сплайн функции наиболее эффективны при решении задач математической физики, в которых конформно-плоская метрика присутствует естественным образом (например, в задачах томографии, геофизики, акустики, интегральной геометрии).

В работе построен программный комплекс в среде MatLab, а также независимо программный комплекс на языке СИ для интерполяции функций многих переменных конформно-плоскими сплайн функциями.

Работа выполнена при финансовой поддержке Российского фонда фундаментальных исследований Совета по грантам Президента РФ для поддержки молодых ученых и ведущих научных школ Российской Федерации (код проекта НШ-6613.2010.1), а также при поддержке ФЦП Научные и научно-педагогические кадры инновационной России на 2009-2013 гг. (гос. контракт № 02.740.11.0457).

Библиографический список

1. Завьялов Ю.С, Квасов Б.И., Мирошничеико В.Л. Методы сплайн-функций. – М.: Наука, Главная редакция физико-математической литературы, 1980.

2. Гладунова О.П., Родионов Е.Д., Славский В.В. Выпуклые многогранники пространства Лобачевского и интерполяция функций // Доклады академии наук – 2011. – Т. 441. – №6. – С. 1–4.

3. Гладунова О.П., Родионов Е.Д., Славский В.В. Конформные сплайн-функции // Метрическая геометрия поверхностей и многогранников: сборник тезисов Международной конференции, посвященной 100-летию со дня рождения Н.В. Ефимова, Москва; 18-21 августа 2010 г. – М.: МАКС Пресс, 2010. – С. 18–19.

Построение регрессии на главные компоненты в архитектуре MapReduce

П.В. Нуждин, С.И. Жилин
АлтГУ, г. Барнаул

Одним из продуктивных подходов к обработке больших массивов данных на вычислительных кластерах, является опубликованная в 2004 компанией Google концепция MapReduce [1], представляющую собой высокоуровневую модель распределенных вычислений, ориентированную на задачи, допускающие распараллеливание по данным. Модель MapReduce требует от пользователя лишь определения содержания работы на вычислительных узлах и позволяет абстрагироваться от вопросов технического характера (распределение вычислительной

нагрузки, востановление после сбоев, и т.п.), возлагая их решение на исполняющую среду. Имеются свободно распространяемые реализации вычислительной среды с архитектурой MapReduce. Наиболее известным продуктом этого рода является Java-фреймворк с открытыми исходными кодами Apache Hadoop [2]. Hadoop широко используется для обработки большого объема данных, а компания Amazon предоставляет возможность аренды Hadoop-кластера [3].

Задача построения регрессии на главные компоненты (РГК) [4] представляет собой задачу восстановления линейной зависимости с некоторой ошибкой между предикторами и зависимой переменной, где, используя метод главных компонент (МГК), в пространстве предикторов предварительно выполняется устранение мультиколлинеарности и понижение размерности. Устранение мультиколлинеарности является необходимым этапом для нахождения устойчивого решения в задаче линейной регрессии.

Существующие алгоритмы решения задачи РГК имеют полиномиальную сложность. Несмотря на это решение конкретных практических задач может занимать недопустимо много времени при обработке большого объема данных.

Целью настоящей работы является алгоритм построения регрессии на главные компоненты для обработки больших данных в архитектуре MapReduce и его реализация для фреймворка Apache Hadoop.

Основная идея алгоритма заключается в распределении компонентов сумм для вычисления эмпирической ковариационной матрицы по узлам вычислительной сети в соответствии с моделью MapReduce.

Алгоритм состоит из следующей последовательности шагов:

1. Центрирование и шкалирование (по необходимости) данных.
2. Вычисление эмпирической ковариационной матрицы.
3. QL-разложение ковариационной матрицы.
4. Выбор главных компонент из числа собственных вектор.
5. Перевод данных в новый базис из главных компонент.
6. Построение модели линейной регрессии для данных в новом базисе.

Все задачи были распараллелены в модели MapReduce, за исключением задачи нахождения собственных чисел и векторов ковариационной матрицы, время решения которой последовательным алгоритмом, реализованным в Apache Commons Math [5], для данных с количеством переменных до 1000, на современном оборудовании занимает менее минуты. Это время сопоставимо с накладными расходами на запуск задачи в Apache Hadoop.

Алгоритм РГК в модели MapReduce реализован для фреймворка Apache Hadoop и испытан в синтетическом тесте (1000000 образцов,

500 переменных), а также на данных спектрометрической съемки в инфракрасном диапазоне (1955731 образцов, 148 переменных), произведенной вдоль русла одной из рек Норвегии. Испытания производились на арендованном у компании Amazon кластере Hadoop, состоящем из 10 вычислительных узлов. Алгоритм показал приемлемое ускорение, как в синтетическом тесте, так и на реальных данных.

Библиографический список

1. Dean J., Ghemawat S. Simplified data processing on large clusters // Operating Systems Design and Implementation. – 2004. – P. 137–149.

2. Apache Hadoop. [Электронный ресурс] Режим доступа – http://hadoop.apache.org.

3. Amazon Elastic MapReduce. [Электронный ресурс] Режим доступа – http://aws.amazon.com/elasticmapreduce/.

4. Дрейпер Н. Прикладной регрессионный анализ. – М.: Финансы и статистика, 1987. – 717 с.

5. Apache Commons Math. [Электронный ресурс] Режим доступа – http://commons.apache.org/math/.

Распределенные алгоритмы построения интервальной регрессии

В.Д. Пятков, С.И. Жилин
АлтГУ, Барнаул

Метод построения регрессионной зависимости по экспериментальным данным при интервальной ошибке в выходной переменной [1] довольно широко используется в практике эмпирического моделирования и для краткости именуется *интервальной регрессией* (ИР). Суть метода сводится к оцениванию множества допустимых значений параметров регрессии, совместных как с используемой моделью регрессии, так и с набором ограничений, вытекающих из интервального характера ошибки наблюдения выходной переменной.

При построении наиболее употребимой линейной по параметрам регрессии задача оценивания множества допустимых значений параметров сводится к решению нескольких задач линейного программирования. Поэтому вычислительная сложность алгоритма построения ИР определяется эффективностью методов линейного программирования: при увеличении числа наблюдений время построения модели, как минимум, растет полиномиально. При обработке большого объема данных распределение процесса построения модели ИР на несколько