Московский государственный университет имени М. В. Ломоносова
Факультет вычислительной математики и кибернетики
Кафедра математических методов прогнозирования

Генерация признаков в задаче классификации сигналов

Дипломная работа

Подготовила:
студентка кафедры ММП 517 группы
Власова Юлия Валерьевна

Научный руководитель:
доцент кафедры ММП, к.ф.-м.н.
Дьяконов Александр Геннадьевич

Москва
2009

Содержание

Введение

Во многих научно–технических областях в основе решения прикладных задач лежит классификация сигналов (конечных последовательностей измерений некоторой величины). В связи с этим актуальной является проблема разработки эффективных методов классификации сигналов. Один из возможных подходов к ее решению состоит в синтезе информативного признакового описания сигналов и сведении проблемы к задаче классификации в признаковом пространстве.

Для многих типов сигналов признаки базируются на временных и спектральных характеристиках [13, 14, 15, 20, 24, 25]. Однако использование подобных методов не всегда гарантирует высокое качество классификации.

В данной дипломной работе предлагается воспользоваться аппаратом генетической оптимизации [1, 2, 3, 4] для построения качественного признакового описания сигнала на основе выделения из него наиболее значимой информации [7]. *Признаком* для сигнала в этом случае называется вещественная функция от ряда. Основная идея метода заключается в представлении признака в виде суперпозиции последовательно применяемых операторов обработки сигнала. Вводится два типа операторов. Операторы первого типа по описанию сигнала получают ряд, возможно, другой длины. Операторы второго типа преобразуют сигнал в число, вычисляя его некоторую характеристику. Любой признак определяется набором операторов первого типа и одним оператором второго типа. Поиск качественного признакового пространства (определение операторов признака) осуществляется посредством генетической оптимизации. Качество отдельного признака оценивается критерием AUC–ROC [5], а совокупности признаков — методом *скользящего контроля с исключением объектов по одному* по алгоритму *k ближайших соседей* [6]. При использовании алгоритма k ближайших соседей осуществляется подбор оптимальной взвешенной евклидовой метрики.

Универсальность предлагаемого подхода заключается в возможности адаптировать систему под любую заданную предметную область, она не является привязанной к какой-либо конкретной прикладной задаче. Для этого необходимо сформировать богатую библиотеку операторов обработки сигналов на основе экспертных знаний о природе исходных данных. В разделах 5.1 и 5.2 продемонстрирована работа метода при решении реальных задач медицинской [10] и технической [26] диагностики. В обоих случаях получено приемлемое качество результата.

Основной целью дипломной работы является разработка, программная реализация и экспериментальное исследование нового метода классификации сигналов.

В первой главе приведена математическая постановка решаемой задачи.

Вторая глава посвящена описанию основных положений теории генетических алгоритмов.

В третьей главе представлено краткое введение в теорию ROC–анализа.

В главе 4 подробно описан собственно предлагаемый метод классификации сигналов.

В главе 5 продемонстрировано использование метода на реальных задачах классификации сигналов: Brain-Computer Interface [10] и Ford Classification Challenge [26].

Изложенный в работе метод программно реализован в системе MATLAB [27].

Работа проводилась в рамках выполнения проекта РФФИ № 08-07-00305-а. По результатам работы сделан доклад на XVI международной научной конференции “Ломоносов — 2009” [9].

1 Постановка задачи классификации сигналов

Проблема разработки качественных методов классификации сигналов является актуальной вследствие огромного числа приложений. Задача классификации сигналов встречается во многих научно–технических областях, таких как: техническая [26] и медицинская [10] диагностика, мониторинг сейсмически активных регионов, распознавание речевых сигналов при идентификации диктора, а также при решении других прикладных задач.

В данном разделе приведена математическая постановка задачи.

Математическая постановка задачи

Будем предполагать, что длины всех сигналов совпадают и равны T, хотя изложенные в работе методы могут применяться и в задачах с сигналами различной длины.

Имеется пространство объектов $\mathcal{X}$ вида $\mathbf{x} = (x_1, ..., x_T) \in \mathbb{R}^T$ и двухэлементное множество имён классов $Y = \{-1, +1\}$. Существует целевая зависимость $y^* : \mathcal{X} \to Y$, значения которой известны только на объектах обучающей выборки $X^L = (\mathbf{x}^i, y^i)_{i=1}^L$, $\mathbf{x}^i \in \mathcal{X} \subset \mathbb{R}^T, y^i = y^*(\mathbf{x}^i)$ (L — количество объектов обучения). Требуется построить алгоритм классификации $a : \mathcal{X} \to Y$, аппроксимирующий целевую зависимость $y^*(\mathbf{x})$ на всём множестве $\mathcal{X}$.

Основные обозначения:

- T — длина сигналов,
- $\mathcal{X}$ — пространство объектов,
- $X^L = \{\mathbf{x}^i = (x_1^i, ..., x_T^i)\}_{i=1}^L$ — обучающая выборка,
- L — объем обучающей выборки,
- $Y = \{-1, +1\}$ — имена классов.

2 Краткий обзор генетических алгоритмов

Эволюция в природе демонстрирует человеку механизм развития и приспособления живых организмов к окружающей среде. Природой заложено так, что в каждый момент времени любая популяция стремится к своему наилучшему варианту развития.

Это свойство привлекло внимание ученых. Исследователи в области компьютерных технологий обратились к природе в поисках новых алгоритмов. Методы, базирующиеся на идее эволюции Дарвина, были названы *эволюционными алгоритмами*. В данной теории выделяют следующие направления:

- генетические алгоритмы (ГА),
- эволюционные стратегии,
- генетическое программирование,
- эволюционное программирование.

Ниже в работе рассматриваются *генетические алгоритмы* (genetic algorithm), именно они используются для поиска оптимального решения — построения качественного признакового пространства для задачи классификации сигналов (см. главу 4).

В данном разделе приведено описание общей схемы функционирования генетического алгоритма. Конкретизация компонент алгоритма для задачи классификации сигналов рассматривается в главе 4.

2.1 Природный механизм

Рассмотрим, как происходит эволюция в природе. Любое живое существо характеризуется своим набором внешних параметров, называемым *фенотипом*. Некоторые из параметров благоприятствуют процветанию особи, другие же вредны или бесполезны с точки зрения выживания и размножения. Данные особи кодируются ее цепью ДНК (*генотипом*). Отдельные участки этой цепи – *гены* – определяют различные характеристики особи.

Особи популяции конкурируют между собой за ресурсы и за привлечение брачного партнера. Те организмы, которые наиболее приспособлены к окружающим условиям, проживут дольше и создадут более сильное и многочисленное потомство. Скрещиваясь, родители передают потомкам часть своего генотипа. Если ребенок совместит в себе части цепи, отвечающие за наиболее удачные качества, то он окажется еще более приспособленным.

Особи, не обладающие характеристиками, способствующими их выживанию, с большей вероятностью не проживут долго и не смогут создать потомство. Чем слабее особь, тем сложнее ей найти партнера, и, таким образом, с большой вероятностью генотип неприспособленных особей исчезнет из генофонда популяции.

Изредка происходит мутация — случайное изменение одной или нескольких позиций в хромосоме. Если полученная цепь будет использоваться для создания потомства, то у детей могут появиться совершенно новые качества. Мутация поддерживает разнообразие особей. Она предлагает новые варианты развития, если предложенные качества неудачны — на этапе отбора они отсеятся.

Естественный отбор, скрещивание и мутация обеспечивают развитие популяции. Поскольку выживают сильнейшие, и именно они участвуют в продолжении рода, то каждое новое поколение в среднем более приспособлено, чем предыдущее.

2.2 Генетические алгоритмы

В [3] генетический алгоритм описан на примере решения оптимизационной задачи. Моделируется эволюционное развитие искусственной популяции: каждая особь — одно из решений поставленной задачи.

Случайным образом создается начальная популяция. При этом ее размер фиксируется, количество особей не изменяется в течении всей работы алгоритма. Далее необходимо формализовать понятие приспособленности особи. Более приспособленные особи соответствуют более подходящим решениям задачи.

Шаг алгоритма [3] состоит из трех стадий:

- **Генерация промежуточной популяции путем отбора.**
  На данном этапе производится отбор оптимальной популяции для дальнейшего размножения. Промежуточную популяцию образуют наиболее приспособленные члены

Рис. 1: Общая схема функционирования генетического алгоритма

популяции. В некоторых случаях имеет смысл также отбрасывать особей с одинаковым набором генов.

- **Скрещивание особей промежуточной популяции, что приводит к формированию нового поколения.**
  Результатом скрещивания двух особей промежуточной популяции являются два потомка с компонентами, определенным образом заимствованными от родителей. Цель оператора скрещивания — распространение хороших генов по популяции.

- **Мутация полученного поколения.**
  Оператор мутаций меняет заданное число генов особей популяции на другие произвольные. С одной стороны, оператор мутации вытягивает из локальных экстремумов, с другой — приносит новую информацию в популяцию.

Существуют различные стратегии отбора, скрещивания и мутации особей [1, 2, 3, 4]. В настоящей работе используется следующий алгоритм. На каждом шаге алгоритма проверяется критерий останова: если условия выполнены, работа алгоритма завершается, иначе производится переход на новую эпоху и повторяется последовательность генетических операторов (отбор, скрещивание, мутация). В качестве критерия останова выбирается ограничение на число циклов алгоритма или стабилизация параметра, определяющего сходимость алгоритма, как правило, путем сравнения приспособленности популяций на нескольких его итерациях. Общая схема работы генетического алгоритма приведена на рис. 1.

Чтобы воспользоваться генетическим алгоритмом, необходимо представить каждый объект набором генов, т. е. правильно закодировать особи, и затем задать функцию приспособленности особи. Все дальнейшее функционирование алгоритма производится на уровне генотипа автоматически, что позволяет не задумываться о характере объекта. Это способствует возможности применения генетических алгоритмов в совершенно различных задачах.

В разделе 4 будет продемонстрирован пример применения генетических алгоритмов для задачи классификации рядов, в частности, для построения качественных признаков, позволяющих классифицировать сигналы.

3 Введение в теорию ROC–анализа

При построении одномерного признакового пространства с использованием генетических алгоритмов качество одного признака оценивается по критерию AUC–ROC (см. раздел 4.2.1). В данной главе изложена краткая теория ROC–анализа [5].

3.1 ROC–анализ

ROC–кривая (Receiver Operator Characteristic) наиболее часто используется для представления результатов бинарной классификации (задача с двумя классами) в машинном обучении. Название пришло из систем обработки сигналов. Поскольку различных классов всего два, то один из них назовем классом с положительными исходами, второй — с отрицательными исходами. ROC–кривая показывает зависимость количества верно классифицированных положительных примеров от числа неверно классифицированных отрицательных объектов. В терминологии ROC–анализа первые называются *истинно положительным*, вторые — *ложно отрицательным* множеством. Что является положительным событием, а что отрицательным, зависит от конкретной задачи. Например, если мы прогнозируем вероятность наличия заболевания, то положительным исходом будет класс “Больной пациент”, а отрицательным — “Здоровый пациент”.

Рассмотрим ошибки двух типов. *Ошибка первого рода*: положительные примеры, классифицированные как отрицательные. *Ошибка второго рода*: ложно положительные случаи. На основе результатов классификации моделью и фактической (реальной) принадлежностью примеров к классам строим таблицу:

Модель	Фактически	
	положительно	отрицательно
положительно	TP	FP
отрицательно	FN	TN

Здесь:

- TP (*True Positives*) — верно классифицированные положительные примеры,
- TN (*True Negatives*) — верно классифицированные отрицательные примеры,
- FN (*False Negatives*) — ложно отрицательные случаи (ошибка I рода),
- FP (*False Positives*) — ложно положительные случаи (ошибка II рода).

При анализе чаще оперируют не абсолютными показателями, а относительными — долями, выраженными в процентах. Введем соответствующие величины:
Доля истинно положительных случаев (True Positives Rate): $TPR = \frac{TP}{TP+FN} \cdot 100\%$
Доля ложно положительных примеров (False Positives Rate): $FPR = \frac{FP}{TN+FP} \cdot 100\%$.

Объективная ценность любого бинарного классификатора определяется следующими двумя показателями:
Чувствительность (Sensitivity) — это и есть доля истинно положительных случаев:

$$Se = TPR = \frac{TP}{TP + FN} \cdot 100\%,$$

Специфичность (Specificity) – доля истинно отрицательных случаев, которые были правильно идентифицированы моделью:

$$Sp = \frac{TN}{TN + FP} \cdot 100\%.$$

(Заметим, что $FPR = 100 - Sp$.)

Модель с высокой чувствительностью часто дает истинный результат при наличии положительного исхода (обнаруживает положительные примеры). Наоборот, модель с высокой специфичностью чаще дает истинный результат при наличии отрицательного исхода (обнаруживает отрицательные примеры). В терминах медицины — задачи диагностики заболевания, где модель классификации пациентов на больных и здоровых называется диагностическим тестом:

- Чувствительный диагностический тест появляется в гипердиагностике — максимальном предотвращении пропуска больных.
- Специфичный диагностический тест диагностирует только доподлинно больных. Это важно в случае, когда, например, лечение больного связано с серьезными побочными эффектами и гипердиагностика не желательна.

3.2 Построение ROC–кривой

Предполагается, что у классификатора имеется некоторый параметр, варьируя который, мы будем получать то или иное разбиение на два класса. Этот параметр часто называют *порогом*, или *точкой отсечения* (cut–off value). В зависимости от него будут получаться различные величины ошибок первого и второго рода.

ROC–кривая строится следующим образом:

1. Для каждого значения порога отсечения, которое меняется от 0 до 1 с шагом dx (например, 0.01) расчитываются значения чувствительности Se и специфичности Sp. В качестве альтернативы порогом может являться каждое последующее значение примера в выборке.
2. Строится график зависимости: по оси y откладывается чувствительность Se, по оси x — доля ложно положительных примеров $FPR = 100\% - Sp$.

В статье [5] приведен канонический алгоритм построения ROC–кривой.

На рис. 2 показаны примеры ROC–кривых, соответствующих двум различным классификаторам: A и B.

График часто дополняют прямой $y = x$.

ROC–кривая отражает зависимость между долей истинно положительных примеров (чувствительность) и долей ложно положительных объектов (100-специфичность). Анализируя эту кривую, можно сделать вывод о качестве классификатора (модели), и, кроме того, выбрать оптимальный порог отсечения.

Для идеального классификатора график ROC-кривой проходит через верхний левый угол, где доля истинно положительных случаев составляет 100% (идеальная чувствительность), а доля ложно положительных примеров равна нулю. Поэтому, чем ближе кривая к верхнему углу, тем выше предсказательная способность модели. Диагональная линия соответствует “бесполезному классификатору”, т. е. полной неразличимости двух классов.

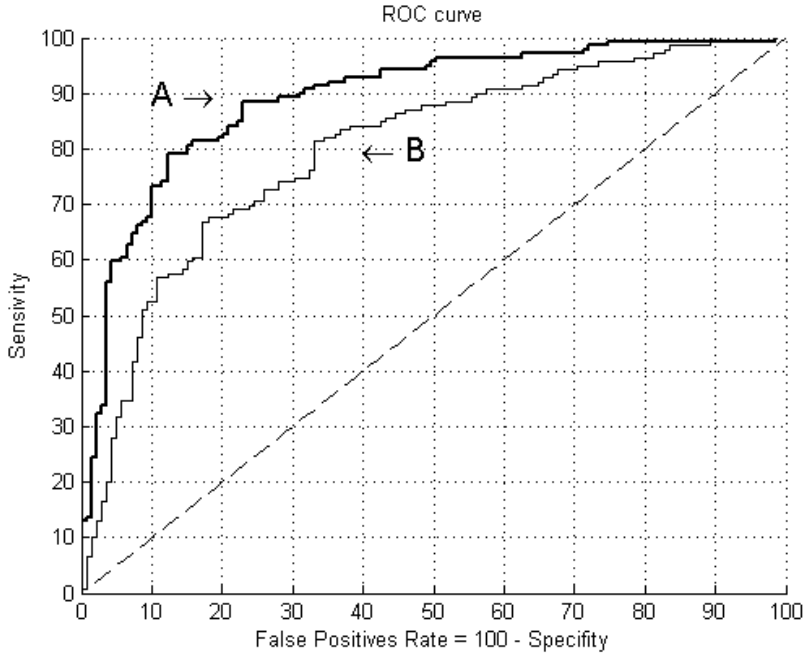

Рис. 2: ROC–кривые, соответствующие двум различным классификаторам: A и B

3.3 Показатель AUC

При визуальной оценке, расположение ROC-кривых друг относительно друга указывает на их сравнительную эффективность. Кривая, расположенная выше и левее, свидетельствует о большей предсказательной способности модели. Так, на рис. 2 представлены ROC-кривые двух моделей. Видно, что классификатор “A” лучше.

Однако визуальное сравнение не всегда позволяет выявить наиболее эффективную модель. Одним из методов сравнения ROC-кривых является оценка площади под кривыми, численный показатель которой называется AUC (Area Under Curve). Теоретически он меняется от 0 до 1.0, но поскольку классификатор всегда характеризуется кривой, расположенной выше положительной диагонали, то обычно говорят об изменениях от 0.5 (“бесполезный” классификатор) до 1.0 (“идеальная” модель).

Вычислить показатель AUC можно, например, с помощью численного метода трапеций:

$$AUC = \int y(x)\,dx = \sum_i \left(\frac{x_{i+1} + x_i}{2}\right) \cdot (y_{i+1} - y_i)$$

С большими допущениями можно считать, что чем больше значение AUC, тем лучшей прогностической способностью обладает модель. Однако следует знать:

- AUC скорее предназначен для сравнения нескольких моделей;
- AUC не содержит никакой информации о чувствительности и специфичности модели.

В [5] приводится следующая экспертная шкала для значений AUC, по которой можно судить о качестве классификатора:

AUC	Качество классификации
0.9-1.0	Отличное
0.8-0.9	Очень хорошее
0.7-0.8	Хорошее
0.6-0.7	Среднее
0.5-0.6	Неудовлетворительное

Идеальная модель обладает 100% чувствительностью и специфичностью. Однако при решении реальных задач добиться этого практически невозможно, более того, невозможно одновременно повысить и чувствительность, и специфичность модели. Компромисс находится с помощью порога отсечения, так как пороговое значение влияет на соотношение Se и Sp. Порог отсечения нужен для того, чтобы применять модель на практике: относить новые примеры к одному из двух классов.

В данной работе критерий AUC используется для оценки качества одного признака (см. пункт 4.2.1).

4 Применение генетического алгоритма для поиска качественных признаков

В данной работе проблема классификации сигналов (см. главу 1) сведена к задаче классификации в сгенерированном признаковом пространстве. Задача построения признакового описания сигнала решается с помощью генетического подхода [7]. В данном разделе конкретизируются компоненты генетического алгоритма.

Отметим, что предложенный метод был реализован в системе MATLAB [27]. Ниже приводится программный код.

4.1 Описание особи популяции

В контексте рассматриваемой задачи (см. главу 1) признаком для сигнала называется вещественная функция от сигнала, ставящая в соответствие ряду определенное число: $f(\mathbf{x}) = f(x_1, ..., x_T) \in \mathbb{R}$. Данную функцию представляем в виде суперпозиции операторов обработки сигнала. Любой признак определяется набором операторов его составляющих. Поиск оптимального набора осуществляется посредством генетического алгоритма.

Введем два типа операторов обработки сигналов и формально опишем особь популяции.

Операторы первого типа

Операторы первого типа по описанию сигнала $\mathbf{x} = (x_1, x_2, ..., x_s)$ получают ряд $\mathbf{y} = (y_1, y_2, ..., y_p)$, возможно, другой длины. Основное их назначение – преобразовать информацию (улучшить или перейти к иному описанию). Заметим, операторы первого типа должны быть достаточно простыми.

Экспертами формируется библиотека операторов первого типа. При решении прикладных задач Brain-Computer Interface [10] и Ford Classification Challenge [26] были использованы операторы, записанные в табл. 1.

Таблица 1: Множество $\mathfrak{B}$ операторов первого типа, реализованных в программной среде MATLAB

№	Описание оператора
B1	фильтрация сигнала
B2	сглаживание сигнала
B3	детрендитизация
B4	взятие производной (конечной разности)
B5	оператор взятия абсолютного значения
B6	покомпонентное логарифмирование ряда
B7	покомпонентное возведение в степень ряда
B8	сортировка значений ряда
B9	“стандартизация” значений ряда
B10	выделение подряда

Обозначим обрабатываемый сигнал через `Signal`, длину сигнала через `len`. Ниже приведен программный код операторов в системе MATLAB.

1. *Фильтрация сигнала — уменьшение уровня шума*

   Параметр: `windowSize` — ширина окна фильтрации

   ```
   Signal = filter(ones(1,windowSize)/windowSize,1,Signal);
   ```

2. *Сглаживание сигнала — замена значения в точке средним значением в окрестности*

   Параметр: `windowSize` — ширина окна сглаживания

   ```
   for i = 1:len/windowSize
     Signal_new(i,:) = mean(Signal((i-1)*windowSize+1 : i*windowSize));
   end
   Signal = Signal_new;
   ```

3. *Детрендитизация*

   Параметр: `windowSize` — ширина окна фильтрации

   ```
   Signal_new = filter(ones(1,windowSize)/windowSize,1,Signal);
   Signal = Signal - Signal_new;
   ```

4. *Взятие производной (конечной разности)*

   Производная конечной разности первой степени вычисляется согласно формуле:

   $$\mathrm{diff}(x_1, x_2, ..., x_s) = (x_2 - x_1, x_3 - x_2, ..., x_s - x_{s-1}).$$

   Применение k раз подряд данного оператора эквивалентно записи на языке MatLab: `Signal = diff(Signal, k);`

   Параметр: `k` — степень производной

   ```
   Signal = diff(Signal, k);
   ```

5. *Оператор взятия абсолютного значения*

   $$\mathrm{abs}(x_1, x_2, ..., x_s) = (|x_1|, |x_2|, ..., |x_s|)$$

   На языке MatLab оператор имеет вид:

   ```
   Signal = abs(Signal);
   ```

6. *Покомпонентное логарифмирование ряда*

```
Signal = (log(Signal + abs(min(Signal)) +0.01));
```

7. *Покомпонентное возведение в степень ряда*

Параметр: `k` — степень
`Signal = Signal.^k;`

8. *Сортировка значений ряда*

```
Signal = sort(Signal);
```

9. *"Стандартизация" значений ряда*

```
Signal = (Signal - repmat(mean(Signal),len,1))./
        repmat(max(abs((Signal))),len,1);
```

10. *Выделение подряда*

Параметры: `a, b` — границы подряда
`Signal = Signal(a:b);`

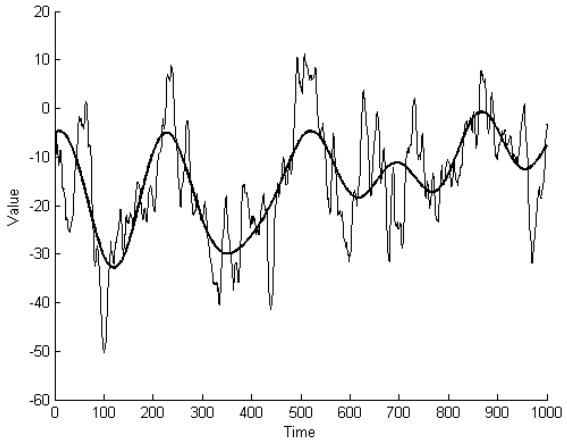

Рис. 3: Фильтрация сигнала

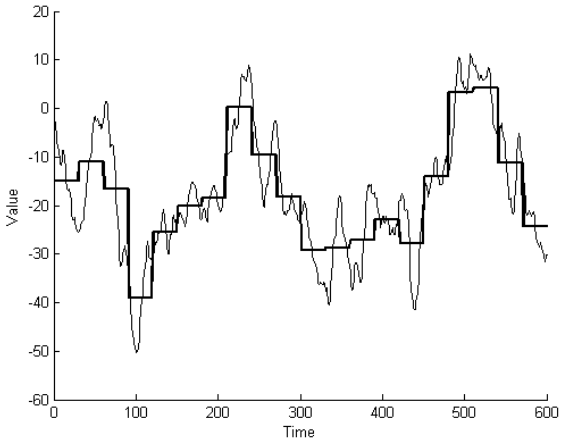

Рис. 4: Сглаживание сигнала

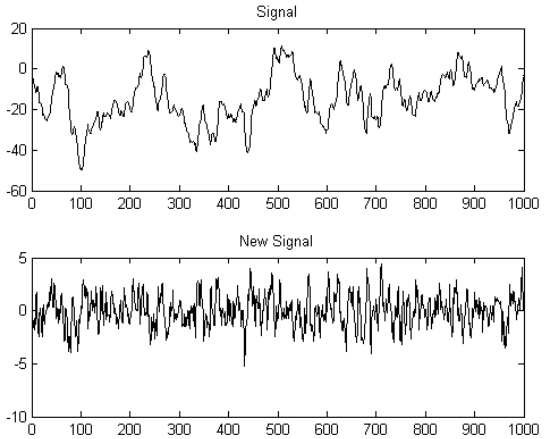

Рис. 5: Взятие производной

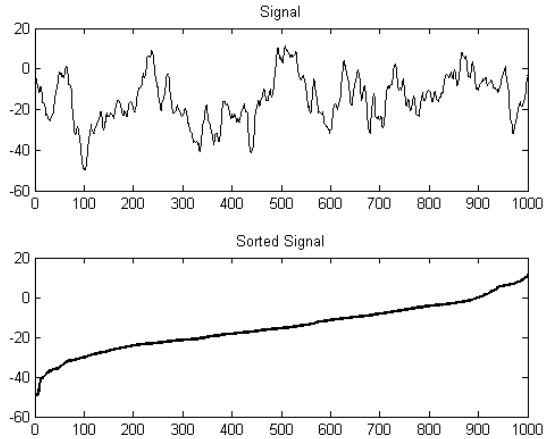

Рис. 6: Сортировка значений ряда

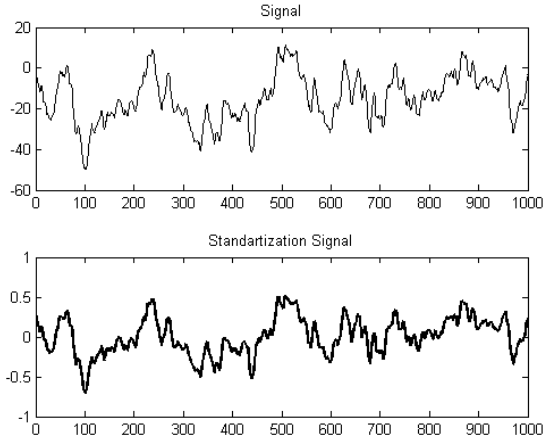

Рис. 7: Стандартизация значений ряда

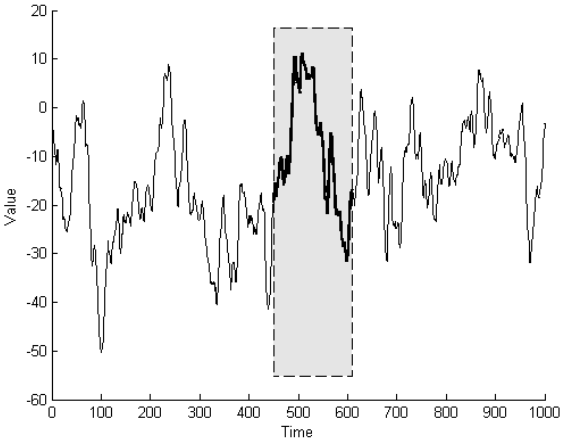

Рис. 8: Выделение подряда

Операторы второго типа

Операторы второго типа преобразуют сигнал в число, вычисляя его некоторую характеристику. Множество операторов второго типа, как и в случае операторов первого типа, формируется на основе экспертных знаний о природе сигналов. При решении прикладных задач Brain–Computer Interface [10] и Ford Classification Challenge [26] были использованы операторы, записанные в табл. 2.

Таблица 2: Множество $\mathfrak{C}$ операторов второго типа, реализованных в программной среде MATLAB

№	Описание оператора
C1	среднее значение сигнала
C2	максимальное значение сигнала
C3	сумма максимального и минимального значений сигнала
C4	стандартное отклонение
C5	число значений сигнала определенного типа
C6	число различных значений сигнала
C7	“центр масс” сигнала

Приведем программный код на языке MATLAB упомянутых операторов второго типа.

1. *Среднее значение сигнала*
```
feature = mean(Signal);
```

2. *Максимальное (минимальное) значение сигнала*
```
feature = max(Signal); (feature = min(Signal);)
```

3. *Сумма максимального и минимального значений сигнала*
```
feature = max(Signal) + min(Signal);
```

4. *Стандартное отклонение*
```
feature = std(Signal);
```

5. *Число значений сигнала определенного типа*

$$\mathrm{num}_X(x_1, x_2, ..., x_s) = |\{i \in \{1, 2, ..., s\}, x_i \in X\}|$$

6. *Число различных значений сигнала*
   ```
   feature = length(unique(Signal));
   ```

7. *"Центр масс" сигнала*
   Вычислим "центр масс" для сигнала $\mathbf{x} = (x_1, x_2, ..., x_s)$:
   $\mathbf{x} \cdot (1, \ldots, s)^{\mathrm{T}}/s$

   На языке MATLAB данный оператор записывается следующим образом:
   ```
   feature = [1:len] * Signal;
   ```

Кодирование особей популяции

Обозначим $\mathfrak{B}$ – множество операторов первого типа

$$B_i \in \mathfrak{B} : \mathbf{x} = (x_1, x_2, ..., x_s) \to \mathbf{y} = (y_1, y_2, ..., y_p),$$

$\mathfrak{C}$ – множество операторов второго типа

$$C_j \in \mathfrak{C} : \mathbf{y} = (y_1, y_2, ..., y_p) \to z.$$

Признак, описывающий сигнал ищем в виде:

$$C_{i_0}(B_{i_k}(...(B_{i_1}(\mathbf{x})))), \tag{4.1}$$

где $k \in \{1, 2, ...\}$, $B_{i_k}, ..., B_{i_1} \in \mathfrak{B}$, $C_{i_0} \in \mathfrak{C}$.

Будем решать задачу поиска оптимального признака (4.1), используя генетический алгоритм. Для этого необходимо закодировать решения (особей). На каждом шаге генетического алгоритма имеется некоторая текущая популяция – множество признаков вида (4.1). Любой признак допускает представление в виде:

$$[i_1]...[i_k][i_0]$$

Таким образом, признак кодируется индексами операторов, которые его определяют. При этом мы разрешаем, чтобы признаки содержали различное число операторов в представлении (4.1). Естественно ограничить максимально возможное количество операторов в представлении признака. Это константа реализации.

В случае поиска оптимальной пары признаков в совокупности особью популяции будет эта пара.

4.2 Функция приспособленности

В данной работе предлагается строить хорошее признаковое описание сигнала с помощью генетического подхода к задаче оптимизации. Чтобы иметь возможность воспользоваться данным методом, необходимо формализовать понятие "качества" признакового пространства. Приспособленность особи должна отражать дискриминантную способность

отдельного признака (совокупности признаков). Чем приспособленность выше, тем более хорошее признаковое пространство построено.

Качество отдельного признака оценивается критерием AUC–ROC [5], а совокупности признаков — методом *скользящего контроля с исключением объектов по одному* (leave-one-out, LOO) по алгоритму *k ближайших соседей* (k nearest neighbors, kNN) [6]. При использовании алгоритма k ближайших соседей осуществляется подбор оптимальной взвешенной евклидовой метрики.

4.2.1 Оценка качества отдельного признака с помощью критерия AUC

Рассмотрим признак $[i_1]...[i_k][i_0]$. Под действием операторов, входящих в состав признака, каждый сигнал преобразуется в вещественное число. Объекты обучающей выборки таким образом представлены точками на прямой.

Существуют различные подходы к оценке информативности отдельного признака. Примерами могут служить коэффициент корреляции Пирсона (между расположением объекта на прямой и номером класса, к которому он принадлежит) или известный критерий Фишера (отношение межклассового рассеяния к внутриклассовому разбросу). Можно предлагать и другие эвристики оценки дискриминантной способности признака.

В рамках данного подхода предлагается воспользоваться критерием AUC–ROC. В главе 3 описаны основы теории ROC–анализа (более подробно см. [5]).

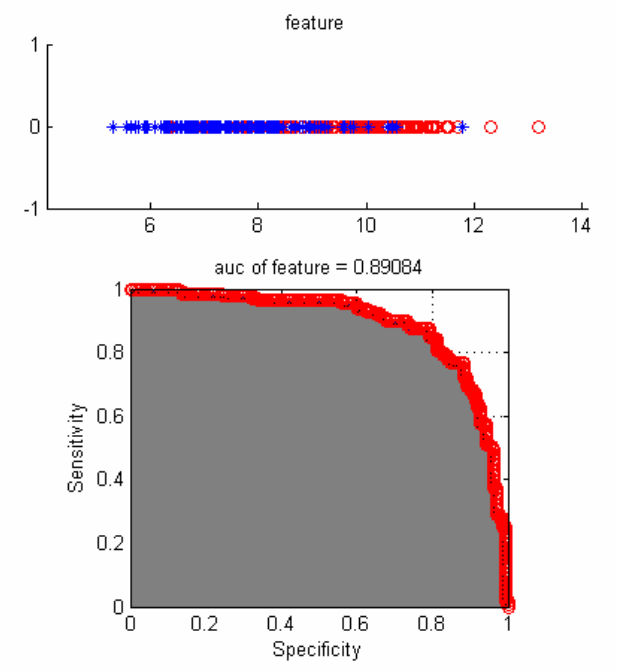

Рис. 9: Пример 1

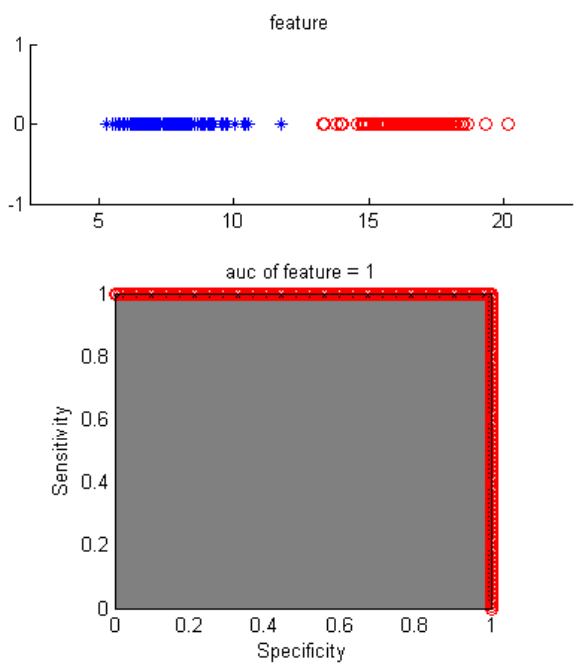

Рис. 10: Пример 2

При построении ROC–кривой значение признака для сигнала будем трактовать как своего рода “вероятность” того, что прецедент относится к положительному классу. Считаем, что, чем правее на оси лежит точка (относительно остальных объектов), тем вероятней она является положительным примером, и, наоборот, чем левее — тем эта вероятность меньше. Идеальный случай, когда все отрицательные примеры левее всех положительных объектов (классы линейно разделимы). В этой ситуации AUC = 1. Мы хотим, чтобы

ROC–кривая отражала зависимость (корреляцию) между расположением точки на прямой относительно других объектов и ее классификацией.

На рис. 9, 10 представлены примеры, иллюстрирующие вышесказанное. Положительный класс обозначается красным кружком, а отрицательный — синей звездочкой.

Понятно, что чем выше показатель AUC, тем признак качественнее.

4.2.2 Оценка качества пары признаков методом скользящего контроля по алгоритму ближайшего соседа

Пусть необходимо оценить качество пары признаков $f_1 = [i_1]...[i_k][i_0]$ и $f_2 = [j_1]...[j_l][j_0]$. Применяя к сигналам соответствующие операторы, представляем объекты обучающей выборки в двумерном признаковом пространстве (f_1, f_2) точками на плоскости (рис. 11).

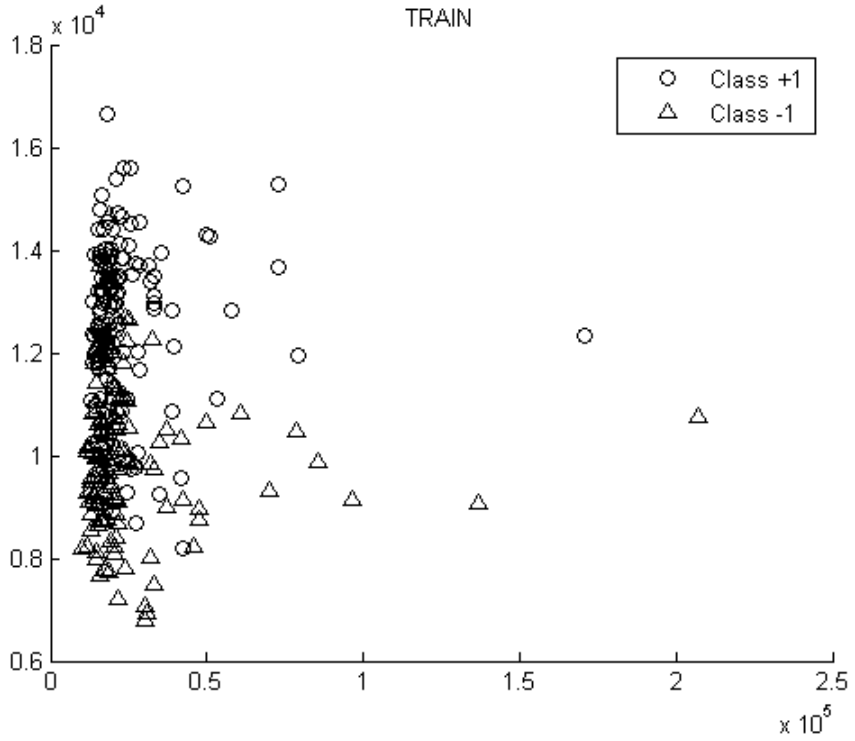

Рис. 11: Представление объектов в двумерном пространстве. В качестве исходных данных использованы сигналы прикладной задачи Brain–Computer Interface (обучающая выборка) [10, 11]. Признаковое пространство: $f1 = [B7, k = 3][B6][B7, k = 5][C4]$, $f2 = [B4, k = 2][B5][C1]$.

Приспособленность особи определяем как долю верно классифицированных объектов обучения методом *скользящего контроля с исключением объектов по одному* (leave-one-out, LOO) по алгоритму k *ближайших соседей* (k nearest neighbors, kNN) [6].

Будем классифицировать объекты обучающей выборки путём голосования по k ближайшим соседям. Для произвольного объекта $u = x^j \in X^L$ расположим элементы обучающей выборки $X^L \backslash \{u\}$ в порядке возрастания расстояний до u:

$$\rho(u, x^{1,u}) \leqslant \rho(u, x^{2,u}) \leqslant ... \leqslant \rho(u, x^{L-1,u}),$$

где через $x^{i,u}$ обозначается i-й сосед объекта u. Аналогичное обозначение введём и для ответа на i-м соседе: $y^{i,u}$. Каждый объект $u \in X^L$ порождает свою перенумерацию выборки $x^{1,u}, ..., x^{L-1,u}$. Каждый из соседей $x^{i,u}, i = 1, ..., k$ голосует за отнесение объекта u к своему

классу $y^{i,u}$. Алгоритм относит объект u к тому классу, который наберёт большее число голосов:

$$a(u; X^L \backslash \{u\}, k) = \arg\max_{y \in \{+1,-1\}} \sum_{i=1}^{k} [y^{i,u} = y].$$

(Здесь скобками обозначен индикатор, принимающий значение 1 при выполненном выражении в скобках и 0 иначе.)

Для каждого объекта обучающей выборки проверяем, правильно ли он классифицируется по своим k ближайшим соседям.

Функционал качества пары признаков в совокупности вычисляется согласно формуле:

$$\text{fitness}(f_1, f_2) = \frac{1}{L} \cdot \sum_{j=1}^{L} \left[a(x^j; X^L \backslash \{x^j\}, k) \neq y^j \right]. \tag{4.2}$$

Преимущества LOO в том, что каждый объект ровно один раз участвует в контроле, а длина обучающих подвыборок лишь на единицу меньше длины полной выборки. Недостатком LOO является большая ресурсоёмкость, так как обучаться приходится L раз.

Следует отметить, что максимальная сумма голосов может достигаться на нескольких классах одновременно. В задачах с двумя классами этого можно избежать, если брать только нечетные значения k.

Метод kNN относится к метрическим алгоритмам классификации, и результат его работы сильно зависит от выбранной метрики. В качестве функции расстояния между двумя объектами $u = (u^1, u^2)$ и $v = (v^1, v^2)$ в двумерном признаковом пространстве была использована стандартная евклидова метрика:

$$\rho(u, v) = \sqrt{(u^1 - v^1)^2 + (u^2 - v^2)^2}.$$

Также исследовалась и взвешенная евклидова метрика:

$$\rho'(u, v) = \sqrt{(u^1 - \text{coeff} \cdot v^1)^2 + (u^2 - \text{coeff} \cdot v^2)^2},$$

где coeff — автоматически настраиваемый параметр. Смысл заключается в растяжении либо сжатии одного из признаков для повышения качества классификации. При этом происходит поиск пары признаков вида $(f_1, \text{coeff} \cdot f_2)$.

4.3 Процедура поиска качественных признаков

В предыдущих разделах 4.1 и 4.2 данной главы были формализованы такие понятия генетического алгоритма, как особь популяции и ее приспособленность. Теперь мы имеем все необходимое, чтобы описать пошаговое действие алгоритма.

В любой момент времени популяция состоит из константного числа особей, не изменяющегося от эпохи к эпохе. Это параметр алгоритма. Обозначим размер популяции через `sizePop`. Начальная популяция, состоящая из заданного количества особей, формируется случайным образом. Шаг алгоритма состоит из трех стадий: генерация промежуточной популяции путем отбора особей из текущего поколения, скрещивание особей промежуточной популяции путем кроссовера, мутация особей. Выполнение трех этапов приводит к формированию нового поколения. Процесс длится пока не достигнем локального максимума.

Следует отметить, что число эпох, необходимых для получения приемлемого результата, зависит от специфики решаемой задачи. Так, при решении реальной задачи классификации сигналов Brain-Computer Interface (см. раздел 5.1) выяснилось, что генетическая оптимизация в предложенном варианте не более, чем за пять итераций строит качественное признаковое пространство, после чего средняя приспособленность популяции повышается, но наиболее качественное найденное решение не изменяется. В разделе 5.1 приведено обоснование быстрой сходимости алгоритма.

Стратегия формирования нового поколения

Для формирования нового поколения используется принцип *элитизма* [3]. В новое поколение обязательно включаются две самые лучшие особи, которые не участвуют в размножении и к которым не применяется оператор мутации. В промежуточную популяцию отбирается `(sizePop - 2)` особи, которые будут участвовать в скрещивании. После этапа скрещивания к популяции применяется оператор мутации. После этого осуществляется переход на следующую эпоху. Новое поколение, таким образом, состоит из полученной популяции размера `(sizePop - 2)` особей и двух сильнейших предыдущего поколения.

Пошаговое действие алгоритма описано ниже.

Алгоритм 4.1 Алгоритм работы генетического метода

Вход: $sizePop, MaxNumIteration, pCross, pMut$;

Выход: $A = \arg\max_{A \in SetA} fit(A)$;

1: Создать $sizePop$ случайных признаков (пар признаков) $SetA = (A_1, ..., A_{sizePop})$;
2: **для** $j = 1, \ldots, MaxNumIteration$
3: Для каждой особи $A_i \in SetA$ вычислить ее приспособленность $fit(A_i)$;
4: Выбрать $sizePop - 2$ особей, используя пропорциональный отбор, занести в $SetANew$;
5: Случайным образом разбить особи популяции $SetANew$ на пары;
6: С определенной вероятностью $pCross$ применить к каждой паре оператор кроссовера;
7: С определенной вероятностью $pMut$ применить к каждой новой особи оператор мутации;
8: Добавить в промежуточную популяцию $SetANew$ две лучшие особи популяции $SetA$;
9: $SetA := SetANew$;
10: Выдать $\arg\max_{A \in SetA} fit(A)$;

4.3.1 Отбор особей для скрещивания

В качестве стратегии отбора родительских хромосом выбран *пропорциональный отбор* [3]. Вероятность каждой особи попасть в промежуточную популяцию пропорциональна ее приспособленности. Данную стратегию отбора можно трактовать как неоднократный запуск рулетки, где размер сектора каждой особи пропорционален ее приспособленности. Для отбора N особей запускаем рулетку N раз, причем ни одна выбранная особь не удаляется с колеса. При таком отборе члены популяции с более высокой приспособленностью с большей вероятностью будут чаще выбираться, чем плохо приспособленные особи. В англоязычной литературе данный тип отбора известен под названием *stochastic sampling*.

4.3.2 Скрещивание

После отбора особи промежуточной популяции случайным образом разбиваются на пары. Каждая из них с вероятностью `pCross = 0.8` скрещивается, в результате чего образуется два потомка. Они заносятся в новое поколение. Если же паре не выпало скрещиваться, то сами особи этой пары остаются в популяции.

Пусть мы хотим скрестить два признака — $[i_1]...[i_k][i_0]$ и $[j_1]...[j_l][j_0]$. Без ограничения общности можно считать, что $l \leqslant k$. Преобразуем код второго признака в эквивалентный ему вид $[j_1]...[j_k][j_0]$, где $j_{l+1} = ... = j_k = 0$. Здесь 0 — индекс, обозначающий отсутствие гена.

Использовались два различных оператора скрещивания:

1. *Одноточечный оператор кроссовера.*
   Для родительских хромосом случайным образом выбирается точка раздела $z \in \{1, 2, ..., k\}$, и они обмениваются отсеченными частями. В результате получаются два потомка:
   $$[i_1]...[i_z][j_{z+1}]...[j_k][j_0],$$
   $$[j_1]...[j_z][i_{z+1}]...[i_k][i_0].$$

2. *Однородный кроссовер.*
   В данном случае каждый ген хромосом родителей с определенной вероятностью p = 0.5 переходит к тому или иному потомку.
   $$[u_1]...[u_k][u_0],$$
   $$[v_1]...[v_k][v_0],$$
   где с вероятностью $\begin{cases} p = 0.5, & u_z = i_z; v_z = j_z; \\ p = 0.5, & u_z = j_z; v_z = i_z. \end{cases}$

Из кода потомков удаляются индексы 0.

В случае поиска оптимальной пары признаков особи скрещиваются посредством независимого скрещивания первых признаков особей и вторых признаков.

4.3.3 Мутация

Для всякой особи применяется оператор мутации. Каждый ген с заданной вероятностью `pMut = 0.009` заменяется на произвольный новый ген.

В случае поиска оптимальной пары признаков мутация может быть нескольких типов: обмен признаков в особи местами, либо замена (с определенной вероятностью) одного из операторов в составе любого признака данной особи на произвольный.

С одной стороны, оператор мутации вытягивает из локальных экстремумов, с другой — приносит новую информацию в популяцию.

4.4 Построение классификатора

После того, как оптимальный признак (оптимальная пара признаков) найден, представляем объекты обучения и контроля исходной задачи (сигналы) в построенном признаковом пространстве. В качестве классификатора используем метод k ближайших соседей [6] со взвешенной евклидовой метрикой.

5 Вычислительные эксперименты

Было проведено исследование предлагаемых методов на реальных задачах классификации сигналов с двумя непересекающимися классами: “Ford Classification Challenge” [26] и “BCI competition” [10, 11]. Природа исходных данных этих прикладных задач различна, однако описанные методы классификации сигналов успешно работают в обоих случаях.

5.1 Brain-Computer Interface

Brain-Computer Interface (BCI) [10, 11] — приложение, находящиеся на стыке таких наук, как: медицина, физиология, нейрология, машинное обучение и обработка сигналов. Основная цель исследований в области Brain-Computer Interface — организация связи между человеческим мозгом и внешним устройством.

Это оказывается возможным благодаря тому, что во время своей активности человеческий мозг генерирует электромагнитные поля. Анализируя данные поля, можно делать выводы о совершаемых человеком ментальных действиях и использовать “физиологические сигналы” для управления механизмами и приборами. Для измерения электрического поля достаточно определить разность потенциалов между электродами, прикреплёнными к скальпу. Существуют различные методы снятия сигнала: electroencephalography (EEG), electrocorticography (ECoG), и т.д. Они отличаются характеристиками принимаемого сигнала (частота, отношение “полезный сигнал/шум”), инвазивностью (имплантирование электродов в кору), размерами электродов, покрываемой площадью (количество нейронов, которые влияют на сигнал, снимаемый одним электродом). Данные электроды не оказывают никакого электрического воздействия на человеческий мозг, они лишь фиксируют электрическую активность мозга.

С помощью специального аппарата с коры головного мозга человека снимаются показания (измеряются значения потенциала в последовательные моменты времени) и представляются в виде ряда. Задача ученных в области машинного обучения — настроить классифицирующий алгоритм на обучающей выборке (сигналы, снятые во время известных ментальных действий) таким образом, чтобы в дальнейшем эффективно определять класс ментальных действий по сигналу. На рис. 12 представлена упрощенная схема функционирования системы Brain-Computer Interface.

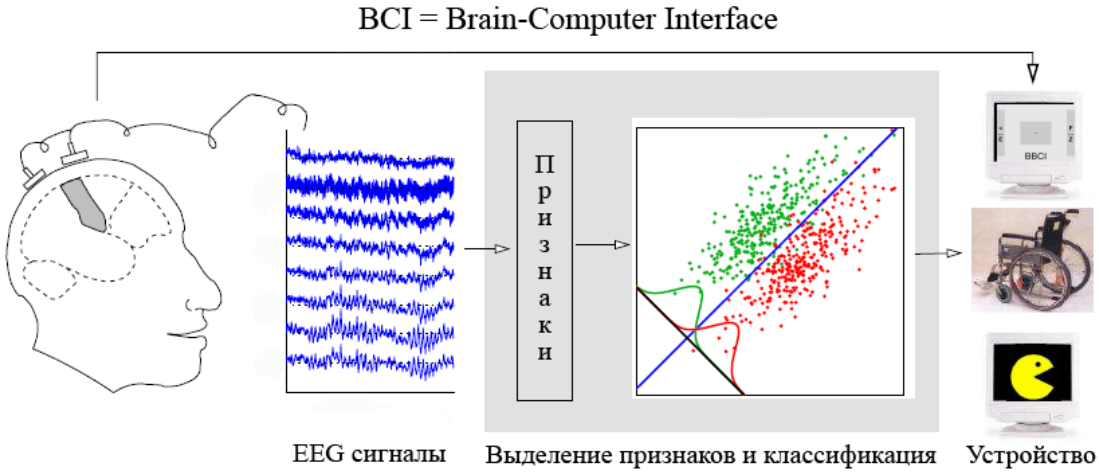

Рис. 12: Общая схема работы системы Brain-Computer Interface

Первая электроэнцефалограмма, снятая с человеческого мозга была записана австралийским психиатром Бергером в двадцатых годах XX века, тем самым было доказано, что мозг обладает электрической активностью. В 1929 году была опубликована его работа [17]. Активными исследованиями проблемы коммуникации человек–ЭВМ занялись ученые Калифорнийского Университета в Лос–Анжелесе [21, 22, 23] в 1970 году. В настоящее время данной проблемой занимаются несколько десятков научных групп и лабораторий [16], сотни ученых смежных областей [13, 14, 15, 20], образованы исследовательские институты [18, 19], проводятся различные конкурсы, предлагающие классифицировать реальные данные задачи Brain-Computer Interface.

Рассматриваемый в дипломной работе подход был протестирован на данных конкурса BCI Competition III [10, 16].

Приложения Brain-Computer Interface: контроль протезов и управление механизмами, организация общения с людьми с ограниченными физическими возможностями, компьютерные игры, исследование активности головного мозга в научных целях.

5.1.1 Данные эксперимента

Характерной особенностью данных соревнований BCI Competition III [10, 16] является то, что обучающая и контрольная выборки были сформированы в различные дни (с интервалом в 1 неделю). Испытуемый мог в один из дней быть более уставшим, что непосредственно отражается на активности головного мозга. Кроме того, электроды могли находиться в разных положениях (быть чуть смещёнными), могли немного измениться сопротивления проводников в системе обработки сигнала и т.д. В результате этого обучающие и контрольные выборки в построенном признаковом пространстве оказались расположенными на некотором расстоянии друг относительно друга. Данная проблема и способ ее решения описаны ниже в главе 6.2.

Во время эксперимента испытуемый выполнял два вида мозговой деятельности. Электрическая активность его головного мозга определялась с помощью ECoG-платиновой сетки из 64 электродов размера 8×8 см. Сигналы записывались в течение 3 секунд с частотой 1000МГц. В первый день было снято 278 записей мозговой активности. Таким образом, в обучении было 278 объектов, которые описывались с помощью 3000×64 значений (64 временных ряда по 3000 точек в каждом), 2 непересекающихся класса. Контрольная выборка состояла из 100 объектов.

При проведении исследований [7] было найдено 7 электродов, сигналы с которых обеспечивают наилучшее качество классификации.

5.1.2 Результаты эксперимента

Приведем пример работы генетического подхода. Была построена пара признаков: $f1 = [B6][B4, k = 5][C2]$, $f2 = [B4, k = 3][B5][C1]$. Здесь $B6$ — покомпонентное логарифмирование ряда, $B4$ — оператор взятия производной, $C2$ — максимальное значения сигнала; $B5$ — оператор взятия абсолютного значения, $C1$ — среднее значение сигнала (см. табл. 1, 2). Ошибка классификации на контроле составляет 13% при использовании процедуры “совмещения” кластеров обучения и контроля (см. раздел 6.2). Данная пара признаков визуализирована на рис. 13.

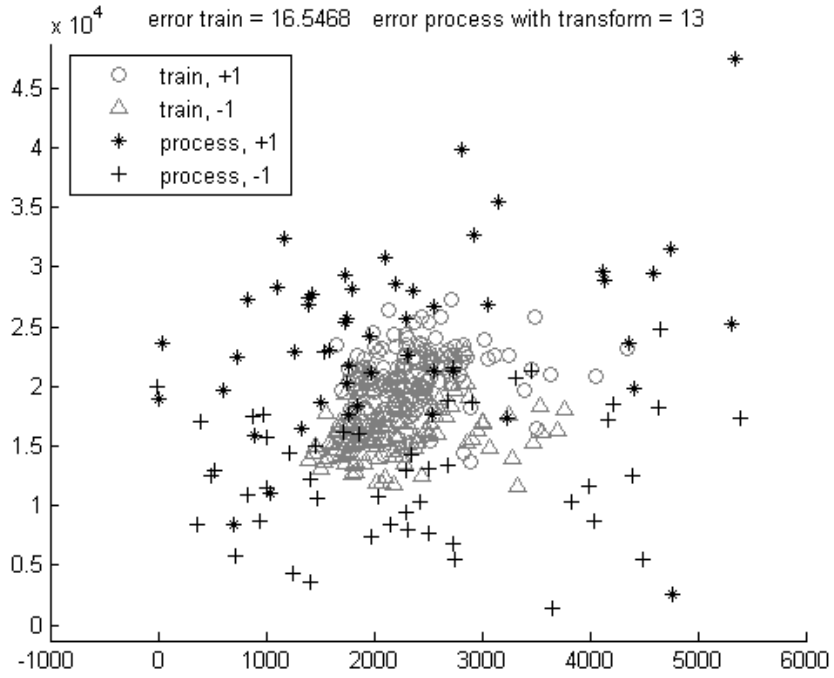

Рис. 13: Пример работы генетического подхода на данных задачи BCI. Построено признаковое пространство: $f1 = [B6][B4, k = 5][C2]$, $f2 = [B4, k = 3][B5][C1]$. Использована процедура "совмещения" кластеров обучения и контроля. Ошибка классификации на контроле составляет 13%.

Для некоторой объективной оценки предлагаемого подхода к решению задачи классификации сигналов было проведено следующее исследование. Все признаки, содержащие в своем представлении (4.1) не более четырех операторов первого типа, были упорядочены по убыванию показателя AUC. Результат представлен на рис. 14.

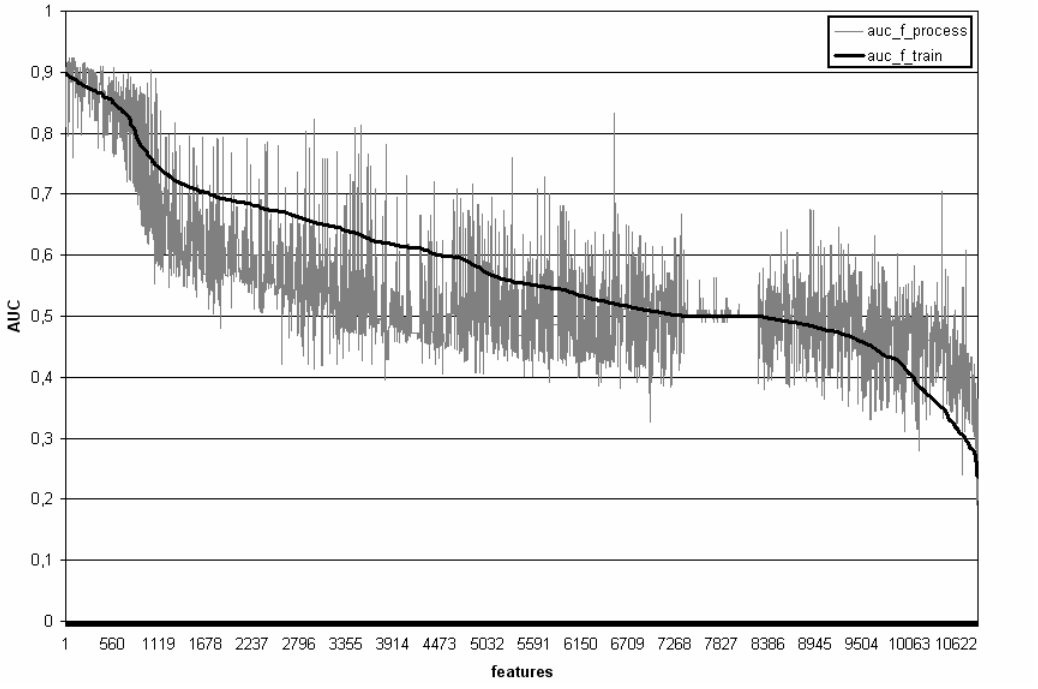

Рис. 14: Показатель AUC на обучении и контроле задачи BCI

На диаграмме серым цветом показаны соответствующие показатели AUC для кон-

троля. Анализируя поведение графиков, можно сделать несколько выводов:

- Виден эффект переобучения — серый график, соответствующий значениям показателя AUC на контроле, лежит ниже черной жирной кривой. Это означает, что в среднем ошибка на контроле больше ошибки на обучении;
- Заметна широкая полоса хороших признаков, для которых показатель AUC на обучении высокий, а переобучение минимально. С большой степенью вероятности мы попадем в данную область. Поэтому генетический алгоритм довольно быстро сходится;
- Графики совпадают при значениях показателя AUC = 0.5, что соответствует "константам" — бесполезным классификаторам.

5.2 Ford Classification Challenge

Также, было проведено исследование описанного подхода на данных соревнований Ford Classification Challenge [26]. Сигнал в данном случае описывает характеристику некоторого технического устройства в последовательные моменты времени. Задача распознавания Ford Classification является двухклассовой — поведение наблюдаемого устройства либо стандартное, либо имеются некоторые отклонения.

5.2.1 Данные эксперимента

Объекты распознавания представляют собой ряды длиной в 500 значений. Время начала записи сигнала не связано с каким-либо внешним обстоятельством или другим аспектом наблюдаемого объекта.

Каждый объект может быть отнесен к одному из двух классов: сигналы, описывающие поведение технического устройства в случае присутствия некоторого симптома (класс имеет пометку $+1$), либо в случае отсутствия симптома — класс помечен -1. Все сигналы разбиты на две выборки — обучение и контроль. Требуется построить классификатор, отличающий объекты первого класса от сигналов второго класса.

Организаторы соревнований [26] предлагают две независимые между собой задачи:

1. **Ford_A**
   Данные, как обучение, так и контроль, собраны при типичных условиях, с минимальным загрязняющим шумом
2. **Ford_B**
   Обучающая выборка собрана при типичных условиях, в то время как контрольная выборка зашумлена

Объемы данных приведены в следующей таблице.

Задача	Число объектов обучения	Число объектов контроля
Ford_A	3271	330
Ford_B	3306	330

Качество классификации предлагается оценивать по двум критериям:

- Точность (Accuracy) — доля верно классифицированных сигналов контроля от общего объема контрольной выборки;

- Доля ложно положительных примеров (False Positives Rate).

5.2.2 Результаты эксперимента

Задача Ford_A

Генетический подход к задаче классификации в данном случае позволил найти информативные признаки. Существует одномерное признаковое пространство, обеспечивающее линейную разделимость классов. Приведем один из них. Оказалось, что для сигналов с положительной меткой выполнено свойство: их значения практически не повторяются, т. е. число различных значений за эпоху (500 точек) крайне близко к 500. Однако это свойство невыполнено для сигналов второго класса.

На рис. 15, 16 признак визуализирован, соответственно, для обучения и контроля.

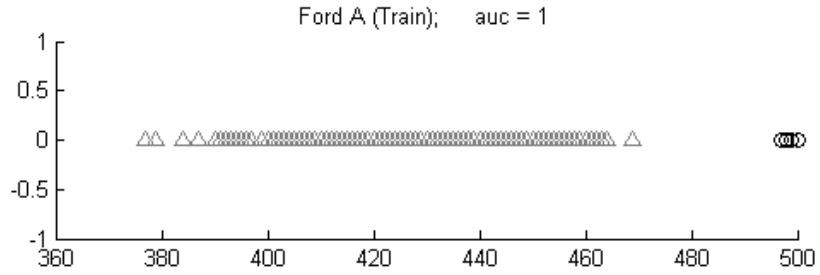

Рис. 15: Признак, линейно разделяющий классы. (Обучающая выборка задачи Ford A)

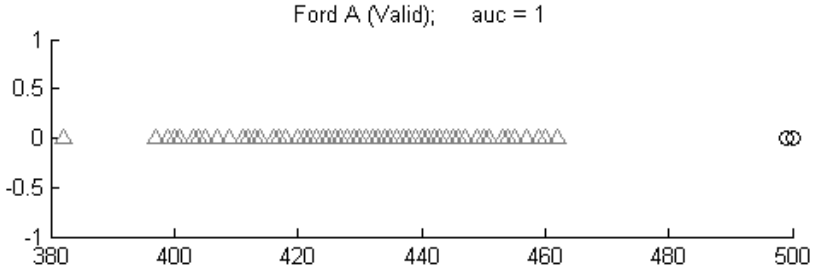

Рис. 16: Признак, линейно разделяющий классы. (Контрольная выборка задачи Ford A)

Задача Ford_B

При решении второй задачи качество классификации на контрольной выборке оказалось ниже, чем в задаче Ford B. Это связано с зашумленностью контрольной выборки при качественной выборки обучения.

Приведем пример работы генетического подхода. Было построено двумерное признаковое пространство. Ошибка классификации на контроле составляет 21%. Данная пара признаков визуализирована на рис. 17.

Также было проведено следующее исследование. Все признаки, содержащие в своем представлении (4.1) не более трех операторов первого типа, были упорядочены по убыванию показателя AUC. Результат представлен на рис. 18. На диаграмме серым цветом показаны соответствующие показатели AUC для контроля.

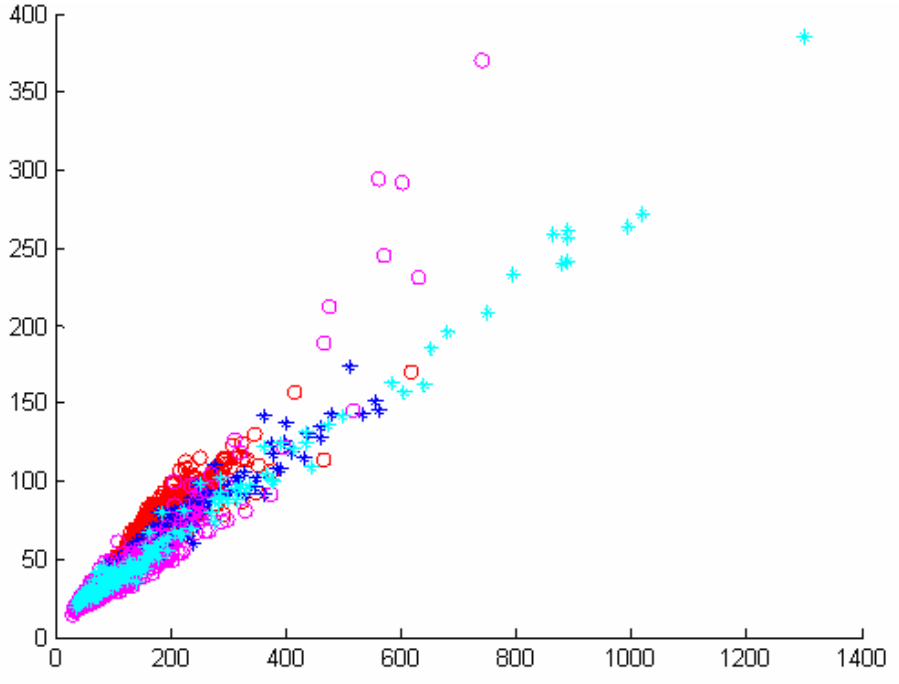

Рис. 17: Пример работы генетического подхода на данных задачи Ford B

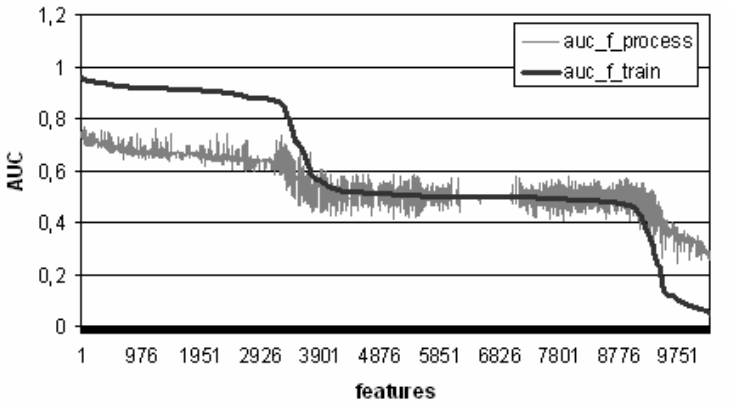

Рис. 18: Показатель AUC на обучении и контроле задачи Ford B

5.3 Сравнение с полным перебором

Результат работы генетического подхода к построению качественного признакового пространства сравнивался с результатом, полученным переборным алгоритмом. При этом базу сигналов составили данные соревнований BCI Competition III (см. пункт 5.1.1).

Полный перебор осуществлялся по всем признакам (парам признаков), содержащим в своем представлении (4.1) не более заданного количества операторов первого типа. При поиске отдельного признака функционалом качества являлся показатель AUC, при поиске пары признаков — критерий cfNN. Переборный алгоритм искал признак с максимальным значением функционала качества на обучении.

Генетический алгоритм строил признаковые пространства по схеме, описанной в разделе 4.3 с теми же функционалам качества AUC и cfNN.

Результаты работы алгоритмов сравнивались по двум критериям:

- ошибка классификации по методу kNN на контроле,
- время работы алгоритма.

В случае поиска отдельного признака f_1 ошибка классификации на контроле вычислялась по методу kNN в признаковом пространстве (f_1, f_1).

Полный результат исследований представлен в таблицах 3, 4. В первых основных колонках таблиц курсивом приведен лучший результат, который потенциально может быть получен. При исследовании генетического алгоритма была проведена серия из 100 экспериментов, после чего все вычисленные показатели были усреднены.

В таблицах 3, 4 использованы следующие обозначения:

- B — любой оператор первого типа (см. пункт 4.1),
- r — количество операторов первого типа типа в представлении признака (4.1),
- er_proc — ошибка классификации по методу kNN на контроле,
- AUC — критерий AUC–ROC (функционал качества отдельного признака, см. пункт 4.2.1),
- er_train — ошибка классификации по методу kNN на обучении (функционал качества совокупности признаков, см. пункт 4.2.2),
- time — время работы алгоритма (в сек.).

Следует отметить, что при поиске пары признаков в совокупности, они могли содержать в своем представлении (4.1) различное число операторов первого типа.

Для наглядности ниже на рис. 19, 20 представлены диаграммы сравнения генетического алгоритма с переборным методом для случая поиска отдельного признака.

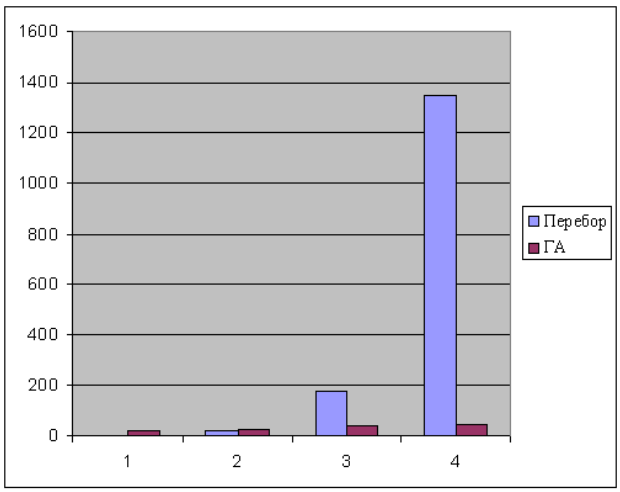

Рис. 19: Время работы алгоритма поиска качественного признака

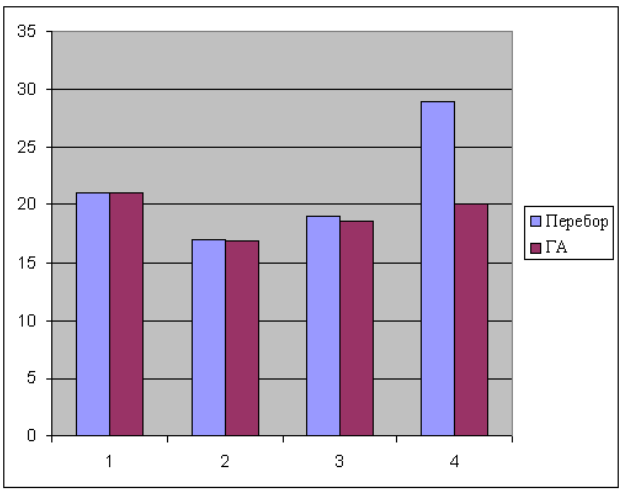

Рис. 20: Ошибка классификации на контроле задачи BCI

Переборный метод работает существенно дольше генетического подхода, и его время работы возрастает по экспоненте, что и подтверждается рис. 19. На второй диаграмме видно, что ошибка классификации на контроле у переборного алгоритма во всех четырех случаях оказалась выше ошибки генетического метода.

Таблица 3: Поиск отдельного признака ($f_2 = f_1$)

	Полный перебор по всем признакам, содержащим не более r операторов B. *Признак ищем так, чтобы минимизировать ошибку на контроле.*		**Полный перебор по всем признакам, содержащим не более r операторов B.** Функционал качества — критерий AUC. Максимизируем функционал AUC на обучении, вычисляем ошибку классификации на контроле.			**Генетический алгоритм.** Искомый признак содержит не более r операторов B. Функционал качества — критерий AUC. `sizePop = 100`, `MaxNumIteration = 5`				
	er_proc	***AUC***	**er_proc**	**AUC**	time	**er_proc ср. зн.**	er_proc мин. зн.	er_proc макс. зн.	**AUC ср. зн.**	time ср. зн.
$r = 1$	*21*	*0.8813*	**21**	**0.8813**	2.106	**21**	21	21	**0.8813**	19.77
$r = 2$	*13*	*0.8857*	**17**	**0.8915**	20.81	**16.86**	13	22	**0.8876**	28.02
$r = 3$	*13*	*0.8857*	**19**	**0.8952**	175.72	**18.63**	13	29	**0.8911**	37.44
$r = 4$	*13*	*0.8857*	**29**	**0.8985**	1344.49	**20.1**	13	29	**0.893**	45.79

Таблица 4: Поиск двух признаков в совокупности (f_1, f_2) (евклидова метрика)

	Полный перебор по всем парам признаков, содержащих не более r операторов B. *Признак ищем так, чтобы минимизировать ошибку на контроле.*		**Полный перебор по всем парам признаков, содержащих не более r операторов B.** Функционал качества — критерий cfNN. Максимизируем функционал cfNN на обучении, вычисляем ошибку классификации на контроле.			**Генетический алгоритм.** Искомый признак содержит не более r операторов B. Функционал качества — критерий cfNN. `sizePop = 100`, `MaxNumIteration = 5`				
	er_proc	***er_train***	**er_proc**	**er_train**	time	**er_proc ср. зн.**	er_proc мин. зн.	er_proc макс. зн.	**er_train ср. зн.**	time ср. зн.
$r = 1$	*17*	*25.8993*	**19**	**18.7**	49.43	**26.51**	17	44	**20.87**	62.54
$r = 2$	*12*	*23.02*	**46**	**15.46**	2259.88	**18.79**	14	42	**18.59**	80.86
$r = 3$						**18.91**	13	50	**17.63**	101.21
$r = 4$						**20.82**	14	46	**17.45**	121.98

6 Усовершенствование алгоритма

6.1 Использование взвешенной евклидовой метрики

В процессе поиска эффективной пары признаков, мы оптимизируем функционал качества, выражающийся долей верно классифицированных объектов по скользящему kNN с использованием стандартной евклидовой метрики. Предлагается воспользоваться взвешенной евклидовой метрикой — растягиваем либо сжимаем один из признаков. Ищем пару признаков вида $(f_1, \text{coeff} * f_2)$, где coeff – автоматически настраиваемый параметр. Ниже представлены результаты экспериментов на данных задачи Brain–Computer Interface (см. главу 5.1).

Были построены все пары признаков (f_1, f_2), содержащих в своем представлении (4.1) не более двух операторов первого типа. При этом признаки f_1 и f_2 пары могли содержать различное число операторов первого типа. Для каждой пары были вычислены ошибки классификации на контроле по методу kNN в признаковых пространствах (f_1, f_2) и $(f_1, \text{coeff} * f_2)$ (параметр coeff настраивался на обучающей выборке). Признаки были отсортированы по возрастанию ошибки классификации с использованием взвешенной евклидовой метрики. Результат приведен на рис. 21. Далее были отобраны те пары признаков, для которых данная ошибка меньше 30%. На рис. 22 представлен результат исследования. Анализируя поведение двух графиков, можно сделать вывод о том, что качество пары признаков может быть существенно улучшено за счет использования взвешенной евклидовой метрики вместо стандартной евклидовой метрики. На рис. 23 признаки отсортированы по возрастанию ошибки классификации с использованием стандартной евклидовой метрики и отобраны те из них, для которых данная ошибка меньше 30%.

На всех трех диаграммах черный график соответствует ошибке классификации с использованием взвешенной евклидовой метрики, серый график — с использованием стандартной евклидовой метрики. По горизонтальной оси отложены признаки, по вертикальной оси — ошибка классификации, выраженная в процентах.

Для некоторых пар признаков (f_1, f_2) ошибка классификации на контроле ухудшается при использовании взвешенной евклидовой метрики (точки, в которых черный график лежит выше серого). Это эффект "переобучения" — коэффициент coeff настраивается из соображений минимизации ошибки на обучении. Однако данный эффект наблюдается лишь для малого количества хороших пар признаков, а для большинства из них применение взвешенной евклидовой метрики дает положительный результат, что видно из рис. 21, 22, 23.

При использовании взвешенной евклидовой метрики при вычислении функционала качества пары признаков в генетическом подходе (см. 4.2.2) удается построить более качественное двумерное признаковое пространство, что видно из табл. 5.

Таблица 5: Серия из 20 экспериментов. Используется генетический подход для построения двумерного признакового пространства. При вычислении функционала качества пары признаков (см. 4.2.2 используется, соответственно, стандартная либо взвешенная евклидова метрика.

BCI	евклидова метрика	взвешенная евклидова метрика
мин.знач.	14 %	14 %
сред.знач	21.15 %	18.65 %

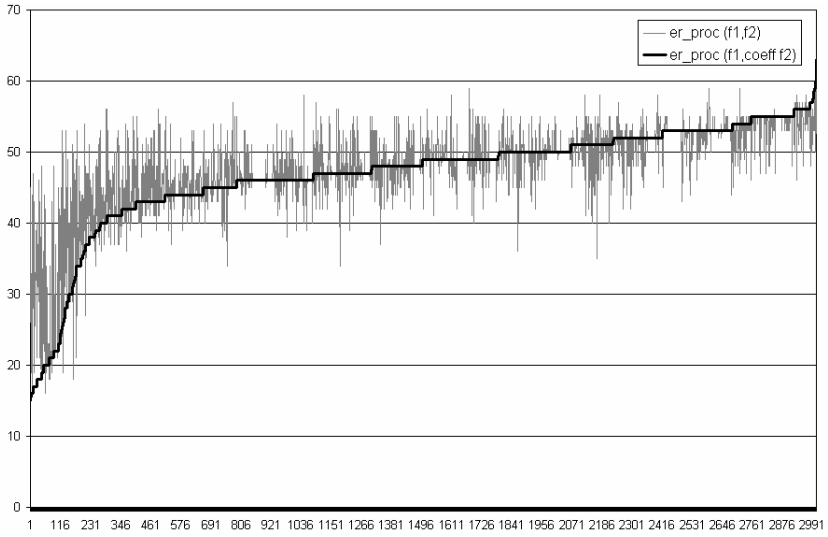

Рис. 21: Ошибки классификации по методу kNN со стандартной/взвешенной евклидовой метрикой для всех пар признаков, содержащих в своем представлении (4.1) не более двух операторов первого типа.

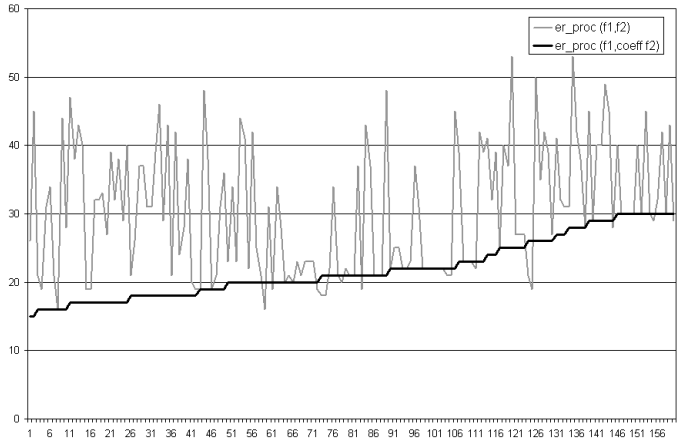

Рис. 22: Ошибки классификации по методу kNN с использованием стандартной/взвешенной евклидовой метрики для всех пар признаков (f_1, f_2), содержащих в своем представлении (4.1) не более двух операторов первого типа. Отобраны пары признаков, для которых ошибка с использованием взвешенной евклидовой метрики меньше 30%.

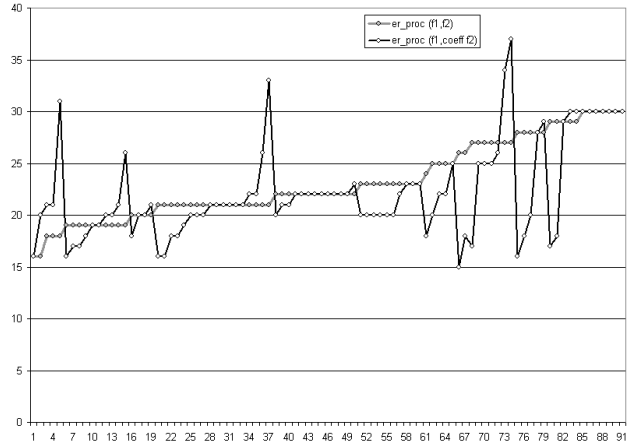

Рис. 23: Ошибки классификации по методу kNN с использованием стандартной/взвешенной евклидовой метрики для всех пар признаков (f_1, f_2), содержащих в своем представлении (4.1) не более двух операторов первого типа. Отобраны пары признаков, для которых ошибка с использованием стандартной евклидовой метрики меньше 30%.

На рис. 24, 25 показан пример пары признаков ($f1 = [B4, k = 1][C4]$, $f2 = [B6][B7, k = 5][C4]$), для которой при использовании взвешенной евклидовой метрики вместо стандартной в методе kNN существенно повышается качество классификации.

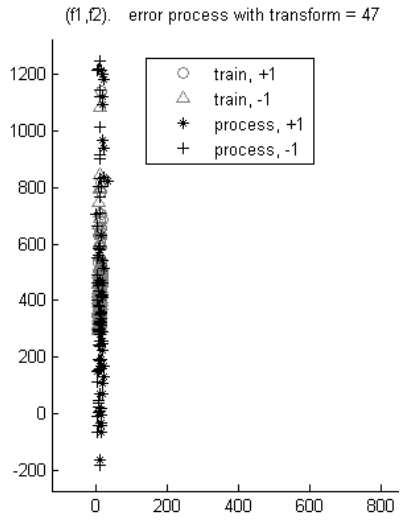

Рис. 24: Исходные данные: сигналы задачи BCI. Признаковое пространство: $f1 = [B4, k = 1][C4]$, $f2 = [B6][B7, k = 5][C4]$. В методе kNN использована **стандартная евклидова метрика**.

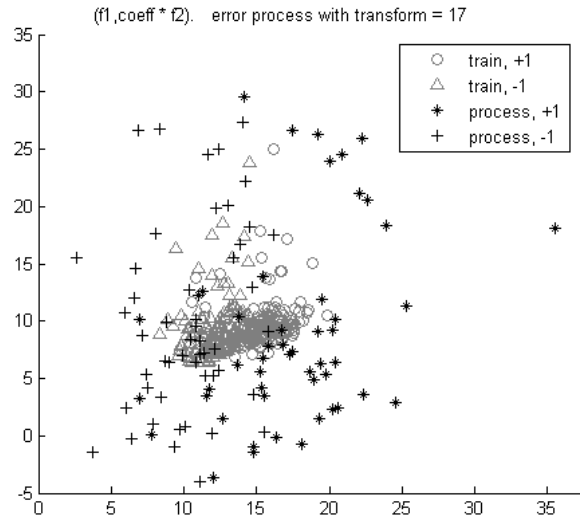

Рис. 25: Исходные данные: сигналы задачи BCI. Признаковое пространство: $f1 = [B4, k = 1][C4]$, $f2 = [B6][B7, k = 5][C4]$. В методе kNN использована **взвешенная евклидова метрика**.

6.2 Совмещение множеств обучения и контроля

При решении реальных задач нередко возникают случаи, когда контрольная выборка в признаковом пространстве располагается на некотором расстоянии от обучения, возможны ситуации, когда выпуклые оболочки этих множеств даже не пересекаются. В основном это происходит, когда обучающие и контрольные выборки формируются при различных условиях (см. пункт 5.1.1). Классификатор должен быть устойчив к подобного рода "перемещением" кластеров обучения и контроля друг относительно друга.

Поскольку в качестве классификатора используется метод kNN, который очень чувствителен к взаимному расположению объектов обучения и контроля, то необходимо это учитывать [8].

Предлагается непосредственно перед классификацией совмещать кластеры контроля и обучения. Было исследовано два вида преобразований — параллельный перенос (чтобы центры кластеров совпали) и поворот (чтобы совпали главные компоненты — ось, проекция на которую дает максимальный разброс). Выяснилось, что поворот не всегда приводит к желаемому результату в следствие наличия объектов-выбросов, в то время как совмещение центров необходимо практически во всех случаях.

Заметим, что в процессе поиска признаков данную трансформацию делать не нужно, так как мы работаем с одним и тем же множеством — обучением.

При решении задачи Brain-Computer Interface (см. главу 5.1) возникла данная проблема в результате того, что сигналы обучения и контроля были записаны в разные дни. На рис. 26, 27 показано сравнение классификации до и после совмещения множеств обучения и контроля на одной из построенных пар признаков. Ошибка классификации на обучении (с использованием скользящего контроля) составляет $17,62$ %. Ошибка классификатора

на контроле без совмещения множеств достигает 50 % (бесполезный классификатор), а при использовании процедуры совмещения кластеров оказывается равной 13 %.

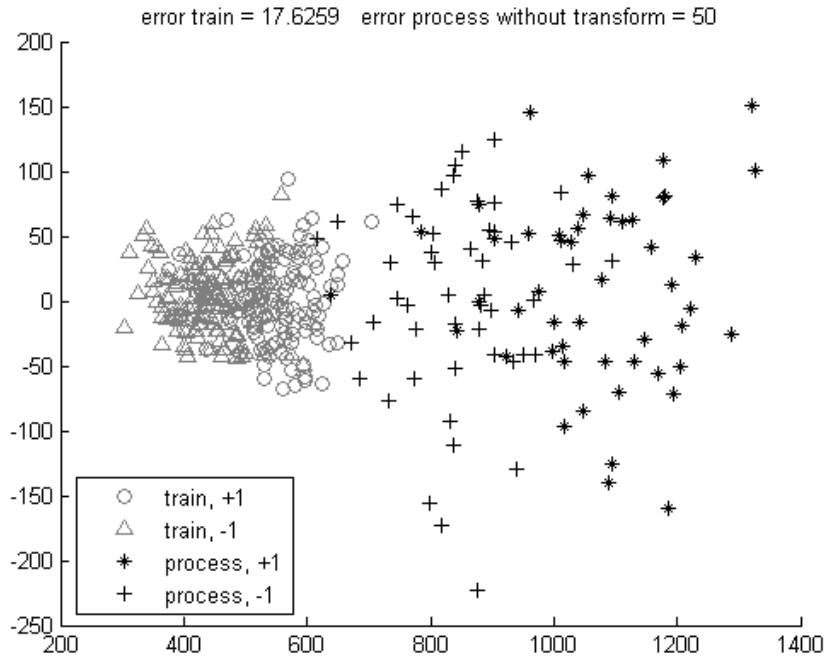

Рис. 26: **Без совмещения кластеров обучения и контроля.** В качестве исходных данных использованы сигналы прикладной задачи Brain–Computer Interface [10, 11]. Признаковое пространство: $f1 = [B4, k = 2][B6][B7, k = 5][C4],\ f2 = [B4, k = 3][C1]$.

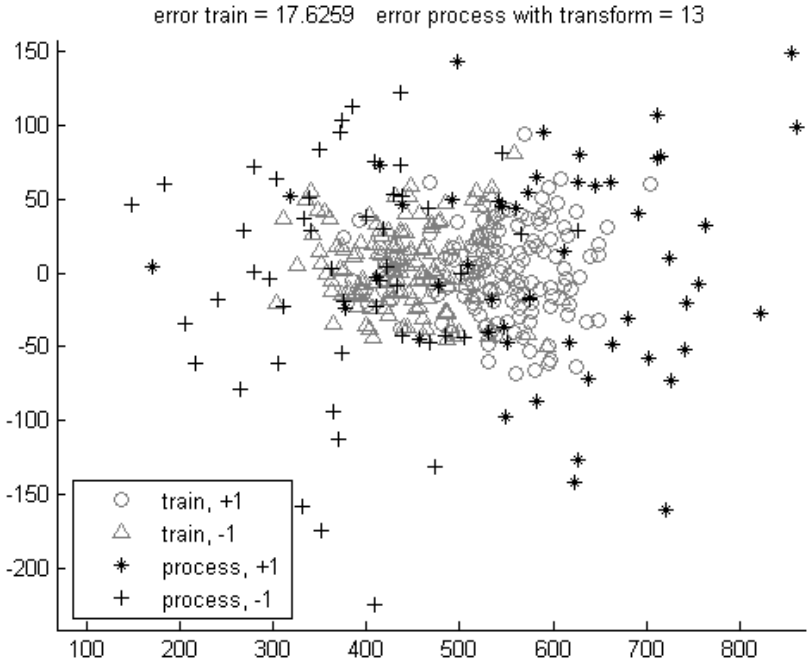

Рис. 27: **С совмещением кластеров обучения и контроля.** В качестве исходных данных использованы сигналы прикладной задачи Brain–Computer Interface [10, 11]. Признаковое пространство: $f1 = [B4, k = 2][B6][B7, k = 5][C4],\ f2 = [B4, k = 3][C1]$.

Была проведена серия из 100 экспериментов — с помощью генетического подхода построено 100 пар признаков, содержащих не более трех операторов первого типа обработки

сигнала. Для каждой пары признаков вычислена ошибка классификации на контроле с использованием и без процедуры совмещения множеств обучения и контроля. Полученные результаты приведены в таблице ниже.

BCI	Без трансформации	С трансформацией
мин. знач.	45 %	13 %
сред. знач	49.94 %	18.59 %

6.3 Использование похожести формы обучения и контроля в процессе построения эффективных признаков

При поиске оптимальной пары признаков функционалом качества пары служил критерий скользящего контроля по методу kNN. При тестировании алгоритма периодически возникали ситуации, когда в построенном признаковом пространстве формы класеров обучения и контроля существенно отличались друг от друга, в результате чего при переходе к тестовой выборке качество классификации значительно ухудшалось. Это наводит на мысль учитывать похожесть форм кластеров обучения и контроля в процессе построения признакового пространства. Таким образом, указано одно из направления дальнейших исследований — разработка различных критериев похожести форм двух множеств и встраивание их в генетический алгоритм.

Заключение

В настоящей дипломной работе:

- Предложен метод генерации признаков в задаче классификации сигналов, основанный на генетической оптимизации.
- Выполнена программная реализация предложенного метода на ЭВМ в виде библиотеки для системы MATLAB.
- Проведены вычислительные эксперименты на реальных данных двух прикладных задач: Brain-Computer Interface [10] и Ford Classification Challenge [26]. Предложенный метод показал приемлемый результат работы. Сделанные выводы подробно изложены в работе.
- Работа была представлена на XVI международную научную конференцию “Ломоносов — 2009”.
- Работа была опубликована [9].

Список литературы

[1] John R. Koza. Genetic Programming IV: Routine Human-Competitive Machine Intelligence // Springer. 2005. pp590.

[2] William B. Langdon, Riccardo Poli. Foundations of genetic programming // Springer. 2002. pp260.

[3] Генетические алгоритмы — математический аппарат
http://www.basegroup.ru/library/optimization/ga_math/

[4] Генетические алгоритмы
http://ru.wikipedia.org/wiki/Генетические_алгоритмы

[5] Логистическая регрессия и ROC-анализ — математический аппарат
http://www.basegroup.ru/regression/logistic.htm —

[6] Воронцов. К.В. Лекции по метрическим алгоритмам классификации. 2007
http://www.ccas.ru/voron/

[7] Дьяконов А.Г. Об одном подходе к решению задач из области BCI // Сборник докладов XII Всеросийской конференции ММРО–12. Москва. ООО МАКС Пресс. 2005. С. 95–97.

[8] Дьяконов А.Г. Анализ кластерных конфигураций в одной проблеме фильтрации спама // Математические методы распознавания образов: 13-я Всеросийская конференция: Сборник докладов. М.: МАКС Пресс. 2007. С. 476-478.

[9] Власова Ю.В. Применение генетических алгоритмов в задаче классификации сигналов (приложение в BCI) // Сборник тезисов XVI международной научной конференции Ломоносов — 2009. М.: МАКС Пресс. 2009. С. 17.
http://lomonosov-msu.ru/rus/archive.html

[10] http://ida.first.fraunhofer.de/projects/bci/competition_iii/ — BCI competition 2003

[11] Blankertz B., Muller K.-R., Curio G., Vaughan T.M., etc. The BCI competition 2003: Progress and perspectives in detection and discrimination of EEG single trials. // IEEE Trans. Biomed. Eng., 51(6):1044-1051, 2004.
http://ida.first.fhg.de/publications/BlaMueCurVauSchWolSchNeuPfuHinSchBir04.pdf

[12] Blankertz B., Curio G., Muller K.-R. Classifying single trial EEG: Towards brain computer interfacing. // In Advances in Neural Inf. Proc. Systems (NIPS 01), T.G. Diettrich, S. Becker, Z. Ghahramani, Eds., 2002, vol. 14, pp. 157–164.
http://ida.first.fhg.de/publications/BlaCurMue02.pdf

[13] Muller K.-R., Anderson C. W., Birch G. E. Linear and non-linear methods for brain-computer interfaces. // IEEE Trans Neural Sys Rehab Eng, 11(2):165-169, 2003.
http://ida.first.fhg.de/publications/MueAndBir03.pdf

[14] K.-R. Muller, M. Krauledat, G. Dornhege, G. Curio, Benjamin B. Machine learning techniques for brain-computer interfaces. // Biomed Tech, 49(1):11-22, 2004.
http://ida.first.fhg.de/publications/MueKraDorCurBla04.pdf

[15] R. Krepki, G. Curio, B. Blankertz, K.-R. Muller. Berlin brain-computer interface — the HCI communication channel for discovery. // Int J Hum Comp Studies, 65:460-477, 2007. Special Issue on Ambient Intelligence.
http://ida.first.fhg.de/publications/KreCurBlaMue07.pdf

[16] http://ida.first.fraunhofer.de/bbci/index_en.html — Berlin Brain-Computer Interface

[17] http://www.biocybernaut.com/about/brainwaves/

[18] http://www.biocybernaut.com/ — The Biocybernaut Institute — Center for Advanced Brain Wave Biofeedback Training using Neurofeedback

[19] http://mmspl.epfl.ch/page33695.html

[20] U. Hoffmann, J.-M. Vesin, T. Ebrahimi. Recent Advances in Brain-Computer Interfaces // IEEE International Workshop on Multimedia Signal Processing (invited Paper), 2007.

[21] http://en.wikipedia.org/wiki/Brain-computer_interface — Brain-Computer Interface

[22] J. Vidal. Toward Direct Brain-Computer Communication // In Annual Review of Biophysics and Bioengineering. L.J. Mullins, Ed., Annual Reviews, Inc., Palo Alto, Vol. 2, 1973, pp. 157-180.

[23] J. Vidal. Real-Time Detection of Brain Events in EEG // In IEEE Proceedings, May 1977, 65-5:633-641.

[24] Jorge Baztarrica Ochoa. EEG Signal Classification for Brain–Computer Interface Applications. // Diploma, The University of Plymouth, 2002

[25] http://web.vrn.ru/wavelet/ — Вейвлетный анализ медико-биологических сигналов

[26] http://home.comcast.net/~nn_classification/ — M. Abou-Nasr, L. Feldkamp. Ford Classification Challenge.

[27] http://www.mathworks.com/