Компьютерная Математика

Системный анализ

УДК 519.217.268

Б.А. БЕЛЕЦКИЙ, Н.А. ГУПАЛ

РАСПРЕДЕЛЕННЫЕ БАЙЕСОВСКИЕ ПРОЦЕДУРЫ РАСПОЗНАВАНИЯ ТЕКСТОВОЙ ИНФОРМАЦИИ

Для распознавания текстовой информации рассмотрены две версии байесовской оптимальной процедуры. Описаны распределенные реализации этих версий на основе MapReduce.

Введение. Известно, что сложные задачи машинного обучения, как правило, требуют построения обучающих выборок большого объема. Эмпирические наблюдения показывают, что объемы глобально накапливаемых и обрабатываемых данных возрастают экспоненциально, удваиваясь приблизительно каждые 20 месяцев [1]. Согласно закону Мура, вычислительная мощность процессоров также возрастает экспоненциально, удваиваясь каждые 18 месяцев [2]. Такой баланс позволял относительно эффективно обрабатывать накапливаемые данные для решения различных задач. Однако к 2010 году обнаружилось, что закон Мура перестал выполняться из-за достижения производителями процессоров технологических пределов для текущей архитектуры процессоров. Для обработки стремительно возрастающих объемов данных требовались новые подходы к организации вычислений. Одним из предложенных подходов стало горизонтальное масштабирование, которое заключается в использовании множества соединенных между собой, относительно простых вычислительных устройств для совместного хранения и обработки данных. Обрабатываемые данные должны храниться и обрабатываться всеми узлами вычислительной системы, при этом работоспособность системы должна сохраняться при динамическом изменении ее топологии. Подход горизонтального масштабирования связан с рядом концептуальных и технических вызовов. Во-первых, принципы организации распределенных вычислений

значительно отличаются от принципов построения классических программ, рассчитанных на последовательное выполнение. Распределенные вычисления выполняются параллельно на множестве вычислительных устройств, что приводит к утрате детерминированности и значительно усложняет разработку и анализ таких алгоритмов. Во-вторых, получен ряд негативных результатов, касающихся концептуальных ограничений присущих распределенным системам (CAP-теорема [3]). В-третьих, пропускная способность сети оказалась серьезным ограничивающим фактором для производительности распределенных вычислительных систем.

Предложен целый ряд подходов, упрощающих построение горизонтально масштабируемых алгоритмов. Одним из популярных на сегодня подходов является модель программирования MapReduce [4], позволяющая относительно просто масштабировать достаточно богатый класс алгоритмов, среди которых и методы машинного обучения.

Распределенные вычисления на основе модели MapReduce. Рассмотрим основные принципы, на которых основывается модель программирования MapReduce. Пусть A, $|A|<\infty$ и B, $|B|<\infty$ – исходный и результирующий типы данных соответственно, и $f: A^l \mapsto B$ – вычислимая функция или алгоритм, обрабатывающий распределенную коллекцию $(a_1,...,a_l) \in A^l$ элементов исходного типа данных и возвращающий элемент результирующего типа данных $f(a_1,...,a_l) \in B$. Для вычисления значения функции $f(a_1,...,a_l)$ в распределенном режиме с помощью MapReduce достаточно представить ее в виде композиции пары специальных операций: $m(\cdot)$ и $\cdot \oplus \cdot$, заданных своими алгебраическими свойствами

$$f(a_1,...,a_l) = m(a_1) \oplus m(a_2) \oplus ... \oplus m(a_l), \tag{1}$$

где $m: A \mapsto B$ – детерминированная функция, преобразующая отдельные элементы $a_i \in A$, $i = \overline{1,l}$ распределенной коллекции в частичные результаты $m(a_i) \in B$. Для получения конечного результата частичные результаты $m(a_i)$ агрегируют с помощью бинарной операции $\oplus: B \times B \mapsto B,$, удовлетворяющей условиям:

$$b_1 \oplus b_2 = b_2 \oplus b_1, \ \forall b_1, b_2 \in B \text{ (коммутативность)}, \tag{2}$$

$$(b_1 \oplus b_2) \oplus b_3 = b_1 \oplus (b_2 \oplus b_3), \ \forall b_1, b_2, b_3 \in B \text{ (ассоциативность)}, \tag{3}$$

$$\exists e \in B: b \oplus e = b, \ \forall b \in B \text{ (существование нейтрального элемента)}. \tag{4}$$

Используя свойства (2) – (4), схему вычислений можно перестроить с сохранением конечного результата таким образом, чтобы частичные результаты $m(a_i)$ вычислялись параллельно на всех узлах вычислительной системы, а их агрегация выполнялась в первую очередь среди элементов одноименных узлов системы, минимизируя нагрузку на сеть при передаче данных между узлами.

Существуют различные реализации модели программирования MapReduce. Рассмотрим одну из таких реализаций – Apache Spark, содержащую специальную библиотеку горизонтально масштабируемых методов машинного обучения, среди которых: дерево принятия решений, случайный лес, метод опорных векторов, байесовский метод, нейронные сети и многие другие методы обучения с учителем и без него. Для байесовского метода доступны две вариации: бернуллиевская и мультиномиальная, которые успешно применяются для решения задач распознавания текстов [5].

Байесовская процедура распознавания текстов. Пусть D – конечное множество объектов произвольной природы, Y – конечное множество меток или классов, ассоциируемых с этими объектами, обозначим множество пар объект-метка $E = D \times Y$. Имеется обучающая выборка таких пар $\tau = (e_1, ..., e_l) \in E^l$, содержащая результаты l независимых реализаций случайного эксперимента с множеством элементарных исходов E и неизвестным распределением вероятности.

Задача обучения с учителем сводится к восстановлению неизвестного распределения путем построения эффективного в некотором смысле классификатора $g : X \mapsto Y$, вычисляющего метку $g(x(d)) \in Y$ объекта $d \in D$ по его описанию $x(d) \in X$, где X – множество описаний объектов $|X| < \infty$. В случае байесовской процедуры обучения классификатор строится в виде

$$g(x) = \arg\max_{y \in Y} P(x \mid y) P(y), \tag{5}$$

где $P(x \mid y)$ – условная вероятность описания $x \in X$ в классе $y \in Y$, $P(y)$ – априорная вероятность класса $y \in Y$.

Сложные объекты $d \in D$ описываются целым набором признаков, $x(d) = (x_1, ..., x_n) \in X$. Предположение об условной независимости вероятности появления признаков в описании объекта в каждом из классов $y \in Y$ позволяет представить условную вероятность описания объекта в виде произведения

$$P(x_1, ..., x_n \mid y) = \prod_{i=1}^{n} P_i(x_i \mid y), \tag{6}$$

где $P_i(x_i \mid y)$ – условная вероятность появления i-го признака со значением x_i в классе y. При вычислениях вероятности $P_i(x_i \mid y)$ и $P(y)$ в (6) заменяются соответствующими оценками $\hat{P}_i(x_i \mid y)$ и $\hat{P}(y)$, а классификатор (3) принимает вид

$$g(x) = \arg\max_{y \in Y} [\ln \hat{P}(y) + \sum_{i=1}^{n} \ln \hat{P}_i(x_i \mid y)]. \tag{7}$$

Различные вариации байесовской процедуры возникают при различных интерпретациях вероятностей $P_i(x_i \mid y)$, $P(y)$ и связанных с ними случайных экспериментов.

Бернуллиевская и мультиномиальная байесовские процедуры обучения. Классификация текстов – одна из областей успешного применения машинного обучения, и байесовского подхода, в частности. Рассмотрим несколько вариаций байесовской процедуры машинного обучения доступных в библиотеке Apache Spark.

При классификации текстов в качестве множества объектов D выступает множество конечных текстов, состоящих из слов из некоторого словаря V, $|V|<\infty$. Для простоты изложения полагаем, что слова в V отсортированы в лексикографическом порядке, и имеют уникальные порядковые номера, с которым они отождествляются. Текстам соответствуют метки из конечного множества меток Y, которые необходимо научиться предсказывать, используя обучающую выборку размеченных текстов. Примеры задач классификации текста: определение тональности сообщения (негативная, позитивная, нейтральная), или темы новостного сообщения (социология, спорт, политика, наука). Рассмотрим, каким образом байесовские процедуры обучения применимы в этом случае.

Бернуллиевская байесовская процедура обучения. Данная процедура основывается на описании текста $d \in D$ в классе $y \in Y$ в виде n, $n=|V|$ независимых бернуллиевских случайных величин, вероятность успеха каждой из которых соответствует вероятности появления каждого из слов из алфавита V в тексте. Текст $d \in D$ описывается булевым вектором $x(d)=(x_1(d),...,x_n(d)) \in X$, $x_i(d) \in \{0,1\}$, $i=\overline{1,n}$, причем $x_i(d)=1$, если слово $i \in V$ встречается в $d \in D$, в противном случае $x_i(d)=0$.

При этом условные вероятности (6) принимают вид

$$P(x_1,...,x_n \mid y)=\prod_{i=1}^{n}[p_{iy}x_i+(1-p_{iy})(1-x_i)], \tag{8}$$

где p_{iy} – вероятность появления слова $i \in V$ в классе $y \in Y$ (доля текстов класса $y \in Y$, которые содержат слово $i \in V$).

При расчетах классификатор (5) принимает вид

$$g_{\pi,\theta}(x_1,...,x_n)=\arg\max_{y \in Y}\pi_y\prod_{i=1}^{n}[\theta_{iy}x_i+(1-\theta_{iy})(1-x_i)], \tag{9}$$

где $\pi \in \mathbb{R}^{|Y|}$ – вектор оценок априорных вероятностей классов $P(i)$, а $\theta \in \mathbb{R}^{n \times |Y|}$ – матрица оценок условных вероятностей p_{ij} слов в классах. Задача построения классификатора (8) на этапе обучения сводится к вычислению вектора π и матрицы θ по обучающей выборке $\tau=(e_1,...,e_l)$. Для этого необходимо обработать объемный массив обучающих данных. Эту задачу можно выполнить в распределенном режиме с помощью MapReduce.

Элементы вектора оценок априорных вероятностей классов $\pi \in \mathbb{R}^{|Y|}$ вычисляются по выборке $\tau = (e_1,...,e_l)$ в виде

$$\pi_i(\tau) = \frac{c_i(\tau)}{l}, \tag{10}$$

где $c_i(\tau)$ – количество примеров класса $i \in Y$ в обучающей выборке $\tau \in E^l$. Вектор $(c_1(\tau),...,c_{|Y|}(\tau)) = c(e_1,...,e_l)$ можно вычислить с помощью MapReduce, представив вычисляющую его вектор-функцию $c: E^l \mapsto \mathbb{Z}^{|Y|}$ в виде

$$c(e_1,...,e_l) = m(e_1) \oplus m(e_2) \oplus ... \oplus m(e_l). \tag{11}$$

Операция $m: E \mapsto \{0,1\}^{|Y|}$ ставит в соответствие обучающему примеру $e_k \in E$ булев вектор $m(e_k) \in \{0,1\}^{|Y|}$ с единицей на позиции, соответствующий номеру класса обучающего примера т. е.

$$m_i(e_k) = ¢(j, y(e_k)), \tag{12}$$

где $y(e_k)$ – метка (ее порядковый номер) обучающего примера e_k, а $\delta(\cdot,\cdot)$ – дельта-функция

$$\delta(x, y) = \begin{cases} 1, x = y \\ 0, x \neq y \end{cases}. \tag{13}$$

Операция $\oplus: \{0,1\}^{|Y|} \times \{0,1\}^{|Y|} \mapsto \mathbb{Z}^{|Y|}$ агрегации пары частичных результатов является стандартной операцией поэлементного сложения векторов. Несложно убедиться, что условия (2) – (4) выполняются, что позволяет применять модель программирования MapReduce.

Элементы матрицы $¯ \in \mathbb{R}^{n \times |Y|}$ оценок условных вероятностей вычисляются по выборке $² = (e_1,...,e_l) \in E^l$ в виде

$$\theta_{ij}(\tau) = \frac{c_{ij}(\tau)}{c_i(\tau)}, \tag{14}$$

где $c_{ij}(\tau)$ – количество обучающих примеров класса $j \in Y$, содержащих слово $i \in V$ в $\tau \in E^l$, $c_i(\tau)$ – количество примеров класса $i \in Y$ в $\tau \in E^l$. Матрица $c(\tau) \in \mathbb{Z}^{n \times |Y|}$ строится в распределенном режиме с помощью MapReduce, учитывая представление

$$c(e_1,...,e_l) = m(e_1) \oplus m(e_2) \oplus ... \oplus m(e_l), \tag{15}$$

где операция $m: E \mapsto \mathbb{Z}^{n\times|Y|}$ ставит в соответствие обучающему примеру e_k, $k=\overline{1,l}$ разреженную матрицу $m(e_k)\in\mathbb{Z}^{n\times|Y|}$, содержащую в $y(e_k)$-том столбце описание $x(d(e_k))\in\mathbb{Z}^n$ текста $d(e_k)\in D$ из обучающего примера $e_k\in D\times Y$,

$$m_{ij}(e_k)=x_i(d(e_k))¢(j,y(e_k)). \tag{16}$$

Тогда операция $\oplus:\mathbb{Z}^{n\times|Y|}\times\mathbb{Z}^{n\times|Y|}\mapsto\mathbb{Z}^{n\times|Y|}$ агрегации пары частичных результатов является стандартной операцией поэлементного сложения матриц, для которой выполняются условия (2) – (4), что позволяет вычислять ее в распределенном режиме.

Мультиномиальная байесовская процедура обучения. Другой способ обучения распознавания на текстовых документах $d\in D$ со словарем V заключается в использовании мультиномиального распределения для моделирования вероятности появления текста в классе $y\in Y$. При этом текст $d\in D$ описывается целочисленным n-мерным, $n=|V|$ вектором $x(d)\in\mathbb{Z}^{|V|}$, чьи элементы $x_i(d)$ соответствуют количеству появлений слова $i\in V$ в тексте d.

Условная вероятность (6) принимает вид

$$P(x_1,...,x_n\mid y)=K(x)\prod_{i=1}^{n}p_{iy}{}^{x_i}, \tag{17}$$

где p_{iy} – вероятность слова $i\in V$ в классе $y\in Y$ (доля слова $i\in V$ среди слов или позиций в текстах класса $y\in Y$), $K(x_1,...,x_n)=\dfrac{(\sum_{i=1}^{n}x_i)!}{\prod_{i=1}^{n}(x_i!)}$ – мультиномиальный коэффициент, не зависящий от y, и не оказывающий влияния на классификацию. Классификатор (5) приобретает вид

$$g_{\pi,\theta}(x_1,...,x_n)=\arg\max_{y\in Y}\pi_y\prod_{i=1}^{n}\theta_{iy}{}^{x_i}. \tag{18}$$

Как и в предыдущем случае (9), процесс обучения сводится к вычислению вектора π и матрицы θ по обучающей выборке $\tau=(e_1,...,e_n)$. Оценки априорных вероятностей классов π вычисляются с помощью MapReduce, аналогично (10), а элементы матрицы оценок условных вероятностей θ вычисляются аналогично (14) в виде отношений

$$\theta_{ij}(\tau)=\frac{c_{ij}(\tau)}{\sum_{k=1}^{n}c_{kj}(\tau)}. \tag{19}$$

Несмотря на различные описания текстов и различные интерпретации связанных с ними условных вероятностей (8), (17) в обоих случаях распределенное вычисление оценок по обучающей выборке с помощью MapReduce производится практически одинаково и имеет одинаковую вычислительную сложность.

Выводы. Для распознавания тестовой информации рассмотрены бернуллиевская и мультиномиальная версии байесовской обучающей процедуры. Описаны области применения различных вариаций байесовской процедуры и соответствующие распределенные реализации на основе MapReduce.

Б.О. Білецький, М.А. Гупал

РОЗПОДІЛЕНІ БАЙЄСІВСЬКІ ПРОЦЕДУРИ РОЗПІЗНАВАННЯ ТЕКСТОВОЇ ІНФОРМАЦІЇ

Для розпізнавання текстової інформації розглянуто дві версії байєсівської оптимальної процедури. Описано розподілені реалізації цих версій на основі MapReduce.

B.A. Biletskyy, M.A. Gupal

DISTRIBUTED BAYESIAN PROCEDURES OF THE TEXT INFORMATION RECOGNITION

In this paper, we consider Bernoulli and Multinomial variations of Bayesian Machine Learning procedures, as well as their distributed implementations based on MapReduce. We discuss their distributed implementation and use cases.

Список литературы

1. Hilbert M., López P. The World's Technological Capacity to Store, Communicate, and Compute Information. *Science*. 2011. Vol. 332. P. 60−65.
2. Moore G. Cramming more components onto integrated circuits. *Electronics*. 1965. Vol. 38. P. 114−117.
3. Gilbert S., Lynch N. Brewer's conjecture and the feasibility of consistent, available, partition-tolerant web services. *ACM SIGACT News*. 2002. Vol. 33. P. 51−59.
4. Dean J., Ghemawat S. MapReduce: simplified data processing on large clusters. Proceedings of the 6th conference on Symposium on Operating Systems Design & Implementation. 2004. Vol. 6. P. 10 − 17.
5. Manning C., Raghavan P., Schütze H. Introduction to information retrieval. *Cambridge University Press*. 2008. 482 p.

Получено 20.05.2018

Об авторах:

Белецкий Борис Александрович,
кандидат физико-математических наук, старший научный сотрудник
Института кибернетики имени В.М. Глушкова НАН Украины,

Гупал Никита Анатольевич,
кандидат физико-математических наук, научный сотрудник
Института кибернетики имени В.М. Глушкова НАН Украины.