80%D0%B5%D0%BE%D0%B1%D1%83%D1%87%D0%B5%D0%BD%D0%B8%D0%B5).

И, наконец, скользящий контроль дает несмещенную точечную, но не интервальную оценку риска. В настоящее время не существует методов построения на основе скользящего контроля точных доверительных интервалов для риска, то есть математического ожидания потерь (в частности, вероятности ошибочной классификации).

В связи с этим рассмотрим ниже еще два метода, помогающие уменьшить степень переобучения.

### 1.3.2 **L1- и L2-регуляризация в машинном обучении**

L1- и L2-регуляризация — эта два тесно связанных метода, которые также можно применять в алгоритмах машинного обучения для уменьшения степени переобучения модели.

Регуляризация — метод добавления некоторой дополнительной информации к условию с целью решить [некорректно поставленную задачу](https://ru.wikipedia.org/wiki/%D0%9D%D0%B0%D1%87%D0%B0%D0%BB%D1%8C%D0%BD%D1%8B%D0%B5_%D0%B8_%D0%B3%D1%80%D0%B0%D0%BD%D0%B8%D1%87%D0%BD%D1%8B%D0%B5_%D1%83%D1%81%D0%BB%D0%BE%D0%B2%D0%B8%D1%8F#.D0.9A.D0.BE.D1.80.D1.80.D0.B5.D0.BA.D1.82.D0.BD.D0.BE.D1.81.D1.82.D1.8C_.D0.BF.D0.BE.D1.81.D1.82.D0.B0.D0.BD.D0.BE.D0.B2.D0.BA.D0.B8_.D0.B3.D1.80.D0.B0.D0.BD.D0.B8.D1.87.D0.BD.D1.8B.D1.85_.D1.83.D1.81.D0.BB.D0.BE.D0.B2.D0.B8.D0.B9) или предотвратить [переобучение](https://ru.wikipedia.org/wiki/%D0%9F%D0%B5%D1%80%D0%B5%D0%BE%D0%B1%D1%83%D1%87%D0%B5%D0%BD%D0%B8%D0%B5). Эта информация часто имеет вид штрафа за сложность модели.

С [байесовской](https://ru.wikipedia.org/wiki/%D0%91%D0%B0%D0%B9%D0%B5%D1%81%D0%BE%D0%B2%D1%81%D0%BA%D0%B0%D1%8F_%D0%B2%D0%B5%D1%80%D0%BE%D1%8F%D1%82%D0%BD%D0%BE%D1%81%D1%82%D1%8C) точки зрения многие методы регуляризации соответствуют добавлению некоторых [априорных распределений](https://ru.wikipedia.org/wiki/%D0%90%D0%BF%D1%80%D0%B8%D0%BE%D1%80%D0%BD%D0%BE%D0%B5_%D1%80%D0%B0%D1%81%D0%BF%D1%80%D0%B5%D0%B4%D0%B5%D0%BB%D0%B5%D0%BD%D0%B8%D0%B5) на параметры модели.

Существуют два типа регуляризации.

Регуляризация ridge regression для [интегральных уравнений](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%82%D0%B5%D0%B3%D1%80%D0%B0%D0%BB%D1%8C%D0%BD%D0%BE%D0%B5_%D1%83%D1%80%D0%B0%D0%B2%D0%BD%D0%B5%D0%BD%D0%B8%D0%B5) позволяет балансировать между соответствием данным и маленькой нормой решения:

|  |  |  |
| --- | --- | --- |
|  |  | (5) |

Первое слагаемое — среднеквадратичная ошибка, второй — регуляризирующий коэффициент.  В процессе работы алгоритма размеры коэффициентов будут пропорциональны важности соответствующих переменных, а перед теми переменными, которые дают наименьший вклад в устранение ошибки, станут околонулевые. Параметр λ в свою очередь позволяет настраивать вклад регуляризирующего оператора в общую сумму. С его помощью мы можем указать приоритет — точность модели или минимальное количество используемых переменных.

Регуляризация [lasso regression](https://en.wikipedia.org/wiki/lasso_regression):

|  |  |  |
| --- | --- | --- |
|  |  | (6) |

Если в ridge по мере роста альфа все коэффициенты получают значения все ближе к нулевым, но обычно при этом все-таки не зануляются. То в LASSO с ростом альфа все больше коэффициентов становятся нулевыми и совсем перестают вносить вклад в модель.

Переобучение в большинстве случаев проявляется в том, что в получающихся многочленах слишком большие коэффициенты. Соответственно, и бороться с этим можно довольно естественным способом: нужно просто добавить в целевую функцию штраф, который бы наказывал модель за слишком большие коэффициенты.

# **2 Метод обнаружения аномалий**

Исследуем более подробно метод обнаружения аномалий. Этот метод является важной задачей машинного обучения, рассматриваемой во многих областях исследования и сферах применения. Выявление аномалий (или детектирование аномалий, поиск аномалий) относится к проблеме нахождения шаблонов данных, не соответствующих ожидаемому поведению.

В общей формулировке детектирование аномалий предполагает поиск нетиповых последовательностей во временных рядах. Пример детектирования аномалий представлен на рисунке 2.1.

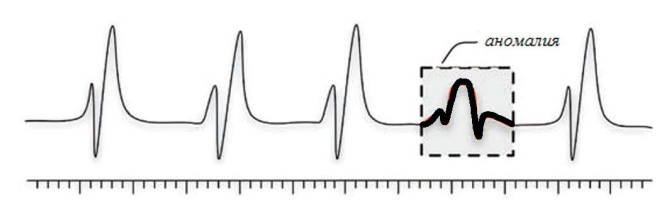


Рисунок 2.1 — Пример детектирования аномалий

При математическом обозначении данная формулировка может быть представлена следующим образом:

Дан временной ряд , найти все последовательности , содержащие ненормальные для исходного временного ряда данные.

Обнаружение аномалий широко используется в таких областях применения, как обнаружение вторжений в компьютерной безопасности, обнаружение мошенничества при проведении банковских транзакций, поиск отказов в системе безопасности, слежение за вражеской активностью и др.

Аномалиями являются такие шаблоны данных, которые не удовлетворяют предопределенному понятию нормального поведения. Рисунок 2.2 иллюстрирует наглядный пример аномалий в двумерном пространстве. Представленные данные разделены на две нормальных области,  и , заключающих в себе большинство наблюдений. Точки, лежащие вне этих областей, т.е. точки , , а также точки области A являются аномалиями [8].

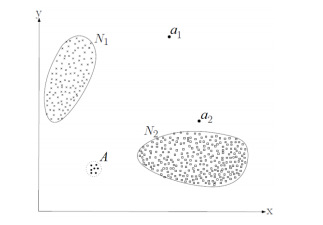


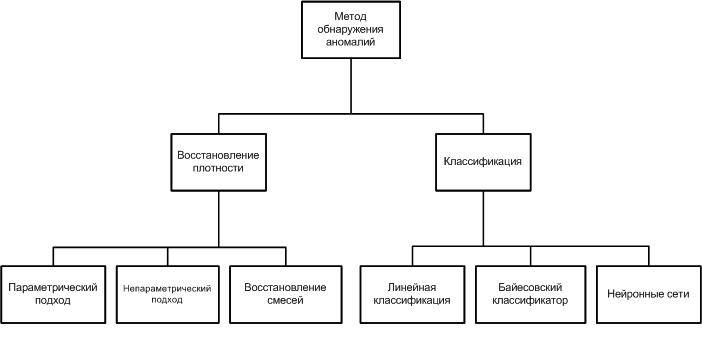
Рисунок 2.2 — Пример аномалий в двумерном пространстве

Проблема детектирования аномалий связана со смежными проблемами, такими как удаление шума и искусственное добавление шума, в виду того, что решение этих проблем предполагает изучение ненужной или вредоносной информации в данных. Шум может быть определен как некоторое явление в исследуемых данных, которое находится вне зоны интереса аналитика, но является помехой для анализа исследуемых данных. Методы удаления шума позволяют избавиться от таких явлений до проведения анализа данных. Методы добавления шума предполагают «иммунизацию» статистической модели к аномальным наблюдениям.

Ключевым аспектом любого подхода к обнаружению аномалий является природа анализируемых данных, которые в общем смысле являются набором объектов. Каждый объект данных может быть описан рядом признаков.

Базовый подход к обнаружению аномалий заключается в определении области нормального поведения и в отнесении любого наблюдения, лежащего вне данной области, к аномальному. Существует ряд трудностей, неблагоприятным образом влияющих на этот, казалось бы, простой подход. К ним относятся: сложность определения области, сочетающей в себе всевозможные линии нормального поведения. При этом граница между нормальным и аномальным поведением не всегда четкая. В связи с такими сложностями, как различное понятие «аномалия» для различных областей применения, сложность доступной маркировки данных для обучения моделей, используемых методами обнаружения аномалий, сложность удаления нежелательной шумовой составляющей, решение проблемы обнаружения аномалий на основе базового подхода не является тривиальным. Поэтому большинство создаваемых методик обнаружения аномалий базируются на специфической формулировке проблемы, основанной на ряде признаков.

Существует два основных метода обнаружения аномалий, один из которых основан на восстановлении плотности, а другой сводится к методу классификации (рисунок 2.3) [9].

Рисунок 2.3 — Классификация методов обнаружения аномалий

### 2.1 Метод, основанный на восстановлении плотности

Рассматривается распределение, описывающее имеющуюся выборку объектов. Далее происходит оценка, насколько вероятно получить конкретный новый объект из этого распределения. Чем меньше эта вероятность, тем больше шансов, что рассматриваемый объект — аномальное явление.

В вероятностном подходе считают, что аномалия – объект, который был получен из другого распределения, а не из того, которое сгенерировало основную обучающую выборку [10]. Необходимо найти распределение, с помощью которого была сгенерирована выборка, тогда будет возможность оценить вероятность появления объекта из этого или же другого распределения. Если вероятность мала, вероятнее всего, объект является аномалией.

Среди методов, основанных на восстановлении плотности, рассмотрим несколько.

#### 2.1.1 Параметрическое восстановление плотности

Существует некоторое вероятностное распределение , где  — параметр распределения, которое можно получить. Необходимо, чтобы параметр распределения был найден так, чтобы вероятность нахождения объектов обучающей выборки в области распределения была максимальна. В таком случае объекты, которые не похожи на эту выборку, будут получать низкую вероятность.

Используется метод максимального правдоподобия, который подбирает такое распределение из параметрического семейства, что объекты обучающей выборки будут как можно более вероятны:

|  |  |  |
| --- | --- | --- |
|  |  | (7) |

Итак, если есть некоторое распределение на объектах, можем использовать некий алгоритм поиска аномалий. Предположим, имеем новый объект x, необходимо понять, аномальный он или же удовлетворяет распределению. Необходимо оценить вероятность p(x) с точки зрения данного распределения и оценить ее относительно порогового параметра t. Если вероятность меньше, можем считать объект x аномалией.

Параметр t можно определять из нескольких соображений: из априорных, например, если 99% объектов расположены внутри области с порогом t, то 1% процент остается вне области, или же по известным аномалиям, когда мы заранее знаем, что несколько объектов являются аномальными и можем отсечь параметром t именно данную область.

#### 2.1.2 Непараметрический подход восстановления плотности

Данный подход заключается в идее восстановления вида распределения, глядя на данные, который позволяет использовать метод Парзена-Розенблатта:

|  |  |  |
| --- | --- | --- |
|  |  | (8) |

т.е. в каждой точке обучающей выборки вычисляется разность между новым объектом, в котором нужно оценить плотность x, и данной точкой обучающей выборки, а h — некий параметр, ширина окна, от которого зависит, будет ли выборка переобученной или же недообученной. Ядро К — вероятность, определяющая на сколько эти точки похожи. Ядро — четная функция, для которой выполняется следующее условие:

|  |  |  |
| --- | --- | --- |
|  |  | (9) |

В основном непараметрический подход используется для восстановления сложных плотностей с большим количеством пиков.

В многомерном случае требуется очень большая выборка для качественного восстановления плотности. Если же удалось восстановить плотность, алгоритм поиска аномалий можно использовать такой же, что и при параметрическом подходе.

#### 2.1.3 Восстановление смесей

Смесью называют такое распределение

|  |  |  |
| --- | --- | --- |
|  |  | (10) |

где — параметрические распределения. В данном методе используется так называемый EM-алгоритм, используемый в математической статистике для нахождения оценок максимального правдоподобия параметров вероятностных моделей, в случае, когда модель зависит от некоторых скрытых переменных, который заключается в повторении E и M шагов до тех пор, пока не будет достигнута сходимость. На Е-шаге вычисляется вероятность того, что i-ый объект принадлежит j-ой компоненте смеси:

|  |  |  |
| --- | --- | --- |
|  |  | (11) |

На М-шаге вычисляется оценка максимального правдоподобия, используются эти вероятности, чтобы обновить значения на параметре θ:

|  |  |  |
| --- | --- | --- |
|  |  | (12) |
|  |  | (13) |

Таким образом, увеличивается ожидаемое правдоподобие, вычисляемое на E-шаге. Затем это значение используется для E-шага на следующей итерации. Происходит в цикле до тех пор, пока параметр  не будет меняться слишком слабо. Благодаря этому алгоритму, можно подогнать смесь из К под необходимую выборку.

### 2.2 Сведение к методу классификации

Существует некоторая связь между методом обнаружения аномалий с методом классификации.

#### 2.2.1 Линейные классификаторы

Одним из основных типов метода классификации является линейный классификатор. Рассмотрим его основные методы.

##### 2.2.1.1 Одноклассовый метод опорных векторов

В задачи классификации необходимо разделить выборку на два класса прямой, а в методе обнаружения аномалий отделить некой кривой область, которая включит в себя выборку и вне себя оставит аномальные объекты [11]. При соединении этих двух методов можем представить, что первый класс – это нормальные объекты, обучающая выборка, а второй класс – аномалии, причем аномалиями считаются объекты начала координат.

Применим метод опорных векторов для обнаружения аномалий. Задача для начала координат будет выглядеть следующим образом:

|  |  |  |
| --- | --- | --- |
|  | . | (14) |

Важный параметр задачи — ν, задаёт верхнюю оценку для доли аномальных объектов в выборке.

Предположение о том, что объекты начала координат - аномальные, например, если выборка центрирована вокруг начала координат, не может быть верным. На самом деле, одноклассовый SVM с линейным ядром никогда не используется. Скалярное произведение заменим на ядро K. Если разделяющая гиперплоскость будет строиться в пространстве более высокой размерности, используется RBF-ядро (радиально-базисные функции), которое вычисляется по формуле:

|  |  |  |
| --- | --- | --- |
|  |  | (15) |

##### 2.2.1.2 Логистическая регрессия

Рассматривается задача бинарной классификации, где в выборке имеются пары <объект – ответ>. Объект описывается n вещественными признаками.

|  |  |  |
| --- | --- | --- |
|  |  | (16) |

где  — вес j-го признака,  — порог принятия решения,  — вектор весов,  – скалярное произведение признакового описания объекта на вектор весов.

Объект относится к одному классу, если скалярное произведение положительное. К другому, если скалярное произведение отрицательное. Введем важное понятие — отступ объекта, равное скалярному произведению, умноженному на правильный ответ, где правильный ответ ±1. Поэтому если у нас на объекте есть ошибка — отступ отрицательный, если ошибка отсутствует — отступ положительный, и функция потерь I, если она монотонно убывает, то она штрафует нас за ошибки и даже штрафует за приближение к границе между классами.

#### 2.2.2 Байесовский классификатор

Байесовский классификатор основан на принципе максимума апостериорной вероятности. Рассмотрим наивный байесовский алгоритм, а так же метод ближайшего соседа.

##### 2.2.2.1 Наивный байесовский алгоритм

Наивный байесовский алгоритм (далее — НБА) — это алгоритм классификации, основанный на [теореме Байеса](https://ru.wikipedia.org/wiki/%D0%A2%D0%B5%D0%BE%D1%80%D0%B5%D0%BC%D0%B0_%D0%91%D0%B0%D0%B9%D0%B5%D1%81%D0%B0) с допущением о независимости признаков. Другими словами, наивный байесовский классификатор предполагает, что наличие какого-либо признака в классе не связано с наличием какого-либо другого признака. Модели на основе НБА достаточно просты и полезны при работе с очень большими наборами данных. При своей простоте НБА способен превзойти даже некоторые сложные алгоритмы классификации. Предположение о независимости признаков существенно упрощает задачу, поскольку оценить n одномерных плотностей легче, чем одну n-мерную.

На практике наивный байесовский классификатор употребляется крайне редко, лишь в качестве алгоритмической композиции или примитивного эталона в сравнении с другими моделями алгоритмов.

##### 2.2.2.2 Метод ближайшего соседа

Метод ближайшего соседа—простейший [метрический классификатор](http://www.machinelearning.ru/wiki/index.php?title=%D0%9C%D0%B5%D1%82%D1%80%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B8%D0%B9_%D0%BA%D0%BB%D0%B0%D1%81%D1%81%D0%B8%D1%84%D0%B8%D0%BA%D0%B0%D1%82%D0%BE%D1%80), основанный на оценивании [сходства](http://www.machinelearning.ru/wiki/index.php?title=%D0%A1%D1%85%D0%BE%D0%B4%D1%81%D1%82%D0%B2%D0%BE&action=edit) объектов. Классифицируемый объект относится к тому классу, которому принадлежат ближайшие к нему объекты обучающей выборки.

Гипотеза компактности: если мера сходства объектов введена достаточно удачно, то схожие объекты гораздо чаще лежат в одном классе, чем в разных.

#### 2.2.3 Нейронные сети

Нейронная сеть — математическая модель, построенная по принципу организации и функционирования [биологических нейронных сетей](https://ru.wikipedia.org/wiki/%D0%9D%D0%B5%D0%B9%D1%80%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F_%D1%81%D0%B5%D1%82%D1%8C) — сетей [нервных клеток](https://ru.wikipedia.org/wiki/%D0%9D%D0%B5%D0%B9%D1%80%D0%BE%D0%BD) живого организма. Это понятие возникло при изучении процессов, протекающих в [мозге](https://ru.wikipedia.org/wiki/%D0%9C%D0%BE%D0%B7%D0%B3), и при попытке [смоделировать](https://ru.wikipedia.org/wiki/%D0%9C%D0%BE%D0%B4%D0%B5%D0%BB%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5) эти процессы [12].

Искусственные нейронные сети представляют собой систему соединённых и взаимодействующих между собой простых процессоров ([искусственных нейронов](https://ru.wikipedia.org/wiki/%D0%98%D1%81%D0%BA%D1%83%D1%81%D1%81%D1%82%D0%B2%D0%B5%D0%BD%D0%BD%D1%8B%D0%B9_%D0%BD%D0%B5%D0%B9%D1%80%D0%BE%D0%BD)). Такие процессоры обычно довольно просты. Каждый процессор подобной сети имеет дело только с [сигналами](https://ru.wikipedia.org/wiki/%D0%A1%D0%B8%D0%B3%D0%BD%D0%B0%D0%BB), которые он периодически получает, и сигналами, которые он периодически посылает другим процессорам. Будучи соединёнными в достаточно большую сеть с управляемым взаимодействием, такие по отдельности простые процессоры вместе способны выполнять довольно сложные задачи.

В качестве образов могут выступать различные по своей природе объекты: символы текста, изображения, образцы звуков и другое. При обучении сети предлагаются различные образцы, как правило, — вектор значений признаков, образов с указанием того, к какому классу они относятся. При этом совокупность всех признаков должна однозначно определять класс, к которому относится образец.

Если признаков недостаточно, сеть может соотнести один и тот же образец с несколькими классами, что неверно.

По окончании обучения сети ей можно предъявлять неизвестные ранее образы и получать ответ о принадлежности к определённому классу.

Количество нейронов в выходном слое, как правило, равно количеству определяемых классов. При этом устанавливается соответствие между выходом нейронной сети и классом, который представляет нейрон. Когда сети предъявляется некий образ, на одном из её выходов должен появиться признак того, что образ принадлежит этому классу. В то же время на других выходах должен быть признак того, что образ данному классу не принадлежит. Если на двух или более выходах есть признак принадлежности к классу, считается, что сеть «не уверена» в своём ответе.

##### 2.2.3.1 Перцептрон

Перцептрон стал одной из первых моделей [нейросетей](https://ru.wikipedia.org/wiki/%D0%98%D1%81%D0%BA%D1%83%D1%81%D1%81%D1%82%D0%B2%D0%B5%D0%BD%D0%BD%D0%B0%D1%8F_%D0%BD%D0%B5%D0%B9%D1%80%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F_%D1%81%D0%B5%D1%82%D1%8C). *Перцептрон* представляет собой сеть, состоящую из S, A, R — элементов, с переменной матрицей взаимодействия V (элементы которой  – весовые коэффициенты), определяемой последовательностью прошлых состояний активности сети [13]. Пример логической схемы перцептрона с тремя выходами представлен на рисунке 2.4.

*S-элементом* (сенсорным элементом) является чувствительный элемент, который от воздействия какого-либо из видов энергии (например, света, звука, давления, тепла и т. п.) вырабатывает сигнал. Если входной сигнал  превышает некоторый порог , то элемент выдаёт выходной сигнал , в противном случае выходной сигнал равен нулю.

*A-элементом* (ассоциативным элементом) называется логический решающий элемент, который выдаёт выходной сигнал, когда алгебраическая сумма его входных сигналов  равна или превышает некоторую пороговую величину . Выходной сигнал , если была превышена пороговая величина ; в противном случае он равен нулю. Если , то говорят, что A-элемент является активным.

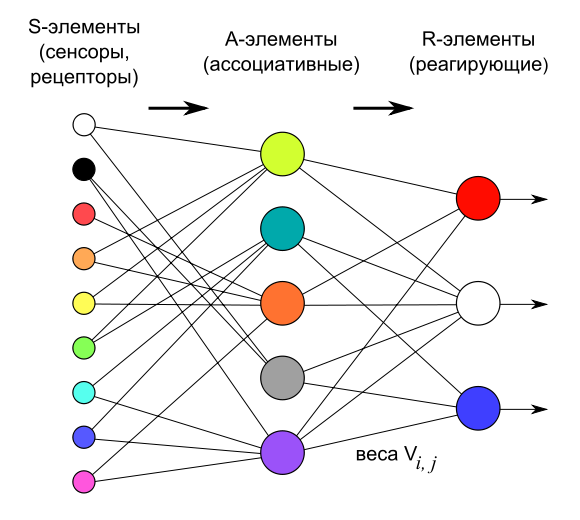


Рисунок 2.4 — Логическая схема перцептрона с тремя выходами

*R-элементом* (реагирующим элементом) называется элемент, который выдаёт выходной сигнал , если сумма его входных сигналов является строго положительной, и сигнал , если сумма его входных сигналов является строго отрицательной. Если сумма входных сигналов равна нулю, выход можно считать либо равным нулю, либо неопределённым.

Каждый рецептор может находиться в одном из двух состояний — покоя или возбуждения. В состоянии возбуждения он передаёт единичный сигнал в следующий слой, ассоциативным элементам. A-элемент активизируется, как только количество сигналов от S-элементов на его входе превысило некоторую величину θ. Сигналы от возбудившихся A-элементов, в свою очередь, передаются в сумматор R, причём сигнал от i-го ассоциативного элемента передаётся с коэффициентом  — вес A—R связи. R-элемент подсчитывает сумму значений входных сигналов, помноженных на веса.  R-элемент, а вместе с ним и элементарный перцептрон, выдаёт «1», если линейная форма превышает порог θ, иначе на выходе будет «−1». Итак, функция, реализуемая R-элементом, представляется в виде:

|  |  |  |
| --- | --- | --- |
|  |  | (17) |

[Обучение](https://ru.wikipedia.org/wiki/%D0%9C%D0%B0%D1%88%D0%B8%D0%BD%D0%BD%D0%BE%D0%B5_%D0%BE%D0%B1%D1%83%D1%87%D0%B5%D0%BD%D0%B8%D0%B5) элементарного перцептрона состоит в изменении весовых коэффициентов   связей A—R. Веса связей S—A (которые могут принимать значения {−1; 0; +1}) и значения порогов A-элементов выбираются случайным образом в самом начале и затем не изменяются. По окончании обучения, перцептрону предъявляются ранее неизвестные ему объекты, и перцептрон должен установить, к какому классу они принадлежат.

##### 2.2.3.2 [Сети адаптивного резонанса](https://ru.wikipedia.org/wiki/%D0%A1%D0%B5%D1%82%D0%B8_%D0%B0%D0%B4%D0%B0%D0%BF%D1%82%D0%B8%D0%B2%D0%BD%D0%BE%D0%B3%D0%BE_%D1%80%D0%B5%D0%B7%D0%BE%D0%BD%D0%B0%D0%BD%D1%81%D0%B0)

[Сети адаптивного резонанса](https://ru.wikipedia.org/wiki/%D0%A1%D0%B5%D1%82%D0%B8_%D0%B0%D0%B4%D0%B0%D0%BF%D1%82%D0%B8%D0%B2%D0%BD%D0%BE%D0%B3%D0%BE_%D1%80%D0%B5%D0%B7%D0%BE%D0%BD%D0%B0%D0%BD%D1%81%D0%B0) включают в себя модели, использующие [обучение с учителем](https://ru.wikipedia.org/wiki/%D0%9E%D0%B1%D1%83%D1%87%D0%B5%D0%BD%D0%B8%D0%B5_%D1%81_%D1%83%D1%87%D0%B8%D1%82%D0%B5%D0%BB%D0%B5%D0%BC) и [без учителя](https://ru.wikipedia.org/wiki/%D0%9E%D0%B1%D1%83%D1%87%D0%B5%D0%BD%D0%B8%D0%B5_%D0%B1%D0%B5%D0%B7_%D1%83%D1%87%D0%B8%D1%82%D0%B5%D0%BB%D1%8F) и используются при решении задач [распознавания образов](https://ru.wikipedia.org/wiki/%D0%A0%D0%B0%D1%81%D0%BF%D0%BE%D0%B7%D0%BD%D0%B0%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5_%D0%BE%D0%B1%D1%80%D0%B0%D0%B7%D0%BE%D0%B2) и предсказания [14].

Большая часть информации, поступающая в человеческий мозг не представляет интереса или неважна и поэтому игнорируется. Однако та небольшая ее составляющая, что имеет ценность, должна быть не только обработана, но и сохранена в долговременной памяти. Важно, чтобы запомнились не только новые образы, но также идентифицировались ранее встречавшиеся. В свою очередь, запоминание новых образов не должно влиять на образы, хранящиеся в памяти.

Именно эти особенности работы человеческого мозга отражены в теории адаптивного резонанса.

Алгоритм включает в себя три стадии жизненного цикла:

* инициализация, в процессе которой устанавливается параметр сходства Rкр (0 < Rкр < 1), причем чем он больше, тем выше должно быть сходство образа и кластера для отнесения к последнему. Затем для первого поданного на входы сети образа создается первый нейрон;
* распознавание образа, в процессе которого на входы сети подается новый входной образ. Для каждого нейрона рассчитывается значение его выхода;
* **кластеризация образа (сравнения)**, когда рассчитывается количественная мера сходства входного образа с кластером, имеющим наибольшее значение выхода нейрона. Если условие Rj > Rкр выполняется, нейрон считается нейроном-победителем. Если нет, этап кластеризации повторяется для остальных кластеров, пока не будет получен нейрон-победитель или не будут использованы все кластеры с положительным выходом нейрона.

##### 2.2.3.3 Сеть радиально-базисных функций

Сеть радиально-базисных функций — искусственная нейронная сеть, у которой функциями активации являются радиальные базисные функции (РБФ), которая содержит промежуточный (скрытый) слой радиально симметричных [нейронов](http://www.machinelearning.ru/wiki/index.php?title=%D0%9D%D0%B5%D0%B9%D1%80%D0%BE%D0%BD%D0%BE%D0%B2&action=edit). Такой нейрон преобразовывает расстояние от данного входного вектора до соответствующего ему "центра" по некоторому нелинейному закону [15].

Общий вид радиальной базисной функции:

|  |  |  |
| --- | --- | --- |
|  |  | (18) |

где х — вектор входных сигналов нейрона, σ — ширина окна функции, ϕ(x) — убывающая функция.

Задача данного метода - построить алгоритм, который бы решал задачу классификации байесовским алгоритмом (частный случай EM-алгоритма) в предположении, что плотность распределения представима в виде смеси гауссовских распределений с диагональными матрицами ковариации.

Преимущества радиально-базисных функций [16]:

* моделируют произвольную нелинейную функцию с помощью всего одного промежуточного слоя, тем самым избавляя разработчика от необходимости решать вопрос о числе слоев;
* параметры линейной комбинации в выходном слое можно полностью оптимизировать с помощью хорошо известных методов линейной оптимизации, которые работают быстро и не испытывают трудностей с локальными минимумами.

Недостаток заключается в том, что сети получаются весьма громоздкими при большой размерности вектора входов

#### 2.2.4 Методы оценки качества классификации

Представим так называемую Матрицу ошибок:

* верное срабатывание (TruePositive) – объект относится к классу y=1 и возвращает нам a(x)=1;
* ложное срабатывание (FalsePositive) – объект относится к классу y=0, а возвращает a(x)=1;
* ложный пропуск (FalseNegative)– объект относится к классу y=1, возвращает a(x)=0;
* верный пропуск (TrueNegative) - объект относится к классу y=0 и возвращает нам a(x)=0.

Ее наглядное представление в таблице 1.

Таблица 1 — Матрица ошибок

|  |  |  |
| --- | --- | --- |
|  | y=1 | y=0 |
| a(x)=1 | TruePositive (TP) | FalsePositive (FP) |
| a(x)=0 | FalseNegative (FN) | TrueNegative (TN) |

Один из самых очевидных методов избавиться от ошибок — доля правильных ответов на выборке (accurary):

|  |  |  |
| --- | --- | --- |
|  |  | (19) |

Можем выразить долю правильных ответов через четыре показателя матрицы:

|  |  |  |
| --- | --- | --- |
|  |  | (20) |

Этот метод соответствует интуитивным представлениям о качестве классификации, но имеет проблемы с интерпретацией на несбалансированных выборках.

Ошибки имеют разную цену, так как в реальных задачах мы можем получать нерелевантную долю правильных ответов.

В случае если разные ошибки имеют разную цену, измеряют две метрики качества — полноту и точность.

Точность (precision) — насколько можно доверять классификатору:

|  |  |  |
| --- | --- | --- |
|  |  | (21) |

Полнота (recall) — как много объектов класса 1 находит классификатор:

|  |  |  |
| --- | --- | --- |
|  |  | (22) |

Свойства точности и полноты:

* в разных задачах можно отдавать приоритет точности или полноте;
* точность и полнота характеризуют разные стороны качества классификатора;
* чем выше точность, тем меньше ложных срабатываний;
* чем выше полнота, тем меньше ложных пропусков.

В задачах машинного обучения все же есть смысл в усреднении точности и полноты для получения одной метрики вместо двух. Решить эту проблему можно с помощью гармонического среднего (F-меры):

|  |  |  |
| --- | --- | --- |
|  |  | (23) |

F-мера является стандартом машинного обучения.

Иногда сталкиваемся с задачами классификации, при которых необходимо оценить не то, к какому классу принадлежит объект, а с какой вероятностью он находится в классе 1:

|  |  |  |
| --- | --- | --- |
|  |  | (24) |

В таком случае оценка качества алгоритма происходит иным способом.

Первый метод основан на PR-кривой (кривой точности и полноты). Для начала необходимо отсортировать объекты по возрастанию оценки b(x) принадлежности к первому классу. Затем перебираем все пороги классификации. Сначала ни один объект не относим к классу 1, затем только первый с максимальной оценкой и далее. Для каждого порога необходимо найти точность и полноту алгоритма. Наносим соответствующую точку на графике в осях «полнота-точность». Соединив точки, мы и получим необходимую Precision-Recall-кривую. Чем больше площадь под графиком, тем лучше работа алгоритма. Данная мера качества для b(x) — AUC-PRC (Area Under Precision-Recall curve).

Второй метод реализуется с использованием ROC-кривой (receiver operating characteristic). По оси x отложена доля ошибочных положительных классификаций (FalsePositiveRate):  (отношение ложных срабатываний к размеру нулевого класса). По оси y отложена доля правильных положительных классификаций (TruePositiveRate):  — чувствительность алгоритма (отношение верных срабатываний к размеру первого класса). Данная мера качества для b(x) — AUC-ROC-кривая, так же является хорошей метрикой.

# **3 Применение метода обнаружения аномалий к задаче аутентификации пользователя смартфона**

Выясним возможность применения методов машинного обучения к задаче аутентификации пользователя мобильного устройства.

## 3.1 Обзор проблемы

Все больше конфиденциальной информации передается по смартфону через обыкновенный звонок. Пользователи сообщают личную или же деловую информацию по телефону. Банки удостоверяются в подлинности транзакции, совершая звонок клиенту на мобильный телефон.

Однако, на ряду с ростом популярности смартфонов и подобных технологий растет так же и рост количества мошенничества. В связи с этим встает вопрос защиты данных, передаваемых по телефону, от несанкционированных пользователей.

Существуют такие средства защиты на мобильный устройствах, как PIN-код, пароль, графический ключ. Их сложно назвать безопасными. Метод перебора всех возможных ключей не является эффективным, так как число комбинаций невероятно велико, но если сторонний пользователь скомпрометирует ввод ключа, у него будет возможность доступа к данным без ведома законного пользователя. Также эти методы трудно назвать удобными. Ввод ключа занимает некоторое время, а также при неверном вводе, что бывает не редко, необходимо начать ввод заново.

Широкое распространение получила технология аутентификации по отпечатку пальца. Отпечаток пальца является одним из составляющих физической биометрии и индивидуальным для каждого пользователя. Таким образом, решена проблема небезопасности методов, рассмотренных выше, но удобство использования остается сомнительным. Аутентификация не всегда проходит с первого раза, в определенные погодные условия это доставляет дополнительные неудобства.

Менее популярные методы — аутентификация по чертам лица или по голосу. Так же являются частью физической биометрии. Обойти защиту этих методов можно с помощью поднесения к камере устройства изображение легитимного пользователя или записи голоса для каждого метода соответственно.

Есть возможность разработать метод аутентификации пользователя смартфона по движению руки при поднесении телефона к уху во время принятия входящего вызова, а также по характерному движению пальца по экрану при разблокировке смартфона пользователем на основе поведенческой биометрии.

Таким образом решается вопрос безопасности, ведь поведенческая биометрия каждого человека уникальна и ее крайне сложно подделать, а так же вопрос удобства, так как от пользователя не потребуется дополнительных действий.

Источниками информации для аутентификации по движению руки при принятии входящего звонка могут являться следующие параметры:

* скорость поднесения к уху телефона;
* ускорение поднесения телефона к уху;
* траектория движения руки с телефоном при поднесении;
* положение телефона относительно уха пользователя.

При аутентификации пользователя по движению пальца по экрану:

* скорость движения пальца по экрану;
* траектория движения пальца по экрану;
* площадь соприкосновения пальца с экраном телефона.

Для получения данной информации нет необходимости в дополнительных датчиках. Они есть во всех современных смартфонах — сенсорный экран, гироскоп, акселерометр.

## 3.2 Роль машинного обучения в задаче аутентификации

Для обработки данных, собранных с датчиков, возможно использование методов машинного обучения.

Обучающей выборкой будут являться экспериментальные данные – до начала пользования технологией многоразовое поднесение телефона к уху или проведение пальцем по экрану, в качестве характерных данных пользователя.

После, активного участия пользователя требоваться не будет, процесс сбора и обработки будет происходить во время пользования.

Метод классификации требует разделение выборки на некоторое количество классов. Первый класс является нормальными явлениями, когда легитимный пользователь использует телефон. Второй же класс включает в себя явления, при которых незаконный владелец берет телефон. В целом, метод подходит для решения задачи аутентификации, но второй класс в таком случае будет крайне мал, более эффективно рассматривать метод обнаружения аномалий.

При использовании смартфона легитимным пользователем, алгоритм определяет это как нормальное явление, так как обучался на явлениях с аналогичными значениями параметров. Если же телефон попадает в руки несанкционированного пользователя, алгоритм относит это явление в класс аномальных, так как значения параметров ему незнакомы.

Рассмотрим существующие методы обнаружения аномалий применительно к задаче аутентификации на основе поведенческой биометрии. Результат представлен в таблице 2.

Таким образом, к задаче могут быть применимы такие методы как восстановление смесей с непараметрическим восстановлением плотности, одноклассовый метод опорных векторов, логистическая регрессия, а также сети адаптивного резонанса и перцептрон.

Таблица 2 — Применимость методов машинного обучения к задаче аутентификации на основе поведенческой биометрии

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Метод обнаружения аномалии | | | Достоинства и недостатки | Применимость |
| Метод, основанный восстановле-нии плотности | Параметрическое восстановление плотности | | Должен быть известен вид функции распределения | Не применим |
| Непараметрическое восстановление плотности | | Вид функции распределения может быть неизвестен, но распределение только одно | Применим (при использовании восстановления смесей) |
| Восстановление смесей | | Должен быть известен вид функции распределения, но распределений может быть несколько | Применим (при использовании непараметричес-кого восстанов-ления плотности) |
| Метод сведения к методу классификации | Линейные классификаторы | Одноклас-совый метод опорных векторов | Один класс. Явления, неподходящие этому классу – нелегитимные пользователи | Применим |
| Логистичес-кая регрессия | Скалярное произведение положительное – нормальное явление. Иначе – ошибка, нелегитимный пользователь | Применим |
| Байесовский классификатор | Наивный байесовский классификатор | Необходима независимость признаков. В задаче аутентификации признаки зависимы | Не применим |
| Метод ближайшего соседа | Необходимо больше одного класса для определения принадлежности объекта классу. | Не применим |
| Нейронные сети | Перцептрон | Устанавливает, какому классу принадлежит объект. Нормальное явление первому классу. Аномальное любому другому. | Применим |

Продолжение таблицы 2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | Сети адаптивного резонанса | Нормальное явление, тогда нейрон считается нейроном-победителем. Если аномалия – прекращает поиск | Применим |
| Сеть радиально-базисных функций | Необходимо гауссовское распределение | Не применим |

Принцип использования метода восстановления смесей с непараметрическим восстановлением плотности изображен на рисунке 3.1.

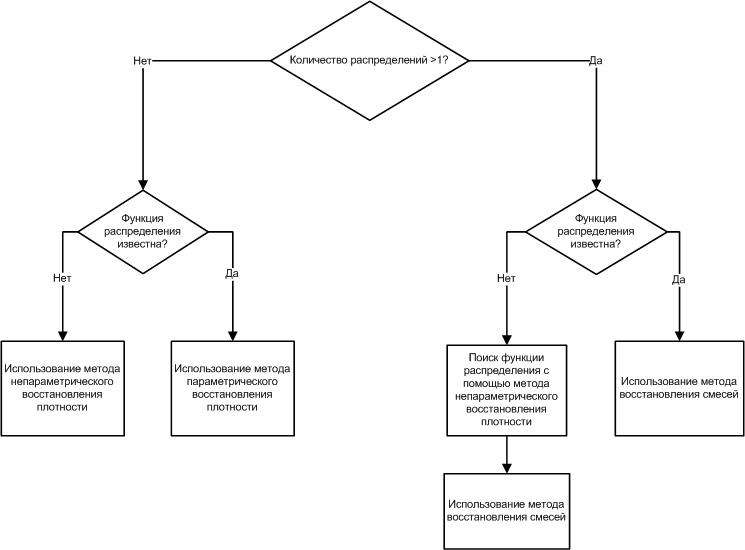


Рисунок 3.1— Блок-схема использования метода восстановления смесей с непараметрическим восстановлением плотности

Для задачи аутентификации пользователя мобильного устройства наиболее важным параметром является точность, нежели полнота, так как важно, чтобы нарушитель не был пропущен. Поэтому, чтобы усилить вес этого параметра, добавим коэффициент в F-меру к параметру точности:

|  |  |  |
| --- | --- | --- |
|  |  | (25) |

# **ЗАКЛЮЧЕНИЕ**

В настоящей работе был проведен анализ методов машинного обучения, применимых к задаче аутентификации пользователей мобильных устройств.

Исследованы ключевые характеристики основных методов машинного обучения, выявлены их достоинства и недостатки.

Проанализированы методы машинного обучения, применимые к задаче аутентификации. На основе них выявлены несколько подходящих методов: восстановление смесей с непараметрическим восстановлением плотности, одноклассовый метод опорных векторов, логистическая регрессия, а также сети адаптивного резонанса и перцептрон.

Целью дальнейшей научно-исследовательской работы станет применение метода обнаружения аномалий к задаче аутентификации.

# **СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ**

1. **Samuel, A.L.** Some Studies in Machine Learning Using the Game of Checkers / A.L. Samuel // IBM Journal. — 1959. — С. 535-554.
2. **Mitchell, T.** MachineLearning. / Mitchell, T. // McGraw-Hill Science/Engineering/Math. — 1997. — 414 с.
3. Введение в машинное обучение. Просто о сложном [Электронный ресурс]. — Режим доступа: <http://savepearlharbor.com/?p=163675> (дата обращения: 13.12.2016).
4. **Золотых Н. Ю.** Машинное обучение и анализ данных. Учебное пособие. / Золотых Н. Ю. // 2016. — 125 с.
5. Выбор алгоритмов машинного обучения [Электронный ресурс]. — Режим доступа: <https://docs.microsoft.com/ru-ru/azure/machine-learning/machine-learning-algorithm-choice> (дата обращения: 12.01.2017)
6. Машинное обучение [Электронный ресурс]. — Режим доступа: [http://www.machinelearning.ru](http://www.machinelearning.ru/) (дата обращения: 13.01.2017).
7. Проблема переобучения модели и метод регуляризации [Электронный ресурс]. — Режим доступа: [http://wiki.technicalvision.ru](http://wiki.technicalvision.ru/) (дата обращения: 13.12.2016).
8. **Arindam B.** Anomaly Detection : A Survey / B. Arindam, C. Varun, K. Virin // Springer Verlak Berlin Heidelberg 2009. — 72 с.
9. **Суханов, А. В.** Интеллектуальные методы обнаружения и прогнозирования аномальных событий в темпоральных данных.: дис. канд. техн. наук: 05.13.01 : защищена 18.02.16 / А. В. Суханов. — М., 2016 – 159 с.
10. **Кузовлев, В. И.** Метод выявления аномалий в исходных данных при построении прогнозной модели решающего дерева в системах поддержки принятия решений / В. И. Кузовлев, , А. О. Орлов**. —** Наука и образование: научное издание МГТУ им. Н.Э. Баумана. 2012. №09. // 2012. – С. 209-218.
11. **Будынков, А. Н.** Проблема выбора ядра в методе опорных векторов. / А. Н. Будынков, О. А. Коршунова // Труды X Международной конференции «Идентификация систем и задачи управления» SICPRO ‘15 // 2015. – С. 220-229.
12. **Хайкин, С.** Нейронные сети: Полный курс. / С. Хайкин // 2006. – 1103 с.
13. **Розенблатт, Ф.** Принципы нейродинамики: Перцептроны и теория механизмов мозга. / Ф. Розенблатт // 1965. – 478 с.
14. **Брюхомицкий, Ю. А.** [Нейросетевые модели для систем информационной безопасности: Учебное пособие](http://window.edu.ru/window_catalog/redir?id=28836&file=tsure062.pdf). / Ю. А. Брюхомицкий, // 2005. – 160 с.
15. **Дударов, С. П.** Портал искусственного интеллекта. [Электронный ресурс]. — Режим доступа: <http://neuronus.com/> (дата обращения: 13.01.2017).
16. Нейронные сети с радиальными базисными функциями. [Электронный ресурс]. — Режим доступа: <http://libraryno.ru/2-9-neyronnye-seti-s-radial-nymi-bazisnymi-funkciyami-iis/> (дата обращения: 13.01.2017).