МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ФАКУЛЬТЕТ ПРИКЛАДНОЙ МАТЕМАТИКИ И ИНФОРМАТИКИ

Кафедра теории вероятностей и математической статистики

Панин Кирилл Романович

РАЗРАБОТКА МЕТОДА ПОСТРОЕНИЯ АНСАМБЛЯ КЛАССИФИКАТОРОВ ДЛЯ БИОМЕДИЦИНСКИХ ПРИЛОЖЕНИЙ

Магистерская диссертация

специальность 1-31 81 12 «Прикладной компьютерный анализ данных»

Научный руководитель:
Наталья Анатольевна. Новоселова,
в.н.с., кандидат технических наук

Допущена к защите

«____» _____ 2017 г.

Зав. кафедрой ТВИМС

_____ Н.Н. Труш

доктор физ.-мат. наук, профессор

Минск, 2017

ОГЛАВЛЕНИЕ

РЕФЕРАТ

Магистерская диссертация, 75 с., 12 рис., 7 табл., 1 прил., 33 источников.

АНСАМБЛЬ КЛАССИФКАТОРОВ, ДЕРЕВЬЯ РЕШЕНИЙ, АДАБУСТ, БЭГГИНГ, СЛУЧАЙНЫЙ ЛЕС, МГК, ВРАЩАЮЩИЙСЯ ЛЕС

Объект исследования: популярные ансамбли классификаторов.

Цель работы: сравнительный анализ эффективности популярных методов ансамблей классифкаторов с предложенным методом вращающихся деревьев.

Методы исследования: методы математической статистики и теории вероятности, методы классификации, методы машинного обучения и интеллектуального анализа данных.

Результат: результаты экспериментов по сравнительному анализу ансамблей, способы оценки эффективности классификаторов и их ранжирования согласно критерию эффективности.

Область применения: задачи классификации в области медицины, биоинформатики

ABSTRACT

The master's thesis, 75 pages, 12 figures., 7 t., 1 app., 33 literature references.

Keywords: CLASSIFIER ENSEMBLES, DECISIONS TREES, ADABOOST, BAGGING, RANDOM FOREST, PCA, ROTATION FOREST.

Research object: popular ensembles of classifiers.

Purpose of the degree work: comparative analysis of the effectiveness of popular methods of ensembles of classifiers with the proposed method of rotation forest.

Research methods: methods of mathematical statistics and probability theory, classification methods, methods of machine learning and data mining.

The results of the work are the results of experiments on the comparative analysis of ensembles, methods for assessing the effectiveness of classifiers and their ranking according to the criterion of effectiveness.

The results can be applied and used in classification tasks in the field of medicine, bioinformatics.

ВВЕДЕНИЕ

Основной задачей в области распознавания образов и машинного обучения является задача классификации объектов данных. Объекты данных как правило описываются набором признаков и представляют собой точки признакового пространства. В случае, когда метки классов для некоторой выборки данных известны, задача заключается в построении классификационной модели (классификатора), которая будет использоваться для предсказания меток классов новых объектов данных. Данная задача решается с использованием различных алгоритмов и методов статистики и машинного обучения и представляет собой задачу обучения с учителем.

Существует большое количество как параметрических, так и непараметрических методов классификации. В основе построения всех классификационных моделей лежит Байесовская теория принятия решений [1], основанная на правиле Байеса, которое определяет оценку апостериорной вероятности принадлежности объекта классу. Параметрические классификаторы используют обучающие данные для оценки параметров некоторой предопределенной вероятностной функции распределения объектов данных в классах. К ним относятся классификаторы, построенные с использованием квадратичного и линейного дискриминантного анализа, наивный Байесовский классификатор. К непараметрическим классификаторам относится классификатор k-ближайших соседей (k-nn). Такие классификаторы как машины опорных векторов, нейронные сети, деревья решений напрямую определяют наилучшую разделяющую границу классов.

Однако с развитием высокопроизводительных информационных систем и накоплением большого объема данных в различных прикладных областях, включая биомедицинские исследования, повысилась сложность решаемых задач. При решении сложных задач построение единственной классификационной модели является недостаточным для получения приемлемой точности классификации. В этом случае имеет смысл строить комбинации классификаторов

(ансамбль классификаторов) в предположении, что их ошибки будут взаимно компенсироваться. Разработка алгоритмов построения комбинаций классификаторов является активной областью исследований в области машинного обучения и распознавания образов [2-8].

Согласно литературным источникам [9-10] использование комбинации классификаторов позволяет повысить точность классификации при решении практических задач. Среди всех имеющихся методов построения ансамбля классификаторов наиболее популярными являются «bagging» и «boosting» [11], которые основаны на манипуляциях с исходным обучающим множеством с целью построения нескольких классификаторов. Похожим на бэггинг методом является метод случайных подпространств (random subspace method, RSM [12]), основанный на создании вариативности при обучении с помощью выбора случайных подмножеств признаков. Широко известным примером использования бэггинга и RSM является случайный лес [13].

Индивидуальные классификаторы, составляющие основу ансамбля, называются базовыми классификаторами. Теоретические и эмпирические результаты показывают, что результат комбинации классификаторов наиболее эффективен, когда базовые классификаторы являются независимыми [2], т.е. классификаторы делают ошибки на различных объектах данных. Переход к применению ансамблей классификаторов для решения сложных задач можно обосновать тем, что практически невозможно найти универсальный алгоритм (классификатор), который бы был эффективен для решения любой задачи. Как правило, обучающий алгоритм является экспертом только для некоторых локальных задач или подмножеств входных данных. Ансамбль классификаторов позволяет исследовать и использовать локальные различия в поведении базовых классификаторов с целью улучшения точности и надежности всей обучающей системы. Теоретическим обоснованием эффективности ансамблей по сравнению с отдельными классификаторами является теорема Кондорсье о жюри присяжных, доказанная в XVIII веке. Согласно теореме вероятности ошибки ансамбля при объединении нескольких слабых классификаторов (ошибка на обучающей выборке меньше 50%, но более 0%) меньше ошибки отдельного классификатора.

Пусть имеется $L = 21$ классификаторов с ошибкой $p = 0.3$, причем ошибки независимы. Тогда общая ошибка ансамбля, полученного методов большинства голосов равна

$$P_{error} = \sum_{(i=\lceil L/2 \rceil)}^{L} C_L^i (1-p)^{L-i} \Rightarrow P_{error} = 0.026 < p = 0.3 \quad . \qquad (1)$$

В случае, если условие независимости не выполняется и ошибки коррелированы, то нет гарантии снижения ошибки при использовании ансамбля классификаторов. Однако последние исследования показали, что не всегда независимые базовые классификаторы имеют преимущество перед классификаторами с коррелированными ошибками. Для построения эффективного ансамбля необходимо определить оптимальное соотношение между независимостью и точностью базовых классификаторов [2].

В настоящей работе рассмотрены основные варианты ансамблей классификаторов, предложен новый вариант ансамбля на основе комбинации деревьев решений, основанный на методе вращающихся деревьев [14] и представляющий собой его модифицированную версию. Метод заключается в осуществлении повторных случайных подвыборок из обучающего множества с последующим применением метода главных компонент и построения каждого базового классификатора на трансформированном признаковом пространстве, что позволяет снизить степень корреляции между ошибками отдельных классификаторов. Выполнен сравнительный анализ различных методов построения ансамблей классификаторов на наборах данных из архивов по машинному обучению [15], и [29], учитывающий анализ зависимости точности ансамблей от количества базовых классификаторов. В заключении представлены выводы, полученные по результатам анализа.

ГЛАВА 1 ОБЗОР МЕТОДОВ ПОСТРОЕНИЯ АНСАМБЛЯ КЛАССИФИКАТОРОВ

1.1 Задача классификации и формальное определение ансамбля классификаторов

Задача классификации являются одной из наиболее важных задач анализа данных, и состоит в отнесении некоторого объекта, характеризующегося набором признаков, к одному из нескольких предопределенных классов. Пусть $C = \{c_1, \ldots, c_K\}$ является множеством меток класса, которому можно поставить в соответствие множество $\{1, 2, \ldots, K\}$ и $X = [x_1, \ldots, x_m]^T \in R^m$ – множество признаков, описывающих объекты данных.

Классификатором является отображение следующего вида:

$$F : R^m \to \{1, 2, \ldots, K\} \tag{2}$$

В этом случае классификатор представляет собой разбиение признакового пространства $X \in R^m$ на K непересекающихся областей или подмножеств $A_1, A_2, \ldots, A_K$, таких, что для объекта $x = (x_1, \ldots, x_m) \in A_k$ определяется класс $\hat{y} = k$. Классификационная функция, допускающая возможность перекрывающихся классов представляется как

$$F : R^m \to [0,1]^K, \tag{3}$$

где $F(x)$ – K-мерный вектор, i-ый компонент которого определяет степень принадлежности объекта x классу $c_i, i = 1, \ldots, K$.

Классификаторы обучаются с использованием специально разработанных алгоритмов на обучающей выборке $L = \{(x_1, y_1), \ldots, (x_{n_L}, y_{n_L})\}$, где n_L - количество элементов в выборке, $y_i, i = 1, \ldots, n_L$ - значение зависимой переменной для объекта обучающей выборки $x_i = (x_{i1}, x_{i2}, \ldots, x_{im})$, описываемого значениями признаков или предикторов. После определения параметров классификационной модели она

используется для предсказания классов тестовой выборки данных или тестового множества $T = \{ \mathrm{x}_1, \mathrm{x}_2, \mathrm{K}, \mathrm{x}_{n_r} \}$, где n_r - количество объектов тестовой выборки. В случае, когда метки классов элементов тестовой выборки известны, можно оценить ошибку классификации путем сравнения действительных и предсказываемых меток класса.

Классификационные модели можно рассматривать с точки зрения теории вероятности, рассматривая переменную класса $c_k, k = 1, \mathrm{K}, K$ как случайную переменную с априорной вероятностью $P(c_k)$. Тогда апостериорная вероятность события принадлежности наблюдаемого объекта x классу k определяется с использованием правила Байеса

$$P(c_k \mid \mathrm{x}) = \frac{p(\mathrm{x} \mid c_k)\, P(c_k)}{\sum_{j=1}^{K} p(\mathrm{x} \mid \mathrm{c}_j)\, P(c_j)}, \tag{4}$$

где $p(\mathrm{x} \mid c_k)$ - плотность вероятности случайной величины x, при условии класса c_k. Согласно правилу Байеса в качестве метки класса объекта x выбирается класс с наибольшей апостериорной вероятностью

$$c^* = \arg\max_{c_k} P(c_k \mid \mathrm{x}). \tag{5}$$

Апостериорная вероятность принадлежности к классу отражает уверенность в предсказании, т.е. чем ближе значение к единице, тем больше уверенность. Данный классификатор минимизирует функцию риска при симметричной функции потерь. Для функции потерь общего вида L правило классификации следующее

$$c^* = \arg\max_{c_k} \sum_{h=1}^{K} L(h, c_k) P(h \mid \mathrm{x}). \tag{6}$$

Байесовский классификатор является основой методов построения многих классификаторов, включая методы дискриминантного анализа, наивный Байесовский классификатор, где оцениваются условные функции плотности вероятности для каждого класса. А также методов, где напрямую оценивается апостериорная вероятность принадлежности к классу $P(c_k \mid \mathrm{x})$, а именно

логистическая регрессия, нейронные сети, классификационные деревья, классификатор k-ближайших соседей и т.д.

В связи с тем, что все анализируемые в работе ансамбли классификаторов в качестве базового классификатора используют деревья решений, то в разделе 1.2 более подробно описана работа этого метода.

Для построения ансамбля классификаторов в качестве базового обычно используется классификатор, который является наиболее неустойчивым, т.е. небольшие изменения обучающей выборки приводят к изменениям классификационной модели и, следовательно, результатов классификации. К таким классификаторам можно отнести деревья решений и нейронные сети. Области классификации, которые получаются при разбиении в каждом узле дерева являются результатом разбиения по значению единственного признака, отобранного в узле. Это приводит к значительным изменения областей классификации при малых изменения в данных. Области классификации, получаемые с использованием нейронных сетей изменяются не только как результат небольших изменений в данных, но также зависит и от инициализации обучающего алгоритма. В нашем исследовании в качестве базовых классификаторов для всех ансамблей используются деревья решений.

Пусть классификатор определен с использованием функции (3), и с использованием некоторого алгоритма построены B базовых классификаторов ансамбля. Для построения ансамбля классификаторов выходы B отдельных классификаторов агрегируют следующим образом:

$$F(x) = A(F_1(x), \ldots, F_B(x)), \tag{7}$$

где A – оператор агрегирования. Выход каждого отдельного классификатора для объекта данных x представляет собой K-мерный вектор $F_i(x) = [f_{i,1}(x), \ldots, f_{i,K}(x)]^T, i = 1, 2, \ldots, B$. Выходом ансамбля классификаторов в свою очередь является K-мерный вектор вида – $F(x) = [g_1(x), \ldots, g_K(x)]^T$. Отбор метки класса c_s для объекта данных x осуществляется согласно максимальному значению степеней принадлежности:

$f_{i,s}(x) \leftarrow f_{i,j}(x)\ \forall j = 1,\ldots,K$ - для отдельных классификаторов;

$g_s(x) \leftarrow g_t(x),\ \forall t = 1,\ldots,K$ - для всего ансамбля.

Существует различные операторы, позволяющие комбинировать выходы отдельных классификаторов ансамбля. К ним относятся: оператор максимума, минимума, произведения, усреднения, решение «большинством голосов» и т.д. В нашем исследовании отдельные классификаторы комбинируется с использованием метода «большинством голосов», который является достаточно популярным и простым в реализации.

Пусть K-мерный вектор $F_i(x) = [f_{i,1}(x),\ldots, f_{i,K}(x)]^T \in [0,1]^K$ представляет собой выход отдельного классификатора $F_i, i = 1,\ldots,B$ для объекта данных x. Значение $f_{i,j}(x) \in [0,1]$ является степенью принадлежности v классу c_j и определяется с использованием классификатора F_i. Для того чтобы определить «голос» классификатора в поддержку единственного класса мы огрубляем классификационное решение, а именно, выбираем класс:

$$c_s \leftarrow f_{i,s}(x) = \max_j \{ f_{i,j}(x) \}. \quad (8)$$

Таким образом, классификационное решение для каждого F_i формулируется как бинарный вектор F_i^h имеющий единицу в позиции s и ноль в остальных позициях:

$$f_{i,j}^h(x) = \begin{cases} 1, & j = s \\ 0, & j \neq s \end{cases}. \quad (9)$$

Решение «большинством голосов» A_{maj} представляет собой K-мерный вектор и рассчитывается следующим образом:

$$A_{maj} \leftarrow F(x) = [f_1(x),\ldots, f_K(x)]^T, \quad f_j(x) \in \{0,1\},\ j=1,\ldots,K$$

и

$$f_j(x) = \begin{cases} 1, & \sum_{i=1}^{B} f_{i,j}^h(x) = \max_{s=1,\ldots,K} \sum_{i=1}^{B} f_{i,s}^h(x) \\ 0, & \text{иначе} \end{cases}, \quad (10)$$

где B – количество отдельных классификаторов, составляющих ансамбль.

В общем случае выход ансамбля классификаторов можно также записать следующим образом

$$c^* = \arg\max_{c_k} \sum_{i=1}^{B} w_i I(F_i(x) = c_k), \quad (11)$$

где $I(.)$ -индикаторная функция, равная 1 при выполнении условия в скобках, и 0 в противном случае. При $w_i = 1, i = 1,\mathrm{K}, B$ выражение (11) сводится к выражению (10). Для оценки надежности предсказания для ансамблей классификаторов вводится понятие силы предсказания

$$PV(\mathrm{x}) = \frac{\max_{c_k} \sum_{i=1}^{B} w_i I(F_i(x) = c_k)}{\sum_{i=1}^{B} w_i}. \quad (12)$$

При $w_i = 1, i = 1,\mathrm{K}, B$ сила предсказания представляет собой пропорцию голосов за выигравший класс, вне зависимости от корректности предсказания.

1.2 Описание методов деревьев решений.

Деревья решения – это инструмент поддержки принятия решений, способ представления правил в иерархической, последовательной структуре, где каждому объекту соответствует единственный узел, дающий решение. На рисунке 2.1 представлен наглядный пример структуры и элементы дерева решений.

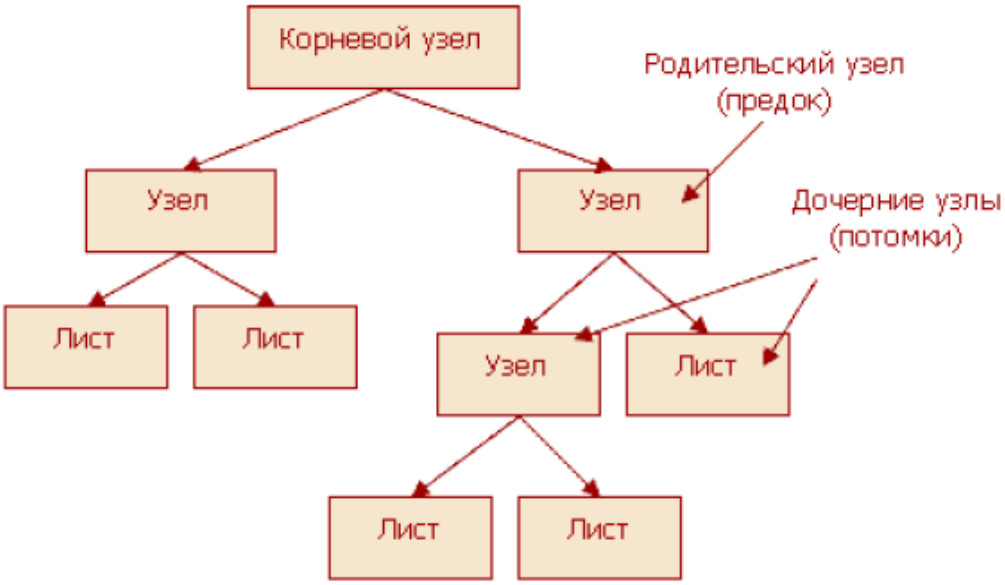

Рисунок 2.1. Элементы дерева решений

Область применения деревьев решений:

- Описание данных. Деревья решений позволяют хранить информацию о данных в компактной форме, вместо них мы можем хранить дерево решений, которое содержит точное описание объектов.
- Классификация. Деревья решений отлично справляются с задачами классификации, т.е. отнесения объектов к одному из заранее известных классов. Целевая переменная должна иметь дискретные значения.
- Регрессия. Если целевая переменная имеет непрерывные значения, деревья решений позволяют установить зависимость целевой переменной от независимых(входных) переменных. Например, к этому классу относятся задачи численного прогнозирования (предсказания значений целевой переменной).

Существует несколько вариантов деревьев решений. Наиболее популярные в использовании является алгоритмы CART [16] и C4.5[30]. CART (Classification and Regression Tree) – это алгоритм построения бинарного дерева решений – дихотомической классификационной модели. Каждый узел дерева при разбиении имеет только двух потомков. Как видно из названия алгоритма, решает задачи классификации и регрессии. C4.5 – алгоритм построения дерева решений,

количество потомков у узла не ограничено. Не умеет работать с непрерывным целевым полем, поэтому решает только задачи классификации.

Рассмотрим алгоритм CART подробнее. Согласно этому методу при построении дерева в каждом его узле выполняется полный перебор всех возможных бинарных разбиений множества объектов с целью поиска разбиения с максимальным значений критерия, отвечающего за качество разбиения. В качестве такого критерия используется Gini индекс загрязнений данных, который равен нулю, в случае когда узел дерева содержит объекты только одного класса (чистый узел) и достигает максимального значения при равной количестве объектов каждого класса. Для построения оптимального классификационного дерева используется алгоритм перекрестной проверки (кроссвалидации). Согласно этому алгоритму весь набор данных разбивается на N равных подмножеств, из который N-1 подмножеств по очереди используются для построения дерева, оставшееся при этом неиспользованное подмножество выступает в качестве тестового для оценки ошибки классификации. Таким образом, каждое подмножество используется N-1 раз как обучающее и один раз как тестовое множество. Ошибка классификации рассчитывается как среднее значений, полученное на N тестовых множествах. Результатом применения алгоритма обрезки дерева является множество деревьев, полученное из исходного дерева путем последовательного отсечения его ветвей вплоть до корневого узла. Для каждого обрезанного дерева рассчитывается ошибка классификации на обучающем множестве и ошибка, полученная методом перекрестной проверки. Дерево наименьшего размера с минимальной ошибкой перекрестной проверки выбирается в качестве оптимального дерева. Далее кратко представлена алгоритмическая схема работы метода CART.

Алгоритм CART

1. Вся обучающая выборка помещается в корневой узел дерева.
2. Выполняется полный перебор всех значений множества признаков для поиска оптимального разбиения S_{opt} множества объектов данных в родительском узле.

3. Множества объектов данных разбивается на два подмножества τ_l и τ_r.

4. Если условие останова не выполняется переходим к п.2, иначе завершаем работу алгоритма. В качестве условия останова алгоритма используется пороговое значение на количество объектов в дерева или чистота узла дерева.

В каждом узле дерева поиск оптимального бинарного разбиения выполняется следующим образом:

1. Множество значений признака A разбивается по значению r, где r выбирается как некоторое промежуточное значение между соседними уникальными значениями числового признака и как $r \in D$, где D подмножество возможных значений категориального признака;

2. При наличии в данных K классов, $p(j \mid \tau)$ определяется как процентное отношение класса j в узле τ, и мера загрязнения узла τ рассчитывается с использованием функции ϕ как

$$v(\tau) = \varphi(p(1 \mid \tau), ..., p(K \mid \tau)) \quad (13)$$

3. Изменение загрязнения узла при выполнении разбиения s, которое разделяет множество объектов S на два подмножества p_R и p_L рассчитывается как:

$$\Delta v(s,\tau) = v(\tau) - p_R \, \varphi(\tau_R) - p_L \, \varphi(\tau_L) \quad (14)$$

4. Среди всех возможных разбиений выбирается разбиение с минимальным значением $\Delta v(s,\tau)$.

Далее рассмотрим алгоритм C4.5.Для того, чтобы с помощью C4.5 построить решающее дерево и применять его, данные должны удовлетворять нескольким условиям. Информация об объектах, которые необходимо классифицировать, должна быть представлена **в виде конечного набора признаков (атрибутов), каждый из которых имеет дискретное или числовое значение. Такой набор атрибутов назовём объектом. Для всех объектов количество атрибутов и их состав должны быть постоянными. Множество классов, на которые будут**

разбиваться объекты, должно иметь конечное число элементов и каждый объект должен однозначно относиться к конкретному классу. Для случаев с нечёткой логикой, когда примеры принадлежат к классу с некоторой вероятностью, C4.5 неприменим. В обучающей выборке количество примеров должно быть значительно больше количества классов, к тому же каждый пример должен быть заранее ассоциирован со своим классом. По этой причине C4.5 является вариантом машинного обучения с учителем. **Представляет собой усовершенствованный вариант алгоритма** ID**3.** Среди улучшений стоит отметить следующие: возможность работать не только с категориальными атрибутами, но также с числовыми. **Для этого алгоритм разбивает область значений независимой переменной на несколько интервалов и делит исходное множество на подмножества в соответствии с тем интервалом, в который попадает значение зависимой переменной. После построения дерева происходит усечение его ветвей. Если получившееся дерево слишком велико, выполняется либо группировка нескольких узлов в один лист, либо замещение узла дерева нижележащим поддеревом. Перед операцией над деревом вычисляется ошибка правила классификации, содержащегося в рассматриваемом узле. Если после замещения (или группировки) ошибка не возрастает (и не сильно увеличивается энтропия), значит замену можно** произвести без ущерба для построенной модели. Один из недостатков алгоритма ID3 является то, что он некорректно работает с атрибутами, имеющими уникальные значения для всех объектов из обучающей выборки. Для таких объектов информационная энтропия равна нулю и никаких новых данных от построенного дерева по данной зависимой переменной получить не удастся. Поскольку получаемые после разбиения подмножества буду содержать по одному объекту. Алгоритм C4.5 решает эту проблему путём введения нормализации. Оценивается не количество объектов того или иного класса после разбиения, а число подмножеств и их мощность (число элементов).

Алгоритм C4.5

Пусть задано множество объектов T, где каждый элемент этого множества описывается m атрибутами. Количество объектов во множестве T будем называть

мощностью этого множества и будем обозначать |T|. Пусть метка класса принимает следующие значения $C_1, C_2 \ldots C_k$.

Задача заключатся в построении иерархической классификационной модели в виде дерева из множества примеров T. Процесс построения дерева будет происходить сверху вниз. Сначала создается корень дерева, затем потомки корня и т.д.

На первом шаге имеется пустое дерево (имеется только корень) и исходное множество T (ассоциированное с корнем). Требуется разбить исходное множество на подмножества. Это можно сделать, выбрав один из атрибутов в качестве проверки. Тогда в результате разбиения получаются n(по числу значений атрибута) подмножеств и, соответственно, создаются n потомков корня, каждому из которых поставлено в соответствие свое подмножество, полученное при разбиении множества T. Затем эта процедура рекурсивно применяется ко всем подмножествам (потомкам корня) и т.д.

Рассмотрим подробнее критерий выбора атрибута, по которому должно пойти ветвление. Очевидно, что в нашем распоряжении m (по числу атрибутов) возможных вариантов, из которых мы должны выбрать самый подходящий. Некоторые алгоритмы исключают повторное использование атрибута при построении дерева, но в нашем случае мы таких ограничений накладывать не будем. Любой из атрибутов можно использовать неограниченное количество раз при построении дерева.

Пусть имеется некоторый атрибут X, который принимает n значений $A_1, A_2 \ldots A_n$. Тогда разбиение T по проверке X даст нам подмножества $T_1, T_2 \ldots T_n$, при X равном соответственно $A_1, A_2 \ldots A_n$. При определении критерия разбиения используется информация о том, каким образом классы распределены в множестве T и его подмножествах, получаемых при разбиении по X.

Пусть $freq(C_j, S)$ – количество примеров из некоторого множества S, относящихся к одному и тому же классу C_j. Тогда вероятность того, что случайно выбранный пример из множества S будет принадлежать к классу C_j:

$$P = \frac{freq(C_j, S)}{|S|} \tag{15}$$

Согласно теории информации, количество содержащейся в сообщении информации, зависит от ее вероятности:

$$\log_2 \left(\frac{1}{p} \right) \tag{16}$$

Поскольку мы используем логарифм с двоичным основанием, то выражение (16) дает количественную оценку в битах.

$$Info(T) = -\sum_{j=1}^{k} \frac{freq(C_j, T)}{|T|} \log_2 \left(\frac{freq(C_j, T)}{|T|} \right) \tag{17}$$

Выражение (17) дает оценку среднего количества информации, необходимого для определения класса примера из множества T. В терминологии теории информации выражение (17) называется энтропией множества T.

Ту же оценку, но только уже после разбиения множества T по X, дает следующее выражение:

$$Info_X(T) = \sum_{i=1}^{n} \frac{|T_i|}{|T|} Info(T_i) \tag{18}$$

Тогда критерием для выбора атрибута будет являться следующая формула:

$$Gain(X) = Info(T) - Info_X(T) \tag{19}$$

Критерий (19) рассчитывается для всех атрибутов. Выбирается атрибут, максимизирующий данное выражение. Этот атрибут будет являться проверкой в текущем узле дерева, а затем по этому атрибуту производится дальнейшее построение дерева. Т.е. в узле будет проверяться значение по этому атрибуту и дальнейшее движение по дереву будет производиться в зависимости от полученного ответа.

Такие же рассуждения можно применить к полученным подмножествам $T_1, T_2 \ldots T_n$ и продолжить рекурсивно процесс построения дерева, до тех пор, пока в узле не окажутся примеры из одного класса.

Одно важное замечание: если в процессе работы алгоритма получен узел, ассоциированный с пустым множеством (т.е. ни один пример не попал в данный узел), то он помечается как лист, и в качестве решения листа выбирается наиболее часто встречающийся класс у непосредственного предка данного листа.

Здесь следует пояснить почему критерий должен максимизироваться. Из свойств энтропии нам известно, что максимально возможное значение энтропии достигается в том случае, когда все его сообщения равновероятны. В нашем случае, энтропия (18) достигает своего максимума когда частота появления классов в примерах множества T равновероятна. Нам же необходимо выбрать такой атрибут, чтобы при разбиении по нему один из классов имел наибольшую вероятность появления. Это возможно в том случае, когда энтропия (18) будет иметь минимальное значение и, соответственно, критерий (19) достигнет своего максимума.

В случае с числовыми атрибутами, следует выбрать некий порог, с которым должны сравниваться все значения атрибута.

Пусть числовой атрибут имеет конечное число значений. Обозначим их $\{v_1, v_2 ... v_n\}$. Предварительно отсортируем все значения. Тогда любое значение, лежащее между v_i и v_{i+1}, делит все примеры на два множества: те, которые лежат слева от этого значения $\{v_1, v_2 ... v_i\}$, и те, что справа $\{v_{i+1}, v_{i+2} ... v_n\}$. В качестве порога можно выбрать среднее между значениями v_i и v_{i+1}:

$$TH_i = \frac{v_i + v_{i+1}}{2} \tag{20}$$

Таким образом, мы существенно упростили задачу нахождения порога, и привели к рассмотрению всего n-1 потенциальных пороговых значений $TH_1, TH_2 ... TH_{n-1}$.

Формулы (17), (18) и (19) последовательно применяются ко всем потенциальным пороговым значениям и среди них выбирается то, которое дает максимальное значение по критерию (19). Далее это значение сравнивается со значениями критерия (19), подсчитанными для остальных атрибутов. Если

выяснится, что среди всех атрибутов данный числовой атрибут имеет максимальное значение по критерию (19), то в качестве проверки выбирается именно он.

Следует отметить, что все числовые тесты являются бинарными, т.е. делят узел дерева на две ветви.

Итак, мы имеем дерево решений и хотим использовать его для распознавания нового объекта. Обход дерева решений начинается с корня дерева. На каждом внутреннем узле проверяется значение объекта Y по атрибуту, который соответствует проверке в данном узле, и, в зависимости от полученного ответа, находится соответствующее ветвление, и по этой дуге двигаемся к узлу, находящему на уровень ниже и т.д. Обход дерева заканчивается как только встретится узел решения, который и дает название класса объекта Y.

Отличия CART и C4.5.

Таблица 1.1 - Отличия алгоритмов CART и C4.5

CART	C4.5
Использует неопределенность Джини	Использует приток информации к сегменту данных в процессе создания дерева решений
Использует механизм отсечения дерева при прореживании. Начиная с низа дерева, CART оценивает ошибку классификации в узле и вне узла. Если погрешность превышает граничную, то ветка отбрасывается	использует однопроходной метод прореживания, чтобы уменьшить переобучение.
Узлы решения имеют две ветки	узлы дерева решений могут иметь две или более ветвей.
использует суррогатные переменные, чтобы передать отсутствующие данные «детям»	распределяет отсутствующие значения между «детьми» на основе вероятностей.

Далее рассмотрим основные подходы построения ансамблей классификаторов, а также представим предложенный вариант ансамбля на основе комбинации деревьев решений, основанный на методе вращающихся деревьев.

1.3 Метод бэггинг

Наиболее популярным методом этой группы является непараметрический бэггинг. В этом случае из обучающей выборки формируются случайные подвыборки с возвращением такого же размера как обучающая выборка. Для каждой сформированной подвыборки строится классификатор, который представляет собой базовый классификатор ансамбля. Все полученные базовые классификаторы агрегируют в ансамбль с использованием метода большинства голосов, при этом для предсказания класса объекта x используется выражение (10) при $w_i = 1, i = 1,\mathrm{K}, B$.

Как описано в [16] метод бэггинг позволяет оценить обобщающую ошибку ансамбля классификаторов без использования отдельной тестовой выборки. Каждая формируемая случайная подвыборка L_b по оценкам из [16] включает только около 63% из объектов исходного обучающего множества. Таким образом, оставшиеся 37% объектов обучающей выборки используются в качестве тестового множества для классификатора $F_i, i = 1,\mathrm{K}, B$, построенного на сформированной подвыборке. Такого рода тестовые множества называют out-of-bag множествами и позволяют оценить точность классификации ансамбля следующим образом

$$\frac{1}{n_L}\sum_i I(y_i = \arg\max_{c_k} \sum_{\{b:(x_i,y_i)\notin L_b\}} I(F(\mathrm{x}_i, L_b) = c_k)), \qquad (21)$$

где n_L - количество объектов исходной обучающей выборки, $F(\mathrm{x}_i, L_b)$ - результат классификации объекта x_i с использованием классификатора, построенного на множестве L_b. Для каждого объекта (x_i, y_i) обучающего множества L предсказание осуществляется путем агрегирования классификаторов

$F(\mathbf{x}, L_b)$ таких что $(\mathrm{x}_i, y_i) \in L_b$. Ошибка классификации, полученная с использованием (21) называется out-of-bag ошибкой.

1.4 Метод бустинг

Метод бустинг [5] основан на осуществлении адаптивных подвыборок из обучающего набора данных, которые определяются весами объектов данных. На каждой последующей итерации соответствующего алгоритма бустинга веса увеличиваются для неправильно классифицированных объектов данных. Агрегирование классификаторов выполняется с использованием взвешенного голосования. Пусть для обучающего множества $L = \{(\mathrm{x}_1, y_1), \ldots, (\mathrm{x}_{n_L}, y_{n_L})\}$ множество $\{p_1, \ldots, p_{n_L}\}$ представляется собой вероятности (веса) каждого объекта данных. В начале работы алгоритма веса одинаковы и равны $p_i = 1/n_L$. Схема работы алгоритма для шага b представлена ниже.

Алгоритм бустинга (AdaBoost)

1. С использованием вероятностей $\{p_1, \ldots, p_{n_L}\}$ случайным образом генерировать подвыборку из обучающего множества L для получения множества L_b размерности n_L.

2. Построить классификатор $F(\mathbf{x}, L_b)$ с использованием L_b.

3. Предсказать класс для каждого объекта L с использованием классификатора $F(\mathbf{x}, L_b)$ и пусть $d_i = 1$ если i-ый объект неправильно классифицирован и $d_i = 0$ в противном случае.

4. Определить

$$e_b = \sum_i p_i d_i$$
$$\beta_b = (1 - e_b) / e_b. \qquad (22)$$

и обновить вероятности генерирования объектов данных для $(b+1)$-го шага алгоритма как

$$p_i = \frac{p_i \beta_b^{d_i}}{\sum_i p_i \beta_b^{d_i}}. \quad (23)$$

После выполнения B шагов алгоритма классификаторы $F(\mathbf{x}, \mathbf{z}_1), \ldots, F(\mathbf{x}, \mathbf{z}_B)$ агрегируются с использованием взвешенного голосования, где классификатор $F(\mathbf{x}, \mathbf{z}_b)$ имеет вес $w_b = \log(\beta_b)$. В случае, если $e_b \geq 1/2$ или $e_b = 0$ вероятности объектов данных снова становятся равными и $p_i = 1/n_L$. Можно заметить, что бэггинг является частным случаем бустинга, при котором вероятности p_i являются равными на каждом шаге алгоритма и базовые классификаторы имеют равные веса при агрегировании.

1.5 Метод случайных лесов

Случайный лес (Random Forest) [13] представляют собой ансамбль классификаторов, где в качестве базовых классификаторов используются деревья решений. Каждый из классификационных деревьев строится с использованием случайной подвыборки с повторениями из обучающей выборки (множества). Случайный лес представляет собой объединение процедуры бэггинга (bagging), т.е. комбинирует нестабильные модели, и выполняет случайный отбор признаков в каждом из узлов дерева. Каждое дерево строится без использования обрезки, а процедуры бэггинга и отбора признаков обеспечивают низкую корреляцию индивидуальных деревьев. Далее представим краткое описание работы алгоритма:

Алгоритм Random Forest

1. Сформировать L случайных подвыборок $\{B_1, \ldots, B_L\}$ из обучающего множества.

2. Использовать каждую подвыборку B_k для построения классификационного дерева T_k для последующего предсказания для объектов

данных, не вошедших в B_k (out-of-bag выборка). Такие предсказания называются out-of-bag оценками.

3. Перед использованием T_k для предсказания принадлежности к классу для out-of-bag выборки рассчитывается важность каждого признака. Данная процедура выполняется путем случайной перестановки значений признака для объектов out-of-bag выборки. В этом случае степень увеличения ошибки классификации отражает важность соответствующего признака.

4. При построении каждого дерева T_k первоначально случайным образом отбираются m признаков, среди которых далее происходит поиск оптимального разбиения.

5. Окончательная out-of-bag ошибка случайного леса равна среднему значению out-of-bag оценок по всем подвыборкам.

В качестве параметров, которые можно использовать для настройки работы алгоритма случайного леса используются такие как количество базовых деревьев, минимальное количество элементов в корневых узлах дерева и количество случайным образом отбираемых признаков m. В качестве значения по умолчанию отбирается количество признаков $m = \sqrt{n}$, где n - размерность пространства признаков. Признаки отбираются случайным образом без повторения.

1.6 Описание предложенного метода построения ансамблей классификаторов.

Для получения эффективных ансамблей классификаторов необходимо построить такие базовые классификаторы, которые являются одновременно как независимыми, так и обладать достаточной точностью. Данные задачи, как правило, являются взаимоисключающими и целью является поиск приемлемого компромисса между точностью и разнообразием базовых классификаторов. Данная цель лежит в основе всех имеющихся методов построения ансамблей классификаторов. Например, в связи с тем, что каждый отдельный классификатор

при бэггинге строится на случайной выборке из данных, то распределение данных в выборке соответствует исходному распределению. Таким образом, базовые классификаторы в бэггинге имеют относительно высокую точность классификации. Разнообразие базовых классификаторов достигается за счет различий в обучающих подвыборках, которое в этом случае является более низким, чем у других методов построения ансамблей, как бустинг и случайные попространства. Для усиления разнообразия базовых классификаторов был придложен вариант случайного леса [13]. Согласно проведенным исследованиям метод бустинг, основанный на объединении «слабых» классификаторов и включающий адаптивные веса объектов данных является в среднем наилучшим методов по сравнению с методом случайных подпространств [12] и бэггинга. Однако значимость различия различных методов построения ансамблей классификаторов [17] падает при увеличении размера ансамбля, а именно количества базовых классификаторов. Таким образом, актуальной задачей является получить ансамбли высокой точности небольшого размера, которые являются менее сложными в вычислительном плане и во многих случая дают близкие к оптимальным решения. В нашем исследовании предлагается использовать в качестве альтернативы метод построения ансамбля классификаторов, являющегося модифицированной версией метода вращающихся деревьев. Идея заключается в разбиении пространства признаков на два подмножества одинакового размера и трансформация каждого из них с использованием метода PCA (анализ главных компонент), причем трансформация выполняется на случайной подвыборке из обучающего множества.

Пусть $\mathrm{x}=[x_1,\mathrm{K},x_m]$ является объектом данных, описанным вектором признаков длины m и пусть X является набором данных, содержащим обучающую выборку и представленной в виде матрицы размера n m. Пусть Y является вектором меток класса $Y=[y_1,\mathrm{K},y_n]$, где $y_i \in \{c_1,\mathrm{K},c_K\}$. Обозначим через D_1,K,D_B классификаторы ансамбля и через G - множество признаков. Первоначально необходимо выбрать количество базовых классификаторов B. Как и в методах бэггинг и случайный лес, базовые классификаторы могут обучаться

параллельно. Для построения обучающего множества для классификатора D_i выполняются следующие шаги:

1. Разбить случайным образом множество признаков G на два непересекающихся подмножества, каждое из которых содержит $M = m/2$ признаков.

2. Обозначим через $F_{i,1}$ и $F_{i,2}$ два подмножества признаков классификатора D_i. Для каждого подмножества признаков отдельно генерируем подвыборку с возвращением равную 75% обучающих данных. Выполним PCA используя M признаков в F_{ij}, $j = 1,2$ и отобранное подмножество X. Сохраним векторы коэффициентов главных компонент $a_{i,j}^{(1)}, \mathrm{K}, a_{i,j}^{(M_j)}$ размером M, $M_j \leq M$ так как некоторые собственные значения могут равняться нулю. Использование подвыборок позволяет избежать идентичных коэффициентов в случае отбора идентичного подмножества признаков для построения другого базового классификатора.

3. Сформируем из полученных векторов коэффициентов главных компонент матрицу R_i размерности $m \times (M_1 + M_2)$

$$R_i = \begin{bmatrix} a_{i,1}^{(1)}, \mathrm{K}, a_{i,1}^{(M_1)} & [0] \\ [0] & a_{i,2}^{(1)}, \mathrm{K}, a_{i,2}^{(M_2)} \end{bmatrix}. \qquad (24)$$

Для составления обучающего множества для классификатора D_i необходимо первоначально выполнить перестановку столбцов матрицы R_i для соответствия исходной последовательности признаков, тогда обучающее множество для классификатора D_i определяется как матрица XR_i^a размерности $n \times m$, где R_i^a - матрица после перестановки столбцов. Использование PCA обосновывается возможностью увеличить разнообразие базовых классификаторов, повысить их независимость. При построении обучающего множества для каждого классификатора используются все полученные главные компоненты, так как PCA не является идеальным методом отбора информативных признаков и при отбрасывании части главных компонент, можно потерять часть информативных

признаков для классификации, тем самым понизив ее точность. Случайные подвыборки из обучающего набора данных гарантируют различие базовых классификаторов в ансамбле. После получения матрицы проекций для подмножеств признаков, весь набор данных трансформируется и используется для обучения базового классификатора.

1.7 Способы оценки результатов классификации

Для сравнительного анализа различных ансамблей классификаторов необходимо выбрать способ оценки их обобщающей способности, т.е. эффективности классификации новых объектов данных.

В связи со случайной природой обучающего множества L класс, предсказываемый для объекта x с использованием построенного классификатора $F(\mathrm{x};L)$ является случайной переменной. Поэтому изменение обучающей выборки L влечет за собой изменение предсказываемого класса объекта. Следовательно, необходимым является осуществление оценки ошибки классификационной модели с целью предсказания ее поведения на объектах данных, не включенных в обучающую выборку L.

1.7.1 Кроссвалидация методом Монте-Карло

Как правило, построение классификатора осуществляется на обучающей выборке L, а оценка работы классификатора – на тестовой выборке T. В случае отсутствия тестовой выборки объекты исходного множества могут быть разбиты на два множества: обучающее множество L_1 и вариационное множество L_2. Классификатор строится на L_1, а ошибка классификации рассчитывается на L_2. Данное деление на практике осуществляется путем случайного разбиения исходного множества на два подмножества. Для сидении дисперсии оценки данная процедура может быть повторно выполнена несколько раз (например 50) с

последующим усреднением ошибки. Единственным недостатком такого подхода является снижение размера выборки для обучения. Нет общепринято рекомендации соотношения размеров обучающего и валидационного подмножества для проведения оценки ошибки классификации. Примером может служить использование 10% случайным образом отобранных объектов исходной выборки в качестве валидацимонной подвыборки. Однако при анализе различных классификаторов такой небольшой процент валидационной подвыборки недостаточен для обеспечение адекватных результатов сравнения. Для этого размер валидационной подвыборки увеличивается до одной трети от исходной выборки.

1.7.2 Блочная кроссвалидация

Метод k-блочной перекрестной проверки (CV) основан на разбиении исходной выборки L на k множеств $L_i, i = 1,\text{K}, k$ одинаковой размерности. Классификаторы строятся на обучающей подвыборке $L - L_i$ ошибки классификации рассчитываются на валидационной подвыборке L_i и усредняются по k блокам. Как правило более низкое значение k приводит к большему смещению меньшей дисперсии оценки ошибки классификации.

Частным случаем метода CV является проверка по отдельным объектам (leave-one-out CV (LOOCV)). Преимущества LOOCV в том, что каждый объект ровно один раз участвует в проверке, а длина обучающих подвыборок лишь на единицу меньше длины полной выборки. Однако LOOCV часто дает большую дисперсию ошибки классификации. Для стабильных классификаторов, таких как k-ближайших соседей, LOOCV обеспечивает качественную оценку обобщающей ошибки.

1.7.3 Оценка с использование бутстрэпинга

Согласно данной процедуре $B1$ ошибка классификации рассчитывается для объекта x_i исходной выборки которые не вошли в подвыборки, сформированные с использованием бутстрэпинга (т.е. случайным образом сформированные подвыборки с возвращением, размер которых равен размеру исходной выборки). Каждая подвыборка включает $0.632n$ наблюдений выборки размера n. Для коррекции смещения оценки ошибки классификации используется выражение $B.632 = 0.368RE + .632B1$, где RE - ошибка методом повторной подстановки (на обучающем множестве) или выражение $B.632+ = (1-\omega)RE + \omega B1$, где ω основано на величине разницы $B1 - RE$.

1.7.4 Подходы к оценке результатов классификации

1.7.4.1 Критерии для оценки результатов классификации

В общем, мы можем представить классификатор, как модель или функцию M, которая предсказывает метки класса $\hat{y}$ для данного входного объекта x:

$$\hat{y} = M(x) \tag{25}$$

,где $x = (x_1, x_2, ..., x_d)^T$ является точкой в d-мерном пространстве и $\hat{y}$ это предсказанный моделью вектор.

Чтобы построить классификационную модель M или классификатор, нам нужен обучающий набор данных, объектов, вместе с присвоенными классами. Различные классификаторы получаются в зависимости от предположений, используемых для построения модели M. Например идея метода опорных векторов SVM состоит в том, что исходные вектора переводятся в пространство с более высокой размерностью и далее в этом пространстве выполняется поиск разделяющей гиперплоскости с максимальным зазором, которые разделяют классы. Байесовский классификатор вычисляет апостериорную вероятность $P(c_k \mid \mathrm{x})$ для каждого класса c_k и предсказывает класс x, как тот, который имеет максимальную апостериорную вероятность $\hat{y} = \arg\max_{c_k}\{P(c_k \mid \mathrm{x})\}$. Как только

модель обучена, мы оцениваем ее качество на тестовом множестве, для которого мы знаем истинные классы. После этого, мы можем использовать модель для прогнозирования на данных для которых мы обычно не знаем к какому классу они принадлежат.

Рассмотрим как оценивать классификаторы и сравнивать множество классификаторов.

Пусть D тестовый набор содержащий n-объектов, в d-мерном пространстве пусть $\{c_1, c_2, ..., c_k\}$ - обозначают метки классов, а M - классификатор. Для объектов $x_i \in D$, y_i обозначает истинные классы, а $\hat{y}_i = M(x_i)$ - предсказанные классы.

Основным критерием, используемым для оценки результатов классификации является оценка его точности. Коэффициент ошибок определяет долю неверных прогнозов классификатора

$$ErrorRate = \frac{1}{n} \sum_{i=1}^{n} I(y_i \neq \hat{y}_i) \quad (26)$$

, где I индикаторная функция, которая имеет значение 1, когда аргумент равен $true$ и 0 в противном случае. Коэффициент ошибок – это оценка вероятности ошибочной классификации. Чем ниже коэффициент ошибок, тем лучше классификатор.

Точность классификатора – это доля правильных предсказаний на тестовом наборе.

$$Accuracy = \frac{1}{n} \sum_{i=1}^{n} I(y_i = \hat{y}_i) = 1 - ErrorRate \quad (27)$$

Точность дает вероятностную оценку безошибочной классификации. Таким образом, чем выше точность, тем лучше классификатор.

Кроме стандартного критерия точности использовались критерии, которые позволяют учитывать дисбаланс между количеством элементов в отдельных классах. Более информативные критерии могут быть получены путем оценки соответствия между действительными и предсказанными метками классов на

тестовом множестве. Пусть $B = \{B_1, B_2, \text{K}, B_c\}$ определяет разбиение тестового множества на группы согласно их реальной метке класса, где $B_j = \{x_i \in B \mid l_i = k_j\}$.

Пусть $|B_i| = n_i$ определяет количество объектов класса k_i. Множество $R = \{R_1, R_2, \text{K}, R_c\}$ определяет разбиение тестовых объектов на основе предсказанных меток класса, где $R_j = \{x_i \in B \mid \hat{l}_i = k_j\}$.

Пусть $|R_j| = m_j$ определяет количество объектов, предсказанных как относящиеся к классу k_j. Таким образом, с использованием множеств R и B матрица сопряженности (confusion matrix) N определяется как

$$N(i, j) = n_{ij} = |R_i \cap B_j| = \left|\{x_a \in B \mid \hat{l}_a = k \text{ и } l_a = k_j\}\right|, \text{ где } 1 \le i, j \le c. \quad (28)$$

Величина n_{ij} соответствует количеству объектов класса k_j, для которых предсказан класс k_i, причем n_{ii} - количество объектов для которых предсказание совпадает с действительной меткой класса, в остальных случаем происходит расхождение между предсказаниями и реальным классом объектов.

С использованием таблицы сопряженности рассчитывается ряд информационных критериев оценки эффективности классификаторов.

Точность для класса $c_i, i = 1, \text{K}, K$ классификатора D определяется как отношение корректных предсказаний ко всем объектам, для которых предсказан класс c_i и определяется как $acc_i = \frac{n_{ii}}{m_i}$, где m_i - количество объектов для которых предсказана метка класса c_i. Общая точность классификатора представляет собой взвешенной среднее точностей для классов

$$Accuracy = \text{Precision} = \sum_{i=1}^{k} \left(\frac{n_i}{n} acc_i\right) = \frac{1}{n} \sum_{i=1}^{k} n_{ii}. \quad (29)$$

Полнота для отдельного класса $c_i, i = 1,\text{K}, K$ является отношение правильных предсказаний ко всем объектам класса c_i и определяется как $recall_i = \frac{n_{ii}}{n_i}$, где n_i - количество объектов класса c_i. Чем выше покрытие, тем лучше классификатор.

В случае если количество классов относительно невелико (не более 100-150 классов), этот подход позволяет довольно наглядно представить результаты работы классификатора. Чем выше точность и полнота, тем лучше. Но максимальная точность и полнота не достижимы одновременно и приходится искать некий баланс. Поэтому используется комплексная метрика, объединяющая информацию о точности и полноте алгоритма классификации и позволяющая оценить эффективность реализации. В качестве такой метрики часто используется F-мера. F-мера представляет собой гармоническое среднее между точностью и полнотой. Она стремится к нулю, если точность или полнота стремится к нулю и является хорошей оценкой качества классификатора.

F-мера представляет собой соотношение между покрытием и точностью. F-мера для класса c_i определяется как

$$F_i = \frac{2}{1/prec_i + 1/recall_i} = \frac{2 \cdot prec_i \cdot recall_i}{prec_i + recall_i} = \frac{2n_{ii}}{n_i + m_i}. \quad (30)$$

Общая F-мера для классификатора D является средним значением F-мер для отдельных классов $F = \frac{1}{k} \sum_{i=1}^{K} F_i$.

Бинарная классификация: Положительный и Отрицательный класс

В случае бинарной классификации имеется два класса, k= 2 класса, где класс c_1 - положительный класс (Positive class), а класс c_2 - отрицательным классом (Negative class). В таблице 1.2 представлена матрица сопряженности для результатов бинарной классификации. Для классификации с количеством классов более двух, матрица сопряженности составляется для каждого класса отдельно.

Таблица1.2 - Матрица неточностей

	Экспертная оценка	
Оценка системы	Положительная(c_1)	Отриацательная(c_2)
Положительная(c_1)	True Positive (TP)	False Positive (FP)
Отриацательная(c_2)	False Negative (FN)	True Negative (TN)

- True Positives (TP): Количество объектов, правильно предсказанных классификатором, как положительные:

$$TP = n_{11} = |\{x_i \mid \hat{y}_i = y_i = c_1\}| \tag{31}$$

- False Positives (FP): Количество объектов, которые предсказаны классификатором как положительные, но принадлежат к отрицательному классу:

$$FP = n_{12} = |\{x_i \mid \hat{y}_i = c_1 \,\&\, y_i = c_2\}| \tag{32}$$

- False Negatives(FN): Количество объектов, которые предсказаны классификатором как отрицательные, но принадлежат к положительному классу:

$$FN = n_{21} = |\{x_i \mid \hat{y}_i = c_2 \,\&\, y_i = c_1\}| \tag{33}$$

- True Negatives (TN): Количество объектов, предсказанных классификатором, как отрицательные:

$$TN = n_{22} = |\{x_i \mid \hat{y}_i = y_i = c_2\}| \tag{34}$$

Коэффициент Ошибок. Error Rate.

Коэффициент ошибок (26) для бинарной классификации получется как доля ошибки (или ложных предсказаний):

$$ErrorRate = \frac{FP + FN}{n} \tag{35}$$

С использованием таблица 1.2 точность классификации (27) рассчитывается как доля корректных прогнозов:

$$Accuracy = \frac{TP + TN}{n} \tag{36}$$

Вышеупомянутые метрики, являются глобальными метриками качества классификации. Далее рассмотрим особые метрики качества классификации.

Точность определения класса.

Точность для положительного и отрицательного класса задается как:

$$prec_p = \frac{TP}{TP + FP} = \frac{TP}{m_1} \tag{37}$$

$$prec_N = \frac{TN}{TN + FN} = \frac{TN}{m_2} \tag{38}$$

где $m_i = |R_i|$ количество объектов предсказанные классификатором M, как класс c_i.

Чувствительность. Sensitivity: True Positive Rate

Истинно положительная оценка, также называемая чувствительностью, представляет собой долю правильных предсказаний по отношению ко всем объектам положительного класса. Данную метрику также называют, полнотой для положительного класса

$$TPR = recall_p = \frac{TP}{TP + FN} = \frac{TP}{n_1}, \tag{39}$$

где n_1-размер положительного класса.

Специфичность. True Negative Rate

Истинно-отрицательная оценка, также называемая специфичностью. Данную метрику также называют, полнотой для отрицательного класса:

$$TNR = specificity = recall_N = \frac{TN}{FP + TN} = \frac{TN}{n_2}, \tag{40}$$

где n_2 - размер отрицательного класса.

Ложно-отрицательная оценка (False Negative Rate, FNR) определяется как

$$FNR = \frac{FN}{TP + FN} = \frac{FN}{n_1} = 1 - sensitivity \tag{41}$$

Ложно-положительная оценка (False Positive Rate, FPR) определяется как

$$FPR = \frac{FP}{FP + TN} = \frac{FP}{n_2} = 1 - specificity \tag{42}$$

1.7.4.2 Анализ с использованием кривой ошибок (ROC анализ) для оценки эффективности классификации

ROC-кривая является популярной стратегией оценки качества бинарной классификации. Анализ ROC требует, чтобы классификатор выводил значения оценки score value для положительного класса для каждого набора признаков из тестового набора. Эти значения оценки или баллы затем могут использоваться для упорядочивания в порядке убывания. Для примера, мы можем использовать апостериорную вероятность $P(c_1 \mid x_i)$ как оценку например Байесовского классификатора. Для метода опорных векторов (SVM) в качестве оценки, мы можем использовать знаковое расстояние от гиперплоскости, потому что чем больше положительное расстояние (positive distances), тем выше степень доверия для c_1, а чем больше отрицательное расстояние, тем ниже степень доверия для c_1 (и следовательно выше степень доверия для отриацтельного класса c_2).

Как правило, бинарный классификатор выбирает порог положительной оценки ρ и классифицирует все объекты с оценкой (score) выше значения ρ как положительный, а остальные объекты как отрицательные. Однако такой порог является нечетким и должен подбираться для каждого набора данных индивидуально.

Вместо этого анализ ROC показывает качество классификатора по всем возможным значениям порогового параметра ρ. В частности, для каждого значения ρ ROC анализ показывает ложно-положительную оценку FPR (42) на оси x, по сравнению с истинно-положительной оценкой TPR (39) нас оси y. Полученный график называется ROC кривой или ROC графиком для классификатора.

Пусть $S(x_i)$ обозначает вещественную оценку для вывода положительного класса классификатором M для каждого объекта x_i. Пусть максимальные и минимальные пороговые значения оценок полученные на тестовом наборе данных D следующие:

$$\rho^{\min} = \min_i \{S(x_i)\} \qquad \rho^{\max} = \max_i \{S(x_i)\}$$

Первоначально мы классифицируем все объекты как отрицательные. Таким образом, TP и FP первоначально равны нулю (как показано в таблице 1.3) и это означает, что TPR и FPR равны нулю. Этим значениям соответствует точка (0 0) в левом нижнем углу на участке ROC графика. Далее, для каждого отдельного значения ρ в диапазоне $[\rho^{\min}, \rho^{\max}]$ мы приводим таблицу положительных объектов:

$$R_1(\rho) = \{x_i \in D : S(x_i) > \rho\} \qquad (43)$$

По которой вычисляются соответсвующие истинные и ложные положительные оценки(TPR и FPR) и определяются, точки на ROC графике. Наконец на последнем шаге все объекты классифицируются как положительные. Значения FN и TN равны нулю (как показано в таблице 1.4), в результате чего значения TPR и FPR равны 1. Эти результаты приводят к точке (1,1) в верхнем правом углу на графике ROC. Идеальный классификатор соответствует верхней левой точке (0,1) что соответствует случаю FPR=0 и TPR=1, т.е. классификатор не имеет неверных предсказаний (FP) и идентифицирует все истинно-положительные TP(как следствие, он также правильно предсказывает все объекты отрицательного класса). Этот случай иллюстрирует таблица 1.5. Таким образом ROC кривая указывает, в какой степени, классификатор оценивает положительные экземпляры выше, чем орицательные. Идеальный классификатор должен оценивать все положительные объекты выше любого отрицательного объекта. Таким образом, классификатор, чья кривая ближе к верхнему левому углу, является лучшим классификатором.

Таблица 1.3 - Начальное: все отрицательные

	True	
Predicted	Pos	Neg
Pos	0	0
Neg	FN	TN

Таблица 1.4 - Финальное: все положительные

	True	
Predicted	Pos	Neg
Pos	TP	FP
Neg	0	0

Таблица 1.5 - Идеальный классификатор

	True	
Predicted	Pos	Neg
Pos	TP	0
Neg	0	TN

Площадь под ROC кривой AUC (Area Under ROC Curve) может использоваться как мера эффективности классификатора. Данная мера определяет количественную интерпретацию ROC и определяет площадь, ограниченную ROC-кривой и осью x. Общая площадь ROC графика равна 1, значение AUC лежит на интервале от [0,1]. Чем выше показатель AUC, тем качественнее классификатор. Значение 0,5 демонстрирует непригодность выбранного метода классификации (соответствует случайной классификации). Значение менее 0,5 говорит о том, что классификатор действует с точностью до наоборот: если положительные объекты назвать отрицательными и наоборот, классификатор будет работать.

Алгоритм ROC / AUC:

ROC-Curve(D,M):

1. $n_1 \leftarrow |\{x_i \in D \mid y_i = c_1\}|$ // размер положительного класса

2. $n_2 \leftarrow |\{x_i \in D \mid y_i = c_2\}|$ // размер отрицательного класса

// классификация, оценка и сортировка всех тестовых объектов

3. L $\leftarrow$ *сортировка множества* $\{(S(x_i), y_i) : x_i \in D\}$

4. FP $\leftarrow$ TP $\leftarrow$ 0

5. $FP_{prev} \leftarrow TP_{prev} \leftarrow 0$

6. $AUC \leftarrow 0$

7. $\rho \leftarrow \leftarrow$

8. foreach $(S(x_i), y_i) \in L$ do

9. if $\rho > S(x_i)$ then

10. $plot\ point\left(\frac{FP}{n_2}, \frac{TP}{n_1}\right)$

11. $AUC \leftarrow AUC + Trapezoid - Area\left(\frac{FP_{prev}}{n_2}, \frac{TP_{prev}}{n_1}, \frac{FP}{n_2}, \frac{TP}{n_1}\right)$

12. $\rho \leftarrow S(x_i)$

13. $FP_{prev} \leftarrow FP$

14. $TP_{prev} \leftarrow TP$

15. if $y_i = c_i$ then $TP \leftarrow TP + 1$

16. else $FP \leftarrow FP + 1$

17. $plot\ point\left(\frac{FP}{n_2}, \frac{TP}{n_1}\right)$

18. $AUC \leftarrow AUC + Trapezoid - Area\left(\frac{FP_{prev}}{n_2}, \frac{TP_{prev}}{n_1}, \frac{FP}{n_2}, \frac{TP}{n_1}\right)$

Trapezoid-Area$((x_1, y_1), (x_2, y_2))$:

19. $b \leftarrow |x_2 - x_1|$ // Основание трапеции

20. $h \leftarrow \frac{1}{2}(y_2 + y_1)$ // Средняя высота трапеции

21. return $(b \cdot h)$

Алгоритм описывает основные шаги построения кривой ROC и вычисляет площади под кривой. В качестве входных данных требуется тестовый набор данных D и классификатор M. Первым шагом является предсказание оценки $S(x_i)$ для положительного класса (c_1) для каждого объекта из тестовой выборки $x_i \in D$. Затем пары $(S(x_i), y_i)$ сортируются в порядке убывания оценки истинных классов (строка 3). Вначале порог положительной оценки устанавливается равным $\rho = \infty$ (строка 7). Цикл (строка 8) проверяет каждую пару $(S(x_i), y_i)$ в отсортированном

порядке и для каждого отдельного значения оценки устанавливает $\rho = S(x_i)$, рассчитывает значения FP и TP и рассчитывает точку ROC кривой

$$(FPR, TPR) = \left(\frac{FP}{n_2}, \frac{TP}{n_1} \right) \tag{44}$$

Истинные и ложные положительные значения расчитываются на основе истинного класса y_i всех контрольных точек $x_i \in D$. Если $y_1 = c_1$ мы увеличиваем на 1 истинные положительные значения TP, в противном случае, мы увеличиваем ложные положительные знчения (строки 15-16). В конце цикла мы строим конечную точку кривой ROC (строка 17). Значения AUC вычисляется по мере того, как каждая новая точка добавляется к графику ROC. Алгоритм сохраняет предыдущие значения ложных и истинных положительных результатов, FP_{prev} и TP_{prev} для предыдущего порога оценки ρ. Учитывая текущие значения FP и TP, мы вычисляем площадь под кривой определяемой четырьмя точками

$$(x_1, y_1) = \left(\frac{FP_{prev}}{n_2}, \frac{TP_{prev}}{n_1} \right) \qquad (x_2, y_2) = \left(\frac{FP}{n_2}, \frac{TP}{n_1} \right)$$

$$(x_1, 0) = \left(\frac{FP_{prev}}{n_2}, 0 \right) \qquad (x_2, 0) = \left(\frac{FP}{n_2}, 0 \right)$$

Эти четыре точки определяют трапецию, когда $x_2 > x_1$ и $y_2 > y_1$, в противном случае, они определяют прямоугольник (который может быть вырожденным, с нулеовой площадью).

Функция TRAPEZOID-AREA вычисляет площадь под трапецией, которая задается как $b \cdot h$, где $b = | x_2 - x_1 |$ - длина основания трапеции и $h = \frac{1}{2}(y_2 + y_1)$ является средней высотой трапеции.

1.7.5 Проверка значимости отличий классификаторов. Сравнение классификаторов: парный t-тест

Рассмотрим метод, который позволяет проверить статистическую значимость разницы между классифкаторами. Обозначим через M_A и M_B две классификационные модели. Мы хотим оценить, какая из них имеет более высокие показатели эффективности классификации по данному набору данных D. Для этого можно применить кросс-проверку K-fold (или повторные выборки с помощью бутстрапа) и сгруппировать результаты некоторого критерия оценки эффективности классификации по каждой подвыборке для каждого классификатора. . Для оценки статистической значимости различий значений критериев эффективности выполняется парный тест, причем оба классификатора обучены и протестированы на тех же самых данных. Пусть $\theta_1^A, \theta_2^A, ..., \theta_k^A$ и $\theta_1^B, \theta_2^B, ..., \theta_k^B$ обозначают значения эффективности классификаторов M_A и M_B. соответственно. Чтобы определить, имеют ли два классификатора разную или сходную эффективность, определяет случайную величину δ_i как разность их эффективностей на i-ом наборе данных:

$$\delta_i = \theta_i^A - \theta_i^B \tag{45}$$

Теперь рассмотрим оценки ожидаемой разности: среднее и дисперсию величины δ:

$$\hat{\mu}_\delta = \frac{1}{K} \sum_{i=1}^{K} \delta_i \tag{46}$$

$$\hat{\sigma}_\delta^2 = \frac{1}{K} \sum_{i=1}^{K} (\delta_i - \hat{\mu}_\delta)^2 \tag{47}$$

Для определения, наличия статистически значимой разницы между производительностью M_A и M_B - создадим структуру тестирования гипотез. Определим следуюущую нулевую гипотезу H_0, которая заключается в том, что

эффективность классификаторов одинакова, то есть истинная ожидаемая разница равна нулю, тогда как альтернативная гипотеза H_a состоит в том, что они не совпадают, то есть истинные ожидаемая разность μ_δ не равна нулю:

$$H_0: \ \mu_\delta = 0 \qquad\qquad H_a: \mu_\delta \neq 0$$

Определим случайную величину z-оценки (z-score) для предполагаемой ожидаемой разности как:

$$Z_\delta^* = \sqrt{K}\left(\frac{\hat{\mu}_\delta - \mu_\delta}{\hat{\sigma}_\delta}\right) \tag{48}$$

При нулевой гипотезе $\mu_\delta = 0$ мы имеем

$$Z_\delta^* = \frac{\sqrt{K}\hat{\mu}_\delta}{\hat{\sigma}_\delta} : t_{K-1}, \tag{49}$$

где обозначение $Z_\delta^* : t_{K-1}$ означает, что Z_δ^* следует t-распределению с K-1 степенями свободы.

Учитывая желаемый уровень достоверности α, заключаем, что

$$P(-t_{\alpha/2,K-1} \leq Z_\delta^* \leq t_{\alpha/2,K-1}) = \alpha \tag{50}$$

Иначе говоря, если $Z_\delta^* \notin (-t_{\alpha/2,K-1} \leq Z_\delta^* \leq t_{\alpha/2,K-1})$, то мы можем отклонить нулевую гипотезу с $\alpha\%$ уверенностью. В этом случае мы заключаем, что существует значительная разница между эффективностями классификаторов M_A и M_B. С другой стороны, если Z_δ^* лежит выше доверительного интервала, то мы принимаем нулевую гипотезу о том, что M_A и M_B имеют по существу такую же производительность.

Алгоритм. Парный t-тест с помощью перекрестной проверки.

Исходные данные (α, K, D):

1. D – случайным образом перемешать D
2. $\{D_1, D_2, ..., D_k\}$ – делим D на K равных частей
3. **foreach** $i \in [1, K]$ **do**

a. M_i^A, M_i^B обучить два разных классификатора $D \setminus D_i$

b. θ_i^A, θ_i^B оценить M_i^A и M_i^B на D_i

c. $\delta_i = \theta_i^A - \theta_i^B$

4. $\hat{\mu}_\delta = \frac{1}{K} \sum_{i=1}^{K} \delta_i$

5. $\hat{\sigma}_\delta^2 = \frac{1}{K} \sum_{i=1}^{K} (\delta_i - \hat{\mu}_\delta)^2$

6. $Z_\delta^* = \frac{\sqrt{K}\hat{\mu}_\delta}{\hat{\sigma}_\delta}$

7. **if** $Z_\delta^* \in (-t_{\alpha/2,K-1}, t_{\alpha/2,K-1})$ **then**

- Принять H_0 нулевую гипотезу; Оба классификатора имеют схожие показатели

8. **else**

- Отклонить H_0 нулевую гипотезу; Классификаторы имеют значительно различную производительность

В нашем исследовании для оценки эффективности базовых классификаторов и ансамбля классификаторов использовалась оценка точности классификации $Accuracy$. Для оценки статистической значимости попарных различий эффективности классификаторов использовались t-тест, каппа Коэна и разработанные способы ранжирования.

1.8 Методы отбора признаков для построения классификационной модели

Одной из важных этапов предобработки данных является выделение наиболее информативных признаков, описывающих объекты данных. Информативность признака в задаче классификации означает, насколько данный признак характеризует состояние объекта, то есть насколько от него зависит

правильное предсказание его класса. Методы отбора признаков для задач классификации, можно разделить в первую очередь на две группы: фильтровочные и упаковочные методы [20]. Фильтровочные методы позволяют удалять неинформативные признаки согласно общим характеристикам данных. Они более просты в вычислительном плане и не привязаны к конкретному алгоритму классификации. Встроенные методы используют алгоритмы машинного обучения для отбора признаков и следовательно позволяют получить подмножества с более высокой точностью классификации с использованием классификатора, применяемого для отбора этого подмножества признаков. Недостатком встроенных методов является высокая вычислительная сложность в связи с необходимостью на каждой итерации отбора применять классификационный алгоритм. Большинство методов, применяемых для отбора информативных признаков для классификации относятся к фильтровочным методам, как например стандартные параметрические тесты – t-тест [21], непараметрические тесты - знаково-ранговый тест Вилкоксона [22].

1.8.1 Методы на основе энтропии.

Методы нацелены на удаление из рассмотрения признаков со случайным образом распределенными по классам значениями. Для оставшихся признаков эти методы позволяют автоматически определить промежуточные значения, такие что, получаемые в результате интервалы значений признаков имеют максимально различное распределение значений по классам. Если все образованные интервалы значений признака содержит объекты только одного класса, то значение энтропии равно нулю. Чем меньше значение энтропии, тем более информативным он является для классификации [23]. Значение энтропии класса Y с учетом признака с X рассчитывается по формуле:

$$H(Y \mid X) = -\sum_{x \in X} p(x) \sum_{y \in Y} p(y \mid x) \log_2 \left(p(y \mid x) \right) . \quad (51)$$

1.8.2 Методы на основе критерия Хи-квадрат и на основе расчета значений корреляции.

Данные методы построены на основе метода оценки энтропии. Хи-квадрат метод [24] оценивает отдельно каждый признак путем расчета статистики Хи-квадрат относительно класса. Для числового признака необходимо предварительная дискретизация значений признака, как например дискретизация на основе энтропии. Значение статистики Хи-квадрат для признака определяется по формуле:

$$\chi^2 = \sum_{i=1}^{m} \sum_{j=1}^{k} \frac{\left(A_{ij} - E_{ij} \right)^2}{E_{ij}}, \qquad (52)$$

где m – количество интервалов, k – количество классов, A_{ij} - количество объектов j-го класса в i-ом интервале, R_i - количество объектов в i-ом интервале, C_j - количество объектов в j-ом классе, N – общее количество объектов данных, и E_{ij} - ожидаемое значение для A_{ij}, рассчитываемое как $E_{ij} = R_i \cdot C_j / N$.

После расчета значения Хи-квадрат для всех рассматриваемых признаков выполняется сортировка значений по убыванию, т.к. большее значение соответствует более важному для классификации признаку. Степень свободы статистики Хи-квадрат равна $(m-1)(k-1)$. Так как в большинстве случаев для каждого признака определяется два интервала $m = 2$, степень свободы равна $k-1$. Далее выбираются признаки, которые имеют значение Хи-квадрат больше, чем 5% уровень значимости статистики.

Метод, основанный на расчете корреляций (CFS) [25] отличается от предыдущего метода тем, что оценивает или ранжирует не отдельные признаки. А подмножества признаков. В связи с тем, что, как правило, признаковое пространство достаточно большое, то используется поисковый алгоритм - «лучший — первый». CFS метод принимает во внимание эффективность отдельных признаков для предсказания класса совместно с уровнем их взаимной корреляции, основываясь на том, что эффективные подмножества признаков

содержат признаки, которые хорошо коррелируют с классом и не коррелируют между собой. Согласно CFS сначала на основе исходных обучающих данных рассчитывается корреляционные матрицы – «признак-признак» и «признак-класс». Затем рассчитывается значение эффективности подмножества признаков по формуле:

$$E_s = \frac{k\overline{r_{cf}}}{\sqrt{k + k(k-1)\overline{r_{ff}}}}, \quad (53)$$

где E_s – значение эффективности подмножества признаков S, содержащего k признаков, $\overline{r_{cf}}$ - среднее значение корреляции «признак-класс», $\overline{r_{ff}}$ - среднее значение взаимной корреляции «признак-признак».

Для оценки корреляции между признаков X и классом Y используется выражение для расчета симметричной неопределенности r_{ij}:

$$r_{ij} = 2\left[\frac{gain}{H(X) + H(Y)}\right], \quad (54)$$

где $gain = H(X) + H(Y) - H(X,Y)$ - информационный выигрыш от использования X для оценки Y, $H(X)$ - значение энтропии признака. Согласно методу CFS поиск начинается с пустого подмножества признаков и используется алгоритм поиска «лучший — первый» и условием остановки – пять последовательных подмножеств с неулучшающимся значением критерия эффективности E_s.

1.8.3 Методы с использованием t-статистики и MIT корреляции.

Наиболее широко используемым методом, применяемым для отбора признаков при условии двух классов является использование t – статистики [26]. Исходными данными для использования метода является набор данных S, состоящий из m векторов: $X^i = (x_1^i, \ldots, x_n^i)$, где $1 \le i \le m$, m - количество объектов данных, n - количество измеренных признаков. Каждый объект соответствует

классу $Y \in \{+1,-1\}$. Для каждого признака x_j с использованием объектов данных с меткой класса +1 (-1) рассчитываются значения μ_j^+ (μ_j^-) и стандартное отклонение δ_j^+ (δ_j^-). Затем значение статистики $T(x_j)$ рассчитывается по формуле:

$$T(x_j) = \frac{\left|\mu_j^+ - \mu_j^-\right|}{\sqrt{\frac{(\delta_j^+)^2}{n_+} + \frac{(\delta_j^-)^2}{n_-}}}, \tag{55}$$

где n_+ (n_-) - количество объектов, помеченных меткой +1 (-1). После расчета значений $T(x_j)$ для каждого признака *N* признаков с наибольшим значением выбираются в качестве наиболее информативных.

Согласно методу MIT корреляции [27] для каждого признака значение критерия эффективности рассчитывается по формуле:

$$MIT(x_j) = \frac{\left|\mu_j^+ - \mu_j^-\right|}{\delta_j^+ + \delta_j^-}. \tag{56}$$

Методы, основанные на энтропии, Хи-квадрат критерии и метод CFS устойчивы к нормализации данных. Это означает, что при применении каждого из методов будут отобраны одни и те же признаки независимо от выбранной функции нормализации, например применение логарифмирования значений признаков. Это условие не выполняется для методов на основе t-статистики и MIT корреляции.

1.8.4 Метод основанный на отношении суммы квадратов между классами и внутри классов.

Для i-го признака отношение суммы квадратов определяется как

$$\frac{BSS(j)}{WSS(j)} = \frac{\sum_i \sum_k I(y_i = c_k)\left(\bar{x}_{kj} - \bar{x}_{.j}\right)^2}{\sum_i \sum_k I(y_i = c_k)\left(x_{ij} - \bar{x}_{kj}\right)^2}, \tag{57}$$

где $\bar{x}_{.j}$ - среднее значение j-го признака по всем объектам, $\bar{x}_{kj}$ - среднее значение j-го признака по объектам класса c_k.

1.9 Построение диаграммы для оценки характеристик ансамблей классификаторов

1.9.1 Каппа Коэна

Коэффициент каппа Коэна - это статистика, которая измеряет взаимное согласие для качественных (категориальных) переменных т.е. измеряет согласие мнений двух экспертов, оценивающих одни и те же объекты. Как правило, считается, что это более надежная мера, чем простой расчет процентных соглашений, поскольку k учитывает возможность возможного возникновения соглашения. Значение 1 указывает на полное согласие. Значение 0 указывает на то, что согласие - не более чем случайность. Каппа основывается на квадратной таблице, в которой значения строк и столбцов измерены в одной и той же шкале. Любой ячейки, которая имеет наблюденные значения для одной переменной, но не имеет для другой, присваивается количество, равное 0. Каппа не вычисляется, если тип хранения данных (текстовый или числовой) не одинаков для обеих переменных. Для текстовых переменных, обе переменные должны иметь одинаковую заданную длину.

Каппа Коэна – мера согласованности между двумя категориальными переменными

X и Y. Каппа Коэна может быть использована для оценки согласованности между двумя оценщиками классифицирующими n объектов по s категориям.

Каппа Коэна k задается как:

$$k = \frac{P_0 - P_e}{1 - P_e} = 1 - \frac{1 - P_0}{1 - P_e} \tag{58}$$

где P_0 относительное наблюдаемое согласование между оценщиками(одинаково и с точностью), и P_e - гипотеттическая вероятность случайного соглашения, используя наблюдемые данные для вычисления вероятностей каждого наблюдателя, случайно наблюдающего каждую категорию, если оценщики

находятся в полном согласии то $k=1$. Если между оценщиками не существует согласия,кроме того, что можно было бы ожидать случайно (как указано P_e), k 0 . Для k категорий (групп), числа элементов N и n_{ki} количества прогнозируемых категорий вероятность случайного солашения определяется как :

$$P_e = \frac{1}{N^2} \sum_k n_{k1} n_{k2} \tag{59}$$

Приведем пример. Предположим, что анализируются данные, относящиеся к группе из 50 пациентов, проходящие обследование в клинике. Каждый пациент осмотрен двумя врачами, и каждый из специалистов оценил пациента, как здорового или больного, т.е. поставил метку 0 или 1("Да" или "Нет"). Матрица согласованности представленная в таблице 1.6, демонстрирует различия согласованность ответов двух классификаторов:A и B. Данные по главной диагонали матрицы (верхний левый и правый нижний) демонстируеют счет согласованности, а данные вне главной диагонали (верхний правый и левый нижний) -счет разногласий или несогласованности:

Таблица 1.6 – Матрица согласованности

		B	
		Да	Нет
A	Да	a	b
	Нет	c	d

Например:

Таблица 1.7 – Пример матрицы согласованности

		B	
		Да	Нет
A	Да	20	5
	Нет	10	15

Наблюдаемое пропорциональное соглашение:

$$P_0 = \frac{a+d}{a+b+c+d} = \frac{20+15}{50} = 0.7 \tag{60}$$

Для вычислений P_e вероятности случайного соглашения (the probability of random agreement) заметим что:

- Специалист А сказал 25 пациентам, что они здоровы, и 25 пациентам, что они больны. Т.е. специалист А сказал «Да» в 50% случаев.
- Специалист В сказал 30 пациентам, что они здоровы, и 20 пациентам, что они больны. Т.е. специалист В сказал «Да» в 60% случаев.

Таким образом, ожидаемая вероятность того, что оба скажут «Да» наугад равна:

$$P_{Да} = \frac{a+b}{a+b+c+d} \cdot \frac{a+c}{a+b+c+d} = 0.5 \cdot 0.6 = 0.3 \quad (61)$$

Идентично

$$P_{Нет} = \frac{c+d}{a+b+c+d} \cdot \frac{b+d}{a+b+c+d} = 0.5 \cdot 0.4 = 0.2 \quad (62)$$

Общая вероятность случайного согласования -это вероятность, того, что они соглосятся либо на «Да», либо на «Нет» т.е.

$$P_{eДа} = P_{Нет} + P = 0.3 + 0.2 = 0.5 \quad (63)$$

Итак применив формулу для каппы Коэна получим:

$$k = \frac{P_0 - P_e}{1 - P_e} = \frac{0.7 - 0.5}{1 - 0.5} = 0.4 \quad (64)$$

Если оценщики полностью согласованны, тогда $k = 1$.

Если $k > 0.75$, согласованность считается высокой,

если $0.4 < k \leq 0.75$ A - хорошой, иначе плохой.

ГЛАВА 2 ОПИСАНИЕ ПРОГРАММНОЙ РЕАЛИЗАЦИИ

2.1

Для разработки был выбран язык программирования R [18] использующийся для статистической обработки данных и работы с графикой, а также является свободно распостроняющейся программной средой вычислений с открытым исходным кодом в рамках проекта GNU.

Для визуализации графики использовались собственные модули, стандартные встроенные функции и пакет ggplot2 [31]. Для реализации алгоритмов использовались как и стандартные функции , так и пакеты rpart (Recursive Partitioning and Regression Trees) [32], для Рекурсивное разделение который реализует деревья решений, adabag [33], который реализует алгоритм адабуст, пакет randomforest ...описание пакетов представлено в таблице 2.1

Таблица 2.1 Описание основных пакетов изпользующихся в исследовании

Пакет	Описание пакета
ggplot2	Этот пакет помогает расширить арсенал языка R, предназначенн для визуализации данных. Для создания графики ggplot2 использует систему абстрактных понятий: массив данных, визуальные средства, геометрические объекты, сопоставление переменных из массива визуальным средствам, статистическое преобразование переменных, системы координа
rpart	построения классификационных и регрессионных моделей с применением двухшаговой процедуры, а результат представляется в виде бинарных деревьев.
adabag	Реализует алгоритм adaBoost
randomForest	Реализует алгоритм random forest
doParallel	Распараллеливание

Таблица 2.2 Описание написанных модулей

Описание R функций	
ReadDataSet.R	Чтение набора данных из файла или загрузка из пакетов. В каждом наборе данных необходимо выделить числовые признаки и номинальные признаки, которые хранятся как factor в конструкции data.frame. Если на вход поступает числовая матрица, то ее нужно перевести в data.frame и номинальные признаки конвертировать с использованием as.factor. На выходе получаем список из необходимых для расчета матрицы, датафрэйма и вектора с классами
Experiment.R	Основная функция эксперимента. Вызывает функцию TestEnsemble(). Выбираются основные параметры метода: ensemble - какие ансамбли строятся , BTree - размеры ансамблей. Также в конце выполняется запись результатов эксперимента в файл write.table (имя файла эквивалентно имени набора данных для эксперимента). Далее вызов функции для графического отображения результатов GraphResult().

TestEnsemble.R	Функция в которой основная функциональность метода. Строит несколько ансамблей классификаторов, которые выбраны в Experiment().
ExperimentP.R	Основная функция эксперимента (версия распараллеливания).
TestEnsembleP.R	Функция в которой основная функциональность метода (версия распараллеливания).
RotateClassifier.R	Реализован новый метод построения ансамбля на основе Rotation Forest реализация которого описанна в пункте 1.6 глава 1.
GraphResult.R	Графический вывод по результатам эксперимента в виде ящиков с усами (boxflag=TRUE). Параметр type позволяет выбрать построение графика результатов для всех ансамблей различных размеров (type=TRUE) или только для одного размера ансамблей (type=FALSE). Эту функцию можно вызывать отдельно, если есть результаты эксперимента для некоторого набора данных. Если функция вызывается вне запуска Experiment(), то нужно указать входные параметры: nfile (имя файла результатов),classifiers (вектор с именами ансамблей, которые строились в процессе эксперимента, записанного в файл, simpleclass (NULL или имя простого классификатора), BTree (вектор размеров ансамблей)
DataResult.R	Дополнительная функция, принимает на вход файл с результатами эксперимента. Строит разные типы графиков представленные в главе 3
Какие строятся классификаторы	ensemble=c("RotateCl","AdaBoost","Bagging","RandomForest") ансамбли и simpleclass=c("SimpleTree") один единичный классификатор (одно дерево)

Были созданы модули для чтения наборов данных и манипуляции с ними. Включает преобразование данных в матрицу, датафпэйм,

После чтения и преобразования набора данных, в параллельном режиме запускается процесс обучения моделей на наборе данных и тестирование, как указано в главе, после процессов формируется матрица с 5 методами, простое дерево решений, и все 4 метода, каждый имеющий метод отсортированные по возрастанию размерности ансамбля.

Были созданы модули для чтения наборов данных и манипуляции с ними. Включает преобразование данных в матрицу, датафпэйм, После чтения и преобразования набора данных, в параллельном режиме запускается процесс обучения моделей на наборе данных и тестирование, как указано в главе, после процессов формируется матрица с 5 методами, просто дерево решений, и все 4 метода э, каждый имеющий отсортированные по возрастанию Далее мы подгружаем выходную матрицу в модуль преобразования, который проводит манипуляции и вводит графические Результаты Преобразование в факторыМетрики качества И так далееМетрики качества И так далее

ГЛАВА 3 РЕЗУЛЬТАТЫ ТЕСТИРОВАНИЯ МЕТОДОВ ПОСТРОЕНИЯ АНСАМБЛЕЙ КЛАССФИКАТОРОВ

Были проведены эксперименты по сравнению методов бэггинг, бустинг, случайные деревья и предложенного метода на наборах данных по машинному обучению [15], [29]. Целью является сравнительная оценка эффективности работы ансамблей с акцентом на предложенный вариант ансамбля вращающихся деревьев. Основными задачами исследования являлись:

Оценить тенденцию изменений эффективности классификации с использованием критерия точности классификации для ансамблей с увеличением из размерности (количества деревьев). Разработать способ обобщения оценок точности классификации ансамблями, полученных по всем наборам данных.

Выполнить сравнительный анализ эффективности классификации для ансамблей с фиксированным размером по всем наборам данных. Особый интерес представляют ансамбли малой размерности, т.к. известно, что эффективности различных методов ансамблевой классификации сравниваются с увеличением числа деревьев. Для сравнения разработать способ ранжирования ансамблей на основе попарного сравнения эффективности по совокупности всех анализируемых наборов данных.

Оценить отличие эффективности предложенного метода на основе вращающихся деревьев от других ансамблей классификаторов с использованием как табличного, так и графического представления результатов классификации.

3.1 Наборы данных

Исследования проводились на наборах данных из архивов по машинному обучению. Характеристики 14 наборов данных из архивов по машинному обучению UCI Machine Learning Repository[15] и KEEL-dataset repository[29] представлены в таблице 3.1. Как видно из таблицы 3.1 мы взяли разнообразные наборы данных для тестирования, чтобы анализировать качество классификации

методов на более непохожих наборах. Наборы данных различаются по количеству объектов, атрибутов, количеству классов.

Таблица 3.1 - Характеристики наборов данных использованных в исследовании

Имя датасета	Атрибуты (R/I/N)	Количество объектов	Количество классов
Appendicitis	7 (7/0/0)	106	2
Balance	4 (4/0/0)	625	3
BreastCancer	5 (5/0/0)	215	3
Bupa	6 (1/5/0)	345	2
Cleveland	13 (13/0/0)	297 (303)	5
Ecoli	7 (7/0/0)	336	8
Heart	13 (1/12/0)	270	2
Ionosphere	33 (32/1/0)	351	2
Iris	4 (4/0/0)	150	3
Led7digit	7 (7/0/0)	500	10
Pima	8 (8/0/0)	768	2
Sonar	60 (60/0/0)	208	2
Vehicle	18 (0/18/0)	846	4
Wine	13 (13/0/0)	178	3

Для реализации метода вращающихся деревьев и выполнения трансформации признакового пространства использовался метод PCA. В связи с тем, что метод PCA определён для числовых признаков, категориальные признаки в ряде наборов из архива по машинному обучению были преобразованы в s бинарных признаков, кодированных 0 или 1, где s - количество категорий признака.

3.2 Постановка эксперимента

В результате экспериментов были проанализирована эффективность работы следующих ансамблей классификаторов: бэггинг, бустинг (AdaBoost), случайные леса и предложенный вариант ансамбля вращающихся деревьев.

Основные этапы эксперимента следующие:

- Применение методов построения ансамбля к наборам данных по машинному обучение. Размер ансамблей варьировался от 10 до 100 с шагом 10. Эффективность классификации сравнивалась для ансамблей равных размеров. Разработка способа графического представления результатов в виде диаграммы слоев.

- Для каждого набора данных и ансамбля эффективность классификации оценивалась с использованием 100 повторных разбиений объектов на обучающее и тестовое множества в пропорции 10:1. Т.е. ансамбль строился на обучающем множестве, а эффективность классификации оценивалась на тестовом. Данный подход позволил оценить среднее значение и стандартное отклонение критерия эффективности классификации рассчитывалась точность классификации и результаты для каждого ансамбля представляются. Оценка среднего значения критерия точности рассчитывалось по 100 повторным разбиениям .

- Построение таблиц оценки результатов классификации для каждого ансамбля фиксированного размера.

- Выполнение ранжирование классификаторов по их эффективности с использованием разработанного способа. (N=100).

- Построение различных графических представлений результатов работы ансамблей на данных. Построения диаграммы зависимости критерия Каппа от точности базовых классификаторов для двух наборов данных генной экспрессии. Оценка преимуществ и недостатков предложенного метода для классификации данных

В качестве базовых классификаторов используются деревья решений.

3.3 Результаты анализа

Размер ансамбля классификаторов B является параметров всех используемых методов и является индикатором сложности ансамбля. Оптимальное значение этого параметра, дающего максимальную точность классификации, может быть

оценено с использованием метода перекрестной проверки. На рисунке. 3.1 и рисунке 3.2 представлено изменение точности классификации при варьирующемся количестве базовых классификаторов от $B=10$ до $B=100$ с применением метода повторных разбиений на обучающее и тестовое множества. На оси x отмечена размерность ансамбля, а на оси y точность классификации. Как видно на рисунках предложенный метод на основе вращающихся деревьев является наиболее точным для ансамблей различного размера.

На рисунке.3.1 представлена диаграммы вида ящик с усами для значений точности классификации и график зависимости точности классификации от размерности ансамбля по каждому методу для набора данных “Balance”.

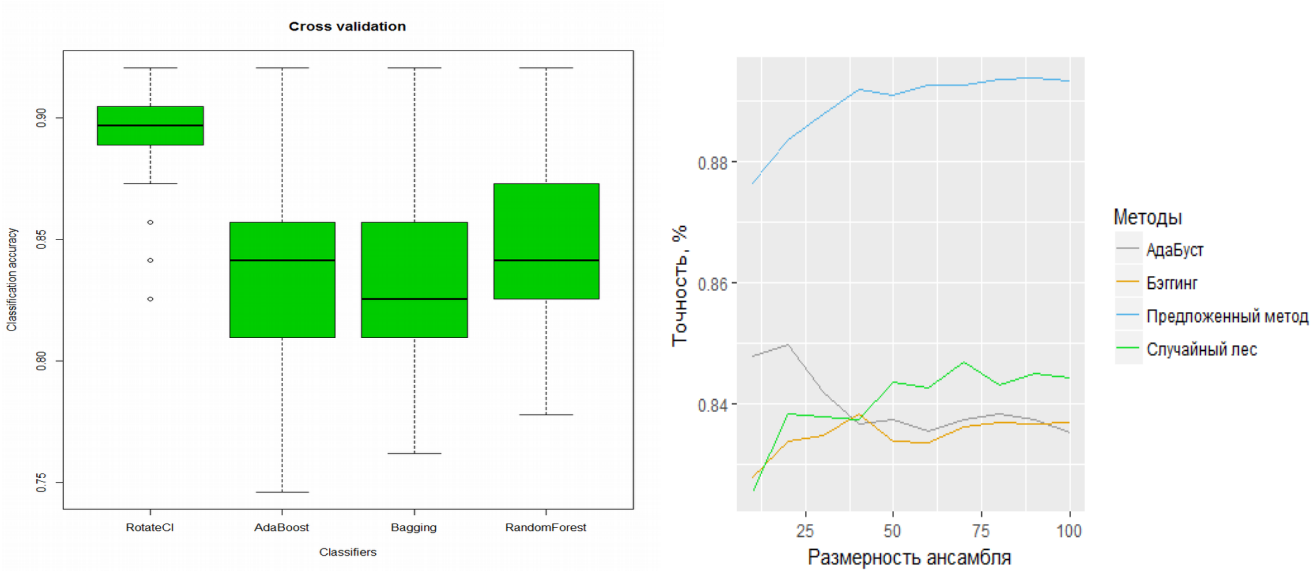

Рисунок. 3.1. Диаграммы точности для набора данных «Balance»

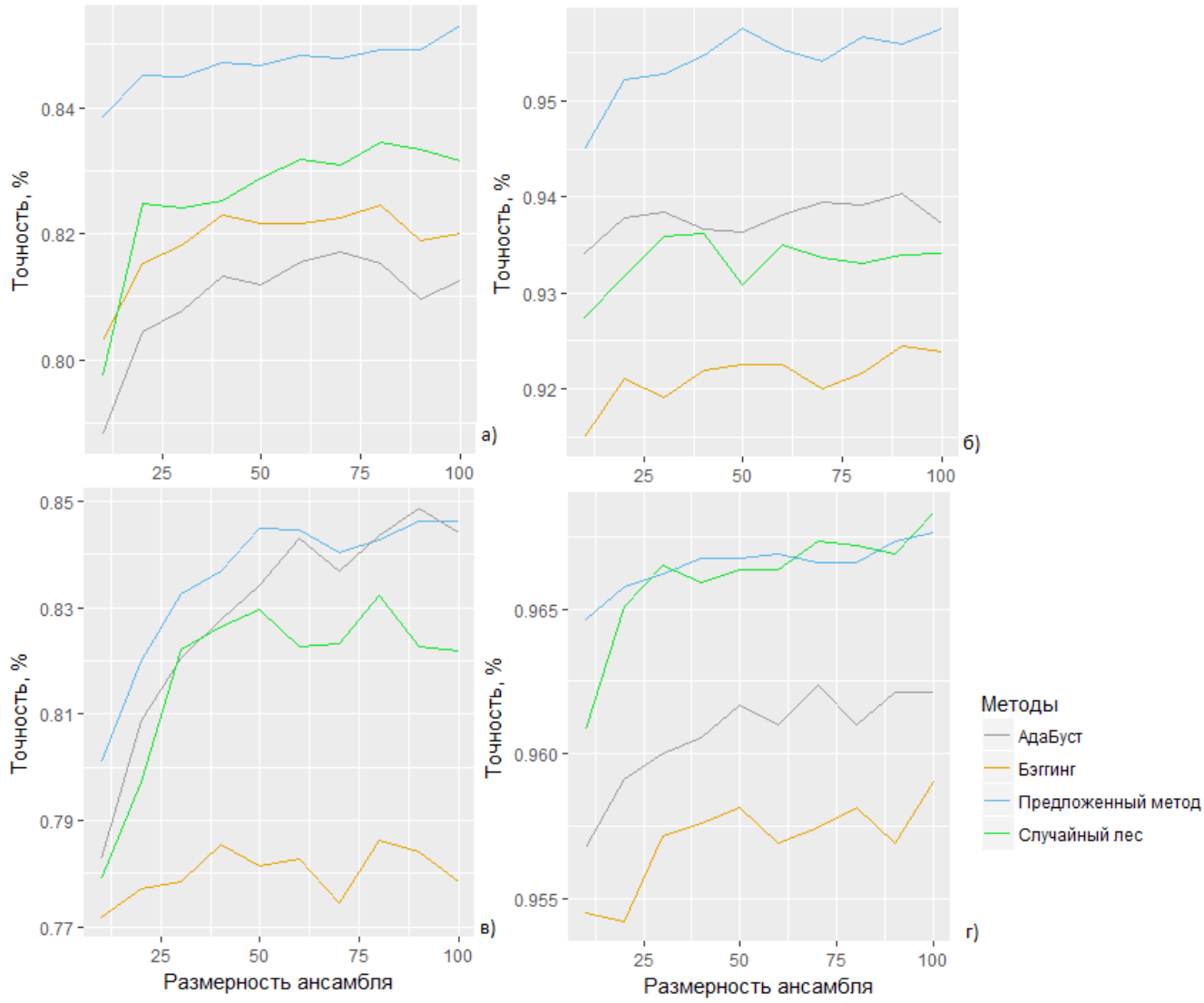

Рисунок. 3.2 График демонстрирует изменение точности классификации от размерности ансамбля для каждого метода на наборе данных:
а) Heart, б) Ionosphere, в) Sonar, г) Breast Cancer

Для каждого набора данных и метода построения ансамбля было оценена среднее значение и стандартное отклонение точности классификации (Таблица 3.2). Оценка выполнялась с использование 100 повторений случайных разбиений исходного набора данных на обучающее и тестовое подмножество.

В нашем эксперименте интерес представляют ансамбли небольшой фиксированной размерности, таким образом, все методы построения ансамблей сравниваются при равном значении параметра B. Размер всех ансамблей был выбран равным B=10.

В таблице 3.2 представлены результаты классификации, где для каждого набора данных точкой отмечен классификатор с наибольшей средней точностью.

Из таблицы 3.2 видно, что предложенный метод построения ансамбля вращающихся деревьев был наиболее эффективным для большинства наборов данных. Предложенный метод особенно хорошо классифицировал наборы данных с бинарной классификацией и данные связанные с биомедициной.

Таблица 3.2 - Точность классификации и стандантное отклонение

Datasets	Rotation Forest	AdaBoost	Bagging	Random Forest
Appendicitis	0.8486±0.0033	0.8427±0.0031	0.8472±0.0035	0.8572±0.0054•
•Balance	0.8895±0.0055•	0.8397±0.0050	0.8349±0.0029	0.8404±0.0061
BreastCancer	0.9665±0.0008•	0.9606±0.0017	0.9570±0.0015	0.9660±0.0020
Bupa	0.7061±0.0056	0.7099±0.0063	0.7202±0.0088	0.7263±0.0151•
Cleveland	0.5545±0.0035•	0.5217±0.0062	0.5351±0.0022	0.5390±0.0044
Ecoli	0.8033±0.0058	0.7849±0.0037	0.7627±0.0028	0.8056±0.0070•
•Heart	0.8469±0.0037•	0.8095±0.0084	0.8188±0.0061	0.8262±0.0107
•Ionosphere	0.954±0.0036 •	0.9377±0.0017	0.9212±0.0027	0.9331±0.0025
Iris	0.9477±0.0029	0.946±0.0025	0.9535±0.0027 •	0.9504±0.0016
Led7digit	0.7261±0.0035	0.7309±0.0019•	0.7195±0.0025	0.7216±0.006
Pima	0.7700±0.0027•	0.743±0.0035	0.7697±0.0026	0.7630±0.0061
Sonar	0.8355±0.0146•	0.829±0.0204	0.7800±0.0047	0.8177±0.0165
Vehicle	0.7319±0.0031	0.7699±0.0108•	0.7090±0.0032	0.7492±0.0017
Wine	0.9722±0.0066	0.9682±0.0007	0.9585±0.0035	0.9762±0.0043•

На рисунке 3.3 представлен другой вариант графического результата. На оси Y представлена точность предложенного метода ансамбля на основе вращающихся дереьвев, а на оси X наилучшая точность классификации среди сравниваемых методов. Диагональной линии, соответствует одинаковым значениям точности. Большинство точек лежит выше диагональной линии, что означает преимущество предложенного метода.

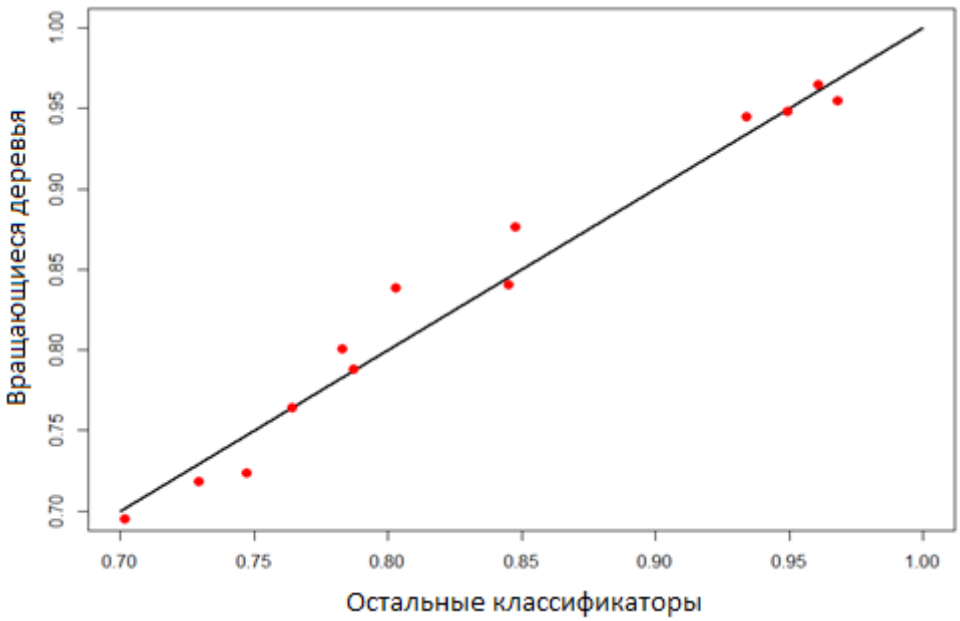

Рисунок 3.3 Диаграмма демонстрирует точность предложенного метода к остальным методам по всем наборам данных

В таблице 3.3 представлен ранжированный список сравниваемых методов, согласно разнице между числом раз, когда метод был лучше или хуже другого метода при проведении попарных сравнений.

Таблица 3.3 Ранжированный список сравниваемых методов

	Вращающиеся деревья	АдаБуст	Бэггинг	Случайный лес
Ранг	20	-12	-22	14
Количество побед	31	15	10	28
Количество поражений	11	27	32	14

Для проведения ранжирования был разработан способ для оценки рангов сравниваемых методов. Для этого для анализируемых ансамблей классификаторов и для каждого набора данных были выполнены следующие шаги:

1. Были определены все возможные пары ансамблей классификаторов.

2. Для каждого ансамбля и для каждого конкретного набора данных подсчитывались следующие показатели: 1) количество раз, когда данный ансамбль был лучше остальных (по средней точности классификации) 2) количество раз когда каждый ансамбль был хуже остальных. Данный подсчет производился для всех наборов данных.

3. Рассчитывался ранг классификатора как разница между количеством выигрышей и проигрышей при классификации для всех наборов данных.

Для обобщения оценок точности классификации ансамблями, полученных по всем наборам данных для изменяющейся размерности ансамблей был разработан способ расчета и построения диаграммы для визуализации полученных результатов оценки. Данная диаграмма позволяет визуально представить зависимость различия базовых классификаторов и их средней точности для каждого метода построения ансамбля. Как и при ранжировании различия пары ансамблей построенных различными методами оценивались с использованием сравнения выходов классификаторов. На рисунке 3.4 представлены процентная диаграмма для 4 ансамблевых методов и соотношение для тех же методов в зависимости от количества ансамблей (деревьев решений) на наборе данных Balance.

Для визуализации полученных данных было реализовано несколько версий графика со слоями. Для всех анализируемых ансамблей классификаторов и для каждого набора данных были выполнены следующие шаги:

1. Были определены все возможные пары ансамблей классификаторов.

2. Для каждого ансамбля и для каждого конкретного набора данных подсчитывались следующие показатели: 1) количество раз, когда данный ансамбль был лучше остальных (по средней точности классификации, по максимальной точности классификции, на каждой выборке из 100 разбиений) и сравнение проводилось для ансамблей с одинаковой размерностью B. 2) количество раз когда каждый ансамбль был хуже остальных. Данный подсчет производился для всех наборов данных.

3. Формировалась матрица с результатами по каждому ансамблю и расчитывалась доля побед каждого ансамбля на конкретной размерности.

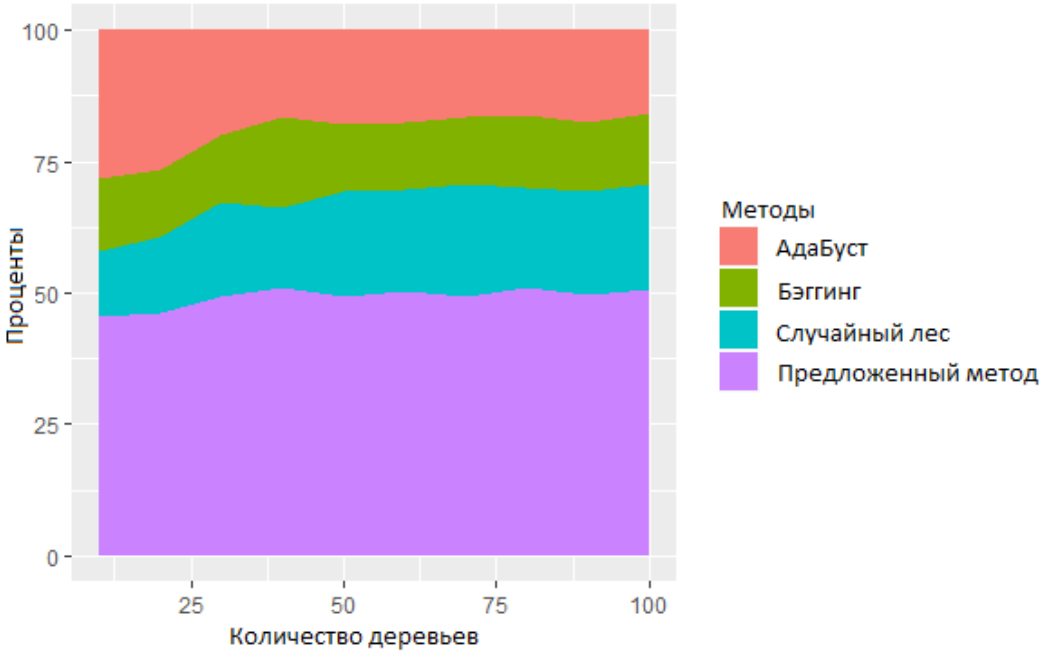

Рисунок 3.4.График со слоями демонстрирует долю выигрышей для каждого метода на наборе данных «Balance»

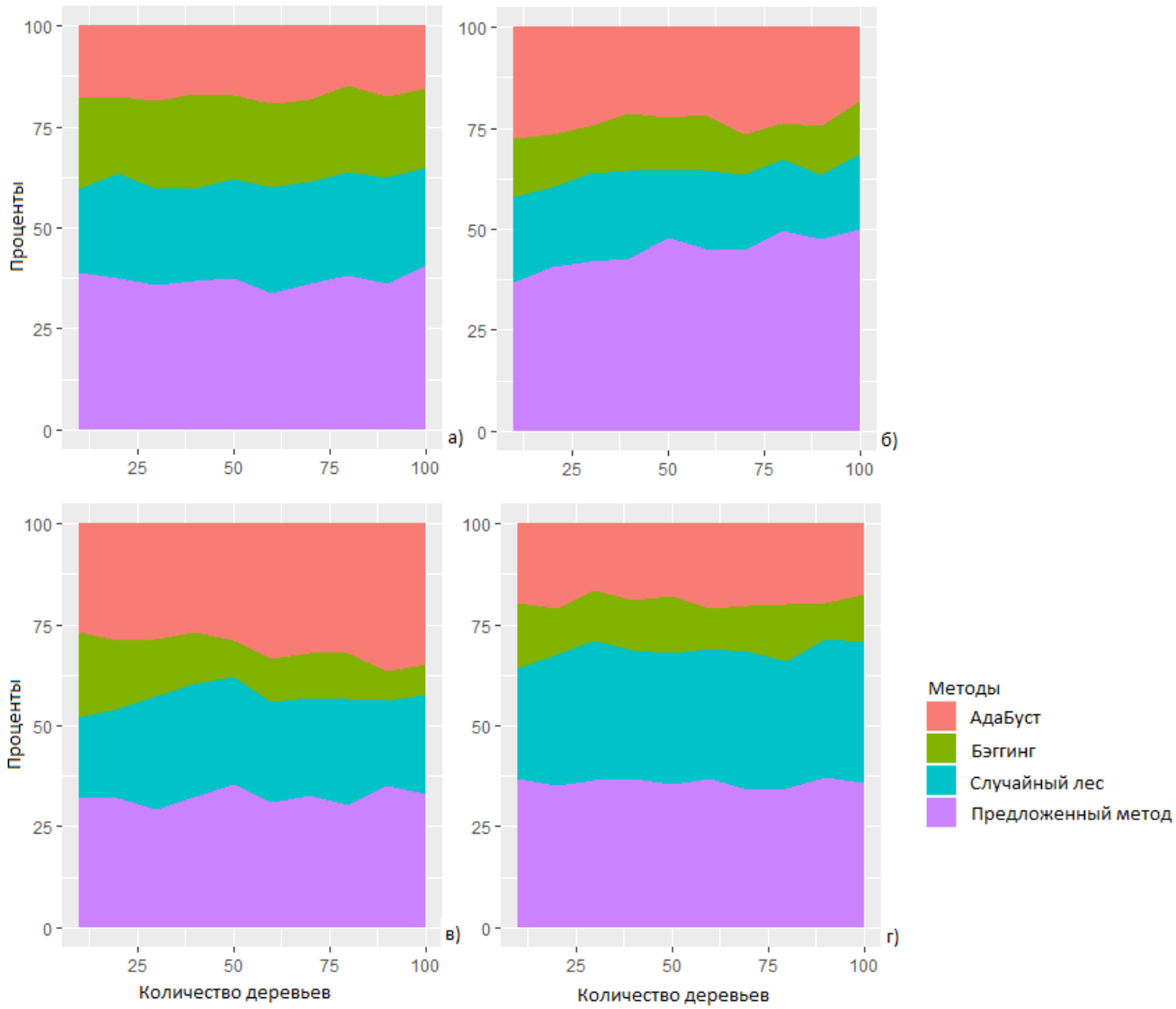

Рисунок. 3.5 График со слоями демонстрирует долю выигрышей от количества деревьев для каждого метода на наборе данных :
а) Heart, б) Ionosphere, в) Sonar, г) Breast Cancer

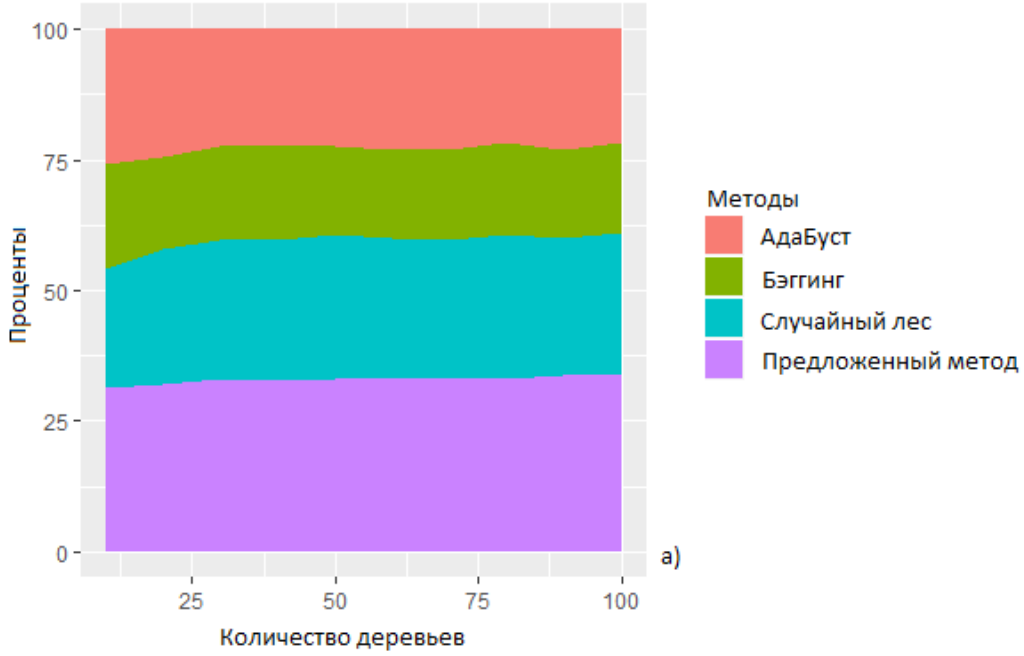

Рисунок 3.6 График со слоями демонстрирует долю выигрышей для каждого метода на всех наборах данных в зависимости от количества деревьев

Согласно анализу рисунков 3.2-3.6 можно сделать следующие выводы:

1) для большинства наборов данных точность метода на основе вращающихся деревьев незначительно изменялась при увеличении размера ансамбля, т.е. данный метод давал достаточно высокую точность классификации в процентном соотношении с другими методами при малой размерности сравниваемых ансамблей.

2) метод на основе вращающихся деревьев показал наибольший процент выигрышей на всех наборах данных, причем на ряде наборов, таких как Balance, Heart, Breast Cancer он имел значительное преимущество перед остальными методами.

3) при сравнении методов на всех наборах данных предложенный метод в большинстве случаев был лучше методов случайные деревья и AdaBoost, причем метод бэггинг показал в среднем наихудшие результаты.

4) Ансамбль случайный лес показал достаточно хорошую точность классификации наборов данных, которая как правило возростала с увеличением размерности ансамбля.

В ходе проведения эксперимента была исследована зависимость различия базовых классификаторов и их средней точности для каждого метода построения ансамбля. Различия пары ансамблей построенных различными методами оценивались с использованием сравнения выходов классификаторов. Для этого использовалась Каппа критерий κ, являющийся мерой согласованности между двумя категориальными переменными. Для K классов, критерий определяется на основе матрицы сопряженности выходов двух классификаторов M размерности $K \times K$.

Низкие значения критерия Каппа означают несогласованность результатов классификации тестового множества и, следовательно, высокий уровень различия пары классификаторов. Если классификаторы независимы, то их согласованность

будет эквивалентна согласованности, полученной случайным образом, и значение Каппа будет равно нулю.

Таким образом, ансамбль из B классификаторов образует $B(B-1)/2$ пары классификаторов D_i, D_j. На оси x диаграммы (рис. 8) представлено значение Каппа для пары классификаторов и на оси y среднее значение ошибки базовых классификаторов D_i, D_j. Малое значение Каппа означает более высокий уровень различия базовых классификаторов и малое значение ошибки означает лучшую точность классификации.

Рис. 3.7 представляет диаграмму зависимости критерия Каппа и точности классификации для 5 наборов данных.

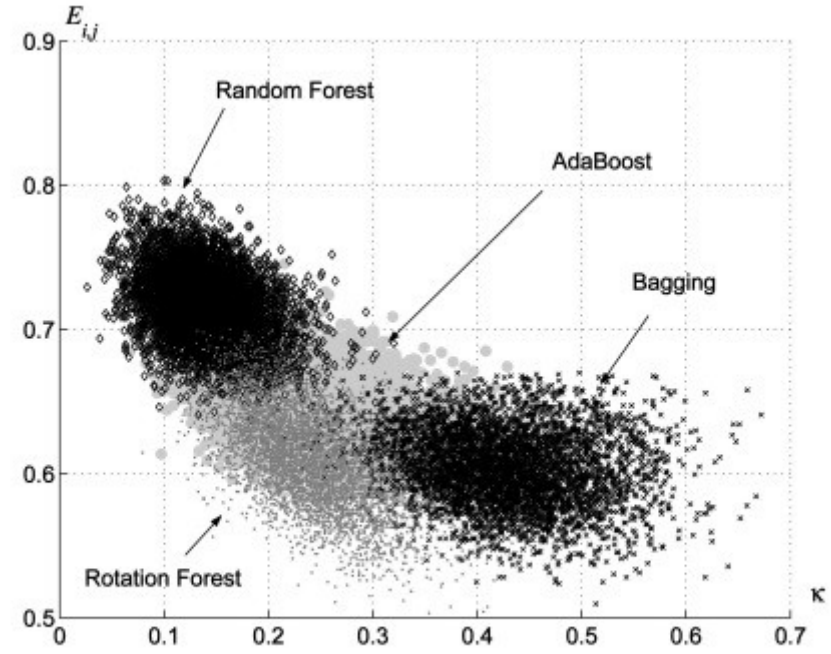

Fig. 7. Kappa-error diagrams for the vowel-n data set.

Рисунок 3.7

Cohen Kappa and Weighted Kappa correlation coefficients and confidence boundaries

	lower	estimate	upper
unweighted kappa	0.91	0.94	0.98
weighted kappa	0.91	0.95	0.98

Из рисунка 3.7 видно, что базовые классификаторы ансамбля AdaBoost являются наиболее независимыми (малое значение Каппа по оси x), однако много пар базовых классификаторов имеют большую ошибку (большое значение Каппа

по оси Y). Остальные три метода имеют близкую точность, причем ансамбль случайных лесов менее точны, чем бэггинг и предложенный метод вращающихся деревьев. Базовые классификаторы предложенного метода более независимы, чем их аналоги в методе бэггинг, однако превосходят данный метод по точности, что возможно и объясняет наилучшую эффективность всего ансамбля в среднем на всех наборах данных.

ВЫВОДЫ

В ходе исследований предложена версия метода ансамблей классификаторов на основе вращающихся деревьев. Проведен сравнительный анализ нескольких ансамблей классификаторов, включая предложенный ансамбль. Гипотеза о преимуществах предложенного ансамбля подтвердилась на большинстве проанализированных наборах данных. Кроме того получены следующие результаты:

- Проанализированы изменения эффективности классификации с использованием критерия точности классификации для ансамблей с увеличением их размерности. Разработан способ обобщения оценок точности классификации ансамблями, полученных по всем наборам данных. Для визуализации результатов классификации разработан вариант графического представления эффективности ансамблей на наборах данных в виде процентной диаграммы.
- Выполнен сравнительный анализ эффективности классификации для ансамблей с фиксированным размером по всем наборам данных. Разработан способ ранжирования ансамблей на основе попарного сравнения эффективности по совокупности всех анализируемых наборов данных.
- С помощью разработанного способа ранжирования ансамблей на основе попарного сравнения эффективности по совокупности всех анализируемых наборов данных получили, что метод вращающихся деревьев стоит на первом месте в списке ансамблей классификаторов, ранжированном по

эффективности. На втором месте стоит случайный лес. При анализе эффективности ансамблей по совокупности наборов данных метод вращающихся деревьев выигрывает для большинства размеров деревьев.

ЗАКЛЮЧЕНИЕ

В ходе выполнения работы предложен метод ансамбля классификаторов на основе вращающихся деревьев. Метод заключается в осуществлении повторных случайных подвыборок из обучающего множества с последующим применением метода главных компонент (МГК) и построения каждого базового классификатора на трансформированном признаковом пространстве, что позволяет снизить степень корреляции между ошибками отдельных классификаторов, т.е. мы были нацелены на разнообразие классификаторов и на высокую точность одновременно. В качестве базового алгоритма использовался вариант деревьев решений CART, потому что деревья решений чувствительны к вращению осей и достаточно точные.

Был проведен сравнительный анализ ансамбля на основе вращающихся деревьев с тремя другими ансамблями классификаторов, а именно Бэггинг, АдаБуст и Случайный лес, доступные в R.

Для проведения анализа было выбрано 14 наборов данных из архивов по машинному обучению UCI Machine Learning Repository [15] и KEEL-dataset repository [29], которые отличались по размерности признакового пространства и количеству классов. Количество классов варьировалось от 2 до 8, а количество признаков было не более 50. Результаты эксперимента по наборам данных показали, что метод на основе вращающихся деревьев превзошел все три метода по точности классификации.

В ходе экспериментов был разработан способ обобщения оценок точности классификации ансамблями, полученных по всем наборам данных, который позволил графически оценить результаты классификации ансамблями наборов данных при изменении размеров ансамбля. В качестве вывода было установлено, что эффективность ансамбля случайного леса как правило повышается с увеличением его размерности, что нельзя сказать о предложенном методе, для которого точность классификации имеет достаточно высокий уровень уже на малом количестве деревьев (до 10).

В ходе сравнительного анализа ансамблей был разработан способ их ранжирования на основе попарного сравнения эффективности по совокупности всех анализируемых наборов данных. На основе проведенного анализа метод вращающихся деревьев показал преимущества на большинстве наборов данных, он стоит на первом месте в списке ансамблей классификаторов, ранжированном по эффективности.

С использованием криетрия Каппа было исследовано соотношения точности и независимости базовых классификаторов, составляющих каждый из ансамблей. Согласно результатам базовые классификаторы предложенного ансамбля оказализь немного более точными и более разнообразными, чем бэггинг, что возможно я является основной причиной их высокой эффективности классификации соответствующего ансамбля.

В качестве направлений дальнейших исследований можно выделить необходимость проведения экспериментов на наборах данных с большой размерностью, а также использование других вариантов базовых классификаторов для построения ансамблей.

Литература

1. R. O. Duda, P. E. Hart, and D. G. Stork. Pattern Classi_cation. John Wiley & Sons, NY, second edition, 2001.
2. L. I. Kuncheva. Combining Pattern Classi_ers. Methods and Algorithms. Wiley, 2nd edition, 2014.
3. J. J. Rodriguez, L. I. Kuncheva, and C. J. Alonso. Rotation forest: A new classi_er ensemble method. IEEE Transactions on Pattern Analysis and Machine Intelligence, 28(10):1619{1630, Oct 2006.
4. L. Rokach. Pattern Classi_cation Using Ensemble Methods. World Sceinti_c, 2010.
5. Robert E. Schapire and Yoav Freund. Boosting: Foundations and Algorithms. MIT Press, 2012.
6. Zhi-Hua Zhou. Ensemble Methods: Foundations and Algorithm. CRC Press { Business & Economics, 2012.
7. Н.А., Новоселова Подход к построению ансамбля классификаторов с использованием генетического алгоритма / Новоселова Н.А., Том И.Э. // Журнал «Искусственный интеллект», Украина / Институт проблем искусственного интеллекта - Донецк : "Наука і освіта", 2009. - № 3. - С. 81-88.
8. Новоселова, Н.А. Evolutionary Design of the Classifier Ensemble / Н.А. Новоселова, И.Э. Том, С.В.Абламейко// Искусственный интеллект. – 2011. – №3. – С.429-438.
9. Multiple Classifier Systems / J. Kittler & F. Roli (editors) // Proc. of 2nd International Workshop, MCS2001, (Cambridge, UK, 2-4 July 2001) / Lecture Notes in Computer Science. – Vol. 2096. – Springer-Verlag, Berlin.
10. Vishwath P. Fusion of multiple approximate nearest neighbor classifier for fast and efficient classification / P. Vishwath, M.N. Murty, C. Bhatnagar, // Information fusion. – 2004. – Vol. 5. – P. 239-250.
11. Quinlan J.R. Bagging, boosting and C4.5 / J.R. Quinlan // Proceedings of AAA/IAAI. – 1996. – Vol. 1. –P. 725-730.

12. Skurichina M., Duin R. P. W. Limited bagging, boosting and the random subspace method for linear classifiers // Pattern Analysis & Applications. 2002. Pp. 121–135.

13. Breiman, Leo. Random Forests // Machine Learning, 45(1), 5-32, 2001.

14. Kuncheva L.I. and C.J. Whitaker, Pattern recognition and classification, Wiley StatsRef-Statistics Reference Online, 2014.

15. C.L. Blake and C.J. Merz, “UCI Repository of Machine Learning Databases,” 1998, http://www.ics.uci.edu/mlearn/MLRepository.html.

16. Breiman, L., Friedman, J.H., Olshen, R.A., and Stone, C.J. Classification and Regression Trees. 1984.

17. T.G. Dietterich, “Ensemble Methods in Machine Learning,” Proc. Conf. Multiple Classifier Systems, pp. 1-15, 2000.

18. R Core Team (2013). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL http://www.R-project.org/.

19. Dudoit S., Fridlyand J., Speed T. Comparison of discrimination methods for the classification of tumors using gene expression data. J. Am. Stat. Assoc. 2002, 97:77–87.

20. Kohavi R, John G: Wrapper for feature subset selection. Artificial Intelligence 1997, 97(1–2):273-324.

21. Thomas, J.G., Olson, J.M., Tapscott, S.J., Zhao, L.P., 2001. An efficient and robust statistical modeling approach to discover differentially expressed genes using genomic expression profiles. Genome Res. 11, 1227–1236.

22. Antoniadis, A., Lambert-Lacroix, S., Leblanc, F., 2003. Effective dimension reduction methods for tumor classification using gene expression data. Bioinformatics 19, 563–570.

23. Liu, X., Krishnan, A. & Mondry, A. An entropy-based gene selection method for cancer classification using microarray data. BMC Bioinformatics 6, 76 (2005).

24. Liu, H. and Setiono, R., Chi2: Feature selection and discretization of numeric attributes, Proc. IEEE 7th International Conference on Tools with Arti_cial Intelligence, 338-391, 1995.

25. Wang Y, Tetko IV, Hall MA, Frank E, Facius A, Mayer KF, Mewes HW: Gene selection from microarray data for cancer classification – a machine learning approach. Comput Biol Chem 2005, 29(1):37-46.

26. Liu H, Li J, Wong L. A comparative study on feature selection and classification methods using gene expression profiles and proteomic patterns. Genome Inform. 2002;13:51-60.

27. Golub,T.R. et al. (1999) Molecular classification of cancer: class discovery and class prediction by gene expression. Science, 286, 531–537.

28. Novoselova N.A . An algorithm to estimate the stability of the individual clusters in the hierarchical context // Intelligent Data Analysis . - 2014. - Vol.18 - №4, p. 531-546.

29. KEEL-dataset citation paper: J. Alcalá-Fdez, A. Fernandez, J. Luengo, J. Derrac, S. García, L. Sánchez, F. Herrera. KEEL Data-Mining Software Tool: Data Set Repository, Integration of Algorithms and Experimental Analysis Framework. Journal of Multiple-Valued Logic and Soft Computing 17:2-3 (2011) 255-287.

30. Quinlan, J. R. C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers, 1993

31. H. Wickham. ggplot2: Elegant Graphics for Data Analysis. Springer-Verlag New York, 2009.

32. Breiman L., Friedman J. H., Olshen R. A., and Stone, C. J. (1984) Classification and Regression Trees. Wadsworth.

ПРИЛОЖЕНИЕ А