Министерство образования и науки Российской Федерации

Федеральное государственное автономное образовательное учреждение высшего профессионального образования

«Уральский федеральный университет имени первого Президента России Б.Н. Ельцина»

Кафедра интеллектуальных информационных технологий

Оценка работы \_\_\_\_\_\_\_\_\_\_

Руководитель от УрФУ \_\_\_\_\_\_\_\_\_\_

**ПОСТРОЕНИЕ ПРИЗНАКОВ ДЛЯ КЛАССИФИКАЦИИ МЕДИЦИНСКИХ ИЗОБРАЖЕНИЙ ПОСРЕДСТВОМ ГЕНЕТИЧЕСКОГО ПРОГРАММИРОВАНИЯ**

ОТЧЕТ

по научно-исследовательской практике

Руководитель от предприятия \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Студент \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Группа \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**ОГЛАВЛЕНИЕ**

[Основные сведения о компании 3](#_Toc472581843)

[ВВЕДЕНИЕ 4](#_Toc472581844)

[Информация о приложении 5](#_Toc472581845)

[Концепция 5](#_Toc472581846)

[Функционал приложения 6](#_Toc472581847)

[Использованные инструменты разработки 7](#_Toc472581848)

[PyCharm 7](#_Toc472581849)

[Использованные технологии 9](#_Toc472581850)

[Git 9](#_Toc472581851)

[GitHub 10](#_Toc472581852)

[Trello 10](#_Toc472581853)

[lxml 11](#_Toc472581854)

[Логирование информации 12](#_Toc472581855)

[Основные сведения 12](#_Toc472581856)

[Практическое применение 13](#_Toc472581857)

[ЗАКЛЮЧЕНИЕ 14](#_Toc472581858)

[СПИСОК ЛИТЕРАТУРЫ 15](#_Toc472581859)

# ВВЕДЕНИЕ

В последнее время одним из актуальных направлений развития компьютерных технологий в медицине становится обработка цифровых изображений. Медицинские изображения – это структурно-функциональный образ органов человека, предназначенный для диагностики заболеваний и изучения анатомо-физиологической картины организма. Распознавание патологических процессов является одной из наиболее важных задач обработки и анализа медицинских изображений.

Классификация заключается в прогнозировании значения категориального атрибута (класса) на основе значений других атрибутов. Цель классификации – взять объект, оцениваемый набором признаков, и назначить его одному дискретному классу. Каждый экземпляр данных определяется в соответствии с набором атрибутов или переменных.

Для решения задачи классификации выполняется определенный вид предварительной обработки, известной как преобразование представления, которое, учитывая исходный вектор признаков *F0* и обучающее множество *L*, состоит в создании представления *F*, полученного из *F0*, которое максимизирует некоторый критерий, и, по крайней мере, так же хорошо, как *F0* по отношению к этому критерию.

Разработка хорошего пространства признаков является предпосылкой для достижения высокой производительности в любой задаче машинного обучения. Однако часто неясно, каким должно быть оптимальное представление признака. В результате общий подход заключается в том, чтобы использовать все доступные атрибуты в качестве признаков и оставить проблему идентификации полезных наборов признаков модели обучения. Такой упрощенный подход не всегда работает хорошо. С развитием технологий аппаратного и программного обеспечения и увеличением объема данных количество признаков, используемых системами машинного обучения, измеряется десятками тысяч и миллионами признаков.

Для решения проблем нерелевантности признаков и взаимодействия признаков используются методы построения признаков. Конструирование признака включает в себя преобразование заданного набора входных признаков для создания нового набора более мощных признаков, которые затем используются для прогнозирования. Поскольку вновь созданные признаки учитывают взаимодействия в предыдущем пространстве признаков, они более значимы и приводят к более кратким и точным классификаторам.

В данной работе применяется генетическое программирование для автоматического конструирования признаков с целью повышения различающей эффективности классификатора.

Актуальность выбранной темы обосновывается тем, что проблема классификации патологических процессов по данным медицинских изображений не может быть правильно решена, если важные взаимодействия и отношения между оригинальными признаками, не принимаются во внимание. Таким образом, многие исследователи согласились, что выделение признаков является наиболее важным ключом к любому распознаванию образов и проблеме классификации. В большинстве случаев выделение признаков осуществляется человеком на основе знаний исследователя, опыта и / или интуиции.

Проблема. Эффективность работы классификатора сильно зависит от входного множества признаков. Как выбрать оптимальное множество признаков для классификатора?

Цель – нахождение и отработка методики построения признаков для решения задачи классификации.

Для достижения поставленной цели необходимо выполнить следующие задачи:

* изучить соответствующую литературу;
* разработать алгоритм построения признаков;
* реализовать программу по данному алгоритму;
* оценить эффективность работы алгоритма и сравнить с классической классификацией;
* обобщить полученные результаты и сделать соответствующие выводы.

Объектом исследования выступает классификация изображений. Предметом исследования является алгоритм построения признаков для классификации.

# Постановка задачи

Конструирование признаков может применяться для достижения двух различных целей: уменьшения размерности данных и улучшения показателей прогнозирования. В данной работе будет рассматриваться использование построения признаков для увеличения эффективности классификации.

## 1.1 Формальные определения

Пусть:

1. *x* ∈ *X*, где *X* – область входных значений.
2. *y* ∈ *Y*, где *Y* – область выходных значений.
3. *S* – набор обучающих примеров таких, что
4. *S'* – набор тестовых примеров таких, что
5. *Errh* будет ошибкой гипотезы *h* по сравнению с истинной основной гипотезой *hT*.

Мы можем рассматривать каждый *x* как вектор фиксированной длины значений признаков в исходном пространстве признаков, т.е.

,

где *fi* ∈ *F0* ∀ *i* = 1…*n* и *n* = |*F0*|.

Цель любой обучающей машины – узнать предиктор из *S*, с маленькой ошибкой *Errh*. В парадигме построения признака каждый исходный вектор признаков *x* преобразуется в новый вектор признаков

,

где *φi* ∈ *FT* ∀ *i* = 1...*N* и *N* = |*FT*|.

Каждое преобразованное значение признака *φi(x)* получается путем оценки некоторой функции по всем исходным *fi*. Мы хотим вывести гипотезу , предполагая, что ее истинная ошибка *Errh'* меньше *Errh*. В большинстве практических сценариев *Errh* и *Errh'* будут вычисляться путем измерения производительности *h* и *h'* на тестовом множестве *S'*.

## Требования к системе

Необходимо разработать алгоритм, который решает задачу построения множества признаков для эффективной классификации медицинских изображений методом генетического программирования. Создаваемый алгоритм должен удовлетворять следующим критериям:

* на основе входных данных в виде множества оригинальных признаков, обучающего и тестового множеств исходных медицинских изображений, набора классификаторов, а также максимально возможной погрешности выдавать результат в виде множеств сконструированных признаков, которые обеспечивают точность классификации не меньше заданной.
* иметь возможность изменения параметров генетического программирования – мощность начальной популяции, максимальная глубина дерева, вероятности скрещивания, репродукции и мутации;
* быть удобной в использовании, т.е. предоставлять промежуточные и конечные результаты работы в удобном для пользователя виде.

# Обзор литературы

## 2.1 Общее описание построения признаков

Концептуально любой метод построения признака можно рассматривать как выполнение следующих действий:

1. Выбор начального пространства признаков *F0* (ручное построение признаков).
2. Преобразование *F0* для построения нового пространства признаков *FN* (преобразование признаков).
3. Выбор подмножества признаков *Fi* из *FN* (выбор признака).
   1. Определение полезности *Fi* для задачи классификации.
   2. Если некоторые критерии завершения достигнуты:
      1. Вернитесь к шагу 3.
   3. Иначе множество *FT* = *Fi*.
4. *FT* – это сконструированное пространство признаков.

Исходное пространство признаков *F0* состоит из признаков, созданных вручную, которые часто кодируют некоторые базовые знания области. Различные методы построения признаков отличаются тем, как они реализуют каждый из этих этапов. Ясно, что тремя важными аспектами любого метода построения признаков являются: метод трансформации, метод выбора подмножества признаков *Fi* и критерий полезности для подмножества признаков.

### 2.1.1 Преобразование признаков

Общим подходом к созданию преобразованных признаков является применение набора операторов (например, {+, –, \*}) к исходным значениям признаков. Выбор операторов основан на знании области и типе признаков. К числу наиболее часто используемых операторов относятся:

1. Логические функции: конъюнкция, дизъюнкция, отрицание и т. д.
2. Номинальные характеристики: декартово произведение и т. д.
3. Численные характеристики: минимум, максимум, сложение, вычитание, умножение, деление, среднее, эквивалентность, неравенство и т. д.

Помимо этого, гиперплоскости, логические правила и битовые строки также могут использоваться для создания новых пространств признаков. Операторы обычно применяются итеративно. Поэтому каждый новый признак *φi* ∈ *FN* можно представить, используя дерево операторов и исходные значения признаков, как показано на рисунке 1.



Рисунок 1 – Древовидное представление сконструированного признака *φi* ∈ *FN*

### 2.1.2 Выбор признаков

Выбор признаков является важным шагом в процессе построения признака. Так как преобразованное пространство признаков *FN* велико, нам нужно выбрать подмножество *FT* из *FN*. Проблема выбора оптимального подмножества является NP трудной, и методы обычно выполняют какой-то неоптимальный жадный поиск. Используемые критерии для измерения полезности пространства признаков *Fi* часто включают в себя получение информации, коэффициент корреляции, точность предсказания на некотором множестве проверки и т. д.

В литературе было представлено множество различных методов отбора (см. [Guyon and Elisseeff, 2003] и [Forman, 2003]). Мы можем свободно классифицировать эти методы по двум категориям: фильтры и обертки [Kohavi and John, 1997].

Фильтрующие методы выбирают подмножества признаков независимо от предиктора. Они, по существу, действуют как этап предварительной обработки данных, прежде чем обучить предиктор. В эту категорию входят подходы с переменным ранжированием, которые включают ранжирование отдельных признаков с использованием теоретико-информационных или корреляционных критериев, а затем построение подмножества с высокими показателями выигрыша. Фильтры имеют преимущество в том, что они быстрее, чем обертки. Более того, они, как правило, обеспечивают общий порядок построения признаков, не настроенный на конкретный метод обучения. Однако недостатком является то, что выбранное подмножество может быть не самым подходящим для конкретного классификатора.

Оберточные методы выбора признаков используют метод обучения, который будет использоваться для прогнозирования как черный ящик для выбора подмножеств признаков. Эти методы обычно делят обучающий набор на набор обучения и проверки (набор тестов является отдельным). Для любого заданного подмножества признаков предиктор обучается набору обучения и тестируется на наборе проверки. Точность прогнозирования на наборе проверки рассматривается как оценка подмножества признаков. Таким образом, мы в конечном счете хотели бы выбрать подмножество с наивысшей оценкой. Из-за повторяющихся циклов обучения и тестирования для каждого подмножества признаков, обертки имеют тенденцию быть намного более вычислительно затратными по сравнению с фильтрами. Обычно цель состоит в том, чтобы пропустить пространство признаков таким образом, чтобы число проверяемых подмножеств было сведено к минимуму. Очевидным преимуществом является то, что выбранное подмножество настроено на предиктор.

## Обзор аналогов

Задача построения соответствующих признаков часто очень специфична для области применения и трудоемка. Таким образом, создание автоматизированных методов построения признаков, требующих минимальных усилий пользователя, является сложной задачей. В частности, необходимы методы, которые:

1. Создают набор признаков, которые помогут улучшить точность прогнозирования.
2. Вычислительно эффективны.
3. Являются обобщаемыми для разных классификаторов.
4. Позволяют легко добавлять знания области.

Был предложен ряд различных методов. В следующих подразделах мы классифицируем эти методы на основе методик, которые они используют для определения и поиска пространства признаков. Ранние методы в основном базировались на деревьях решений, в то время как последние подходы были в большей степени сосредоточены на индуктивном логическом программировании и генетическом программировании. Методы на основе генетического программирования являются гибкими в операторах, которые они могут использовать, в то время как основанные на индуктивном логическом программировании методы позволяют легко внедрять знания из разных источников.

### Деревья решений

Один из ранних алгоритмов построения признаков принадлежит Пагалло [Pagallo, 1989], создателю FRINGE, который адаптивно увеличивал первоначальный набор атрибутов для изучения концепции дизъюнктивной нормальной формы (ДНФ). На каждой итерации новые признаки строятся путем комбинирования пар признаков в существующем пространстве признаков с использованием операторов отрицания и конъюнкции. Поскольку данные операторы являются полным набором булевых операторов, общее пространство признаков состоит из всех булевых функций исходных признаков. Чтобы справиться с проблемой чрезвычайно большого пространства новых признаков, FRINGE объединяет только пары признаков, которые появляются на краю каждой из положительных ветвей дерева решений. Процесс создания признака повторяется до тех пор, пока не будут созданы новые признаки.

CITRE [Matheus and Rendell, 1989] и DC Fringe [Yang et al., 1991] – два других алгоритма построения признака на основе деревьев решений. Алгоритмы используют конъюнкции и дизъюнкции для объединения различных операндов, таких как корень (выбирает первые два признака каждой положительной ветви), границу (подобно FRINGE), корневую границу (комбинация корня и границы), смежную (выбирает все соседние пары вдоль каждой ветви) и все (всё вышеперечисленное).

Проблема с алгоритмами на основе деревьев решений заключается в том, что с тех пор, как новые признаки добавляются в пространство признаков на каждой итерации, количество входных признаков, которые необходимо передать в алгоритм построения дерева решений, становится очень большим, делая процесс вычислительно неэффективным. В результате в каждой итерации некоторые низко оцениваемые признаки отбрасываются. Помимо использования обычных методов обрезки дерева решений, все признаки, которые не использовались при построении дерева решений, также могут быть исключены.

Все методы, рассмотренные ранее, использовали только логические операторы для генерации признаков. Чтобы разработать более гибкий подход, Маркович и Розенштейн [Маркович и Розенштейн, 2002] представили FICUS, общую структуру построения признаков. Их подход обеспечивает набор функций (операторов) конструктора вместе с исходным набором признаков и примерами для алгоритма построения признаков. Затем алгоритм обогащает исходное пространство признаков добавлением дополнительных перспективных признаков. Функции конструктора могут быть либо одним, либо несколькими обычно используемыми операторами, либо могут поставляться некоторым экспертом предметной области.

Вход в FICUS задается с использованием «языка спецификаций признаков» (ЯСП). Пользователь может предоставить информацию о типе (например, номинальном, непрерывном и т.д.), области и диапазоне основных признаков и функций конструктора. Также пользователь может дополнительно указать набор булевых ограничений на тип признаков, которые могут быть сгенерированы или использованы с определенными функциями. Таким образом, новое пространство признаков представляет собой набор всех признаков, которые могут быть сгенерированы на основе ЯСП. Обход пространства поиска осуществляется с помощью четырех разных операторов:

1. Композиция. Один или два признака используются в качестве входных данных. На их основе вычисляется новый набор признаков с использованием всех допустимых функций.
2. Вставка. На вход подаются два признака. Новый признак получается путем вставки одного признака в другой.
3. Замена. Из двух начальных признаков формируется новый признак с помощью замены какого-либо компонента одного признака на компонент другого признака.
4. Интервал. Из одного исходного признака получаются новые логические признаки, проверяющие, лежит ли начальный признак в каком-либо определенном диапазоне.

Стратегия поиска является вариантом лучевого поиска. На каждом шаге поддерживаются два набора признаков: текущий набор признаков и предыдущий набор признаков, который создал текущий набор через применение четырех указанных выше операторов. Сохранение предыдущего набора позволяет системе выполнить один уровень обратного отслеживания. Полезность набора признаков вычисляется на основе размера и сложности дерева решений, которое генерируется по множеству примеров. Полезность отдельных признаков в наборе вычисляется с использованием критериев разделения (получения информации). На каждой итерации используется лучший набор признаков для создания нового набора признаков. Авторы показали, что их метод достиг значительного прироста производительности в разных областях и классификаторах.

Обладая высокой гибкостью, с одной стороны, у FICUS было два потенциальных недостатка, с другой стороны. Во-первых, их критерии выбора подмножества признаков не учитывали взаимодействия признаков, что приводило к несколько узкому поиску в пространстве признаков. Во-вторых, как обсуждалось ранее, при заданной проблеме выбор операторов часто неясен, что затрудняет пользователям предоставление правильного набора функций конструктора.

### Индуктивное логическое программирование

Индуктивное логическое программирование (ИЛП) используется для разработки описаний предикатов из примеров и базовых знаний. Методы построения признаков на основе ИЛП могут обеспечить обобщенную структуру для включения знаний области в процесс создания признаков. Первое использование предикатов первого порядка в качестве признаков было применено в программе LINUS [Lavrac et al., 1991]. В последующей литературе проблема идентификации хороших признаков с представлением первого порядка была рассмотрена в пропозициональных (основанных на характеристиках) подходах [Kramer et al., 2000].

В работах Specia et. Al. [Specia et al., 2007; 2009] использовался подход, основанный на ИЛП, для повышения точности прогнозирования в задаче определения значения слова. Основная идея заключается в применении двухэтапного подхода:

1. Конструирование признака. ИЛП используется для изучения нового набора конъюнктивных признаков.
2. Выбор признака. Выбор подмножества признаков на основе их полезности в процессе прогнозирования.

Их система изучает предложения вида

,

где .

– это конъюнкция над некоторыми значениями признака экземпляра *x* и вычисляется как TRUE (1), если конъюнкцией является, иначе – FALSE (0). Предложения *hi* можно рассматривать как правила вида: «Если какой-либо признак *φ(x)* = 1, то экземпляр *x* должен принадлежать классу *c*. Когда такие правила выводятся из набора примеров, пространство признаков обогащается добавлением всех индивидуальных конъюнкций *φi* в качестве признаков. Таким образом, они используют ИЛП для идентификации подмножества конъюнкций из пространства всех возможных конъюнкций исходных признаков. Дополнительным преимуществом использования ИЛП является то, что конъюнктивные предложения *hi* также могут быть получены из источников, отличных от тренировочных примеров. Это позволяет легко встраивать знания области в процесс построения признака. После создания исходного набора признаков, они затем используют метод выбора признаков в соответствии с подходом обертки, о котором говорилось выше. На каждой итерации оценивается производительность прогнозирования набора признаков и генерируются два новых набора признаков. Один, исключая худший признак, а другой, добавляя наиболее эффективный признак в другой случайный образец признаков из исходного набора признаков. Полученная модель показала впечатляющий прирост производительности в задаче разрешения лексической многозначности.

### Аннотации

Подходы, основанные на аннотациях, позволяют пользователям предоставлять знания области в виде аннотаций вместе с примерами обучения. Затем на основе этих аннотаций будет изучено пространство признаков. Таким образом, исключается необходимость в определении операторов.

Ротт и Смолл [Roth и Small, 2009] предложили интерактивный протокол построения пространства признаков. Вместо того, чтобы предопределять большое пространство признаков с помощью операторов, их подход позволяет создавать динамическое пространство признаков, основанное на взаимодействии между обучающей машиной и экспертом предметной области. Во время учебного процесса, когда учащийся представляет какой-то пример эксперту по предметной области, эксперт использует знания области, чтобы предоставить дополнительные аннотации (а не только ярлыки). Учащийся теперь должен включить дополнительные знания через модификации пространства признаков и изучить модель. Таким образом, эксперт области может напрямую кодировать знания об области без необходимости определения операторов. Они применили свой метод к проблеме распознавания именованного объекта при использовании семантически связанных списков слов (СССС) [Fellbaum, 1998; Pantel and Lin, 2002] в качестве внешнего знания. СССС группируют вместе лексические элементы, принадлежащие к одной семантической категории. Например, Направление\_Компаса = {восток, запад, север, ... и т.д.}. Протокол работает следующим образом:

1. Учащийся начинает с изучения первоначальной гипотезы *h* в исходном пространстве признаков *F0* и оценки всех экземпляров, использующих его.
2. Затем функция запроса *Q* используется для выбора лучших экземпляров и представления их эксперту области. Цель состоит в том, чтобы выбрать *Q* таким образом, чтобы он минимизировал взаимодействие с пользователем при максимальном воздействии.
3. Эксперт рассматривает экземпляры и отмечает некоторые признаки для абстракции. Например, на рисунке 2 ученик ошибается при маркировке «Chicagoland» как организации.
4. Эксперт отмечает признаки, относящиеся к «западу» для абстракции.
5. Основываясь на этом взаимодействии, пространство признаков изменяется путем замены признаков *φeast*, *φwest*, *φnorth* в исходном пространстве признаков *F0* на *φCompass*-*Direction*= *φeast* ∨ *φwest* ∨*φnorth*



Рисунок 2 – Пример работы протокола СССС

Рисунок 2 демонстрирует работу протокола. Все лексические элементы с членством в СССС обведены квадратом. Элементы, используемые для неверного предсказания для «Chicagoland», имеют двойное обведение. Эксперт может выбрать любой обведенный элемент для проверки СССС.

Другой интерактивный метод принадлежит Рагхавану и Аллену [Raghavan and Allan, 2007], которые представляют тандемный алгоритм обучения для выбора признаков для классификации текста. Алгоритм начинается с небольшого количества помеченных экземпляров и на каждой итерации рекомендует экземпляры и признаки для людей для маркировки. Однако в отличие от СССС, в их случае пространство признаков остается статичным.

Другие методы, которые позволяют эксперту области напрямую указывать информацию об пространстве признаков с помощью аннотаций, включают [Хуан и Митчелл, 2006; Зайдан и Эйснер, 2007 год; Druck et al., 2008; Зайдан и Эйснер, 2008 год; Lim et al., 2007]. В частности, Лим и др. [Lim et al., 2007] предлагают метод построения признаков, основанный на объяснительном обучении, для задачи распознавания китайских символов. В их случае знания кодировались в виде «штрихов», которые использовались для генерации символов (или экземпляров). Затем признаки генерировались на основе наличия / отсутствия одинаковых штрихов в разных классах..

# ЗАКЛЮЧЕНИЕ

Достоинством использования метода генетического программирования для решения задачи установления функциональная зависимости данных является точность совпадения полученной функции в точках выборки. Если у нас есть дополнительная информация о функции кроме ее точек, мы легко можем использовать ее при применении операции репродукции. Концептуальная простота генетического программирования также является важным преимуществом его практического применения.

В ходе прохождения практики была разработана система ведения журнала событий программного комплекса. Полученные данные позволили провести анализ состояний системы, привели к улучшению работы программы. В связи с этим можно считать разработанную систему успешно внедренной в программный комплекс.

# СПИСОК ЛИТЕРАТУРЫ

1. Koza, John R. Genetic programming: on the programming of computers by means of natural selection. MIT Press, Cambridge, MA, USA, 1992.
2. Langdon, William B. Genetic programming and data structures. Department of Computer Science, University College London, 1996.
3. Mitchell Melanie. An introduction to Genetic Algorithms. MIT Press, Cambridge, London, 1998.
4. Xinjie Yu, Mitsuo Gen. Introduction to Evolutionary Algorithms. Springer, London Limited 2010.
5. Библиотека lxml [Электронный ресурс]. URL: http://lxml.de/ (дата обращения – 01.09.2016)