Московский Государственный Университет им. М. В. Ломоносова
Механико-математический факультет
Кафедра вычислительной математики

Деветьяров Дмитрий
группа 502

.

Дипломная работа

Использование эволюционных алгоритмов для построения деревьев решений при классификации биологически активных молекул.

Научный руководитель
д. ф.-м. н.
Кумсков М. И.

Москва, 2006

В работе предложен, проанализирован и реализован новый метод решения задачи «структура-свойство» (QSAR-задачи) для молекулярных графов в применении к конкретной выборке молекул амбровых одорантов с целью определения их биологической активности/неактивности.

По завершении этапов вычисления множества особых точек на молекулярных поверхностях, формирования алфавита дескрипторов и построения матрицы «молекула-дескриптор» был разработан и применен новый метод поиска функциональной зависимости биологической активности от значений дескрипторов в виде дерева решений.

Проект реализован в системе Matlab 7.0. При этом использовались как основной пакет программ, так и дополнительный пакет (Toolbox) Statistics, предлагающий широкий выбор инструментов для статистического исследования. Проект состоит из порядка 15 модулей-процедур.

Проведены вычислительные эксперименты на выборке молекул амбровых одорантов, состоящей из 50 молекул. Из них 37 являются биологически активными, 13 - неактивными. Обработано две матрицы «молекула-признак», сформированных на основе разных наборов дескрипторов численностью 703 и 231 соответственно. В результате, на одной из них удалось выделить два крупных кластера из 28 и 10 элементов с хорошей прогностической оценкой на скользящем контроле 78,6% и 80% соответственно.

Кроме того, была реализована обратная связь между этапом формирования дескрипторов и этапом поиска функциональной зависимости. После нахождения информативных 3D - дескрипторов предыдущего уровня реализована возможность построения на их основе 3D- дескрипторов следующего уровня.

Оглавление.

Введение.

Одним из наиболее интенсивно развивающихся в настоящее время направлений молекулярного моделирования является поиск взаимозависимостей между структурами химических соединений и их свойствами посредством построения математических моделей. Данная методология получила название QSAR, что означает Quantitative Structure Activity Relationships. Полученные модели используются для скрининга молекулярных баз данных, поиска новых потенциально активных веществ. В развитых странах работы в области QSAR ведутся постоянно возрастающими темпам, так как применение методов QSAR при создании новых соединений с заданными свойствами позволяет значительно сократить затраты на скрининг и осуществлять более целенаправленный синтез соединений, обладающих заданным набором свойств. Так QSAR-задача обеспечивает подход к решению ОСНОВНОЙ задачи химии - синтезу химических соединений с определенными нужными свойствами без лишних затрат времени и денег.

В нашем случае была предложена выборка амбровых одорантов[1] со значениями их биологической активности/неактивности с целью поиска дерева решений, строящих функциональную зависимость между структурой и активностью. В основу было решено положить известный Метод Группового Учета Аргументов.

Стандартные методы построения классификаторов молекулярных графов обладают рядом недостатков. Причем, степень их эффективности зависит от конкретной выборки объектов. Так и Метод Группового Учета Аргументов «в чистом виде» далеко не всегда дает хорошие результаты. Основные проблемы этого алгоритма – неоднородность выборки и большое время ее обработки. Эти проблемы решаются в данной работе следующими методами:

- нормирование столбцов матрицы и результирующего вектора с целью устранения влияния абсолютного значения признака и степени его вариации на результаты проведения алгоритма

[1] Низкомолекулярные органические соединения, обладающие амбровым запахом

- проведение иерархического кластерного анализа для выявления «сгустков однородности» с последующим удалением из выборки «выбросов», мешающих выявить функциональную зависимость на основных кластерах
- фильтрация выборки для отсеивания близких к константным признаков на каждой селекции МГУА
- фильтрация выборки для отсеивания признаков, сильно коррелирующих с остальными, на каждой селекции МГУА
- использование наиболее информативных структурных 3D-дескрипторов низшего уровня для отбора 3D-дескрипторов высшего уровня

Кроме того, известные реализации некоторых из вышеописанных методов имеют ряд недостатков:

- при проведении селекций МГУА и при последующей оценке результатов не учитывается тот факт, что результирующий вектор состоит только из 0 и 1
- реализованный в системе Matlab метод иерархического кластерного анализа неудобен тем, что строит разбиение на заданное число кластеров; такое разбиение может оказаться неэффективным, если большинство полученных кластеров будут мелкими
- после построения классификаторов и выявления информативных дескрипторов не реализована возможность возврата от дескрипторов к 3D-формам, возможность визуализации фрагментов молекулярного графа, чьи дескрипторы вошли в классификатор

Для решения вышеописанных проблем и исправления недостатков были предложены, программно реализованы и применены к выборке амбровых одорантов новые модификации уже известных алгоритмов поиска функциональной зависимости.

При этом в комплексе использованы различные методы распознавания образов. Сначала для выявления информативных дескрипторов и сокращения длины обрабатываемой матрицы был реализован Метод Группового Учета Аргументов (МГУА). На новой матрице был запущен иерархический кластер-анализ, в результате чего вся выборка была поделена на два больших кластера.

Далее на каждом кластере был снова запущен МГУА и построена классифицирующая функция. Чтобы оценить полученный результат, на выбранных информативных столбцах запускался скользящий контроль с линейной регрессией и выдавался процент совпадений исходных и полученных значений биологической активности.

Особенностью полученной реализации является то, что, исходя из качества полученного прогноза, формируются рекомендации по изменению параметров детализации, использованных на этапе описания молекул и формирования дескрипторов. К ним, в частности, относятся параметры разбиения интервала значений расстояний и электрического заряда, а также сложность формируемых дескрипторов. Далее весь алгоритм запускается заново уже с новыми параметрами детализации. Таким образом реализуется обратная связь между этапом описания и этапом построения классификаторов.

Научная новизна работы:

- В работе предложены способы построения дерева решений при поиске функциональной зависимости с помощью оптимизации работы метода группового учета аргументов в применении к конкретной выборке
- Успешно программно реализованы новые модификации алгоритмов
- Проведены вычислительные эксперименты и построены деревья решений для прогнозирования амбрового запаха с хорошей прогностической оценкой
- Производится построение и обработка дескрипторов, основанных на трехмерном графе, в вершинах которого находятся не атомы, а особые точки (точки локальных экстремумов, геометрических или физико-химических свойств) на молекулярных поверхностях

В первой главе данной работы приведена постановка общей задачи «структура-свойство» (QSAR-задачи) как частного случая из области распознавания образов. Вторая глава содержит постановку конкретной задачи, определяет этапы ее решения и описывает те из них, которые предшествуют поиску функциональной зависимости биологической активности от значений дескрипторов. И наконец, в третьей главе раскрывается этап построения классификаторов, рассматриваются применяемые методы и схемы реализованных алгоритмов. В четвертой главе описаны особенности программной реализации проекта, а также полученные результаты по построению классификатора. Сам текст программ, наряду с форматами и пояснениями, приведен в Приложении.

Глава 1. Общая постановка задачи «структура-свойство» (QSAR-задачи) для молекулярных графов.

План:

*)Распознавание образов с «учителем»

- обзор методов РО («РО»,Журавлев, Рязанов, Сенько) диплом Пиманихина

*) QSAR-задача как частный случай РО: (диплом Захарова)

- общая постановка задачи, понятия, термины (обзор по Similarity)
- основные стадии QSAR-задачи
- стадия описания:
  (а) дескрипторы, основные понятия
  (б) виды дескрипторов
  (в) обзор топологических дескрипторов (обзор по Similarity)
  (г) обзор 2D-дескрипторов (обзор по Similarity, Бибигон)
- стадия анализа и построения модели функциональной зависимости
  (а) применение различных методов РО (линейные, кластерный анализ, МГУА, эволюционные методы)
  (б) построение линейной прогнозирующей функции (пересекается с п. 3.1)

*) Выводы:

- QSAR – частный случай РО
- могут быть применены все алгоритмы распознавания образов

Параллельно вводятся основные определения и термины:

- молекулярный граф
- инвариант
- дескриптор
- молекулярный фрагмент
- …

Глава 2. Наша постановка задачи, основные определения и термины.

План:

*) Постановка нашей задачи, что дано, что хотим
- описание данной выборки: примеры структурных формул + 3D-поверхности
- общая схема: поиск описания, адаптированный (по сложности) к свойствам; стремимся к минимальной сложности описания

Основной подход: поиск описания молекулярных графов, адекватных для заданной активности -> обратная связь:

Параметры детализации:
(а) дискретизация (разбиение на интервалы значений расстояний/зарядов)
(б) сложность анализируемых фрагментов

Параметры описания (запротоколированы) -> результаты ->
рекомендации по детализации -> новые параметры описания

Здесь: блок-схема

*) Этапы задачи

*) Этап формирования ОТ (подробности у Вики !!!)
- что и в каком виде дано (3D-представление (.mol, .sdf) + заряд)
- построение молекулярной поверхности с зарядами, ее триангуляция
- нахождение особых точек на молекулярной поверхности

Здесь: рисунки молекулярных поверхностей

*) Этап формирования дескрипторов и построения матрицы
- основные определения, термины
- описание построения структурных 3D-дескрипторов
- требования к структуре (обработка каждой молекулы отдельно, возврат от дескрипторов к конкретным особым точкам)

*) Сводная таблица деятельности по вышеописанным этапам (от Светы!!!)

*) Выводы: что на входе моего анализа

Глава 3. Этап построения дерева решений в виде линейных классификаторов.

В данной главе дано подробное описание этапа анализа матрицы «структура-свойство» и поиска дерева решений в виде линейных классификаторов. Здесь рассматриваются известные применяемые методы, приводятся их модификации, наряду с предпосылками к их использованию и нюансами реализации, а также даны схемы реализованных алгоритмов.

3.1. Постановка задачи.

Имея в наличии матрицу из значений сформированных дескрипторов для каждой молекулы выборки, перейдем к решению основной задачи - нахождению метода распознавания новых поступающих на вход молекулярных графов и прогнозированию его свойств. Необходимо найти некоторую зависимость, функцию $\varphi = \varphi(x_1, x_2, \dots, x_m)$ (где $x_1, x_2, \dots, x_m$ – значения отобранных дескрипторов), которая приближала бы свойства объектов, представленных молекулярными графами.

Чаще всего для нахождения таких зависимостей используются линейные функции φ. Поэтому в данной работе за основу взят **линейный вид классификаторов**. В результате для решения задачи построения прогнозирующей функции необходимо решить систему линейных уравнений

$$X \times \bar{a} = \bar{y}$$

где X – матрица N x M (N – количество объектов обучающей выборки, M – количество выявленных признаков объектов), a – искомый вектор коэффициентов линейной функции, y – вектор истинных свойств объектов (биологической активности). Как показано в предыдущей главе, матрица получается очень «широкой», то есть $M >> N$ (N = 50, M = 231 и 703). Можно, конечно, решать эту систему и находить зависимости для таких матриц полным перечислением, но при этом возникает несколько трудностей:

- Технически сложно держать и работать с таким количеством памяти.
- В стандартных пакетах анализа данных модулей для обработки таких «широких» таблиц отсутствует.

Для преодоления этих трудностей зачастую используют **генетический алгоритм Метода Группового Учета Аргументов**, позволяющий динамически подбирать самые существенные для прогноза столбцы матрицы, не рассматривая остальные столбцы. Именно его мы применяем для нахождения линейной функциональной зависимости. Подробное описание данного алгоритма приводится в п. 3.2.

Чтобы найденная зависимость значения активности от значений дескрипторов была точнее, будем ее искать в виде дерева решений.

Деревья решений – это способ представления правил в иерархической, последовательной структуре, где каждому объекту соответствует единственный узел, дающий решение.

Под правилом понимается логическая конструкция, представленная в виде «если … то …».

<иллюстрация, блок-схема>!!!

Область применения деревья решений в настоящее время широка, но все задачи, решаемые этим аппаратом, могут быть объединены в следующие три класса:

- описание данных, хранение информации
- классификация данных
- регрессия, численное прогнозирование

В нашем случае мы хотим построить дерево решений для классификации обрабатываемых векторов из значений дескрипторов и последующего поиска функциональной зависимости отдельно на каждом из классов. Иными словами, перед нами стоит цель разбить выборку на классы, внутри которых искомая функциональная зависимость, предположительно, общая, а уже затем для каждого из классов найти свой классификатор.

Далее перед нами встает вопрос, как выделить искомые классы элементов, сходные по функциональной зависимости. Логично использовать следующую «гипотезу компактности»: близкие по функциональной зависимости объекты будут близки как точки в некотором многомерном пространстве в некоторой фиксированной метрике (координатами точек выступают соответствующие вектора значений дескрипторов). Для выделения подобных классов обычно применяют методы **кластерного анализа**.

Кластер-анализ представляет собой совокупность методов и подходов к разбиению исходного множества объектов на определенное число классов (*кластеров*, «сгустков»), причем внутри кластера должны содержаться «близкие» друг другу объекты. Обычно методы кластерного анализа используются при распознавании без учителя, т.е. нет необходимости задавать значения внешнего признака, а только описание объектов в виде векторов чисел-признаков. Отсюда, кстати, следует наглядная геометрическая интерпретация кластерного анализа: в *n*-мерном метрическом пространстве объекты изображаются в виде точек; кластеры же – это совокупности близких (в некоторой фиксированной метрике) точек. Выбор метрики не накладывает жестких ограничений на решение задачи и в конечном итоге зависит лишь от природы проблемы и от предпочтений человека-эксперта. Обычно выбирается евклидова метрика.

В нашей реализации был выбран метод иерархического кластерного анализа с центроидальным алгоритмом объединения кластеров и евклидовой метрикой. Подробное описание этого и других методов кластерного анализа приводится в п. 3.3.

Таким образом, для решения поставленной задачи необходимо научиться из полученных на этапах описания молекул и формирования дескрипторов векторов (признаков) объектов исходной выборки создавать правила, по которым новый объект будет, во-первых, относится к одному из найденных выше кластеров; во-вторых, сопоставлен некоторой его характеристике - числовому признаку (биологической активности/неактивности). Первая подзадача решается очевидным образом, т.к. имея метрику, определяющую близость свойств объектов (близость точек в метрическом пространстве), можно легко определить, к какому классу лучше отнести новый объект (или же отказаться от объекта, если он далек от всех классов выборки). Вторая же подзадача в данной работе решена модификациями Метода Группового Учета Аргументов (МГУА), которые будет описаны и проанализированы ниже.

В результате, получаем следующую краткую схему алгоритма.

<блок-схема>

3.2. Метод Группового Учета Аргументов.

3.2.1. Обзор метода, его преимущества.

Метод Группового Учета Аргументов применяется в самых различных областях для анализа данных и отыскания знаний, прогнозирования и моделирования систем, оптимизации и распознавания образов. Индуктивные алгоритмы МГУА дают уникальную возможность автоматически находить взаимозависимости в данных, выбрать оптимальную структуру модели или сети, и увеличить точность существующих алгоритмов.

Этот подход самоорганизации моделей принципиально отличается от обычно используемых дедуктивных методов. Он основан на индуктивных принципах - нахождение лучшего решения основано на переборе всевозможных вариантов.

При помощи перебора различных решений подход индуктивного моделирования пытается минимизировать роль предпосылок автора в результатах моделирования. Алгоритм сам определяет структуру модели и законы, действующие в объекте. Он может быть использован как советчик для отыскания новых решений в проблемах искусственного интеллекта.

Метод Группового Учета Аргументов состоит из нескольких алгоритмов для решения разных задач. В него входят как параметрические, так и алгоритмы кластеризации, комплексирования аналогов, ребинаризации и вероятностные алгоритмы. Этот подход самоорганизации основан на переборе постепенно усложняющихся моделей и выборе наилучшего решения согласно минимуму внешнего критерия. В качестве базисных моделей используются не только линейные функции, как в нашем случае, но и полиномы, а также нелинейные, вероятностные функции или кластеризации. В частности, в конце работы предлагается использовать модификации МГУА для логических и вероятностных функций.

Направление МГУА может быть полезным по следующим причинам:

- Находится оптимальная сложность структуры модели, адекватная уровню помех в выборке данных. (Для решения реальных проблем с зашумленными или короткими данными, упрощенные прогнозирующие модели оказываются более точными.)

- Количество слоев и нейронов в скрытых слоях, структура модели и другие оптимальные параметры нейросетей находятся автоматически.
- Гарантируется нахождение наиболее точной или несмещенной модели - метод не пропускает наилучшего решения во время перебора всех вариантов (в заданном классе функций)
- Любые нелинейные функции или воздействия, которые могут иметь влияние на выходную переменную, используются как входные параметры
- Автоматически находятся интерпретируемые взаимосвязи в данных и выбираются эффективные входные переменные
- Переборные алгоритмы МГУА довольно просто запрограммировать
- Метод использует информацию непосредственно из выборки данных и минимизирует влияние априорных предположений автора о результатах моделирования
- Подход МГУА используется для повышения точности других алгоритмов моделирования
- МГУА дает возможность отыскания несмещенной физической модели объекта (закона или кластеризации) - одной и той же для всех будущих выборок.

3.2.2. Общая схема алгоритма МГУА.

Пусть заданы:

- матрица «объект-признак» X, состоящая из N строк и M столбцов;
- *результирующий вектор* - вектор y, который необходимо «предсказать» по матрице X;
- класс функций C для построения классифицирующей функции;
- *функционал качества* $\varphi(F, X, y)$, $F \in C$;
- количество отбираемых на каждом этапе наилучших функций Q;

- условие остановки алгоритма (максимальное число используемых переменных в классификаторе или предельный уровень функционала качества).

Тогда эволюционный алгоритм МГУА состоит из следующих этапов:

1) Перебором строим всевозможные функции $F \in C$ от одной переменной (столбцов матрицы X);

2) Оцениваем множество всех построенных на данный момент функций при помощи функционала качества φ и выбираем из них Q наилучших (данный набор функций, отобранных на определенном этапе, называется *селекцией*), причем для каждого шага число Q может иметь отдельное значение;

3) Перебором присоединяем к каждой из отобранных функций новую переменную (в рамках класса функций C), формируем классифицирующие функции новой селекции;

4) Проверяем, достигнуто ли условие остановки алгоритма; если оно не достигнуто, переходим к п. 2).

<блок-схема>

Таким образом, алгоритмы МГУА воспроизводят схему массовой селекции. На каждой селекции классифицирующие модели усложняются и отбираются лучших из них. Прогнозирующая функция (*прогнозирующая модель*) $\varphi = \varphi(x_1,x_2,x_3,...,x_m)$, где φ — некоторая элементарная функция, в нашем случае линейная, заменяется несколькими рядами "частных" описаний:

1-ый ряд селекции: $y_1 = f(x_1, x_2)$, $y_2 = f(x_1, x_3),..., y_s = f(x_{m-1}, x_m)$,

2-ой ряд селекции: $z_1 = f(x_1, y_1)$, $z_2 = f(x_1, y_2),..., z_p = f(x_m, y_s)$, где s=m*(m-1)/2, p=s*m и т.д.

Сложность комбинаций на каждом ряду обработки информации возрастает (как при массовой селекции), пока не будет получена единственная модель оптимальной сложности.

Каждое частное описание является функцией только двух аргументов. Поэтому его коэффициенты легко определить по данным обучающей выборки. Исключая промежуточные переменные (если это удается), можно получить "аналог" полного описания.

Ряды селекции наращиваются до тех пор, пока повышается значение классифицирующей функции φ. Как только достигнут минимум ошибки, селекцию следует остановить. Практически селекцию останавливают даже несколько раньше достижения полного минимума, как только ошибка начинает падать слишком медленно. Это приводит к более простым и более достоверным уравнениям.

3.2.3. Случай линейных классифицирующих функций.

В нашем случае C представляет собой класс линейных функций от нескольких переменных. Тогда на каждом шаге селекции мы строим линейные прогностические модели на основе уже построенных добавлением одной переменной, а затем из них с помощью функционала качества отбираем лучшие. Когда же условие остановки алгоритма будет выполняться, рассматривается лучшая по функционалу качества прогностическая модель, в которой переходя обратно от последних селекций к первой, раскрывая скобки, можно получить прогностическую линейную функцию φ от большого числа переменных. Для полного определения алгоритма необходимо предоставить правило построения новой функции от двух переменных (см. шаг 3) и функционал качества.

Алгоритм построения линейной функции от двух переменных.

В качества данного алгоритма используется **линейная регрессия**.

<….описание….>

Линейную регрессию можно использовать и для построения линейных функций от большого числа переменных, т.е. было возможно сразу построить линейную функцию однократным запуском линейной регрессии на всей матрице. Но…..????

???? Как показано в предыдущей главе, матрица получается очень «широкой», то есть M >> N (N = 50, M = 231 и 703). Можно, конечно, решать эту систему и находить зависимости для таких матриц полным перечислением, но при этом возникает несколько трудностей:

- *Технически сложно держать и работать с таким количеством памяти.*

- *В стандартных пакетах анализа данных модулей для обработки таких «широких» таблиц отсутствует.*

????

Функционалы качества.

(а) Обычно в случае линейного классификатора роль функционала качества играет величина **среднеквадратической ошибки** – нормированная сумма квадратов отклонений полученного прогноза от исходного результирующего вектора :

<формула R^2 = …>

В результате проведения селекций значение R^2 увеличивается. С одной стороны, необходимо получить значение R^2, близкое к единице (когда ошибка минимальна), с другой – как можно более простую прогнозирующую модель (т. е. на более раннем этапе селекции).

(б) С целью определения прогностической способности модели используют метод **скользящего контроля (cross-validation)**. Значение среднеквадратической ошибки отображает качество прогнозирования на данной выборке, но ничего не говорит о прогностической способности модели при добавлении новой переменной. Среднеквадратическая ошибка, вычисленная с применением метода скользящего контроля, R^2_{CV} (нижний индекс «CV» – от «cross validation») позволяет это сделать.

Значение R^2_{CV} считается следующим образом:

1) Переходя обратно от последней селекции к первым и раскрывая скобки в полученной модели, получим линейную прогнозирующую модель от нескольких переменных. Выделим переменные, задействованные в модели, и оставим в исходной матрице X только им соответствующие столбцы. В результате получим матрицу Z размерностью N x M_{new}, где N – первоначальное число элементов выборки, M_{new} - число задействованных в прогнозирующей модели переменных.

2) Далее перебираем все строки матрицы Z. Для каждой строки под номером $1 \le i \le N$ проводим следующую процедуру:

(а) Из полученной матрицы Z и результирующего вектора y сформируем новую матрицу Z' и новый вектор y' путем вычеркивания i-ой строки z_i из Z и i-ого элемента y_i из y.

(б) На сокращенных матрице и векторе (размерностью (N - 1) x M_{new} и (N - 1) x 1 соответственно) проведем линейную регрессию. В качестве приближения получим вектор-строку b_i длины M_{new}.

(в) Вычислим ошибку $\varepsilon_{i,CV}$ прогноза вектором b_i линейной зависимости вычеркнутой строки z_i от вычеркнутого элемента y_i:

$\varepsilon_{i,CV} = | y_i - z_i \times b_i |$

3) Вычисляем значение среднеквадратической ошибки по той же формуле, беря вместо ошибок ε_i ошибки скользящего контроля $\varepsilon_{i,CV}$:

<формула R^2 CV= …>

В то время как при проведении селекции значение R^2 неуклонно растет и приближается к единице, R^2_{CV} в некоторый момент, не доходя до единицы, достигает максимума и начинает уменьшаться. Именно это и является показателем, что селекцию необходимо остановить.

(в) Иногда в качестве данного показателя используется **коэффициент корреляции**.

<формула, описание>????

3.3. Иерархический кластерный анализ.

Из классических методов распознавания образов в работе используется кластерный анализ. В данной части приводится описание его основных методов, их параметров и модификаций.

3.3.1. Краткий обзор кластерного анализа.

Кластерный анализ - это совокупность методов, позволяющих классифицировать многомерные данные, которые описываются определенным набором исходных переменных. Целью кластерного анализа является образование групп схожих между собой объектов, которые принято называть кластерами (родственные понятия, используемые в литературе, - класс, таксон, сгущение).

Кластерный анализ приводит к разбиению на группы с учетом всех группировочных признаков одновременно, т.е. они учитываются все сразу при отнесении наблюдения в ту или иную группу. При этом, как правило, не указаны

четкие границы каждой группы, и зачастую неизвестно заранее, сколько же групп целесообразно выделить в исследуемой совокупности.

Результатом проведения кластерного анализа является получение групп сходных объектов. Различные методы кластерного анализа позволяют получать кластеры, различающиеся по размеру и форме.

Методы кластерного анализа представляют собой очень эффективное средство статистической обработки данных, позволяющее решать следующие задачи:

- изучение структуры совокупности с целью выделения групп объектов, схожих между собой по нескольким признакам, отражающим сущность, природу объектов
- проверка выдвигаемых предположений о наличии некоторой структуры в изучаемой совокупности объектов, т.е. поиск существующей структуры;
- построение новых классификаций для слабоизученных явлений, когда необходимо установить наличие связей внутри совокупности и попытаться привнести в нее структуру
- снижение размерности признакового пространства перед проведением корреляционно-регрессионного анализа без потери информации о взаимосвязи между переменными

Выделяют различные методы кластерного анализа:

a) *агломеративные* (в начале алгоритма каждый элемент представляет собой отдельный кластер, а затем происходит пошаговое объединение имеющихся кластеров в более крупные до момента выполнения условия завершения алгоритма);
b) *дивизимные* (в данном случае, наоборот, в начале алгоритма все элементы объединены в один кластер, далее происходит пошаговое деление одного из кластеров на два меньших кластера);
c) *итеративные* (остальные, например, метод *k*-средних)

В работе применен *иерархический кластерный анализ*, самый распространенный из агломеративных методов. Для проведения любого из методов нужно ввести общие понятия, такие как мера сходства и критерий качества, о чем и пойдет речь в следующих главах.

3.3.2. Меры сходства.

Для проведения классификации необходимо ввести понятие сходства объектов по наблюдаемым переменным. В каждый кластер должны попасть объекты, имеющие сходные характеристики.

В кластерном анализе для количественной оценки сходства вводится понятие метрики. Сходство или различие между классифицируемыми объектами устанавливается в зависимости от метрического расстояния между ними. Если каждый объект описывается *k* признаками, то он может быть представлен как точка в *k*-мерном пространстве, и сходство с другими объектами будет определяться как соответствующее расстояние. Чем больше расстояние, тем меньше сходство, и наоборот. В кластерном анализе используются различные меры расстояния между объектами:

1) евклидово расстояние;

<формула>

2) взвешенное евклидово расстояние

<формула>

Вопрос о придании переменным соответствующих весов должен решаться после проведения исследователем тщательного анализа изучаемой совокупности и сущности классифицирующих переменных. Веса задаются пропорционально степени важности переменных.

3) расстояние city-block (Хэммингово расстояние, или L_1-норма)

<формула>

4) расстояние Минковского

<формула>

5) расстояние Махаланобиса

<формула>

Если в числе классификационных признаков встречаются качественные или квазиколичественые характеристики (например, кодовые обозначения объектов), то необходимо произвести оцифровку этих признаков по определенному правилу, т.е. заменить значения признаков числовыми метками. В нашем случае такой необходимости нет, так как все признаки представлены числовыми значениями.

Выбор меры расстояний и весов для классифицирующих переменных – очень важный этап кластерного анализа, так как от этих процедур зависят состав и количество формируемых кластеров.

Если алгоритм кластеризации основан на измерении сходства между переменными, то в качестве мер сходства могут использованы:

- линейные коэффициенты корреляции
- коэффициенты ранговой корреляции
- коэффициенты контингенции и т. д. [ссылка на эту книгу]

В зависимости от типов исходных переменных выбирается один из видов показателей, характеризующих близость между ними.

3.3.3. Методы иерархического кластерного анализа.

Из всех методов кластерного анализа самыми распространенными являются иерархические агломеративные методы. Сущность этих методов заключается в том, что на первом шаге каждый объект выборки рассматривается как отдельный кластер. Процесс объединения кластеров происходит последовательно: на основании матрицы расстояний или матрицы сходства объединяются наиболее близкие объекты. Если матрица сходства первоначально имеет размерность m x m, то полностью процесс кластеризации завершается за m – 1 шагов, в итоге все объекты будут объединены в один кластер. Последовательность объединения легко поддается геометрической интерпретации и может быть представлена в виде графа-дерева (дендрограммы). На дендрограммах, полученных в результате наших опытов, указываются номера объединяемых объектов и расстояние, при котором произошло объединение.

<дендрограмма>

Множество методов иерархического кластерного анализа различается не только используемыми мерами сходства (различия), но и алгоритмами классификации. Из них наиболее распространены метод одиночной связи, метод полных связей, метод средней связи, метод Уорда.

Метод одиночной связи.

Алгоритм образования кластеров следующий: на основании матрицы сходства (различия) определяются два наиболее схожих или близких объекта, они и образуют первый кластер. Таким объектом будет тот, который имеет наибольшее сходство хотя бы с одним из объектов уже включенных в кластер.

При совпадении данных на основании одинаковых мер сходства (различия) будет идти образование сразу нескольких кластеров.

К достоинствам этого метода следует отнести нечувствительность алгоритма к преобразованиям исходных переменных и его простоту. Недостатками являются необходимость постоянного хранения матрицы сходства и невозможность определения по результатам кластеризации, сколько же кластеров можно образовать в исследуемой совокупности объектов.

Метод полных связей.

Включение нового объекта в кластер происходит только в том случае, если расстояние между объектами не меньше некоторого заданного уровня.

Метод средней связи.

Для решения вопроса о включении нового объекта в уже существующий кластер вычисляется среднее значение меры сходства, которое затем сравнивается с заданным пороговым уровнем.

Если речь идет об объединении двух кластеров, то вычисляются расстояния между их центрами и сравнивают ее с заданным пороговым значением.

Метод Уорда.

Данный метод предполагает, что на первом шаге каждый кластер состоит из одного объекта. Первоначально объединяются два ближайших кластера. Для них определяются средние значения каждого признака и рассчитывается сумма квадратов отклонений V_k:

$V_k = \Sigma_{i=1}^{n(k)} \Sigma_{j=1}^{p} (x_{ij} - x_{jk\,cp})^2$, где

k – номер кластера,

i – номер объекта,

j – номер признака,

p – количество признаков, характеризующих каждый объект,

$n(k)$ – количество объектов в k-ом кластере.

В дальнейшем на каждом шаге работы алгоритма объединяются те объекты или кластеры, которые дают наименьшее приращение величины V_k. Метод Уорда приводит к образованию кластеров приблизительно равных размеров с минимальной внутрикластерной вариацией. В итоге все объекты оказываются объединенными в один кластер. При больших объемах исходной совокупности это приводит к значительным затратам машинного времени и требует больших объемов памяти для вычислений.

3.3.4. Алгоритм иерархического кластерного анализа.

Алгоритм иерархического кластерного анализа можно представить в виде последовательных процедур:

1) Значения исходных переменных нормируются одним из вышеперечисленных способов.
2) Рассчитывается матрица расстояний или матрица мер сходства (в нашей реализации это делает функция *pdist*, реализованная в MatLab)
3) Находится пара самых близких кластеров. По выбранному алгоритму объединяются эти два кластера. Новому кластеру присваивается меньший из номеров объединяемых кластеров.
4) Процедуры 2, 3 и 4 повторяются до тех пор, пока все объекты не будут объединены в один кластер или заданное количество кластеров либо до достижения заданного порога сходства.

3.3.5. Меры сходства для объединения двух кластеров.

Меры сходства для объединения двух кластеров п. 3 определяются четырьмя методами:

Метод «ближайшего соседа» - степень сходства оценивается по степени сходства между наиболее схожими (ближайшими) объектами этих кластеров.

Метод «дальнего соседа» - степень сходства оценивается по степени сходства между наиболее отдаленными (несхожими) объектами кластеров.

Метод средней связи – степень сходства оценивается как средняя величина степеней сходства между объектами кластеров.

Метод медианной связи – расстояние между любым кластером S и новым кластером, который получился в результате объединения кластеров p и q, определяется как расстояние от центра кластера S до середины отрезка, соединяющего центры кластеров p и q.

Если процесс объединения в иерархическом кластерном анализе проводится вручную, то оценка сходства может быть дана либо визуально по матрице сходств (расстояний), либо на основании значений значений d_{US}:

$d_{US} = \alpha_p d_{ps} + \alpha_q d_{qs} + \beta\, d_{pq} + \gamma |\, d_{ps} - d_{qs}|$,

где d_{ps}, d_{qs}, d_{pq} - расстояния между соответствующими кластерами;

α_p, α_q, β, γ - параметры, учитывающие особенности конкретного алгоритма кластеризации данных; их значения приведены в таблице ниже. В формулах таблицы n_p, n_q – количество объектов в соответствующих кластерах.

Константные значения параметров, учитывающие особенности метода кластеризации данных.

Метод	**Параметры**
Ближайшего соседа	$\alpha_p = \alpha_q = 0.5$, $\beta = 0$, $\gamma = -0.5$ – расстояние, $\gamma = 0.5$- подобие
Дальнего соседа	$\alpha_p = \alpha_q = 0.5$, $\beta = 0$, $\gamma = 0.5$ – расстояние, $\gamma = -0.5$ - подобие
Средней связи	$\alpha_p = n_p / (n_p + n_q)$, $\alpha_q = n_q / (n_p + n_q)$, $\beta = \gamma = 0$
Центроидный	$\alpha_p = n_p / (n_p + n_q)$, $\alpha_q = n_q / (n_p + n_q)$, $\beta = - n_p * n_p / (n_p + n_q)^2$
Медианной связи	$\alpha_p = \alpha_q = 0.5$, $\beta = -0.25$, $\gamma = 0$

Использование различных алгоритмов объединения в иерархических агломеративных методах приводит к различным кластерным структурам и сильно влияет на качество проведения кластеризации. Поэтому алгоритм должен выбираться с учетом имеющихся сведений о существующей структуре совокупности наблюдаемых объектов или с учетом требований оптимизации математических критериев.

Выбор алгоритма классификации во многом зависит от принимаемого критерия качества разбиения на классы.

Если алгоритм иерархического кластерного анализа реализуется в программе для ЭВМ. То оценка сходства между кластерами дается только на основании значений d_{US}.

3.3.6. Критерии качества классификации.

При использовании различных методов кластерного анализа для одной и той же совокупности могут быть получены различные варианты разбиения. Существенное влияние на характеристики кластерной структуры оказывают: во-первых, набор признаков, по которым осуществляется классификация, во-вторых, тип выбранного алгоритма. Например, иерархические и итеративные

методы приводят к образованию различного числа кластеров. При этом сами кластеры различаются и по составу, и по степени близости объектов. Выбор меры сходства также влияет на результат разбиения. Если используются методы с эталонными алгоритмами, например, метод k-средних, то задаваемые начальные разбиения в значительной степени определяют конечный результат разбиения.

После завершения процедур классификации необходимо оценить полученные результаты. Для этой цели используется некоторая мера качества классификации, которую принято называть функционалом или критерием качества. Наилучшим по выбранному функционалу следует считать такое разбиение, при котором достигается экстремальное (минимальное или максимальное) значение целевой функции – функционала качества.

В большинстве случаев алгоритмы классификации и критерии качества связаны между собой, т. е. определенный алгоритм обеспечивает получение экстремального значения соответствующего функционала качества. Например, использование метода Уорда приводит к получению кластеров с минимальной внутриклассовой дисперсией.

Рассмотрим наиболее распространенные функционалы качества:

1. Сумма квадратов расстояний до центров классов.

<формула> См. Пиманихин с. 23

2. Сумма внутриклассовых расстояний между объектами

<формула> См. Пиманихин с 23

В этом случае наилучшим следует считать такое разбиение, при котором достигается минимальное значение *Q*, т. е. получены кластеры большой «плотности». Объекты, попавшие в один кластер, близки между собой по значениям тех переменных, которые использовались для классификации.

3. Суммарная внутриклассовая дисперсия.

<формула> См. Пиманихин с 23

В данном случае разбиение, при котором сумма внутриклассовых дисперсий будет минимальной, следует считать оптимальной.

Если оценивать качество разбиения по степени удаленности кластеров друг от друга, то можно использовать функционал средних межклассовых расстояний:

<формула> $= \Sigma d_{ij}$ (i из S_q j из S_l) / $\Sigma_{l<q} n_l n_q$ - max

Судить о качестве разбиения позволяют и некоторые простейшие примеры. Например, сравнение средних значений признаков в отдельных кластерах со средним значением в целом по всей совокупности объектов. Если отличие групповых средних от общего среднего значения существенное, то это может являться признаком хорошего разбиения. Оценка существенности различий может быть выполнена с помощью t-критерия Стьюдента.

Перечисленные выше способы оценки качества разбиения предполагают чисто формальный подход и являются для исследователя только вспомогательными средствами. Основная роль принадлежит содержательному анализу результатов классификации. Выбор лучшего варианта разбиения облегчается в значительной мере серьезной подготовительной работой, в частности выбором признаков, характеризующих классифицируемые объекты, откуда вытекает необходимость предварительного отбора значимых для всей выборки дескрипторов. В зависимости от количества признаков, их взаимосвязи, выбранного масштаба измерения подбирается наиболее подходящий алгоритм классификации, задаются начальные параметры разбиения. Все это облегчает интерпретацию разбиения и позволяет судить о его качестве с точки зрения поставленной задачи.

3.4. Описание предложенных модификаций.

Ознакомившись с классическими вариантами применяемых нами методов, перейдем к описанию используемых модификаций. Рассмотрим отдельно каждое изменение в алгоритме, уделяя внимание предпосылкам к их введению, а также нюансам реализации.

3.4.1. Проведение иерархического кластерного анализа перед МГУА.

Цель всей работы состоит в нахождении линейной зависимости значений биологической активности от вектора признаков (значений дескрипторов). Как показано в п. 3.1, необходимо решить систему линейных уравнений

$$X \times \bar{a} = \bar{y},$$

где X – очень широкая матрица (количество столбцов матрицы на порядок больше количества строк). В результате возникают трудности с решением

данной системы (см. п. 3.1). Но их можно преодолеть, используя **эволюционный алгоритм Метода Группового Учета Аргументов**, который позволяет отобрать самые информативные для данной активности признаки. Поэтому за основу при построении классификаторов в работе берется именно МГУА (см. п. 3.2).

Далее, опираясь на гипотезу, что «сходные» векторы признаков могут иметь «сходную» функциональную зависимость, хотим выделить классы элементов выборки, «сходных» по своим векторам. В итоге мы хотим построить дерево решений для классификации обрабатываемых векторов из значений дескрипторов и последующего поиска функциональной зависимости отдельно на каждом из классов. Иными словами, перед нами стоит цель разбить выборку на классы, внутри которых искомая функциональная зависимость, предположительно, общая, а уже затем для каждого из классов найти свой классификатор.

Далее перед нами встает вопрос, как выделить искомые классы элементов, «сходные» по структуре векторов признаков.

Если исходные данные представлены в форме матрицы X, то эти точки являются непосредственным геометрическим изображением многомерных наблюдений $x^{(1)}\ x^{(2)}\ ...,\ x^{(n)}$ в n-мерном пространстве $П^n\ (X)$ с координатными осями $Ox^{(1)},\ Ox^{(2)}\ ...,\ Ox^{(n)}$. Тогда проблема классификации состоит в разбиении анализируемой совокупности точек — наблюдений на сравнительно небольшое число (заранее известное или нет) классов таким образом, чтобы объекты, принадлежащие одному классу, находились бы на сравнительно небольших расстояниях друг от друга. Для этого мы используем метод кластерного анализа (см. п. 3.3).

Таким образом, сначала мы проводим на выборке кластерный анализ, разделяя выборку на близкие классы, а затем на каждом кластере отдельно находим функциональную зависимость Методом Группового Учета Аргументов.

3.4.2. Особенности реализованного иерархического кластерного анализа.

В работе используется иерархический агломеративный метод кластерного анализа. Метрика для меры сходства и метод объединения двух кластеров являются параметрами, которые можно задать произвольно. Но в проекте задействованы обычная евклидова метрика и центроидальный метод

объединения кластеров (т.е. расстояние между кластерами определяется, как расстояние между их центрами тяжести).

Реализованный в системе Matlab метод иерархического кластерного анализа неудобен тем, что строит разбиение на заданное число кластеров; такое разбиение может оказаться неэффективным, если большинство полученных кластеров будут мелкими. Мы же хотим получить разбиение на заданное число кластеров (в нашем случае этот параметр равен 2) достаточно большой мощности, т.е. содержащих не менее заданного числа элементов (в реализации – переменная *clustmin*). Естественно, желательно получить наибольшее значение переменной *clustmin*. Поэтому иерархический кластерный анализ реализован в работе следующим образом:

1) для поиска оптимального значения переменной *clustmin* запускается цикл по этой переменной от достаточно большого числа с шагом (-1);

2) в теле цикла запускаем новый цикл по переменной *clusti* – количество кластеров разбиения – от 2 до максимального числа кластеров (равного числу элементов выборки);

3) в теле второго цикла строим разбиение на *clusti* кластеров и считаем количество кластеров мощности не меньшей *clustmin*;

4) если таких кластеров не меньше двух (вместо 2 можно использовать другое значение), то выходим из обоих циклов, оптимальное разбиение найдено; в противном случае, продолжаем работу во внутреннем цикле.

В результате получаем разбиение на два достаточно крупных кластера. Оставшиеся элементы выборки, назовем их «выбросы», мы не анализируем, так как они составляют меньшую часть выборки и, внося помехи в общие данные, препятствуют нахождению функциональной зависимости.

3.4.3. Реализация МГУА на полной выборке для определения метрики иерархического кластерного анализа.

Как отмечалось ранее, значительное влияние на характеристики кластерной структуры и на качество полученного разбиения оказывает набор признаков, по которым осуществляется классификация. И обычно выбор лучшего варианта разбиения облегчается в значительной мере серьезной подготовительной работой, в частности, выбором признаков, характеризующих классифицируемые объекты.

В нашем случае имеется огромное количество признаков (на порядок превышающее число объектов-молекул). И ясно, что далеко не все из них играют информативную роль в функциональной зависимости. Поэтому перед применением иерархического метода кластерного анализа следует выделить наиболее информативные признаки, по которым он будет проводиться, т. е. те, которые войдут в метрику – меру сходства. Для определения информативных признаков применяем все тот Метод Группового Учета Аргументов, только на полной выборке объектов.

Также следует отметить, что набор отобранных столбцов не отразится на МГУА на отдельных кластерах. Последующее применение МГУА к строкам, соответствующим элементам кластера, проводится на полном наборе признаков, т.е. включая и столбцы, отброшенные в результате первого запуска МГУА на полной выборке.

Количество признаков *Q*, задействованных в прогнозирующей модели, определяется как *N / 5*, где *N* – число строк обрабатываемой матрицы. Таким образом, для первого применения МГУА на полной выборке из 50 молекул этот параметр равен 10, для МГУА на отдельных кластерах он вычисляется исходя из размера кластера.

3.4.4. Фильтрация на сильно коррелирующие и близкие к константным столбцы.

Вышеописанный алгоритм требует проведения алгоритма МГУА на достаточно широкой матрице. Так как селекции МГУА основаны на переборе всех полученных прогнозирующих моделей и всех имеющихся переменных, проведение расчетов занимает много времени. Поэтому с целью уменьшить затраты времени и памяти при переборе на каждом шаге селекции, а также перед первой селекцией, проводится фильтрация полученных прогнозирующих моделей, каждая из которых представлена в виде вектора прогнозируемых значений активности:

1) сначала отсеиваем столбцы, близкие к константным:

Вводится порог дисперсии значений свойств объектов (близкий к 0). Если дисперсия столбца, соответствующего признаку, меньше данного порога, то свойство отбрасывается.

2) затем отсеиваем признаки, сильно коррелирующих с остальными:

Вводится порог корреляции значений свойств объектов (близкий к 1). Если для данного свойства есть другое, такое что корреляция соответствующих столбцов больше данного порога, то свойство отбрасывается. В программной реализации эти значения задаются как параметры. По умолчанию для первого ряда селекции порог корреляции равен 0.99, для остальных – 0.85.

3.4.5. Нормирование столбцов перед запуском алгоритма.

Оценка сходства между объектами при проведении иерархического кластерного анализа сильно зависит от абсолютного значения признака и от степени его вариации в совокупности. Чтобы устранить подобное влияние на процедуру кластеризации, можно значения исходных переменных нормировать одним из следующих способов:

1) $z_{ij} = (x_{ij} - x_{j\,cp}) / \sigma_j$, где σ_j – среднеквадратичное отклонение j-ого столбца

2) $z_{ij} = x_{ij} / x_{max\,j}$

3) $z_{ij} = x_{ij} / x_{cp\,j}$

4) $z_{ij} = x_{ij} / x_{min\,j}$.

В нашей работе применен первый из способов нормировки, причем, нормируются как столбцы матрицы X, так и результирующий вектор y. Нормирование происходит один раз перед запуском всего алгоритма, коэффициенты $x_{j\ cp}$ и σ_j сохраняются и используются вновь по завершении алгоритма для получения корректных коэффициентов лучших линейных моделей-классификаторов.

3.4.6. Функционал качества, адаптированный к логическому формату прогнозируемого вектора.

Стандартный алгоритм Метода Группового Учета Аргументов не учитывает тот факт, что прогнозируемый столбец имеет логический формат (что соответствует биологической активности и неактивности). Поэтому он состоит из двух значений: 0 и 1.

Предлагается воспользоваться этим свойством при проведении МГУА на этапе отбора прогнозирующих моделей, а также при оценке найденного лучшего классификатора. Для этого вместо стандартной среднеквадратичной ошибки R^2 введем новый функционал качества, оценивающий прогностическую достоверность модели.

Пусть X и y – матрица значений дескрипторов и прогнозируемый вектор, b – прогнозирующая линейная модель (представляет собой вектор-строку коэффициентов). Тогда предсказываемое значение вектора y равно $X * b = y'$. Заметим, что вектор логического формата в данном случае прогнозируется вектором из действительных значений. Логично было бы преобразовать действительные значения в логический формат. Таким образом, по вектору y' строится его «округление» y'_{round} : для каждого из значений преобразуется к ближайшему из значений 0 или 1. И наконец, вычисляется процент совпадений значений элементов векторов y и y'_{round}.

Заметим, что предшествующее нормирование столбцов нисколько не препятствует использованию нового функционала качества. Прогнозируемый вектор все так же состоит из 2 значений, на этот раз $(- y_{cp}) / \sigma_y$ (соответствует 0) и $(1 - y_{cp}) / \sigma_y$ (соответствует 1), где σ_y – среднеквадратичное отклонение столбца y. И округление y' будет происходить в сторону ближайшего из этих значений.

Следует отметить, что использование описанного функционала качества дало значительное улучшение прогностического качества полученных классификаторов.

3.4.7. Реализация обратной связи между этапами формирования дескрипторов и поиска классификаторов.

Примененный способ формирования структурных 3D-дескрипторов позволяет строить не только дескрипторы, соответствующие двойкам и тройкам особых точек, но и дескрипторы более высоких порядков. Но возникает проблема при их использовании на этапе построения классификаторов. Уже при минимальном числе интервалов дискретизации расстояний между особыми точками на дескрипторах четвертого порядка получаются очень широкие матрицы. Это делает практически невозможным проведение расчетов Методом Группового Учета Аргументов вследствие невероятных по масштабу требуемых затрат времени. Кроме того, в целом нецелесообразно перебирать всевозможные четверки, пятерки и т. д. особых точек. Поэтому предлагается строить дескрипторы более высокого порядка на основе уже отобранных наиболее информативных дескрипторов меньшего порядка.

Алгоритм состоит в следующем:

1) формируем структурные 3D-дескрипторы 2-ого порядка (двойки особых точек), из них строим векторы признаков и матрицу признаков;

полученную матрицу признаков подаем на алгоритм построения классификаторов в виде дерева решений;

2) если в результате получили прогностически устойчивую модель зависимости, останавливаемся на данном шаге, так как целью работы является получение одновременно и достаточно простой, и достаточно точной модели;

в противном случае формируем алфавит структурных 3D-дескрипторов 3-его порядка, но не полным перебором всевозможных вариантов, а только тех, что получены добавлением к отобранным дескрипторам 2-ого порядка еще одной особой точки;

аналогично, полученную матрицу признаков подаем на алгоритм построения классификаторов в виде дерева решений;

3) повторяем шаг 2, строя дескрипторы n-ого порядка на основе лучших дескрипторов (n - 1)-ого порядка, до тез пор пока не получим хорошей прогностической способности модели.

<блок-схема>

3.5. Полная схема предложенного алгоритма.

В итоге в работе на этапе построения классификаторов предложено реализовать следующий алгоритм:

1) нормирование столбцов матрицы и результирующего вектора, сохранение коэффициентов нормирования;

2) первоначальная фильтрация столбцов матрицы на близкие к константным и сильно коррелирующие между собой столбцы;

3) запуск МГУА на полной выборке с фильтрацией прогнозирующих моделей на близкие к константным и коррелирующие между собой на каждом шаге селекции, а также с функционалом качества, адаптированным к логическому формату прогнозируемого вектора; последующее удаление из матрицы неинформативных столбцов;

4) запуск иерархического кластерного анализа на получившейся матрице;

5) на каждом кластере отдельный запуск МГУА с последующим удалением неинформативных для данного кластера столбцов;

6) для каждого кластера вычисление коэффициентов классифицирующей функции и номеров информативных столбцов (в том числе с учетом первоначального нормирования столбцов);

7) запуск скользящего контроля (основанного на линейной регрессии) и вычисление качества линейной модели с помощью функционала качества, адаптированного к логическому формату прогнозируемого вектора.

<блок-схема>

3.6. Выводы.

На основе стандартных методов распознавания образов путем их модификации был предложен алгоритм построения дерева решений – линейных классификаторов. Внесенные изменения позволяют преодолеть трудности проведения расчетов и улучшить качество прогнозирующих моделей, в том числе и за счет адаптации к конкретной задаче.

Кроме того, предложенный алгоритм позволяет использовать обратную связь между этапами описания, формирования дескрипторов и построения классификаторов: результаты предложенного алгоритма можно использовать для формирования нового алфавита дескрипторов с возможно лучшей прогностической оценкой.

Глава 4. Программная реализация предложенного алгоритма решения задачи «структура-свойство». Ее особенности. Результаты вычислений.

4.1. Архитектура программы.

4.1.1. Вопрос выбора среды программирования: Matlab или C/C++.

При реализации вышеописанных методов возникали следующие сомнения технического плана по поводу архитектуры программы: В какой среде лучше программировать: в Matlab или на C++? Оба имеют свои преимущества и недостатки.

Достоинства языка C++:

1) в C++ предоставлен широкий ряд инструментов для использования классов и других приемов объектно-ориентированного программирования;

2) скомпилированные модули, написанные на C или C++, теоретически должны производить расчеты быстрее, чем их аналоги на Matlab.

Достоинства среды Matlab:

1) Имеется возможность легко орудовать векторами и матрицами. Любая переменная в данной системе (даже отдельное число или вектор) представляет собой двухмерный массив, поэтому реализованы все стандартные технические приемы работы с векторами и матрицами.

2) В системе Matlab и дополнительных пакетах программ (Toolboxes) реализованы практически все известные примы различных разделов современной математики; все новое, что появляется в математике, через определенное время реализуется в дополнительном пакете программ. В результате при работе в Matlab есть возможность использовать готовые реализации стандартных математических методов.

3) При необходимости с помощью дополнительного пакета Matlab Compiler предоставляется возможность в дальнейшем сочетать полученные модули с модулями, написанными на C++, сгенерировав текст аналогичной программы на C++, который выполняет процедуры так, как это реализовано в Matlab.

Сравнив два языка программирования и учитывая особенности предложенного алгоритма, было решено реализовать его в среде Matlab:

- в приведенных алгоритмах работа в основном проводится над матрицами и векторами
- используются стандартные методы многомерного статистического анализа, имеющие готовые реализации в пакете Matlab Statistics
- в перспективе использование аппарата нечеткой логики, реализация которого приведена в Toolbox'е Fuzzy Logic
- предоставляется возможность формирования текста аналогичной программы на C++ с помощью пакета Matlab Compiler

4.1.2. Описание среды Matlab.

Matlab – это высокопроизводительный язык для технических расчетов. Он включает в себя вычисления, визуализацию и программирование в удобной среде, где задачи и решения выражаются в форме, близкой к математической. Типичное использование Matlab – это:

- Математические вычисления
- Создание алгоритмов
- Моделирование
- Анализ данных, исследования и визуализация
- Научная и инженерная графика
- Разработка приложений, включая создание графического интерфейса

Matlab – это интерактивная система, в которой основным элементом данных является двухмерный массив, или матрица. Это позволяет решать задачи, связанные с техническими вычислениями, особенно в которых используются вектора и матрицы, в несколько раз быстрее, чем при написании программ с использованием «скалярных» языков программирования, таких как Си или Фортран.

Matlab развивался в течении нескольких лет, ориентируясь на различных пользователей. В университетской среде, он представлял собой стандартный инструмент для работы в различных областях математики, машиностроения и науки. В промышленности, Matlab – это инструмент высокопродуктивных исследований, разработок и анализа данных.

В Matlab важная роль отводится специализированным группам программ, называемых Toolboxes. Они очень важны для большинства пользователей Matlab, так как позволяют изучать и применять специализированные методы. Toolboxes –это всесторонняя коллекция функций Matlab (M-файлов), которые

позволяют решать частные классы задач. Toolboxes применяются для обработки сигналов, систем контроля, нейронных систем, нечеткой логики, моделирования и т.д.

Из всех реализованных в Matlab Toolboxes нас на данный момент прежде всего интересует пакет программ, носящий название Statistics, предлагающий широкий выбор инструментов для статистического исследования.

Тексты программных модулей на языке Matlab сохраняются в файлах с расширением *.m*. Они представляют собой текстовые файлы, содержащие внешние определения команд и функций системы. Модули реализуются в виде процедур и функций.

4.1.3. Особенности архитектуры.

Предложенный в работе алгоритм построения деревьев решений - линейных классификаторов реализован в среде Matlab 7.0. Программная реализация представлена в виде вызывающих друг друга процедур со вложенными функциями. Написано порядка 15 модулей-процедур, которые изменялись по мере совершенствования алгоритма.

При написании программ активно использовался дополнительный пакет программ Matlab Statistics Toolbox, позволяющий выполнять различные статистические расчеты. Применены реализации следующих алгоритмов статистического анализа:

- линейная регрессия (linear regression):

функции *regress, glmfit*

- гребневая регрессия (ridge regression):

функция *ridge*

- иерархический кластерный анализ:

функции *pdist, linkage, dendrogram*

Работу всего алгоритма запускает главный модуль *main.m*, вызывая остальные, вспомогательные модули. Значения всех параметров алгоритма задаются в главном модуле *main.m* и при желании пользователя могут принимать другие значения. К таким параметрам относятся:

1) параметры иерархического кластерного анализа:

- метрика кластерного анализа – мера сходства (*mname*);

- алгоритм объединения двух кластеров (*linkage_method*);
- количество находимых кластеров (*clusttot*)
- первоначальное минимальное число элементов в кластере (*clustmin*)

2) параметры МГУА:

- глубина МГУА (количество отбираемых прогнозирующих моделей на шаге селекции) – отдельные значения для МГУА на полной выборке и МГУА на отдельных кластерах, для первого ряда селекции и остальных рядов (*q_single_1, q_mult_1, q_single_2, q_mult_2*)
- порог дисперсии для фильтрации признаков, близких к константным (*vartol*)
- порог корреляции для фильтрации сильно коррелирующих признаков – отдельные значения для первого ряда селекции и остальных рядов (*corrtol_single, corrtol_mult*)

Параметры запуска алгоритма и результаты работы протоколируются и выводятся в файлы:

1) *parameters.txt* - содержит вышеперечисленные параметры алгоритма;

2) *clusters.txt* – содержит описание кластеров разбиения (число элементов и номера входящих элементов для каждого кластера);

3) *process.txt* – содержит промежуточные и конечные результаты:

- номера столбцов, оставшихся после первой фильтрации на близкие к константным и сильно коррелирующие столбцы
- номера столбцов, задействованных в прогнозирующей модели, полученной с помощью МГУА на полной выборке, и, как следствие, входящих в метрику кластерного анализа

для каждого кластера:

- число элементов
- номера входящих элементов
- коэффициенты построенной с помощью МГУА классифицирующей функции
- номера столбцов, задействованных в классифицирующей функции
- коэффициент качества прогнозирующей линейной модели

Структура данных, их форматы, подробное описание работы модулей, а также сам текст программ приведены в Приложении к текущей работе.

4.2. Проведение расчетов.

4.2.1. Проблемы при проведении расчетов и их преодоление.

С целью выявления зависимости значений биологической активности от молекулярной структуры проведены вычислительные эксперименты на выборке молекул амбровых одорантов, состоящей из 50 молекул. Из них 37 являются биологически активными, 13 - неактивными. Обработано две матрицы «молекула-признак», сформированных на основе разных наборов дескрипторов, построенных численностью 703 и 231 соответственно. Таким образом, в каждом случае обрабатываемые матрицы оказались очень широкими, количество столбцов на порядок превышало число строк. По этой причине возникают проблемы при проведении расчетов. Так как реализованный алгоритм несколько раз запускает алгоритм Метода Группового Учета Аргументов на широких матрицах, который на каждом шаге селекции фактически перебирает всевозможные комбинации уже полученных прогнозирующих моделей и имеющихся переменных, алгоритм работает крайне медленно. (Даже выполнение оптимизированного алгоритма занимает порядка 20 часов на персональной ЭВМ с процессором Intel Pentium 4 2.40GHz и 512 Мб оперативной памяти.)

Для решения данной проблемы были предложены следующие решения:

1) *фильтрация каждого ряда селекции МГУА для отброса сильно коррелирующих и близких к константным столбцов*

Данный прием позволяет сократить перебор линейных моделей и перменных-признаков.

2) *использование наиболее информативных структурных 3D-дескрипторов низшего уровня для отбора 3D-дескрипторов высшего уровня*

Этот прием дает возможность сократить алфавит дескрипторов высокого уровня, входящих в обрабатываемую таблицу признаков.

3) *преобразование программ в аналогичные по выполняемым процедурам программы на языке C с последующим линкованием функций библиотек Matlab и компиляцией в исполнимый .exe-файл* (все это позволяет сделать пакет Matlab Compiler)

Этот прием мог бы дать ускорение работы программы, так как теоретически программа, написанная на Matlab, работает медленнее, чем ее аналог на C.

Первые два из перечисленных решений, как видно из предыдущих глав, были применены и дали свои результаты.

4.2.2. Численные результаты расчетов.

При обработке матрицы из структурных 3D-дескрипторов размерностью 50x703 были получены следующие численные результаты.

Матрица обрабатывалась трижды, с разными модификациями алгоритма. В первый раз в иерархическом кластерном анализе использовался метод ближнего соседа объединения кластеров и стандартная среднеквадратическая ошибка в качестве функционала качества. Результат был неудовлетворительным:

- выделены кластеры в 26 и 5 элементов
- для них построены прогнозирующие модели из 6 и 1 признаков соответственно с прогностической оценкой на скользящем контроле 0.22 и -1.19

Во второй раз метод объединения кластеров иерархического кластерного анализа был заменен на центроидальный и использовался функционал качества, адаптированный к логическому формату прогнозируемого столбца. В итоге, получен хороший результат:

- выделены кластеры в 28 и 10 элементов
- для них построены прогнозирующие модели из 6 и 2 признаков соответственно с прогностической оценкой на скользящем контроле 0.79 и 0.8

В оба предыдущих раза МГУА на отдельных кластерах обрабатывал матрицы, содержащие только те столбцы, что входили в метрику кластерного анализа. В третий раз были взяты те же параметры, что и при втором запуске, но МГУА на отдельных кластерах обрабатывал матрицы с полным набором признаков. Результат получился по качеству соотносимый со вторым разом:

- выделены кластеры в 27 и 8 элементов
- для них построены прогнозирующие модели из 6 и 2 признаков соответственно с прогностической оценкой на скользящем контроле 0.67 и 0.88

На проведение последних расчетов (однократного запуска алгоритма) затрачено 19 ч. 44 мин. на персональной ЭВМ с процессором Intel Pentium 4

2.40GHz и 512 Мб оперативной памяти. Все параметры запуска алгоритма и промежуточные и конечные результаты запротоколированы.

4.3. Выводы.

Реализация предложенных алгоритмов и выполнение численных экспериментов на конкретной выборке амбровых одорантов продемонстрировали хорошие численные результаты: получено разбение на крупные кластеры, для них построены линейные классифицирующие модели с высокой прогностической оценкой. Таким образом, предложенный алгоритм и его реализацию можно считать успешными.

Заключение.

В дипломной работе разработан и программно реализован новый метод поиска функциональной зависимости биологической активности от значений дескрипторов в виде дерева решений.

Предложена модификация Метода Группового Учета Аргументов с использованием иерархического кластерного анализа, что позволяет выявить «сгустки однородности», а затем удалить из выборки так называемые «выбросов», мешающие выявить функциональную зависимость на основных кластерах.

Предложен метод выбора метрики для проведения иерархического кластерного анализа для более верного выделения кластеров и сокращения вычислений. Метод состоит в использовании МГУА на полной выборке молекул и отборе наиболее информативных столбцов для всей выборки.

Предложен и использован новый метод отбора прогностических моделей при каждой селекции МГУА. Его достоинством является то, что он использует тот факт, что результирующий вектор состоит только из двух значений: 0 и 1.

Было предложено использовать обратную связь между этапом описания объектов (молекул) и формирования дескрипторов и этапом поиска функциональной зависимости. Его суть состоит в том, что после запуска алгоритма, исходя из качества полученного прогноза, формируются рекомендации по изменению параметров детализации, использованных на этапе описания молекул и формирования дескрипторов. Далее весь алгоритм запускается заново уже с новыми параметрами детализации. В частности, найденные наиболее информативные структурные 3D - дескрипторы предыдущего уровня используются при построении 3D - дескрипторов следующего уровня.

Также предложены и использованы инструменты для уменьшения объема вычислений при проведении селекции. Они состоят в фильтрации набора векторов каждой селекции МГУА для отсеивания признаков, сильно коррелирующих с остальными, и признаков, близких к константе.

Различные модификации вышеописанного алгоритма успешно реализованы в системе Matlab 7.0. При этом использовались как основной пакет программ, так и дополнительный пакет (Toolbox) Statistics, предлагающий

широкий выбор инструментов для статистического исследования. Проект состоит из порядка 15 модулей-процедур.

Полученная реализация дает возможность после построения дерева решений в виде классификаторов вернуться от дескрипторов к 3D-формам на молекулярном графе, т. е. возможность визуализации фрагментов молекулярного графа, чьи дескрипторы вошли в классифицирующую функцию.

С целью выявления зависимости значений биологической активности от молекулярной структуры проведены вычислительные эксперименты на выборке молекул амбровых одорантов, состоящей из 50 молекул. Из них 37 являются биологически активными, 13 - неактивными. Обработано две матрицы «молекула-признак», сформированных на основе разных наборов дескрипторов, численностью 703 и 231 соответственно. Причем, в обоих случаях дескрипторы были построены по 3D-графу, в вершинах которого находятся не атомы, а особые точки (точки локальных экстремумов, геометрических или физико-химических свойств) на молекулярных поверхностях. В результате, на одной из них удалось выделить два крупных кластера из 28 и 10 элементов с хорошей прогностической оценкой на скользящем контроле 78,6% и 80% соответственно. На проведение расчетов затрачено 19 ч. 44 мин. на персональной ЭВМ с процессором Intel Pentium 4 2.40GHz и 512 Мб оперативной памяти.

Можно выделить следующие основные направления развития данной работы:

1) решение проблемы поиска «выбросов» - элементов выборки, сильно отличающихся по своему описанию от основных кластеров и, как следствие, не поддающихся общему анализу; уменьшение влияния наличия «выбросов» в выборке на общий результат классификации молекул;

2) использование структурных 3D-дескрипторов более высокого порядка при формировании матрицы «структура-свойство», опираясь на уже построенные информативные дескрипторы низшего прядка, а не перебирая всевозможные дескрипторы (в данном проекте удалось задействовать только структурные дескрипторы, соответствующие двойкам и тройкам особых точек);

3) реализация и использование при построении деревьев решений-классификаторов других алгоритмов кластерного анализа, например, метода *k* – средних;

4) преобразование исходной матрицы «структура-свойство» к матрице данных логического формата (т.е. состоящую из 0 и 1) путем дискретизации интервала значений дескрипторов и использование в качестве классификатора модификации МГУА, основанные на дизъюнктивной нормальной форме;

5) использование подхода нечеткой логики (Fuzzy Logic) как на этапе описания молекул, так и на этапе построения классификаторов: введение нечетких интервалов, формирование основанных на них структурных дескрипторов, построение классификаторов – модификаций МГУА, основанных на аналоге дизъюнктивной нормальной формы.

Работа в данных направлениях, возможно, могла бы предоставить дерево решений-классификаторов с еще лучшей прогностической оценкой.

Список литературы и других источников.

Книги и статьи:

1. Журавлев Ю.И., Рязанов В.В., Сенько О.В.

«Распознавание». Математические методы. Программная система. Практические применения.

М.: ФАЗИС, 2006. 176 с.

2. Скворцова М.И., Станкевич И.В., Палюлин В.А., Зефиров Н.С.

Концепция молекулярного подобия и ее использование для прогнозирования свойств химических соединений. (в публикации)

3. Сошникова Л.А., Тамашевич В.Н., Уебе Г., Шефер М.

Многомерный статистический анализ в экономике.

4. …

Интернет-ресурсы:

1. www.qsar.ru
2. http://www.basegroup.ru/trees/description.htm

Приложение.