# Отчёт о решении задачи конкурса CardioQVARK

# «Разработка алгоритма определения курящего человека по его кардиограмме»

Александр Дьяконов

ф-т ВМК, МГУ имени М.В. Ломоносова

**Москва, 14.03.2016**

**Оглавление**

## 0. Задача конкурса

Первый этап конкурса CardioQVARK[1] проходил с 25 декабря 2015 по 01 марта 2016. Участникам было необходимо разработать алгоритм определения курящего человека по его кардиограмме. Организаторами были предоставлены:

- Обучающая выборка – 100 кардиограмм, с указанием курит ли человек, у которого сняли кардиограмму (в выборке было 50 курящих и, соответственно, 50 некурящих),
- Контрольная выборка – 250 кардиограмм без указания значения целевого признака (курит ли человек).

Необходимо было для каждой из 250 неразмеченных кардиограмм определить целевое значение: 1 – курит, 0 – нет, т.е. фактически сформировать 250-мерный бинарный вектор.

Кроме самих кардиограмм, организаторы предоставили некоторые вычисленные по ним признаки, а также фильтрованные кардиограммы (которые не использовались автором отчёта).

Полученный ответ можно было отослать организаторам, которые вычисляли чувствительность (Se) и специфичность (Sp) предоставленного решения, по каждому из этих критериев строилась турнирная таблица участников, и определялось место в таблице. Сумма мест по чувствительности и специфичности – итоговая оценка решения (чем меньше – тем лучше). Первые 10 участников по этой сумме мест допускались во второй этап.

Автор данного отчёта занял второе место по описанной системе определения рейтинга. Отчёт является конкурсной работой для второго этапа. Как будет отмечено в отчёте, построенный алгоритм имеет наилучшее качество среди всех алгоритмов других участников по многим стандартным критериям качества (например, по F1-мере).

[1] http://www.cardioqvark.ru/challenge/

## 1. Краткое описание содержания отчёта

Вместе с отчётом предоставлен код для генерации финального решения. Ниже полностью описаны все этапы создания финального решения, и отчёт является своеобразным путеводителем по коду. Таким образом, организаторы могут использовать все идеи и алгоритмы автора в дальнейшем.

Кроме того, в отчёте описываются некоторые идеи, которые проверял автор, но они не вошли в реализацию финального решения.

В процессе работы над задачей автор ставил своей целью не просто получить качественное решение, но и проверить как можно больше подходов к решению, причём как подходов к извлечению информации из признаков, так и подходов (алгоритмов) классификации. Некоторые результаты работы автор планирует популярно изложить в своём блоге[2], если получит разрешение организаторов.

### 1.1. Подход автора к решению задачи

Задача сведена к признаковой задаче классификации. Сначала сигналы предобрабатывались: проводилась фильтрация низких и высоких частот, выделялись отрезки, соответствующие разным кардиоциклам. Затем по каждому сигналу строилось его признаковое описание. Признаки генерировались на основе разных подходов:

- Фурье-анализ
- Сингулярное разложение
- Статистики (не вошли в финальное решение)
- Параметры, предоставленные организаторами
- по В.М. Успенскому (не вошли в финальное решение)
- Вейвлет-анализ (не вошли в финальное решение)

Ниже каждый подход подробно описан и обоснован (см. разделы с соответствующими названиями). В процессе работы над задачей проводилась многоэтапная селекция признаков. Основная идея – сделать классификацию с разнородными признаками, взять от каждого подхода около 10 лучших признаков.

[2] https://alexanderdyakonov.wordpress.com/

Селекция также подробно описана в отчёте (раздел **Генерация и селекция признаков**).

Классификация выполнялась регрессионными алгоритмами, которые оценивали степень принадлежности к классу 1 («курильщики»), затем она [степень] сравнивалась с порогом. Такая организация классификации позволяла находить компромисс между значениями чувствительности (Se) и специфичности (Sp) - полное объяснение см. раздел **Классификация сигналов**.

Были подробно исследованы возможности следующих алгоритмов:

- Линейная и гребневая регрессии
- Логистическая регрессия
- Случайный лес
- Экстремально случайные деревья
- Бустинг над деревьями

В отчёте приведены подробные таблицы качества их настройки на локальном тесте (с помощью скользящего контроля по одному – LOO).

В результате получено решение, которое превосходит решения других участников по F-мере и функционалу Se+Sp (в турнирной таблице оно заняло 2е место).

### 1.2. Реализация подхода

Технически задача решалась следующим образом. В системе Matlab 2012b была произведена загрузка и обработка данных. Затем для каждого сигнала (из обучающей и контрольной выборки) были вычислены признаки. Все они были сохранены в виде csv-файлов (в которых записаны стандартные признаковые матрицы объект-признак). Затем признаковые матрицы загружались средствами Python 3.5.0 (Anaconda 2.4.0 64-bit) и использовались для настройки регрессоров из библиотеки scikit-learn. Файл с ответами регрессора сохраняется на диск. Файл для отправки организаторам формируется в среде Matlab (считывается файл с ответами и переводится в нужный формат).

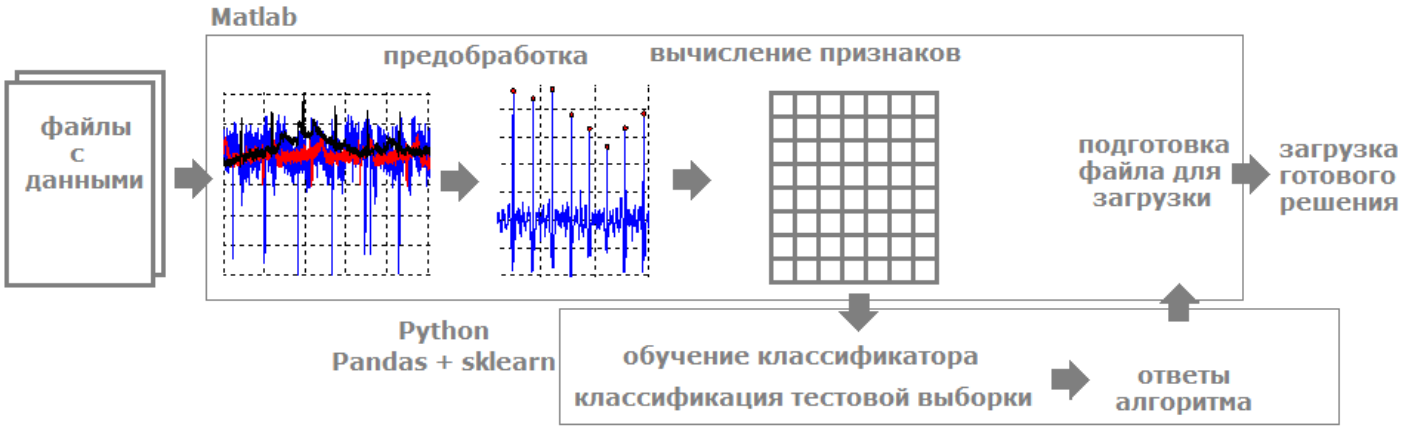



**Рис. 1.1. Общий план формирования итогового ответа**

Использование двух разных сред (Matlab и Python) объясняется утилитарными соображениями: в Matlab удобнее работать с сигналами и выделять признаки, для Python есть библиотека машинного обучения scikit-learn, в которой реализованы все потенциально полезные алгоритмы. При желании весь код может быть переписан, например на языке программирования Python.

### 1.3. Как воспроизвести финальное решение

Далее в отчёте разбирается весь код «по кусочкам». Здесь же – напишем инструкцию для воспроизведения финального результата.

1. Создайте каталог и перенесите в него весь код. Сделайте в нём два подкаталога train и test, разместите в них соответственно обучающую[3] и контрольную выборки (можно обучающие и контрольные файлы держать и в других каталогах, но тогда их надо явно прописать в процедурах cardio_loadtrain и cardio_loadtest). Откройте Matlab и сделайте созданный каталог текущим.

2. Запустите скрипт **cardio_runme.m** (можно просто набрать команду cardio_runme1 в системе Matlab). В результате посчитаются и сохранятся в файлы все признаки.

3. Запустите код для IPython notebook **cardio-makefinalsolution.ipynb**. В результате выполнится регрессия, и результаты будут сохранены в файле.

4. Запустите скрипт **cardio_makemyfinans.m** в среде Matlab – будет сформирован файл ответа идентичный тому, что был прислан организаторам.

[3] Это содержание архива **Обучающая выборка.rar**, предоставленного организаторами. Только в нём все файлы лежат в каталоге с названием «Обучающая выборка» – их надо перенести в каталог train. Аналогично с контрольной выборкой.

Выполнение всех этапов занимает чуть более 30 минут. Время может быть существенно сокращено (например, при вычислении каждой группы признаков все сигналы загружаются в память – это можно сделать лишь один раз в начале вычислений).

## 2. Загрузка данных

Вручную был создан файл **traininfo.txt**. Он содержит ту же информацию, что и файл **_Обучающая выборка_.txt**, но в более удобной для считывания форме, см. таблицу 2.1 – последний столбец данных - целевой вектор (1 - если человек курящий, 0 – иначе). Для загрузки этого файла в Matlab используется функция **cardio_importfile**.

**Табл. 2.1. Файлы со значением целевого признака (данный и созданный).**

| файл _Обучающая выборка_.txt | файл traininfo.txt |
|---|---|
| \| ФИО \| пол \| возраст \|<br><br>1.1. Обучающая выборка (50 человек). Курящие.<br><br>BRA \| М \| 37 \|<br>CZA \| Ж \| 31 \|<br>KNN \| М \| 52 \|<br>PPV \| М \| 32 \|<br>SAE \| М \| 47 \|<br>ZLS \| Ж \| 27 \|<br>ATA \| Ж \| 46 \|<br>... | BRA, 1, 37, 1<br>CZA, 0, 31, 1<br>KNN, 1, 52, 1<br>KVA, 1, 32, 0<br>PPV, 1, 32, 1<br>SAE, 1, 47, 1<br>ZLS, 0, 27, 1<br>ATA, 0, 46, 1<br>АШМ, 1, 22, 1<br>БАС, 1, 27, 0<br>БВА, 1, 44, 1<br>БДИ, 1, 33, 0<br>... |

Для загрузки данных написаны следующие m-процедуры (они в дальнейшем используются в процедурах генерации признаков и создании файла-решения)

**Табл. 2.2. Процедуры для загрузки сигналов.**

| | |
|---|---|
| **cardio_loadtrain.m** | Загрузка wav-файлов для обучения.<br><br>В результате в массиве ячеек S – сигналы, а в массиве ячеек FN – названия файлов (которые соответствуют сигналам).<br><br>При желании использовать – необходимо указать в коде каталог с данными! |

| | |
|---|---|
| **cardio_loadtest.m** | Загрузка wav-файлов для контроля.<br><br>В результате в массиве ячеек S2 – сигналы, а в массиве ячеек FN2 – названия файлов (которые соответствуют сигналам).<br><br>При желании использовать – необходимо указать в коде каталог с данными! |
| **cardio_importfile.m** | Загрузка csv-файла. |

На рис. 2.1–2.3 показана визуализация считанных сигналов[4]

- ATA_02-01-2015_21-39-50_300_926258366804_1000hz_int16.wav (сигнал 8)
- КПА_13-09-2015_08-59-49_300_906005931418_1000hz_int16_I.wav (сигнал 41)
- BRA_10-12-2015_11-58-45_300_212626192412_1000hz_int16_I.wav (сигнал 1)

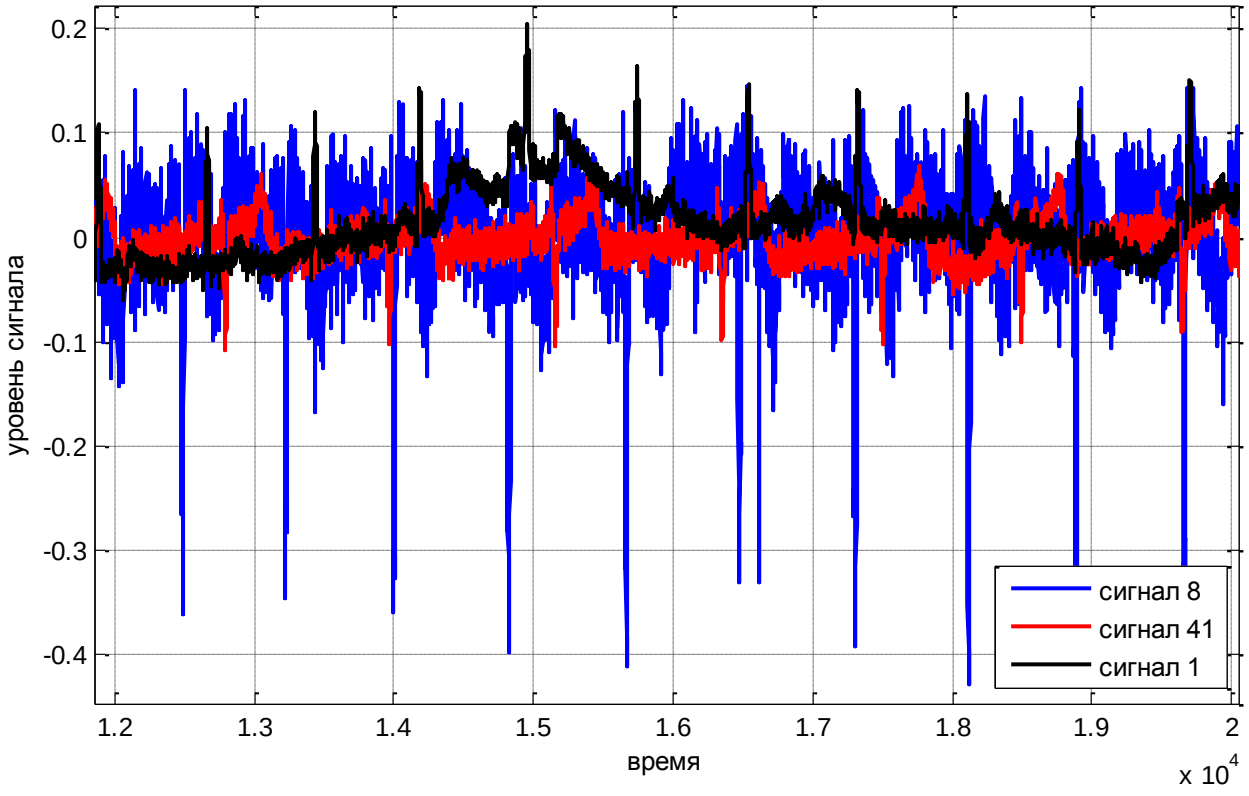



**Рис. 2.1. Визуализация загруженных сигналов.**

[4] Все графики построены в системе Matlab.

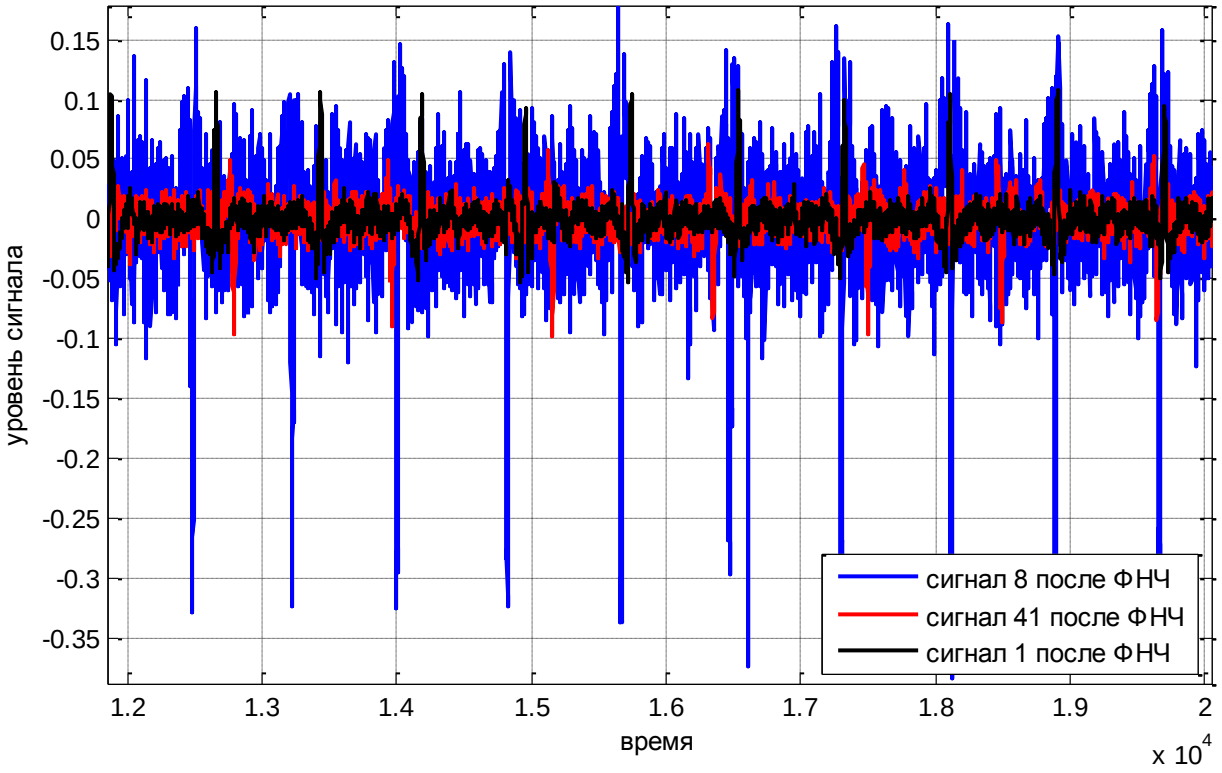



**Рис. 2.2. Сигналы после использования фильтра низких частот.**

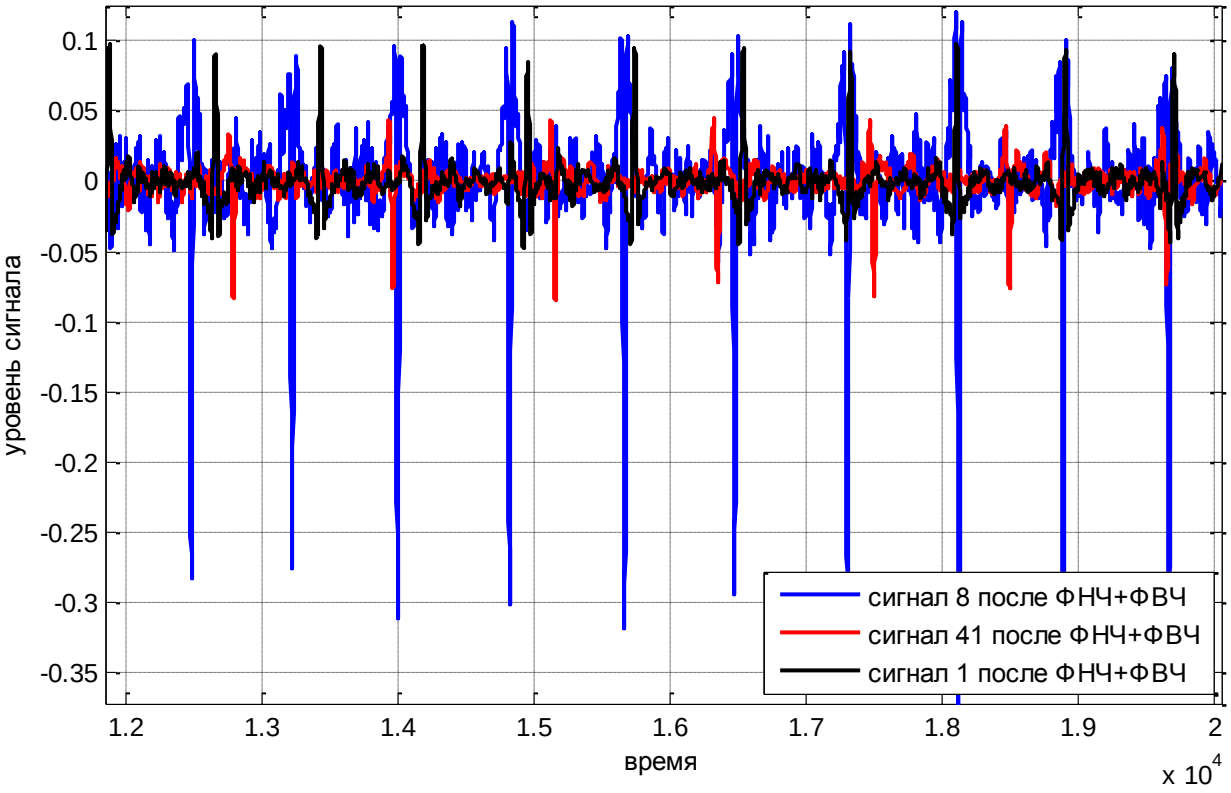



**Рис. 2.3. Сигналы после использования фильтров низких и высоких частот.**

Также для загрузки данных, которые предоставили организаторы созданы

| | |
|---|---|
| **cardio_readalldata.m** | Загрузка данных с помощью функций<br>**cardio_readucc**<br>**cardio_readpar** |
| **cardio_readucc.m** | Загрузка данных из файла вида *ucc*.csv (см. раздел **Признаки, предоставленные организаторами**) |
| **cardio_readpar.m** | Загрузка данных из файла вида *.csv, в названии которого нет подстроки 'ucc' (см. раздел **Признаки, предоставленные организаторами**) |

## 3. Предобработка данных

### 3.1. Разведочный анализ данных

Вначале сигналы были визуализированы. Нетрудно видеть, что некоторые сигналы являются «перевёрнутыми» кардиограммами (см. рис. 2.1–2.3) – их следует домножить на (–1), чтобы привести к нормальному виду и в дальнейшем выделять в них участки, соотвествующие отдельным циклам работы сердца (кардиоциклам).

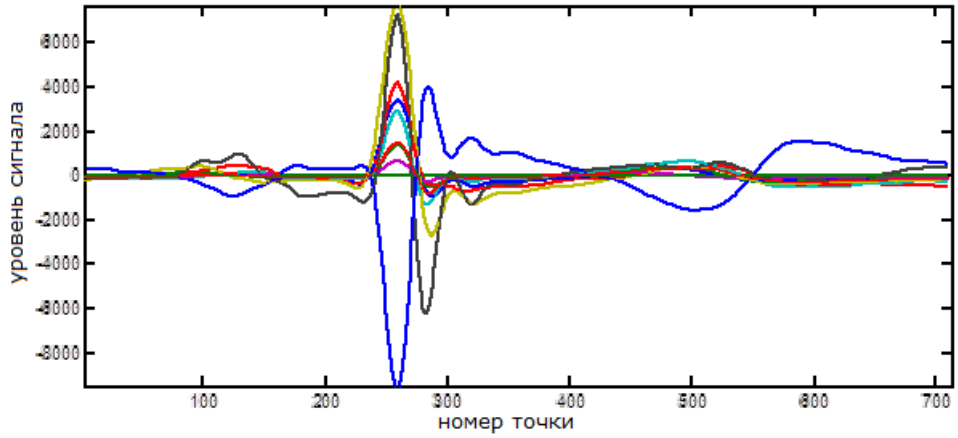



**Рис. 3.1. Главные волны, предоставленные организаторами конкурса**

Кстати, на рис. 3.1 показаны волны, описания которых содержались в *ucc.csv файлах. Видно, что некоторые (например, синяя) выделены с точностью до множителя (–1).

### 3.2. Фильтрация данных

Для предварительной обработки сигналов написаны фильтры низких частот и высоких частот:

- **`cardio_removelowerfrequencies.m`,**

- **cardio_removehighfrequencies.m**

Фильтрация производится на основе дискретного преобразования Фурье (ДПФ): проводится ДПФ, зануляются нужные коэффициенты и производится обратное преобразование ДПФ. Какие именно коэффициенты занулять «подсмотрено» в сторонних библиотеках Matlab для анализа кардиограмм. Небольшой несложный код для иллюстрации:

```
% фильтр низких частот
function corrected = cardio_removelowerfrequencies(ecg, samplingrate)
    fresult=fft(ecg);
    fresult(1 : round(length(fresult)*5/samplingrate))=0;
    fresult(end - round(length(fresult)*5/samplingrate)+1 : end)=0;
    corrected=real(ifft(fresult));
```

Результаты работы фильтров можно посмотреть на рис. 2.1-2.3.

### 3.3. Выделение кардиоциклов

Для работы с сигналом может потребоваться разбиение его на отрезки, соответствующие отдельным циклам работы сердца. В принципе, организаторы предоставили подобное разбиение – информация о нём содержится в файлах *_filtered.csv. Но, как правило, оно не очень точное. На рис. 3.2 показаны концы отрезков, построенные по информации из файла *_filtered.csv для одного конкретного сигнала (ИСН_30-09-2015_15-02-25_300_488414082503_1000hz_int16_I.wav).

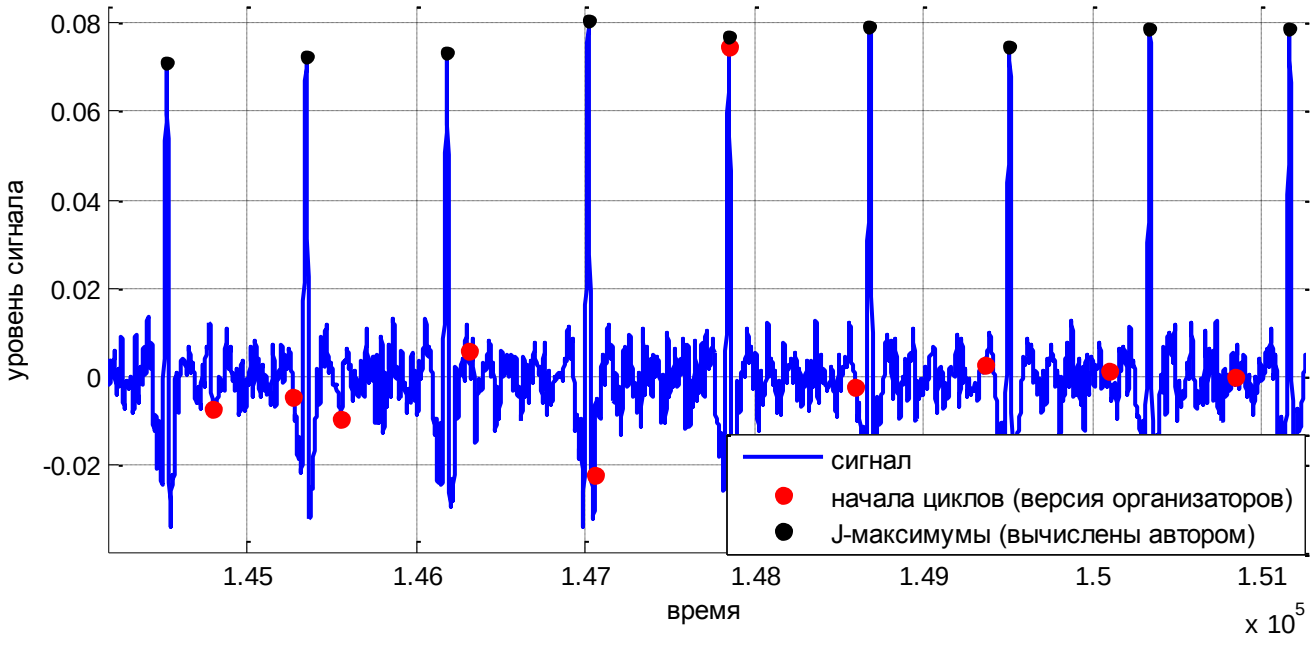



**Рис. 3.2. Концы отрезков отдельных кардиоциклов**

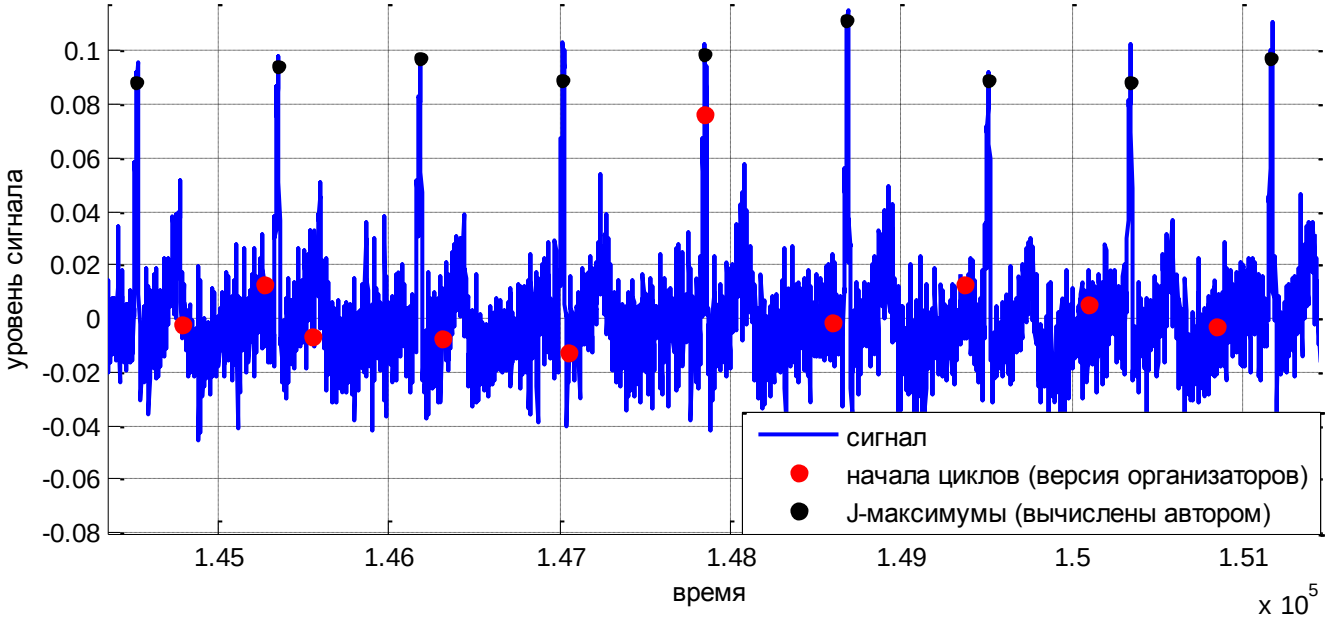



**Рис. 3.3. Концы отрезков отдельных кардиоциклов на исходном сигнале**

Видно, что разбиение, которое восстанавливается по информации, предоставленной организаторами, не совсем верно разбивает сигнал на искомые отрезки. Поэтому решено было написать свою функцию разбиения – поиска т.н. $J$-максимумов (определение ниже), которые соответствуют вершинам QRS-зубцов[5]. С биологической точки зрения, видимо, некорректно выбирать точкой начала цикла подобный максимум, но на большинство признаков, которые генерируются для классификации, выбор начала или не влияет, или признаки пересчитываются по расширенному интервалу (см. дальше).

Сначала использовались методы разбиения сигналов из различных библиотек, доступных в Интернете[6]. В результате, наилучшее разбиение показал авторский метод, который реализован в функции **`cardio_findmax2`**. Метод заключается в следующем. В сигнале

$$(x_1, x_2, \ldots, x_n)$$

находятся точки такие точки $x_i$, что

$$x_i = \max( x_{i-k_{\text{pred}}}, x_{i-k_{\text{pred}}+1}, \ldots, x_{i+k_{\text{next}}} ),$$

[5] https://ru.wikipedia.org/wiki/Электрокардиография

[6] Например, http://www.mathworks.com/matlabcentral/fileexchange/45404-ecg-q-r-s-wave-online-detector

то есть в этой точке достигается локальный максимум, причём это глобальный максимум в окрестности $k_{\text{pred}}$ точек слева и $k_{\text{next}}$ точек справа[7], и

$$\begin{gathered}\min(\, x_{i-r_{\text{pred}}}, x_{i-r_{\text{pred}}+1}, \ldots, x_i\,) < 0, \\ \min(\, x_i, x_{i+1}, \ldots, x_{i+r_{\text{next}}}\,) < 0,\end{gathered} \qquad (3.1)$$

то есть сигнал опускается ниже нулевого уровня слева и справа от точки $x_i$. Все такие найденные номера точек $i$ будем называть $J$-максимумами и считать, что они образуют множество $J$, $J \subseteq \{1,2,\ldots,n\}$.

Отметим, что $J$-максимумы ищутся уже на отфильтрованном сигнале (ФВН+ФНЧ), среднее значение которого лежит около нуля (см. рис. 3.2), поэтому нужны условия (3.1), они же позволяют отсеять другие локальные максимумы. При желании, по найденным точкам можно обнаружить пики и исходного сигнала (до фильтрации, см. рис. 2).

Параметры $k_{\text{pred}} = k_{\text{next}} = 400$, $r_{\text{pred}} = r_{\text{next}} = 50$ были подобраны вручную и позволяют в большинстве случаев правильно находить концы отрезков кардиоциклов (см. рис. 3.4).

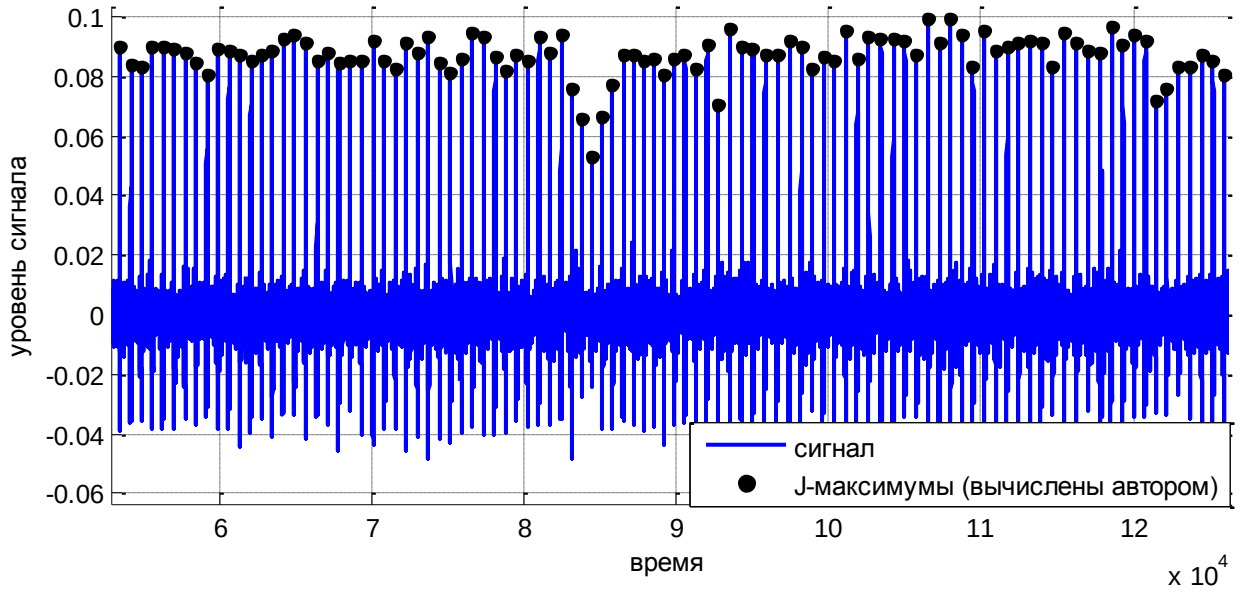



**Рис. 3.4. Концы отрезков отдельных кардиоциклов на первом сигнале**

[7] Ясно, что если точка $x_i$ лежит близко к краю сигнала, то

$$x_j = \begin{cases} x_1, & j < 1, \\ x_n, & j > n. \end{cases}$$

## 4. Генерация и селекция признаков

Задачу классификации признаков решено свести к классической признаковой классификации, когда каждый объект (в данном случае – сигнал) описывается фиксированным набором вещественных значений (признаков). Простейший пример признака: максимальное значение сигнала. Однако, признаки должны быть выбраны так, чтобы максимально точно описать сигнал[8], учесь физику описываемых сигналом процессов, при генерации признаков могут быть использованы различные эвристики (ранее накопленные специалистами при исследовании кардиограмм).

Для генерации признаков использовалось несколько подходов (ещё несколько были исследованы, но их результаты не использованы в финальном решении). Все они описаны ниже. Каждый подход даёт целый набор признаков, иногда несколько сотен, как например при использовании коэффициентов в разложении Фурье. Однако в классификаторах использованы не все признаки: от каждого подхода не больше 10. Это связано с тем, что многие признаки, полученные одним подходом, коррелируют, а большинство не улучшает качество классификации или даже приводит к переобучению[9]. Поэтому после генерации набора признаков производился отбор.

**Отбор признаков проходил в два этапа**. **Первый** – оценка априорного качества признака (до его использования для классификации). Он подробно описан в разделе **Оценка признаков**. **Второй** – по результатам классификации. Здесь имеется в виду

1) эксперименты с настройкой классификаторов на разных признаковых пространствах и оценкой качества классификации на скользящем контроле,

2) результаты оценки качества решений организаторами при предоставлении решений во время соревнования.

Второй этап не совсем формальный. Например, если классификатор, построенный на признаках, сгенерированных с помощью определённого подхода, показывал низкое качество на тестовой выборке (организаторы сообщали, что значения Se и Sp небольшие – по крайней мере, существенно ниже, чем в локальных тестах), то все признаки могли быть исключены из дальнейшего использования. В условиях конкурса (когда видишь текущие результаты других участников – и это подстёгивает искать

[8] Не описывая ненужную «шумовую» информацию о сигнале.

[9] **Переобучение** – эффект, возникающий при решении задач классификации, когда качество на тестовой выборке существенно ниже, чем на обучающей. Как правило, связан с использованием очень сложных моделей классификации, которые «слишком хорошо» настраиваются на обучающую выборку, вместо того, чтобы экстраполировать найденные простые закономерности.

более эффективные подходы) никакого конкретного порога (при непревышении которого признаки с соответствующим качеством отвергались) здесь не было.

Отметим, что проверка (1) нацелена на борьбу с переобучением и выявляет признаки с большой разделяющей способностью[10]. Проверка (2) нацелена на исследование инвариантности признаков, т.е. меняется ли распределение классов в соответствующем признаковом пространстве при переходе от обучающей к контрольной выборке. Как будет упоминаться далее, обучающая и тестовая выборки, действительно, отличались по своим статистическим свойствам.

Некоторые потенциально перспективные группы признаков не использовались из-за проверки (2). В данном отчёте мы не будем подробно выводить всю статистику по оценке признаков (её очень много, она перегрузит отчёт и не нужна для понимания принципов работы финального алгоритма).

## 5. Оценка признаков

Опишем метод, который использовался на первом этапе оценки качества признаков. Этот метод часто применяется автором при решении прикладных задач, правда, позволяет отобрать лишь признаки определённого вида (может пропустить хороший признак, у которого, например, все объекты одного класса имеют небольшие по модулю значения, а все объекты другого класса – большие).

Для оценки некоторого признака, значения которого равны $f_1,\ldots,f_m$ соответственно для $m$ объектов обучающей выборки, предположим, что в нашей задаче классификации на 2 класса 0 и 1 мы должны выдать вероятность принадлежности к классу 1. Напрямую используем для этого значения $f=(f_1,\ldots,f_m)$ нашего признака:

$$f_N=\left(\frac{f_1-\min(f)}{\max(f)-\min(f)},\ldots,\frac{f_m-\min(f)}{\max(f)-\min(f)}\right),$$

здесь $\max(f)=\max(f_1,\ldots,f_m)$, $\min(f)=\min(f_1,\ldots,f_m)$, $i$-й элемент вектора $f_N$ – это наш ответ на $i$-м объекте. Для ответов $f_N\in[0,1]^m$ и вектора правильных ответов $y\in\{0,1\}^m$ можно вычислить качество AUC ROC[11], которое принимает значение из отрезка $[0,1]$. Поскольку

[10] Неформально: определённые значения признаков (и близкие к ним) соответствует сигналам одного класса, а другие значениям – сигналам второго класса. На поиск таких признаков нацелен и первый этап.

[11] https://ru.wikipedia.org/wiki/ROC-кривая

$$\mathrm{AUCROC}(1-f_N, y) = 1 - \mathrm{AUCROC}(f_N, y),$$

$$(-f)_N = 1 - f_N,$$

то логично качество признака определить как

$$\max[\mathrm{AUCROC}(f_N, y), 1 - \mathrm{AUCROC}(f_N, y)] = |\mathrm{AUCROC}(f_N, y) - 0.5| + 0.5.$$

Действительно, хороший признак – это признак с помощью которого можно решить нашу задачу классификации с достаточно хорошим качеством. Мы предложили вычислять это качество в случае, когда для решения используется только этот признак ($f$) или его инверсия ($-f$).

Описанную оценку качества назовём априорной. Она позволяет оценить сразу все признаки, полученные с помощью некоторого метода генерации, найти среди них перспективную группу или отвергнуть этот метод для генерации признаков. Напомним, что в нашем подходе есть также и другие этапы оценки качества признаков (см. раздел **Генерация и селекция признаков**).

## 6. Признаки, основанные на разложении Фурье

При анализе сигналов один из самых стандартных и успешных методов – использование преобразования Фурье, в данном случае – одномерного дискретного прямого преобразования Фурье[12] (ДПФ). На рис. 6.1. видно, что вроде бы есть некоторые отличия в модулях[13] коэффициентов ДПФ у сигналов класса 1 (показаны красным цветом) и класса 0 (синим). Необходимо убедиться, насколько качественные признаки могут быть получены из этих коэффициентов для нашей задачи классификации сигналов.

Используя коэффициенты ДПФ добиться хороших результатов в классификации не удалось[14], в результате экспериментов решено было раскладывать не весь сигнал, а его участки. Затем разложения можно усреднить (получив более стабильные

---

AUC = Area under the curve, ROC = receiver operating characteristic, здесь речь о критерии качества, который часто используется для оценки качества бинарных классификаторов (точнее, целых параметрических семейств таких классификаторов). Мы здесь не будем подробно описывать определение функционала.

[12] http://www.exponenta.ru/soft/matlab/potemkin/book2/chapter8/fft.asp

[13] Можно рассматривать и вещественную часть.

[14] Стоит отметить, что именно с этих признаков автор и начал исследования. Сейчас, набрав некоторый опыт, автор не уверен, что рассмотрел все возможности использования коэффициентов ДПФ.

признаки)[15]. В системе Matlab есть специальная функция, которая делит сигнал на отрезки и делает ДПФ каждого их них – spectrogram (спектрограмма).

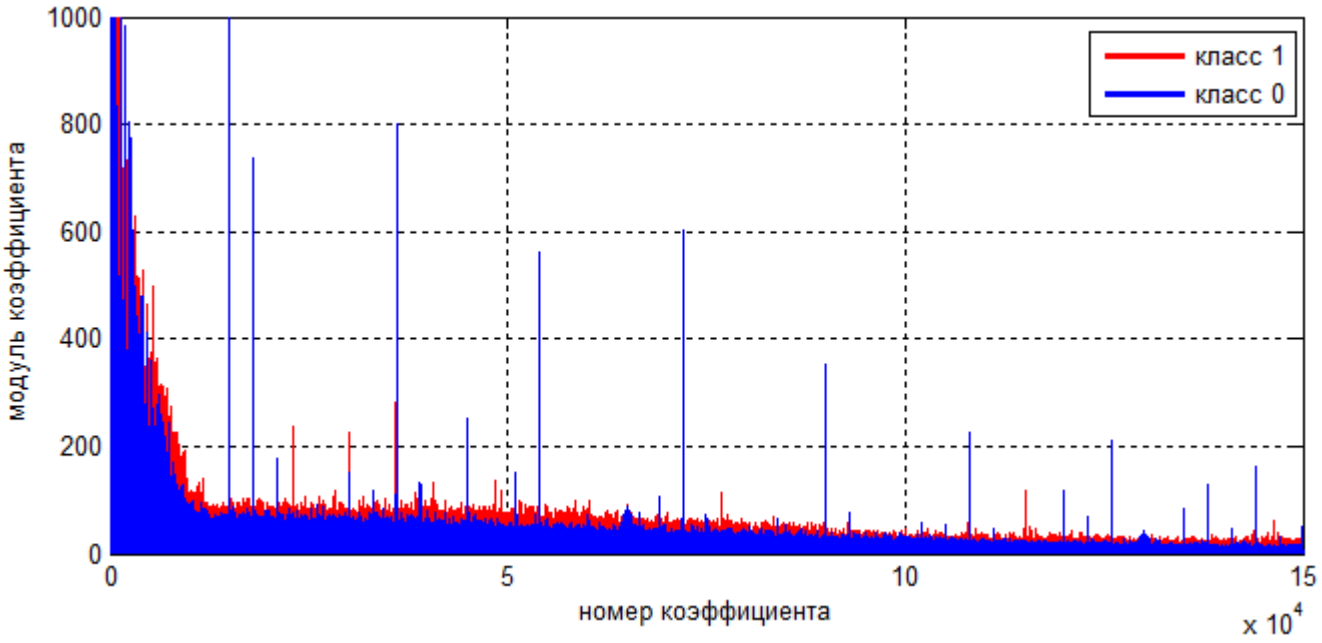



**Рис. 6.1. Значение модулей коэффициентов в ДПФ.**

Для генерации признаков, основанных на спектрограммах необходимо запустить

| | |
|---|---|
| **cardio_runSVDs.m** | Загрузка данных (обучение и контроль)<br>Формирование признаков.<br>Запись признаковых матриц для обучения и контроля:<br>**features_tempfftabsdiff1.txt**<br>**features_tempfftabsdiff2.txt** |

Для каждого сигнала $(x_1, x_2, \ldots, x_n)$ строится спектрограмма: сигнал разбивается на участки по 1000 точек (1 сек), всего $k \approx 300$ участков[16], для каждого сигнала выполняется ДПФ, берутся модули коэффициентов (всего 513 штук). Таким образом, получается вещественная матрица $H = \| h_{ij} \|$ размера $513 \times k$. По этой матрице строится 513-мерный вектор признаков

$$(f_1, \ldots, f_{513}),$$

$$f_i = \frac{1}{k-1} \sum_{j=1}^{k-1} |\log(h_{i,j+1} / h_{ij})| .$$

[15] Можно усреднять и коэффициенты в ДПФ всего сигнала, но использование предложенного подхода даёт возможность проанализировать гораздо больший набор признаков.

[16] т.е. коэффициенты будут описывать не весь сигнал, а только небольшой его фрагмент (чуть больше длины кардиоцикла).

На самом деле, изначально были опробованы различные функции над матрицей $H = \| h_{ij} \|$, например, такое усреднение

$$f_i = \frac{1}{k} \sum_{j=1}^{k} h_{ij} \qquad (6.1)$$

даёт просто усреднение коэффициентов разных ДПФ (на разных отрезках сигнала). Все они оказывались не очень качественными признаками. Идея полученной в итоге формулы – по аналогии с признаками В.М. Успенского[17], учитывать изменение сигнала в соседних отрезках кардиоциклов. Чуть больше одной пульсовой волны попадает в окно из 1000 точек (этим можно объяснить выбор такой длины),

$$| \log(h_{i,j+1} / h_{ij}) | = | \log h_{i,j+1} - \log h_{ij} | -$$

– это просто модуль изменения логарифма от модуля конкретного коэффициента ДПФ. Чем меньше это значение – тем больше текущий кардиоцикл похож на следующий. На рис. 6.2. показаны значения построенных признаков для 4х сигналов (два класса 0 и два класса 1)[18].

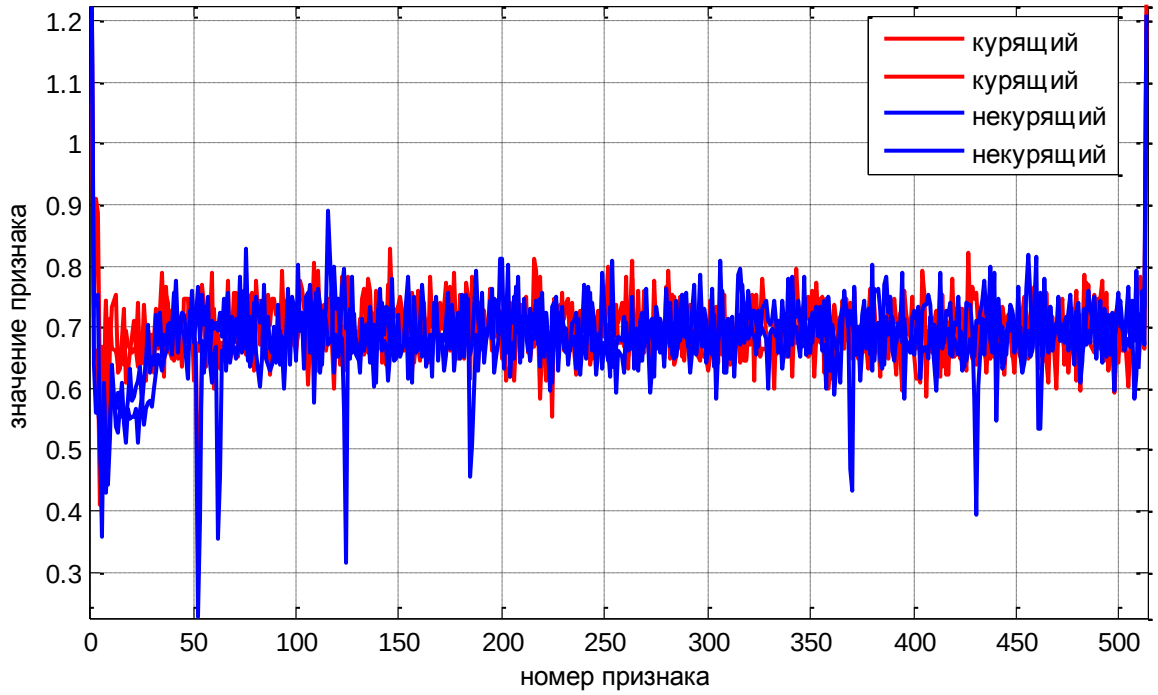



**Рис. 6.2. Значения признаков, построенных по спектрограммам для 4х сигналов.**

[17] См. далее и в http://www.machinelearning.ru/wiki/images/9/9a/Voron-2014-10-20-ecg.pdf

[18] Здесь и далее берутся первые 2 сигнала класса 0 и первые 2 сигнала класса 1.

В результате отбора использовались не все признаки (513 шт.), а лишь 5 из них – с номерами 8, 104, 296, 369, 506.

Если строить классификатор только на этих признаках, то получается качество FF-мера=**0.64**[19] с помощью Ridge-регрессора и выбора порога (см. раздел **Классификация сигналов**). С одной стороны, это не очень высокое качество, с другой – мы использовали только 5 признаков! И, как показали исследования, эти признаки успешно комбинируются с другими.

В заключение проиллюстрируем априорное качество признаков (6.1), т.е. фактически классического ДПФ (если ДПФ делать для всего сигнала картина будет примерно такая же) – см. рис. 6.2. Видно, что модули первых коэффициентов имеют высокое качество: 0.7 AUC ROC. Интересно, что именно эти коэффициенты зануляются в фильтре низких частот (и, по логике, должны быть бесполезны для решения данной задачи). Как показали загрузки решений – именно использование этих признаков приводит к низкому качеству на тестовой выборке.

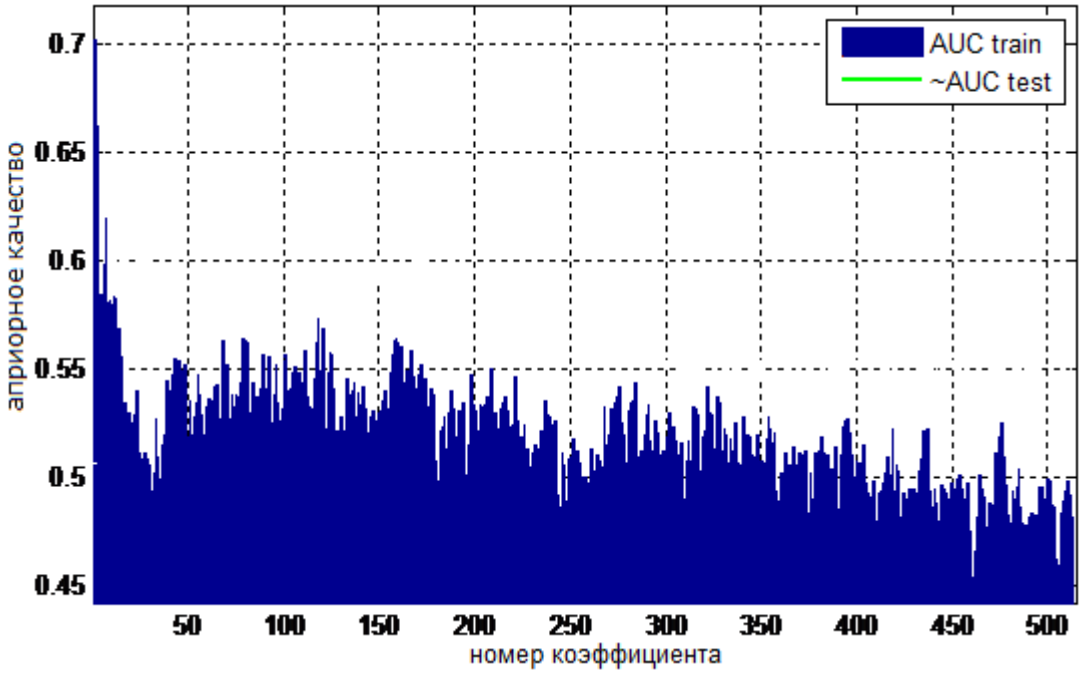



**Рис. 6.2. Априорное качество признаков (6.1).**

[19] Так мы (совсем нетрадиционно) называем среднее гармоническое чувствительности и специфичности. Напомним, что F-мера – среднее гармоническое точности и полноты (чувствительности). FF-мерой выражено желание заказчика найти компромисс между специфичностью и чувствительностью. https://en.wikipedia.org/wiki/Precision_and_recall

## 7. Признаки, основанные на сингулярном разложении

В анализе данных часто используются различные матричные разложения[20], т.е. представления одной матрицы в виде произведения других. Как правило, на множители накладывают дополнительные условия, в первую очередь – небольшой размер, тогда получаются экономные (часто – низкоранговые) приближения исходной матрицы. Особенно популярно классическое сингулярное разложение[21] (SVD = Singular Value Decomposition), с его помощь находят лучшее приближение в L2-норме среди всех матриц ранга не выше *k* (для заданного *k*). В данной задаче решено было также использовать всю мощь этого математического аппарата.

Для генерации признаков, основанных на сингулярном разложении матрицы необходимо запустить

| | |
|---|---|
| **cardio_runSVDs.m** | Загрузка данных (обучение и контроль)<br>Генерация признаков – вызов функции **cardio_extractSVDs**.<br>Формирование признаков.<br>Запись признаковых матриц для обучения и контроля:<br>**features_temp1.txt**<br>**features_temp2.txt** |

Реализованы следующие этапы обработки сигнала:

1. Фильтрация (ФНЧ+ФВЧ)
2. Нахождение $J$ -максимумов
3. Выделение непрерывных участков сигнала

$$(x_i, x_{i+1}, \ldots, x_{i+999}),\ i \in J\ .$$

Каждый участок начинается в точке $J$ -максимума и имеет длину 1000 пунктов (1 сек.).

4. Формирование матрицы $1000 \times k$ размера $1000 \times k$, где $k$ – число выделенных отрезков[22], в столбцах которой записаны векторы $(x_i, x_{i+1}, \ldots, x_{i+999})^{\mathrm{T}}$.
5. Выполнение сингулярного разложения полученной матрицы. Вычисляем только первые 3 компоненты разложения:

$$X \approx U_{1000\times 3} \cdot L_{3\times 3} \cdot V_{3\times k}\ .$$

При желании все элементы полученных матриц $U, L$ можно использовать в качестве признаков[23]. Но, во-первых, поскольку компоненты сингулярного разложения находятся с точностью до знака – мы будем брать абсолютные значения элементов матриц $U$. Во-вторых, подвергнем все эти признаки нашей процедуре селекции.

[20] http://www.math.mrsu.ru/text/courses/mcad/3.10.htm

[21] https://ru.wikipedia.org/wiki/Сингулярное_разложение

[22] Число $k$ может не совпадать с $|J|$, поскольку после последней точки $J$ -максимума может быть меньше 1000 измерений.

[23] При желании, можно и матрицу $V$ использовать.

Матрицу $L$ не будем использовать, поскольку априорное качество признаков, которые мы по ней получили низкое (около 50%). В качестве признаков были отобраны следующие элементы матрицы $|U|$[24]:

(1, 467), (1, 655),
(2, 138), (2, 266),
(2, 590), (3, 431),
(3, 574), (3, 812).

Выбор именно этих элементов обусловлен тем, что у этих признаков было достаточно высокое априорное качество (>0.6)[25], кроме того, только их использование уже показывало неплохие результаты в задаче (см. дальше).

На рис. 7.1 показаны непрерывные отрезки ряда, каждый из которых начинается в точке $J$-максимума. Выбор длины отрезка – 1 сек – обусловлен тем, чтобы в отрезок заведомо попадала одна пульсовая волна и, быть может, ещё небольшой участок сигнала. На рис. 7.2-7.4 показаны значения всех признаков, основанных на SVD, для 4х сигналов. Визуально заметны отличия в значениях некоторых признаков для кардиограмм курящих и некурящих. Однако, итоговый отбор признаков проводится не на основе визуального контроля, а на основе анализа априорного качества и качества по классификации.

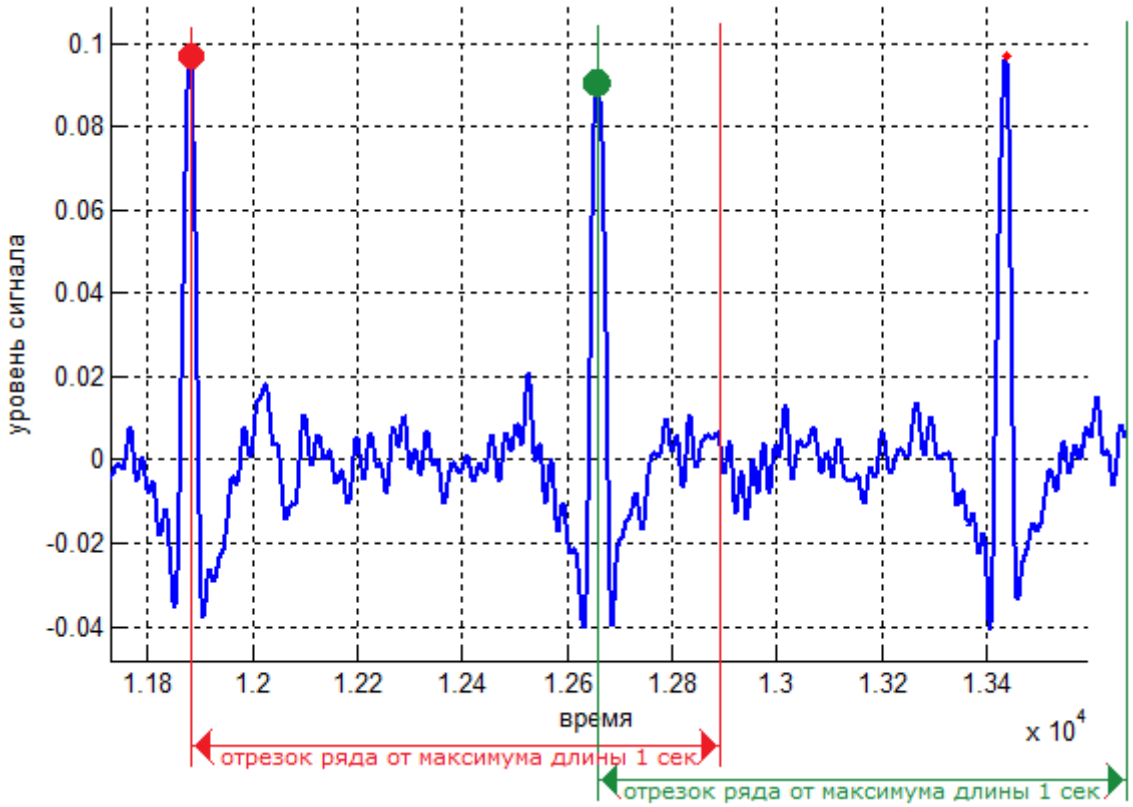



**Рис. 7.1. Участки сигнала, которые используются в SVD.**

[24] Здесь и далее $|U|$ – матрица, составленная из модулей элементов матрицы $U$ (не путать с обозначением мощности множества $U$, если $U$ множество, а не матрица).

[25] При подготовке этого отчёта автор заметил, что так было не у всех признаков.

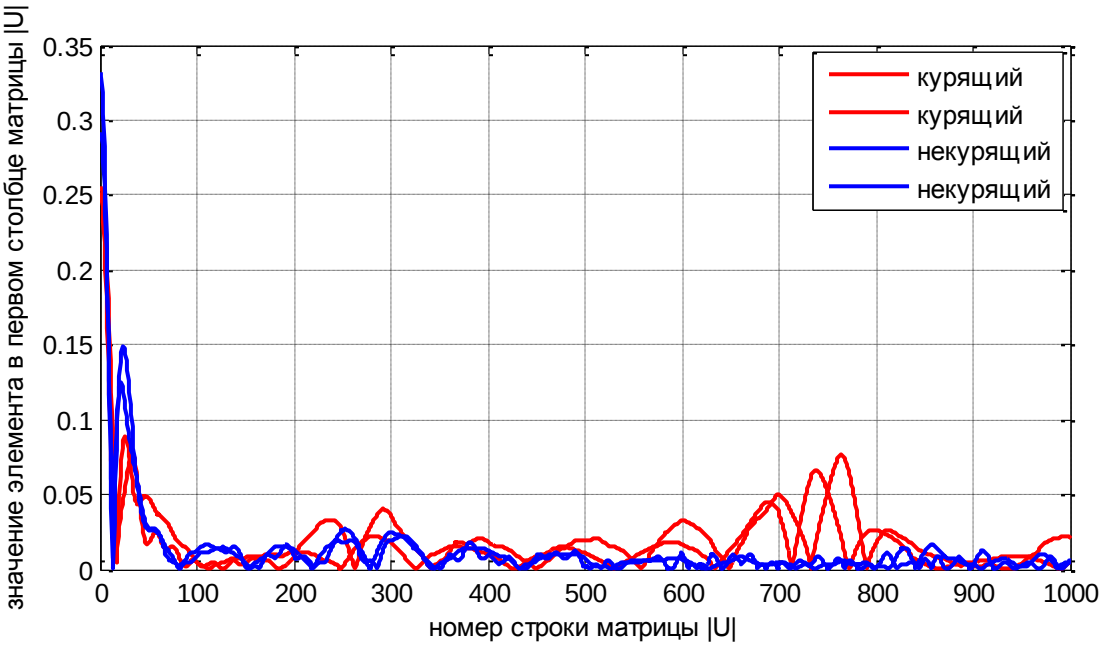



**Рис. 7.2. Значение признаков, основанных на SVD (I)**

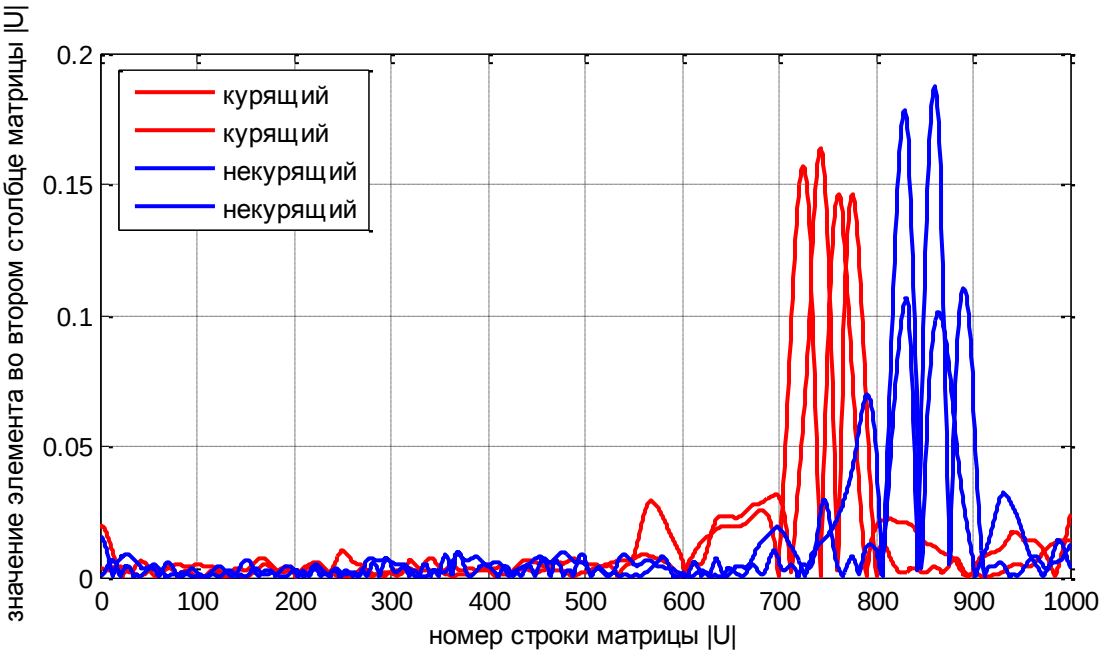



**Рис. 7.2. Значение признаков, основанных на SVD (II)**

Ещё одно замечание, которое необходимо сделать. Мы построили 8 признаков, в признаковой матрице, которая сохраняется для последующего использования для обучения 17 признаков: исходные 8, их квадраты и константный единичный признак (поскольку изначально признаки планировалось использовать в обобщённой линейной регрессии и там, кроме самих признаков, иногда полезно использовать их произведения).

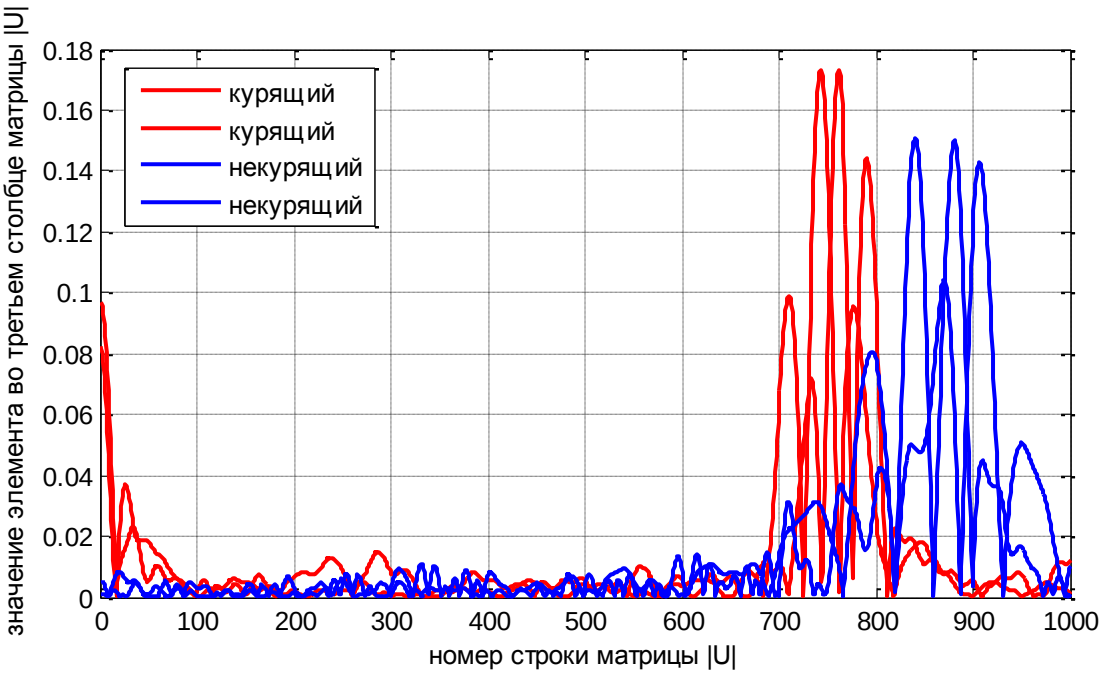



**Рис. 7.4. Значение признаков, основанных на SVD (III)**

Если строить классификатор только на этих признаках, то на локальном скользящем контроле качество может достигать FF-мера=(Se+Sp)/2=**0.72** (высокий для данной задачи показатель!). При этом на тестовой выборке максимальное качество (Se+Sp)/2=**0.66**. Интересно, что при применении бустинга качество и на локальном тесте и на итоговом – одинаковое (0.66), а случайные леса показывают 0.72 на локальном и только 0.63 – на итоговом. Качество Ridge-регрессии существенно ниже.

Подробнее об экспериментах по классификации в разделе **Классификация сигналов**.

## 8. Признаки, основанные на случайном сингулярном разложении

Предыдущая группа признаков описывала, фактически, циклы работы сердца, поскольку сингулярному разложению подвергались участки сигнала, которые начинались в J-максимумах. Возникла идея, что аналогичным образом можно описать сигнал в целом. Для этого надо в качестве начал 1000-точечных отрезков брать всевозможные точки сигнала[26]. В этом случае, правда, вычисления могут быть достаточно долгими (размер матрицы для сингулярного разложения будет около $1000 \times 300000$), поэтому было решено брать в качестве таких начал случайные точки (это множество точек сигнала случайно, но фиксировано).

[26] Можно, конечно, и увеличивать длину отрезка, но в итоговом решении использовалась такая фиксированная длина.

| | |
|---|---|
| **cardio_runSVDs_all.m** | Загрузка данных (обучение и контроль)<br>Генерация признаков – вызов функции<br>**cardio_extracttotalSVD**.<br>Формирование признаков.<br>Запись признаковых матриц для обучения и контроля:<br>**features_temp_allrand1.txt**<br>**features_temp_allrand2.txt** |

Были реализованы следующие этапы обработки сигнала:

1. Фильтрация (ФНЧ+ФВЧ)

2. Генерация множества $I$ случайных точек от 1 до 290000 (после них заведомо будет участок сигнала длины 1000), $|I|=10000$.

3. Выделение непрерывных участков сигнала

$$(x_i, x_{i+1}, \ldots, x_{i+999}),\ i \in I .$$

Каждый участок начинается в точке из $I$ и имеет длину 1000 пунктов (1 сек.).

4. Формирование матрицы $X$ размера $1000 \times 10000$

5. Выполнение сингулярного разложения полученной матрицы. Вычисляем только первые 2 компоненты разложения:

$$X \approx U_{1000\times 2} \cdot L_{2\times 2} \cdot V_{2\times 10000}.$$

Как показывают рис. 8.1-8.2 здесь сингулярные компоненты отличаются от компонент на рис. 7.2-7.4, поскольку описывают не пульсовые волны а участки сигнала в выбранном случайном множестве точек[27].

В итоге, в качестве признаков были отобраны следующие элементы матрицы $|U|$:

(1, 102), (1,232), (1,416), (1,714), (1,781),

т.е. использовалась только первая компонента разложения. Как и для предыдущей группы признаков, здесь были сгенерированы признаки-квадраты и константный единичный (всего 11 признаков).

Здесь признаков меньше, при классификации по этим признакам FF-мера и (Se+Sp)/2 не превосходят порога **0.62** на локальных тестах, правда, на тестовой выборке качество может быть даже лучше, чем при локальном тестировании: (Se+Sp)/2 = **0.63** (см. раздел **Классификация сигналов**).

[27] Хотя множество точек случайное, оно построено с помощью генератора псевдослучайных чисел с предустановленным random.seed, поэтому финальное решение может быть в точности повторено.

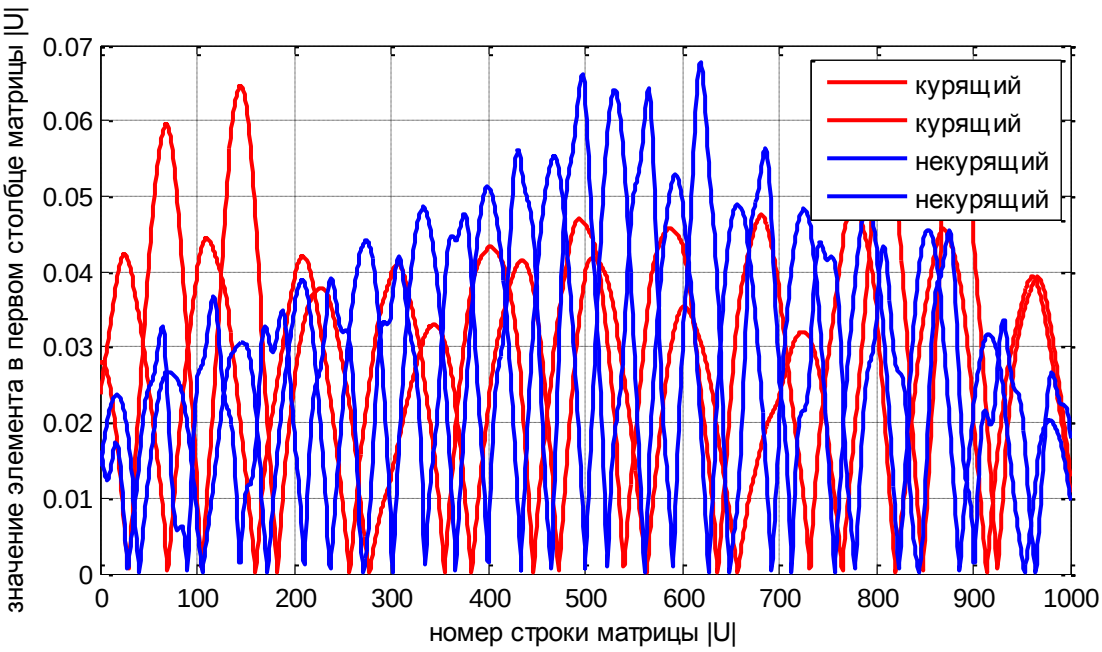



**Рис. 8.1. Значение признаков, основанных на случайном SVD (I)**

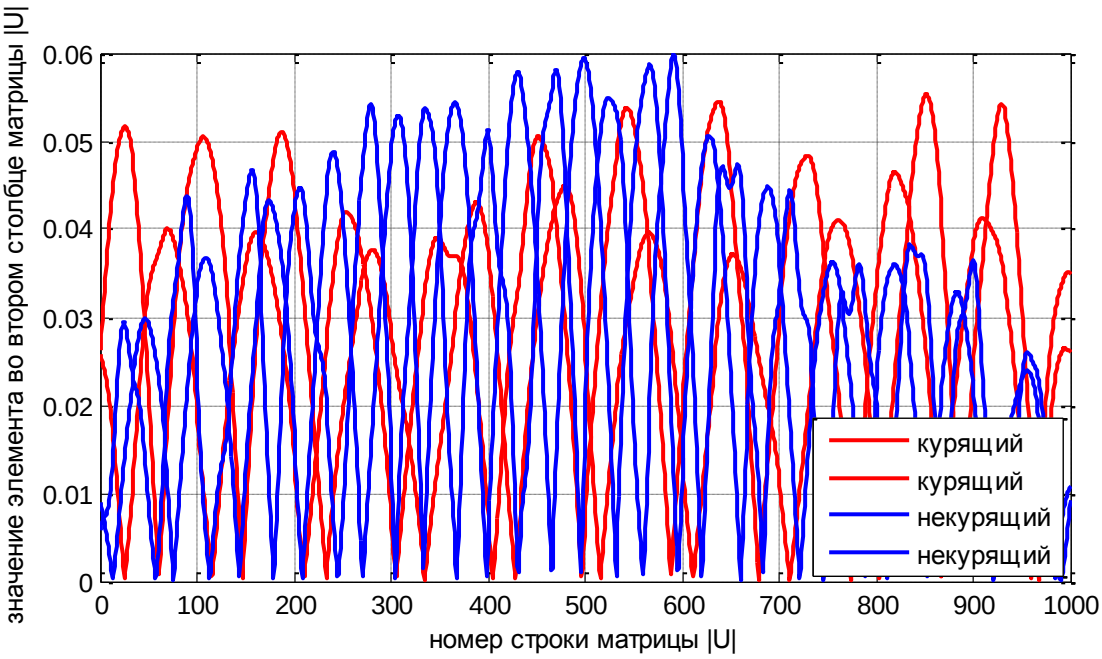



**Рис. 8.2. Значение признаков, основанных на случайном SVD (II)**

## 9. Признаки, предоставленные организаторами

Здесь идёт речь о некоторых параметрах сигналов, которые предоставили организаторы. См. табл. 9.1 – все эти значения загружались из файлов и считались признаковыми описаниями. В итоге – в финальной модели – использовались не все признаки, а только следующие (в табл. 9.1 показаны красным):

из *ucc.csv файла – quality, sqrs, 6-е числовое значение;
из другого csv-файла – HR, SI, VLF.

Для считывания этих признаков необходимо запустить

| | |
|---|---|
| **cardio_runstandartfeatures.m** | Загрузка данных с помощью функции<br>**cardio_readalldata,**<br>которая, в свою очередь, вызывает функции<br>**cardio_readucc**<br>**cardio_readpar**<br>Запись признаковых матриц для обучения и контроля:<br>**features_tempsf1.txt**<br>**features_tempsf2.txt** |

**Табл. 9.1. Содержимое некоторых файлов, предоставленных организаторами**

| **Фрагмент csv-файла, содержащий в названии "ucc":**<br><br>**CZA_09-12-2015_02-27-31_300_87302162855_1000hz_int16_l_ucc.csv** | **Фрагмент csv-файла, НЕ содержащий в названии "ucc":**<br><br>**KNN_09-12-2015_13-16-17_300_88649561808_1000hz_int16_l.csv** |
|---|---|
| error, 0<br>**quality, 48**<br>spqrst, 115.801916104722240<br>spq, 14.190019818114955<br>**sqrs, 41.586460006008991**<br>sst, 59.911544840698710<br><br>-43.498732254252765<br>-43.316289798709320<br>-43.140302991464743<br>-42.993737422959335<br>-42.897042604584314<br>**-42.862033117616917**<br>-42.887039393198449<br>-42.954988907959326<br>-43.035577360470192<br>... | **HR, 75, bpm**<br>SDNN, 30, ms<br>CV, 3.7, %<br>**SI, 120.3,**<br>IRSA, 7,<br>NArr, 1.6, %<br>NN50, 10,<br>pNN50, 2.7, %<br>VLF, 12.4, %<br>LF, 86.2, %<br>HF, 13.8, %<br>TP, 1106, ms^2<br>**VLF, 137, ms^2**<br>LF, 834, ms^2<br>HF, 135, ms^2<br>LF/HF, 6.23,<br>IC, 8.0, |

Если только на этих признаках построить классификаторы, то получаются такие результаты – см. табл. 9.2. Красным показано качество решения, отправленного организаторам (т.е. на тестовой выборке). Видно, что всего на 6 признаках, которые уже были предоставлены вместе с данными, качество на локальном контроле очень высокое – до 0.7134 по FF-мере, однако на тестовой выборке оно сильно падает (фактически, до качества случайного решения, см. раздел **Классификация сигналов**). Это позволило предположить, что контрольная выборка отличается по своим

свойствам от обучающей и заставило искать более стабильные признаки, однако все перечисленные 6 использовались в финальном решении.

**Табл. 9.2. Результаты классификации с помощью предоставленных признаков**

| Алгоритм | FF-мера / (Se+Sp)/2 |
|---|---|
| **RandomForestRegressor (n_estimators=100, criterion='mse', min_samples_split=2, min_samples_leaf=1, min_weight_fraction_leaf=0.0, max_features=2, random_state=1)** | 0.693 / 0.71 |
| **GradientBoostingRegressor(n_estimators=100, learning_rate=0.01, max_depth=2, random_state=1, loss='ls')** | 0.7134 / 0.73<br>0.5519[28] |
| **Ridge(alpha=0.001, normalize=True)** | 0.6519 / 0.69 |

## 10. Генерация собственных статистических признаков

Эта группа признаков показывала неплохие результаты в локальных экспериментах, но не вошла в финальное решение. Сначала каждый сигнал предобрабатывался, затем вычислялись некоторые статистические признаки как для самого сигнала, так и для его производных. Вот формальный перечень этапов для генерации признаков:

1. Фильтрация (ФНЧ/ФВЧ/сглаживание)

2. Для сигнала $(x_1, x_2, \ldots, x_n)$, его модуля $(|x_1|, |x_2|, \ldots, |x_n|)$, его производной $(x_2 - x_1, \ldots, x_n - x_{n-1})$, модуля производной $(|x_2 - x_1|, \ldots, |x_n - x_{n-1}|)$ и её производной вычисляются следующие значения признаков (покажем на примере исходного сигнала)

1) среднее значение сигнала $\text{mean} = \frac{x_1 + x_2 + \ldots + x_n}{n}$

2) стандартное отклонение $\text{std} = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}\left(x_i - \frac{x_1 + x_2 + \ldots + x_n}{n}\right)^2}$

3) доля пересечений с уровнем $a$ (для $a = 0$, $a = \text{mean}$, $a = \text{mean} + \text{std}$)

[28] Здесь и далее красным цветом показано качество на тестовой выборке (после деления на 2), если соответствующее решение отправлялось организаторам.

$$\frac{|\{i \in \{1,2,\ldots,n-1\} \,|\, (x_i - a)\cdot(x_{i+1} - a) \le 0\}|}{n-1}$$

4) разность долей пересечений с уровнем $a = \text{mean} + \text{std}$ и $a = \text{mean} - \text{std}$

Заметим, что здесь может применяться сглаживание сигнала: значение $x_i$ заменяется на

$$\frac{x_{i-k} + x_{i-k+1} + \ldots + x_i + \ldots + x_{i+k}}{2k+1}$$

(для более адекватных значений признаков, которые вычисляются по производным).

Причина использования такого набора признаков простая: первые два – стандартные статистики, третий часто используется при анализе сигналов (правда, звуковых). Четвёртый оценивает «симметрию сигнала» (хотя интуитивно ясно, что он несимметричен относительно горизонтальной оси).

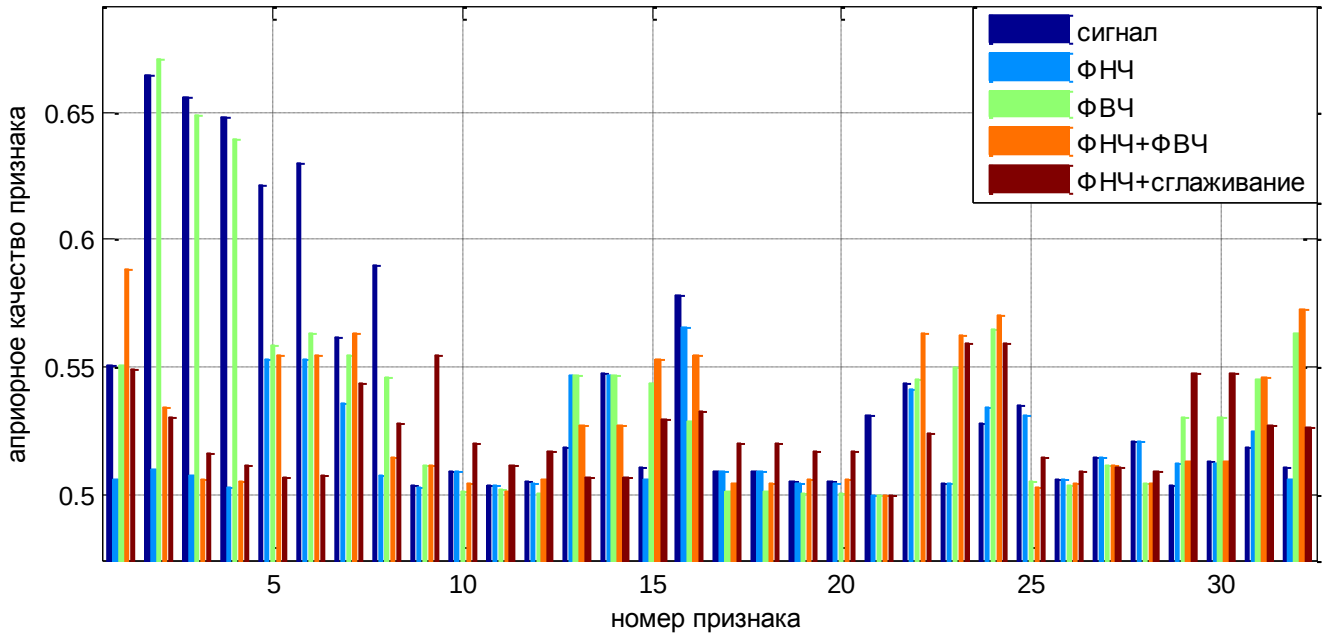



**Рис. 10.1. Априорное качество статистических признаков**

Всего было построено 32 признака. На рис. 10.1 показано их априорное качество, видно, что оно не очень высокое и достигает максимальных значений на исходном сигнале или при использовании только ФВЧ.

Если эти признаки использовать для классификации (32*5=160 признаков), то качество также не очень высокое (см. табл. 10.1), например FF-мера не превышает 0.65. В комбинации с другими признаками также не наблюдалось улучшения качества.

**Табл. 10.1. Результаты классификации с помощью статистических признаков**

| Алгоритм | FF-мера / (Se+Sp)/2 |
|---|---|
| **RandomForestRegressor (n_estimators=100, criterion='mse', min_samples_split=2, min_samples_leaf=1, min_weight_fraction_leaf=0.0, max_features=3, random_state=1)** | 0.5775 / 0.64 |
| **GradientBoostingRegressor(n_estimators=100, learning_rate=0.01, max_depth=1, random_state=1, loss='ls')** | 0.6448 / 0.66 |
| **Ridge(alpha=0.001, normalize=True)** | 0.5727 / 0.66 |

## 11. Признаки по В.М. Успенскому и К.В. Воронцову

В последнее время в русскоязычном сообществе аналитиков широкую известность приобрёл подход В.М. Успенского к анализу кардиограмм, в основном, благодаря работам К.В. Воронцова[29]. Очень подробно здесь описывать этот подход мы не будем, судя по рекламной публикации на habrahabr.ru[30], он знаком организаторам. В общих чертах, суть в следующем. Сигнал разбивается на отрезки, соответствующие разным кардиоциклам. Для каждого отрезка вычисляется несколько характеристик – они показаны на рис. 11.1:

$$R_n, T_n, \alpha_n, \ n - \text{номер цикла.}$$

Затем весь сигнал кодируется некоторым словом, $n$-я буква в котором зависит от знаков выражений

$$R_{n+1} - R_n, \quad T_{n+1} - T_n, \quad \alpha_{n+1} - \alpha_n.$$

Всего возможны 6 вариантов знаков[31], поэтому в слове будут 6 различных букв. Затем по всем триграммам слова (последовательные три буква) вычисляется их частота, см. рис. 11.2. Заметим, что всего возможно $6^3 = 216$ триграмм.

[29] http://www.machinelearning.ru/wiki/images/9/9a/Voron-2014-10-20-ecg.pdf

[30] https://habrahabr.ru/post/277287/

[31] У трёх независимых чисел возможно 8 комбинаций знаков: (+,+,+), (+,+,-), ..., (-,-,-), но здесь выражения зависимые.

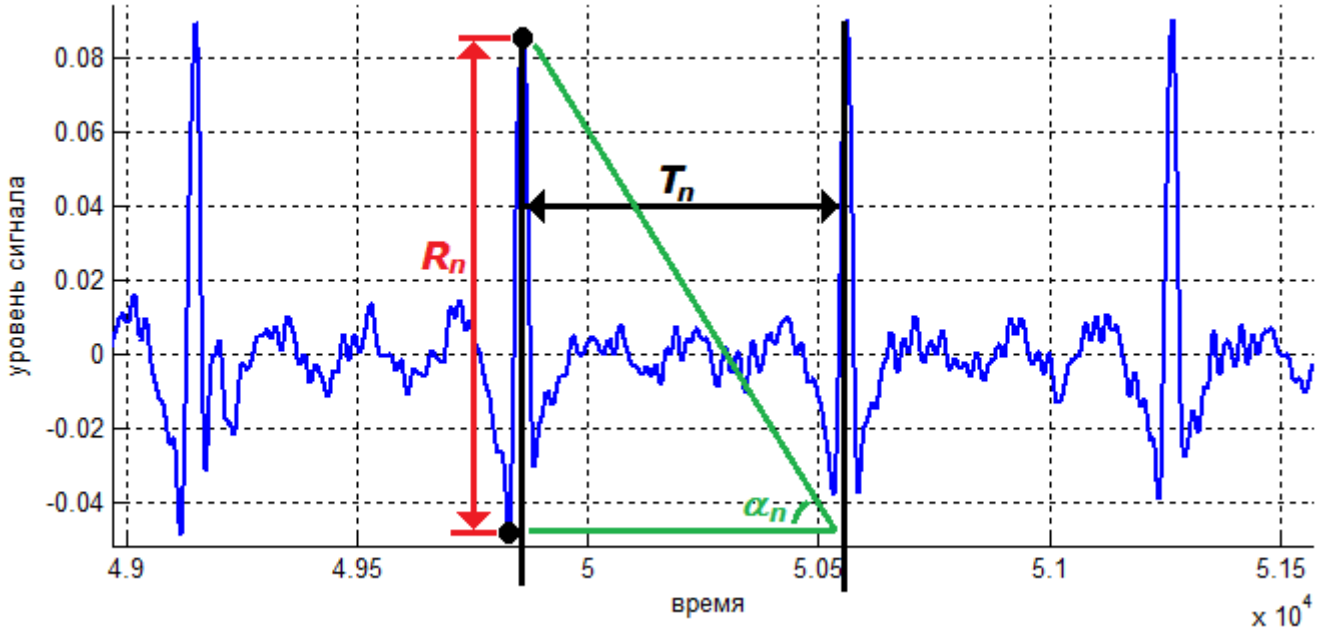



**Рис. 11.1. Длины отрезков, которые используются в методе В.М. Успенского**

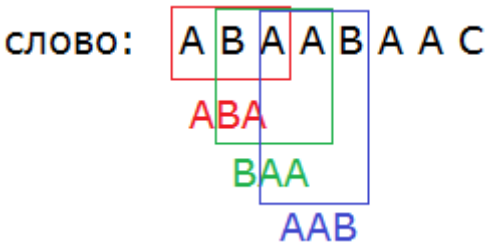



**Рис. 11.2. Триграммы слова.**

В работах В.М. Успенского и К.В. Воронцова показано, что частоты триграмм могут быть хорошими признаками при диагностике заболеваний (причём, не только напрямую связанных с проблемами сердца). Поэтому мы провели аналогичную генерацию признаков, кроме триграмм, рассматривали частоты букв и биграмм, всего

$$6^3 + 6^2 + 6 = 258 \text{ признаков.}$$

На рис. 11.3 показано априорное качество найденных признаков (признаки упорядочены по возрастанию качества). К сожалению, оно не очень высокое. Парадоксально, но если сгенерировать случайную матрицу признаков, то их качество будет выше – см. рис. 11.3. Качество классификации при использовании этих признаков также низкое: FF-мера не превышает **0.57**, (Se+Sp)/2 – **0.59**.

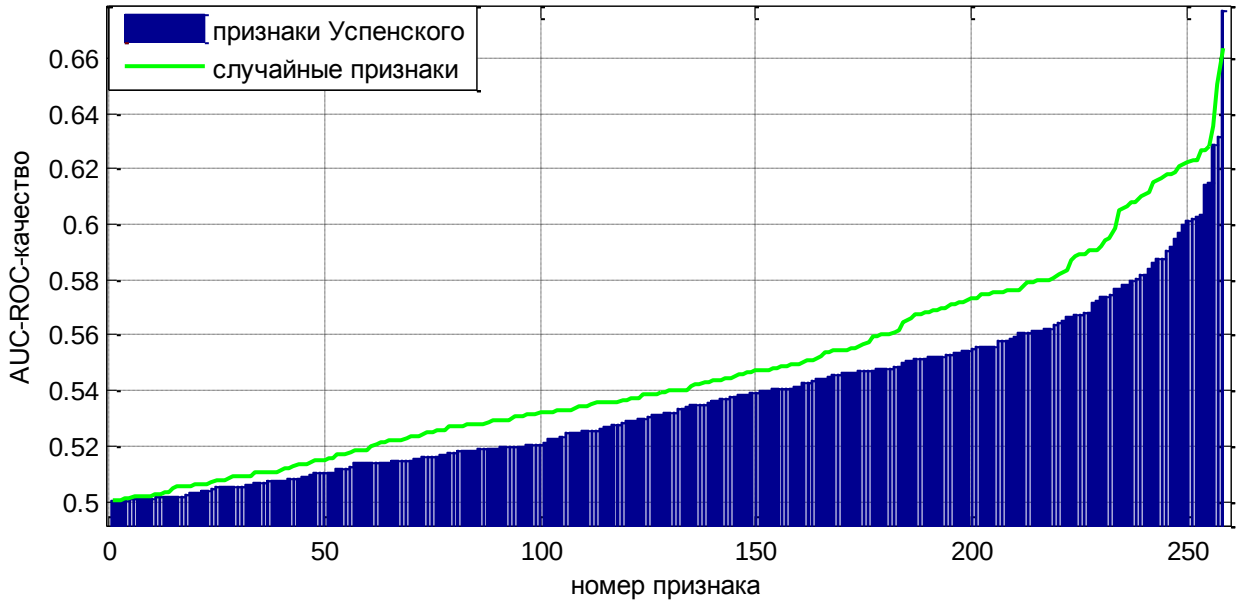



**Рис. 11.3. Качество признаков.**

Параллельно были исследованы и другие признаки, связанные с анализом визуальных характеристик сигналов разных кардиоциклов и их изменениями, к сожалению, качественных признаков построить не удалось. Ниже опишем возможные причины повала этого подхода к синтезу признаков. Отметим, что мы не опровергли работы В.М. Успенского и К.В. Воронцова, поскольку, во-первых, решали другую задачу, во-вторых, наши сигналы даже визуально сильно отличаются от сигналов на презентациях К.В. Воронцова, и, судя по всему, те сигналы сняты более точным оборудованием. Кроме того, нет уверенности, что мы не используем менее качественные процедуры выделения кардиоциклов или не знаем некоторых тонкостей в формировании кодовых слов.

Разведочный анализ данных подтвердил, что качество данных, к сожалению, не позволит в полной мере проверить, насколько хорошо в данной задаче могут быть использованы признаки В.М. Успенского. В сигналах много артефактов (которые отсутствуют в других датасетах с кардиограммами, найденными в интернете) – проиллюстрируем некоторые из них следующими рисунками.

На рис. 11.4 показан один из сигналов – виден резкий перепад амплитуды, который не устраняется с помощью фильтрации. Поскольку работа над задачей велась без тесного общения с постановщиком задачи, не понято, в чём причина подобных перепадов и как извлекать признаки из такого сигнала (из его первой части или второй).

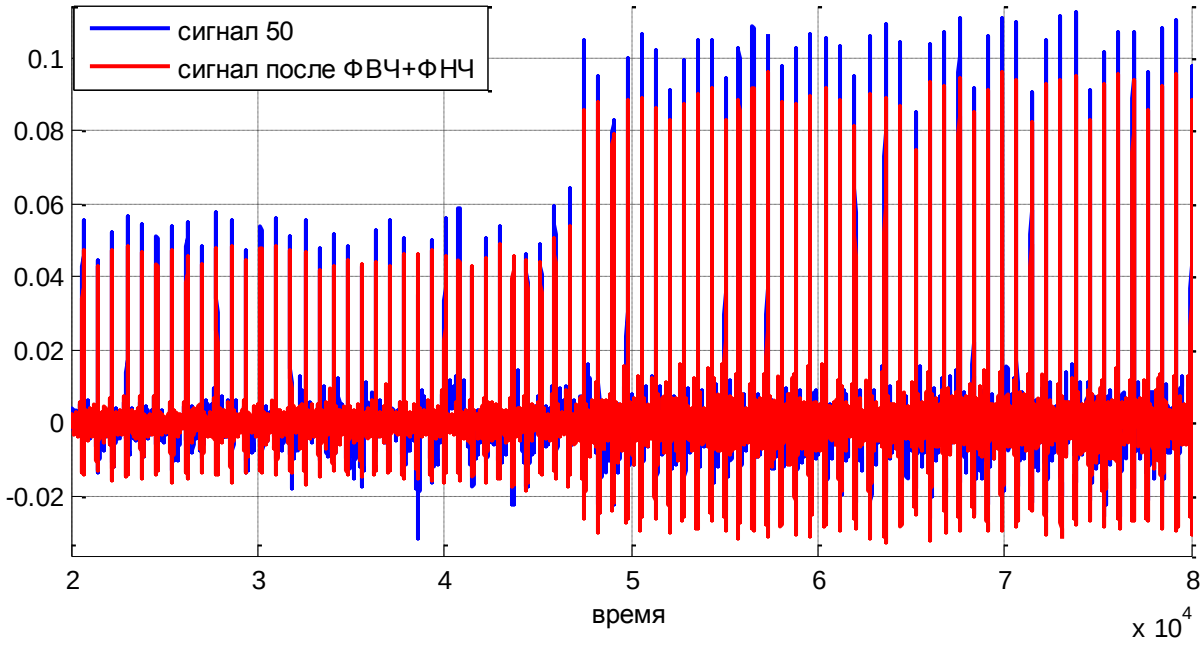



**Рис. 11.4. Сигнал ЛТ0_25-03-2015_20-12-13_300_261792289964_1000hz_int16_I_kwIRdqZ.wav**

Но хуже дела обстоят с сигналами, один из которых изображён на рис. 11.5. Как видно, фильтрация полностью испортила сигнал (хотя отлично работает на большинстве сигналов задачи). Но если увеличить изображение – рис. 11.6, то причина становится понятной: исходный сигнал не был даже похож на кардиограмму (по крайней мере, с точки зрения автора отчёта).

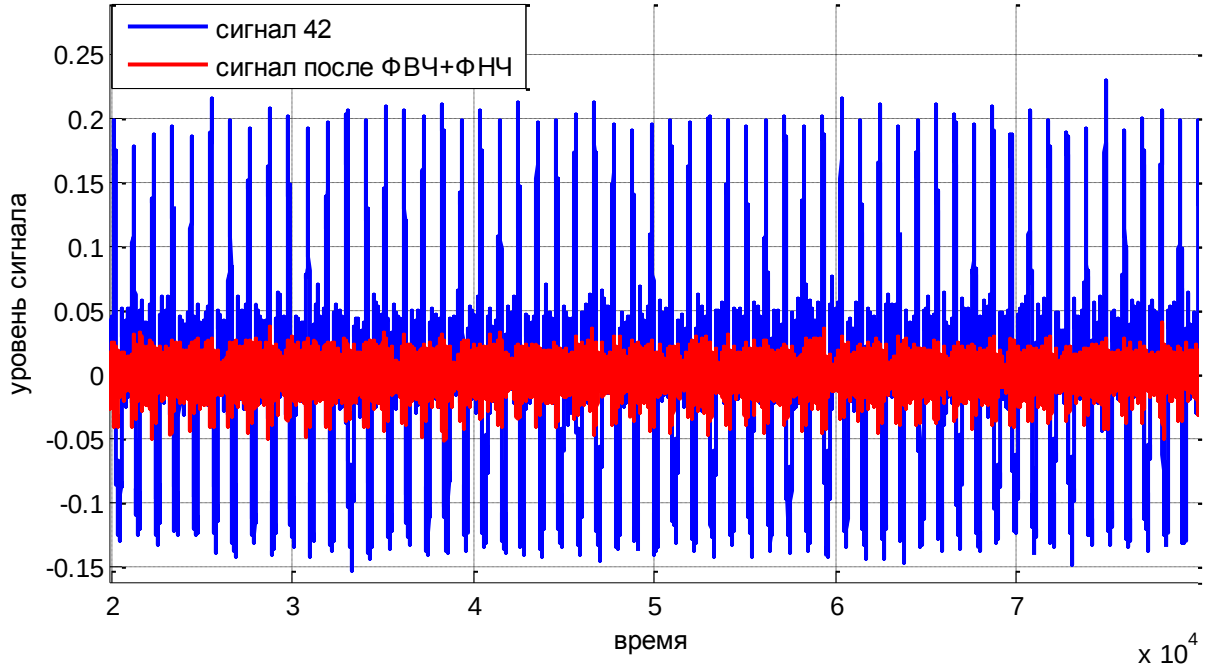



**Рис. 11.5. Сигнал КПЮ_01-10-2015_17-02-02_300_854384923152_1000hz_int16_I.wav**

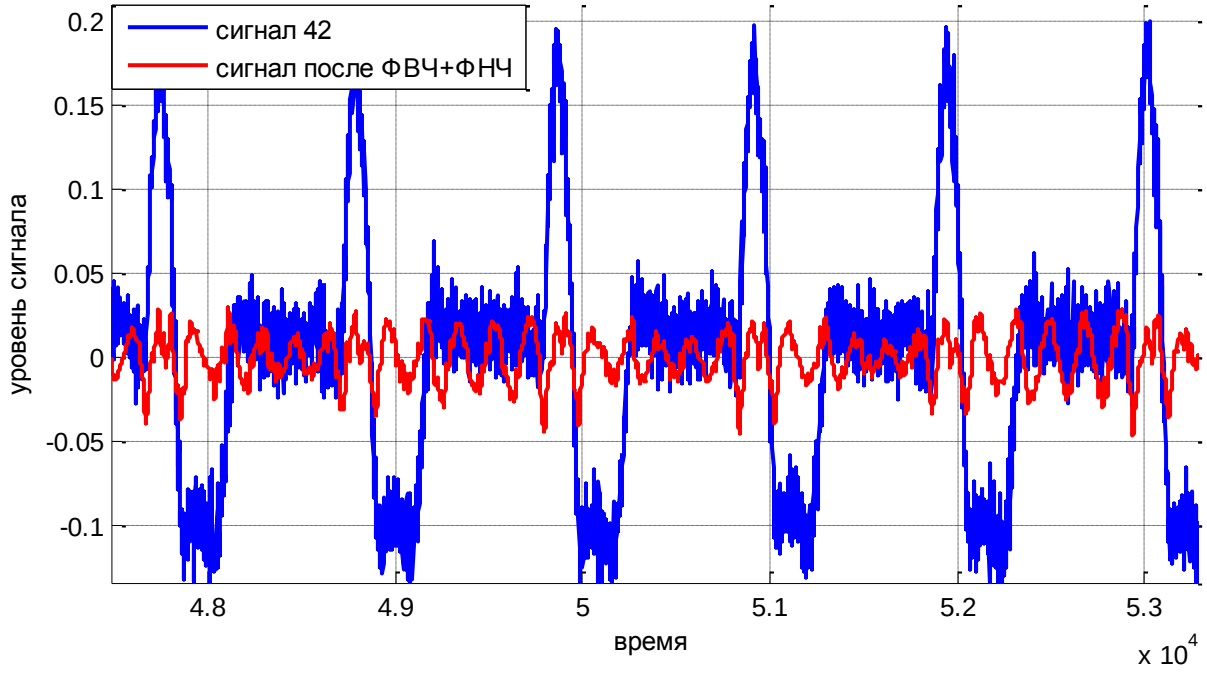



**Рис. 11.6. Сигнал с рис. 11.5 (увеличенный вид).**

## 12. Признаки, построенные с помощью вейвлетов

Вейвлеты[32] также являются мощным математическим аппаратом при анализе сигналов и часто хорошо применимы там, где плохо работает традиционное разложение Фурье. Сходу этим подходом получить качественные признаки не удалось, но на полный анализ причин и предложение вариантов их устранения, к сожалению, в режиме соревнования времени не хватило. Здесь не приводим статистику всех экспериментов, поскольку их было сделано немного.

## 13. Классификация сигналов

После того как построены все описанные выше признаки (признаковые матрицы записаны в файлы), строится классификатор из библиотеки scikit-learn[33] для языка Python. Эта библиотека выбрана, поскольку является самой лучшей по составу реализованных алгоритмов. Поэтому часть решения получена за пределами системы Matlab - в Ipython Notebook[34].

[32] https://ru.wikipedia.org/wiki/Вейвлет

[33] http://scikit-learn.org/stable/

[34] http://ipython.org/notebook.html

Реализованы (файл с кодом – **dj_cardio-makefinalsolution-checkpoint.ipynb**) следующие этапы:

1. Загрузка признаковых матриц

2. Их конкатенация (разные группы признаков у нас записаны в разных матрицах). В итоге получаем обучающую выборку и тестовую (на которой необходимо получить финальное решение).

3. Обучаем и запускаем регрессор sklearn.ensemble.GradientBoostingRegressor (градиентный бустинг над деревьями) с параметрами

n_estimators=1000,
learning_rate=0.01,
max_depth=2,
random_state=100,
loss='ls'

В результате получаем 250 вещественных чисел – ответы регрессора.

4. Записываем их в файл **myans_tmp.csv**

Затем в системе Matlab загружаем ответы регрессора и формируем окончательный файл-ответа **myfinans.csv**: для каждого сигнала из тестовой выборки указываем название файла с сигналом и его предполагаемое значение целевого признака:

1 – если ответ регрессора > 0.882

0 – иначе.

Хотя решается задача классификации с двумя классами (0 и 1) мы решаем её не классификатором (который также выдаёт ответы 0 или 1), а регрессором, который выдаёт числа из отрезка [0,1], которые можно интерпретировать как степень уверенности в том, что сигнал принадлежит классу 1. Использование регрессора вызвано желанием иметь ещё один параметр для настройки: порог (выше значения которого принимается решение о принадлежности к классу 1). Ясно, что при увеличении порога для хорошо настроенного регрессора чувствительность повышается, а специфичность уменьшается, см. также рис. 13.1. Поэтому варьированием порога можно «контролировать своё место в турнирной таблице» (выбирая нужные значения чувствительности и специфичности). Кроме того, при варьировании порога меняются значения исследуемых функционалов качества – FF-меры и Se+Sp, см. рис. 13.1.

Кроме градиентного бустинга над деревьями, который стал финальным алгоритмом, были исследованы следующие алгоритмы:

- sklearn.ensemble.RandomForestRegressor (случайный лес),
- sklearn.linear_model.SGDRegressor (линейная модель + стохастический градиентный спуск)
- sklearn.ensemble.ExtraTreesRegressor ("экстремально случайные деревья")
- sklearn.linear_model.LogisticRegression (логистическая регрессия)
- sklearn.linear_model.Ridge (гребневая регрессия)

Фактически, перебирались все регрессоры, реализованные в библиотеке scikit-learn.

В табл. 13.1 показано качество перечисленных алгоритмов с помощью контроля по одному (Leave-One-Out[35]). В качестве основных критериев качества выбраны FF-мера (среднее гармоническое чувствительности и специфичности), а также среднее арифметическое чувствительности и специфичности: (Se+Sp)/2. Выбор таких критериев связан с тем, что напрямую оптимизировать критерий качества, предложенный организаторами, невозможно (он зависит от расположения участников соревнования в турнирной таблице). Однако, организаторы вычисляют (Se+Sp) присланных решений – и сообщают это значение участникам[36] (т.е. можно контролировать, совпадают ли результаты локального тестирования с финальной проверкой), а FF-мера является примером «другого среднего» чувствительности и специфичности.

На рис. 13.1 приведены графики значений качества в зависимости от порога для самых интересных классификаторов и значений параметров. Был, естественно, исследован значительно больший набор различных параметров алгоритмов. Автор отчёта решил не перегружать отчёт лишней информацией. Красным цветом в таблице показано значение (Se+Sp)/2 соответствующего присланного организаторам решения (если такая посылка была).

Итак, анализируя статистику можно сделать следующие выводы. SGDRegressor не удаётся хорошо настроить на данные, у остальных линейных методов (LogisticRegression и Ridge) есть большое различие в качестве на локальном контроле и при тестировании организаторами (их решено также не использовать для решения задачи). Наиболее стабильны алгоритмы основаны на построении деревьев. При этом RandomForestRegressor явно лучше ExtraTreesRegressor. Бустинговый алгоритм GradientBoostingRegressor также показывает качество ниже на отосланных решениях, но это меняется при селекции признаков.

[35] Из выборки последовательно исключается по одному объекту, на оставшейся выборке настраивается классификатор, а затем его результат сравнивается с истинным целевым значением удалённого объекта. Точнее: формируем вектор ответов на обучающей выборке, *i*-й элемент вектора – ответ регрессора на *i*-м сигнале (регрессор обучен на остальных сигналах). Вектор ответов сравниваем с истинным целевым вектором.

[36] Среднее арифметическое (Se+Sp)/2 удобнее суммы тем, что как и FF-мера лежит в отрезке [0,1].

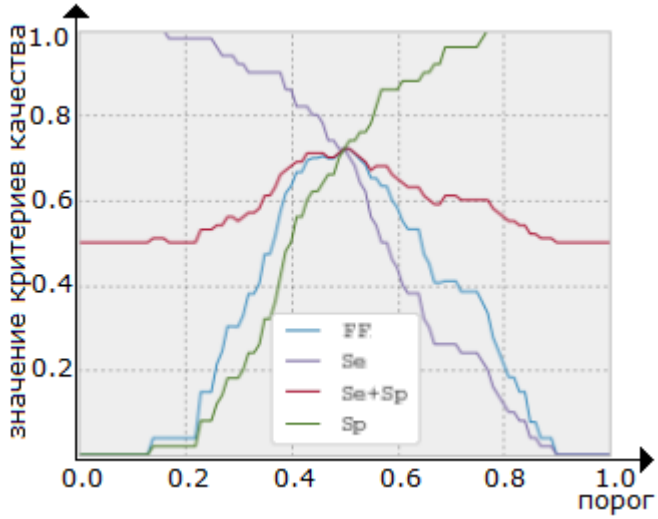
Табл. 13.1. Качество различных алгоритмов (FF-мера / (Se+Sp)/2)

| Алгоритм | Параметры алгоритма | | |
|---|---|---|---|
| **RandomForestRegressor (n_estimators=100, criterion='mse')** | **max_features=5** | **max_features=10 (*1)** | **max_features=15** |
| | 0.6949 / 0.7 | 0.72 / 0.72<br>**0.67** | 0.7/ 0.69 |
| **SGDRegressor()<br>С нормировкой<br>sk.preprocessing.normalize()** | **alpha=1.0** | **alpha=0.1 (*2)** | **alpha=0.01** |
| | 0.3632 / 0.52 | 0.4124 / 0.52 | 0.3962 / 0.52 |
| **ExtraTreesRegressor(criterion='mse')** | **n_estimators=20<br>max_depth=2 (*3)** | **n_estimators=40<br>max_depth=2** | **n_estimators=40<br>max_depth=1** |
| | 0.6627 / 0.69<br>0.6175 | 0.6314 / 0.69 | 0.5994 / 0.69 |
| **GradientBoostingRegressor (n_estimators=1000, learning_rate=0.01, max_depth=2, random_state=1, loss='ls')** | **n_estimators=1000, max_depth=2,** | **n_estimators=1000, max_depth=1** | **n_estimators=100, max_depth=1** |
| | 0.68 / 0.7 | 0.6928 /0.72<br>0.6048<br>(*4) | **0.7467 / 0.75**<br>0.5972 |
| **LogisticRegression()** | **C=1.0<br>(*5)** | **C=0.1** | **C=0.01** |
| | 0.6986 / 0.71<br>0.5467 | 0.6725 / 0.69 | 0.6794 / 0.69 |
| **Ridge(normalize=True)** | **alpha=0.1<br>(*6)** | **alpha=0.01** | **alpha=0.001** |
| | 0.6862 / 0.71<br>0.6316 | 0.6794 / 0.71 | 0.672 / 0.7 |

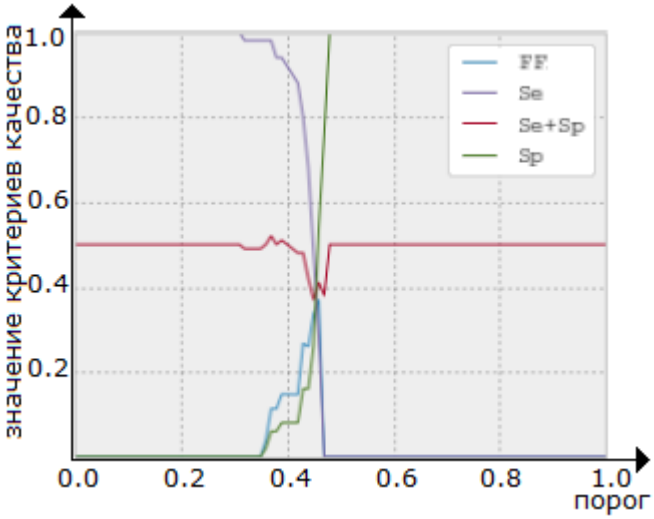
Обычно при построении современных алгоритмов классификации для повышения качества используют ансамбли алгоритмов[37]. Пример ансамбля – «блендинг» – линейная комбинация нескольких алгоритмов. Такой ансамбль хорошо вписывается в наш метод классификации: можно взять линейную комбинацию регрессоров и для неё подбирать порог. Эксперименты показали, что большого прироста качества такие ансамбли не дают, поэтому они не использовались в финальном решении.

[37] http://mlwave.com/kaggle-ensembling-guide/

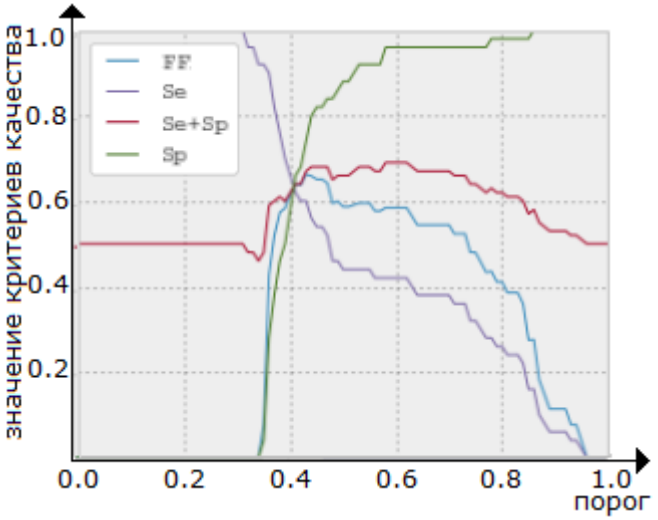
Рис. 13.1. Качество решений на локальном тесте при варьировании порога:



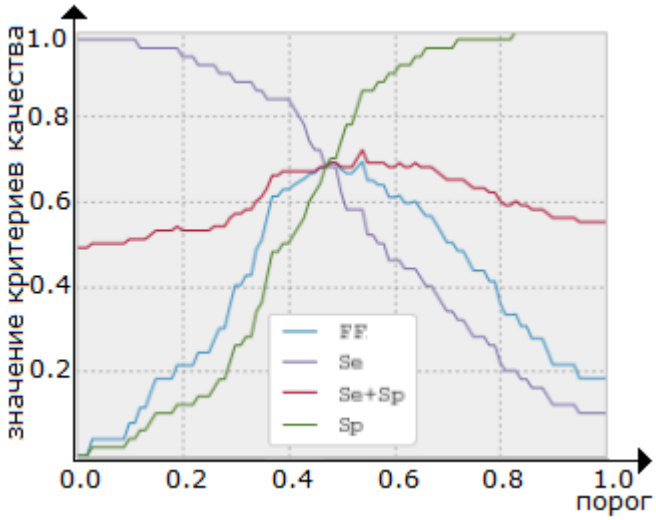

**Качество алгоритма (*1).**



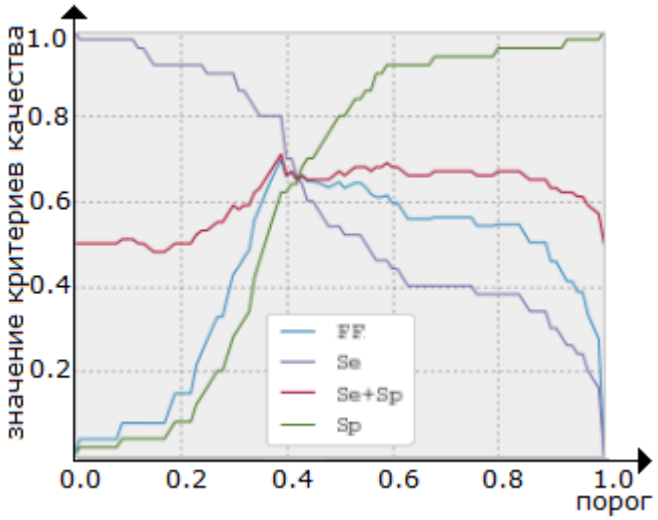

**Качество алгоритма (*2)**



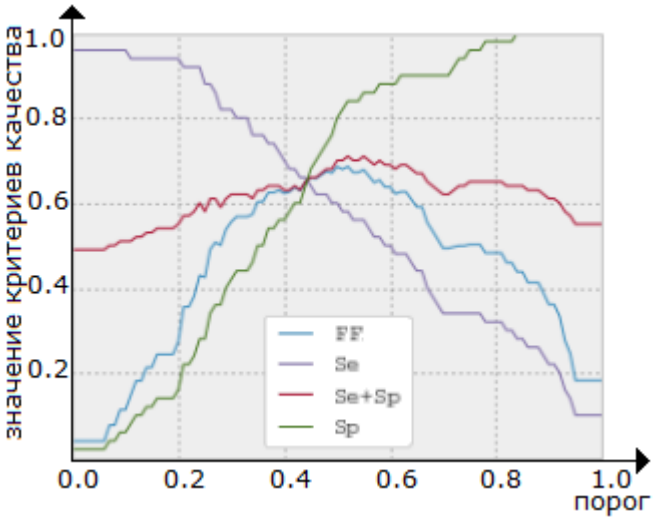

**Качество алгоритма (*3)**




**Качество алгоритма (*4)**




**Качество алгоритма (*5)**




**Качество алгоритма (*6)**

Последним этапом построения классификатора был окончательный отбор признаков. Теперь уже признаки исключались по одному и на локальном контроле вычислялось качество решения. Если удаление признака не сильно ухудшало качество, то он окончательно удалялся. Так был построена финальная версия классификатора. Как было сказано выше, результат лучшей загрузки в турнирной таблице был у градиентного бустинга над деревьями (GradientBoostingRegressor) – (Se+Sp)/2=**70.59**, при этом на локальном контроле его результаты чуть ниже: FF-мера=0.6699, (Se+Sp)/2=0.69.

## 14. Итоговое качество полученного результата

На рис. 14.1 показаны результаты участников из Top10 соревнования CardioQVARK[38]. Видно, что, не смотря на второе место автора отчёта, разработанный им алгоритм показал лучшее значение по среднему арифметическому чувствительности и специфичности (**0.7059**), FF-мере (**0.7047**) и по F1-мере(**0.5497**). Для сравнения, у победителя значение FF/F1-мер – 0.6864/0.5455, у участников из конца десятки сильнейших – доходит до 0.51/0.37 (т.е. фактически соответствует случайному решению).

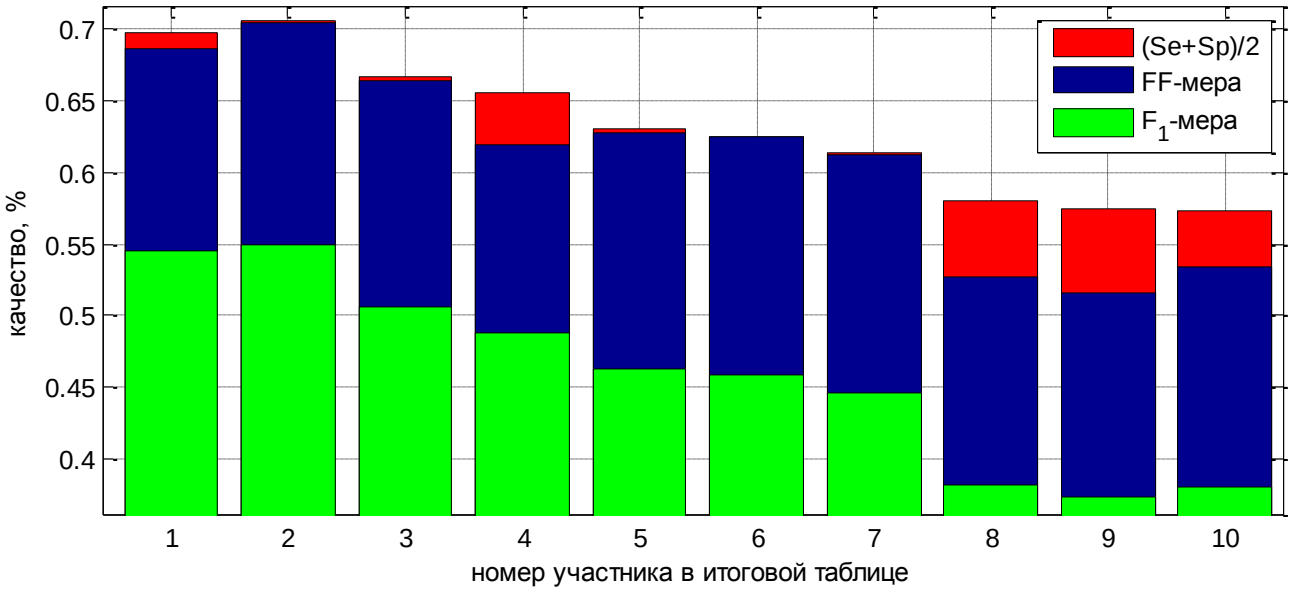



**Рис. 14.1. Качество решений участников CardioQVARK из Top10**

**Интересное наблюдение.** Автор в своих локальных тестах попробовал использовать случайную модель, чтобы оценить возможности тривиального решения (т.е. её решение – случайный бинарный вектор). Лучшее решение (из 100) имеет качество

[38] http://www.cardioqvark.ru/challenge/

Se+Sp = 1.1934 (см. рис. 14.2). В следующей таблице подкрашены результаты участников, которые превзошли данный порог. Из 72 участников только 15 смогли улучшить случайное решение!

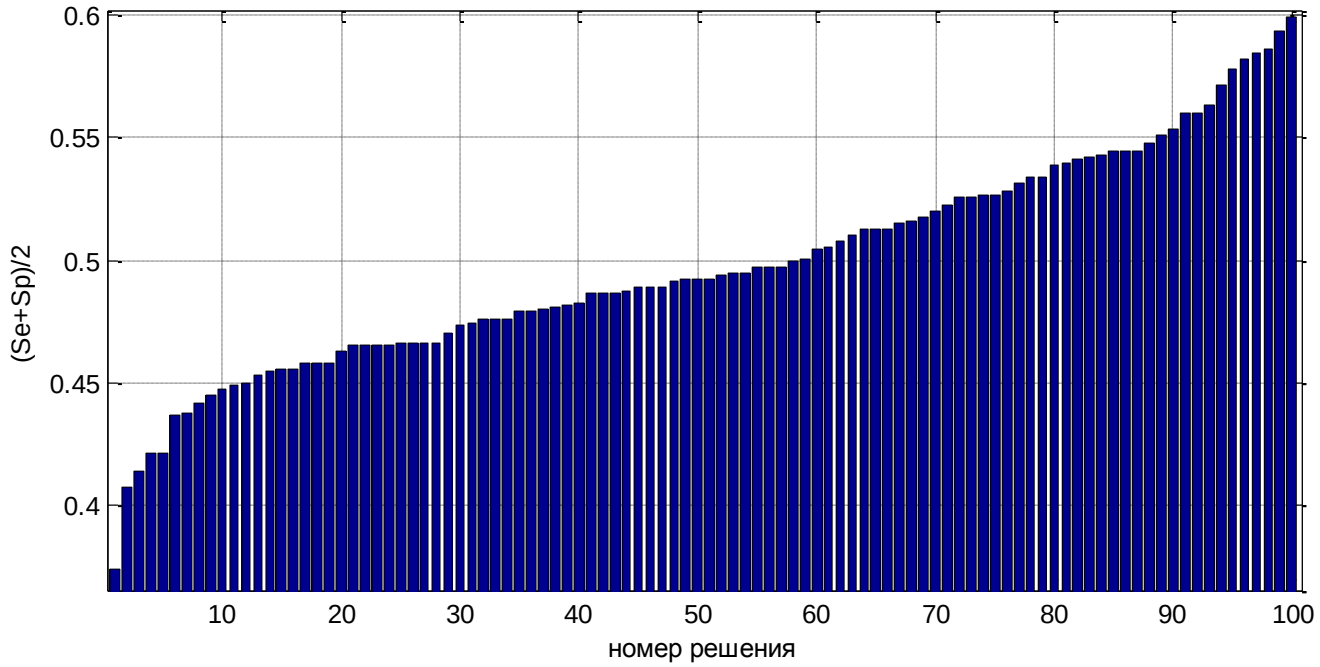



**Рис. 14.2. Качество случайных решений**

| Место | Логин | Чувствит. (Se) Лучшее решение | Результат Se | Спец. (Sp) Лучшее решение | Результат Sp | Сумма | Se + Sp |
|---|---|---|---|---|---|---|---|
| 1 | belavin | 60,94% | 15 | 78,57% | 3 | 18 | 139,51% |
| 2 | djakonov | 73,44% | 10 | 67,74% | 9 | 19 | 141,18% |
| 3 | IRV | 62,50% | 14 | 70,97% | 7 | 21 | 133,47% |
| 4 | ibryukhanov | 50,00% | 22 | 81,18% | 2 | 24 | 131,18% |
| 5 | alex_dok | 59,38% | 16 | 66,67% | 11 | 27 | 126,04% |
| 6 | eugtsa | 60,94% | 15 | 63,98% | 14 | 29 | 124,92% |
| 6 | potom20 | 57,81% | 17 | 65,05% | 12 | 29 | 122,87% |
| 7 | dubnov | 40,63% | 26 | 75,27% | 5 | 31 | 115,89% |
| 7 | AlexSemenov | 39,06% | 27 | 75,81% | 4 | 31 | 114,87% |
| 7 | leksotat | 42,19% | 25 | 72,58% | 6 | 31 | 114,77% |
| 7 | art-ya | 18,75% | 30 | 93,55% | 1 | 31 | 112,30% |
| 8 | andreyi | 46,88% | 24 | 68,82% | 8 | 32 | 115,69% |
| 9 | prikhodko | 51,56% | 21 | 65,05% | 12 | 33 | 116,62% |
| 10 | nikmort | 62,50% | 14 | 57,53% | 20 | 34 | 120,03% |
| 10 | LyapinNR | 60,94% | 15 | 58,06% | 19 | 34 | 119,00% |
| 10 | gavrikos | 51,56% | 21 | 64,52% | 13 | 34 | 116,08% |
| 11 | schiavoni | 67,19% | 12 | 55,91% | 23 | 35 | 123,10% |
| 11 | parametric | 67,19% | 12 | 55,91% | 23 | 35 | 123,10% |
| 11 | ilya25 | 59,38% | 16 | 58,60% | 19 | 35 | 117,98% |
| 11 | vlad.alt.ov | 54,69% | 19 | 61,29% | 16 | 35 | 115,98% |
| 11 | iv.bagaew | 53,13% | 20 | 61,83% | 15 | 35 | 114,96% |
| 12 | kirska | 84,38% | 4 | 46,24% | 32 | 36 | 130,62% |
| 12 | celyh | 70,31% | 11 | 52,69% | 25 | 36 | 123,00% |
| 12 | p2004r | 70,31% | 11 | 52,69% | 25 | 36 | 123,00% |
| 12 | MSM | 54,69% | 19 | 60,75% | 17 | 36 | 115,44% |
| 13 | virus | 75,00% | 9 | 50,54% | 28 | 37 | 125,54% |
| 14 | glivec | 53,13% | 20 | 59,14% | 18 | 38 | 112,27% |
| 15 | vartemkin | 53,13% | 20 | 58,06% | 19 | 39 | 111,19% |
| 15 | karpenko | 48,44% | 23 | 61,29% | 16 | 39 | 109,73% |
| 15 | n_sib | 35,94% | 29 | 67,20% | 10 | 39 | 103,14% |
| 16 | iamuser | 75,00% | 9 | 47,31% | 31 | 40 | 122,31% |
| 16 | panin | 62,50% | 14 | 52,15% | 26 | 40 | 114,65% |
| 17 | ident | 56,25% | 18 | 55,91% | 23 | 41 | 112,16% |
| 17 | ikaserg | 54,69% | 19 | 56,45% | 22 | 41 | 111,14% |
| 18 | a2bogdanov | 56,25% | 18 | 53,76% | 24 | 42 | 110,01% |
| 18 | mike | 51,56% | 21 | 56,99% | 21 | 42 | 108,55% |
| 19 | kobets | 56,25% | 18 | 52,69% | 25 | 43 | 108,94% |
| 19 | fedorov86 | 56,25% | 18 | 52,69% | 25 | 43 | 108,94% |
| 19 | nesterjuk-petr | 56,25% | 18 | 52,69% | 25 | 43 | 108,94% |
| 19 | DIChernov | 56,25% | 18 | 52,69% | 25 | 43 | 108,94% |
| 20 | dmaslennikov | 73,44% | 10 | 43,55% | 34 | 44 | 116,99% |
| 20 | king3 | 70,31% | 11 | 44,09% | 33 | 44 | 114,40% |
| 20 | KhomutovNikita | 59,38% | 16 | 50,54% | 28 | 44 | 109,91% |
| 21 | vitalikm | 76,56% | 8 | 37,63% | 38 | 46 | 114,19% |
| 21 | markochev | 48,44% | 23 | 55,91% | 23 | 46 | 104,35% |
| 22 | fedork | 65,63% | 13 | 43,55% | 34 | 47 | 109,18% |
| 22 | zhelyazik | 53,13% | 20 | 51,61% | 27 | 47 | 104,74% |
| 23 | akimov | 70,31% | 11 | 40,86% | 37 | 48 | 111,17% |
| 23 | e_vd0vina | 54,69% | 19 | 50,00% | 29 | 48 | 104,69% |
| 24 | chuprakov | 90,63% | 2 | 26,34% | 47 | 49 | 116,97% |
| 24 | vasenev | 82,81% | 5 | 27,96% | 44 | 49 | 110,77% |
| 24 | sensanek | 76,56% | 8 | 32,26% | 41 | 49 | 108,82% |
| 24 | alexeyk | 65,63% | 13 | 41,94% | 36 | 49 | 107,57% |
| 24 | victor | 51,56% | 21 | 50,54% | 28 | 49 | 102,10% |
| 25 | perepechkin | 54,69% | 19 | 47,31% | 31 | 50 | 102,00% |
| 26 | ggofat | 85,94% | 3 | 22,04% | 48 | 51 | 107,98% |
| 26 | aleksei.r | 73,44% | 10 | 32,26% | 41 | 51 | 105,70% |
| 26 | ozherelev.ilya | 59,38% | 16 | 42,47% | 35 | 51 | 101,85% |
| 27 | abramov | 92,19% | 1 | 19,35% | 51 | 52 | 111,54% |
| 27 | Victor | 75,00% | 9 | 30,11% | 43 | 52 | 105,11% |
| 27 | edloginova | 67,19% | 12 | 33,33% | 40 | 52 | 100,52% |
| 27 | stepandrapak | 50,00% | 22 | 49,46% | 30 | 52 | 99,46% |
| 28 | maximkochmin | 85,94% | 3 | 19,89% | 50 | 53 | 105,83% |
| 28 | daniil | 70,31% | 11 | 30,65% | 42 | 53 | 100,96% |
| 28 | lar10 | 92,19% | 1 | 5,91% | 52 | 53 | 98,10% |
| 29 | volodia2 | 60,94% | 15 | 36,02% | 39 | 54 | 96,96% |
| 30 | maxxk | 84,38% | 4 | 19,35% | 51 | 55 | 103,73% |
| 30 | KATOK | 75,00% | 9 | 26,88% | 46 | 55 | 101,88% |
| 30 | maxim.b | 79,69% | 7 | 22,04% | 48 | 55 | 101,73% |
| 30 | daryas | 81,25% | 6 | 20,43% | 49 | 55 | 101,68% |
| 31 | vla | 70,31% | 11 | 27,42% | 45 | 56 | 97,73% |
| 32 | irbiscat | 37,50% | 28 | 43,55% | 34 | 62 | 81,05% |

## 15. Замечания и соображения по конкурсу

1. Автор считает, что критерий качества, выбранный организаторами для оценки участников совревнования, не способствовал получению действительно качественного решения и не мог, например, оптимизироваться на локальном контроле (т.к. зависел от результатов других участников). Интересно, что на протяжении последних нескольких недель конкурса бóльшую часть времени в турнирной таблице лидировали решения, которые были далеко не первыми по всем классическим функционалам качества. В качестве функционала качества в этом конкурсе лучше было бы использовать F1-меру или какое-нибудь[39] среднее чувствительности и специфичности. Тем не менее, заказчик всегда прав – и автору пришлось ориентироваться именно на этот функционал.

2. Некоторые сигналы оказались «перевёрнутыми кардиограммами», другие – не были похожи на кардиограммы вообще или имели большие артефакты (например, скачки). Понятно, что реальные данные именно такие, но они сильно отличаются от многих датасетов, найденных в интернете. Также отметим, что большинство найденных готовых библиотек программ совсем не годились для работы с данными сигналами.

3. Размеры выборок (контрольной и обучающей) всё-таки очень малы. По своим статистическим свойствам контрольная выборка отличалась от обучающей. Это иллюстрировалось, например, качеством работы простых моделей на признаках, предоставленных организаторами. На обучении, даже при честном скользящем конроле, качество было очень высоким, на тестовой выборке – низким. Возможно, так получилось из-за малого размера выборок, или из-за разного процентного соотношения классов в обучении и контроле, или из-за разных способов формирования обучения и контроля (у участников нет информации, является ли контрольная выборка случайной подвыборкой из всех данных организаторов).

[39] арифметическое, геометрическое и т.п.

## 16. Выводы

Был разработан алгоритм, который по распространённым критериям качества (Se+Sp, F-мера и т.п.) превосходит решения других участников. В итоговой турнирной таблице алгоритм занял второе место.

Алгоритм создан с помощью предложенного подхода к решению задачи:

- предобработка сигнала, выделение кардиоциклов
- генерация признаков (с помощью разных методов)
- селекция признаков
- настройка регрессора
- выбор порога и формирование окончательного результата

На локальных тестах (на обучающей выборке) признаки, построенные на основе Фурье-анализа, сингулярного разложения, статистик и стандартные признаки (предоставленные организаторами) оказались достаточно неплохими: используя только одну такую группу признаков можно было получить решение с качеством не ниже 0.64 по FF-мере. Признаки по В.М. Успенскому давали более низкое качество классификации: (ниже 0.6). Использование разнородных признаков существенно повышает качество решения (до 0.747).

Наиболее стабильные алгоритмы классификации (у которых результаты на локальном контроле согласовывались с контролем на тестовой выборке) основаны на построении деревьев. Линейные модели (гребневая, линейная и логистическая регрессии), в целом, наиболее нестабильны. Бустинг над деревьями и случаные леса по-разному ведут себя на разных признаковых пространствах (иногда лучше бустинг, иногда леса).

Разработанный автором подход вполне применим для построения действующего алгоритма (с качеством классификации выше 0.7 по FF-мере), однако для его практической реализации необходима выборка бóльшего объёма и, возможно, качества. Ни один из рассмотренных подходов не позволил (даже на локальных тестах) получить качество больше 0.75 по FF-мере. По мнению автора, в подобной постановке задачи на предоставленных данных такое качество недостижимо.

## Благодарности

Спасибо организаторам за необычное соревнование (с такой практически полезной задачей, связанной с кардиограммами, автор сталкивается впервые).

Москва, 14.03.2016 / Дьяконов А.Г.