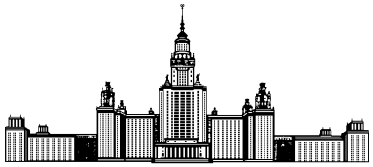
Московский государственный университет имени М. В. Ломоносова



Факультет Вычислительной Математики и Кибернетики

Кафедра Математических Методов Прогнозирования

КУРСОВАЯ РАБОТА СТУДЕНТА 517 ГРУППЫ

«Автоматическое выделение признаков в задаче классификации сигналов»

Выполнил:

студент 5 курса 517 группы

Викулин Всеволод Александрович

Научный руководитель:

д.ф-м.н., профессор

Дьяконов Александр Геннадьевич

Москва, 2017

Содержание

1 Введение **2**

1.1 Постановка задачи классификации сигналов 2

1.2 Обзор литературы 3

2 Стохастический алгоритм синтеза признакового пространства **5**

2.1 Представление признака через базисные функции 5

2.2 Оценка качества признака 9

2.3 Схема метода 11

3 Вычислительные эксперименты **15**

4 Выводы **21**

Список литературы **22**

1 Введение

Сигнал – последовательность измерений некоторой величины. Задача классификации сигналов часто встречается во множестве различных прикладных задач – от медицины до приборостроения [1, 6]. По этой причине разработка алгоритмов классификации сигналов становится важной и актуальной задачей.

Одним из популярных подходов к решению задачи классификации сигналов является нахождение оптимального признакового пространства, в котором объекты (сигналы) могут наиболее просто быть разделены с помощью классических алгоритмов классификации.

Данная работа посвящена методу автоматического построения признакового пространства с помощью максимизации критерия качества признака (здесь и далее признаком сигнала будем называть любую вещественную функцию от сигнала). В данной работе использовался метод оптимизации, который является обобщением жадного поиска, но использование конкретно этого метода оптимизации совершенно не обязательно. Поиск оптимального признакового пространства при этом являлся стохастическим, то есть в самом алгоритме заложена рандомизация, что позволяет генерировать постоянно новые признаки, отличающиеся от предыдущих. Критериев качества для оценки признака было несколько, и они тоже выбирались для каждого признака случайно. Это так же помогало генерировать непохожие друг на друга признаки, так как не существует универсального метода оценки качества, при этом нельзя максимизировать сразу их все. Благодаря стохастике данный алгоритм позволял за N итераций почти всегда найти N не похожих друг на друга признака. Далее синтезированное множество признаков может использоваться любым классическим классификатором.

1.1 Постановка задачи классификации сигналов

В данном разделе рассмотрим математическую постановку задачи. Пусть задано множество объектов X, множество допустимых ответов Y, и существует функция $y : X \to Y$, значения которой $y_i = y$ известны только на конечном подмножестве объектов $\{x_1, \ldots, x_l\} \subset X$. Задача заключается в том, чтобы по имеющимся

парам объект-ответ восстановить исходную зависимость, то есть построить решающую функцию $a : X \to Y$, которая приближала бы целевую функцию y, причём не только на известных объектах, но и на всём множестве X.

Признаком объекта x назовем результат измерения характеристики объекта. Другими словами, признаком называется отображение $f : X \to D_f$, где D_f – множество допустимых значений признака. Нахождению оптимального множества таких отображений $\{f\}$ посвящена данная работа.

В данном случае под множеством объектом X будем всегда иметь в виду множество сигналов, то есть конечную последовательность вещественных чисел, а под множеством допустимых ответов Y двухэлементное множество {-1, 1}.

1.2 Обзор литературы

Нахождение оптимального признакового пространства является крайне сложной задачей. В большинстве случаев в задачах анализа сигналов для конструирования этого пространства применяются классические приемы, основанные в своем большинстве на преобразовании Фурье и вейвлет-преобразовании [8,11,13]. Такой подход имеет ряд существенных недостатков. Он требует глубокого понимания от исследователя природы сигнала, и исследователь должен сам подбирать необходимое спектральное разложение, не существует некоторого универсального преобразования, которое бы позволяло всегда выделять оптимальное признаковое пространство из сигнала. Из-за этого недостатка те качественные признаки, которые были найдены в прошлой задаче анализа сигнала, в новой задаче могут быть абсолютно неприменимы. В новой задаче анализа сигналов исследователю необходимо с нуля конструировать новое признаковое пространство, опираясь исключительно на свою интуицию и опыт, полученный при решении прошлых задач.

Большая популярность нейронных сетей способствовала продвижению нового подхода к анализу сигналов, лишенного вышеперечисленных недостатков – нейросетевого подхода [3, 4]. Нейронные сети являются отличным инструментом, который позволяет работать с неструктурированной информацией, они сами способны извлекать необходимые для классификации признаки из неструктурированных данных, избавляя исследователя от необходимости ручного построения признакового про-

странства. Успешность применения нейронных сетей в задачах анализа изображений и текста подтолкнуло исследователей на мысль о возможном применении их еще и в задаче анализа сигналов.

Одним из популярных подходов в анализе кардиосигналов является подход, который был предложен В.М.Успенским при разработке метода информационного анализа электрокардиосигналов [14, 17]. Этот подход основан на предположении о том, что амплитуды и интервалы кардиоциклов несут в себе информацию о состоянии организма человека. С помощью специальной кодировки амплитуд и интервалов кардиоциклов, кардиосигнал может быть представлен с помощью набора триграмм, а затем этот набор триграмм можно анализировать классическими методами анализа текстов, например с помощью методов тематического моделирования. Несмотря на то, что данный подход изначально предполагался для анализа электрокардиограмм человека, он может быть успешно применен и в анализе сигналов другого рода. Этот метод значительно упрощает хранение информации о сигнале, ее обработку и дает возможность работать с сигналами методами, которые изначально сами по себе для сигналов не предназначены.

Подход, который используется в данный работе заключается в автоматическом построении признакового пространства путем максимизации качества признака. Данный подход уже применялся несколько раз в анализе сигналов [2, 5, 10, 12]. В этих работах используется генетический алгоритм для нахождения оптимального признакового пространства. Генетический алгоритм является алгоритмом оптимизации, который базируется на механизмах, в какой-то степени аналогичных механизмам эволюции в живой природе. В качестве функции, которая оптимизируется генетическим алгоритмом, выступает какая-либо мера качества признака. Например, качество предсказания алгоритма, построенного на синтезированном признаке на кросс-валидации. Основным недостатком данных работ является четкая привязка как к методу оптимизации, так и к выбору оценки качества признака. Из-за жесткой привязки к оценке качества генетический алгоритм является хорошим выбором, потому что он не старается наивным образом подобрать себе лучшее решение, как это делает жадный алгоритм, например. Если бы в этих работах использовался жадный алгоритм, то признаковое пространство было бы бедным и все время одинаковым,

так как он не обладает нужной вариативностью. Одной из важнейших задач данной работы является построение метода, в который легко бы встраивался абсолютно любой метод оптимизации, то есть предлагается обеспечивать нужную вариативность не с помощью метода оптимизации, а с помощью стохастической природы поиска оптимального признакового пространства. В этом случае метод оптимизации может быть любым, он не будет определяющим в конструкции.

2 Стохастический алгоритм синтеза признакового пространства

В данном разделе рассмотрим предлагаемый в текущей работе алгоритм синтеза признакового пространства.

2.1 Представление признака через базисные функции

Напомним, что признаком сигнала называется функция от сигнала, которая ставит в соответствие сигналу какое-то число. Будем раскладывать каждую такую функцию через набор заранее определенных базисных функций, в рамках которых мы и будем проводить оптимизацию. Таким образом выбор базисных функций однозначно определит пространство, в котором будет происходить оптимизация. Каждый признак при этом будет представлять собой суперпозицию базисных функций, которые будут применяться поочередно, формируя в итоге значение признака. Пусть мы выбрали множество $\{b\}$ мощности N базисных функций, тогда любой признак сигнала может быть представлен в виде:

$$f(x) = [b_1][b_2][b_3]\dots[b_{last}](x), \tag{1}$$

где b_i - очередная базисная функция из множества $\{b\}$, прямоугольные скобки используются для разделения базисных функций и отдельного смысла не несут. Далее будем подразумевать, что функции в выражении (1) применяются слева направо. Эта форма записи не согласуется с привычными правилами записи подобных выражений в математике, но была выбрана из соображений наглядности.

Заметим, что в формуле (1) каждый признак может быть представлен через любое число базисных функций. Конкретно взятая базисная функция может быть использована неограниченное, но обязательно конечное число раз в представлении признака.

Отметим еще, что если мы определили, что признаком сигнала является число, то последняя из базисных функций в формуле (1) обязана быть $b_{last} : R^m \to R$. Остальные функции должны быть согласованы по областям задания и областям значений: $b_i : R^{m_i} \to R^{m_{i+1}}$, $b_{i+1} : R^{m_{i+1}} \to R^{m_{i+2}}$.

Для того, чтобы лучше понять смысл выражения (1), рассмотрим простой пример. Пусть множество $\{b\}$ стоит из трех элементов: $\{\sin(x), x^2, x+1\}$. Тогда признак $f(x) = \sin((x+1)^2)$ может быть записан через базисные функции как:

$$f(x) = [x+1][x^2][\sin(x)](x). \tag{2}$$

В данной работе из-за специфичности задачи анализа сигналов любой признак описывается не формулой (1), а ее несколько усложненным вариантом, что позволяет как сузить оптимизируемое пространство, так и использовать априорные знания о том, какие базисные функции вообще должны применяться в задаче обработке сигналов, в каком порядке они должны применяться.

Все базисные функции $\{b\}$ разделены на 3 непересекающихся класса.

- функции инициализации – множество $\{i\}$. Это функции, с которых каждый признак в представлении (1) должен начинаться. Должна быть ровна одна функция инициализации (она может быть тождественной, то есть не меняющей сигнал). В множество функций инициализаций стоит включить те преобразования, которые в предметной области чаще всего используются для предобработки данных, это позволит напрямую использовать знания о предметной области при поиске признакового пространства. Например, если Вы анализируете сигналы, то стандартным подходом является сглаживание сигнала скользящим окном или использование фильтра высоких частот. В данном случае сглаживание сигнала и фильтр высоких частот – функции инициализации. Это функции $i : R^m \to R^n$.

- функции трансформации – множество $\{t\}$. Это функции, которые отвечают за преобразования сигнала, который прошел через инициализацию. В каждом сигнале их может быть любое количество, число функций трансформаций может быть ограничено только из соображений вычислительной сложности получаемых признаков. Эти функции в отличие от функций инициализации слабее зависят от предметной области и представляют собой по большей части нелинейные преобразования. Это функции $t : R^m \to R^n$

- функции агрегации – множество $\{a\}$. Для того, чтобы получить из сигнала число, необходимо в конце цепочки базисных функций поставить функцию, которая бы агрегировала бы всю полученную информацию в одно число, поэтому необходимо ввести функции агрегации. Они практически совсем не зависят от предметной области. Функциями агрегации могут быть, например, среднее значение последовательности, максимальное значение последовательности и так далее. Это функции $a : R^m \to R$.

Таким образом формула (1) может быть переписана в виде:

$$f(x) = [i][t_1][t_2] \dots [t_{last-2}][a](x), \tag{3}$$

где i – какая-то из функций инициализации, $t_1, \dots t_{last}$ – какие-то функции трансформации, a – какая-то функция из функций агрегации.

Задача генерации признакового пространства заключается в том, чтобы найти N признаков, представимых в форме (3), которые бы были бы оптимальны с точки зрения оценки качества алгоритма, обученного на этом пространстве признаков. В свою очередь это означает, что для каждого из N признаков необходимо найти функцию инициализации, последовательность функций трансформации и функцию агрегации для данного признака.

Таблица 1: Используемые функции инициализации

Функция инициализации	Описание
Тождественная	Не изменяет исходный сигнал
Медианное сглаживание	Значение сигнала заменяется медианой по некоторой окрестности
Фильтр верхних частот	Пропускает высокие частоты сигнала, подавляет частоты ниже определенной частоты
Фильтр нижних частот	Пропускает низкие частоты сигнала, подавляет частоты выше определенной частоты
Фильтр верхних и нижних	Последовательное применение двух фильтров
Прямое преобразование Фурье	$\hat{f}(\omega) = \frac{1}{\sqrt{2\pi}} \int\limits_{-\infty}^{\infty} f(x) e^{-ix\omega}\, dx.$
Обратное преобразование Фурье	$f(x) = \frac{1}{\sqrt{2\pi}} \int\limits_{-\infty}^{\infty} \hat{f}(w) e^{ix\omega}\, dw$
Время максимумов	Функция возвращает значения времени при которых сигнал имеет локальный максимум
Значение максимумов	Функция возвращает значения сигнала в локальных максимумах

В таблицах 1, 2, 3 приведены функции инициализации, трансформации и агрегации, которые использовались для экспериментов. Такой выбор был обусловлен интуицией и опытом, вполне допускает выбор других функций, удовлетворяющих правилам представления (3).

Включение в функции инициализации функции поиска локальных максимумов сигналов обусловлен успехами анализа кардиосигналов в работах [14, 17].

Важно отметить, что многие из функций трансформаций заключают в себе еще и параметры. Например, можно применять периодическую функцию не $\sin(x)$, а $\sin(2x)$. Выбор с каким параметром будем использовать функция трансформации должен решаться только непосредственно перед ее применением к сигналу. В экспериментах параметры выбирались случайно из заранее выбранного множества.

Таблица 2: Используемые функции трансформации

Функция трансформации	Описание
Логарифмирование	Каждая точка заменяется ее логарифмом
Возведение в степень	Каждая точка возводится в степень параметра
Показазательная	Параметр возводится в степень каждой точки
Периодическая	Над сигналом применяется какая-либо периодическая функция
Конечная разность порядка n	Конечная разность порядка n от сигнала
Модуль	Точка заменяется своим абсолютным значением
Шкалирование	Значения сигнала переводятся на отрезок [0,1]
Стандартизация	Новый сигнал будет с нулевым средним и единичной дисперсией

Таблица 3: Используемые функции агрегации

Функция агрегации	Описание
Среднее значение	Среднее значение последовательности
Медиана	Медиана последовательности
Дисперсия	Дисперсия последовательности
Максимум и минимум	Максимальное/минимальное значение последовательности
Отношение максимума к минимуму	Отношение среднего к медиане
Отношение среднего к медиане	Отношение максимума к минимуму
Центр масс сигнала	Скалярное произведение сигнала на вектор $(0, 1, \dots len(signal))$

2.2 Оценка качества признака

Для того, чтобы построить хорошее признаковое пространство, удовлетворяющее условию (3), необходимо четко определить критерий, по которому будет проходить

поиск нового признака. Необходимо решить, что мы будем подразумевать под хорошим признаком, а затем максимизировать данный критерий каким-либо методом оптимизации. Таким образом нам необходимо ввести **критерий качества признака**.

Критерии качества признака сильно связаны с методами фильтрации признаков, которых на данных момент известно уже немало. Обзор методов фильтрации признаков можно найти в [7]. Похожие подходы можно использовать и в оценке качества признака.

Самый простой способ оценить качество признака – проверить, насколько статистически признак связан с целевой переменной. В этой области существует невероятно большое число исследований. Перечислим лишь несколько способов, которые в дальнейшем будем использовать для экспериментов:

- Количество неправильно ранжированных пар целевой переменной при сортировке ее по значениям данного признака. Самая простая оценка качества. Полагаем, что значения признака есть выход некоторого классификатора. Отсортируем целевую переменную по данному признаку и проверить качество этой сортировки. Заметим, что без разницы, по возрастанию или по убыванию проводить сортировку.

- Корреляция Пирсона между целевой переменной и признаком, то есть мера линейной зависимости признака от целевой переменной. При этом разумно брать модуль, так как нам неважен знак этой линейной зависимости.

- Взаимная информация между целевой переменной и признаком, то есть величина, описывающая количество информации, содержащегося в целевой переменной относительно признака. В качестве оценки качества разумно брать нормированное на отрезок [0,1] значение.

Статистические методы обладают важным достоинством – они очень быстро считаются. Из-за этого получили широкое распространение в задачах, где признаковое пространство состоит из огромного числа признаков, но при этом не очень важно выделить оптимальное подмножество признаков из пространства, а гораздо важнее

убрать совершенно бесполезные или даже вредные признаки. Основным недостатком этих методов является недостаточная описательная способность, любой статистический критерий не способен исчерпывающе описать степень зависимости одной величины от другой, очень высокий риск ошибки в оценке качества.

Существует другой обширный класс методов оценки качества признаков, который проверяет качество алгоритма, обученного на одном этом признаке. В экспериментах использовался один из самых простых алгоритмов – алгоритм k ближайших соседей (k nearest neighbors, KNN [15]). Описать этот алгоритм довольно просто – объект относится к тому классу, к которому относится большинство из его k соседей, то есть k ближайших к нему объектов обучающей выборки. Степень близости можно определять как угодно, в данной работе использовалась стандартное евклидово расстояние. Оценка качества проводилась методом скользящего контроля с исключением объектов по одному (leave-one-out, LOO). Это очень популярный метод оценки качества алгоритма k ближайших соседей. В этом методе каждый объект по очереди исключается из обучающей выборки, для него происходит предсказание, вычисляется оценка качества, а затем это качество усредняется по всем объектам.

Последний метод заключается в проверке того, что признак хорошо используется алгоритмом. В экспериментах использовался алгоритм дерево решений (decision tree [16]). К тестируемому признаку прибавлялся случайный признак, затем на этих двух признаках строилось дерево решений фиксированной глубины, оценивалось во сколько раз тестируемый признак лучше, чем случайный признак с помощью оценки уменьшение impurity по разбиениям дерева решений [16].

2.3 Схема метода

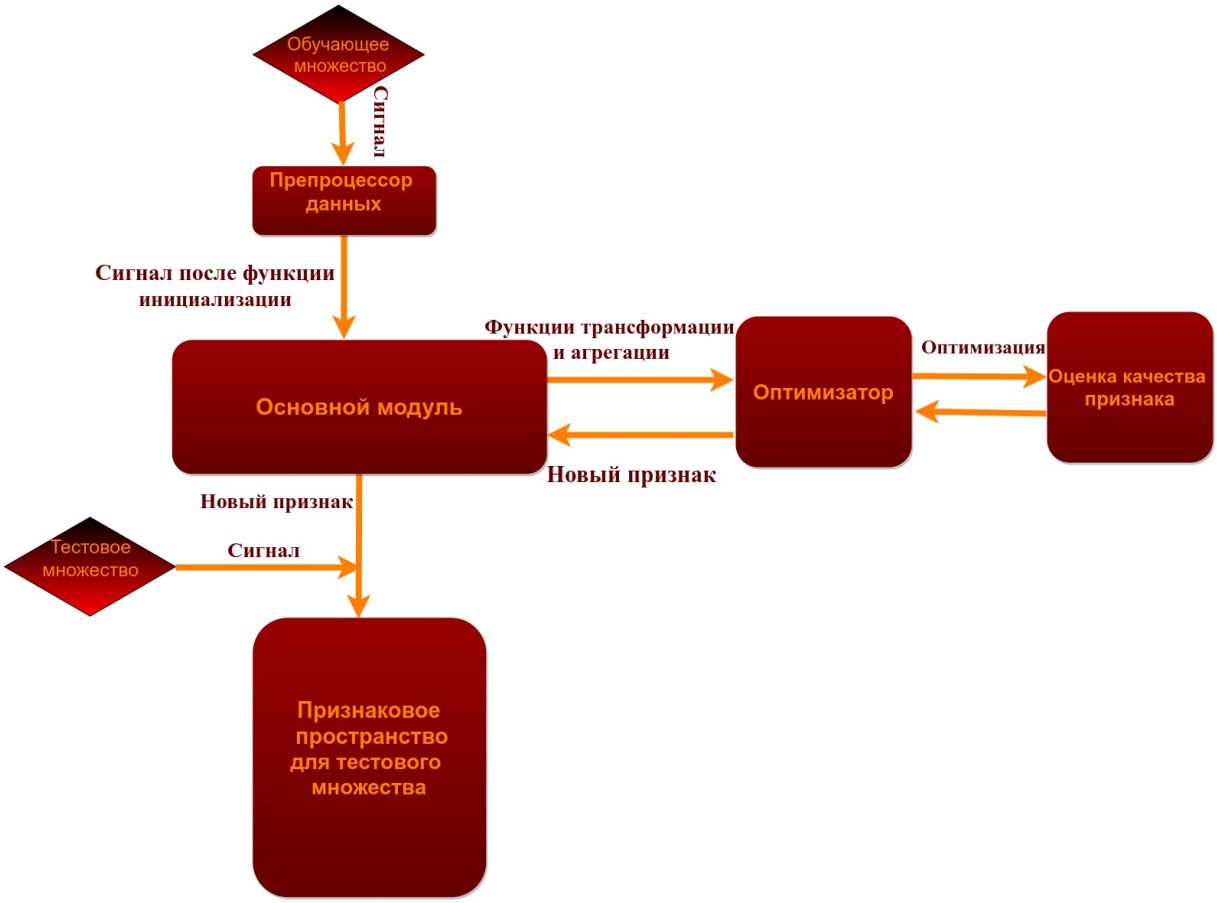


Рис. 1: Составляющие элементы метода

На рисунке 1 показаны составляющие, из которых конструируется алгоритм. Основными функциональными элементами являются:

- Основной модуль, отвечающий за определение множества функций инициализации, трансформации и агрегации и за связь остальных элементов.
- Препроцессор, который преобразует признак, используя функцию инициализации.
- Оптимизатор, который подбирает функции трансформации и агрегации, используя оценку качества.
- Метод оценки качества признака.

Основным достоинством данного подхода является то, что в зависимости от задачи можно изменять каждый модуль в отдельности, оставляя весь подход целостным и

Algorithm 1 Стохастический алгоритм синтеза признакового пространства

procedure FIND_FEATURES($sigs, N, k, init\_funcs, trans\_funcs, agg\_funcs, criteria$)
    $i = 0$
    $features = \{\}$
    while $i \neq N$ **do** ▷ Ищем N признаков
        $subs = random\_subsample(sigs, k)$ ▷ Взяли подмножество сигналов
        $new\_init = get\_random(init\_funcs)$ ▷ Случайная инициализация
        $init\_subs = new\_init(subs)$ ▷ Применили инициализацию
        $new\_crit = get\_random(criteria)$ ▷ Случайный критерий
        $set\_parameters(trans\_funcs)$ ▷ Проставили параметры, если нужны
        $new\_feat = optimize(init\_subs, new\_crit, trans\_funcs, agg\_funcs)$
        if $new\_feat \notin features$ **then** ▷ Проверили, что новый признак
            $i = i + 1$
            features.insert(new_feat)
    return $features$ ▷ Вернули найденные признаки

неизменным. Этот факт сильно упрощает работу с применением метода для новой задачи.

Алгоритм (1) описывает работу метода с помощью псевдокода. Метод работает примерно так:

- Случайно взяли подвыборку из k элементов из множества сигналов.
- Применили к этой подвыборке случайную функцию инициализации.
- Выбрали случайный критерий качества признака.
- Установили параметры в функциях трансформации, если не устраивают стандартные. Например, можно возводить значение сигнала не в квадрат, а в куб. В экспериментах параметры трансформации брались случайно из заранее выбранного множества, но можно использовать любой другой подход, например, можно решать еще внутреннюю задачу оптимизации по параметру.

- Нашли новый признак методом оптимизации. Если его еще не видели, то запомнили его. Если нужно больше признаков, начали процесс сначала (с новой случайной подвыборки).

Построение признака по случайной подвыборке решает одновременно несколько задач. Признаки будут не очень похожи на друг друга, так как они подстраивались под разные множества. Уменьшается риск переобучиться под обучающее множество, то есть риск построить признаковое пространство, которое работает только на определенном наборе объектов. Позволяет избавиться от проблемы несбалансированных классов, можно брать подвыборку с равным количеством объектов каждого класса. Уменьшает вычислительную сложность оценки качества признака, которая во многих методах очень большая.

Алгоритм (2) иллюстрирует работу жадного оптимизатора. Он наращивает функции трансформации жадным образом. Наращивает до тех пор, пока не превысит заранее установленный лимит, или пока качество не перестанет расти. При добавлении новой трансформации просматриваются все возможные функции агрегации.

Algorithm 2 Жадный оптимизатор

```
1:  procedure OPTIMIZE(init_subs, new_crit, trans_funcs, agg_funcs)
2:      best_qual = −inf
3:      found_trans = {}                          ▷ Храним трансформации
4:      while len(found_trans) ≠ MAX_SIZE do      ▷ Макс. длина признака
5:          for new_trans in trans_funcs do       ▷ Перебираем все трансформации
6:              found_better = False              ▷ Индикатор успеха
7:              for new_agg in agg_funcs do       ▷ Перебираем все агрегации
8:                  feature = create(found_trans + {new_trans}, new_agg)
9:                  if qual(feature) > best_qual then   ▷ Нашли лучший вариант
10:                     found_better = True
11:                     new_best_trans, best_agg = new_trans, new_agg
12:         if notfound_better then               ▷ Остановились, если не нашли
13:             break
14:         found_trans.insert(new_best_trans)
15:     return found_trans, best_agg              ▷ Вернули лучший признак
```

3 Вычислительные эксперименты

Эксперименты проводились на сигналах, которые представляют собой электрокардиограммы пациентов. Для каждой кардиограммы известно, болен ли ее владелец ишемической болезнью сердца. Это классическая задача бинарной классификации, где класс 1 означает, что пациент с данной кардиограммой болен, класс 0 – здоров. Выборка состояла из 1798 сигналов, из которых 743 сигнала принадлежало больным, а 1055 сигналов принадлежало здоровым пациентам.

Оценка качества синтезированного множества признаков будет происходить с помощью измерения качества алгоритма, обученного на этих признаках, Для этого может использоваться любой классический классификатор. В качестве базового классификатора был выбран *случайный лес* [9]. Это ансамбль решающих деревьев. Каждое решающее дерево строится по случайным подвыборкам, полученным в резулатате сэмплирований с возвращениями объектов обучающей выборки.

Для оценки качества будем использовать 20-кратную кросс-валидацию. Важно отметить, что в обучающей выборке многим пациентам принадлежит сразу несколько кардиограмм, поэтому валидация проводилась таким образом, чтобы кардиограммы любого пациента не могли попасть и в обучение, и в контроль одновременно. Это более честная оценка, так как кардиограммы одного и того же пациента очень похожи, и алгоритму проще выдать правильный ответ, так как он уже ранее видел похожую кардиограмму.

Размер случайной подвыборки составлял 100 объектов (50 объектов каждого класса). В экспериментах использовалось 2 функционала качества. Первый функционал – точность предсказания по пациентам. Вычисляется он так: для каждой кардиограммы каждого пациента делается предсказания о наличие у пациента болезни, затем для каждого пациента считается процент правильно классифицированных его кардиограмм, затем все эти значения усредняются по пациентам. Вторым функционалом является площадь под ROC-кривой (сокращенно AUC – Area Under Curve). Она считается уже не по пациентам, а по сигналам. Не вдаваясь подробно в ROC-анализ упомянем лишь, что ROC-кривая показывает зависимость между долей истинно положительных (в нашем случае – доля правильно классифицированных кардиограмм больных пациентов) примеров и долей ложно положительных (в на-

шем случае – доля неверно классифицированных кардиограмм здоровых пациентов) примеров. С помощью анализа этой кривой можно явным образом судить о качестве классификации, но визуальное сравнение может не позволить сразу оценить эффективность модели, необходимо сопоставить кривой какое-то число. Площадь под этой кривой является классическим методом оценки качества классификации, этот показатель меняется от 0 до 1.0. Значение 0.5 соответствует случайному классификатору, а значение 1.0 соответствует идеальной с точки зрения AUC модели. Более подробно про эту и другие метрики качества классификации можно найти в [15].

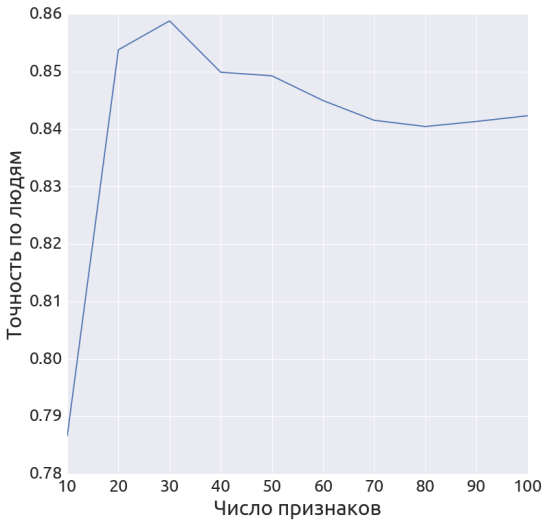


Рис. 2: Зависимость точности по пациентам от количества признаков.

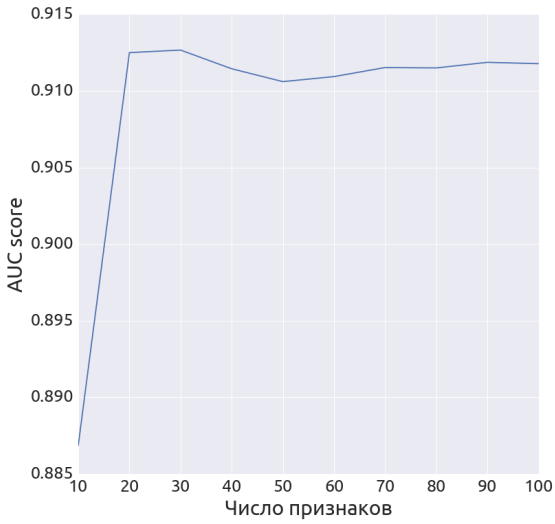


Рис. 3: Зависимость AUC от количества признаков.

Все эксперименты проводились с алгоритмом случайный лес, который состоял из 100 деревьев. На рисунке 2 и рисунке 3 показана зависимость метрик качества от количества признаков с шагом в 10 признаков. На них видно, что оптимальное количество признаков находится около 30. Дальнейшее увеличение признакового пространства не приводит к росту качества. Это свидетельствует о том, что многие из сгенерированных признаков являются шумовыми, поэтому предсказательные возможности алгоритма не растут. Важно отметить, что алгоритм постоянно создает новые признаки, они не повторяются уже со старыми. Это наводит на мысль о необходимости создания качественной методики отбора признаков, которая позволила бы

из этого большого множества признаков выделить оптимальное подмножество. Максимальное значение точности по пациентам – 0.859, максимальное значение AUC – 0.913.

Все дальнейшие эксперименты проводились для множества, состоящее из 300 синтезированных признаков. На рисунках 4 и 5 показаны самые часто встречаемые функции трансформации и агрегации. Описание самих функций можно найти в таблицах 2 и 3. Как видно из этих рисунков, среди функций трансформаций нет определенной доминирующей функций, среди функций агрегаций с большим отрывом выигрывает медиана.

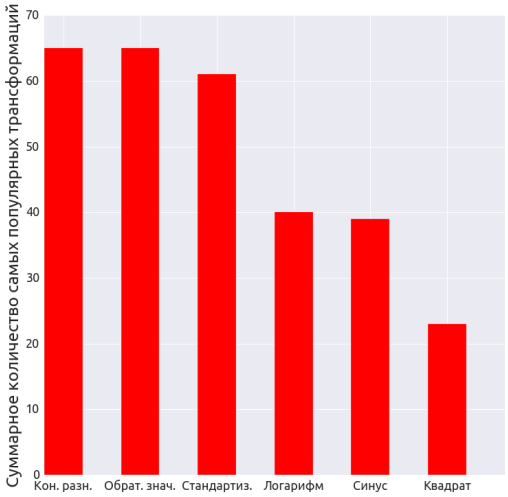


Рис. 4: Самые популярные функции трансформации.

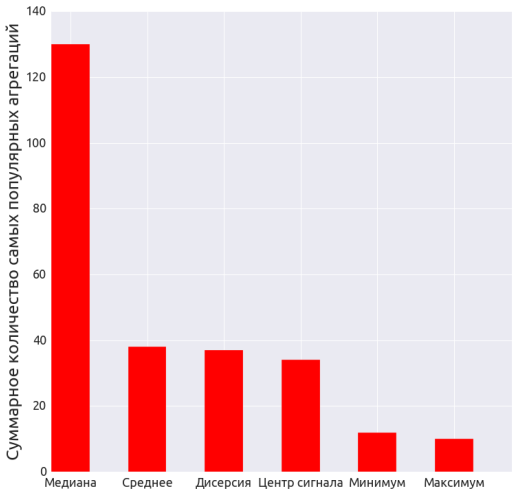


Рис. 5: Самые популярные функции агрегации.

Рисунки 6 и 7 показывают различие в поведении жадного алгоритма при различных методов оценки качества признака. Обозначения: ДНРП – доля неверно ранжированных пар (целевая переменная сортируется по значению признака), качество по ДР – качество признака по оценке дерева решений (см. раздел 2.2 про эти и другие оценки качества). Процент увеличения качества считается по формуле $\frac{FinalQual - InitQual}{InitQual}$, где $FinalQual$ – финальное качество признака, $InitQual$ – начальное качество. Начальное качество определяется качеством лучшей функцией агрегации при отсутствии функций трансформации. Из графиков видно, что зависимость меж-

ду длиной последовательности трансформации и изменении качества практически обратная – чем длиннее получается последовательность трансформаций, тем слабее изменяется качество. Средняя длина последовательности трансформаций равна 2.4 трансформации на один признак.

.

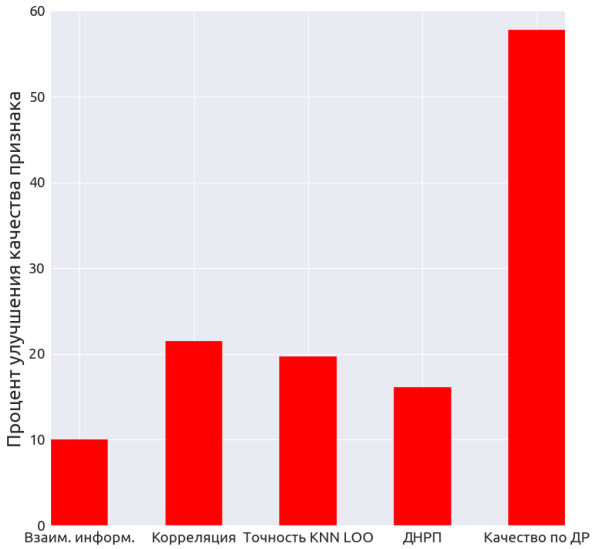


Рис. 6: Средний процент увеличения качества признака.

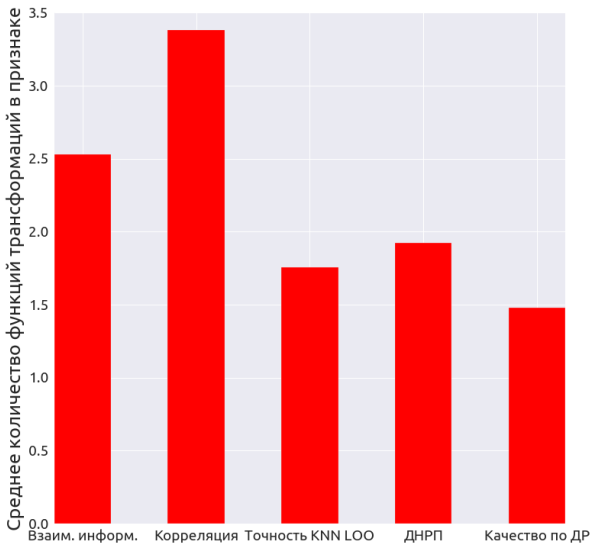


Рис. 7: Средняя длина трансформаций в признаке.

Важным критерием выбора качества признака является скорость подсчета этого качества. На рисунке 8 показана доля каждого метода оценки качества от общего времени, потраченного на оценку качества. Как и ожидалось, тяжелее всего считать LOO валидацию, на втором месте доля неверно ранжированных пар, остальные три метода считаются намного быстрее. Среднее время построения одного признака около 3 минут, то есть суммарно 300 признаков строились около 15 часов.

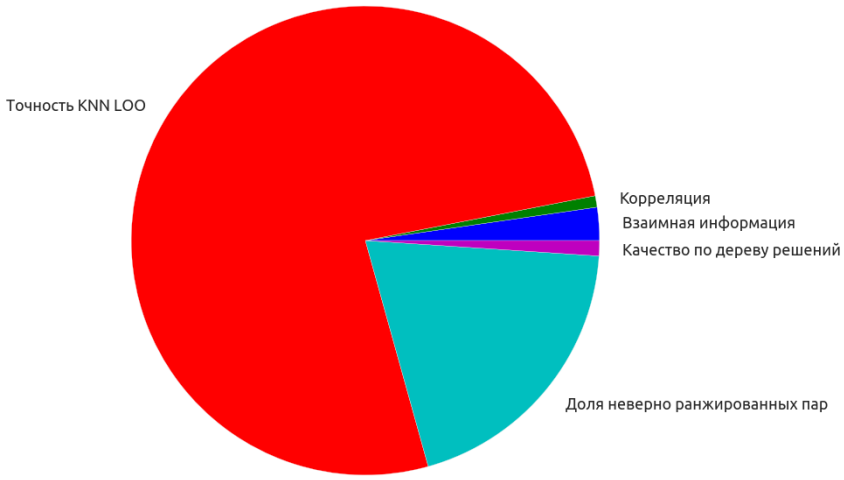


Рис. 8: Доля каждого метода оценки качества от общего времени работы.

Важной особенностью данного алгоритма является его стохастическая составляющая, поэтому среди данных 300 признаков наряду с хорошими признаками появляются бесполезные и даже вредные.

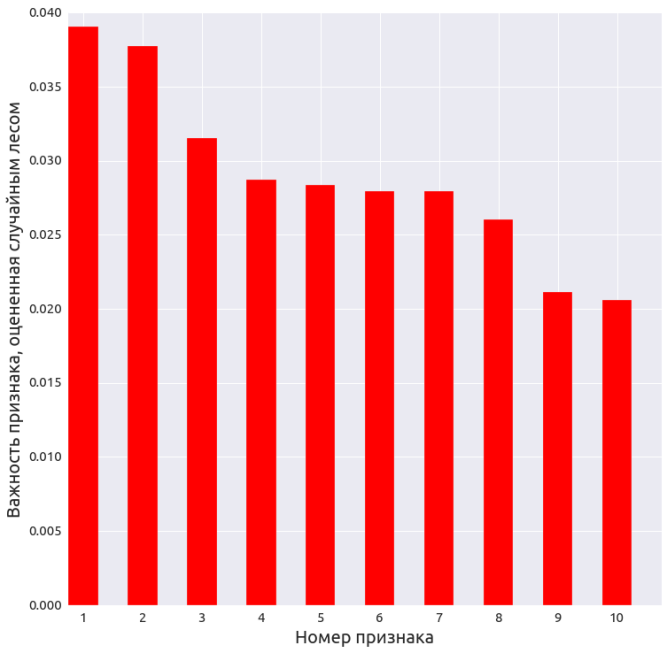


Рис. 9: Самые важные признаки по оценке случайного леса.

На рисунке 9 представлено качество 10 самых лучших признаков по оценке случайного леса. Качество признака определяется долей уменьшения impurity при выборе данного признака в качестве того, по которому будет разбиение в узле дерева. Далее это значение усредняется по всем деревьям, подробнее об impurity можно найти в [15]. Из этого рисунка видно, что среди 300 признаков нет какого-то определяющего, который бы и обеспечивал все качество. Важность признаков уменьшается плавно, без резких скачков. Среди этих 300 признаков было около 20 признаков, которые алгоритм случайный лес вообще не использовал.

10 самых важных признаков по оценке случайного леса:

1. фильтр низких частот[первая конечная разность][обратное значение][медиана], критерий – число неверно ранжированных пар.
2. фильтр низких частот[первая конечная разность][синус][шкалирование][медиана], критерий – число неверно ранжированных пар.
3. фильтр низких частот[первая конечная разность][обратное значение][синус] [среднее], критерий – KNN LOO.
4. фильтр высоких и низких частот[первая конечная разность][стандартизация] [обратное значение][медиана], критерий – взаимная информация.
5. фильтр высоких и низких частот[первая конечная разность][логарифм][обратное значение][квадрат][шкалирование][квадратный корень][обратное значение][медиана], критерий – корреляция.
6. фильтр высоких и низких частот[первая конечная разность][стандартизация] [обратное значение][медиана], критерий – взаимная информация.
7. фильтр высоких и низких частот[модуль][среднее], критерий – дерево решений.
8. фильтр высоких и низких частот[первая конечная разность][обратное значение][медиана], критерий - число неверно ранжированных пар.
9. медианный фильтр[2 в степени значение сигнала][стандартизация][синус] [стандартизация][медиана], критерий – взаимная информация

10. фильтр высоких и низких частот[первая конечная разность][обратное значение][синус][стандартизация][минимум], критерий – корреляция.

4 Выводы

В настоящей работе предложен алгоритм автоматического построения признакового пространства, были проведены вычислительные эксперименты для задачи бинарной классификации кардиограмм. Данный алгоритм в вычислительных экспериментах показал свою способность конструировать признаковое пространство, которое позволило бы решать задачу классификации сигналов с высокой точностью. Основными достоинства данного алгоритма являются:

- Автономность. Алгоритм сам создает нужные признаки, используя только оценки качества этих признаков. Работа исследователя заключается только в выборе базисных функций, которые специфичны в его задаче, если задача сильно отличается от задачи классификации сигналов.

- Модульность. Алгоритм состоит из нескольких отдельных частей: начальный набор базисных функций, метод оценки качества признака, оптимизатор. Каждая из этих частей может независимо от других настраиваться на конкретную задачу. Те варианты модулей, которые были приведены в данной работе, являются не более чем тестовыми вариантами, для каждой задачи они могут подбираться индивидуально.

- Универсальность. Возможности алгоритма не ограничиваются его применением исключительно в задаче классификации сигналов. При изменении функций инициализации, трансформации и агрегации он может быть применен в любой дугой задаче распознавания неструктурированных данных, например, в задаче классификации текстов или изображений.

- Вариативность. Благодаря своей стохастической природе алгоритм с каждой новой итерацией создает признак, который сильно отличается по своему методу построения от предыдущих. В процессе экспериментов алгоритм за 300 итераций всего несколько раз выдал признак, который был уже получен на

прошлых итерациях. Таким образом, чем больше итераций проведет алгоритм, тем больше вероятность, что среди полученных признаков будет подмножество действительно качественных.

- Расширяемость. Данный алгоритм просто улучшить. Алгоритм способен генерировать огромное число непохожих друг на другу признаков, но некоторые из них бесполезны или даже вредны для решения данной задачи. Самым очевидным расширением данного алгоритма является добавление еще одного модуля, который перед тем, как подавать на вход алгоритму классификации сгенерированные признаки, провел бы их селекцию и убрал бы ненужные признаки.

Реализация данного алгоритма доступна по адресу
https://github.com/VVVikulin/Automatic-feature-extraction-from-signal.

Список литературы

[1] C. Abadie, T. Billard, and T. Lebey. Numerical signal processing methods for partial discharge detection in more electrical aircraft. In *2016 IEEE International Conference on Dielectrics (ICD)*, volume 1, pages 540–543, July 2016.

[2] H. Al-Sahaf, K. Neshatian, and M. Zhang. Automatic feature extraction and image classification using genetic programming. In *The 5th International Conference on Automation, Robotics and Applications*, pages 157–162, 2011.

[3] N. T. H. Anh, T. H. Hoang, D. T. Dung, V. T. Thang, and T. T. Q. Bui. An artificial neural network approach for electroencephalographic signal classification towards brain-computer interface implementation. In *2016 IEEE RIVF International Conference on Computing Communication Technologies, Research, Innovation, and Vision for the Future (RIVF)*, pages 205–210, 2016.

[4] T. Debnath, M. M. Hasan, and T. Biswas. Analysis of ecg signal and classification of heart abnormalities using artificial neural network. In *2016 9th International Conference on Electrical and Computer Engineering (ICECE)*, pages 353–356, 2016.

[5] Katharina Morik Ingo Mierswa. Automatic feature extraction for classifying audio data. *Machine Learning*, 58(2):127, 2005.

[6] B. U. Kohler, C. Hennig, and R. Orglmeister. The principles of software qrs detection. *IEEE Engineering in Medicine and Biology Magazine*, 21(1):42–57, Jan 2002.

[7] Dr.J. Jebamalar Tamilselvi K.Sutha. A review of feature selection algorithms for data mining techniques. *International Journal on Computer Science and Engineering*, 7(6), 2015.

[8] P. R. U. Lallo. Signal classification by discrete fourier transform. In *MILCOM 1999. IEEE Military Communications. Conference Proceedings (Cat. No.99CH36341)*, volume 1, pages 197–201 vol.1, 1999.

[9] Breiman Leo. Random forests. *Machine Learning*, 45:5–32, 2001.

[10] Julian Dorado Cristian R. Munteanu Alejandro Pazos Ling Guo, Daniel Rivero. Automatic feature extraction using genetic programming: an application to epileptic eeg classification.

[11] A. Prochazka, J. Kukal, and O. Vysata. Wavelet transform use for feature extraction and eeg signal segments classification. In *2008 3rd International Symposium on Communications, Control and Signal Processing*, pages 719–722, 2008.

[12] B. Dal Seno, M. Matteucci, and L. Mainardi. A genetic algorithm for automatic feature extraction in p300 detection. In *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*, pages 3145–3152, 2008.

[13] A. T. Tzallas, M. G. Tsipouras, and D. I. Fotiadis. The use of time-frequency distributions for epileptic seizure detection in eeg recordings. In *2007 29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 3–6, 2007.

[14] Tselykh V. Bunakov V Uspenskiy V., Vorontsov K. Information function of the heart: Discrete and fuzzy encoding of the ecg-signal for multidisease diagnostic system.

Advanced Mathematical and Computational Tools in Metrology andTesting X, Series on Advances in Mathematics for Applied Sciences., 2015.

[15] Воронцов К.В. Курс лекций Математические методы обучения по прецедентам, МФТИ. 2004-2008.

[16] Воронцов К.В. Лекции по логическим алгоритмам классификации. 2007.

[17] Успенский В. М. Информационная функция сердца. Теория и практика диагностики заболеваний внутренних органов методом информационного анализа электрокардиосигналов. *Экономика и информатика*, 2008.