Распределенная реализация метода средних решающих правил с умными выборками для больших задач SVM

М.Ю. Курбаков
Тульский государственный университет
Тула, Россия
muwsik@mail.ru

А.В. Копылов
Тульский государственный университет
Тула, Россия
andkopylov@gmail.com

С.Д. Двоенко
Тульский государственный университет
Тула, Россия
sergedv@yandex.ru

В.В. Сулимова
Тульский государственный университет
Тула, Россия
vsulimova@yandex.ru

***Аннотация*—Ранее нами был предложен метод средних решающих правил с умными выборками (SS-KMDR) для эффективного решения двухклассовых задач обучения по методу опорных векторов (SVM) в условиях большого числа объектов и было экспериментально показано, что он позволяет получить результаты, близкие по качеству к эталонным за существенно меньшее время по сравнению с традиционными методами. В данной работе мы представляем его высокопроизводительную реализацию, которая позволяет дополнительно существенно повысить скорость вычислений за счет разработки эффективных параллельных алгоритмов и применения современных технологий параллельных и распределенных вычислений.**

***Ключевые слова— SVM, большая обучающая совокупность, высокопроизводительные вычисления*.**

1. Введение

При решении больших задач SVM возникают проблемы высокой вычислительной сложности, нехватки памяти и быстрого произвольного доступа к объектам, осложняющиеся тем, что традиционный формат хранения данных (libsvm) не позволяет вычислить положение объекта с заданным номером.

Практически все существующие методы, детальный обзор которых приведен в [1], направлены на решение только одной из указанных проблем, а также, как правило, имеют и другие недостатки, наиболее важными из которых является отсутствие возможности введения нелинейности и итерационная природа с многочисленными зависимостями по данным.

Предложенный нами ранее метод средних решающих правил с умными выборками (SS-KMDR) [1] лишен указанных недостатков и позволяет получить результаты, близкие по качеству к эталонным за ощутимо меньшее время. В данной работе мы представляем его распределенную версию, обеспечива-ющую дополнительное повышение производительности.

2. Метод средних решающих правил с умными выборками (SS-KMDR)

A. Основная идея

Основная идея заключается в формировании небольших обучающих подвыборок, независимом обучении по каждой из них с последующим объединением в общее решение исходной задачи. В [1] показано, что и в пространстве признаков, и в пространстве, порожденном потенциальной функцией, объединение может быть осуществлено путем усреднения, но необходимо специфическое понимание усреднения для каждого из этих случаев.

В отличие от традиционных для выборочных методов [2], [3] случайных обучающих подвыборок предлагается использовать специально сформированные (умные) выборки, составленные из объектов, расположенных вблизи разделяющей гиперплоскости, а именно, опорных объектов, полученных при обучении по небольшим случайным подвыборкам.

Б. Оптимизация работы с данными

В [1] предложена специальная схема, основанная на осуществлении предварительной разметки файла с данными для определения расположения в нем объектов. Предварительная разметка в совокупности с применением механизма отображения файла в память, позволяет осуществить быстрый произвольный доступ к любому объекту, не требуя одновременной загрузки в память всех объектов и снимая практическое ограничение на размер обучающей совокупности.

В. Двухуровневая схема параллельных вычислений

Метод SS-KMDR может быть алгоритмически реализован несколькими способами. В данной работе используется версия с фиксированным числом умных выборок. Исходными параметрами в этом случае являются: количество умных выборок, размер одной умной выборки, размер одной случайной подвыборки и полная обучающая совокупность.

Поскольку каждая умная выборка может быть сформирована и обработана независимо от других, то известное заранее количество умных выборок позволяет легко реализовать массивный параллелизм верхнего уровня при помощи процессов. Процессы могут работать на одном или нескольких узлах вычислительной системы. Последнее является более предпочтительным, поскольку позволяет выделить большее количество ресурсов для каждой задачи.

Задача формирования одной умной выборки, в свою очередь, может быть разбита на ряд подзадач формирования случайных подвыборок и обучения по

ним. Однако необходимое количество случайных подвыборок нельзя определить заранее, так как неизвестно сколько опорных объектов будет получено в результате обучения по каждой из них. Это препятствует организации параллельного выполнения данного этапа.

Для решения этой проблемы предлагается смешанная (последовательно-параллельная) схема формирования умной выборки, реализуемая при помощи потоков. Эта схема заключается в том, чтобы сформировать и параллельно решить пул задач обучения на случайных подвыборках, объединить полученные опорные объекты и, если их количество меньше заданного объема умной выборки, сформировать и решить новый пул задач и т. д., пока не будет достигнут размер умной выборки.

3. Экспериментальное исследование

Исследование проводилось с использованием суперкомпьютера НИИ ВЦ МГУ «Ломоносов-2» [4] на наборах данных из репозиториев LibSVM (https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/) и PiSVM (https://sourceforge.net/projects/pisvm/files/pisvm-datasets/pisvm-dataset-1.0/), их основные характеристики приведены в таблице I.

Таблица1. Основные характеристики наборов данных

Набор данных	Объектов на обучении/ контроле	Число признаков (доля не нулевых)
mnist-784	60 000 / 10 000	784 (61.73%)
kddcup99	4 898 430 / 11 029	122 (10.23%)
SUSY	4 500 000 / 500 000	18 (98.82 %)
HIGGS	10 500 000 / 500 000	28 (92.11 %)

В таблице II приведены время работы и точность распознавания на контрольной выборке для предложенного метода (SS-KMDR), библиотеки LibSVM [5] (являющейся эталоном качества), ее параллельной версии PiSVM [6], а также python scykit-learn реализаций бэггинга [2] с ансамблем из 1 и 5 моделей, а также стохастических методов SGD [7] и ASGD [8]. Результаты для метода SS-KMDR приведены для нескольких наборов значений параметров, указанных в скобках в следующем формате: (<число умных выборок> / <число процессов>; <размер умной выборки>). Размер случайной выборки везде был принят равным 300. Для остальных методов были взяты параметры, принятые по умолчанию. Для mnist-784 для SS-KMDR (*/*; 20000) вычисления не проводились, из-за относительно небольшого числа объектов.

Таблица 2. Результаты сравнения SS-KMDR с другими методами обучения SVM

Метод	Набор данных							
	mnist-784		*kddcup99*		*SUSY*		*HIGGS*	
	время (с)	*точность*	*время (с)*	*точность*	*время (с)*	*точность*	*время (с)*	*точность*
LibSVM	387	99.15	> 3600	-	> 86 400	-	> 604 800	-
PiSVM	302	99.15	Нехватка ОП		Нехватка ОП		Нехватка ОП	
Бэггинг-1	157	99.47	2241	92.74	> 86 400	-	> 86 400	-
Бэггинг-5	775	99.49	> 3600	-	> 86 400	-	> 86 400	-
SGD	0.633	94.39	16.98	91.96	40.5	77.14	230	62.55
ASGD	0.675	94.47	15.97	91.95	35.2	78.10	144	63.62
SS-KMDR (1/1; 5000)	6.24	98.54	0.358	91.95	5.27	81.12	23.6	65.24
SS-KMDR (8/1; 5000)	58	99.45	3.29	92.35	72	84.32	213	70.23
SS-KMDR (8/8; 5000)	8.06	99.47	0.479	92.33	6.82	84.35	26.7	70.22
SS-KMDR (8/8; 20000)	-	-	0.74	92.57	63.6	86.93	170	75.41
SS-KMDR (16/16; 20000)	-	-	0.89	92.64	66.2	87.27	173	80.05

4. Заключение

Эксперименты показывают, что классические LibSVM, его параллельная версия PiSVM и Бэггинг не позволяют получить решение для большинства рассмотренных наборов данных из-за ограничений по времени и памяти. Быстрые методы SGD и ASGD не допускают введение нелинейности, поэтому проигрывают по качеству. Предложенный подход позволяет достаточно быстро найти решение, близкое к эталонному, а его параллельная реализация обеспечивает дополнительное повышение производительности, характеризуется высокой эффективностью использования процессоров и, более того, в ряде случаев позволяет повысить качество решения за счет охвата большего объема данных за меньшее время.

Благодарности

Работа выполнена при поддержке РФФИ, проекты № 20-07-00055 и № 20-07-00441 с использованием оборудования Центра коллективного пользования сверхвысокопроизводительными вычислительными ресурсами МГУ имени М.В. Ломоносова [4].

Литература

[1] Makarova, A. Mean Decision Rules Method with Smart Sampling for Fast Large-Scale Binary SVM Classification / A. Makarova, M. Kurbakov, V. Sulimova // ICPR. – 2021. – P. 8212-8219.

[2] Breiman, L. Bagging predictors / L. Breiman // ML. – 1996. – Vol. 24(2). – P. 123-140.

[3] Chauhan, V.K. Mini-batch block-coordinate based stochastic average adjusted gradient methods to solve big data problems / V.K. Chauhan, D. Kalpana, S. Anuj // Asian Conf. on Machine Learn., PMLR. – 2017. – P. 49-64.

[4] Voevodin, Vl. Supercomputer Lomonosov-2: Large Scale, Deep Monitoring and Fine Analytics for the User Community / Vl. Voevodin, A. Antonov, D. Nikitenko, P. Shvets, S. Sobolev, I. Sidorov, K. Stefanov, Vad. Voevodin, S. Zhumatiy // Supercomputing Frontiers and Innovations. – 2019. – Vol. 6(2). – P. 4-11. DOI:10.14529/jsfi190201.

[5] Chang, C.-C. LIBSVM: a library for support vector machines / C.-C. Chang, C.-J. Lin. – 2001. URL: http://www.csie.ntu.edu.tw/ cjlin/libsvm.

[6] Chang, E.Y. Psvm: Parallelizing support vector machines on distributed computers / E.Y. Chang, K. Zhu, H. Wang, H. Bai, J. Li, Z. Qiu, H. Cui / NIPS. – 2007. – Vol. 20.

[7] Bottou, L. Stochastic Learning / L. Bottou, O. Bousquet, U. von Luxburg // Advanced Lect. on ML, LNAI 3176. – Springer Verlag, Berlin, 2004. – P. 146-168.

[8] Averaging Stochastic Gradient Descent on Riemannian Manifolds / N. Tripuraneni, N. Flammarion, F. Bach, M. Jourdan // arXiv:1802.0912, 2018.