Тема 3. Компьютерный анализ данных

Лекция 9. Введение в компьютерный анализ

1. Методы классификации и кластеризации данных.

С ростом информации, обрабатываемой, хранимой и получаемой в ходе работы информационных процессов, возникающих в различных сферах деятельности человека, ее обработка, в полученном виде, становится затруднительна. Появляется необходимость первоначальной обработки информации для ее структурирования, выделения характерных признаков, обобщения, сортировки. Для этого применяют процессы классификации и кластеризации, позволяющие в полной мере производить требуемую обработку информации, для ее последующего анализа.

Классификация данных — процесс упорядочения или распределения объектов (наблюдений) по классам с целью отражения отношений между ними.

Класс — это множество документов, имеющих определенный общий признак, отличающий эту совокупность от других объектов.

В качестве классификационного деления можно принять различные признаки, зависящие от цели классификации. В основу класса всегда кладут наиболее важный признак документа, отвечающий цели классификации.

Классифицировать объект — значит, указать номер (или наименование) класса, к которому относится данный объект. Классификация объекта — номер или наименование класса, выдаваемый алгоритмом классификации в результате его применения к данному конкретному объекту.

Обучение классификатора — процесс построения алгоритма в случае, когда задано конечное множество объектов, для которых известно, к каким классам они относятся. Это множество называется выборкой. Классовая принадлежность остальных объектов не известна.

Кластеризация — процесс разбиения заданной выборки объектов (наблюдений, данных) на непересекающиеся подмножества, называемые кластерами, так, чтобы каждый кластер состоял из схожих объектов, а объекты разных кластеров существенно отличались.

Одной из целью кластеризации является понимание данных путём выявления кластерной структуры. Разбиение наблюдений на группы схожих объектов позволяет упростить дальнейшую обработку данных и принятия решений, применяя к каждому кластеру свой метод анализа (стратегия «разделяй и властвуй»).

Кластеризация применяется при необходимости сжатия данных. Если исходная выборка избыточно большая, то можно сократить её, оставив по одному наиболее типичному представителю от каждого кластера.

Так же кластеризация служит для обнаружения новизны. Выделяются нетипичные объекты, которые не удаётся присоединить ни к одному из кластеров. Для решения задач методами кластерного анализа, необходимо задавать количество кластеров заранее. В первом случае число кластеров стараются сделать поменьше. Во втором случае важнее обеспечить высокую степень сходства объектов внутри каждого кластера, а кластеров может быть сколько угодно. В третьем случае наибольший интерес представляют отдельные объекты, не вписывающиеся ни в один из кластеров.

Классификация методов кластерного анализа.

Методы по способу обработки данных:

Иерархические методы:

Агломеративные методы AGNES (Agglomerative Nesting):

- CURE;
- ROCK;
- CHAMELEON и т.д.

Дивизимные методы DIANA (Divisive Analysis):

- BIRCH;
- MST и т.д.

Неиерархические методы.

Итеративные

- K-средних (k-means)
- PAM (k-means + k-medoids)
- CLOPE
- LargeItem и т.д.

Методы по способу анализа данных:
- Четкие;
- Нечеткие.
Методы по количеству применений алгоритмов кластеризации:
- С одноэтапной кластеризацией;
- С многоэтапной кластеризацией.
Методы по возможности расширения объема обрабатываемых данных:
- Масштабируемые;
- Немасштабируемые.
Методы по времени выполнения кластеризации:
- Потоковые (on-line);
- Не потоковые (off-line).

Алгоритмы кластеризации.

Иерархическая кластеризация. При иерархической кластеризации выполняется последовательное объединение меньших кластеров в большие или разделение больших кластеров на меньшие.

Агломеративные методы AGNES (Agglomerative Nesting). Эта группа методов характеризуется последовательным объединением исходных элементов и соответствующим уменьшением числа кластеров. В начале работы алгоритма все объекты являются отдельными кластерами. На первом шаге наиболее похожие объекты объединяются в кластер. На последующих шагах объединение продолжается до тех пор, пока все объекты не будут составлять один кластер.

Алгоритм CURE (Clustering Using REpresentatives). Выполняет иерархическую кластеризацию с использованием набора определяющих точек для определения объекта в кластер. Назначение: кластеризация очень больших наборов числовых данных. Ограничения: эффективен для данных низкой размерности, работает только на числовых данных. Достоинства: выполняет кластеризацию на высоком уровне даже при наличии выбросов, выделяет кластеры сложной формы и различных размеров, обладает линейно зависимыми требованиями к месту хранения данных и временную сложность для данных высокой размерности. Недостатки: есть необходимость в задании пороговых значений и количества кластеров.

Описание алгоритма:

Шаг 1: Построение дерева кластеров, состоящего из каждой строки входного набора данных.

Шаг 2: Формирование <кучи> в оперативной памяти, расчет расстояния до ближайшего кластера (строки данных) для каждого кластера. При формировании кучи кластеры сортируются по возрастанию дистанции от кластера до ближайшего кластера. Расстояние между кластерами определяется по двум ближайшим элементам из соседних кластеров. Для определения расстояния между кластерами используются манхеттенская (r=|x|+|y|), евклидова (r=sqrt(x^2+y^2)), чебышовская (r=max(|x|,|y|)) метрики или похожие на них функции.

Шаг 3: Слияние ближних кластеров в один кластер. Новый кластер получает все точки входящих в него входных данных. Расчет расстояния до остальных кластеров для новообразованного кластера. Для расчета расстояния кластеры делятся на две группы: первая группа - кластеры, у которых ближайшими кластерами считаются кластеры, входящие в новообразованный кластер, остальные кластеры - вторая группа. И при этом для кластеров из первой группы, если расстояние до новообразованного кластера меньше чем до предыдущего ближайшего кластера, то ближайший кластер меняется на новообразованный кластер. В противном случае ищется новый ближайший кластер, но при этом не берутся кластеры, расстояния до которых больше, чем до новообразованного кластера. Для кластеров второй группы выполняется следующее: если расстояние до новообразованного кластера ближе, чем предыдущий ближайший кластер, то ближайший кластер меняется. В противном случае ничего не происходит.

Шаг 4: Переход на шаг 3, если не получено требуемое количество кластеров.

Алгоритм BIRCH (Balanced Iterative Reducing and Clustering using Hierarchies). В этом алгоритме предусмотрен двухэтапный процесс кластеризации. Назначение: кластеризация очень больших наборов числовых данных. Ограничения: работа с только числовыми данными. Достоинства: двухступенчатая кластеризация, кластеризация больших объемов данных, работает на ограниченном объеме памяти, является локальным алгоритмом, может работать при одном сканировании входного набора данных,

использует тот факт, что данные неодинаково распределены по пространству, и обрабатывает области с большой плотностью как единый кластер. Недостатки: работа с только числовыми данными, хорошо выделяет только кластеры сферической формы, есть необходимость в задании пороговых значений.

Алгоритм MST (Algorithm based on Minimum Spanning Trees). Назначение: кластеризация больших наборов произвольных данных. Достоинства: выделяет кластеры произвольной формы, в т.ч. кластеры выпуклой и невыпуклой формы, выбирает из нескольких оптимальных решений самое оптимальное.

Неиерархическая кластеризация.

Алгоритм k-средних (k-means). Алгоритм k-средних строит k кластеров, расположенных на возможно больших расстояниях друг от друга. Основной тип задач, которые решает алгоритм k-средних, - наличие предположений (гипотез) относительно числа кластеров, при этом они должны быть различны настолько, насколько это возможно. Выбор числа k может базироваться на результатах предшествующих исследований, теоретических соображениях или интуиции. Общая идея алгоритма: заданное фиксированное число k кластеров наблюдения сопоставляются кластерам так, что средние в кластере (для всех переменных) максимально возможно отличаются друг от друга. Ограничения: небольшой объем данных. Достоинства: простота использования; быстрота использования; понятность и прозрачность алгоритма. Недостатки: алгоритм слишком чувствителен к выбросам, которые могут искажать среднее; медленная работа на больших базах данных; необходимо задавать количество кластеров.

Алгоритм PAM (partitioning around medoids). Ограничения: небольшой объем данных. Достоинства: простота использования; быстрота использования; понятность и прозрачность алгоритма, алгоритм менее чувствителен к выбросам в сравнении с k-means. Недостатки: необходимо задавать количество кластеров; медленная работа на больших базах данных. Этот алгоритм аналогичен алгоритму k-средних, только при работе алгоритма перераспределяются объекты относительно медианы кластера, а не его центра.

Алгоритм CLOPE. Назначение: кластеризация огромных наборов категорийных данных. Достоинства: высокие масштабируемость и скорость работы, а так же качество кластеризации, что достигается использованием глобального критерия оптимизации на основе максимизации градиента высоты гистограммы кластера. Он легко рассчитывается и интерпретируется. Во время работы алгоритм хранит в RAM небольшое количество информации по каждому кластеру и требует минимальное число сканирований набора данных. CLOPE автоматически подбирает количество кластеров, причем это регулируется одним единственным параметром - коэффициентом отталкивания.

Самоорганизующиеся карты Кохонена. Назначение: кластеризация многомерных векторов, разведочный анализ данных, обнаружение новых явлений. Достоинства: используется универсальный аппроксиматор - нейронная сеть, обучение сети без учителя, самоорганизация сети, простота реализации, гарантированное получение ответа после прохождения данных по слоям. Недостатки: работа только с числовыми данными, минимизация размеров сети, необходимо задавать количество кластеров. Самоорганизующая карта Кохонена - нейронная однослойная сеть прямого распространения.

Алгоритм HCM (Hard C - Means). Назначение: кластеризация больших наборов числовых данных. Достоинства: легкость реализации, вычислительная простота. Недостатки: задание количества кластеров, отсутствие гарантии в нахождении оптимального решения.

Нечеткая кластеризация.

Алгоритм Fuzzy C-means. Назначение: кластеризация больших наборов числовых данных. Достоинства: нечеткость при определении объекта в кластер позволяет определять объекты, которые находятся на границе, в кластеры. Недостатки: вычислительная сложность, задание количества кластеров, возникает неопределенность с объектами, которые удалены от центров всех кластеров.

2. Статические и динамические модели данных.

В ***статических моделях*** система представляется неизменной во времени. Такие модели удобны, когда нужно описать структуру системы, то есть из каких объектов она состоит, как эти объекты связаны с друг с другом и каковы свойства этих объектов. Образно говоря, статическая модель представляет собой как бы "фотографию" существенных свойств системы в некоторый момент времени. Примеры статических моделей: карта местности, схема персонального компьютера, перечень планет Солнечной системы с указанием их массы.

Динамические модели содержат информацию о поведении системы и ее составных частей. Для описания поведения обычно используются записанные в виде формул, схем или компьютерных

программ соотношения, позволяющие вычислить параметры системы и ее объектов, как функции времени. Примеры динамических моделей: набор формул небесной механики, описывающий движение планет Солнечной системы; график изменения температуры в помещении в течение суток; видеозапись извержения вулкана.

В зависимости от цели моделирования для одной и той же системы могут создаваться как статические, так и динамические модели. Построение динамических моделей обычно сложнее, чем статических, поэтому, если значения свойств системы изменяются редко или медленно, то лучше построить статическую модель системы и при необходимости вносить в нее коррективы.

С позиций изменчивости можно выделить два класса моделей: статические, динамические и квазидинамические. К статическим относят модели инвариантные относительно времени. Они служат для описания процессов и явлений, независящих от времени. Динамические модели не только допускают изменение параметров и структур во времени, но и служат для описания изменения процессов и моделей именно во времени. Построение динамических моделей (например для задач управления) как правило более сложно чем построение статических. Поэтому в некоторых случаях применяют квазидинамические модели как упрощение динамических. Квазидинамические модели - это модели, в которых временной интервал действия модели разбивается на периоды, для каждого из которых строится статическая модель. Таким образом, квазидинамические модели можно рассматривать как совокупность меняющихся и взаимосвязанных статических моделей.

Примерами динамических и статических моделей в ГИС могут служить два вида электронных карт. Электронные карты в режиме разделения времени (электронные атласы) представляют реализацию статических моделей, в то время как электронные карты в реальном масштабе времени (навигационные системы) могут служить примером динамической модели.

Следует подчеркнуть, что понятие изменчивости моделей данных в информационных системах - относительно, так как вся информация носит временной характер и через какой-то период времени требует обновления (актуализации). Поэтому применение понятий статистические и динамические модели данных требует указания периода времени, который используется в процессе исследований или указания альтернативной модели при сравнении с исходной.

Мы говорили об информационных моделях и об одном из видов программного обеспечения для работы с ними. Вы помните, что СУБД позволяет хранить большое количество информации и находить среди нее нужную. Однако, зачастую требуется не просто хранить некоторую информацию (знания об объектах), но и динамически их изменять в соответствии с заданными функциями.

Чаще всего правилами, описывающими функционирование системы являются математические формулы. В этом случае модель называют математической. Одним из средств для построения компьютерных математических моделей являются электронные таблицы (или табличные процессоры). Как ясно из названия, электронная таблица предназначена для табличных расчетов. Представление информации в ней похоже на реляционную базу данных. Но, в отличие от таблицы в базе данных, здесь строки совсем не обязательно должны быть однотипными.

Но вот мы создали математическую модель, занесли данные в таблицу, выполнили расчеты -- и получили большое количество чисел. Хорошо бы представить результаты вычислений понагляднее. Тут нам тоже поможет табличный процессор. Оказывается, он умеет строить диаграммы. Диаграмма -- условное графическое изображение числовых величин или их соотношений.

Рассмотрим три их разновидности: столбчатую, линейную и круговую диаграммы.

На столбчатой диаграмме каждая величина изображается в виде столбика. Его высота показывает в соответствующем масштабе (он наносится на вертикальной оси) числовое значение этой величины. Что обозначает каждый столбик может быть написано либо непосредственно около него, либо в так называемой "легенде" -- табличке, где указано чему соответствует каждый цвет. На практике одинаково часто встречаются диаграммы как с вертикальным, так и с горизонтальным расположением столбиков.

Линейная диаграмма наиболее часто используется, когда хотят показать изменение какой-либо величины, например, с течением времени. При ее построении отмечаются точки, расстояние которых от горизонтальной оси соответствует (в заданном масштабе) значениям величины, а затем эти точки соединяются отрезками. На горизонтальной оси указывают, чему соответствует каждое значение.

По круговой диаграмме , в отличие от двух предыдущих, нельзя определить значения величин. Это круг, разделенный на сектора, размеры которых соотносятся также как изображаемые ими числовые величины.

Табличные процессоры позволяют строить не только несколько разновидностей диаграмм, но и графики (график, в отличие от диаграммы, изображает зависимость одной величины от другой).

Типы и структуры данных.

Основные типы данных:

- Фактические данные: числовые, символьные, текстовые, графические и т.д.
- Абстрактные данные

Абстрактный тип данных (АТД) - это математическая модель для типов данных, где тип данных определяется поведением (семантикой) с точки зрения пользователя данных, а именно в терминах возможных значений, возможных операций над данными этого типа и поведения этих операций.

Формально АТД может быть определен как множество объектов, определяемое списком компонентов (операций, применимых к этим объектам, и их свойств). Вся внутренняя структура такого типа спрятана от разработчика программного обеспечения - в этом и заключается суть абстракции. Абстрактный тип данных определяет набор функций, независимых от конкретной реализации типа, для оперирования его значениями. Конкретные реализации АТД называются структурами данных.

Запах -> анализатор данных -> цифровая модель данных -> обработка и анализ данных -> устройство для воспроизводства данных (распылитель ароматизаторов)

Физ. объект -> анализатор данных -> цифровая модель данных -> обработка и анализ данных -> устройство для воспроизводства данных (3D принтер)

В программировании абстрактные типы данных обычно представляются в виде интерфейсов, которые скрывают соответствующие реализации типов. Программисты работают с абстрактными типами данных исключительно через их интерфейсы, поскольку реализация может в будущем измениться. Такой подход соответствует принципу инкапсуляции в объектно-ориентированном программировании. Сильной стороной этой методики является именно сокрытие реализации. Раз вовне опубликован только интерфейс, то пока структура данных поддерживает этот интерфейс, все программы, работающие с заданной структурой абстрактным типом данных, будут продолжать работать. Разработчики структур данных стараются, не меняя внешнего интерфейса и семантики функций, постепенно дорабатывать реализации, улучшая алгоритмы по скорости, надежности и используемой памяти.

Различие между абстрактными типами данных и структурами данных, которые реализуют абстрактные типы, можно пояснить на следующем примере. Абстрактный тип данных список может быть реализован при помощи массива или линейного списка, с использованием различных методов динамического выделения памяти. Однако каждая реализация определяет один и тот же набор функций, который должен работать одинаково (по результату, а не по скорости) для всех реализаций.

Абстрактные типы данных позволяют достичь модульности программных продуктов и иметь несколько альтернативных взаимозаменяемых реализаций отдельного модуля.

Примеры ADT: Список, Стек, Очередь, Дерево, Ассоциативный массив, Очередь с приоритетом и т.д.

Основные структуры данных

Работа с большими наборами данных автоматизируется проще, когда данные упорядочены, то есть образуют заданную структуру. Существует три основных типа структур данных: линейная, табличная и иерархическая. При создании любой структуры данных необходимо обеспечить решение двух задач: как разделять элементы данных между собой и как разыскивать нужные элементы.

Линейные структуры – это хорошо знакомые списки. Список – это простейшая структура данных, отличающаяся тем, что каждый элемент данных однозначно определяется своим уникальным номером в массиве (списке).

Табличные структуры данных подразделяются на двумерные и многомерные.

Двумерные табличные структуры данных (матрицы) – это упорядоченные структуры, в которых адрес элемента определяется номером столбца и номером строки, на пересечении которых находится ячейка, содержащая искомый элемент.

Многомерные таблицы – это упорядоченные структуры данных, в которых адрес элемента определяется тремя и более измерениями. Для отыскания нужного элемента в таких таблицах необходимо знать параметры всех измерений (размерностей).

Линейные и табличные структуры являются простыми. Ими легко пользоваться, поскольку адрес каждого элемента задаётся числом (для списка), двумя числами (для двумерной таблицы) или несколькими числами для многомерной таблицы. Они также легко упорядочиваются. Основным методом упорядочения таких данных является сортировка. Недостатком простых структур данных является трудность их обновления. При добавлении, например, произвольного элемента в упорядоченную структуру возникает необходимость изменения адресных данных у других элементов.

Иерархические структуры – это структуры, объединяющие нерегулярные данные, которые трудно представить в виде списка или таблицы. В иерархической структуре адрес каждого элемента определяется маршрутом, ведущим от вершины структуры к данному элементу. Эти структуры по форме сложнее, чем линейные и табличные, но они не создают проблем с обновлением данных. Их легко развивать путём создания новых уровней. Недостатком иерархических структур является относительная трудоёмкость записи адреса элемента данных и сложность упорядочения. Поэтому для упорядочения в таких структурах применяется метод предварительной индексации. При этом каждому элементу данных присваивается свой уникальный индекс, который используется при поиске, сортировке и тому подобное. В качестве примера иерархической структуры может служить система почтовых адресов.

Гипертекст и гипермедиа

Термин гипертекст был введён Тедом Нельсоном (Ted Nelson) ещё в 60-х годах.

Гипертекст представляет собой тот же текст, что и, например, текст MS Word 6.0-7.0, но отличается тем, что некоторые его части (символы, слова, фразы, рисунки) являются интерактивными ссылками на другие документы. Теперь большой документ можно разбить на отдельные темы и связать их через ключевые фрагменты (символы, слова, фразы или рисунки). Эти фрагменты, выделенные в документе особым образом для их идентификации, служат для перехода на связанные с ними по смыслу другие темы, или содержат в себе вызов других приложений. Таким образом, можно сказать, что *гипертекст* представляет собой содержание, внедрённое непосредственно в документ. Упрощая чтение, делая информацию нагляднее и понятнее, *гипертекст* создаёт комфортную работу с документом, автоматически выполняя многие действия, которые пользователю приходилось делать до этого вручную.

Термином ‘мультимедиа’ (multimedia) обозначаются интерактивные компьютерные системы, обеспечивающие работу с разнообразными типами данных – неподвижными и движущими изображениями (включая видео), а также с текстом, речью и высококачественным звуком. В соответствующих базах данных хранится не только текстовая информация, но и оцифрованные видеоклипы, звуки и музыка, факсимильные изображения и многое другое. Современные системы управления мультимедийными базами данных поддерживают технологию ‘клиент /сервер’, а сами базы данных оказываются, распределенными по узлам всемирной компьютерной сети. При этом возникает новая ситуация, которая в ближайшие годы будут определять развитие цивилизации – большинство знаний, накопленных человечеством, оказывается интегрированным в глобальную информационную систему, а доступ к этим знаниям открыт для каждого члена общества. С развитием технологий мультимедиа появились также системы, обеспечивающие возможность установления гиперсвязей для изображений. Они позволяют рассматривать фрагменты изображений, хранящиеся как отдельные изображения, полученные с большей детализацией, а также устанавливать связи фрагментов изображений с другими изображениями, поясняющим текстом, звуком и т. п.

Гипермедиа – более широкое понятие, которым обозначают документы, включающие в себя мультимедиа-информацию, например, звук или видео.

В современный документ можно вставить практически любой объект из любого приложения и это делает его интерактивной средой. Фактически можно прийти к тому, что скоро граница между документом и приложением может стать очень тонкой.