Лекция 10. Инструменты для анализа данных

Когда люди сталкиваются с бизнес-задачами, им нужны инструменты для преодоления проблем и поиска путей создания ценности. В аналитическом смысле это часто начинается с принятия решения о том, какой инструмент следует использовать. В этой лекции подробно описаны некоторые широко используемые инструменты.

WEKA

Weka (Waikato Environment for Knowledge Analysis) - это программное обеспечение для анализа данных с открытым исходным кодом, полностью реализованное на языке Java и разработанное в основном в Университете Вайкато, Новая Зеландия. Weka отличается широким набором чрезвычайно продвинутых алгоритмов обучения, графическим интерфейсом пользователя (GUI) и инструментами визуализации данных. Weka предоставляет пользователям доступ к своим сложным процедурам добычи данных через графический интерфейс, предназначенный для продуктивного анализа данных, интерфейс командной строки и интерфейс прикладного программирования Java (API). Однако Weka не очень хорошо масштабируется для анализа больших данных, поскольку она ограничена доступными ресурсами оперативной памяти, как правило, на одной машине. Пользователи с 64-разрядными операционными системами имеют доступ к гораздо большим ресурсам оперативной памяти, но документация Weka направляет пользователей к алгоритмам предварительной обработки и фильтрации данных для выборки больших данных перед анализом. Поскольку многие из наиболее мощных алгоритмов, доступных в Weka, недоступны в других средах без разработки собственного программного обеспечения, выборка может быть лучшей практикой для случаев использования, требующих применения Weka.

В последних выпусках Weka сделала шаг вперед в сторону многопоточности и простой многозадачности. В Weka 3.6 некоторые классификаторы могут одновременно обучать модели на нескольких складках перекрестной валидации, а развертывание сервера Weka позволяет выполнять задачи по анализу данных одновременно на одной машине или кластере. Рутины Weka также предоставляют расширенные аналитические компоненты для двух сред анализа больших данных на базе Java: Pentaho и MOA (Massive Online Analysis).

Weka - хороший вариант для людей, у которых может не быть опыта программирования на Java и которым нужно быстро начать работу, чтобы доказать свою ценность.

Языки JAVA и JVM

Для организаций, желающих разработать собственные аналитические платформы с нуля, Java и некоторые другие языки, работающие на виртуальной машине Java (JVM), являются обычным выбором. Для больших, параллельных и сетевых приложений Java предоставляет значительные преимущества в разработке по сравнению с языками более низкого уровня без чрезмерных жертв производительности, особенно в сфере аналитики больших данных. В то время как языки, выполняющиеся непосредственно на аппаратном обеспечении, в частности FORTRAN и C, продолжают превосходить Java в вычислениях в оперативной памяти и на центральном процессоре, технологические достижения платформы Java привели ее производительность почти в соответствие с родными языками для процессов

ввода/вывода и сетевых процессов, таких как те, которые лежат в основе многих приложений больших данных с открытым исходным кодом.

Вероятно, самая узнаваемая среда обработки больших данных на базе Java, Apache Hadoop, обошла несколько других технологий в конкурсе TeraByte Sort Benchmark в 2008 и 2009 годах. Это была первая технология на базе Java, победившая в этом уважаемом конкурсе бенчмарков. По мере того как аналитические приложения значительно увеличивались в масштабе и сложности, время разработки стало более серьезной проблемой, а производительность, связанная с памятью и процессором, стала менее важной.

В конкретном контексте создания приложений для добычи данных сильные стороны Java вытекают из ее эффективности разработки: богатые библиотеки, множество прикладных фреймворков, встроенная поддержка параллелизма и сетевых коммуникаций, а также уже существующая база открытого кода для функциональности добычи данных, например, библиотеки Mahout и Weka.

Несмотря на некоторую потерю производительности, преимущества разработки также связаны с платформой JVM. В дополнение к переносимости, JVM обеспечивает управление памятью, профилирование памяти и автоматическую обработку исключений.

Scala и Clojure - более новые языки, которые также работают на JVM и используются в приложениях для добычи данных. Scala - это язык с открытым исходным кодом, разработанный в Швейцарии в Федеральной политехнической школе Лозанны. Впервые он был выпущен в 2003 году и призван стать усовершенствованием Java.

Scala используется для создания приложений для добычи данных, поскольку она включает в себя эффективность разработки на Java - благодаря совместимости с Java Runtime Environment - и добавляет функциональное программирование, причем с более лаконичным синтаксисом. Scala позволяет разработчикам переключаться между парадигмами функционального и объектно-ориентированного программирования (ОО), в то время как грамматика и синтаксис Java, как правило, навязывают ОО-проектирование. Функциональные языки положительно оцениваются в приложениях для добычи данных, поскольку они работают с памятью иначе, чем ОО и процедурные языки. Например, основной проблемой в процедурных и ОО многопоточных приложениях является предотвращение одновременного изменения одной и той же переменной несколькими потоками. Функциональные языки программирования избегают этого, никогда не изменяя переменные. Несмотря на то, что это менее распространенная парадигма программирования, функциональное программирование является ценным инструментом, который не стоит сбрасывать со счетов. Scala набирает популярность и используется для реализации крупных проектов с открытым исходным кодом, таких как Akka и Spark. Akka - это набор инструментов для создания больших, параллельных и распределенных приложений на JVM, а Spark - высокопроизводительная среда для добычи данных из AMP Lab Калифорнийского университета Беркли. Scala также используется на коммерческой основе в различных целях компаниями FourSquare, Guardian, LinkedIn, Novell, Siemens, Sony, Twitter и другими.

Язык **Clojure** был написан Ричем Хикки и выпущен в 2007 году. Это функциональный язык программирования и диалект языка программирования Lisp, с дополнительными функциями скриптинга и параллелизма. Хотя Clojure не был разработан специально для приложений для работы с большими данными, эти присущие ему характеристики делают его отличным кандидатом для создания программного обеспечения для анализа данных. Clojure по умолчанию использует

неизменяемые структуры данных, устраняя целую категорию проблем, связанных с безопасностью потоков, по сравнению с процедурными и ОО языками. Clojure относится к данным очень целостно, предоставляя разработчикам множество удобных способов обработки обширного набора структур данных и наследуя концепцию кода как данных от Lisp.

Такой подход к последовательностям данных устраняет еще один класс распространенных ошибок программирования, связанных с граничными условиями, особенно по сравнению с C и FORTRAN. Clojure также унаследовал от Lisp простой синтаксис, ленивую оценку и высокоэффективный механизм макросов. Средства макросов позволяют создавать языки, специфичные для конкретной области (DSL), и уже созданы DSL для SQL (ClojureQL) и интеграции Hadoop (Cascalog). Когда уникальные преимущества Clojure объединяются с преимуществами платформы JVM и доступом к библиотекам Java, он становится отличным средством программирования общего назначения. Первыми пользователями Clojure, как правило, были небольшие, технологически или аналитически продвинутые компании и стартапы. Однако крупные корпорации и учреждения, такие как Citigroup и Институт Макса Планка, также сообщают об использовании Clojure.

R

R - это язык программирования с открытым исходным кодом четвертого поколения, предназначенный для статистического анализа. R вырос из коммерческого языка S, который был разработан в Bell Laboratories в конце 1970-х годов. R был перенесен из S и SPLUS - еще одной коммерческой реализации, лицензированной сейчас компанией TIBCO - исследователями из Оклендского университета (Новая Зеландия), и впервые появился в 1996 году.

R занял видное место в широком и растущем сообществе специалистов по науке о данных, широко используется в академической среде и находит все большее признание в частном секторе. Известно, что крупные компании, такие как Bank of America, Google, Intercontinental Hotels, Merck, Pfizer и Shell, используют R для различных целей. По результатам общего исследования языков программирования, проведенного TIOBE в сентябре 2021 года, R занял 18-е место по общей популярности языков разработки, приблизившись к MATLAB (16-е место) и перегнав SAS (23-е место) и (TIOBE, 2021). R также очень легко настраивается. Существуют тысячи расширений для R, по сравнению с примерно 1 600 в 2009 году. Пакеты расширений включают в базовые функции анализа R все: от анализа речи до геномной науки и анализа текста. R также может похвастаться впечатляющей графикой, бесплатными и отшлифованными интегрированными средами разработки (IDE), программным доступом ко многим языкам общего назначения, интерфейсами с популярными собственными аналитическими решениями, включая MATLAB и SAS, и даже коммерческой поддержкой от Revolution Analytics.

R переживает стремительный рост популярности и функциональности, и многие проблемы с памятью, которые затрудняли работу с большими данными в предыдущих версиях R, были решены. В настоящее время ведется значительная работа по обходу оставшихся ограничений памяти с помощью кластерных вычислений и интерфейсов к Hadoop. Короткий цикл обновления R сделал эту работу доступной для потребителей R через официальные пакеты, такие как snow, multicore и parallel, и молодые проекты, такие как RHadoop и RHIPE. За исключением Rhadoop и RHIPE, эти пакеты предназначены не для настоящего анализа больших данных, а для вычислений, требующих больших затрат процессора и памяти, "неловко параллельных", часто

выполняемых с помощью конструкции lapply() в R, где функция применяется ко всем элементам списка.

Snow используется для выполнения вычислений R на кластере компьютеров. Он использует сокеты, MPI, PVM или NetWorkSpaces для связи между узлами кластера. Пакет snow использует традиционную архитектуру распараллеливания master-worker, хранит результаты в памяти на главном узле и требует, чтобы аргументы для вызова функций snow помещались в памяти. Таким образом, будучи привязанным к памяти, пакет используется в основном для вычислений, требовательных к процессору, таких как моделирование, бутстраппинг и некоторые алгоритмы машинного обучения.

Multicore - это простой в установке и внедрении пакет, который разделяет последовательные процессы R, запущенные на одной POSIX-совместимой машине (OS X, Linux или UNIX), на многопоточные процессы на той же машине. Процессы, выполняемые одновременно с использованием многоядерности, ограничены одномашинной оперативной памятью и другими ограничениями общей памяти.

Parallel - это пакет параллельного выполнения, который входит в стандартную поставку R 2.14 и выше. Он расширяет возможности параллельного выполнения до уровня кластера для POSIX-совместимых операционных систем и до уровня мультипроцессора для Windows. Parallel имеет много общего в синтаксисе, функциональности и ограничениях со snow и multicore, а также может использовать объекты кластера snow.

Таким образом, большие вычислительные задачи могут быть решены более экономически эффективно за счет использования совокупной мощности и памяти многих компьютеров. Программное обеспечение очень портативно. NetWorkSpaces (NWS) - это мощный пакет программного обеспечения с открытым исходным кодом, который позволяет легко использовать кластеры из языков сценариев, таких как Matlab, Python и R. Хотя эти языки сценариев являются мощными и гибкими инструментами разработки, присущая им простота использования во многих случаях может привести к естественным ограничениям.

Rhadoop и **Rhipe** обеспечивают программный доступ к Hadoop, используя язык R. Задачи Map и reduce могут выполняться из R на кластере Hadoop.

PYTHON

С момента своего создания в конце 1980-х годов Python был задуман как расширяемый язык высокого уровня с большой стандартной библиотекой и простым, выразительным синтаксисом. Python можно использовать интерактивно или программно, и он часто применяется для написания сценариев, численного анализа, разработки ОО-приложений общего назначения и веб-приложений. Python является интерпретируемым языком и обычно выполняет задачи с интенсивными вычислениями медленнее, чем родные, компилируемые языки и языки JVM.

Однако программы на Python могут вызывать скомпилированные двоичные файлы FORTRAN, C и C++ во время выполнения через несколько документированных API. Программы на Python могут быть многопоточными, и, как и в языках JVM, платформа Python осуществляет управление памятью и обработку исключений, освобождая разработчика от этой обязанности. Язык Python также был перенесен на платформу JVM в проекте под названием Jython.

Общие сильные стороны программирования Python в сочетании с множеством библиотек баз данных, математических и графических библиотек делают его подходящим для проектов в области исследования и добычи данных. Свойственные Python возможности работы с файлами и текстами, а также возможность подключения к большинству баз данных SQL и NoSQL позволяют программам на Python с

относительной легкостью загружать и обрабатывать необработанные и табличные данные. Математические и статистические модели могут быть реализованы с помощью библиотек Scipy и Numpy, которые имеют сильную поддержку линейной алгебры, численного анализа и статистических операций. Более новые библиотеки Python, такие как Orange и Pattern, предоставляют высокоуровневые API для добычи данных, а библиотека Matplotlib позволяет создавать сложные и тонкие визуализации данных. Python также можно использовать для написания директив map и reduce для заданий MapReduce с помощью утилиты Hadoop Streaming. С появлением инструментария scikit-learn Python стал более широко использоваться в сообществах специалистов по анализу данных и науке о данных. В настоящее время этот инструментарий находится в релизе 0.14, но предлагает ряд полезных алгоритмов и методов манипулирования. Python scikit не имеет никакого пользовательского интерфейса, кроме IDE для программирования, что обеспечивает гибкость, но делает кривую первоначального обучения крутой и снижает производительность при выполнении рутинных задач.

SAS

SAS является ведущим аналитическим программным обеспечением на рынке. Система SAS была впервые продана в 1976 году, и двое из четырех ее основателей до сих пор работают в компании - доктор Джим Гуднайт и Джон Салл. Система SAS подразделяется на ряд областей продуктов, включая статистику, исследование операций, управление данными, механизмы для доступа к данным и бизнес-анализ (BI). Для нас наиболее актуальными продуктами являются SAS/STATR, SAS R Enterprise Miner™ и пакет текстовой аналитики SAS. В последних исследованиях Forrester Wave и Gartner Magic Quadrant компания SAS была признана ведущим поставщиком в области предиктивного моделирования и интеллектуального анализа данных. Эти рейтинги показывают широту и глубину возможностей программного обеспечения.

Система SAS делится на две основные области: процедуры для выполнения анализа и язык четвертого поколения, который позволяет пользователям манипулировать данными. Этот язык известен как DATA Step.

Одним из уникальных достижений SAS является обратная совместимость, которая сохраняется из выпуска в выпуск.

Система SAS обрабатывает данные в памяти, когда это возможно, а затем использует системные страничные файлы для эффективного управления данными любого размера. SAS внесла значительные изменения в свою архитектуру, чтобы лучше использовать преимущества вычислительной мощности и снижения цены за FLOP (операции с плавающей запятой в секунду) на современных вычислительных кластерах.

Список использованных источников:

1. Big Data, Data Mining, and Machine Learning: Value Creation for Business Leaders and Practitioners (Wiley and SAS Business Series) 1st Edition.