Сравнительными характеристиками решенной задачи являются мягкий и жёсткий рейтинги. Мягкий рейтинг отражает среднюю сравнительную эффективность этого решения по сравнению с другими решениями, оказавшимися наилучшими при различных способах учета неопределенности для данного варианта. Жёсткий рейтинг есть доля способов учета неопределенности.

В докладе описывается программная система, реализующая метод уверенных суждений. Это веб-преложение, использующее архитектурный паттерн MVC, что позволяет разделить логику от отображения. Сайт написан на языке PHP 5.4. В качестве хранилища данных используется РСУБД Postgresql 9.

Главный экран системы содержит ссылки: пользователи, задачи, доработки, администратор. Вход в систему осуществляется вводом логина и пароля. Регистрация осуществляется через обращение к администратору системы. Система содержит банк пользователей, каждый из зарегистрированных пользователей имеет в своём распоряжении банк своих задач, которые может создавать, инициировать, редактировать.

Отдельная задача представляет собой таблицу, строки которой отвечают аллтернативам, а столбцы частным критериям, по которым оценивается эффективность альтернатив. По каждому критерию задаётся направление оптимизации (на минимум, нейтрально, на максимум), шкала измерения значений критериев (количественная или порядковая) и группа важности. Введение групп важности является положительной отличчительной особеностью метода уверенных суждений, поскольку позволяет ЛПРу отражать свои предпочтения в порядоковой, а не в количественной шкале, поскольку измерение своих предпрочтений в количественной шкале является невыполнимым требованием к ЛПРу.

Система обеспечивает добавление и исключение альтернатив вариантов, ввод и редактирование их названий, ввод значений критериев для различных альтернатив непосредственно на экране. Кроме того возможна подготовка по аналогичному шаблону исходной информации в формате электронных таблиц OpenOffice Calc, Microsoft Excel и импорт их в систему. При нажатии на кнопку «Расчёт» происходит вычисление жёсткого и мягкого рейтингов альтернатив, причём объём случайной выборки моделирующей множество неопределнной весовых коэффициентов в методе уверенных суждений устанавливается пользователем в зависимости от желаемого им времени получения результатов. При нажатии на заголовок любого столбца таблицы происходит её упорядочение по элементам этого столбца.

Результаты решения могут быть сохранены в базе задач пользователя и выведены для документирования и дальнейшей работы с ними в формате электронных таблиц OpenOffice Calc, Microsoft Excel. Кроме того, реализована функция версионности. Этот механизм позволяет лицу принимающему решение углубляться в раздумия, корректировать условия решаемой задачи, сопоставляя соответствующие результаты в виде дерева возможных модификаций задачи принятия решения.

Система использовалась при сравнительной оценки структуры проектных решений по беспилотным летательным аппаратам (совместно с В. С. Брусовым) и сравнении эффективности национальных исследовательских университетов РФ (совмество с В. В. Малышевым), а также в ряде других приложений.

Литература

1. A Brief History of Decision Support Systems // Decision Support Systems Resources URL: http://dssresources.com/history/dsshistory.html (дата обращения: 13.03.2016)

2. The Decision Deck project URL: http://www.decision-deck.org/project/ (дата обращения: 13.03.2016).

3. Design, execute and share MCDA methods URL: http://www.decision-deck.org/diviz/ (дата обращения: 13.03.2016).

4. Multi-Criteria Decision Aiding tool // MCDA-ULaval URL: http://cersvr1.fsa.ulaval.ca/mcda/?q=en (дата обращения: 13.03.2016).

5. EURO Working Group Multicriteria Decision Aiding URL: http://www.cs.put.poznan.pl/ewgmcda/ (дата обращения: 13.03.2016).

6. С.А.Пиявский Два новых понятия верхнего уровня в онтологии многокритериальной оптимизации

7. Малышев В.В., Пиявский С.А. Метод “уверенных суждений” при выборе многокритериальн, Известия Российской академии наук., серия «Теория и системы управления», №5. 2015 – с. 90-101

О.Г. Костанян, А.В. Куприянов

КЛАСТЕРИЗАЦИЯ БОЛЬШИХ ОБЪЁМОВ ДАННЫХ

(Самарский национальный исследовательский университет
имени академика С.П. Королёва,
Институт систем обработки изображений РАН)

На сегодняшний день в мире накоплено огромное количество разной неструктурированной информации, из которой не сразу можно получить скольнибудь значимые знания. С целью упрощения восприятия огромных объемов данных разработано много алгоритмов кластеризации.

Кластеризация позволяет упростить дальнейшую обработку данных, сократить объем хранимых данных, выделить нетипичные объекты, построить иерархию множества объектов.

Кластеризация – это объединение схожих объектов в группы. В области анализа данных и Data Mining, кластеризация является фундаментом. Область применения кластеризации очень широка:

- сегментация изображений;
- анализ текстов;
- борьба с мошенничеством;

– маркетинг;
– прогнозирование.

Например, в маркетинге для того, чтобы провести аналитику, сначала, с помощью кластеризации выделяют группы схожих клиентов, товаров, покупателей, затем для каждой группы строят отдельную модель и стратегию, вместо того, чтобы построить одну общую модель и стратегию на всех данных.

Решение задачи кластеризации данных, вообще говоря, является неоднозначной, так как не существует точной постановки задачи кластеризации. Существует много критериев качества кластеризации, много эвристических методов кластеризации, число кластеров, как правило, неизвестно заранее, результат кластеризации существенно зависит от метрики, которую эксперт задает субъективно.

Выделим наиболее популярные методы кластеризации:

– статистические алгоритмы
– k-средних (k-means)
– иерархическая кластеризация (таксономия)
– графовые алгоритмы
– нейронная сеть Кохонена

Статистические алгоритмы кластеризации основаны на разделении смеси вероятностных распределений по конечной метрике. Другими словами задача кластеризации совпадает с задачей разделения смеси вероятностных распределений и для решения используют EM-алгоритм. В результате кластеризации данным алгоритмом, на выходе мы получим множество объектов, которые с определенной вероятностью будут принадлежать к каждому кластеру.

Т.е возможна ситуация, что один объект будет с вероятностью 0,5 принадлежать одному кластеру и 0,5 другому. Это является своеобразным недостатком данного алгоритма.

Метод k-средних (k-means), является одной из разновидностей EM-алгоритма, а вернее его упрощением, так как в отличие от EM, который каждому объекту ставит в соответствие кластер с определенной вероятностью, k-means жестко привязывает каждый объект к конкретному кластеру. Существует большое количество модификаций алгоритма k-средних, однако общий алгоритм схож:

сначала произвольным образом выбираются центры кластеров, затем каждый объект относится к ближайшему центру. После того как все объекты распределены по кластерам, происходит пересчет центра масс кластеров. Таким образом центр кластеров изменяется, следовательно, пересчитав все попарные расстояния от объектов до новых центров масс, мы получим новое распределение объектов по кластерам. Таким образом, продолжая эти итерации мы добьёмся того, что центры масс устоятся и не будут изменять свое положение, в этом случае кластеризация завершается и мы получаем на выходе объекты, которые приписаны к конкретному кластеру. Одним из значимых недостатков данного алгоритма является высокая чувствительность к выбору начальных центров и то,

что необходимо явно задавать количество кластеров. Решить данные проблемы можно немного модернизировав этот алгоритм: задавать начальные точки так, чтобы расстояния между центрами были максимальны, и проводить кластеризацию для разных параметров k до тех пор, пока не увидим значительного улучшения.

Еще одну группу алгоритмов составляют графовые алгоритмы: алгоритм выделения связных компонент, кратчайший незамкнутый путь, ФОРЭЛ (ФОРмальные ЭЛементы) и т.д.

Идея графовых алгоритмов заключается в том, что все объекты представляются в виде графов, а их рёбра это расстояние между заданными объектами.

В алгоритме выделения связных компонент, задается некое расстояние, по которому решают удалять ребро, связывающее два объекта, или нет. Таким образом, задав некое расстояние, можно кластеризовать множество объектов. Недостатком такого алгоритма является большая чувствительность к шуму и необходимость задания параметра R.

Еще более простым является алгоритм «кратчайший незамкнутый путь». Здесь мы находим пару вершин с наименьшим расстоянием и соединяем их ребрами, затем пока в выборке есть изолированные точки, мы соединяем их с ближайшим элементом. Когда все элементы будут соединены, тогда мы просто удаляем К-1 самых длинных ребер. Таким образом, мы получим К кластеров. Также недостатком является чувствительность к шуму и задание количества кластеров.

С целью качественного сравнения одного кластерного разбиения с другим вводится понятие функционала качества кластеризации. Приведем наиболее популярные функционалы качества:

– Минимизация среднего внутрикластерного расстояния F1 (суммируем все попарные расстояния объектов, которые попадают в один кластер). O(n^2)
– Максимизация среднего межкластерного расстояния F2 (суммируем все попарные расстояния объектов, которые попадают разные кластеры).
– Минимизация частного среднего внутрикластерного расстояния и среднего межкластерного расстояния F1/F2
– Минимизация среднего внутрикластерного расстояния Ф1 (суммируем все расстояния до центров своего кластера). O(n)
– Максимизация среднего межкластерного расстояния Ф2 (суммируем все расстояния между центрами кластеров).
– Минимизация частного среднего внутрикластерного расстояния и среднего межкластерного расстояния Ф1/Ф2

Работа с большим объемом данных невозможна в обычном однопотоковом программировании, т.к. для решения таких задач требуется большое количество процессорного времени, поэтому решением данной проблематики являются различные распределенные реализации. Одной из наиболее популярных

моделей для кластеризации больших объемов данных является технология распределённых вычислений MapReduce.

Модель программирования MapReduce в общем случае содержит несколько функций Map (распределитель) и Reduce (редуктор). Функция Map своими входными параметрами принимает пары key/value. Значение value каждого объекта представляет собой строку, состоящую из координат n-мерного вектора параметров. Входные данные распределяются между Mapper-ами. Начальное множество кластерных центров передаются каждому Mapper-у. Каждый центроид задается своим идентификатором в качестве ключа key и значением центроида в качестве значения value. Map функция, сравнивая значение очередного объекта со значениями кластерных центров, определяет, его принадлежность к заданным кластерам. Выходными значениями для каждого Mapper-а, пара ключ/значение для каждого объекта, где ключ – это номер ближайшего центроида, значение – вектор параметров в n-мерно векторе.

После работы функции Map, значения записываются в распределенную файловую систему. Между стадией Map и Reduce промежуточные данные сортируются и тассуются. Входными данными Reduce-ов будут выходные данные Mapper-ов.

Функция Reduce содержит все пары ключ/значение, которые получены от функции Map. Для всех пар ключ/значение, имеющих один и тот же ключ, их значение сохраняются в некотором итераторе, затем функция Reduce вычисляет среднее значение, которое имеет один и тот же ключ. Таким образом, получаем новые центроиды, и выходные данные передаются функциям Map. Этот процесс продолжается до тех пор, пока центры массы не перестанут изменяться.

Для решения задачи кластеризации с помощью парадигмы MapReduce можно выбрать платформу Hadoop, которая предназначена для создания и запуска распределенных приложений, работающая с большими объемами данных. Одним из основных компонентов платформы Hadoop является его файловая система HDFS (Hadoop distributed file system). Основной функцией распределенной файловой системы является разделение данных конкретного пользователя на блоки данных и репликация заданных блоков по локальным дискам узлов кластера.

Список источников

1. Батуркин С.А., Гостин А.М., А.В. Пруцков и др. Система внутреннего тестового контроля знаний РГРТУ: методические указания/ Рязан. гос. радиотехн. ун-т. – Рязань, 2007. – 68 с.

2. Мансурова М.Е., Шоманов А., Тулепбергенов Б., Параллельный алгоритм кластеризации для обработки гиперспектральных изображений на основе MapReduce Hadoop// Международная конференция “ИКТ: образование, наука, инновации”, Алматы, 20-21 мая 2013 г. – с. 56-61.

В.Д. Ленчук, Д.О. Маркин

СИСТЕМА МОНИТОРИНГА ОБМЕНА ЭЛЕКТРОННЫМИ СООБЩЕНИЯМИ УДАЛЕННЫМИ ПОЛЬЗОВАТЕЛЯМИ НА ОСНОВЕ ИСПОЛЬЗОВАНИЯ ОСОБЕННОСТЕЙ СИСТЕМЫ ДОМЕННЫХ ИМЕН

(Академия Федеральной службы охраны Российской Федерации)

Особое место среди источников утечек конфиденциальной информации занимает электронная почта. Хоть ее доля и уменьшается, электронная почта по-прежнему имеет весомый процент среди каналов утечек, о чем свидетельствует диаграмма, представленная на рисунке 1 [1].

Рисунок 1 – Распределение утечек по каналам
за первые полугодия 2014 и 2015 годов

В процессе информационного взаимодействия при работе системы обмена электронными сообщениями существенную роль на безопасности информации оказывают факторы и угрозы, определяемые стандартами [2, 3], относящиеся к классу сетевых, то есть, реализуемых с использованием протоколов межсетевого взаимодействия. Особое значение в сетевом взаимодействии играет система адресации и маршрутизации, важным элементом которой является система доменных имен. К числу угроз, использующих систему доменных имен, относятся угрозы, основанные на модификации пакетов DNS-транзакций, которые относятся к классу угроз, направленных на создание в сети ложного маршрута. Их потенциальная опасность заключается в возможности перехвата данных, передающихся между клиентами сетевых сервисов и серверами этих сервисов. Очевидно, что использование определенных возможностей системы доменных имен, позволяющих получить доступ к информационному взаимодействию удаленных пользователей, позволит реализовать функции мониторинга, аудита и фильтрации передаваемых данных, включая и сообщения электронной почты.