УДК [004.6+004.93.14]:378(476-25)

СИСТЕМА ОБРАБОТКИ БОЛЬШИХ ДАННЫХ НА ОСНОВЕ ВЫЧИСЛИТЕЛЬНОГО КЛАСТЕРА БГУИР

Д.И.Самаль
Заведующий кафедрой электронных вычислительных машин БГУИР, кандидат технических наук, доцент

А.И. Демидчук
Заведующий лабораторией высокопроизводительных вычислений БГУИР

Н.А. Искра
Старший преподаватель кафедры электронных вычислительных машин БГУИР, магистр технических наук

Д.Ю. Перцев
Ассистент кафедры кафедрой электронных вычислительных машин БГУИР

М.М. Татур
Профессор кафедры электронных вычислительных машин БГУИР, доктор технических наук, профессор

Белорусский государственный университет информатики и радиоэлектроники, Республика Беларусь
E-mail: Liv@bsuir.by

Аннотация. Выделены общие принципы построения систем анализа данных. Описана разработанная многоуровневая система исполнения алгоритмов Data Mining, на основе которой возможно построение полноценной системы анализа.

Ключевые слова: Data Mining, Анализ данных, Принятие решений, Big Data, Параллельные вычисления.

Введение. Для эффективного решения прикладных задач из области анализа данных необходимо принимать обоснованные решения о выборе способов работы с данными. Интересной представляется задача сравнения различных подходов в области анализа данных. Актуальной задача также является быстрое прототипирование решений и их повторное использование для групп родственных задач.

Для построения алгоритмов решения задач могут использоваться различные функции и операции, предоставляемые такими проектами по анализу и обработке в области Data Mining, как TensorFlow[1], Theano[2], Weka[3]. Также существует множество проектов на Python, C++, каждый из которых имеет свои плюсы и минусы. Например, реализации с использованием scikit-learning[4] не поддерживают работу с графическим процессором, но включают большое число уже готовых к использованию алгоритмов. Проекты TensorFlow, Theano наоборот включают ограниченное число уже готовых алгоритмов, но предоставляют интерфейс для авторских разработок, кроме того, поддерживается обработка вычислений на GPU.

В данной работе описываются аспекты разработки учебно-исследовательской системы, позволяющей унифицировать интерфейс доступа к различным реализациям алгоритмов анализа данных, динамически формировать цепочки вызовов обработчиков, собирать и анализировать статистику исполнения, оценивать эффективность применения того или иного метода.

Общие принципы анализа данных. Большинство способов решения задач из области анализа данных сводятся к определенной последовательности действий[5] (рис.1).

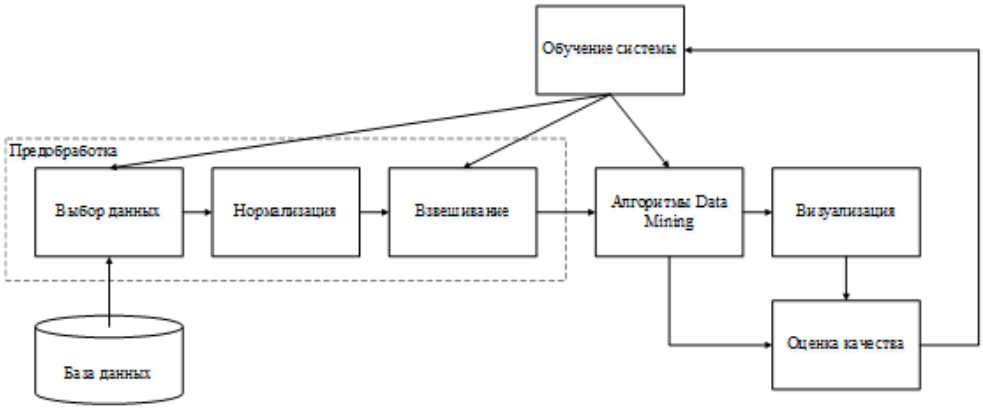

Рисунок 1. Общая последовательность решения задачи в области Data Mining

На первом шаге выполняется выборка исходных данных (в том числе фильтрация, проверка на правильность формата, полноту и т.д.), которая может располагаться в распределенной базе данных, в HDFS или на жестком диске компьютера.

Следующим этапом является нормализация и оценка значимости (взвешивание) данных, выбранных на предыдущем шаге.

После того как данные были подготовлены, выполняется их непосредственный анализ. На данном этапе применяется один или более алгоритмов Data Mining. Для упрощения восприятия результата применяются алгоритмы визуализации, позволяющие в наглядной форме отобразить результат работы.

Финальным этапом обработки некоторого множества данных является оценка полученного результата. При этом принимается решение о качестве полученных результатов, их достоверности и надежности. В зависимости от типа применяемых алгоритмов возможна корректировка входных параметров (обучение) и повторный запуск описанной выше последовательности.

Оценка качества методов анализа данных. Как отмечалось выше, важной проблемой является оценка и сравнение методов анализа данных. В нашем случае под точностью вычислений понимается величина, которая определяется математическим методом и алгоритмом вычислений, репрезентативностью обучающей выборки, критерием принятия решений и т. п.

В случае многопроцессорной обработки больших объемов данных время вычисления тестовой задачи определяется с момента подачи входных данных до получения результата. Этот параметр может измеряться как в абсолютных величинах, так и в условных – модельных тактах. В первом случае на оценку будут влиять технические характеристики аппаратной платформы (тактовая частота процессора, время доступа к данным в оперативной памяти и др.), во втором случае – способ определения модельного времени. Производительность в рассматриваемом случае будет представлена в задачах кластеризации и (или) классификации и т. д. в единицу времени. Очевидно, что это время будет складываться как из времени непосредственных вычислений на параллельном сопроцессоре, так и времени «накладных расходов», связанных с загрузкой либо выгрузкой данных сопроцессора, а также реализации участков программ, не поддающихся распараллеливанию, подготовительных операций, сервисных

функций и т. п.

Влияние указанных параметров друг на друга и будет характеризовать эффективность вычислений[6]. Графики качественных зависимостей такого влияния для различных аппаратных платформ показаны на рис. 2.

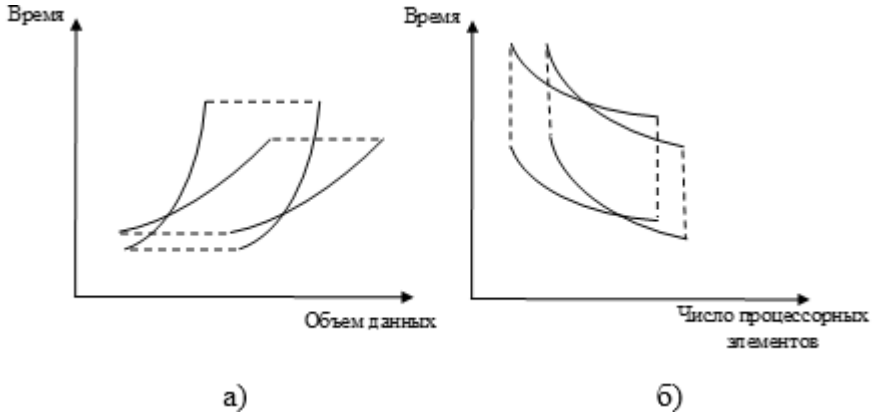

Рисунок 2. Качественные зависимости времени вычислений тестовой задачи: а) от объема обрабатываемых данных; б) от числа процессорных элементов

На рис. 2.а изображен вид экспоненциальной зависимости, поскольку задачи анализа данных носят переборный характер. Однако характер наклона может существенно изменяться в зависимости от того, насколько качественно выполнено распараллеливание алгоритмов применительно к конкретной аппаратной платформе или насколько удачно спроектирована архитектура проблемно-ориентированной машины.

На рис. 2.б показано проявление закона Амдаля – Густафсона, согласно которому производительность параллельной машины не может возрастать линейно с увеличением числа процессорных элементов. Между тем наклон этой кривой может значительно изменяться в зависимости от используемой архитектуры процессоров и технологий вычислений. Для обоих графиков полагается, что другие существенные параметры (точность, объем данных или число процессорных элементов) принимаются постоянными.

На рис. 2 одноименные кривые образуют некоторый диапазон пространства, в котором может проходить конкретная кривая, а ее реальное положение будет определено техническими характеристиками аппаратной платформы.

Система интеллектуального анализа данных. Несмотря на то, что представленная выше последовательность операций (рис.1) является универсальной, на практике на каждом шаге могут применяться алгоритмы из различных библиотек. Необходимость этого может быть связана с применением более качественной реализации, применением специализированных программных или аппаратных решений (например, применение тензоров в NVIDIA CUDA или Google Tensor Processing Unit).

Основными критериями при разработке системы анализа данных были выбраны следующие:

– создание единого интерфейса доступа к конкретным реализациям самих алгоритмов, независимо от библиотек, в которых они были реализованы;

– предоставление возможности пошаговой настройки последовательности операций, выполняемых над данными;

– формирование шаблона решения задачи с возможностью ее последующего применения;

– оценка возможностей и ограничений различных реализаций алгоритмов анализа данных.

При разработке системы сформировалось 3 программных уровня (рис.3):

– уровень сервисов, обеспечивающий взаимодействие с конкретными библиотеками алгоритмов;

– уровень алгоритма анализа данных, взаимодействующий с сервисами;

– Web-интерфейс, через который предоставляет доступ конечному пользователю (интерфейс пользователя).

Уровень сервисов.

Библиотеки алгоритмов анализа данных (Theano[2], Weka[3], scikit-learn[4], MLlib[7]) предоставляют собственный интерфейс доступа, поддерживают разные языки программирования, при этом эффективно справляясь с определенной задачей (таблица 1).

Ключевой идеей уровня сервисов является:

– создание сервиса для каждой поддерживаемой библиотеки, ее подключение либо информирование о ее отсутствии или неполадках;

– инициализация библиотеки;

– проверка корректности входных параметров;

– предоставление интерфейса доступа к поддерживаемым функциям.

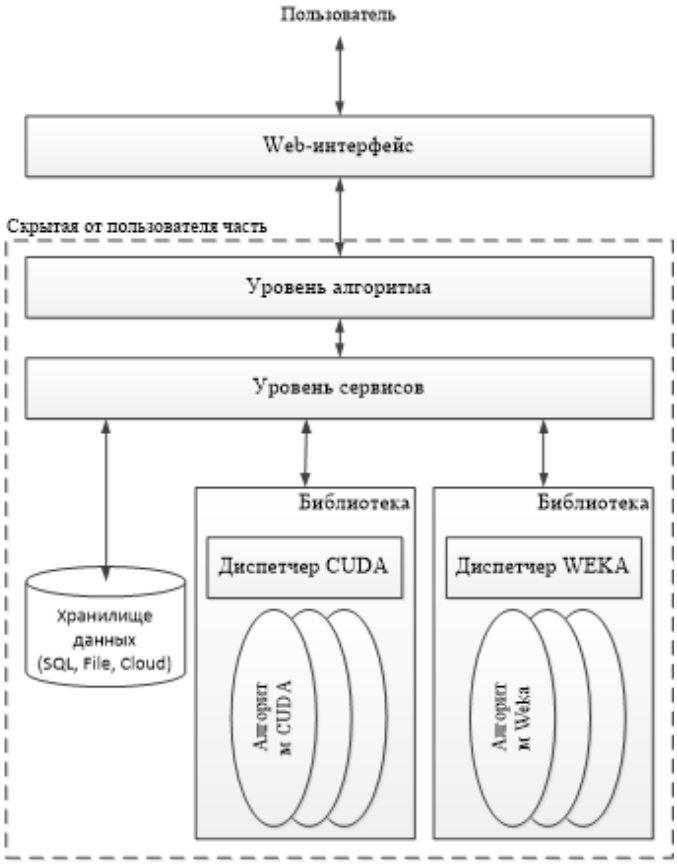

Рисунок 3. Структурная схема системы интеллектуального анализа данных

Важной особенностью данного уровня является формирование единого API, независящего от самой библиотеки. Например, алгоритм кластеризации k-means представлен в библиотеках Weka и MLlib. При подключении соответствующих библиотек, в каждой из них реализуется метод KMeans с одинаковыми входными и выходными параметрами. На текущий момент выполняется экспериментальное подключение библиотек Weka, MLlib и собственных разработок.

Таблица 1

Сравнительный обзор библиотек для анализа данных

Библиотека	Готовые алгоритмы	Работа с большими данными	Параллельные вычисления
Theano	Нет	Нет	Да
Sklearn	Да	Нет	Нет
MLlib	Да	Да	Частично

Уровень алгоритма анализа данных.

Данный уровень обобщает и систематизирует поддерживаемые алгоритмы анализа данных, предоставляет соответствующую информацию Web-интерфейсу для ее отображения. Его основными задачами является:

– получение информации о подключенных сервисах;

– получение информации о поддерживаемых алгоритмах в рамках каждого сервиса;

– обобщение и систематизация полученной информации;

– формирование единого прозрачного интерфейса доступа к сети сервисов.

По мере сбора статистики и анализа результатов на данном уровне возможно внедрение модуля выбора оптимальной версии алгоритма с учетом пожеланий пользователя.

Например, данный уровень формирует запрос на подключение библиотек Weka и MLlib. В случае если ниже лежащий уровень сообщил об успешной инициализации соответствующего сервиса, формируется запрос на поддерживаемые алгоритмы обработки и некоторую статистическую информацию (например, о технических ограничениях, требуемых настройках). Уровень алгоритма анализа данных обобщает полученные данные и передает информацию Web-интерфейсу. Если найдены сервисы, поддерживающие один и тот же алгоритм (например, алгоритм k-means), формируется единый интерфейс доступа. При данном подходе возможна реализация автоматического выбора оптимальной версии алгоритма на основе полученной статистической информации (при условии, что пользователь четко не определил, какую версию необходимо использовать).

Web-интерфейс.

Конечный уровень – Web-интерфейс, основными задачами которого являются:

– удобное предоставление полученной от предыдущих уровней информации для последующего использования;

– получение инструкций от пользователя и их передача на сервер;

– предоставление полученных результатов работы;

– контроль прав доступа.

В результате взаимодействия пользователя с интерфейсом формируется цепочка операций с необходимыми настройками, данная информация передается на сервер. На сервере полученная цепочка раскручивается и передается на второй уровень системы для получения результатов работы.

Учитывая универсальность реализации второго уровня, в перспективе возможно создание клиентского приложения, работающего на основе клиент-серверной архитектуры, и замещающего работу Web-интерфейса. Дополнительной альтернативой могут быть плагины к наиболее распространенным средам разработки, в т.ч. мобильное приложение.

Пример использования системы.

Разрабатываемая и представленная система поддерживает пошаговое формирование и настройку операций, необходимых для решения задач в Data Mining. При этом каждый этап обработки является отдельным алгоритмом, реализованным в той или иной библиотеке. В качестве источника данных на момент написания статьи поддерживается только CSV-файл, размещенный в файловой системе HDFS. Выборку данных и их предобработку можно выполнить при загрузке фрагмента данных либо при просмотре результирующего flowchart.

В качестве примера работы системы на рис.4 показан завершающий этап цепочки операций для алгоритма кластеризации k-means. Интерфейс определяет следующую цепочку операций: загружаемый файл (вкладка Data load), выборка данных и их взвешивание (вкладка Select features, выбранная в качестве активной на рисунке), затем будет вызван алгоритм кластеризации k-means, после чего в качестве примера вызывается PCA. Переключаясь между вкладками, Data Scientist может скорректировать любой из этапов обработки. При нажатии на кнопку Execute данная последовательность сохраняется для последующего применения, и формируются инструкции для исполнения на стороне сервера. Результат вычислений будет доступен в кабинете пользователя.

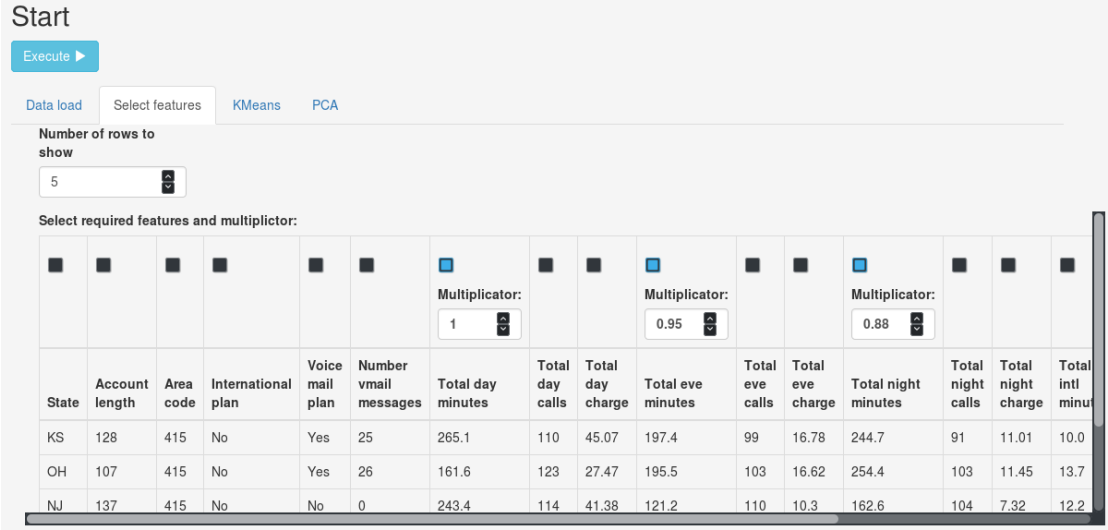

Рисунок 4. Пример работы системы (результирующий Flowchart)

Заключение. В представленной статье описан результат разработки единой системы анализа больших данных, приводится описание основных функциональных элементов.

Достоинствами представленной версии системы является:

– универсальность и гибкость, что позволяет выполнить быстрое подключение и настройку любой библиотеки алгоритмов;

– единый пошаговый интерфейс. От пользователя требуется единожды сформировать цепочку операций, выполнить необходимую настройку, после чего проект будет сохранен в шаблонах и доступен для выбора;

– архитектура системы допускает перспективную реализацию надстроек к популярным средам разработки (например, к Eclipse).

Дальнейшая разработка предполагает отладку и тестирование системы анализа больших данных, внедрение дополнительных библиотек алгоритмов.

Список литературы

[1]. TensorFlow [Electronic resource]. – Mode of access: https://www.tensorflow.org. – Date of access: 01.03.2018.

[2]. Theano 1.0 [Electronic resource]. – Mode of access: http://deeplearning.net/software/theano. – Date of access: 01.03.2018.

[3]. Weka 3: Data Mining Software in Java [Electronic resource]. – Mode of access: https://www.cs.waikato.ac.nz/ml/weka. – Date of access: 01.03.2018.

[4]. scikit-learn: machine learning in Python [Electronic resource]. – Mode of access: http://scikit-learn.org. – Date of access: 01.03.2018.

[5]. Witten, I.A. Data Mining, Fourth Edition: Practical Machine Learning Tools and Techniques / I.A. Witten. – Morgan Kaufmann, 2016. – pp.654.

[6]. Татур, М.М. Особенности построения вычислителей интеллектуальной обработки данных / М.М. Татур // Информатика. – 2015. – № 1(45). – с.39-44.

[7]. MLlib | Apache Spark [Electronic resource]. – Mode of access: https://spark.apache.org/mllib. – Date of access: 01.03.2018.

THE BIG DATA PROCESSING SYSTEM BASED ON BSUIR CLUSTER

D.I.SAMAL, PhD
Head of the Department of Electronic Computing Machines of the BSUIR, Associate Professor

A.I. DEMIDCHUK
Head of the High Performance Computing Laboratory of the BSUIR

N.A. ISKRA
Senior lecturer of the Department of Electronic Computers of the BSUIR, Master of Technical Sciences

D.YU. PERTSEV
Assistant of the Department of the Chair of Electronic Computers of the BSUIR

M.M. TATUR,
Doctor of Technical Sciences
Professor of the Department of Electronic Computers BSUIR, Professor

Belarusian State University of Informatics and Radioelectronics, Republic of Belarus
E-mail: samal@bsuir.by.

Abstract. In this paper general principles of data analysis systems creation are described. A multi-level system for performing analytical calculations on large data volumes that supports a variety of methods and approaches to data analysis is proposed

Key words: Data Mining, Data Analysis, Decision Making, Big Data, Parallel Computing