Технология очистки и трансформирования данных в рамках Knowledge Discovery in Databases (KDD) для ускоренного применения методов Data Mining

Ю. А. Шичкина[1,a], А. Б. Дегтярев[2,b], А. А. Коблов[1,c]

[1] Санкт-Петербургский государственный электротехнический университет «ЛЭТИ» имени В. И. Ульянова,
ул. Профессора Попова, 5, Санкт-Петербург, Ленинградская область, 197022

[2] Санкт-Петербургский государственный университет,
Университетская наб., 7-9, Санкт-Петербург

E-mail: [a] strange.y@mail.ru, [b] deg@csa.ru, [c] koblow.a.a@gmail.com

К большому объему данных сегодня приводит целый ряд причин. Рост объема баз данных предъявляет новые и весьма серьезные требования к аппаратуре центров обработки данных и в последние годы требует все больших вложений в оборудование, программное обеспечение, соответствующие работы и управление. Решение основных проблем, связанных с собственно аппаратной инфраструктурой центров обработки данных, надо сказать, обходится значительно дешевле, чем усовершенствование программного обеспечения. Но это только временное решение. Необходимо искать глобальное решение проблемы обработки больших данных. Надо усовершенствовать методы обработки данных с помощью той аппаратуры, которая имеется в наличии. В настоящей статье рассматриваются методы очистки и трансформирования данных в рамках технологии Knowledge Discovery in Databases для ускоренного применения методов интеллектуального анализа данных. В частности, показывается, каким способом можно значительно сократить выборку данных для построения запросов в noSQL базах данных на примере MongoDB.

Ключевые слова: большие данные, базы данных, MongoDB, очистка данных, запрос

Введение

«Большие данные» (Big Data) вот уже несколько лет как стали одним из самых популярных терминов в IT-области. Это является следствием того, что цифровые технологии полностью подчинили себе жизнь современного человека, сегодня все непрерывно и разных форматах пишется, читается, обрабатывается. Объем данных о самых разных сторонах жизни растет, и одновременно растут возможности хранения информации. К большим данным приводит и развитие вычислительных компьютерных сетей, обусловившее новые возможности в организации и ведении баз данных, позволяющие каждому пользователю иметь на своем компьютере свои данные и работать с ними и в то же время позволяющие работать всем пользователям со всей совокупностью данных как с единой централизованной базой данных. К большим данным приводит развитие новых направлений в IT-отрасли, таких как интернет вещей, которые способствуют генерации больших объемов информации. Таким образом, источников, генерирующих большие данные сегодня предостаточно в противовес технологиям, их обрабатывающим. И как показывает опыт последнего десятилетия решать проблему обработки больших информационных потоков только с помощью аппаратных средств невозможно. Скорость прироста объёмов информации превышает скорость развития аппаратных мощностей. Решить успешно проблемы, связанные с обработкой данных, можно, сделав в первую очередь акцент на развитии моделирования и программного обеспечения.

Долгое время первенство по популярности удерживали реляционные базы данных. Их существование исчисляется с 1970 года. Реляционные базы данных работают по логической схеме, являясь для многих организаций надежным «левым полушарием» и обеспечивая четыре ключевых условия: атомарность, согласованность, изолированность, надежность. Прежде всего, реляционные СУБД позволяют компаниям сохранить ссылочную целостность данных. Для промышленных предприятий и даже медицины это — золотой стандарт.

Но, в связи с присущими таким базам ограничениями вертикального масштабирования появились новые продукты, нестрого выполняющие фундаментальные требования к СУБД. Жестко заданные модели данных, обеспечивающие надежные гарантии консистентности данных и строгое следование стандарту SQL, уступили место моделям, лишенным жестких схем и преднамеренно денормализованным, со слабой консистентностью и с проприетарным API. Такие noSQL-системы (Cassandra, Riak, MongoDB и др.) предоставили программистам доступ к внутренним механизмам управления данными и обычно рассчитаны на горизонтальное масштабирование посредством кластеров из недорогих серверов умеренной производительности [Banker, 2016]. В таких системах высокое быстродействие, эластичность хранения и доступность обеспечиваются за счет секционирования и тиражирования срезов данных в пределах кластера. Но, даже эти системы вызывают проблемы, связанные с замедленной обработкой больших объемов информации.

Одним из способов ускорить выполнение запросов к базам данных, это провести предварительную очистку и трансформацию данных с применением технологии Knowledge Discovery in Databases.

Методика извлечения знаний из баз данных

Методика извлечения знаний из баз данных (Knowledge Discovery in Databases (KDD)) берет свое начало во второй половине 20 века и описывает не конкретные алгоритмы, а определенную последовательность действий, которой необходимо следовать для извлечения полезного знания [Piatetsky-Shapiro, 1991].

Изначально исходные данные необходимо получить из источников данных. Зачастую такие данные необходимо не просто собрать, а консолидировать из нескольких источников. После того, как данные получены и помещены в некоторое хранилище их необходимо отчистить

от «мусорных» данных (не имеющих ценности для получения полезного знания). На третьем этапе отчищенные данные необходимо подвергнуть трансформации. Трансформация представляет собой подготовку отчищенных данных к дальнейшей аналитике, поскольку многие из методов предполагают определенный формат данных для их применения. Четвертый этап — это Data Mining, процесс извлечения полезных знаний из исходных данных. На последнем этапе осуществляется интерпретация полученных знаний.

Например, отслеживание GPS-треков устройств представляет собой запись координат и их последующее хранение. Если фиксировать координаты одного устройства каждые 10 секунд на протяжении одного месяца, то мы получим, в итоге, более 15 миллионов записей, что является достаточно внушительной цифрой в перспективе слежения за более чем тысячей устройств.

Когда речь идет о больших данных подход KDD можно применить с некоторыми модификациями, а именно пройти этапы отчистки и трансформирования данных неоднократно. Данный подход позволит сократить количество полезных данных, что позволит более быстро и эффективно использовать методы извлечения знаний.

Пример предварительной отчистки и трансформации данных

Набор данных, которые можно собрать с помощью GPS-приемника, обширен, и включает в себя не только данные о местоположении. Например, база данных, которая была получена в результате участия транспортных средств в программе Safety Pilot Model Deployment (SPMD), содержит более 15 наборов свойств по транспортным средствам. Одним из ключевых наборов является "BsmP1", который хранит данные, описывающие текущее местоположение, скорость, и направление движения транспортного средства.

Если рассматривать задачу: «Генерация оптимальных правил маршрутизации для сети DTN среди устройств, расположенных на транспортных средствах. Устройства связываются между собой посредством Wi-Fi.», то можно пренебречь такими параметрами как ускорение и скорость, а также и многими техническими параметрами, входящими в набор данных BsmP1 и оставить только идентификатор устройства, координаты и время. Таким образом первый этап технологии KDD будет выполнен путем отбрасывания данных, которые не несут полезной нагрузки.

На основании выбранных параметров можно приступать к трансформации данных. Главной идеей упрощения обработки и обращения к данным является группирование данных по общему признаку. Поскольку для набора данных BsmP1 общими параметрами, не зависящими от других, являются координаты и время, то будет крайне логичным объединять данные в группы по времени и местоположению.

Самым простым и в то же время правильным будет объединение записей в группы по некоторому участку пространства, в котором устройства будут находиться. Самым тривиальным будет случай разбиения плоскости на прямоугольники или квадраты, ввиду двухмерности получаемых от GPS-спутника координат. Такое разбиение будем считать координатной сеткой.

Для нанесения координат на плоскость необходимо выбрать единичную меру, например, метр, и было бы все гораздо проще, если бы градусы долготы и широты однозначно переводились в метрическую систему. Как известно, длина меридиана постоянна, и, исходя из этого, можно однозначно перевести требуемую единицу в градусы широты. Для долготы сложнее, так как длина 1 градуса в метрах изменяется в зависимости от координаты. Если при нанесении двухмерной координатной сетки выполнять точные расчеты, то для обработки каждой записи необходимо заранее произвести расчеты длины метра в зависимости от координат объекта, и хранить все эти данные в отдельном хранилище. Данный подход требует дополнительных хранилищ и обращений к этим хранилищам. Поэтому, для локализованных вычислений координатной сетки в рамках одного города или района, допустимо использовать приближенное среднее значение, исходя из разности крайних координат, деленной на два.

Шаг сетки можно выбрать в зависимости от конечного назначения полученной выборки.

Для ускоренной навигации по координатной сетке следует добавить индексируемое поле nSquare, которое является уникальным идентификатором ячейки координатной сетки.

Переходя к группировке по времени стоит заметить, что загруженность определенных областей различными транспортными средствами, по которым возможно собрать телеметрию геолокации, достаточно периодична относительно дней недели. Каждую неделю график рабочей недели, в большинстве своем, не отличается, а предпочтительные места проведения досуга в выходные тоже могут совпадать из недели в неделю. Например, каждую пятницу менеджер Иванов, окончив рабочий день в 18:00, садится в свой автомобиль и двигается по направлению к дому. Находясь в 18:25 на улице Северодвинская Иванов стоит в длинной пробке из-за светофора и смотрит на серый рекламный стенд без рекламы. В 19:00 Иванов останавливается около гипермаркета, проводит там полчаса и, спустя еще полчаса, оказывается дома. А утром субботы Иванов выезжает в 12.00 для того, чтобы отвести свою дочь Валентину в бассейн на секцию плавания и т.д. Эти и многие другие действия выполняются периодично с достаточно высокой вероятностью. Таким образом, используя подобные данные можно определить места, где наиболее целесообразна установка рекламы, вышек связи.

Исходя из периодичности данных имеет смысл разбить весь набор данных на 7 частей по дням недели и работать с каждым новым набором в отдельности. Данный подход позволит в дальнейшем распределить обработку вновь поступающих данных на кластер из нескольких машин, каждая из которых будет обрабатывать меньшую часть данных, например, только один из дней недели [Shichkina, 2016].

Также имеет смысл ввести еще одну временную координату помимо дня недели – временная зона суток. Выбор подобной координаты обусловлен не только вопросом упрощения навигации по временной оси, но и для сглаживания погрешностей местонахождения устройства в определенной ячейке координатной сетки. Исходя из изначально поставленной задачи, можно варьировать шаг временной зоны суток для получения данных, наиболее полезных для аналитики. Например, анализируя подобный набор данных и устанавливая шаг временной зоны суток на 1 час, можно составить оптимальный график работы кассиров, чтобы минимизировать время ожидания покупателя в очереди. Таким же образом можно составить и график работы мерчендайзеров, чтобы покупатель всегда мог приобрести весь ассортимент товаров по актуальным ценам.

Таким образом, после трансформирования данных, будет получена новая структура данных, в которой поле *loc* содержит координаты устройства, поле *fileId* содержит идентификатор устройства, *dayTime* содержит количество полных минут, прошедших с начала суток, *weekDay* — порядковый номер дня недели, *timeZone* — номер временной зоны суток, *nSquare* — уникальный идентификатор сектора координатной сетки и поле *square* содержит координаты сектора в сетке:

```
{
"loc" : { "long" : 39.778412, "lat" : -86.269676 },
"fileId" : NumberLong(56),
"dayTime" : 1015,
"weekDay" : 7,
"timeZone" : 203,
"nSquare" : NumberLong(4994024),
"square" : { "i" : NumberLong(676), "j" : NumberLong(3074) },
}
```

После первой трансформации количество записей не уменьшилось, но количество полей изменилось и остались заполненными только необходимые и полезные данные.

Глубокая отчистка и трансформация данных методом применения нечетких срезов

После получения предварительно обработанных данных возникает задача отсечения записей, которых недостаточно для построения предсказательной модели и последующей кластеризации.

В нашем конкретном случае использовался метод нечетких срезов в базах данных. Данный подход позволяет реализовать не только кластеризацию данных, но также вычислить степень принадлежности к срезу, что в конечном итоге позволит отчистить данные, порог доверия к которым не удовлетворяет заявленным параметрам. Остается один вопрос - по каким параметрам необходимо выполнить срезы.

Исходя из того, что важные параметры для решения задачи уже были определены, логично будет реализовать фильтры по временной зоне и местоположению. Таким образом, применяя аппарат нечеткой логики [Zadeh, 1965] можно будет связать эти параметры воедино не прибегая к строгим ограничениям.

Формируя срез по местоположению, можно определить, какие транспортные средства с наибольшей вероятностью будут находиться в интересующем исследователя секторе координатной сетки. А срез по временной зоне поможет выделить из всех устройств те, которые будут находиться в этом секторе в интересующее время.

В конечном итоге, отчистка данных с помощью аппарата нечеткой логики позволит сформировать набор сеток для каждой временной зоны суток. Каждый набор будет содержать в себе координатную сетку (поле *square*) с актуальными данными для выбранного дня недели (поле *weekDay*) и временной зоны (поле *timeZone*). Эти данные включают в себя количество устройств (поле *devCount*) в каждом секторе сетки, их уникальные идентификаторы (поле *devices.fileId*) и степень принадлежности каждого устройства (поле *devices.devProb*), а также дисперсию степени принадлежности всех устройств, находящихся в секторе (поле *dispersion*):


```
{
"timeZone" : 122,
"weekDay" : 6,
"nSquare" : NumberLong(12596496),
"square" : { "i" : NumberLong(1704), "j" : NumberLong(4514) },
"devCount" : 1,
"dispersion" : 0,
"devices" : [ { "fileId" : NumberLong(1310830), "devProb" : 0.8081734778982486 } ]
}
```


Данные телеметрии транспортных средств получаются с некоторой временной периодичностью. Эта особенность позволяет получить степени принадлежности для каждого транспортного средства путем отбора тех, количество записей о которых встречается в пределах одного координатного сектора. Например, если телеметрия собирается ежесекундно и необходимо узнать, какие транспортные средства будут находиться в заданном секторе более минуты, то количество записей с одинаковым идентификатором транспортного средства, удовлетворяющее нашему запросу, будет не меньше 60. Для всех таких устройств степень доверия равна единице. Однако, устройства, записи о присутствии которых, имеются в меньшем количестве, тоже попадут в результирующую выборку, но уже с меньшим коэффициентом доверия.

Таким образом, определив коэффициент доверия для каждого из устройств в секторе, можно исключить из дальнейшего рассмотрения устройства, коэффициент принадлежности которых проходит ниже установленного порога.

Заключение

Проведя описанные выше операции очистки и трансформации данных над тестовым набором данных в рамках данного исследования, удалось из 2 267 163 140 исходных записей получить 8825 записей отчищенных данных, которые полностью готовы для применения сложных запросов к базе данных (рис.1.).

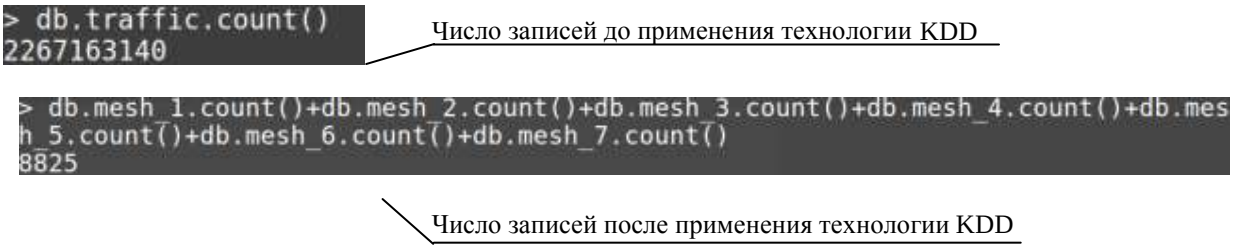

Рис.1. Количество записей до и после применения KDD

Предложенный подход к трансформации и отчистке данных позволяет сократить, в нашем случае, количество записей более чем в 256 000 раз.

Такое ощутимое уменьшение количества записей позволяет не только уменьшить затраты на хранение подготовленных данных, но и намного ускорить применение методов извлечения знаний.

Список литературы

Zadeh L. A. Fuzzy Sets // Information and Control. — 1965. — Vol. 8, No. 3 —P.338-353.

Banker K., Bakkum P., Verch S., Garrett D., Hawkins T. MongoDB in Action, Second Edition: Covers MongoDB version 3.0 // A K Peters/CRC Press. — March 2016. — ISBN 9781617291609. — 480 pages.

Piatetsky-Shapiro G., Frawley W. Knowledge Discovery In Databases // AAAI Press. — December 1991. — ISBN 9780262660709. — 539 pages.

Shichkina Y., Degtyarev A., Gushchanskiy D., Iakushkin O. Application of Optimization of Parallel Algorithms to Queries in Relational Databases // LNCS — 2016 — Vol.9787 — P.366-378.

Technology of cleaning and transforming data using the Knowledge Discovery in Databases (KDD) technology for fast application of Data Mining methods

Y. A. Shichkina [1,a], A. B. Degtyarev[2,b], A. A. Koblov[1,c]

[1] Saint Petersburg Electrotechnical University "LETI"
ul. Professora Popova 5, 197376 St. Petersburg, Russian Federation

[2] Saint-Petersburg State University,
University emb., 7-9, St.Petersburg

E-mail: [a] strange.y@mail.ru, [b] deg@csa.ru, [c] koblow.a.a@gmail.com

For large data amounts today brings a number of reasons. The increase in databases makes new and very serious hardware requirements of data centers and in recent years, it requires greater investment in hardware, software, the corresponding work and management. The decision of the main problems associated with the actual hardware infrastructure of data center, it must be said, it is much cheaper than the improvement of the software. But this is only a temporary solution. We must look for a global solution to the problem of large data. It is necessary to improve the methods of data processing with the help of the equipment that is available. This article discusses methods of cleaning and transforming data within the Knowledge Discovery in Databases technology for fast applying data mining techniques. In particular, the article shows how the method can significantly reduce the data selection for query building in noSQL databases on the example of MongoDB.

Keywords: big data, database, MongoDB, data cleaning, query