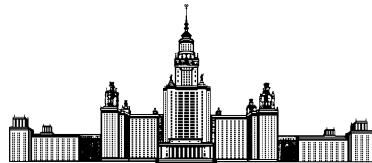


Московский государственный университет имени М. В. Ломоносова



Факультет Вычислительной Математики и Кибернетики

Кафедра Математических Методов Прогнозирования

**КУРСОВАЯ РАБОТА СТУДЕНТА 317 ГРУППЫ**

**«МЕТОДЫ ДЕТЕКТИРОВАНИЯ АНОМАЛИЙ»**

Выполнил:

студент 3 курса 317 группы

*Иванов Сергей Максимович*

Научный руководитель:

д.ф-м.н., профессор

*Дьяконов Александр Геннадьевич*

Москва, 2017

# Содержание

<b>1 Введение</b>	<b>3</b>
1.1 Понятие аномальности . . . . .	3
1.2 Постановка задачи . . . . .	4
<b>2 Подходы и методы решения</b>	<b>6</b>
2.1 Вероятностный подход . . . . .	6
2.2 Линейные методы . . . . .	7
2.3 Метрические методы . . . . .	10
2.4 Гармоническое и полиномиальное оценивание . . . . .	12
2.5 LOF (Local Outlier Factor) . . . . .	13
2.6 Isolation Forest . . . . .	15
<b>3 Методы улучшения алгоритмов</b>	<b>17</b>
3.1 Сэмплирование . . . . .	17
3.2 Случайное масштабирование . . . . .	19
3.3 Rotated Bagging . . . . .	20
3.4 Ансамблирование . . . . .	21
3.5 Итеративный отбор . . . . .	22
<b>4 Вычислительные эксперименты</b>	<b>23</b>
4.1 Исходные данные . . . . .	23
4.2 Сравниваемые методы и использованные реализации . . . . .	26
4.3 Сравнение модификаций . . . . .	27
4.4 Сравнение алгоритмов . . . . .	30
<b>5 Заключение</b>	<b>33</b>
<b>Список литературы</b>	<b>35</b>

## **Аннотация**

Выделение в данных нетипичных, аномальных представителей является особой задачей в машинном обучении. Помимо ряда практических применений (обнаружение сбоев в показаниях датчиков, хакерских атак, необычных результатов диагностики), эту задачу можно считать этапом построения любого алгоритма машинного обучения, на котором данные проверяются на консистентность, очищаются от выбросов и шума.

В работе приведён обзор существующих алгоритмов, идей и подходов к решению задачи, а также практические методы их модификаций. Также приводятся результаты эксперимента по сравнению алгоритмов и модификаций на ряде разнородных датасетов.

# 1 Введение

## 1.1 Понятие аномальности

Задачи детектирования аномалий не имеют единой формулировки и зачастую интерпретируются по-разному в зависимости от характера данных и поставленной цели [1, 3, 5]. На интуитивном уровне аномалиями называют то, что не вписывается в общие правила и законы, справедливые для представленных данных. Такое определение нуждается в формальном уточнении, прежде чем решать задачу математическими методами.

Любые методы детектирования аномалий опираются на некоторое строгое представление того, что означает «отклонение от нормы». В более частных случаях, уже это представление может отталкиваться от контекста задачи и априорной информации. Например, в простейшем случае, если данные представляют собой набор элементов некоторого множества, любой элемент, не принадлежащий этому множеству, априори можно полагать аномалией. Такое «характеристическое» множество, по принадлежности которому можно делать вывод об аномальности элемента, однако, существует не всегда. Например, если объектами являются показания некоторого датчика, а аномалиями – показания сломанного датчика, может случиться так, что и сломанный, и правильно работающий датчики выдадут один и тот же результат.

Существенно разнится и природа появления аномалий в данных. Это может быть как шум, то есть данные, образовавшиеся чаще всего случайным образом, так и редкие явления, представляющие интерес и требующие дополнительного изучения (например, появление посторонних объектов на снимках). В последнем случае аномалии имеют особую природу и могут даже выделяться в обособленные кластера (рис. 1). Подходы к обнаружению аномалий различных «видов» могут существенно отличаться.

Поскольку разные алгоритмы распознавания аномалий исходят от разных определений аномальности, различается не только их результат, но и подход. Возможно и обратное: предлагаемый подход определяет интерпретацию аномальности. Например, если в качестве данных взять набор чисел, то можно пред-

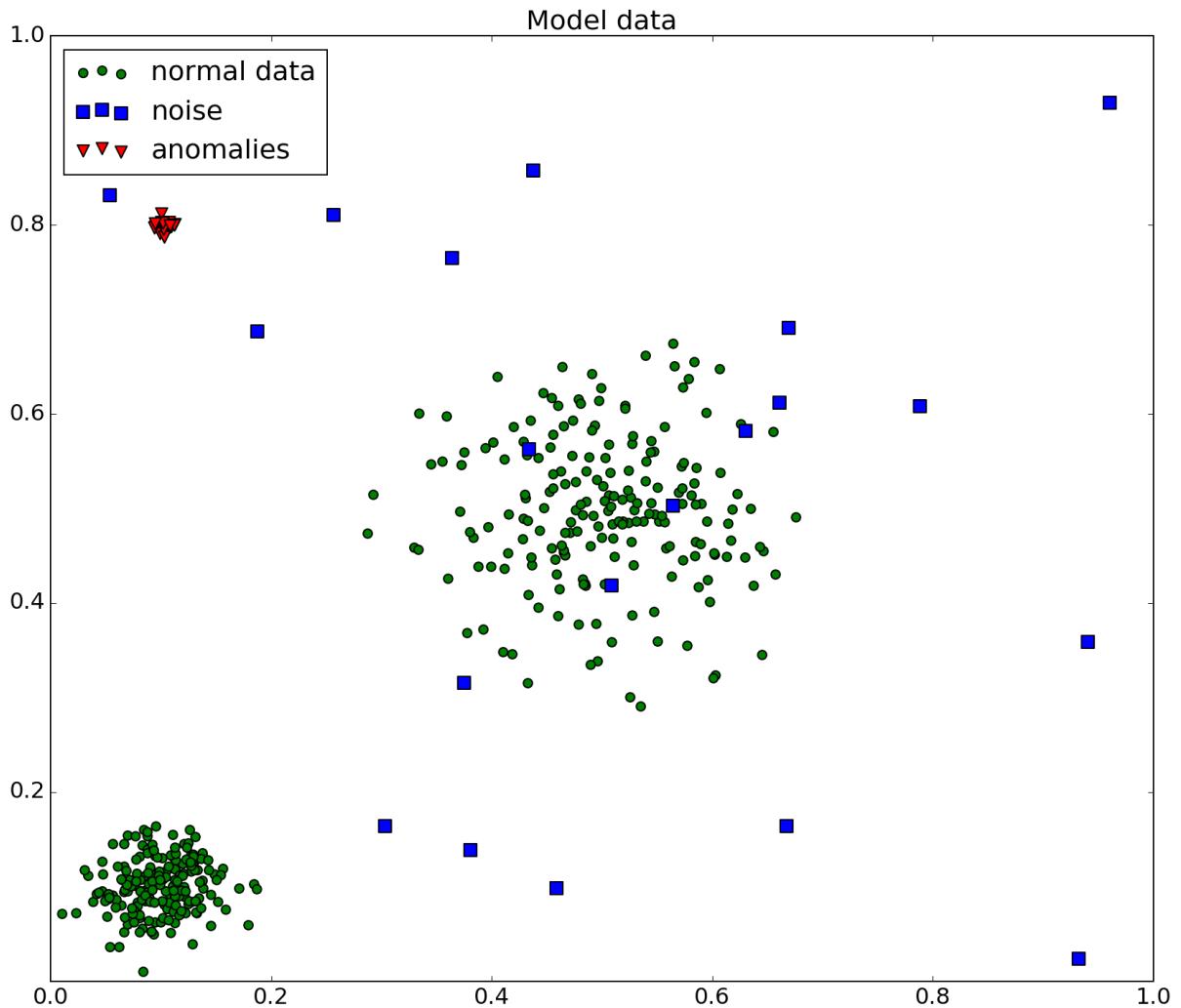


Рис. 1: Пример модельной задачи. Нормальные данные представляют собой два кластера, сэмплированных из нормального распределения. Аномалии образуют отдельный кластер малого размера. Вдобавок, имеется шум, сэмплированный из равномерного распределения

положить, что удаление из него аномалий понизит дисперсию; затем можно найти в наборе подмножество чисел (например, заданного размера), удаление которых максимально понизит дисперсию, и объявить эти точки аномалиями.

## 1.2 Постановка задачи

В дальнейшем будем полагать, что данные имеют признаковое представление, то есть каждый объект  $x$  задан в виде некоторого вектора из  $\mathbb{R}^d$ . В классической постановке задача детектирования аномалий формулируется так: в заданном множестве  $X$  для каждого элемента выдать 0, если этот объект относится

к классу нормальных данных, и 1, если этот объект аномален. Такая задача относится к классу задач обучения без учителя, поскольку правильных ответов на части входных данных не предоставляется.

В аналогичной задаче обучения с учителем на некоторой части  $X_{train}$  входных данных известен верный ответ, то есть для каждого объекта  $x \in X_{train}$  известны метки  $y(x) \in \{0, 1\}$  – является ли данный объект аномалией. Задача выдачи меток для новых данных,  $X_{test}$ , формально является задачей бинарной классификации, и, значит, может решаться при помощи любых алгоритмов машинного обучения с учителем. Однако, возможен и «промежуточный вариант», когда все метки  $y(x), x \in X_{train}$  равны 0, то есть заданы примеры только нормальных («проверенных», «хороших») данных. В таком особом случае алгоритмы решения задачи бинарной классификации будут выдавать нерелевантный константный прогноз.

Стоит отметить, что эта проблема, пусть в меньшей степени, присутствует и в случае, когда в обучающей выборке есть примеры аномалий: алгоритмы обучения с учителем на данных, не имеющих аналогов в обучении, выдают в общем случае случайный ответ (причём часто, единица выдаётся с вероятностью  $p(y(x) = 1)$ , оценённой по обучающей выборке). По этой причине имеет смысл рассматривать задачу именно как задачу обучения без учителя, а наличие обучающей выборки (меток) – как возможность дополнительно настраивать параметры алгоритма.

Практически все алгоритмы детектирования аномалий сводятся к построению некоторой функции  $anomaly\_score(x)$ , которая по данному объекту выдаёт некоторый «рейтинг» аномальности. После этого разделение на класс аномалий и класс нормальных данных производится бинаризацией по некоторому поро-

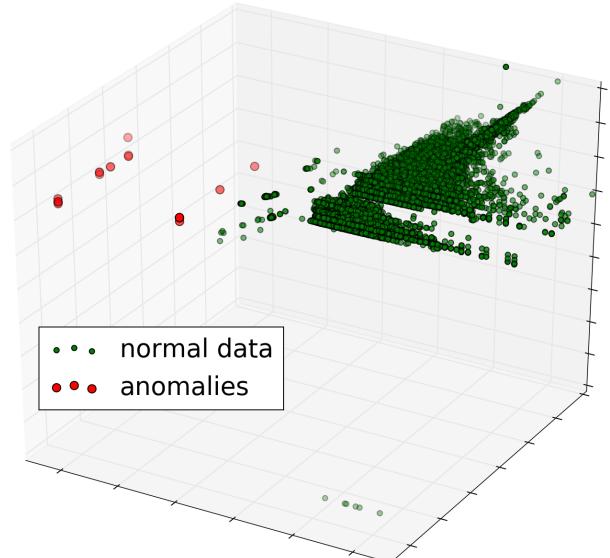


Рис. 2: Данные датасета Smtp из репозитория [12]

гу, выбор которого является особым этапом решения задачи. В отсутствии меток или априорной информации, единственной имеющейся информацией является одномерное распределение значений *anomaly\_score* на имеющихся данных, чего для обоснованного выбора недостаточно. Чаще всего, известна приблизительная доля аномалий в данных; в таких случаях в качестве порога выбирается соответствующий квантиль.

## 2 Подходы и методы решения

### 2.1 Вероятностный подход

В генеративном подходе предлагается подобрать вероятностную модель распределения, из которого были сэмплированы нормальные данные, то есть найти плотность распределения  $p(x)$ . Аномалиями при этом являются объекты, имеющие низкое правдоподобие, то есть в качестве *anomaly\_score* выступает сама функция  $p$ .

Однако, такой подход сталкивается с принципиальными трудностями. Для построения  $p(x)$  требуется решить задачу вида

$$\prod_{x \in X_{norm}} p(x, \theta) \rightarrow \max_{\theta},$$

где  $X_{norm}$  – нормальные данные предоставленной выборки,  $\{p(x, \theta) \mid \theta \in \Theta\}$  – семейство плотностей распределений, параметризованные  $\theta$ . Если метки отсутствуют, эта задача вынужденно заменяется на другую:

$$\prod_{x \in X} p(x, \theta) \rightarrow \max_{\theta}, \tag{1}$$

где  $X$  – все имеющиеся данные, включая аномалии. Эти две задачи неэквивалентны, и имеют существенно различное решение, особенно при высокой доли аномалий в данных. Для конкретных семейств распределений существуют методы повышения робастности, теоретически позволяющие преодолеть эту проблему (например, вместо оценки мат. ожидания средним оценивать его медианой, устойчивой к выбросам) [7].

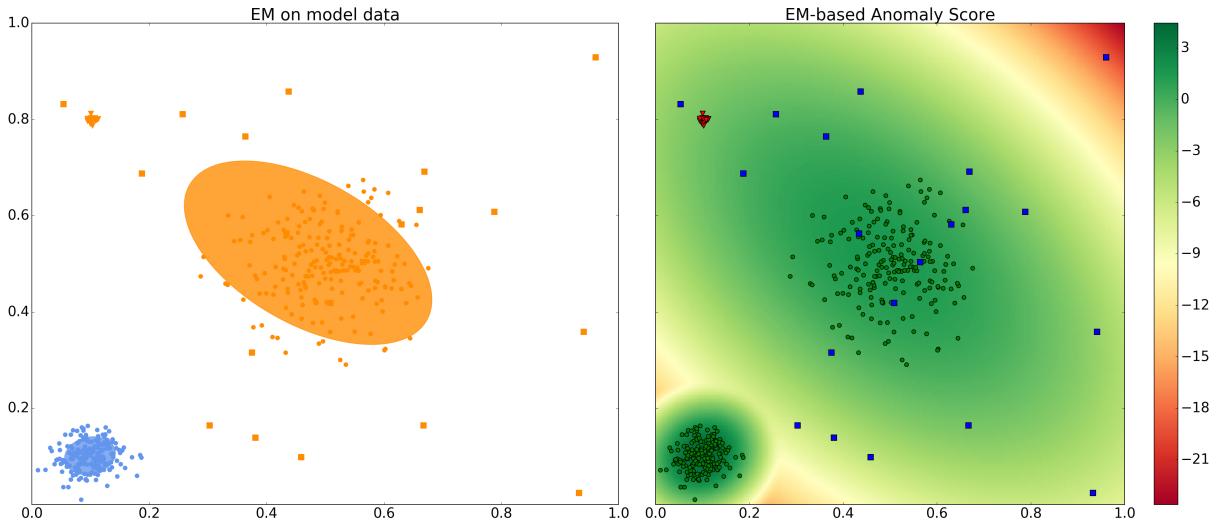


Рис. 3: Пример применения ЕМ-алгоритма [8] на модельной задаче. На рисунке справа приведена карта значений *anomaly\_score*

Тем не менее, на практике метод плохо применим: сложно как проверить полученную модель на адекватность, так и убедиться в правильном выборе семейства распределений. Это связано с тем, что низкое значение функционала (1) может означать как неудачное моделирование, так и вырождено низкое значение правдоподобия для аномальных объектов, что наоборот является успехом. Отличить одно от другого чаще всего затруднительно. В результате, в случае обучения без учителя, вероятностное моделирование может применяться только с опорой на априорную информацию, потому что иначе выдаваемый алгоритмом результат безоснователен.

## 2.2 Линейные методы

Базовая идея линейных методов заключаются в построении такой линейной модели, значительные отклонения от которой будут характеризовать аномалии. Нелинейная природа данных на практике приводит к тому, что требуется либо строить ряд таких моделей и некоторым образом усреднять отклонения, либо применять ядерный переход. Главное предположение линейных методов заключается в том, что нормальные данные располагаются в подпространстве пространства признаков  $\mathbb{R}^d$  размерности меньше  $d$  [1].

Одним из возможных алгоритмов является следующий: объявить  $i$ -ый признак выборки целевой переменной и решать задачу линейной регрессией на основе оставшихся признаков. Отклонение прогноза от истины полагается значением  $anomaly\_score_i$ . Итоговым ответом объявляется усреднение этого результата по всем признакам. Такой алгоритм предполагает найти некоторую линейную зависимость между признаками, которая будет нарушаться для аномалий.

Обобщением этого метода является метод, основанный на свойствах главных компонент  $e_1, e_2 \dots e_d$  — нормированных собственных векторов матрицы  $X^T X$ , соответствующих собственным значениям  $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_d$ .

**Теорема 1.** [9] Главные компоненты, соответствующие  $k$  наибольшим собственным значениям, образуют базис  $k$ -мерного подпространства, проекция на которое наилучшим (по норме Фробениуса) образом аппроксимирует исходные данные.

Теорема (1) позволяет найти подпространства заданной размерности  $k$ , расстояние от точки до которых можно интерпретировать как степень её аномальности (рис. 4). Существенным недостатком алгоритма является необходимость выбора  $k$ ; слишком точная аппроксимация не сможет уловить отклонения, а в слишком грубой аппроксимации станут неотличимы от части нормальных данных.

Более тонкий алгоритм можно получить на основе другого свойства главных компонент:

**Теорема 2.** [9] Дисперсия строк матрицы  $X$  вдоль оси  $e_i$  равна  $\lambda_i$ .

Эта теорема означает, что координаты точек в базисе главных компонент имеют тем меньший разброс, чем меньше значение соответствующего собственного значения. Практическое значение теоремы в следующем: можно для  $i$ -го собственного вектора посчитать отклонения координаты точки  $x$  вдоль оси  $e_i$  от

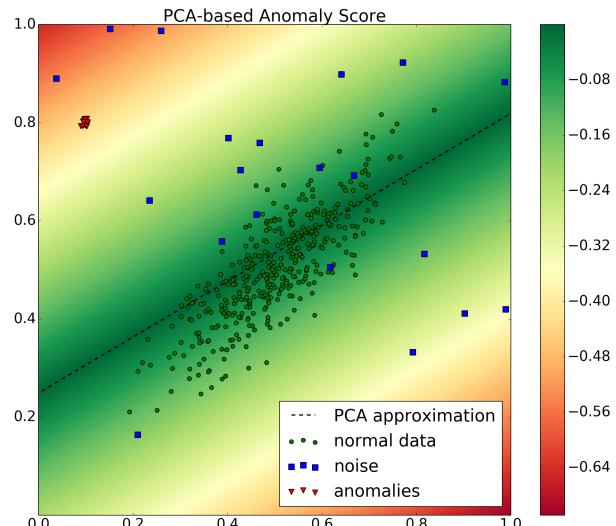


Рис. 4: Расстояние до одномерного аппроксимирующего подпространства

среднего значения, получив тем самым некоторую характеристику аномальности точки. Такая характеристика, как и в случае с расстоянием до аппроксимирующих подпространств, имеет низкую обобщающую способность (рис. 5).

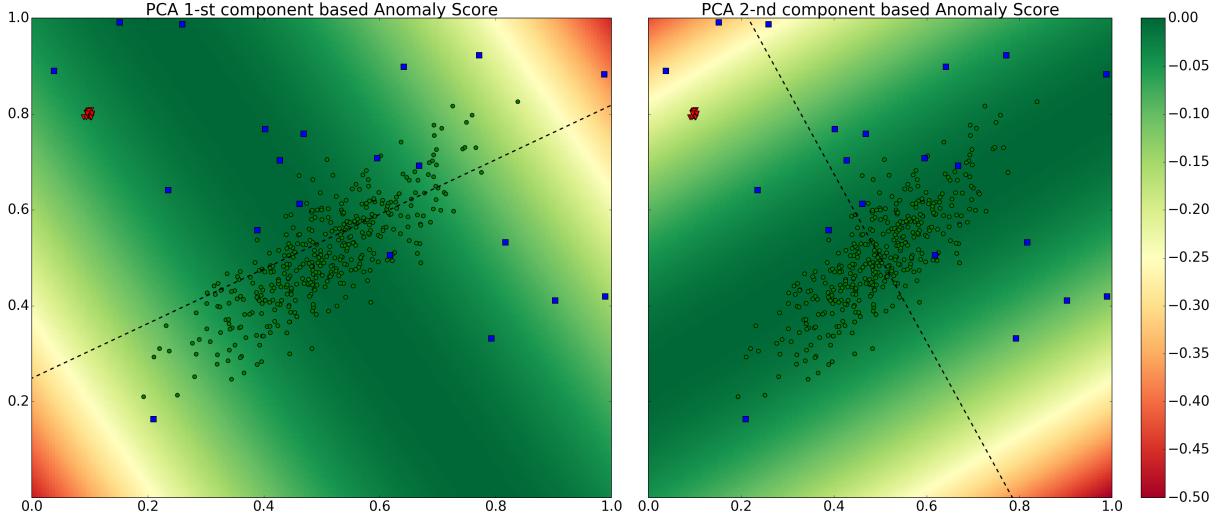


Рис. 5: Построение *anomaly\_score* индивидуально вдоль каждой из главных компонент

Однако теорема (2) предоставляет способ усреднить её значение по всем собственным векторам [1]:

$$anomaly\_score(x) = \sum_{i=1}^d \frac{\langle x - \mu, e_i \rangle^2}{\lambda_i}, \quad (2)$$

где  $\mu$  – средний вектор по выборке. В выражении (2) существенно, что в знаменателе стоит  $\lambda_i$  – именно отклонения вдоль главных компонент, соответствующих малым собственным значениям, являются характерными для аномалий. На самом деле, вычисление рейтинга аномальности по формуле (2) эквивалентно нахождению расстояния Махalanобиса [10] до центроида выборки. Пример результата работы алгоритма приведён на рис. 6.

Алгоритмы, основанные на линейных методах и, в частности, на свойствах главных компонент, на практике применимы только при наличии в данных каких-то линейных зависимостей или трендов. Если подпространство, в котором сосредоточены нормальные данные, менее тривиально, то определить вид этого подпространства становится нелегко (рис. 7). Чтобы обойти эту проблему, можно применить ядерный переход и перейти в пространство большей размерности, где отделение аномалий и нормальных данных будет возможно гиперплоскостью.

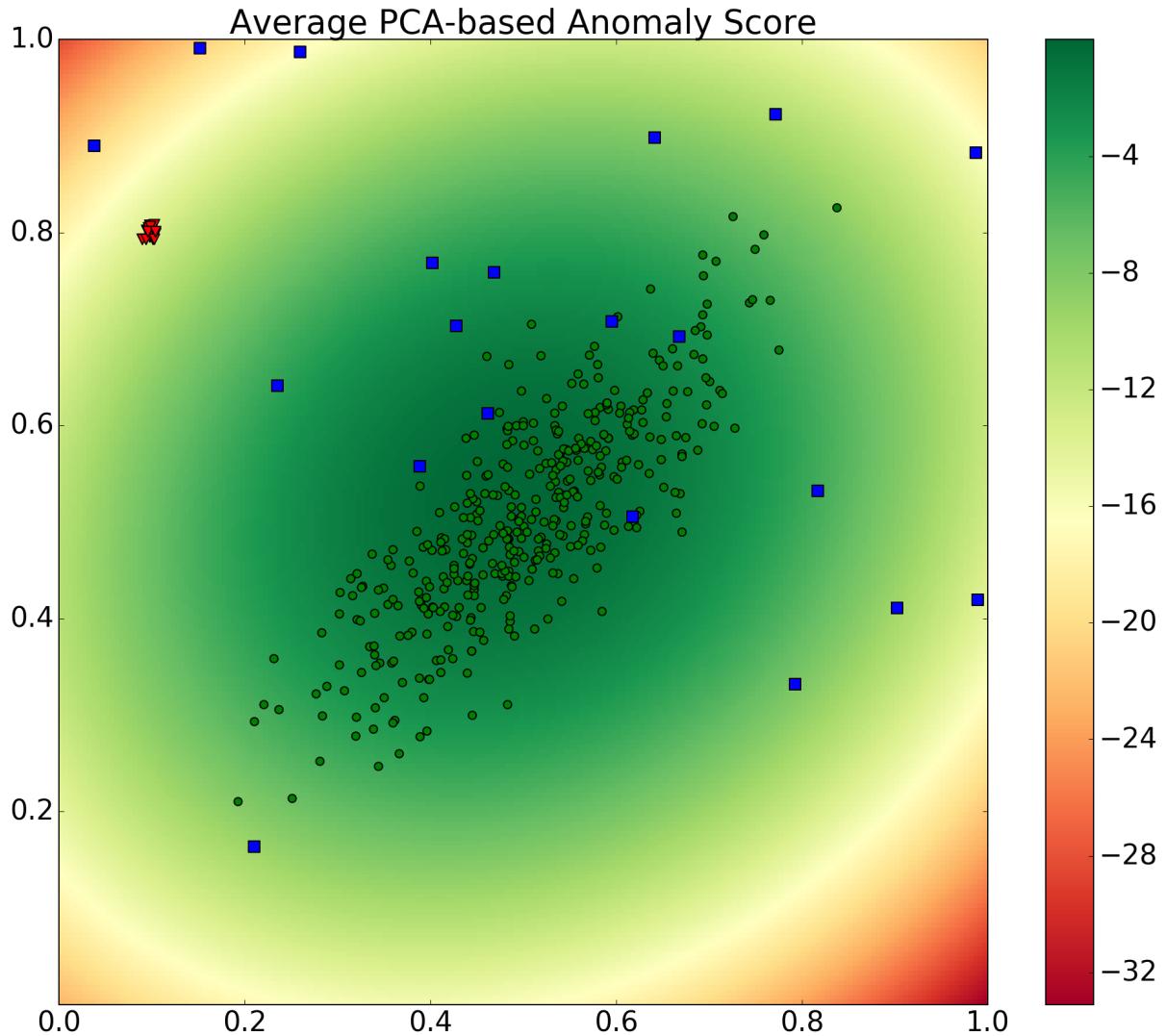


Рис. 6: Усреднённое по формуле (2)  $anomaly\_score$

## 2.3 Метрические методы

Метрические методы пытаются найти в данных точки, в некотором смысле изолированные от остальных [4, 5]. Если в пространстве задана некоторая метрика  $\rho(x_1, x_2)$ , то можно вводить следующие представления аномальности:

- Аномалии – точки, не попадающие ни в один кластер. К данным применяется один из алгоритмов кластеризации; размер кластера, в котором оказалась точка, объявляется её  $anomaly\_score$ .
- Локальная плотность в аномальных точках низкая. Для данной точки  $anomaly\_score$  объявляется локальная плотность, которая оценивается неко-

торым непараметрическим способом (например, ядерной оценкой плотности Розенблата — Парзена).

- Расстояние от данной точки до ближайших соседей велико. В качестве *anomaly\_score* может выступать:
  1. расстояние до  $k$ -го ближайшего соседа;
  2. среднее расстояние до  $k$  ближайших соседей;
  3. медиана расстояний до  $k$  ближайших соседей;
  4. гармоническое среднее до  $k$  ближайших соседей;
  5. доля из  $k$  ближайших соседей, для которых данная точка является не более чем  $k$ -ым соседом.

Использование подобных алгоритмов с параметрами (например,  $k$ ) на-кладывает требование наличия априорной информации о потенциальных размерах кластеров аномалий (рис. 8). Такие кластера могут остаться не выявленными, если алгоритм рассматривает малое число ближайших соседей; при большом же числе минимальное значение построенной функции может оказаться вдали от подпространства нормальных данных (рис. 7).

Главное достоинство метрических методов – интерпретируемость. Можно сказать, что именно метрическими методами задачу нахождения аномалий решает человек. Главным же недостатком является необходимость подбора метрики. На практике в данных интерпретируемая метрика отсутствует; она аппроксимируется стандартными метриками – например, манхэттоновской, евклидовой или чебышёвской. Выбор метрики в отсутствии меток может оказаться выбором слепым, а аппроксимация даже при наилучшем подборе – грубой.

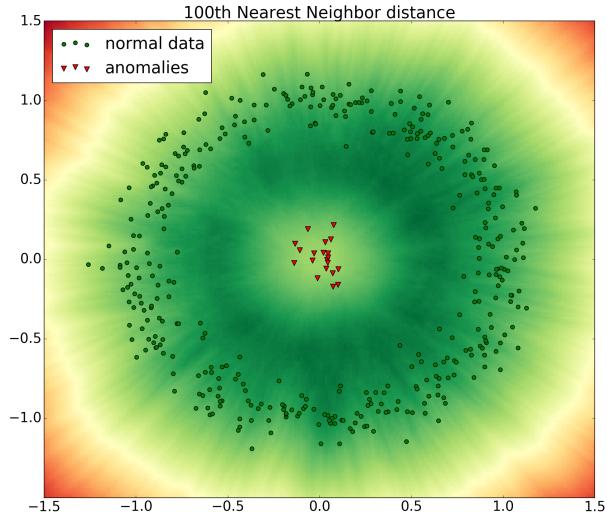


Рис. 7: Пример задачи с нетривиальным подпространством нормальных данных, в котором аномалии расположены в центре масс.

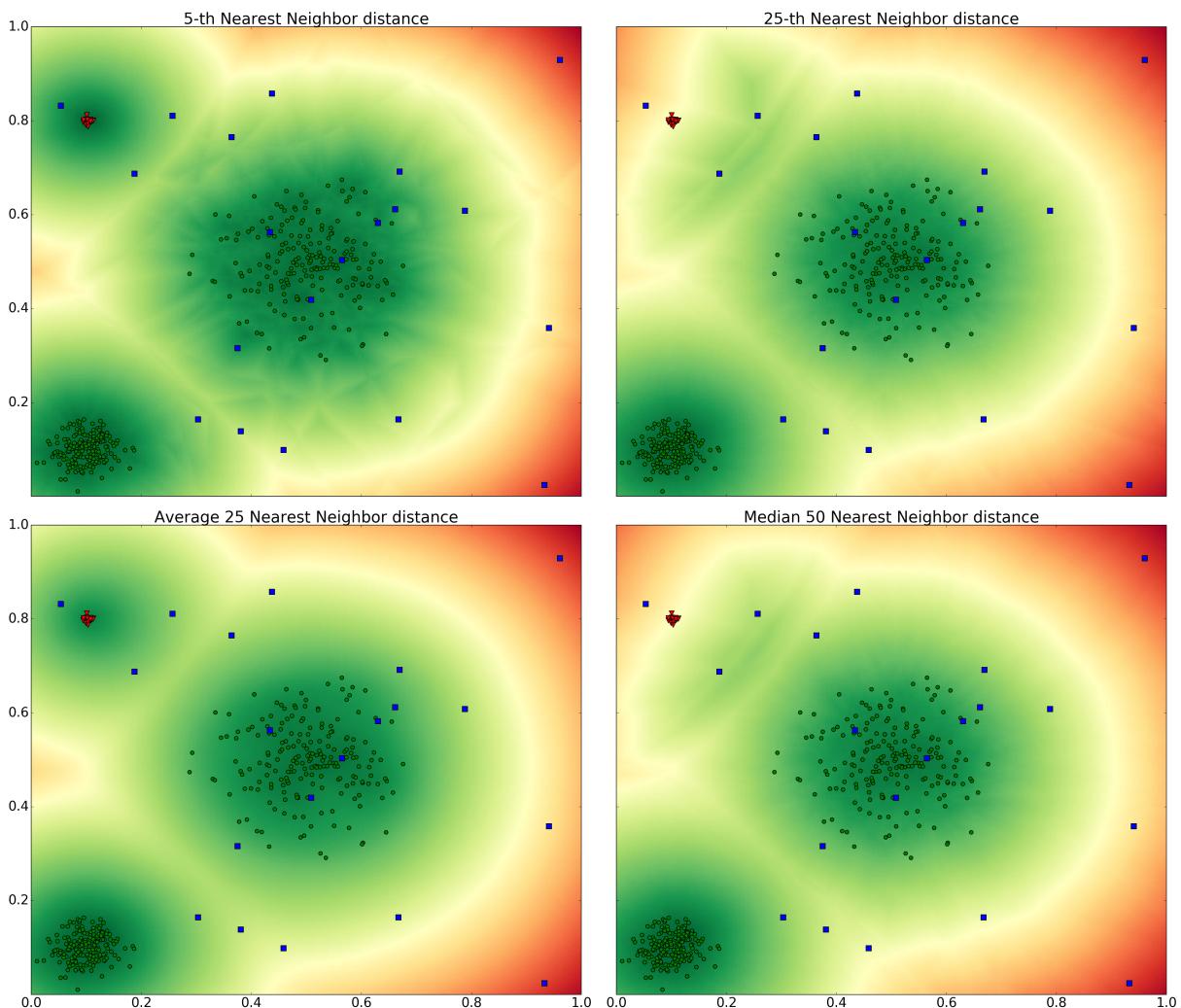


Рис. 8: Примеры работы метрических алгоритмов

## 2.4 Гармоническое и полиномиальное оценивание

Предположим, что в исходных данных  $X$  имеются только нормальные данные, то есть аномалии отсутствуют. В рамках метрического подхода рассмотрим требования, которые разумно наложить на функцию  $\text{anomaly\_score}(x)$ : положим, что это непрерывная функция, которая принимает значение 0 в точке  $x$ , когда про эту точку точно известно, что она нормальна. Иначе говоря,  $\text{anomaly\_score}(x) = 0 \Leftrightarrow x \in X$ . Всего таких точек должно быть столько же, сколько и данных в имеющейся выборке –  $n$ .

Таким требованиям удовлетворяет гармоническая функция

$$anomaly\_score(x) = \frac{n}{\sum_{y \in X} \frac{1}{\rho(x,y)}} \quad (3)$$

и полиномиальная функция:

$$anomaly\_score(x) = \sqrt[n]{\prod_{y \in X} \rho(x,y)} \quad (4)$$

Эти функции полезны тем, что выдают для новой точки оценку того, насколько «близка» она к точкам выборки. Полиномиальная функция по сравнению с гармонической более «гладкая» (рис. 9) и на изолированных точках выборки выдаёт малые значения в меньшей окрестности по сравнению с гармонической, что позволяет использовать её и в случае, если доля аномалий мала.

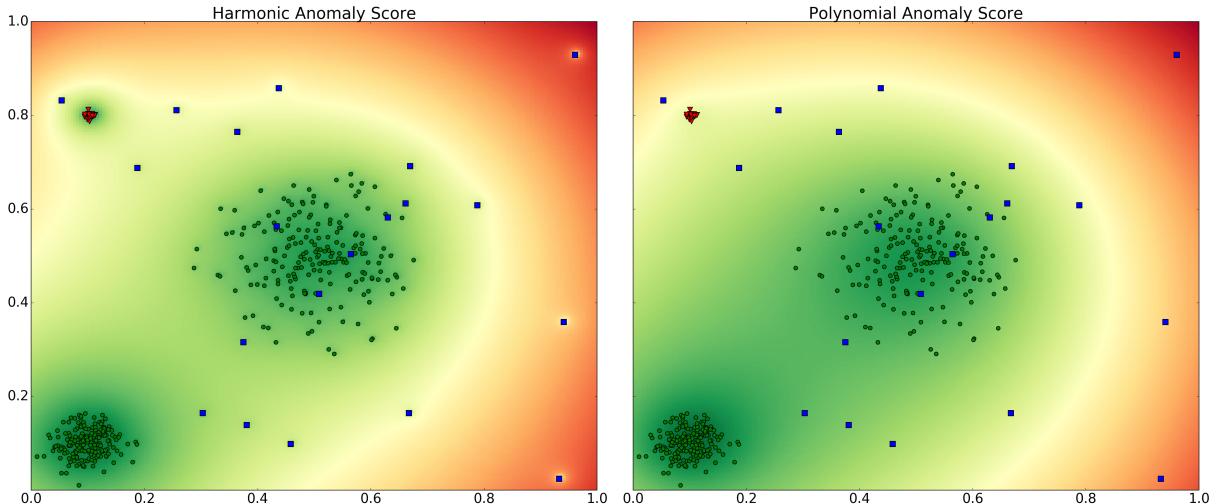


Рис. 9: Примеры работы гармонической и полиномиальной функции

## 2.5 LOF (Local Outlier Factor)

Более тонкой проблемой метрических методов является тот факт, что все лежащие в их основе предположения справедливы лишь в дополнении друг с другом: так, локальная плотность точки, лежащей в центре небольшого кластера аномалий, может оказаться выше, чем для любой точки из большого кластера нормальных данных. Возможно и обратное: изолированная точка-аномалия может располагаться, например, в центре масс кластера нормалий, и тогда среднее расстояние от неё до соседей будет меньше, чем для нормальных точек. Это

«свойство» метрических алгоритмов пытается учесть алгоритм LOF (Local Outlier Factor) [6].

**Определение 2.1.** Пусть  $D_k(y)$  – расстояние от точки  $y$  до  $k$  ближайшего соседа. Досягаемостью (reachability distance) точки  $x$  относительно точки  $y$  называется величина

$$R_k(x, y) = \max(\rho(x, y), D_k(y)) \quad (5)$$

**Определение 2.2.** Пусть  $AR_k(x)$  – средняя досягаемость точки  $x$  относительно  $k$  своих ближайших соседей,  $N_k(x)$  – множество  $k$  ближайших соседей  $x$ . Тогда:

$$LOF_k(x) = \underset{y \in N_k(x)}{\text{mean}} \frac{AR_k(x)}{AR_k(y)} \quad (6)$$

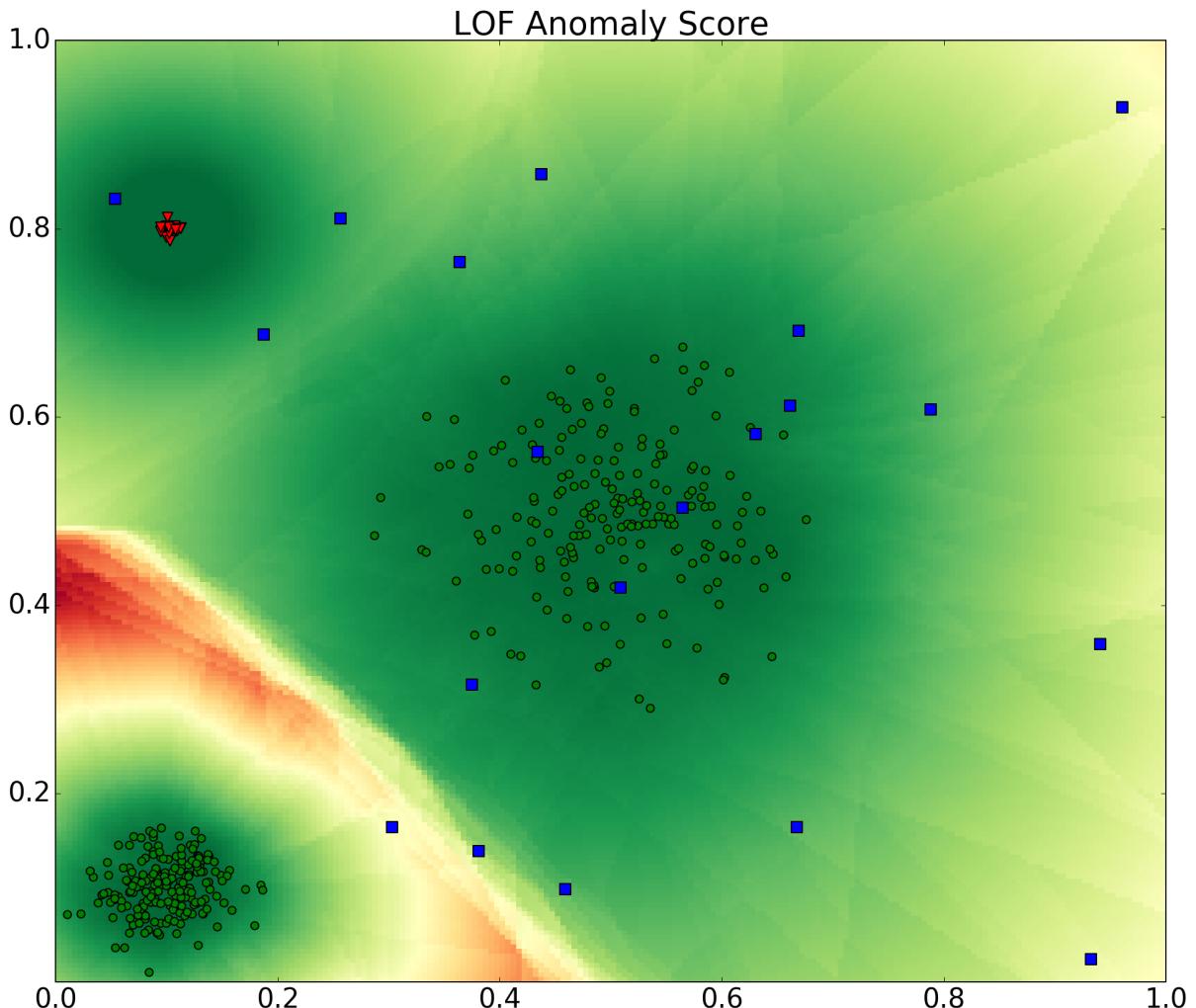


Рис. 10: Пример работы Local Outlier Factor

Интуиция формулы (6) заключается в том, чтобы сравнить среднюю досягаемость точки и её ближайших соседей. Для представителей нормальных данных верно не только, что оценка (5) локальной плотности мала, но и что она незначительно отличается от такой же оценки для ближайших соседей. Пример работы алгоритма приведён на рис. 10.

## 2.6 Isolation Forest

Идея изолирующего леса (Isolation Forest) [2] основана на принципе Монте-Карло: проводится случайное разбиение пространства признаков, такое что в среднем изолированные точки отсекаются от нормальных, кластеризованных данных. Окончательный результат усредняется по нескольким запускам стохастического алгоритма.

Алгоритм изолирующего дерева (Isolation Tree) заключается в построении случайного бинарного решающего дерева. Корнем дерева является всё пространство признаков; в очередном узле выбирается случайный признак и случайный порог разбиения, сэмплированный из равномерного распределения на отрезке от минимального до максимального значения выбранного признака. Критерием останова является тождественное совпадение всех объектов в узле, то есть решающее дерево строится полностью. Ответом в листе, который также соответствует *anomaly\_score* алгоритма, является глубина листа в построенном дереве (рис. 11).

Утверждается, что аномальным точкам свойственно оказываться в листьях с низкой глубиной, то есть в листьях, близким к корню, когда же для разбиения гиперплоскостями кластера нормальных данных дереву потребуется построить ещё несколько уровней. При этом количество таких уровней пропорционально размеру кластера; следовательно, пропорционально и *anomaly\_score* для лежащих в нём точек. Это означает, что объекты из кластеров малых размеров, которые потенциально являются аномалиями, будут иметь *anomaly\_score* ниже, чем из кластеров нормальных данных.

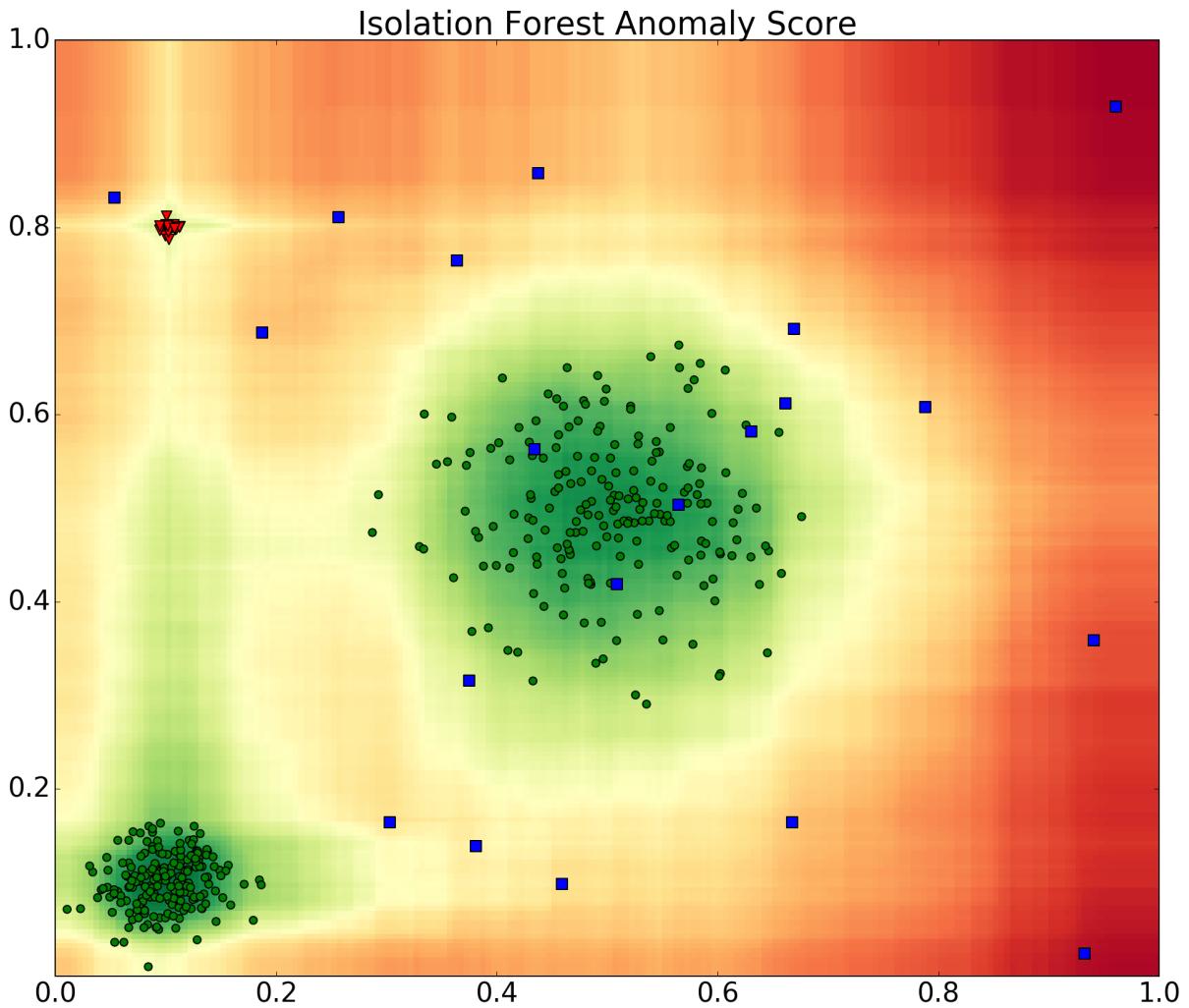


Рис. 11: Пример работы Isolation Forest

Алгоритм обладает рядом существенных преимуществ:

- Алгоритм распознаёт аномалии различных видов: как изолированные точки с низкой локальной плотностью, так и кластеры аномалий малых размеров.
- Сложность изолирующего дерева –  $O(n \log n)$ , что эффективнее большинства других алгоритмов.
- Не требует существенных затрат по памяти, в отличие от, например, метрических методов, зачастую требующих построения матрицы попарных расстояний.
- Отсутствуют параметры, требующие подбора.

- Инвариантен к масштабированию признаков; не требует задания метрики или другой априорной информации об устройстве данных.
- Устойчив к проклятию размерности.

## 3 Методы улучшения алгоритмов

### 3.1 Сэмплирование

Большинство алгоритмов распознавания аномалий успешно работают на выборках малых размеров. Идея сэмплирования заключается в том, чтобы запустить алгоритм на нескольких случайных подвыборках и усреднить результат. Размер случайных подвыборок может быть как фиксированный, так и случайный из некоторого диапазона, но чаще всего он на порядки меньше размера исходной выборки. Интуиция такого выбора заключается в том, что шумовые объекты попадут в подвыборку с низкой вероятностью; кластера нормальных данных будут представлены несколькими представителями, а кластера аномалий выродятся в изолированные точки (рис. 12). Обычно на таких подвыборках алгоритмы строят функцию  $anomaly\_score(x)$ , не сильно уступающую результату, полученному по всем исходным данным, даже несмотря на то, что в подвыборках может потеряться глобальная структура пространства признаков. Усреднения по небольшому числу запусков, обычно порядка 100, на практике хватает для решения этой проблемы.

Помимо значительной оптимизации вычислительной сложности, сэмплирование уменьшает вероятность «подгона» результата алгоритма под конкретные имеющиеся данные. В силу особенностей задачи, необходимое условие отсутствия параметризации алгоритмов зачастую означает их детерминированность (в отсутствие стохастичности  $anomaly\_score$  однозначно определяется по заданной выборке). Это значительно в том случае, если предполагается использовать полученную  $anomaly\_score$  на новых данных. В общем случае, при появлении таких новых данных, любой алгоритм обучения без учителя можно запустить заново на объединении всей имеющейся информации; в случае, если используется

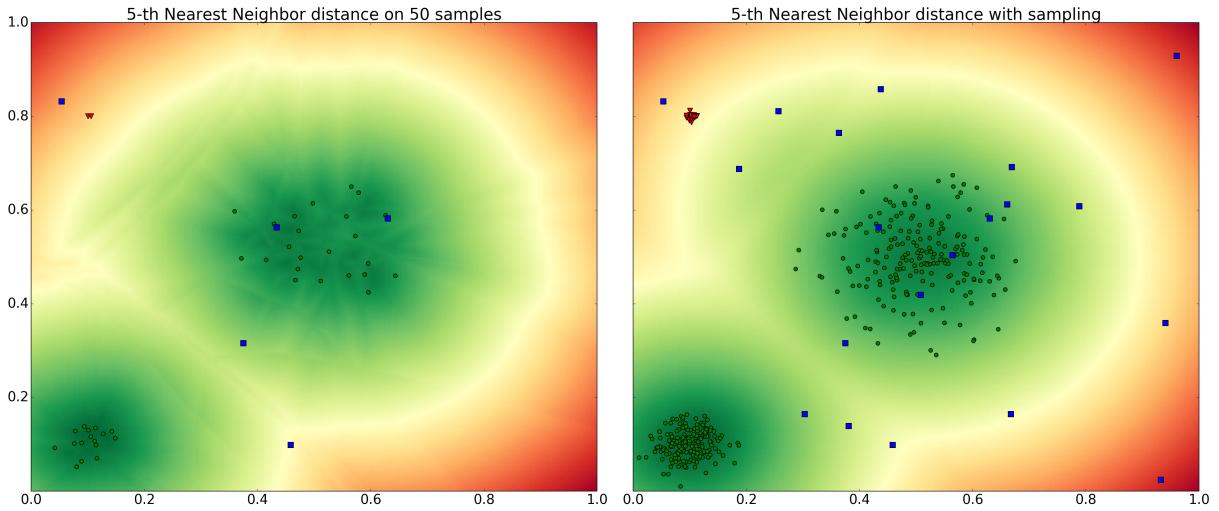


Рис. 12: Расстояние до 5-го соседа на случайной подвыборке из 50 элементов (слева) и результат усреднения по 100 случайным подвыборкам (справа)

сэмплирование, можно не переобучать алгоритм полностью, а добавить запуски в ансамбль (т. н. *warm start*).

Для изолирующего леса сэмплирование непосредственно «встроено» в сам алгоритм: поскольку сам алгоритм является ансамблем из нескольких изолирующих деревьев, каждое из деревьев строится по подвыборке из *max\_samples* элементов. На практике *max\_samples* полагают равным 250, поскольку стохастическая природа алгоритма делает настройку этого параметра бессмысленной. Таким образом, для больших выборок алгоритм не использует все имеющиеся данные, что увеличивает его обобщающую способность. Хорошая работоспособность алгоритма на малых выборках обусловлена тем, что в каждом узле решающего дерева используется лишь информация о минимальном и максимальном значении признака; а вот использование большого числа точек для построения одного дерева может привести к слишком объемному дереву, несущему избыточную и, часто, бесполезную информацию.

Для полиномиальной функции (4) и особенно гармонической (3), по построению рассчитанных на случай, когда в обучении только нормальные данные, сэмплирование позволяет обобщить алгоритм на случай, когда в данных есть аномалии. Так, при построении функций (3), (4) на данных с несколькими аномалиями, в силу изолированности этих точек, выдаваемый функцией ответ будет некорректен (близок к нулю), но лишь в небольшой окрестности. Сэмплирова-

ние позволяет усреднить несколько таких функций по разным подвыборкам, что существенно стяживает этот недостаток.

## 3.2 Случайное масштабирование

Алгоритмы детектирования аномалий зачастую не инвариантны относительно линейных преобразований пространства. Это приводит к зависимости результата от признакового представления объектов.

Так, одним из важных достоинств изолирующего леса является инвариантность к масштабированию признаков. Например, для метрических алгоритмов существенным является нормализация признаков (приведение значений к диапазону  $[-1, 1]$  или центрирование и деление на дисперсию). Однако, такая стандартная нормировка приводит к потере потенциальной информации о значимости признаков. Часто плохие результаты работы метрического алгоритма связаны именно с выбором стандартной метрики на нормированном пространстве, что не соответствует физическому смыслу признаков (который, в свою очередь, обычно скрыт) [1].

При наличии меток эту проблему можно решать параметризацией метрики и дальнейшим подбором параметров; однако, в их отсутствии это невозможно. Потенциальным решением выступает следующее рассуждение: если использование стандартно нормализованных признаков необоснованно, то можно взять несколько различных необоснованных нормализаций и усреднить по ним. Иными словами, для  $j$ -го алгоритма в ансамбле используется преобразованная выборка:

$$\begin{aligned} w(j) &\sim \text{Uniform}[0, 1]^d \\ \forall j : \widetilde{X}(j)_i &= w(j)X_i, \end{aligned} \tag{7}$$

где  $X_i$  –  $i$ -ая компонента исходной, предварительно нормализованной стандартным образом, выборки.

Это довольно общий способ борьбы с проблемой неизвестной метрики, однако гарантий того, что результат окажется лучше, не предоставляет. Более того, нет способа сравнить результат, полученный со случайными масштабированиями, и без.

### 3.3 Rotated Bagging

В отличии от масштабирования, изолирующий лес не инвариантен относительно поворотов пространства признака. Так, при построении линий уровня *anomaly\_score* изолирующего леса на модельных данных ( $d = 2$ ) можно обнаружить (см. рис. 11), что, отходя от кластеров нормальных данных вдоль осей координат, значение *anomaly\_score* изменяется незначительно. Попадающие в эту область потенциальные аномалии могут быть спутаны с граничными объектами кластера, несмотря на то, что они выражено изолированы. Причина заключается в том, что все деления пространства изолирующий лес проводит гиперплоскостями, параллельными осям координат.

Общая идея Rotated Bagging заключается в выборе случайных подпространств, на которые проецируется выборка для очередного алгоритма в ансамбле. Таким образом, для  $j$ -го алгоритма выбирается  $Q(j) \in R^{d' \times d}$  – произвольная матрица из  $d'$  нормированных попарно ортогональных векторов, после чего преобразованная выборка получается по формуле

$$\forall j : \tilde{X}(j) = XQ(j)^T \quad (8)$$

В частности, полагая  $d' = d$ , получаем произвольное ортогональное преобразование пространства признаков. Для изолирующего леса это равносильно использованию гиперплоскостей, параллельных осям случайно выбранного ортогонального базиса; усреднение позволяет получить стяженные линии уровня (рис. 13).

Недостатком такой модификации является потеря инвариантности к масштабированию признаков. Поскольку в модифицированном пространстве каждый признак является линейной комбинацией исходных, доминирование значений одного из них по модулю приведёт к завышенной значимости этого признака. Поэтому использование этого метода для изолирующего леса требует предварительной нормировки и потенциально может привести к тем же проблемам, что и использование метрических алгоритмов.

Теоретически Rotated Bagging, в том числе и в случае  $d' = d$ , может быть применён и для других алгоритмов, не инвариантных относительно поворотов про-

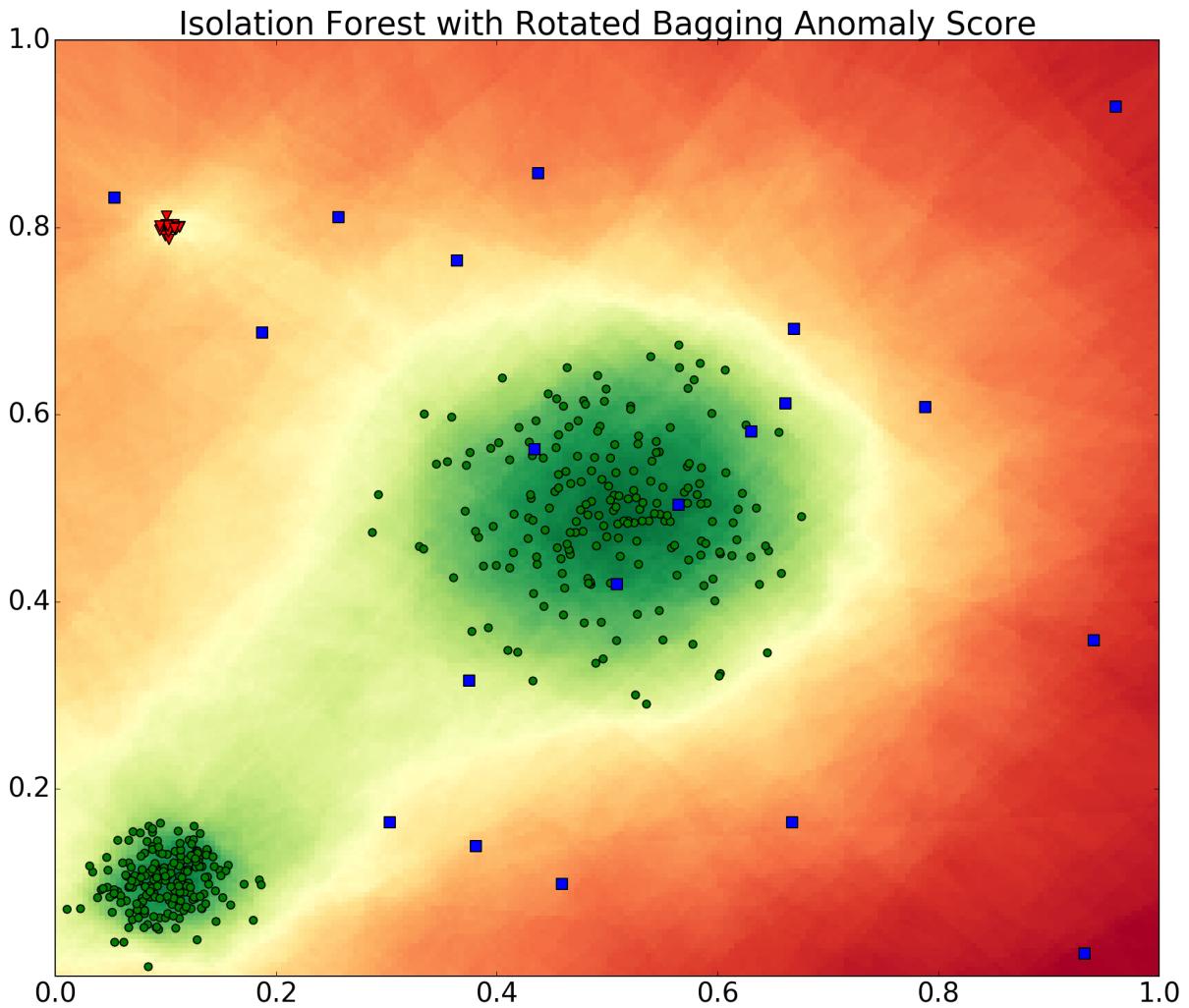


Рис. 13: Результат работы изолирующего леса с использованием Rotated Bagging

странства, в том числе и для метрических методов (при использовании неевклидовой метрики).

### 3.4 Ансамблирование

В более общем виде, ансамблирование в задаче аномалий может быть задано как использование нескольких различных алгоритмов с последующим усреднением их *anomaly\_score*. Однако, для различных алгоритмов выдаваемая ими *anomaly\_score* имеет различные шкалы и масштабы. Поэтому традиционное приведение значений функций разных алгоритмов к одному диапазону (например, к  $[0, 1]$ ) с последующим усреднением лишено своих обычных достоинств.

На практике большего успеха можно добиться ансамблированием одного и того же алгоритма на модифицированных выборках, например, при помощи сэмплирования, традиционного бэггинга, случайного масштабирования или Rotated Bagging. Объединение же результатов разнородных алгоритмов, основанных на несходих принципах и подходах, лучше проводить не усреднением их *anomaly\_score*, а голосованием по конечному ответу, является ли точка аномальной или нет. Для этого для каждого из видов алгоритмов сначала следует индивидуально определить порог бинаризации.

### 3.5 Итеративный отбор

Итеративный отбор в задаче детектирования аномалий является особым методом ансамблирования нескольких алгоритмов. [1]

Допустим, построена некоторая модель, описывающая нормальные данные. Эта модель построена на всех имеющихся данных, содержащих аномалии, и поэтому её точность может быть невелика. Однако, её приближения достаточно, чтобы выявить явные аномалии. Отсортировав все точки по *anomaly\_score*, можно выбрать  $a$  самых аномальных объекта в данных и исключить из данных. Затем модель строится заново, причём точность её приближения будет выше. Этот процесс можно повторить несколько раз или ввести какие-либо критерии останова (рис. 14).

Идея итеративного отбора может быть обобщена различными способами. Результат работы одного алгоритма может быть использован для отсеивания явных аномалий и настройки нового алгоритма, не обязательно совпадающего с предыдущим, на оставшихся данных. Возможна и противоположная механика: по результатам работы одного алгоритма отбираются явные, гарантированные представители нормальных данных, и исключительно на них строится модель, их описывающая (рис. 15).

Важно, что одни алгоритмы работают тем лучше, чем меньше аномалий в данных (построение моделей), когда другие предполагают наличие в данных аномалий (например, изолирующий лес). Если последним подать на вход только нормальные данные, построенная ими функция *anomaly\_score* обычно не сможет

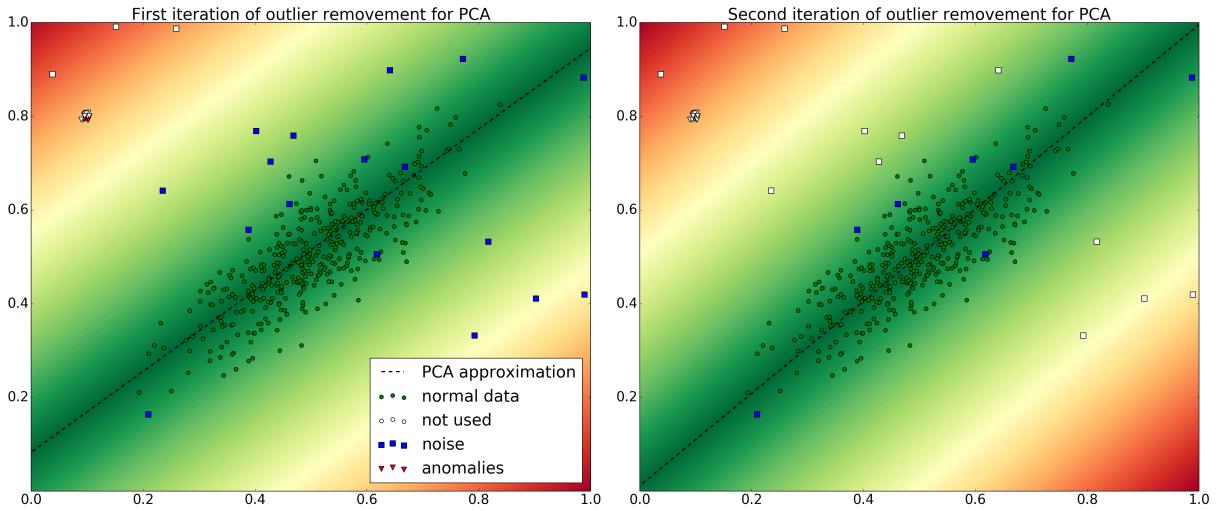


Рис. 14: Итеративное построение аппроксимирующего одномерного подпространства. На каждом шаге удаляется 15 самых аномальных объектов, после чего модель перстраивается

различать граничные нормальные данные с аномалиями, в том числе явными. Поэтому итеративный отбор следует организовывать так, чтобы первые алгоритмы отсеивали явные аномалии и/или отбирали представителей нормальных (Isolation Forest, LOF), а финальные алгоритмы были модельными, то есть строящими функцию, исходя из предположения о нормальности входной информации (например, линейные модели, ЕМ-алгоритм, полиномиальная функция).

Более специфическим инструментом является возможность встраивания результатов работы одного алгоритма «внутрь» других. Например, можно проводить построение модели по всем имеющимся данным с весами, обратно пропорциональными их рейтингу аномальности, выданным некоторым другим алгоритмом.

## 4 Вычислительные эксперименты

### 4.1 Исходные данные

Целью эксперимента является сравнение основных обобщаемых методов детектирования аномалий и их основных модификаций. Сравнение производится на семи датасетах из UCI Machine Learning Repository [11] и двух датасетах из ре-

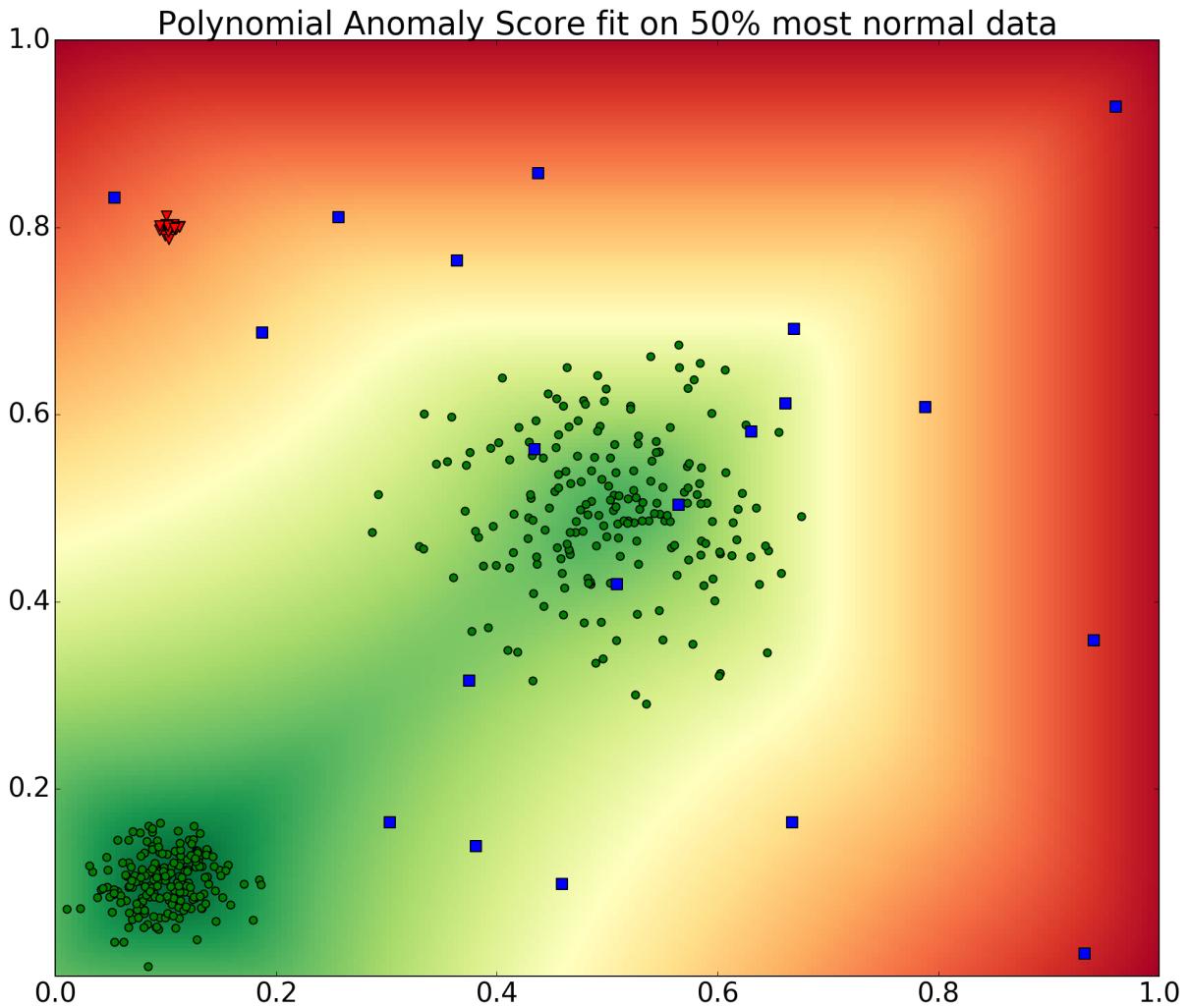


Рис. 15: Полиномиальная функция, построенная на 50% отобранных изолирующим лесом данных

позитория ODDS Library [12], для каждого из которых проведено разбиение на обучающую и тестовую выборку:

- Ionosphere – данные радаров, состоящие из 33 вещественных показателей. На обучающей выборке в 200 объектов половина аномальны. Тестовая выборка – 147 объектов, однако доля аномалий всего 16 %.
- Arrythmia – датасет классифицированных по 16 группам больных. Количество признаков – 279, что всего в два раза меньше размера датасета. Аномальными считаются классы с номерами 3-9 и 14-15. Доля аномалий составляет 15 %.

- Breastw – датасет по диагностике и классификации рака. Аномальным считается 4-ый класс. Для обучения выделяются первые 600 объектов, среди которых 36 % аномальны; особенностями датасета является особый первый из десяти признаков, принимающий сильно большие значения, и меньшее число аномалий (21 %) на контроле.
- Pima – датасет по диагностике диабета. Аномальным считается класс больных. Обучающая выборка состоит из 575 объектов; объекты представлены всего 7 признаками. Доля аномалий составляет 35 %.
- Sattelite – датасет фрагментов космических снимков, снятых в четырёх каналах. Для каждого пикселя предоставлены значения в нём и в соседях, всего - 36 признаков с одинаковым физическим смыслом. Решается задача классификации, что изображено на снимке; классы 2, 4, 5 (32 %) считаются аномальными. Размер обучающей выборки - 4 435 объектов.
- Thyroid – датасет для обнаружения тиреоидита. Для обучения 3 772 объектов, заданных 6 вещественными признаками. Классы 1 и 2 считаются аномальными (7 %).
- Shuttle – датасет по классификации шаттлов. Классы 1 и 4 считаются нормальными и составляют 94 % датасета. Обучающая выборка состоит из 43 500 объектов, заданных 9 вещественными признаками.
- Mnist – переработанная подвыборка оригинального датасета Mnist по классификации цифр. Все изображения были центрированы, после чего случайные 100 пикселей были объявлены признаками. 6903 цифры ноль считаются нормальными данными; 700 случайных изображений цифры шесть - аномальными. Перед разделением на обучающую (в 5 000 объектов) и тестовую выборку датасет был перемешан.
- ForestCover – датасет по классификации лесных массивов, переработанный под задачу распознавания аномалий (нормальными данными считается класс 2, аномалиями – класс 4). Используется только 10 вещественных

признаков. Доля аномалий при этом составляет всего 1 %. Перед разделением на обучающую (в 200 000 объектов) и тестовую (86 048 объектов) датасет был перемешан.

Все данные предварительно нормируются по обучающей выборке.

## 4.2 Сравниваемые методы и использованные реализации

Сравниваются три основных метода детектирования аномалий:

- Isolation Forest – используется реализация библиотеки sklearn [13]. Параметр `n_estimators` установлен в 100 деревьев, что на малых датасетах заводом хватает для сходимости, а при больших не приводит к значительному изменению результата. Встроенное в алгоритм сэмплирование отключено.
- Local Outlier Factor – используется реализация библиотеки sklearn v.0.19, на момент написания работы находящейся в разработке. В силу невозможности практического подбора параметра  $k$  – количества соседей – и незначительной зависимости качества на teste от изменений этого параметра, его значение оставлено дефолтным, 20.
- Полиномиальная оценка – собственная реализация (4). По определению, оценка строится для оценивая новизны новых данных, поэтому при отсутствии сэмплирования имеет смысл рассматривать этот метод только на тестовой выборке.

Полиномиальной оценке и алгоритму LOF требуется задание метрики, поэтому для каждого датасета была выбрана и использована в экспериментах наилучшая из манхэттоновой, евклидовой и чебышёвской. Любопытно, что для семи датасетов таковой оказалась манхэттонова, а для двух оставшихся (Ionosphere, ForestCover) – чебышёвская.

Для методов исследуются потенциальные способы улучшения результата:

- Сэмплирование – в силу малой зависимости качества от количества сэмплов для каждого запуска и теоретической работоспособности алгоритмов на выборках малого объёма, установлено «классическое» значение в 250 сэмплов.

На датасетах большого объёма (Shuttle, ForestCover) вычисления без сэмплирования затруднительны, поэтому на них сравнения не проводится.

- Случайное масштабирование – реализуется согласно (7).
- Rotated Bagging – реализуется согласно (8).

Дополнительно проводится сравнение с результатом итеративного отбора, проводящегося по следующей схеме: по результату изолирующего леса на обучении отбирается доля наименее аномальных данных. На отобранных данных строится полиномиальная оценка.

Во всех таблицах ниже приведены следующие сокращения:

n/m	без модификаций
S	С сэмплированием
RFW	С случайным масштабированием
RB	С использованием Rotated Bagging

Критерием качества выступает AUC-ROC выдаваемой алгоритмом *anomaly\_score*.

### 4.3 Сравнение модификаций

В таблице ниже приведено сравнение работы алгоритмов на датасетах малого размера с сэмплированием и без.

	Ionosphere	Arrythmia	Breastw	Pima	Sattelite	Thyroid	Mnist
IsolationForest	<b>0.8129</b>	0.845	0.9945	<b>0.7416</b>	<b>0.7638</b>	<b>0.7633</b>	0.8115
IsolationForest S	0.7943	<b>0.8595</b>	<b>0.9957</b>	0.708	0.6866	0.762	<b>0.8117</b>
LOF	<b>0.8748</b>	0.8234	0.348	0.6596	0.5315	0.6531	0.7422
LOF S	0.8733	<b>0.8268</b>	0.4261	<b>0.6852</b>	<b>0.7184</b>	<b>0.6896</b>	<b>0.8546</b>
Polynom	<b>0.8272</b>	<b>0.8433</b>	<b>0.9872</b>	<b>0.7361</b>	0.6451	0.6765	<b>0.8553</b>
Polynom S	0.827	<b>0.8433</b>	<b>0.9872</b>	0.7359	<b>0.6464</b>	<b>0.6779</b>	0.8543

Видно, что для полиномиальной функции использование сэмплирования практически не влияет на качество. Изолирующий лес без сэмплирования на нескольких датасетах показывает чуть лучшие результаты; а вот для алгоритма

LOF сэмплирование на относительно крупных обучающих выборках приводит к значительному росту качества.

В целом, этот результат показывает, что на крупных датасетах использование алгоритмов с сэмплированием является разумной подменой обучению на всём объёме данных.

В таблицах далее для каждого из трёх рассматриваемых алгоритмов приведено сравнение на тестовой выборке между различными модификациями.

	Isolation Forest (AUC-ROC)					
	n/m	RB	RB-RFW	RB-RFW-S	RB-S	S
Ionosphere	<b>0.8129</b>	0.8058	0.7924	0.8114	0.808	0.7943
Arrythmia	0.845	0.787	0.7809	0.8156	0.7893	<b>0.8595</b>
Breastw	0.9945	0.9664	0.9713	0.9658	0.967	<b>0.9957</b>
Pima	<b>0.7416</b>	0.7112	0.7205	0.7129	0.7214	0.708
Sattelite	<b>0.7638</b>	0.684	0.7101	0.6763	0.6817	0.6866
Thyroid	<b>0.7633</b>	0.6316	0.631	0.6502	0.624	0.762
Shuttle	-	-	-	0.9894	0.9877	<b>0.995</b>
Mnist	0.8115	0.8394	0.8531	<b>0.8569</b>	0.8414	0.8117
ForestCover	-	-	-	0.9148	<b>0.9411</b>	0.8829

По результатам эксперимента, наилучший результат показывает версия без модификаций и версия только с сэмплированием. Rotated Bagging даёт существенный прирост на датасетах Mnist и ForestCover; для первого из них это можно объяснить тем, что физический смысл признаков – пиксели изображения. Практически на всех остальных датасетах Rotated Bagging, несмотря на предварительную нормировку, приводит к потере качества как при использовании сэмплирования, так и без, что свидетельствует о потере какой-то информации при таком преобразовании. Совмещение Rotated Bagging и случайного масштабирования обычно приводит к небольшой «компенсации» потеряного качества.

	Local Outlier Factor (AUC-ROC)							
	n/m	RB	RB-RFW	RB-RFW-S	RB-S	RFW	RFW-S	S
Ionosphere	0.8748	0.8109	0.8251	0.8484	0.8394	0.8546	<b>0.8862</b>	0.8733
Arrythmia	0.8234	0.7728	0.7755	0.7786	0.7772	0.8241	0.8257	<b>0.8268</b>
Breastw	0.348	0.4231	0.3358	0.4505	0.4676	0.323	0.4383	0.4261
Pima	0.6596	0.6422	0.6556	0.6826	0.6815	0.6671	<b>0.6919</b>	0.6852
Sattelite	0.5315	0.5279	0.5285	<b>0.7207</b>	0.7127	0.5301	0.7029	0.7184
Thyroid	0.6531	0.6392	0.6829	0.6903	0.6461	0.6763	<b>0.7181</b>	0.6896
Shuttle	-	-	-	0.9621	0.9405	-	<b>0.9664</b>	0.9606
Mnist	0.7422	0.7352	0.7566	0.8739	<b>0.8867</b>	0.7449	0.8518	0.8546
ForestCover	-	-	-	0.9579	0.9667	-	0.9754	<b>0.9761</b>

Для алгоритма LOF ситуация обстоит иначе: видно, что сэмплирование даёт значительный прирост при использовании любых модификаций. Это означает, что алгоритм в базовой формулировке (6) плохо работает на выборках большого объёма. Дополнительно это позволяет алгоритму рассматривать для данной точки не только  $k$  ближайших соседей, но значительно больше информации из обучения.

За исключением датасета Mnist, Rotated Bagging приводит к небольшой потере качества; а вот случайное масштабирование стабильно даёт прирост, что, вероятно, связано с проблемой неизвестной значимости признаков и метрическим характером алгоритма.

	Полиномиальное оценивание (AUC-ROC)							
	n/m	RB	RB-RFW	RB-RFW-S	RB-S	RFW	RFW-S	S
Ionosphere	<b>0.8272</b>	0.8103	0.8077	0.7994	0.8121	0.8179	0.8115	0.827
Arrythmia	0.8433	0.7964	0.7964	0.8011	0.7971	<b>0.8443</b>	<b>0.8443</b>	0.8433
Breastw	<b>0.9872</b>	0.978	0.9817	0.9829	0.978	<b>0.9872</b>	<b>0.9872</b>	<b>0.9872</b>
Pima	0.7361	0.7341	0.7323	0.7381	0.7338	0.7311	<b>0.7404</b>	0.7359
Sattelite	0.6451	<b>0.6578</b>	0.6507	0.6522	0.6566	0.645	0.6444	0.6464
Thyroid	0.6765	0.6191	0.6186	0.6282	0.6191	0.6733	<b>0.6856</b>	0.6779
Shuttle	-	-	-	0.9882	0.9889	-	0.9906	<b>0.9908</b>
Mnist	0.8553	0.8649	0.8632	<b>0.8705</b>	0.8653	0.8532	0.8536	0.8543
ForestCover	-	-	-	0.9198	0.9308	-	0.9441	<b>0.9608</b>

Сравнительно с двумя предыдущими алгоритмами, полиномиальная функция к рассматриваемым модификациям устойчивее: результат меняется менее значительно, особенно при сэмплировании. В остальном результат эксперимента похож на LOF: случайное масштабирование хорошо показывает себя на тех же датасетах, что и для LOF-а, когда Rotated Bagging приводит к росту качества только на датасете Mnist.

#### 4.4 Сравнение алгоритмов

Помимо трёх исследуемых алгоритмов, в сравнении также участвует следующий итеративный алгоритм: к обучающей выборке применяется изолирующий лес, по результату работы которого отбрасывается  $t$  наиболее аномальных данных. На оставшемся наборе строится полиномиальная функция (4) с применением сэмплирования, что позволяет применять рассматриваемую схему в том числе на больших датасетах.

На рис. (16) приведена зависимость качества на обучении и контроле от выбранного порога  $t$  для датасетов Sattelite, Thyroid и Shuttle.

Видно, что сильного прироста качества по сравнению с простым применением полиномиальной функции (что равносильно  $t = 0$ ) такой алгоритм не даёт. В силу принципиального отсутствия возможности подобрать значение  $t$  на кон-

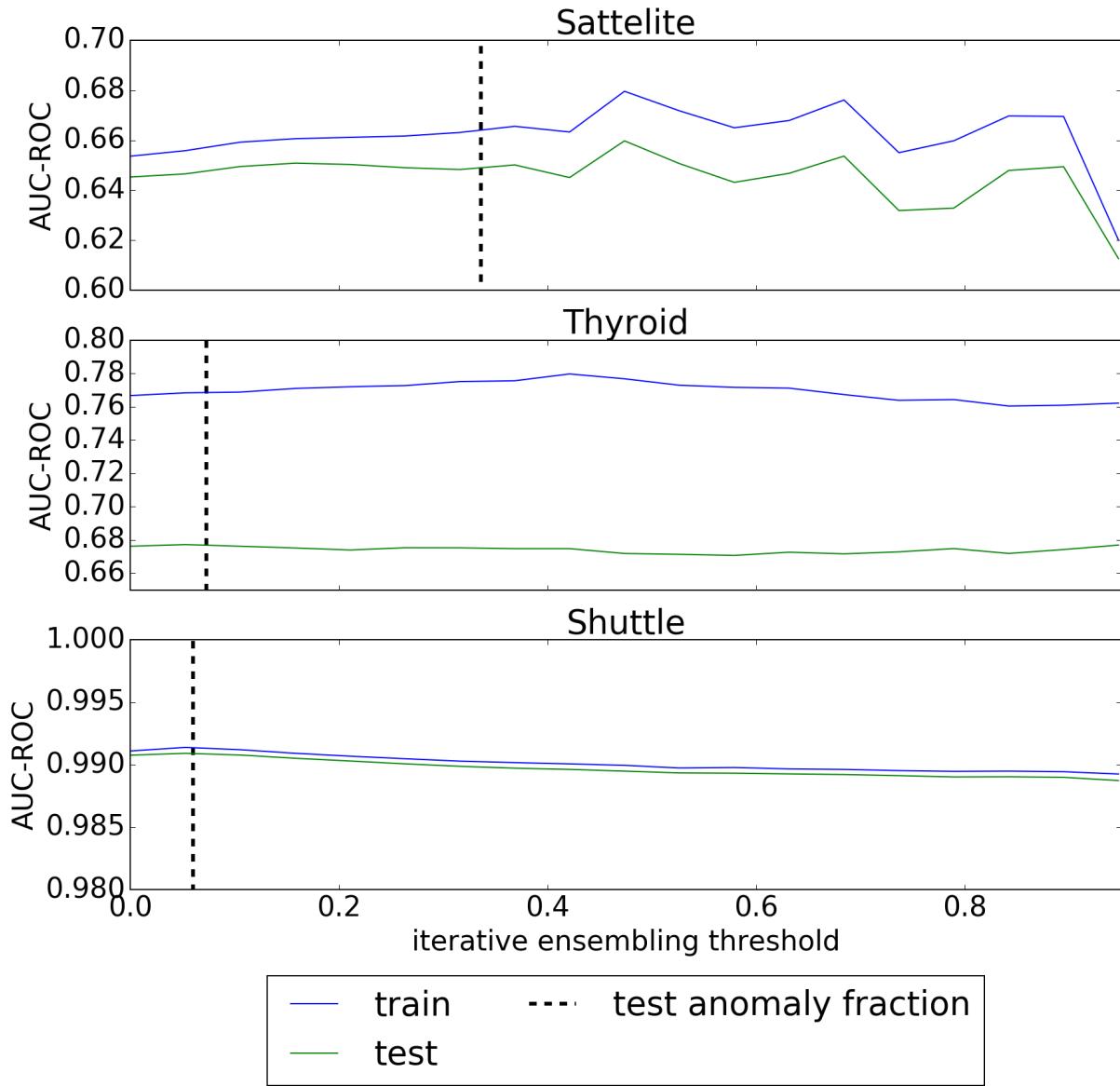


Рис. 16: Зависимость AUC-ROC от выбранного порога отбора данных

крайнем датасете, эксперименты проведены со значением  $t = 0.5$ , что исходит из разумного предположения о том, что аномалий в данных не более 50%.

Для каждого из остальных трёх алгоритмов взят наилучший достигнутый им результат с учётом модификаций, рассмотренных в 4.3. Итоговый результат на обучении и контроле приведён в таблице ниже.

	Сравнение алгоритмов (AUC-ROC)							
	Isolation Forest		LOF		Polynom		IterativeEnsemble	
	train	test	train	test	train	test	train	test
Ionosphere	0.9372	0.8129	<b>0.9604</b>	<b>0.8862</b>	0.8904	0.8272	0.9152	0.8271
Arrythmia	0.7835	<b>0.8595</b>	0.8018	0.8268	<b>0.9978</b>	0.8443	0.7893	0.8436
Breastw	0.978	0.9957	0.549	0.4676	<b>0.9989</b>	<b>0.9872</b>	0.9882	<b>0.9872</b>
Pima	0.6794	0.7416	0.6597	0.6919	0.684	0.7404	<b>0.7146</b>	<b>0.7719</b>
Sattelite	<b>0.7677</b>	<b>0.7638</b>	0.7303	0.7207	0.6699	0.6578	0.6661	0.6452
Thyroid	<b>0.8365</b>	<b>0.7633</b>	0.8002	0.7181	0.7742	0.6856	0.7765	0.6724
Shuttle	<b>0.9958</b>	<b>0.995</b>	0.9639	0.9664	0.9911	0.9908	0.9899	0.9894
Mnist	0.8483	0.8569	<b>0.8776</b>	<b>0.8867</b>	0.8647	0.8705	0.8441	0.854
ForestCover	0.9404	0.9411	<b>0.976</b>	<b>0.9761</b>	0.9598	0.9608	0.9419	0.9435

Наиболее стабильный результат выдаёт изолирующий лес – на всех датасетах его AUC-ROC не сильно уступает наилучшему. Исключение составляют датасеты Ionosphere, Mnist и ForestCover, на которых результат LOF выше, хотя сам алгоритм LOF, согласно результатам, значительно менее стабилен (например, он отчего-то «ломается» на датасете Breastw, что согласуется с результатами экспериментов, например, в [2]).

Полиномиальная оценка на большинстве датасетов превосходит результат LOF-а и сравнима с результатом изолирующего леса. Необычно высокий результат был выдан ему на обучении в датасете Arrythmia (который, тем не менее, не распространился на контроль и в силу небольшого размера датасета может быть списан на случайность). В то же время итеративный алгоритм, строящий полиномиальную функцию только на половине данных, не дал ожидаемого прироста (за исключением датасета Pima). Вероятно, это связано с тем, что с проблемой наличия аномалий в данных, использующихся для построения модели нормального поведения, сэмплирование справляется лучше удаления подозрительных точек, вместе с которыми убираются также и граничные нормальные данные.

Сравнение результатов алгоритмов на обучении и teste показывает, что все алгоритмы обладают достаточной обобщающей способностью, чтобы потреб-

ность в перестроении модели при появлении новых данных отсутствовала. Также заметно, что изменение результата на контроле по сравнению с обучением коррелирует для всех алгоритмов.

## 5 Заключение

Выбор метода для детектирования аномалий зависит в первую очередь от поставленной задачи, данных и имеющейся априорной информации. Рассмотренные подходы являются лишь математическими моделями понятия аномальности и отталкиваются от интерпретации задачи.

В отсутствие априорной информации при необходимости построить модель, описывающую нормальные данные, стоит использовать полиномиальную или гармоническую функцию. Использование сэмплирования позволяет обобщить эти алгоритмы на случай, если в данных есть небольшая доля аномалий, причём чаще всего такая модификация работает лучше итеративных методов.

Наиболее интерпретируемыми алгоритмами являются метрические, для которых необходим правильный подбор метрики и нормировка признаков. Если последняя процедура потенциально может привести к потере информации, может оказаться полезным случайное масштабирование. Сэмплирование для этого вида алгоритмов приводит как к росту качества, так и снижает вычислительную сложность.

Самым стабильным и общим алгоритмом остаётся Isolation Forest, который не требует никакой априорной информации. При этом, какие-либо модификации, «метрически исправляющие» пространство признаков, для него не нужны и чаще приводят к потере качества.

## **Список литературы**

- [1] Aggarwal, Charu C. Outlier Analysis // - 2017. -C. 1-247)
- [2] Liu F. T., Ting K. M., Zhou Z. H. Isolation Forest // Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on. – IEEE, 2008. – C. 413-422.
- [3] He Z. et al. Fp-outlier: Frequent pattern based outlier detection // Computer Science and Information Systems. – 2005. – T. 2. – №. 1. – C. 103-118.
- [4] Hodge V., Austin J. A survey of outlier detection methodologies // Artificial intelligence review. – 2004. – T. 22. – №. 2. – C. 85-126.
- [5] Knorr E. M., Ng R. T., Tucakov V. Distance-based outliers: algorithms and applications // The VLDB Journal—The International Journal on Very Large Data Bases. – 2000. – T. 8. – №. 3-4. – C. 237-253.
- [6] Breunig M. M. et al. LOF: identifying density-based local outliers // ACM sigmod record. – ACM, 2000. – T. 29. – №. 2. – C. 93-104.
- [7] De Veaux R. D., Krieger A. M. Robust estimation of a normal mixture // Statistics & Probability Letters. – 1990. – T. 10. – №. 1. – C. 1-7.
- [8] Dempster A. P., Laird N. M., Rubin D. B. Maximum likelihood from incomplete data via the EM algorithm // Journal of the royal statistical society. Series B (methodological). – 1977. – C. 1-38.
- [9] Eckart C., Young G. The approximation of one matrix by another of lower rank //Psychometrika. – 1936. – T. 1. – №. 3. – C. 211-218.
- [10] Mahalanobis P. C. On the generalized distance in statistics // Proceedings of the National Institute of Sciences (Calcutta). – 1936. – T. 2. – C. 49-55.
- [11] Asuncion A., Newman D. UCI machine learning repository. – 2007.  
URL: <http://archive.ics.uci.edu/ml> (Апрель, 2017)
- [12] Shebuti Rayana. ODDS Library. Stony Brook, - 2016. NY: Stony Brook University, Department of Computer Science.  
URL: <http://odds.cs.stonybrook.edu> (Апрель, 2017)

[13] Pedregosa F. et al. Scikit-learn: Machine learning in Python // Journal of Machine Learning Research. – 2011. – Т. 12. – №. Oct. – С. 2825-2830.  
URL: <http://scikit-learn.org/stable/index.html> (Апрель, 2017)

[14] Репозиторий с материалами и экспериментами:  
URL: <https://github.com/FortsAndMills/AnomalyDetectionMethods>