**Инженерный вестник Дона**, №8 (2021)
ivdon.ru/ru/magazine/archive/n8y2021/7166

# Коррекция обучающих выборок

*Н.И. Гданский, А.М. Крашенинников*

*Московский государственный университет пищевых производств, Москва*

**Аннотация:** Рассмотрен шум в обучающих выборках, основную часть которого составляют выбросы и новизна. Дан анализ основных причин возникновения выбросов в обучающих выборках, применяемых при построении классификаторов при обучении с учителем. Изложен характер их влияния на качество разрабатываемых решающих функций. Рассмотрена сущность основных существующих подходов к определению выбросов в обучающих выборках. В качестве самых распространенных рассмотрены измерительные ошибки, возникающие при определении численных значений свойств объектов исследуемой предметной области. На основе использования метода ближайших соседей предложена модифицированная методика сравнения обобщенных расстояний от объектов до классов, которая дает возможность учесть влияние ошибок измерений в пространстве значений признаков объектов на принятие решений о включении объекта в тот или иной класс. Для основных видов метрик, применяемых в пространствах значений признаков, найдены обоснованные значения коэффициентов запаса, используемые в данной методике. Для программной оценки качества обучающей выборки и обоснованного выбора способа коррекции выбросов в ней предложено применение допустимых долей корректируемых и удаляемых выбросов. Дан алгоритм анализа наличия выбросов в наборе обучающих примеров, в котором при помощи дополнительного коэффициента учтены возможные погрешности численных измерений свойств примеров. Приведена оценка сложности алгоритма по длине входа задачи. Разработан алгоритм оценки и коррекции обучающих выборок, позволяющий по данным анализа давать общую оценку обучающей выборки, автоматически выбирать оптимальный способ коррекции и реализовывать его. Для учета возможного косвенного влияния выбросов на результаты процесса их коррекции предложено повторное применение данной методики. Оно позволяет устранить последствия такого влияния. Приведен пример использования предложенной методики для анализа и коррекции обучающей выборки.

**Ключевые слова:** задача классификации, классификатор, решающая функция, обучающая выборка, прецедент, ошибочные данные, анализ, коррекция, искусственный интеллект, гипотеза компактности, новизна, обучение.

## Введение

При использовании методов искусственного интеллекта для распознавания объектов некоторой предметной области Ω, как правило, данные объекты характеризуются при помощи численных значений их свойств $\{\bar{x}\} = \{x_1, x_2, \ldots, x_n\}$. Величины данных свойств $\{a_1, a_2, \ldots, a_n\}$, дают возможность адекватно выделить конкретный объект в области Ω среди аналогичных ему.

Использование набора свойств $\{a_1,a_2,\ldots,a_n\}$ позволяют дать их удобную геометрическую интерпретацию в виде точки с соответствующими координатами в n-мерном пространстве значений признаков рассматриваемых объектов $U = Ox_1x_2\ldots x_n$.

Проблема автоматического распознавания (классификации) объекта по набору значений его признаков $\bar{a} = \{a_1,a_2,\ldots,a_n\}$ решается при помощи классификатора – специальной решающей функции или алгоритма μ, которые практически реализуют отображение вектора $\bar{a}$ на множество заданных классов $\{A\} = \{A_1,A_2,\ldots,A_m\}$ ($\mu : (\bar{a}) \rightarrow \{A\}$), представляющим собой группы аналогичных объектов с близкими свойствами. За счет такой близости для обработки их можно применять аналогичные методы. Данная методология лежит в основе функционирования интеллектуальных систем самого различного рода.

В случае сложных объектов области Ω классификатор μ, решающий проблему распознавания, имеет также достаточно сложную структуру. Для его автоматического построения по методике обучения с учителем применяется набор обучающих примеров $TE = \{ te^s \} = \{(\bar{a}^s, ncl^s)\}$. Каждый обучающий пример $te^s = (\bar{a}^s, ncl^s)$ состоит из конкретного объекта $\bar{a}^s = \{a_1,a_2,\ldots,a_n\}$ предметной области Ω, который уже сопоставлен своему классу из набора {A}, заданному своим номером $ncl^s$. Структура классификатора и алгоритм его построения определяются конкретным используемым методом (нейросети, геометрические методы, статистические методы и др.) Для количественной оценки качества построенного классификатора μ, как правило, применяется тестовый набор примеров, аналогичных обучающим.

Успешное построение качественного классификатора определяется в основном качеством используемых обучающих примеров ТЕ.

Основной предпосылкой хорошей разделимости рассматриваемого набора обучающих примеров ТЕ является выполнение гипотезы

компактности [1-3] для объектов, составляющих рассмотренные классы $\{A_1,A_2,\ldots,A_m\}$. По данной гипотезе точки, соответствующие этим классам, должны задавать в пространстве значений свойств объектов U отдельные кластеры, представляющие собой компактно расположенные множества точек пространства, которые можно достаточно просто отделить друг от друга при помощи специальных гиперповерхностей. Данные гиперповерхности используются затем классификаторами для определения принадлежности новых предъявляемых объектов к тому или иному классу из набора $\{A_1,A_2,\ldots,A_m\}$. Выполнение гипотезы компактности дает возможность относительно легко построить правильный классификатор.

Соответственно, нарушение данной гипотезы создает существенные сложности, как для построения удовлетворительного классификатора, так и для его последующего успешного применения при распознавании новых объектов.

Нарушение гипотезы компактности в обучающих примерах ТЕ относительно выделяемых ими классов $\{A_1,A_2,\ldots,A_m\}$, называемое шумом, не только существенно усложняет построение классификатора по такой совокупности. В итоге классификаторы, все же построенные таких обучающих данных, при последующем их применении для распознавания новых объектов повторяют ошибки, заложенные в них. [4,5]. Т.е. происходит, как бы, заранее прогнозируемое снижение качества классификатора.

Шум в обучающих примерах с нарушением гипотезы компактности может быть обусловлен действием целого ряда причин – как человеческих, так и методических, ошибок измерений и др. Подробный их перечень дан, например, в работе [6].

Общие методы устранения шума в наборах примеров, используемых при построении классификаторов, можно разделить на следующие 3 основные группы:

1) статистические, основанные, как правило, на предположении о нормальном распределении данных [6],

2) геометрические, например, DBSCAN, в которых применяется метрика в пространстве U [7-9] и

3) структурные, например, изолирующий лес, в котором для выявления шума применяются бинарные деревья [10-12].

Одним из направлений является обработка зашумленных обучающих данных по особым алгоритмам (алгоритмы надежного обучения нейронных сетей), которые учитывают наличие шума. [13-15]

В то же время практика показывает предпочтительность препроцессорных методов выявление и удаление шума из обучающих данных.

Шум в обучающих данных в основном обусловлен наличием следующих нарушений.

1. Выбросы. Это такие обучающие примеры $te^s = (\bar{a}^s, ncl^s)$, для которых неверно указаны их классы $ncl^s$ и они по своим свойствам существенно отличаются от свойств основной группы элементов класса с номером $ncl^s$

2. Новизна. Под новизной понимают такие объекты из обучающей выборки, которые по своим свойствам значительно отличаются от характерных объектов всех рассмотренных классов $\{A_1, A_2, \ldots, A_m\}$.

На практике, как обнаружение, так и коррекция обоих видов нарушений могут осуществляться как одновременно (они рассматриваются совместно, как аномалии, подлежащие удалению из обучающих данных), так и поочередно. Однако в силу того, что данные нарушения имеют разную природу, свои существенные особенности и по-разному проявляют себя, представляется, что для более полного учета данных особенностей обработка выбросов и новизны должна производиться последовательно – вначале

должны быть обнаружены и скорректированы все выбросы, после чего из скорректированных обучающих данных должна быть выявлена и уделена новизна.

Рассмотрим первую фазу данного процесса – выявление и коррекцию выбросов в обучающих данных.

Неучет выбросов в обучающих данных, во-первых, дает искаженную картину моделируемой предметной области Ω, а также приводит к ряду негативных последствий. В частности, таких, как практическая невозможность полного разделения всех обучающих примеров по заданным классам *{A}*. Такие ситуации характерны, в первую очередь, для нейросетевых методов построения классификаторов. Варьирование весов синаптических связей не дает возможности полностью разделить классы *{A}*, а только изменяет наборы неправильно классифицируемых примеров из ТЕ. В случае использования методов, позволяющих все же выполнить полное разделение зашумленных данных по классам *{A}*, построенный классификатор не только получается громоздким, но и повторяет все ошибки из обучающих данных.

На основе геометрического подхода к анализу данных в статье предложен препроцессорный алгоритм анализа и коррекции набора обучающих примеров ТЕ, состоящий из примеров - объектов с уже выделенными для них классами, который учитывает различную природу и характер влияния разных погрешностей на качество обучающих данных, а также позволяет заранее наложить общие требования к методам их коррекции и качеству исправленных данных. Для оценки обобщенного расстояния от объекта до класса использован подход метода ближайших соседей. [16-18].

В п.1. дан краткий анализ причин появления шумов и их влияния на качество обучающих примеров ТЕ, предложен способ их количественного учета.

В п.2 на основе метода ближайших соседей дано модифицированное условие включения объекта в класс, который учитывает погрешности измерения характеристик рассматриваемых объектов. Для основных видов метрик дано обоснование величины коэффициента запаса, обеспечивающее выполнение данного условия.

В п.3 сформулирована общая задача удаления выбросов. Рассмотрен специальный алгоритм выявления выбросов в обучающих данных, позволяющий за счет применения модифицированного условия включения объекта в класс практически учесть погрешности измерений характеристик объектов.

В п.4. приведена оценка сложности для алгоритма анализа выбросов.

В п.5 дан алгоритм общей оценки выбросов и определения оптимального метода коррекции обучающих данных.

## 1. Причины появления шума и выбросов в обучающих данных, их численный учет

С точки зрения причин возникновения и влияния на качество обучающей выборки все многообразие причин возникновения шумов можно разделить на две основные группы.

1. Измерительные ошибки, обусловленные погрешностями применяемых приборов, а также погрешностями, вносимыми в данные при их последующем преобразовании к машинному виду. Величины таких ошибок невелики. Поскольку они являются следствием применения для измерений того или иного оборудования, их величины можно довольно точно предсказать. Существенно особенностью их является то, что получаемые данные всегда содержат ошибки такого рода. Обобщенно

назовем их ошибками измерений. Хотя такие ошибки относительно невелики, они могут привести к неправильному отнесению объекта $\bar{a}_s$ к соответствующему классу $A_s \subset \{A\}$ – особенно для объектов с пограничным положением в классах. Это приводит к появлению выбросов и снижает общее качество обучаемых данных и построенных на них классификаторов. Ошибки измерений, как малые и трудно определяемые, трудно исправлять, однако учитывать их наличие при анализе обучающей выборки необходимо.

2. Ошибки-промахи. Это грубые ошибки, обусловленные человеческим фактором, некорректными методиками получения данных или неправильными алгоритмами их обработки. Ошибки такого рода обычно достаточно велики по сравнению с ошибками измерений, поскольку они не являются характерными, их сложно предсказать заранее. Однако число ошибок-промахов может быть достаточно велико, они могут существенно исказить обучающие данные. При небольшом числе обучающих примеров существенно исказить общую картину могут и единичные ошибки-промахи. Поскольку грубые ошибки существенно снижают качество обучающей выборки, они недопустимы и подлежат устранению.

Такой принципиально отличный характер двух рассмотренных групп ошибок дает основание для применения к ним различных видов учета:

1) ошибки измерений относительно невелики и несущественно искажают истинные свойства примеров обучающей выборки ТЕ и классифицируемых новых объектов, поэтому нет необходимости устранять их в процессе коррекции ТЕ, но необходимо учитывать в процессе анализа выборки, поскольку они могут стать причиной появления выбросов,

2) ошибки-промахи, как грубо искажающие истинные свойства объектов, недопустимы в обучающих выборках ТЕ и должны исключаться из них, поскольку само их присутствие искажает данные анализа выборки, при

наличии ошибок-промахов анализ и коррекция должны выполняться повторно.

Рассмотрим численные характеристики выбросов в обучающих выборках, по которым можно оценить качество выборки и методы и коррекции. Обозначим общий объем обучающих данных ей выборки $TE = \{ te_s \} = \{ (\bar{a}^s = (a^s_1, a^s_2, \ldots, a^s_n), cl_s)\}$ через $N$, а числа объектов в классах из $\{A\}$ – через $N_1, N_2, \ldots, N_m$, $(N_1+N_2+\ldots+N_m = N)$.

1. Предельно допустимая доля удаляемых выбросов, при которой можно без ущерба для общей информативности ТЕ удалить из нее все выбросы δdel.

2. Предельно допустимая доля корректируемых выбросов, при превышении которой обучающие данные ТЕ уже нельзя считать достаточно информативными для построения классификатора δcor.

Поскольку задачи классификации для разных предметных областей Ω имеют существенные отличия, разные объемы N и особенности методов построения классификаторов, то при назначение пороговых величин δdel и δcor необходимо учитывать специфику данных областей, решаемых задач и методов решения.

## 2. Проверка правильности включения объекта в класс с учетом погрешностей измерений. Усиленное условие включения

В обучающих данных для всех объектов $\bar{a}^s$ в примерах $te^s$ уже заданы номера $ncl^s$ соответствующих им классов из набора $\{A\}$. При проверке наличия выбросов необходимо выяснить правильность данного отображения $\bar{a}^s \to \{A\}$.

В основе геометрического метода анализа обучающих данных $TE$ лежит использование некоторой базовой метрики $\rho(\bar{a}^t, \bar{a}^u)$ в пространстве $U$, которая задает для точек пространства $\bar{a}^t$ и $\bar{a}^u$ метод расчета расстояния.

С использованием базовой метрики $\rho(\bar{a}^t, \bar{a}^u)$ пространстве $U$ вводится производная от нее мера близости $R(\bar{a}^s, A_q)$ объекта $\bar{a}^s \in U$ к произвольному классу объектов $A_q \subset \{A\}$. В качестве такой обобщенной меры принята модель KNN (K-nearest neighbors) [16-18], в которой близость объекта $\bar{a}^s$ к классу $A_q$ определяется по $k$ самым ближним точкам класса:

$$R(\bar{a}^s, A_q) = \min\{ \rho(\bar{a}^s, \bar{a}_{j1}) + \rho(\bar{a}^s, \bar{a}_{j2}) + \ldots + \rho(\bar{a}^s, \bar{a}_{jp}) \}, \quad (1)$$

где $1 \le j_1 < j_2 < \ldots < j_k \le N_q$; $\bar{a}_{jr} \neq \bar{a}^s$; $(r = 1,\ldots,k)$ .

При отсутствии погрешностей определения характеристик объектов и точных расстояниях $R_p(\bar{a}^s, A_f)$ геометрически условие включения объекта $\bar{a}^s$ в класс $A_f$ имеет вид:

$$R_p(\bar{a}^s, A_f) \le R_p(\bar{a}^s, A_g), \quad (2\ а)$$

где $A_g$ - любой класс из $\{A\}$, отличный от $A_f$ $(A_g \in \{A\}, A_g \neq A_f)$.

Однако величины характеристик объектов всегда включают погрешности измерений. Для пограничных объектов это может привести к ошибочному включению объекта не в ближайший к нем класс из $\{A\}$. Такое неправомерное отображение объекта $\bar{a}^s \in A_f$ в другой класс $A_g$ возникает в том случае, когда в случае точных значений характеристик выполняется соотношение: $R_p(\bar{a}^s, A_f) \le R_p(\bar{a}^s, A_g)$, а для приближенных: $R(\bar{a}^s, A_f) \ge R(\bar{a}^s, A_g)$.

Поскольку для первоначального включения объекта $\bar{a}^s$ в класс $A_f$, как правило, существуют некоторые весомые аргументы, то предложено дать этому классу некоторое преимущество перед всеми остальными за счет искусственного уменьшения величины расстояния $R(\bar{a}^s, A_f)$, которое рассчитано на основании реальных значений характеристик объектов, содержащих погрешности измерения. Для обеспечения преимущества

использован безразмерный коэффициент запаса ε>0. Усиленное условие включения объекта $\bar{a}^s$ в класс $A_f$ имеет вид:

$$(1-\varepsilon)\cdot R(\bar{a}^s, A_f) \le R(\bar{a}^s, A_g), \text{ где } A_g \in \{A\}, A_g \ne A_f. \qquad (2\ б)$$

Таким образом, замена в формуле (2а) точных расстояний $\{R_p(\bar{a}^s, A_f), R_p(\bar{a}^s, A_g)\}$ приближенными $\{R(\bar{a}^s, A_f), R(\bar{a}^s, A_g)\}$, рассчитанными по реальным величинам характеристик объектов, учтена при помощи коэффициента запаса ε>0. Его величина, с одной стороны, должна обеспечивать компенсацию влияния погрешности измерений. Однако коэффициент не должен принимать слишком большое значение, поскольку в этом слоучае могут быть пропущены грубые ошибки (ошибки-промахи).

Рассмотрим коэффициент ε. Он представляет собой некоторую малую величину, которая зависит в общем случае от двух основных факторов: 1) погрешностей измерений характеристик ($\{\delta_i\}$, $1 \le i \le n$) и 2) базовой метрики ρ, принятой в пространстве U. Таким образом, $\varepsilon = \varepsilon(\rho, \{\delta_i\})$.

Для обоснованного выбора коэффициента запаса ε рассмотрим влияние на него обоих фактов - погрешностей $\{\delta i\}$ и метрики ρ пространства U.

I. Погрешности $\{\delta i\}(1 \le i \le n)$. При заданной метрике ρ в пространстве U расстояние $\rho(\bar{a}^s, \bar{a}^t)$ задает длину вектора $(\bar{a}^s - \bar{a}^t)$, соединяющего точки $\bar{a}^s$ и $\bar{a}^t$ пространства. Максимальное относительное изменение (увеличение или уменьшение) всех компонент вектора $(\bar{a}^s - \bar{a}^t)$ ограничено величиной:

$$\delta_{max} = \max \delta_i,\ 1 \le i \le n. \qquad (3)$$

II. Рассмотрим метрики ρ пространства $U$. Поскольку они по-разному влияют на погрешности векторов, рассмотрим их отдельно.

1. ρ: 1) “манхеттенское расстояние” и 2) “евклидово расстояние”.

В них длины векторов ( $\bar{a}^s$ – $\bar{a}^t$) пропорциональны их линейным размерам. Вследствие этого, при одновременном увеличении всех компонент вектора в $q$ раз, длина всего вектора ( $\bar{a}^s$ – $\bar{a}^t$) также увеличится в $q$ раз.

Допустим, объект $\bar{a}^s$ входит в класс $A_f$ и не входит в класс $A_g$ ( $\bar{a}^s \in A_f$ , $\bar{a}^s \notin A_g$). Примем максимальное относительное изменение компонент вектора ( $\bar{a}^s$ – $\bar{a}^t$) равным $\delta_{max}$ (3).

В метриках 1) и 2) из пропорциональности изменения отдельных компонент вектора ( $\bar{a}^s$ – $\bar{a}^t$) и всего расстояния $\rho(\bar{a}^s, \bar{a}^t)$ следует, что в худшем случае можно ожидать следующее:

- точные расстояния в $R_p$ ( $\bar{a}^s$, $A_f$) ( $\bar{a}^s \in A_f$) из-за возникновения погрешностей увеличатся в $(1+\delta_{max})$,

- точные расстояния в $R_p$ ( $\bar{a}^s$, $A_g$) ( $\bar{a}^s \notin A_g$) изменятся в $(1- \delta_{max})$ раз, т.е. уменьшатся.

Таким образом, в результате подстановки в расчётные формулы приближенных измеренных значений характеристик полученное приближенное значение расстояния $R$ ( $\bar{a}^s$, $A_p$) может стать меньше аналогичного приближенного расстояния $R$ ( $\bar{a}^s$, $A_f$). В резуотате этого и будет принято неправильное решение о включении $\bar{a}^s$ в класс $A_p$.

Математически такую ошибку, возникающую из-за погрешностей измерений характеристик объектов, можно компенсировать следующим образом:

- полученное приближенное расстояние $R$ ( $\bar{a}^s$, $A_f$) разделить на $(1 + \delta_{max})$,

- все остальные приближенные расстояния $R$ ( $\bar{a}^s$, $A_h$), включая $R$ ( $\bar{a}^s$, $A_g$), разделить на $(1–\delta_{max})$.

Относительное изменение всех расстояний будет сохранено, если приближенное значение расстояния $R$ ( $\bar{a}^s$, $A_f$) умножить на общий коэффициент $(1–\delta_{max})/(1+\delta_{max})$, не изменяя все остальные расстояния.

Преобразование такого общего коэффициента с точностью до малых первого порядка по $\delta_{max}$ дает для коэффициента следующий результат:

$$(1–\delta_{max})/(1+\delta_{max}) = (1–\delta_{max})^2/[(1+\delta_{max})\ (1–\delta_{max})] = (1–2\delta_{max}+\delta_{max}^2)\ /(1–\delta_{max}^2)]\approx (1–2\delta_{max}).$$

Отсюда получаем, что в метриках “манхеттенское расстояние” и “евклидово расстояние” искомый коэффициент запаса, компенсирующий возможные ошибки измерений характеристик объектов, равен:

$$\varepsilon(\rho,\{\delta i\}) = 2\delta_{max}. \qquad (4а)$$

2. В метрике $\rho$ “квадрат евклидова расстояния” величина коэффициента запаса может быть получена сходным образом. Отличие заключается в том, что в данной метрике пропорциональное увеличение компонент вектора в $q$ раз приводит к увеличению всего расстояния $\rho(\bar{a}^s, \bar{a}^t)$ в $q^2$ раз.

Из данной зависимости вытекает, что в этой метрике из-за наличия погрешностей измерений в худшем случае все расстояния, входящие в сумму $R$ ( $\bar{a}^s$, $A_f$) ( $\bar{a}^s \in A_f$) изменятся в $(1+\delta_{max})^2$ раз, а все расстояния, входящие в $R$ ( $\bar{a}^s$, $A_g$) ( $\bar{a}^s \notin A_g$), изменятся в $(1- \delta_{max})^2$ раз.

Как и в первом случае, для компенсации таких предельных изменений размеров для объекта $\bar{a}^s$, расстояние $R$ ( $\bar{a}^s$, $A_f$) можно разделить на $(1+\delta_{max})^2$, а все остальные расстояния, включая $R$ ( $\bar{a}^s$, $A_g$), разделить на $(1–\delta_{max})^2$. Также по аналогии, требуемое относительное изменение всех расстояний можно получить за счет умножения расстояния $R$ ( $\bar{a}^s$, $A_f$) на приведенный коэффициент $(1–\delta_{max})^2/(1+\delta_{max})^2$ и не изменять при этом все другие расстояния.

После эквивалентных преобразований данного приведенного множителя с точностью до малых первого порядка по $\delta_{max}$ получим его следующее выражение:

$$(1–\delta_{max})^2/(1+\delta_{max})^2 = (1–\delta_{max})^4/[(1+\delta_{max})^2 (1–\delta_{max})^2] = (1–4\delta_{max}+6\delta_{max}^2 – 4\delta_{max}^3 +\delta_{max}^4) /(1–\delta_{max}^2)^2]\approx (1–4\delta_{max}).$$

Из полученного выражение вытекает, что в метрике ρ: “квадрат евклидова расстояния” в качестве коэффициента запаса должна быть принята величина:

$$\varepsilon(\rho,\{\delta i\}) = 4\delta_{max}. \qquad (4б)$$

Таким образом, для исключения влияния погрешностей измерения характеристик объектов на а принятие решения: оставить объект $\bar{a}^s$ в его исходном классе $A_f$ или переместить его в другой класс из $\{A\}$, предложено:

1) использовать усиленное условие преимущественного включения объекта $\bar{a}^s$ в класс $A_f$ (2 б) и

2) для базовых метрик ρ в пространстве U использовать соответствующие им коэффициенты запаса $\varepsilon(\rho,\{\delta i\})$ (4а) – (4б).

## 3. Общая постановка задачи корректировки выбросов в обучающей выборке. Алгоритм анализа выбросов в выборке

Предложено разбить общую задачу на два последовательных этапа. Это позволяет более гибко оценивать результаты исследования.

1. Анализ наличия и положения выбросов в обучающей выборке с учетом относительных погрешностей измерения.

Для заданной обучающей выборки ТЕ с учетом относительных погрешностей измерения характеристических признаков $\{\delta_i\}$ по условию (2б) включения объекта в класс при конкретной метрике ρ на обучающих данных *ТЕ* требуется выяснить наличие выбросов. Если они присуствуют в выборке, то необходимо сформировать предварительные данные для последующей их

общей оценки и коррекции, т.е. требуется определить общее число выбросов и их положение в обучающих данных.

2. Общее оценивание обучающей выборки, коррекция выбросов в ней. Модификация обучающих данных.

На основе полученных данных анализа обучающей выборки вычислить долю выбросов $\delta$ и для заданной предельной доли корректируемых выбросов $\delta_{cor}$ вначале проверить общую пригодность *TE* для построения классификатора, т.е. условие $\delta \le \delta_{cor}$. Если условие не выполнено, то выборка признается некачественной и ее обработка прекращается.

В том случае, когда условие выполнено, то по заданной предельной доле удаляемых выбросов $\delta_{del}$ необходимо определить способ исправления данных: а) удаление или б) исправление выбросов. После чего необходимо скорректировать обучающую выборку требуемым образом и получить модифицированные обучающие данные *TE*1.

Существенной особенностью решаемой задачи является то, что при значительном числе выбросов процесс коррекции следует повторить. Причина этого заключается в том, что выбросы, как объекты ошибочно отнесенные к другим классам, на этапе анализа могут существенно исказить суммарные расстояния и у других объектов, вызывая при этом неправильную оценку их включения в классы.

Если при повторном анализе выбросы обнаружены, то они корректируются обычным образом - как правило, их число невелико и они просто удаляются из выборки.

В итоге препроцессорной обработки в обучающей выборке устраняются все выбросы. За счет этого упрощается процесс построение классификатора и повышается качество его работы.

Такая же препроцессорная обработка данных требуется и для контрольной выборки, поскольку она также обычно определяется эмпирически и может содержать выбросы.

***Алгоритм OUTLIERS ANALYSIS анализа обучающих данных***

В алгоритме анализа обучающих данных в качестве основной вспомогательной структуры данных предложено использовать модифицированную симметричную матрицу M($N \times N$) расстояний ρ( $\bar{a}^t$; $\bar{a}^u$) между всеми парами обучающих примеров ( $\bar{a}^t$; $\bar{a}^u$) ($1 \le t \le n$; $1 \le u \le n$).

Всем диагональным элементам M([$s$] [$s$]) ($1 \le s \le n$) данной матрицы, соответствующим нулевым величинам ρ( $\bar{a}^s$; $\bar{a}^s$), специально присваивается значение $+\infty$.

Такая модификация дает возможность без выполнения дополнительных проверок автоматически исключить элементы ρ( $\bar{a}^s$; $\bar{a}^s$) из процесса определения $p$ минимальных расстояний до объекта $\bar{a}^s$ в формуле (4) в том случае, когда рассчитывается расстояние $R$( $\bar{a}^s$, $A_f$) от объекта $\bar{a}^s$ до класса $A_f$, содержащего его.

Второй вспомогательной структурой данных является массив Near []. В него заносятся расстояния от всех объектов текущего обрабатываемого класса $A_q$ до анализируемого объекта $\bar{a}^s$ при расчете расстояния $R$( $\bar{a}^s$, $A_q$).

В качестве третьей вспомогательной структуры используется несимметричная матрица RO ($N \times$k), в которой заданы расстоянииия $R$( $\bar{a}^s$, $A_q$) от объектов $\bar{a}^s$ до классов $A_q$.

***Исходные данные алгоритма***:

1) k - общее число выделенных классов в пространстве U,

2) N - общий объем обучающих данных *TE*,

3) n - число характерных признаков объектов,

4) PRV([N][n]) - массив координат точек из *TE*, упорядоченных по вхождению в классы 0, 1, …, k-1,

5) first[k]; last[k] – массивы номеров начальных и конечных объектов в классах 0, 1, …, k-1,

6) Ncl[N] - массив номеров классов для объектов из *TE*,

7) N[k] - массив чисел объектов в классах,

8) p - количество ближайших точек в классе при расчете близости,

9) eps - коэффициент запаса при проверке расстояний при известных погрешностях измерений и выбранной функции расстояний ρ между объектами,

10) RO_EV_Q(n, PRV[t][n], PRV[u][n]) – функция ρ( $\bar{a}^t$; $\bar{a}^u$) расчета расстояния между объектами $\bar{a}^t$ и $\bar{a}^u$.

***Решаемая задача***: анализ выбросов в обучающей выборке.

***Выходные данные***:

1) NV - число выбросов в обучающих данных *TE*,

2) NVb[NV] - массив номеров выбросов в обучающих данных *TE*,

3) Ncor[NV] - массив номеров корректирующих классов для выбросов.

***Вспомогательные структуры данных***:

1) M [$N\times N$] – модифицированная матрица расстояний ρ( $\bar{a}^p$; $\bar{a}^q$) между объектами ( $\bar{a}^t$; $\bar{a}^u$),

2) буферный массив Near [],

3) RO [$N\times$ k] - матрица расстояний $R(\bar{a}^s, A_q)$ между объектами $\bar{a}^s$ и классами $A_q$.

***Алгоритм Outliers analysis анализа обучающей выборки.***

{ Шаг 1. Начальные действия. Формирование матрицы M расстояний между объектами

Цикл по объектам No от 0 до N - 1

{

1.1. M[No][No] = +∞;//Инициализация диагонального элемента

1.2. Цикл по номерам объектов по i от No + 1 до N-1

{

M[No][i] = RO_EV_Q(n, No, i);//Расчет расстояния между объектами No и i

M[i][No] = M[No][i];//Присвоение значения симметричному элементу матрицы

}

};

Шаг 2. Формирование матрицы RO расстояний между объектами и классами.

Цикл по объектам No от 0 до N - 1

Цикл по классам NumC от 0 до k-1

{

2.1. Инициализация элемента матрицы RO [No][NumC]

RO [No][NumC]=0;

2.2. Формирование массива Near[NumC] расстояний от объектов класса NumC до No

Для i=0 до N(k) -1 {Near[i]= M[No][first[NumC]+i]};

2.3. Сортировка массива Near [NumC] по неубыванию величин элементов

2.4. Вычисление элемента RO [No][NumC]

Для j=0 до p-1 {RO [No][ NumC] += Near[j]};

Шаг 3. Анализ наличия выбросов

3.1. Инициализация числа выбросов NV и массивов NVb, Ncor.

3.2. Цикл по объектам по No от 0 до N-1

{

3.2.1. Определение граничного значения для исходного класса Ncl[No] с учетом коэффициента eps

ROR = (1 - eps)*RO[No][Ncl[No]];

3.2.2. Определение минимума расстояния по всем другим классам

min = ROR;

Цикл по NumC от 0 до k – 1

Если ((NumC! = Ncl[No])&& (min > RO[No][NumC]))

{min = RO[No][NumC]; Nmin = NumC; };

3.2.3. Проверка наличия выброса, коррекция NV, NVb [] и Ncor []

Если (min < ROR) {NV++; NVb[NV - 1] = No; Ncor[NV - 1] = Nmin;};

};

}

Завершение работы алгоритма.

Использование полной матрицы M ($N \times N$) максимально сокращает объем вычислений в алгоритме, поскольку все расстояния ρ( $\bar{a}^s$; $a^s$) в этом случае вычисляются только $n(n\text{-}1)/2$ раза. Но в этом случае требуется выделение значительной вспомогательной памяти $O(n^2)$. Если ее размер ограничен, то возможно хранить только половину матрицы (с небольшим увеличением числа расчетных операций) либо вообще отказаться от нее. В последнем случае общее число вычислений существенно. Оптимальный вариант алгоритма, как и в случае алгоритма DBSCAN, зависит от соотношения производительности вычислительного устройства и доступной ему оперативной памяти.

Рассмотрим пример, в котором: число характерных признаков объектов n = 2, общее число выделенных классов k = 2, общий объем

обучающих данных N = 26, количество ближайших точек при расчете по методу ближайших соседей p = 6, числа объектов в классах N[k] = {15;11}, метрика ρ в пространстве U - евклидово расстояние. относительные точности измерения характеристик по координатам δ = (0,04; 0,05). Координаты точек в пространстве U и их принадлежность к классам показаны на рис.1.

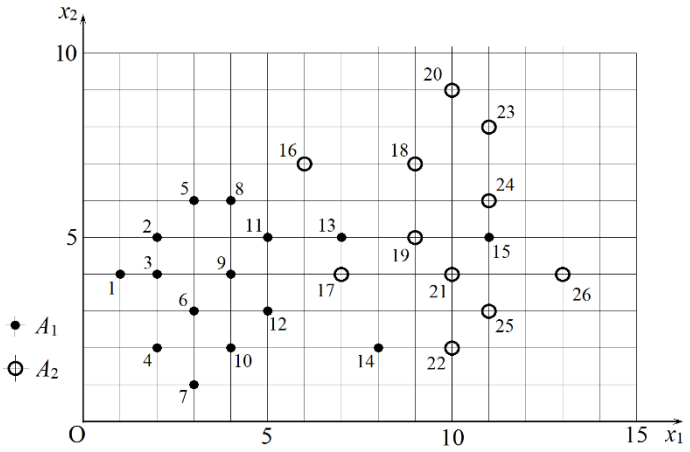


Рис. 1. – Координаты точек и их принадлежность к классам $A_1$ и $A_2$

Таблица № 1

Результаты расчета расстояний от объектов до классов $A_1$ и $A_2$

| N | *ncl* | $R(\bar{a}^s, A_1)$ | $R(\bar{a}^s, A_2)$ | N | *ncl* | $R(\bar{a}^s, A_1)$ | $R(\bar{a}^s, A_2)$ |
|---|---|---|---|---|---|---|---|
| 1 | 1 | 12.714 | 46.656 | 14 | 1 | 23.281 | 18.389 |
| 2 | 1 | 10.536 | 40.457 | 15 | 1 | 34.709 | 11.478 |
| 3 | 1 | 9.650 | 40.933 | 16 | 1 | 17.599 | 24.338 |
| 4 | 1 | 11.892 | 44.252 | 17 | 2 | 14.313 | 19.732 |
| 5 | 1 | 11.950 | 34.695 | 18 | 2 | 25.983 | 14.870 |
| 6 | 1 | 9.656 | 36.800 | 19 | 2 | 20.733 | 13.877 |
| 7 | 1 | 13.981 | 41.852 | 20 | 2 | 37.130 | 20.407 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 8 | 1 | 12.640 | 29.072 | 21 | 2 | 23.602 | 13.064 |
| 9 | 1 | 10.242 | 29.860 | 22 | 2 | 26.334 | 17.910 |
| 10 | 1 | 11.071 | 34.217 | 23 | 2 | 36.506 | 17.851 |
| 11 | 1 | 11.892 | 23.874 | 24 | 2 | 29.914 | 14.536 |
| 12 | 1 | 12.485 | 26.686 | 25 | 2 | 29.030 | 15.016 |
| 13 | 1 | 18.315 | 15.349 | 26 | 2 | 38.828 | 20.265 |

Проверка по таблице по условию (5б) показывает наличие 5 выбросов в точках с номерами:

13 ($ncl$ =1: (1-ε)$R(\ \bar{a}^s, A_1) > R(\ \bar{a}^s, A_2)$);

14 ($ncl$ =1: (1-ε)$R(\ \bar{a}^s, A_1) > R(\ \bar{a}^s, A_2)$);

15 ($ncl$ =1: (1-ε)$R(\ \bar{a}^s, A_1) > R(\ \bar{a}^s, A_2)$);

16 ($ncl$ =2: (1-ε)$R(\ \bar{a}^s, A_2) > R(\ \bar{a}^s, A_1)$);

17 ( $ncl$ =2: (1-ε)$R(\ \bar{a}^s, A_2) > R(\ \bar{a}^s, A_1)$).

## 4. Сложности алгоритма анализа обучающей выборки

Вход алгоритма в основном составляет массив PRV([N][n]) координат точек, задающих значения характеристических признаков объектов обучающей выборки. Поэтому длина L входа задачи пропорциональна n·N.

В алгоритме 4 последовательных шага. Так как на Шаге 4 число операций незначительно, то сложность всего алгоритма определяется максимальной из сложностей выполнения Шагов 1- 3. Проанализируем их.

Шаг 1. Формирование матрицы M расстояний.

Рассмотрим использованием базовой метрики “евклидово расстояние”. При числе характеристических признаков n вычисление одного расстояния требует выполнения: 1) n вычитаний, 2) n возведений в квадрат и 3) (n-1) сложений и 4) одно извлечение квадратного корня.

Диагональные элементы матрицы M равны $+\infty$ она является симметричной. Поэтому в первой ее строке расстояние вычисляется (N-1)

раз, в строке 2 - (N -2) раз, …, в строке N – 0 раз. Суммарное число вычислений расстояния равно N (N -1)/2. Поскольку порядок для основных операций совпадает, то сложность выполнения Шага 1 равна $O(n \cdot N^2)$.

Шаг 2. Формирование матрицы RO расстояний между объектами и классами.

При формировании матрицы RO выполняется вложенный цикл по N объектам и k классам. При расчете каждого элемента RO[*i*][*i*] (соответствующего объекту $i$ $(1 \le i \le N)$ и классу $j$ $(1 \le j \le k)$) основной объем расчётов затрачивается на выполнение следующих действий.

п.2.2. Выполняется $N_j$ операций присваивания (по числу объектов в классе *j*) при формировании массива Near(*i*, *j*). Всего для класса с *j* алгоритма операция присваивания выполняется $N \cdot N_j$ раз. В сумме по всем классам данная операция выполняется $N \cdot (N_1 + N_2 + \ldots + N_k) = N \cdot N = N^2$ раз.

п.2.3. Сортировка массива Near(*i*, *j*) по неубыванию величин элементов. У оптимальных алгоритмов сложность сортировки равна $O(N_j \cdot \log_2 N_j)$. Так как средний размер классов равен N/k, то число операций при сортировке будет пропорционально величине $N \cdot N \cdot \log_2(N/k) = N^2 \cdot \log_2(N/k)$.

п.2.4. Окончательный расчет элемента RO[*i*][*i*] требует выполнения **(**p-1) операции сложения. Так как вычисление элементов RO[*i*][*i*] выполняется $N \cdot k$ раз, то общее число операций сложения в этой операции равно $N \cdot k \cdot p$.

Поскольку N>> k,·N>>p, то сложность выполнения Шага 2 равна $O(N^2 \cdot \log_2(N/k))$.

Шаг 3. Анализ наличия выбросов

Основная часть операций выполняется во вложенном цикле 3.2. по объектам и классам, внутри которого производится несколько проверок логических условий и присваиваний. Таким образом, итоговая сложность выполнения Шага 3 равна $O(N \cdot k)$.

Сравнение итоговых сложностей выполнения Шагов 1-3 показывает, что максимальную из них составляют сложности Шага 1 ($O(n \cdot N^2)$) или Шага 2 ($O(N^2 \cdot \log_2(N/k))$). С учетом того, что длина входа задачи L пропорциональна $n \cdot N$ и $n << N$, $k << N$, то сложность алгоритма анализа обучающей выборки выше линейной по L, но меньше квадратичной.

## 5. Алгоритм оценки и коррекции обучающих данных

Полученные в результате предварительного анализа выбросов в обучающей выборке результаты {NV, NVb[NV], Ncor[NV]} должны быть оценены и соответствующим образом обработаны с учетом рассмотренных выше 1) предельно допустимой доли корректируемых выбросов $\delta_{cor}$ и 2) предельно допустимой доли удаляемых из обучающих данных объектов $\delta_{del}$. На основе этих предварительных оценок принимается решение о возможном использовании обучающих данных *TE* для построения классификатора μ и ее коррекции.

Вначале определяется доля выбросов в обучающей выборке: $\delta = NV/N$.

По величине δ проверяется пригодной обучающей выборки лоя построения классификатора:

$$\delta \le \delta_{cor}. \qquad (5)$$

Если условие (5) не выполняется, то обучающие данные *TE* признаются неинформативными для построения классификатора. В таких случах необходим анализ адекватности выделяемых характерных признаков $\{ \bar{x} \}$ у рассматриваемых объектов, а также методов их получения.

В том случае, если условие выполнено, обучающая выборка *TE* признается достаточно качественной для последующего построения классификатора μ. Однако в случае наличия выбросов должно быть дополнительно принято решение о том, каким способом необходимо

выполнять их коррекцию. Рассматриваются два альтернативных варианта: 1) удаление всех выбросов в *TE* или 2) коррекция - исправление во всех обучающих примерах ( $\bar{a}^s, n^s) \in TE$, являющихся выбросами, номера класса $n^s$, в которые они ошибочно включены.

Для выбора способа коррекции дополнительно производится проверка условия:

$$\delta < \delta_{del}. \qquad (6)$$

Выполнение условия (6) означает, что выбросы составляют малую долю от общего числа учебных примеров, и их можно удалить без существенной потери информативности выборки *TE*. В простейшем случае выбросы программно удаляются из *TE*. Учитывая их малое число, также возможен человеко-машинный анализ выбросов.

Если условие (6) не выполнено, то это означает, что доля выбросов достаточно велика и они составляют существенную часть всех обучающих данных. Отбрасывание выбросов приведет к утрате значительной их части. Поскольку вручную анализ такого объема ошибочных данные выполнить невозможно из-за большого их объема, то предложена автоматизированная коррекция выбросов, отмеченных в массиве *NV*. При этом каждому выбросу в соответствующем обучающем примере ( $\bar{a}, A_f$) класс $A_f$ заменяется классом $A_r$, для которого номер $r$ определяется из массива *NCOR*. В результате такой замены условие (2б) корректности обучающего примера будет выполнено и выброс будет устранен.

Для выполнения общей оценки и коррекции обучающей выборки предложен алгоритм ***EVAL_COR***.

Помимо общих данных по *TE*, он учитывает предельно допустимые доли удаляемых $\delta_{del}$ и корректируемых $\delta_{cor}$ выбросов, а также результаты предварительного анализа *TE* алгоритмом *OUTLIERS ANALYSIS* - число *NV* выбросов в *TE*, массивы *NV* и *NCOR*.

***Входные данные***:

1) k - общее число выделенных классов,

2) N - общий объем обучающих примеров *TE*,

3) Ncl[N] – исходный массив номеров классов для объектов из *TE*,

4) N[k] - массив чисел объектов в классах,

5) PRV([N][n]) - массив координат точек из *TE*, упорядоченных по вхождению в классы 0, 1, …, k-1,

6) first[k]; last[k] – массивы номеров начальных и конечных объектов в классах 0, 1, …, k-1,

7) DelDel – допустимая доля удаляемых выбросов в *TE*,

8) DelCor - допустимая доля корректируемых выбросов в *TE*,

9) NV - число выбросов в *TE*,

10) NVb[NV] - массив номеров выбросов в *TE*,

11) Ncor[NV] - массив номеров корректирующих классов для выбросов.

***Решаемая задача***: оценка качества и возможная коррекция обучающих данных *TE*.

***Выходные данные:***

1) Q – показатель качества (Q=true – качество данных удовлетворительное, Q=false - нет),

2) N1 - результирующий общий объем обучающих данных *TE*,

3) Ncl1[N] – результирующий массив номеров классов для объектов из *TE*,

4) N1[k] - результирующий массив чисел объектов в классах,

5) PRV1([N][n]) - результирующий массив координат точек из *TE*, упорядоченных по вхождению в классы 0, 1, …, k-1,

6) first1[k]; last1[k] – результирующие массивы номеров начальных и конечных объектов в классах 0, 1, …, k-1.

Шаг 1. Расчет доли выбросов и общая оценка качества обучающих данных *TE*

1.1. Расчет доли выбросов в обучающих данных *TE*

DelV = 1.0*NV/N;

1.2. Определение показателя качества Q и общая оценка качества обучающих данных *TE*

Q= false; if(DelV <= DelCor) Q=true;

if(Q= true) {//качество обучающая выборка удовлетворительное

Шаг 2. Определение способа устранения выбросов. Корекция выбросов

Если (DelV> DelDel) // проверка возможности удаления выбросов

***{2.1. Коррекция общих данных***

**2.1.1. К**оррекция чисел объектов в классах

for i=0 to k-1 do Nn1[i]=Nn[i];// инициализация новых чисел объектов

for i=0 to NV-1 do Nn1[Ncl[NVb[i]]]=Nn1[Ncl[NVb[i]]]-1;//увеличение числа объектов

for i=0 to NV-1 do Nn1[Ncor[i]]=Nn1[Ncor[i]]+1;//уменьшение числа объектов

2.1.2. Коррекция номеров крайних элементов классов в общем массиве

first1[0]=0; last1[0]=Nn1[0]-1;

for i=1 to k-1 do

{ first1[i]=last1[i-1]+1;last1[i]=first1[i]+Nn1[i]-1;}

***2.2. Построение исправленного массива координат PRB1. Производится по классам i=0 to k-1.***

2.2.1. Формирование списка удалений для класса i

NDEL=0;for j=0 to NV-1

```
{if (Ncl[NVb[j]]==i) { DEL[NDEL]=NVb[j]; NDEL=NDEL+1;}
}
```

2.2.2. Формирование списка вставок для класса i

```
NSUM=0;for j=0 to NV-1
{if (Ncor[j]==i) { SUM[NSUM]=NVb[j]; NSUM=NSUM+1;}
}
```

2.2.3. Формирование участка PRB1 для класса i

```
//Запись в PRB1 всех неисключенных объектов из PRB
ii=0; //счетчик по PRB
jj=0;//счетчик по PRB1
kk=0;//счетчик по DEL
while (first[i]+ ii<=last[i]) {
if (first[i]+ii<>DEL[kk])
{
for mm=0 to n-1 do PRB1[first1[i]+jj][mm]=PRB[first[i]+ii][mm]; jj=jj+1;
 }
else if(kk<NDEL) then kk=kk+1;
ii = ii+1;
};
//Запись в PRB1 всех вновь включаемых объектов из PRB
for kk = 0  to NSUM-1
{ for mm=0 to n-1
{PRB1[first1[i]+jj][mm]=PRB[SUM[kk]][mm];}
  jj=jj+1;
}
}
else
```

2.2. Удаление выбросов

2.2.1. Формирование списка удалений для класса i

```
NDEL=0;for j=0 to NV-1
{if (Ncl[NVb[j]]==i) { DEL[NDEL]=NVb[j]; NDEL=NDEL+1;}
}
```

2.2.2. Формирование участка PRB1 для класса i

```
//Запись в PRB1 всех неисключенных объектов из PRB
ii=0; //счетчик по PRB
jj=0;//счетчик по PRB1
Kk=0; //счетчик по DEL
while (first[i]+ ii<=last[i]) {
I f (first[i] +ii<>DEL[kk])
{
for mm=0 to n-1 do PRB1[first1[i]+jj][mm]=PRB[first[i]+ii][mm]; jj=jj+1;
 }
else if(kk<NDEL) then kk=kk+1;
ii = ii+1;
};
}
```

***Завершение работы алгоритма***

Сложность алгоритма EVAL_COR линейна по длине входа задачи.

При повторном проверочном применении алгоритма OUTLIERS ANALYSIS возможны следующие ситуации:

1) выбросов нет (NV = 0),

2) выбросы обнаружены (NV>0).

В первом случае обработка ТЕ завершается, во втором повторно вызывается алгоритм EVAL_COR.

Обработка результатов анализа рассмотренного выше примера {NV=5; NVb=(12,13,14,15,16); NCor=(1,1,1,0,0)} при заданных долях DelCor

= 0.2 и DelDel = 0.1 с применением алгоритма EVAL_COR дает следующие результаты.

Шаг 1. Расчет доли выбросов и общая оценка качества обучающих данных *TE*

DelV = 5/26 ≈ 0.192 < DelCor = 0.2; Q = true - качество обучающей выборки удовлетворительное.

Шаг 2. Проверка возможности коррекции выбросов.

Так как DelV= 0.192 > DelDel= 0.1, то выполняется условие коррекции выбросов.

2.1. Коррекция общих данных *TE*

2.1.1. Коррекция чисел объектов в классах

Инициализация обновленных чисел объектов в классах: **{**Nn1[i]} = {Nn[i]} = {15; 11}.

Уменьшение чисел объектов в классах: **{**Nn1[i]} = {15-3; 11-2} = {12; 9}.

Увеличение чисел объектов в классах: **{**Nn1[i]} = {12+2; 9+3} = {14;12}.

2.1.2. Коррекция номеров крайних элементов классов в общем массиве

first1 [0] = 0; last1 [0] = Nn1 [0]-1 = 13; first1 [1] = 14; last1 [1] = 25.

2.2. Результат построения исправленного массива объектов PRB1 с обновленной нумерацией точек показан на рис.2.

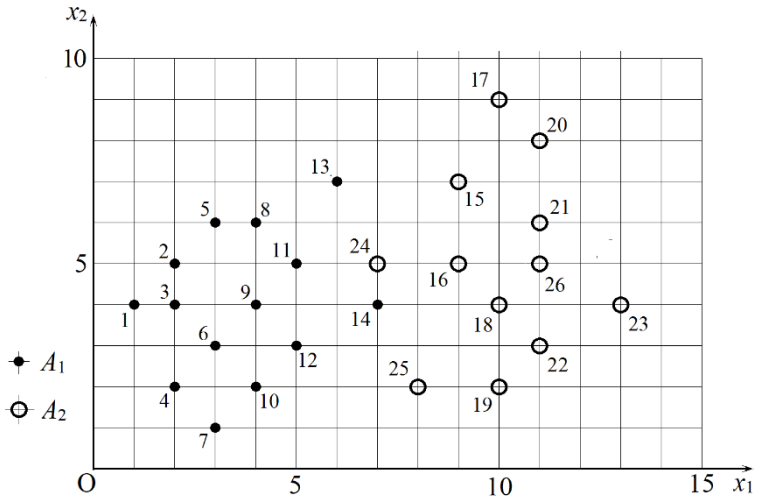


Рис.2. – Исправленный массив обучающих примеров с обновленной нумерацией точек

Итоговый вид обучающей выборки дан на рис.3.

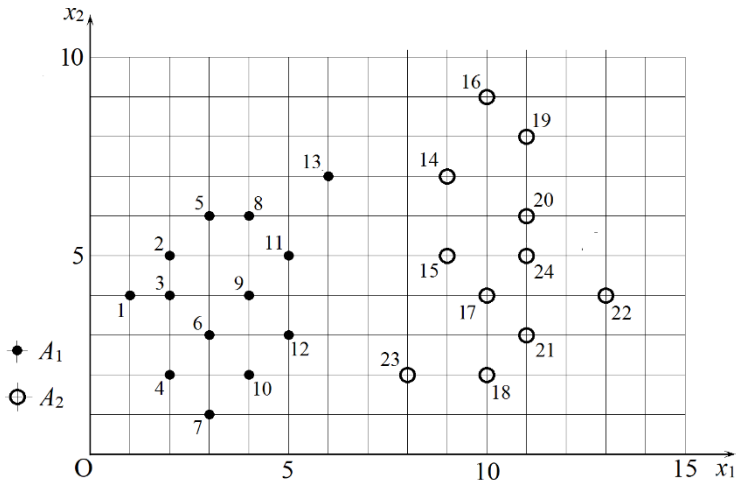


Рис.3. – Итоговый вид обучающей выборки

Как видно из рис.3, применение скорректированных классов $A_1$ и $A_2$ не вызовет трудностей при построении классификаторов любого типа.

## Выводы

Рассмотренная задача устранения шума в обучающих выборках при обучении с учителем является одной из наиболее актуальных для систем искусственного интеллекта.

Описанные алгоритмы анализа и коррекции обучающих данных имеют довольно простую структуру и полиномиальную сложность по длине входа, не превышающую квадратичную.

За счет применения дополнительных коэффициентов алгоритмы позволяют не только учитывать влияние погрешностей измерений характеристик объектов, но и гибко подходить к общему анализу качества обучающих данных и коррекции выбросов в них.

Получаемые в результате обучающие выборки задают более плавные границы классов $\{A\}$ в пространстве значений признаков U, которые удовлетворяют гипотезе компактности. Это существенно упрощает

отделимость классов и дает возможность построения классификаторов с более простой структурой, которые затем затрачивают меньшее число вычислительных операций при классификации новых объектов.

Одним из преимуществ первоначального удаления выбросов в обучающей выборке является то, что получаемые в результате скорректированные обучающие данные существенно упрощают решение последующей задачи устранения новизны, поскольку она уже изолирована в отдельных классах. Это свойство скорректированных обучающих данных существенно снижает объемы перебора в алгоритмах, применяемых для выявления новизны.

## Литература

1. Глушков В. М., Амосов Н. М., Артеменко И. А. Энциклопедия кибернетики. Статья “Гипотеза компактности”, автор Шлезингер М. И., с.229. Том 1. Киев, 1974, 608 с.

2. Гречуха Е. И. Управление изменениями на основе гипотезы компактности. Восточно-европейский журнал передовых технологий. 2013, №10. с.58-60. URL: cyberleninka.ru/article/n/upravlenie-izmeneniyami-na-osnove-gipotezy-kompaktnosti

3. Моттль В., Середин О., Красоткина О. Гипотеза компактности, потенциальные функции и исправление линейного пространства в машинном обучении. URL: link.springer.com/chapter/10.1007/978-3-319-99492-5_3

4. Чен Д., Джайн Р. 1994. Надежный алгоритм обучения методом обратного распространения ошибки для аппроксимации функций. IEEE Trans Neural Netw 5 (3): 467–479. PMID: 18267813. DOI: 10.1109 / 72.286917

5. Лиано К. 1996. Надежная мера ошибок для контролируемого обучения нейронной сети с выбросами. Транснейронная сеть IEEE 7 (1): 246–250 PMID: 18255577. DOI: 10.1109 / 72.478411

6. Серхио Сантойо. Краткий обзор методов обнаружения выбросов. Что такое выбросы и как с ними бороться? 2017. URL: towardsdatascience.com/a-brief-overview-of-outlier-detection-techniques-1e0b2c19e561

7. Кампелло, Рикардо Дж. Г. Б.; Мулави, Давуд; Зимек, Артур; Сандер, Йорг 2015. Иерархические оценки плотности для кластеризации данных, визуализации и обнаружения выбросов. ACM-транзакции при обнаружении знаний из данных. 10 (1): 1–51. DOI: 10,1145 / 2733381. ISSN 1556-4681. S2CID 2887636.

8. Шуберт, Эрих; Сандер, Йорг; Эстер, Мартин; Кригель, Ганс Петер; Сюй, Сяовэй. 2017. Возвращение к DBSCAN, повторение: почему и как (по-прежнему) следует использовать DBSCAN. ACM Trans. База данных Syst. 42 (3): 19: 1–19: 21. DOI: 10,1145 / 3068335. ISSN 0362-5915. S2CID 5156876.

9. Шуберт, Эрих; Гесс, Сибилла; Морик, Катарина. Связь DBSCAN с матричной факторизацией и спектральной кластеризацией (PDF). 2018, Lernen, Wissen, Daten, Analysen (LWDA). С. 330–334. CEUR-WS.org.

10. Шаффер, Клиффорд А. Структуры данных и анализ алгоритмов в Java (3-е изд. Дувра). 2011, Минеола, Нью-Йорк: Dover Publications. ISBN 9780486485812. OCLC 721884651.

11. Дин, Чжиго; Фей, Минруй. Подход к обнаружению аномалий, основанный на алгоритме изолированного леса для потоковой передачи данных с использованием скользящего окна. 3-я Международная конференция МФБ по интеллектуальному управлению и автоматизации. 2013.

12. Дилини Талагала, Прияnга; Гайндман, Роб Дж.; Смит-Майлз, Кейт. Обнаружение аномалий в данных большой размерности. 2019, arXiv: 1908.04000 [stat.ML].

13. Мирослав Кордос, Анджей Русецкий. Снижение шумового воздействия на тренировку MLP. Мягкие вычисления. 2015. Т. 20. С. 49–65. DOI: 10.1007 / s00500-015-1690-9. URL: link.springer.com/article/10.1007/s00500-015-1690-9

14. Кордос М., Русецки А. Повышение производительности нейронной сети MLP за счет уменьшения шума. Конспект лекций по информатике. 2013. TPNC, Vol. 8273, pp. 133–144. URL: link.springer.com/chapter/10.1007/978-3-642-45008-2_11

15. Русецкий А. Алгоритм робастного обучения, основанный на итеративном методе наименьшей медианы квадратов. 2012. Neural Process Lett 36 (2): 145–160, URL: link.springer.com/article/10.1007/s11063-012-9227-z

16. Нигш, Флориан; Бендер, Андреас; ван Бюрен, Бернд; Тиссен, Йос; Нигш, Эдуард; Митчелл, Джон Б. О. "Прогнозирование точки плавления с использованием алгоритмов k-ближайшего соседа и оптимизации генетических параметров". Журнал химической информации и моделирования. 2006, 46 (6): 2412–2422. DOI: 10.1021 / ci060149f. PMID 17125183.

17. Холл, Питер; Сэмворт, Ричард Дж., Выбор порядка соседей в классификации ближайших соседей. Анналы статистики. 2008. 36 (5): 2135–2152. arXiv: 0810.5276. Bibcode: 2008arXiv0810.5276H. DOI: 10.1214 / 07-AOS537. S2CID 14059866.

18. Бремнер, Дэвид; Демейн, Эрик; Эриксон, Джефф; Яконо, Джон; Лангерман, Стефан; Morin, Pat; Туссен, Годфрид Т. Чувствительные к выходу алгоритмы для вычисления границ решения ближайшего соседа. Дискретная и вычислительная геометрия. 2005, 33 (4): 593–604. DOI: 10.1007 / s00454-004-1152-0.

19. Журавлев Ю. И., Рязанов В. В., Сенько О. В. «Распознавание». Математические методы. Программная система. Практические применения. — М.: Фазис, 2006. ISBN 5-7036-0108-8.

## References

1. Glushkov V. M., Amosov N. M., Artemenko I. A. E`nciklopediya kibernetiki. Stat`ya “Gipoteza kompaktnosti”, avtor Shlezinger M. I., p.229. Tom 1. Kiev, 1974, 608 p.

2. Grechuxa E. I. Vostochno-evropejskij zhurnal peredovy`x texnologij. 2013, №10. pp.58-60. URL: cyberleninka.ru/article/n/upravlenie-izmeneniyami-na-osnove-gipotezy-kompaktnosti

3. Mottl` V., Seredin O., Krasotkina O. Gipoteza kompaktnosti, potencial`ny`e funkcii i ispravlenie linejnogo prostranstva v mashinnom obuchenii. [Compactness hypothesis, potential functions and linear space correction in machine learning]. URL: link.springer.com/chapter/10.1007/978-3-319-99492-5_3

4. Chen D., Dzhajn R. 1994. Nadezhny`j algoritm obucheniya metodom obratnogo rasprostraneniya oshibki dlya approksimacii funkcij [Robust backpropagation learning algorithm for function approximation]. IEEE Trans Neural Netw 5 (3): 467–479. PMID: 18267813. DOI: 10.1109 / 72.286917

5. Liano K. (1996) Nadezhnaya mera oshibok dlya kontroliruemogo obucheniya nejronnoj seti s vy`brosami. [A robust measure of error for supervised learning of an outlier neural network]. Transnejronnaya set` IEEE 7 (1): 246–250 PMID: 18255577. DOI: 10.1109 / 72.478411

6. Serxio Santojo. Kratkij obzor metodov obnaruzheniya vy`brosov. Chto takoe vy`brosy` i kak s nimi borot`sya? [A brief overview of outlier detection methods. What are emissions and how to deal with them?] 2017. URL: towardsdatascience.com/a-brief-overview-of-outlier-detection-techniques-1e0b2c19e561

7. Kampello, Rikardo Dzh. G. B.; Mulavi, Davud; Zimek, Artur; Sander, Jorg, 2015. Ierarxicheskie ocenki plotnosti dlya klasterizacii danny`x, vizualizacii i obnaruzheniya vy`brosov. ACM-tranzakcii pri obnaruzhenii znanij iz danny`x. 10 (1): 1–51. DOI: 10, 1145 2733381. ISSN 1556-4681. S2CID 2887636.

8. Shubert, E`rix; Sander, Jorg; E`ster, Martin; Krigel`, Gans Peter; Syuj, Syaove`j, 2017. Vozvrashhenie k DBSCAN, povtorenie: pochemu i kak (po-prezhnemu) sleduet ispol`zovat` DBSCAN». ACM Trans. Baza danny`x Syst. 42 (3): 19: 1–19: 21. DOI: 10, 1145 3068335. ISSN 0362-5915. S2CID 5156876.

9. Shubert, E`rix; Gess, Sibilla; Morik, Katarina. Svyaz` DBSCAN s matrichnoj faktorizaciej i spektral`noj klasterizaciej (PDF). 2018, Lernen, Wissen, Daten, Analysen (LWDA). pp. 330–334. CEUR-WS.org.

10. Shaffer, Klifford A. Struktury` danny`x i analiz algoritmov v Java (3-e izd. Duvra). [Data Structures and Algorithm Analysis in Java (Dover 3rd ed.)]. 2011, Mineola, N`yu-Jork: Dover Publications. ISBN 9780486485812. OCLC 721884651.

11. Din, Chzhigo; Fej, Minruj. 3-ya Mezhdunarodnaya konferenciya MFB po intellektual`nomu upravleniyu i avtomatizacii. 2013.

12. Dilini Talagala, Priyanga; Gajndman, Rob Dzh; Smit-Majlz, Kejt. Obnaruzhenie anomalij v danny`x bol`shoj razmernosti. 2019, arXiv: 1908.04000 [stat.ML].

13. Miroslav Kordos, Andzhej Ruseczkij. Snizhenie shumovogo vozdejstviya na trenirovku MLP. Myagkie vy`chisleniya. 2015. T. 20. pp. 49–65. DOI: 10.1007 s00500-015-1690-9. URL: link.springer.com/article/10.1007/s00500-015-1690-9

14. Kordos M., Ruseczki A. Povy`shenie proizvoditel`nosti nejronnoj seti MLP za schet umen`sheniya shuma. Konspekt lekcij po informatike. 2013. TPNC, Vol. 8273, pp. 133–144. URL:link.springer.com/chapter/10.1007/978-3-642-45008-2_11

15. Ruseczkij A. Algoritm robastnogo obucheniya, osnovanny`j na iterativnom metode naimen`shej mediany` kvadratov. 2012. Neural Process Lett 36 (2): 145–160. URL: link.springer.com/article/10.1007/s11063-012-9227-z

16. Nigsh, Florian; Bender, Andreas; van Byuren, Bernd; Tissen, Jos; Nigsh, E`duard; Mitchell, Dzhon B. O. Zhurnal ximicheskoj informacii i modelirovaniya. 2006, 46 (6): 2412–2422. DOI: 10.1021 ci060149f. PMID 17125183.

17. Xoll, Piter; Se`mvort, Richard Dzh., Annaly` statistiki. 2008. 36 (5): 2135–2152. arXiv: 0810.5276. Bibcode: 2008arXiv0810.5276H. DOI: 10.1214 07-AOS537. S2CID 14059866.

18. Bremner, De`vid; Demejn, E`rik; E`rikson, Dzheff; Yakono, Dzhon; Langerman, Stefan; Morin, Pat; Tussen, Godfrid T. Diskretnaya i vy`chislitel`naya geometriya. 2005, 33 (4): 593–604. DOI: 10.1007 / s00454-004-1152-0.

19. Zhuravlev Yu. I., Ryazanov V. V., Sen`ko O. V. «Raspoznavanie». Matematicheskie metody`. Programmnaya sistema. Prakticheskie primeneniya. ["Recognition". Mathematical methods. Software system. Practical applications]. M.: Fazis, 2006. ISBN 5-7036-0108-8.