УДК 519.217.2

Б.А. БЕЛЕЦКИЙ

РАСПРЕДЕЛЕННЫЕ БАЙЕСОВСКИЕ ПРОЦЕДУРЫ МАШИННОГО ОБУЧЕНИЯ

Аннотация. Рассмотрены бернуллиева и мультиномиальная варианты байесовской процедуры обучения и области их применения, а также их распределенная реализация на основе модели программирования MapReduce. Предложена распределенная категориальная байесовская процедура обучения, описана специфика ее распределенной реализации и область применения.

Ключевые слова: байесовские процедуры обучения распознаванию, распределенные методы машинного обучения, MapReduce.

ВВЕДЕНИЕ

Известно, что более сложные задачи машинного обучения распознаванию с учителем, как правило, требуют более сложных моделей, а последние, в свою очередь, требуют более объемных обучающих выборок. Эмпирические наблюдения показывают, что объемы глобально накапливаемых и обрабатываемых данных возрастают экспоненциально (удваиваются приблизительно каждые 20 месяцев) [1]. Однако согласно закону Мура вычислительная мощность процессоров также возрастает экспоненциально (удваивается каждые 18 месяцев) [2]. Такой баланс позволяет относительно эффективно обрабатывать накапливаемые данные для решения различных задач (в том числе и задач машинного обучения).

Однако в 2010 г. обнаружилось, что закон Мура перестал выполняться вследствие достижения производителями процессоров технологических пределов для их текущей архитектуры. Для обработки стремительно возрастающих объемов данных требовались новые подходы к организации вычислений.

Одним из предложенных подходов стало горизонтальное масштабирование, которое заключается в использования множества соединенных между собой, относительно простых (в смысле производительности и емкости накопителей данных) вычислительных устройств для совместного хранения и обработки данных. Обрабатываемые всеми узлами вычислительной системы данные необходимо хранить, при этом в случае динамического изменения топологии системы (добавление и удаление узлов, потеря связи между ними) ее работоспособность должна сохраняться.

Как выяснилось, предложенный подход горизонтального масштабирования связан с рядом концептуальных и технических трудностей. Во-первых, существенно отличны принципы организации распределенных вычислений и 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 (так называемая CAP-теорема [3]). В-третьих, пропускная способность сети оказалась серьезным ограничивающим фактором для производительности распределенных вычислительных систем.

Предагался ряд подходов, упрощающих построение горизонтально масштабируемых алгоритмов. Одним из популярных в настоящее время подходов является модель программирования MapReduce [4], позволяющая относительно просто масштабировать достаточно большой класс алгоритмов, среди которых и методы машинного обучения.

Рассмотрим основные принципы, на которых основана модель программирования MapReduce. Пусть A, $|A|<\infty$, и B, $|B|<\infty$, — исходный и результирующий типы данных соответственно, $f: A^l \mapsto B$ — вычислимая функция или алгоритм, обрабатывающий распределенную коллекцию $(a_1, \dots, a_l) \in A^l$ элементов исходного типа данных и возвращающий элемент результирующего типа данных $f(a_1, \dots, a_l) \in B$. Для вычисления значения функции $f(a_1, \dots, a_l)$ 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 $f(a_1, \dots, a_l) = m(a_1) \oplus m(a_2) \oplus \dots \oplus m(a_l)$, 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 $\oplus: B \times B \mapsto B$, которая должна удовлетворять условиям коммутативности, ассоциативности и существования нейтрального элемента соответственно:

$$b_1 \oplus b_2 = b_2 \oplus b_1 \quad \forall b_1, b_2 \in B, \tag{1}$$

$$(b_1 \oplus b_2) \oplus b_3 = b_1 \oplus (b_2 \oplus b_3) \quad \forall b_1, b_2, b_3 \in B, \tag{2}$$

$$\exists e \in B : b \oplus e = b \quad \forall b \in B. \tag{3}$$

Последовательное вычисление значения функции $f(a_1, \dots, a_l) \in B$ можно представить в виде бинарного дерева (рис. 1, *а*), листья которого соответствуют вычислению частичных результатов $m(a_i) \in B$, а их цвет обозначает принадлежность элементов $a_i \in A$ различным узлам распределенной системы. Вершины дерева соответствуют операциям агрегации пары частичных результатов $m(a_i) \oplus m(a_j)$, $i, j = \overline{1, l}$, $i \neq j$. Используя свойства (1)–(3), схему вычислений можно перестроить с сохранением конечного результата таким образом (рис. 1, *б*), 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.

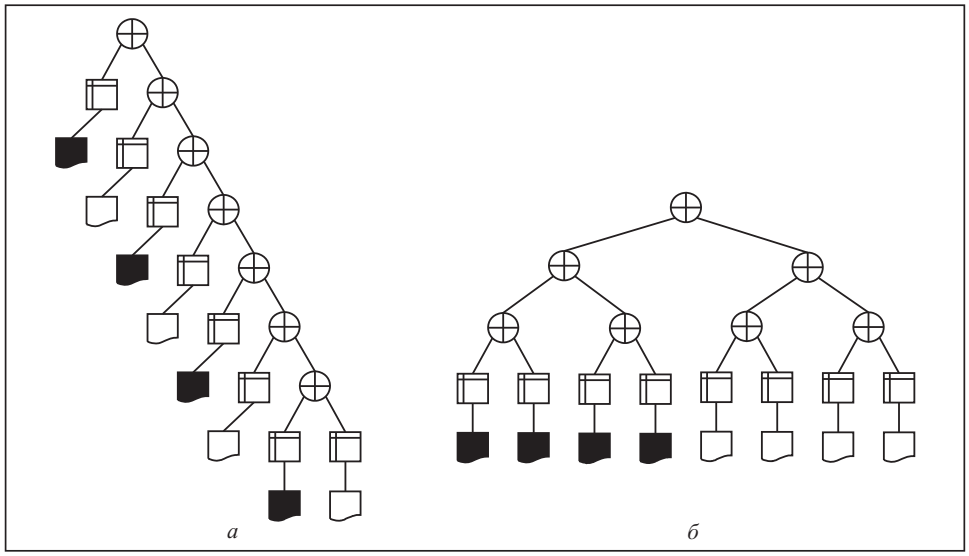

Рис. 1. Схемы вычислений: последовательное (*а*), распределенное (*б*)

Существуют различные реализации модели программирования MapReduce. Рассмотрим одну из них, а именно 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 два варианта процедур: бернуллиева и мультиномиальная, которые успешно применяются для решения задач распознавания текстов [5].

БАЙЕСОВСКАЯ ПРОЦЕДУРА ОБУЧЕНИЯ РАСПОЗНАВАНИЮ ОБРАЗОВ

Пусть D — конечное множество объектов, Y — конечное множество меток или классов, ассоциируемых с этими объектами. Обозначим $E = D \times Y$ множество пар объект–метка. Пусть имеется обучающая выборка таких пар $\tau = (e_1, \ldots, e_l) \in E^l$, содержащая результаты l независимых реализаций случайного эксперимента с множеством элементарных исходов E и неизвестным распределением вероятности на нем.

Задача обучения с учителем сводится к восстановлению неизвестного распределения построением эффективного в некотором смысле классификатора $g : X \mapsto Y$, который вычисляет 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

$$g(x) = \arg\max_{y \in Y} P(x \mid y) P(y), \tag{4}$$

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 набором признаков $x(d) = (x_1, \ldots, x_n) \in X$. Предположение об условной независимости вероятности появления признаков в описании объекта в каждом классе $y \in Y$ позволяет представить условную вероятность описания объекта в виде произведения

$$P(x_1, \ldots, x_n \mid y) = \prod_{i=1}^{n} P_i(x_i \mid y), \tag{5}$$

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 (5) заменяются соответствующими оценками $\hat{P}_i(x_i \mid y)$ и $\hat{P}(y)$, а классификатор (4) принимает вид

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

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

БЕРНУЛЛИЕВА И МУЛЬТИНОМИАЛЬНАЯ БАЙЕСОВСКИЕ ПРОЦЕДУРЫ ОБУЧЕНИЯ

Классификация текстов — одна из областей успешного применения машинного обучения и байесовского подхода в частности. Рассмотрим несколько вариантов байесовской процедуры машинного обучения для задач классификации текстов, доступных в библиотеке Apache Spark.

При классификации текстов в качестве множества объектов D используется множество конечных текстов, состоящих из слов из некоторого словаря V, $| V | < \infty$.

Для простоты изложения предположим, что слова в V отсортированы в лексикографическом порядке и имеют уникальные порядковые номера, с которыми будем их отождествлять. Текстам соответствуют метки из конечного множества меток Y, которые необходимо научиться предсказывать, используя обучающую выборку размеченных текстов. Примерами задач классификации текста могут являться определения интонации сообщения (негативная, позитивная, нейтральная) или темы новостийного сообщения (спорт, политика, наука). Рассмотрим, как применяются байесовские процедуры обучения к задаче классификации текстов.

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

При этом условные вероятности (5) принимают вид

$$P(x_1, \dots, x_n | y) = \prod_{i=1}^{n} [p_{iy} x_i + (1 - p_{iy})(1 - x_i)], \tag{6}$$

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

При расчетах классификатор (4) принимает вид

$$g_{\pi,\theta}(x_1, \dots, x_n) = \arg\max_{y \in Y} \pi_y \prod_{i=1}^{n} [\theta_{iy} x_i + (1 - \theta_{iy})(1 - x_i)], \tag{7}$$

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 (6) на этапе обучения сводится к вычислению вектора π и матрицы θ по обучающей выборке $\tau = (e_1, \dots, e_l)$. 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 $\tau = (e_1, \dots, e_l)$ в виде

$$\pi_i(\tau) = \frac{c_i(\tau)}{l}, \tag{8}$$

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

$$c(e_1, \dots, e_l) = m(e_1) \oplus m(e_2) \oplus \dots \oplus m(e_l). \tag{9}$$

Операция $m: E \mapsto \{0, 1\}^{|Y|}$ строит по обучающему примеру $e_k \in E$ булев вектор $m(e_k) \in \{0, 1\}^{|Y|}$ с единицей на позиции, которая соответствует номеру класса обучающего примера, т.е. $m_i(e_k) = \delta(j, y(e_k))$, где $y(e_k)$ — класс, в который попадает обучающий пример e_k (точнее, порядковый номер класса), а $\delta(\cdot, \cdot)$ — дельта-функция

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

Агрегация пары частичных результатов $\oplus : \{0,1\}^{|Y|} \times \{0,1\}^{|Y|} \mapsto \mathbb{Z}^{|Y|}$ является стандартной операцией поэлементного сложения векторов. Несложно убедиться, что условия (1)–(3) выполняются, это позволяет применять модель программирования MapReduce.

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

$$\theta_{ij}(\tau) = \frac{c_{ij}(\tau)}{c_i(\tau)}, \tag{10}$$

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 с учетом представления

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

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 $y(e_k)$-м столбце описание $x(d(e_k)) \in \mathbb{Z}^n$ текста $d(e_k) \in D$ из обучающего примера $e_k \in D \times Y$,

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

Тогда агрегация пары частичных результатов $\oplus : \mathbb{Z}^{n \times |Y|} \times \mathbb{Z}^{n \times |Y|} \mapsto \mathbb{Z}^{n \times |Y|}$ является стандартной операцией поэлементного сложения матриц, для которой выполняются условия (1)–(3), что позволяет вычислять ее в распределенном режиме.

Мультиномиальная байесовская процедура обучения. Этот способ обучения распознаванию 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 (5) принимает вид

$$P(x_1, \dots, x_n | y) = K(x) \prod_{i=1}^{n} p_{iy}^{x_i}, \tag{12}$$

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 $y \in Y$), $K(x_1, \dots, x_n) = \left(\sum_{i=1}^{n} x_i \right)! / \prod_{i=1}^{n} (x_i!)$ — мультиномиальный коэффициент, не зависящий от y и не оказывающий влияния на классификацию. Классификатор (4) принимает вид

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

Как и в (7), процесс обучения сводится к вычислению вектора π и матрицы θ по обучающей выборке $\tau = (e_1, \dots, e_n)$. Оценки априорных вероятностей классов π вычисляются с помощью MapReduce аналогично (8), а элементы матрицы оценок условных вероятностей θ — аналогично (10) в виде отношений

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

Несмотря на различные описания текстов и интерпретации связанных с ними условных вероятностей (6), (12), в обоих случаях распределенное вычисление оценок по обучающей выборке с помощью MapReduce проводится практически одинаково и имеет одинаковую вычислительную сложность.

Категориальная байесовская процедура обучения. Рассмотрим случай, когда приведенные байесовские методы распознавания трудно применимы. Пусть объект $d \in D$ описывается с помощью n категориальных случайных величин $x(d) = (x_1, \dots, x_n)$, $x_i \in X_i$, где X_i — множество значений (которые будем отождествлять с их порядковыми номерами) i-го признака, $i = \overline{1, n}$. Такими признаками могут являться типы нуклеотидов в ДНК (А, Г, Ц, Т), пол пациента (М, Ж) или семейное положение.

В этом случае условная вероятность (5) принимает вид

$$P(x_1, \dots, x_n | y) = \prod_{i=1}^{n} P_i(x_i | y),$$

где $P_i(x_i | y)$ — вероятность, что i-й признак принимает значение $x_i \in X_i$. Тогда классификатор (6) будет иметь вид

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

где $\pi \in \mathbb{R}^{|Y|}$ — оценки априорных вероятностей классов, которые вычисляются аналогично (8), $\theta_1, \dots, \theta_n$ — матрицы оценок условных вероятностей признаков в классах $\theta_i \in \mathbb{R}^{|X_i| \times |Y|}$ и $\theta_{ijk} = \hat{P}_i(j | k)$, элементы которых вычисляются аналогично (10) по обучающей выборке τ в виде отношений

$$\theta_{ijk}(\tau) = \frac{c_{ijk}(\tau)}{\sum_{m=1}^{|X_i|} c_{imk}(\tau)}, \tag{13}$$

где $c_{ijk}(\tau)$ — количество обучающих примеров класса $k \in Y$ в выборке τ, в котором i-й признак описания принимает значение $j \in X_i$. Для обучения категориального байесовского классификатора (13) требуется вычислить n матриц $\theta_1, \dots, \theta_n$ оценок условных вероятностей. Распределенное вычисление вектора π и матриц $\theta_1, \dots, \theta_n$ выполняется аналогично (9), (11).

В работах [6, 7] показано, что категориальная байесовская процедура обучения имеет полиномиальные оценки сложности и оптимальна для объектов, которые описываются независимыми дискретными признаками. Масштабируемая категориальная байесовская процедура обучения реализована в виде проекта с открытым кодом [8] на языке Scala на основе библиотеки Spark, реализующей модель программирования MapReduce. Одной из тестовых задач, на которых проверялась работоспособность метода, была задача установления диагноза пациента по его симптомам. При этом в качестве обучающей выборки использовались данные [9] с 120 симптомами и соответствующими булевыми диагнозами. В качестве симптомов рассматривались температура пациента и пять булевых признаков о наличии различных болей или воспалений. После округления значения температуры до ближайшего целого все признаки в описании объекта рассматривались как категориальные, что позволило применять категориальную байесовскую процедуру обучения для установления диагноза по симптомам пациента. Точность диагностирования оценивалась пятикратным случайным разбиением выборки на

непересекающиеся обучающую и тестовую части, содержащие 2/3 и 1/3 от общего числа примеров, которые использовались для обучения и оценки точности распознавания. При этом точность распознавания достигала 97 %.

ЗАКЛЮЧЕНИЕ

Рассмотрены бернуллиева и мультиномиальная варианты байесовской обучающей процедуры. Предложена реализация категориальной байесовской процедуры. Описаны области применения различных вариантов байесовской процедуры и соответствующие распределенные реализации на основе MapReduce.

СПИСОК ЛИТЕРАТУРЫ

1. Hilbert M., López P. The world's technological capacity to store, communicate, and compute information. *Science*. 2011. Vol. 332, Iss. 6025. P. 60–65.
2. Moore G.E. Cramming more components onto integrated circuits. *Electronics*. 1965. Vol. 38, N 8. 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, Iss. 2. P. 51–59.
4. Dean J., Ghemawat S. MapReduce: simplified data processing on large clusters. *Proc. of the 6th Conference on Symposium on Opearting Systems Design & Implementation*. 2004. Vol. 6. P. 10–17.
5. Manning C.D., Raghavan P., Schütze H. Introduction to information retrieval. New York: Cambridge University Press, 2008. 482 p.
6. Sergienko I.V., Gupal A.M., Pashko S.V. Complexity of classification problems. *Cybernetics and Systems Analysis*. 1996. Vol. 32, N 4. P. 519–533.
7. Beletskiy B.A, Vagis A.A., Vasilyev S.V, Gupal N.A. Complexity of Bayesian procedure of inductive inference. Discrete case. *Journal of Automation and Information Sciences*. 2006. Vol. 38, Iss. 11. P. 56–73.
8. URL: https://github.com/bbiletskyy/categorical-bayes.
9. Czerniak J., Zarzycki H. Application of rough sets in the presumptive diagnosis of urinary system diseases. In: *Artificial Intelligence and Security in Computing Systems. The Springer International Series in Engineering and Computer Science*. Sollдek J., Drobiazgiewicz L. (Eds.). Boston, MA: Springer, 2003. Vol. 752. P. 41–51.

Надійшла до редакції 24.04.2018

Б.О. Білецький

РОЗПОДІЛЕНІ БАЄСІВСЬКІ ПРОЦЕДУРИ МАШИННОГО НАВЧАННЯ

Анотація. Розглянуто бернуліїву та мультиноміальну варіанти баєсівської процедури машинного навчання, а також їхню розподілену реалізацію на основі моделі програмування MapReduce. Запропоновано категорійну баєсівську процедуру машинного навчання, обговорено специфіку її розподіленої реалізації та сферу її застосування.

Ключові слова: баєсівські процедури навчання распізнаванню, розподілені методи машинного навчання, MapReduce.

B. Biletskyy

DISTRIBUTED BAYESIAN MACHINE LEARNING PROCEDURES

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

Keywords: Bayesian machine learning procedures for recognition, distributed methods machine learning, MapReduce.

Белецкий Борис Александрович,
кандидат физ.-мат. наук, старший научный сотрудник Института кибернетики им. В.М. Глушкова НАН Украины, Киев, e-mail: borys.biletskyy@gmail.com.