Реализация параллельного алгоритма обучения в методе градиентного бустинга деревьев решений для систем с распределенной памятью*

П.Н. Дружков, А.Н. Половинкин

Нижегородский государственный университет им. Н.И. Лобачевского

В работе описывается реализация одного из наиболее перспективных алгоритмов обучения с учителем - градиентного бустинга деревьев решений (Gradient Boosting Trees). Предлагается схема параллельной реализации алгоритма обучения для систем с распределенной памятью. Приводятся результаты вычислительных экспериментов и анализ эффективности предложенного подхода к распараллеливанию.

1. Введение

Машинное обучение является подразделом весьма обширной области науки, изучающей искусственный интеллект. Алгоритмы, относящиеся к данному направлению, используются при решении задач, для которых зачастую сложно или невозможно придумать явный алгоритм решения: предсказание погоды, прогнозирование экономических и социальных процессов, медицинская диагностика, детектирование объектов на фото или видео, распознавание текста, речи, создание антивирусных программ и алгоритмов фильтрации рекламы и спама.

В настоящее время известно достаточно много алгоритмов обучения с учителем, предназначенных для решения задачи восстановления регрессии или классификации: машина опорных векторов [15], метод *K* ближайших соседей [10], нейронные сети [10], AdaBoost [10], деревья решений [2] и их различные ансамбли (случайные деревья [1], полностью случайные деревья [9], градиентный бустинг деревьев решений [7, 8]). В рамках данной работы рассматривается программная реализация одного из наиболее перспективных алгоритмов обучения с учителем – алгоритма градиентного бустинга деревьев решений (GBT – gradient boosting trees), которая является первой полнофункциональной C/C++ реализацией данного метода с открытым кодом. Результаты вычислительного эксперимента, проведенного с использованием широко распространенных наборов реальных данных, взятых из репозитория UCI [14], свидетельствуют о конкурентоспособности предлагаемой реализации по сравнению с реализациями других алгоритмов. Разработанный код интегрирован в одну из наиболее известных свободно распространяемых библиотек компьютерного зрения OpenCV [4, 11].

Необходимо отметить, что многие из решаемых в настоящее время практических задач машинного обучения и компьютерного зрения требуют обработки значительного объема входных данных. Это связано с тем, что каждый исследуемый объект может быть описан вектором признаков, содержащим сотни или даже тысячи переменных, а обучающая и тестовая выборки могут содержать десятки тысяч описаний объектов. В связи с этим, наряду с качеством предсказания на одно из первых мест встает вопрос производительности используемого алгоритма. В работе дается обзор различных аспектов параллельной реализации алгоритма обучения модели и предсказания на новых данных, а также предлагается и анализируется подход к распараллеливанию алгоритма обучения модели для систем с распределенной памятью.

* Авторы благодарят И.Б. Меерова и Н.Ю. Золотых (ННГУ) за ценные замечания и полезные обсуждения. Работа выполнена в рамках программы «Исследования и разработки по приоритетным направлениям развития научно-технологического комплекса России на 2007-2013 годы», государственный контракт № 11.519.11.4015.

2. Градиентный бустинг деревьев решений

2.1 Постановка задачи

Одной из задач, изучаемой в машинном обучении, является *задача обучения с учителем*. В рамках этой задачи дано некоторое множество *объектов* $X = X_1 \times X_2 \times \ldots \times X_p$ *(пространство признаков)*. Каждому объекту $x = (\xi_1, \xi_2, \ldots, \xi_p) \in X$ поставлена в соответствие величина y, называемая *выходом*, или *ответом*, и принадлежащая множеству допустимых ответов Y. Упорядоченная пара «объект-ответ» (x,y), где $x \in X$, $y \in Y$ называется *прецедентом*. Требуется восстановить зависимость между входом и выходом, основываясь на данных о конечном наборе прецедентов, называемом *обучающей выборкой*: $\{(x_i, y_i) \mid x_i \in X, y_i \in Y, i = 1, ..., n\}$. Другими словами, задача состоит в построении функции $f: X \to Y$ из некоторого множества K, которая, получив на вход x, предсказала бы значение ответа y как можно точнее. В случае конечного Y, говорят о *задаче классификации*, если $Y = \mathbf{R}$ – *задаче восстановления регрессии* [10]. Процесс нахождения f называется *обучением* (*тренировкой*, *настройкой)* модели, процесс определения выхода по некоторому входу с помощью уже построенной модели – *предсказанием*.

2.2 Метод решения

Один из общих подходов решения задач обучения заключается в комбинировании моделей. Две основные конкурирующие идеи данного подхода – бэггинг (*bagging* от *Bootstrap Aggregating*) [3] и бустинг (*boosting*) [6]. Первая из них состоит в построении множества независимых между собой моделей с дальнейшим принятием решения путем голосования в случае задачи классификации и усреднения в случае регрессии. Данный подход реализован в алгоритме случайных деревьев (*random trees* или *random forest*). Бустинг, в противоположность бэггингу, обучает каждую следующую модель с использованием данных об ошибках предыдущих моделей. Распространенным выбором базовой модели в упомянутых выше алгоритмах являются деревья решений [2], что обусловлено их универсальностью и наличием эффективных алгоритмов обучения. Напомним, что дерево решений рекурсивно разбивает пространство признаков на непересекающиеся области с помощью правил вида $\xi_j \le t$, если $\xi_j \in \mathbf{R}$, и вида $\xi_j \in A \subset X_j$, где X_j – конечное множество возможных значений переменной ξ_j. Каждой из полученных областей присваивается некоторое значение результирующей переменной y. Таким образом, дерево решений определяет кусочно-постоянную функцию.

Алгоритм градиентного бустинга деревьев решений является развитием бустинг-идеи. Он позволяет строить аддитивную функцию в виде суммы деревьев решений итерационно по аналогии с методом градиентного спуска. Данный подход позволяет расширить круг решаемых этим алгоритмом задач, а также зачастую получить выигрыш в точности предсказания.

Далее приводится краткое описание алгоритма обучения градиентного бустинга деревьев решений для задачи восстановления регрессии в случае использования квадратичной функции потерь. Пусть обучающая выборка содержит n прецедентов, y_i – значение ответа для i-го прецедента, $T_j(x_i)$ – значение, предсказанное j-м деревом в ансамбле для i-го объекта, ν – коэффициент масшабирования. Тогда псевдоостатком для i-го объекта на m-м шаге алгоритма обучения называется значение $\tilde{y}_i = y_i - T_0(x_i) - \nu \cdot \sum_{s=1}^{m-1} T_s(x_i)$, что соответствует разности между истинным значением ответа и предсказанием ансамбля деревьев решений, построенным на (m – 1)-м шаге алгоритма обучения. Пусть M – общее число деревьев в ансамбле, тогда общая схема тренировки модели, изображенная на рис.1, может быть сформулирована следующим образом:

1. Обучить дерево T_0 на исходном наборе данных $(x_i, y_i), i = 1,...,n$
2. Для каждого $m = 1,2,...,M$
   - для всех объектов в обучающей выборке вычислить псевдоостатки $\tilde{y}_i, i = 1,...,n$
   - добавить в ансамбль новое дерево, обученное на наборе данных $(x_i, \tilde{y}_i), i = 1,...,n$

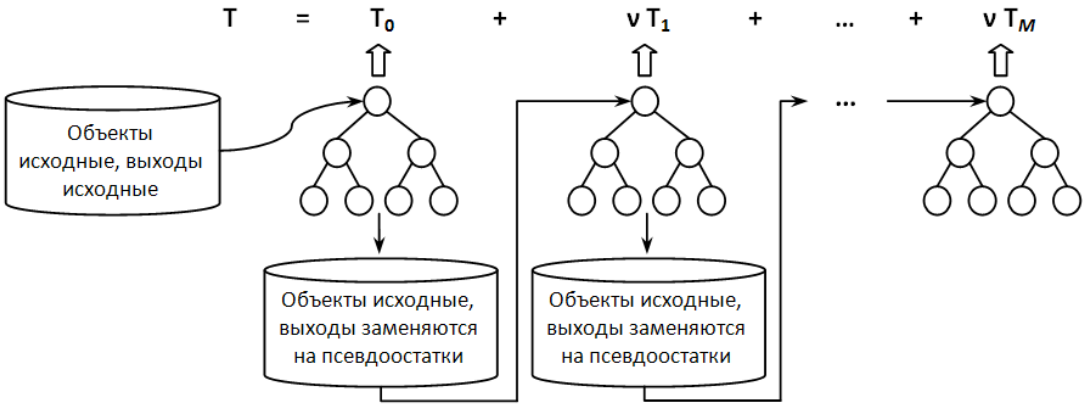

Рис. 1. Алгоритм обучения модели градиентного бустинга деревьев решений

Детальное описание алгоритма обучения, а также подробности, связанные с тренировкой отдельных деревьев решений, и особенности реализации алгоритма для задач восстановления регресии с другими функциями потерь, а также классификации с двумя и более классами можно найти в [7]. Здесь же отметим лишь, что построение одиночного дерева решений), если считать количество возможных значений каждой категориальной переменной константным, имеет трудоемкость $O(pn\log(n)+pnd)$, где p – размерность пространства признаков, n – объем обучающей выборки, d –глубина дерева. Алгоритм построения модели градиентного бустинга деревьев решений имеет трудоемкость $O(MK(pn\log(n)+pnd))$, где M – общее число деревьев в одном ансамбле, K – количество категорий в задаче классификации, для регрессионных задач K=1. При этом исходные данные (обучающая выборка) занимают порядка $O(pn)$ памяти.

Таким образом, на выходе алгоритма обучения мы получаем набор из M деревьев решений, и для осуществления предсказания, т. е. определения выхода y для нового объекта x, следует вычислить сумму $y = T_0(x) + \nu \cdot \sum_{m=1}^{M} T_m(x)$.

Таблица 1. Результаты экспериментального сравнения алгоритмов обучения с учителем: ошибки, полученные методом перекрестного контроля

Название набора данных	*Градиентный бустинг* (GBT)	*Дерево решений* (CvDTree)	*Случайные деревья* (CvRTrees)	*Случайные деревья* (CvERTrees)	*Машина опорных векторов* (CvSVM)
auto-mpg	2	2.238	*1.879*	2.147	2.981
Computer hardware	12.62	15.62	11.62	*9.631*	37
Concrete slump	2.257	2.923	2.6	2.359	*1.767*
Forestfires	18.74	17.26	17.79	16.64	*12.9*
Boston housing	*2.033*	2.602	2.135	2.196	4.049
imports-85	1306	1649	*1290*	1487	1787
Servo	*0.238*	0.258	0.247	0.42	0.655
Abalone	*1.47*	1.604	1.492	1.498	2.091

2.3 Реализация алгоритма градиентного бустинга деревьев решений

Авторами данной работы была выполнена программная реализация алгоритма градиентного бустинга деревьев решений [16], включающая как алгоритм тренировки модели, так и ее дальнейшее использование для предсказания. В данном разделе приведены некоторые экспериментальные результаты, показывающие достоинства и недостатки метода градиентного бустинга. Наряду с описываемым подходом были рассмотрены конкурирующие алгоритмы: одиночные деревья решений (алгоритм CART) [2], случайные деревья (случайные леса) [1, 8], ма-

шина опорных векторов [15]. Программной основой проведенных экспериментов является открытая библиотека компьютерного зрения OpenCV: все результаты, относящиеся к конкурирующим алгоритмам, были получены непосредственно с помощью ее компонентов: CvDTree, CvRTrees, CvERTrees и CvSVM. Эксперименты проводились на наборах реальных данных, взятых из репозитория UCI, при этом мерой качества модели считалась средняя абсолютная ошибка 10-кратного перекрестного контроля, см. табл. 1. Приведенные результаты говорят о том, что алгоритм градиентного бустинга деревьев решений для различных задач дает лучшие, или сопоставимые по качеству результаты в сравнении с другими алгоритмами.

3. Параллельная реализация алгоритма градиентного бустинга деревьев решений для систем с распределенной памятью

3.1 О подходах к распараллеливанию вычислений в алгоритме градиентного бустинга

Значительное количество задач из области машинного обучения характеризуется большим объемом входных данных, обусловленным как количеством объектов в обучающей и тестовой выборках, так и размерностью пространства признаков. В зависимости от характера использования алгоритма в прикладной задаче наиболее требовательной к быстродействию может оказаться как стадия обучения модели, так и стадия предсказания результата на новых данных.

Несмотря на то, что предсказание в алгоритме градиентного бустинга деревьев решений является менее трудоемким по сравнению с построением модели, время работы этого алгоритма зачастую также является критичным. На практике часто приходится выполнять не единичные предсказания, а целые серии: например, в некоторых алгоритмах, решающих задачу детектирования объектов на изображении методом «скользящего окна», может потребоваться выполнение десятков тысяч предсказаний на одно изображение [5]; кроме того, в некоторых прикладных задачах из данной области требуется работа в режиме реального времени.

В работе [17] авторами были предложены и реализованы два различных подхода к распараллеливанию алгоритма предсказания для систем с общей памятью с использованием обученной модели градиентного бустинга деревьев решений. Напомним, что для получения выхода y по некоторому входу x необходимо вычислить сумму предсказаний всех деревьев из имеющегося ансамбля. Так как значение каждого слагаемого может быть получено независимо от остальных, можно рассмотреть первую параллельную схему предсказания, в которой для одного объекта x значения предсказаний отдельных деревьев $T_m(x)$ в сумме $y = T_0(x) + \nu \cdot \sum_{m=1}^{M} T_m(x)$ вычисляются параллельно несколькими потоками. Другой рассмотренный подход: распараллеливание по данным – основан на необходимости одновременного предсказания для большого количества новых объектов. В данном случае выполняется параллельное вычисление значений $y_i = T_0(x_i) + \nu \cdot \sum_{m=1}^{M} T_m(x_i)$ для нескольких объектов x_i. Очевидно, что предложенные схемы распараллеливания алгоритма предсказания применимы и к системам с распределенной памятью.

Возможные подходы к распараллеливанию алгоритма обучения не являются столь тривиальными. К сожалению, бустинг-алгоритмы в целом являются плохо распараллеливаемыми: подход, связанный с одновременным построением нескольких классификаторов в ансамбле, здесь невозможен в силу того, что тренировка каждого следующего дерева решений требует результатов предсказания всех предыдущих. В связи с этим, основные усилия предпринимаются для создания параллельных алгоритмов на уровне базовых моделей, в частности деревьев решений [2]. Большинство таких подходов основано на параллелизме по данным и упрощенных алгоритмах построения деревьев решений [12, 13]. Таким образом, можно говорить, об ориентированности большинства методов на большие объемы исходных данных. В данной работе предлагается схема параллельной реализации алгоритма обучения для отдельного широко распространенного класса задач: классификация объектов с большим числом категорий (к дан-

ному классу относятся, например, задача классификации изображений, автоматизированное построение электронных каталогов научных текстов). Предлагаемый подход не затрагивает алгоритм построения отдельных деревьев решений, что позволяет достичь ускорения процесса обучения модели без потери ее качества. Также, данный метод может быть скомбинирован с параллельными алгоритмами построения базовых моделей, что может, в свою очередь, привести к улучшению масштабируемости параллельного алгоритма градиентного бустинга в целом. Также следует отметить, что, несмотря на наличие других подходов к распараллеливанию рассматриваемого алгоритма, функциональность существующих открытых реализаций весь ограничена, что не позволяет произвести непосредственное сравнение с ними.

3.2 Параллельная реализация алгоритма градиентного бустинга деревьев решений для задач классификации

В отличие от приведенного выше алгоритма обучения модели для задач восстановления регрессии, где строится одна последовательность деревьев, модель для решения задачи классификации [7] состоит из K последовательностей деревьев решений заданной длины M, где K – число категорий, к которым может относиться выходная переменная. Каждая из таких последовательностей $\{T_{0k}, T_{1k}, ..., T_{Mk}\}, k = 1, ..., K$ оценивает вероятность принадлежности объекта к k-му классу. Таким образом, в этом случае общее число обучаемых деревьев увеличивается в K раз, что оказывает серьезное негативное влияние на производительность процедуры обучения. Общая схема построения модели градиентного бустинга деревьев решений при использовании в качестве функции потерь кросс-энтропии, применяемой в текущей реализации для решения задач классификации, выглядит следующим образом:

1. Для всех $m = 1, ..., M$
   a. Для всех $k = 1, ..., K$
      a. Вычислить антиградиент функции потерь $\tilde{y}_{ik}, i = 1, ..., n$ [7]
      b. Обучить дерево решений T_{mk} на данных $(x_i, \tilde{y}_{ik}), i = 1, ..., n$
   b. Обновить модель, добавив деревья $T_{m1}, T_{m2}, ..., T_{mK}$ в соответствующие последовательности

Необходимо отметить, что значение функции антиградиента для определенного k зависит от значения функций предсказания, соответствующих всем уже построенным последовательностям, поэтому ансамбль деревьев решений для одного класса не может быть получен независимо от остальных последовательностей. При этом построение K деревьев решений на m-й итерации алгоритма может быть выполнено параллельно. Так как при обучении одиночного дерева решений производится интенсивная работа с памятью и реализация процесса построения дерева в библиотеке OpenCV является параллельной на общей памяти, параллельное построение нескольких деревьев решений (особенно при большом числе категорий K) имеет смысл производить на распределенной памяти.

Основываясь на этой идее, была выполнена реализация алгоритма обучения модели градиентного бустинга деревьев решений с использованием технологии MPI. Построение модели осуществляется K процессами, каждый из которых строит деревья одной последовательности. После построения очередного дерева решений работа всех процессов синхронизируется, и каждый процесс рассылает всем остальным данные о текущих предсказаниях соответствующего ему ансамбля на объектах обучающей выборки, которые необходимы для выполнения следующей итерации алгоритма. Данные сообщения представляют собой вектора значений типа float из n компонент, где n – объем обучающей выборки. По окончанию построения деревьев производится пересылка всех полученных последовательностей на нулевой процесс, где происходит сбор модели. Общая схема предлагаемого параллельного подхода представлена на рис. 2.

С использованием описанной реализации были проведены эксперименты, оценивающие целесообразность использования данного подхода. Вычислительный эксперимент проводился с использованием кластера ННГУ, в состав которого входят 64 двухпроцессорных двухъядерных сервера Intel Xeon 3.2 GHz, 4 Gb RAM.

С помощью алгоритма градиентного бустинга деревьев решений были решены две задачи: распознавание рукописных букв латинского алфавита (задача взята из репозитория UCI [14]) и рубрикация статей базы журнала «Вестник ННГУ». Описание использованных наборов данных приведено в табл. 2. Следует отметить, что коллекции данных, являющиеся стандартом при оценке качества работы алгоритмов обучения с учителем, такие как UCI, содержат малое количество многоклассовых задач, на решение которых и ориентирована предлагаемая параллельная реализация. В связи с этим, выбор «стандартных» задач был существенно ограничен.

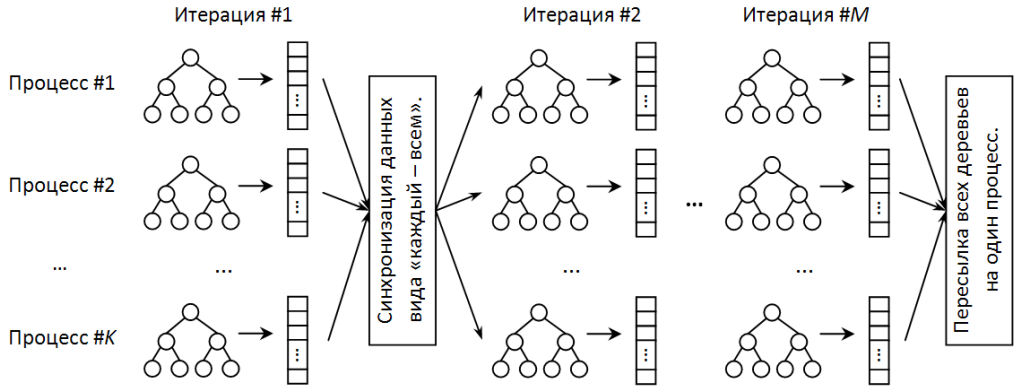

Рис. 2. Схема параллельной версии алгоритма обучения модели градиентного бустинга деревьев решений для решения задачи классификации

Таблица 2. Описание наборов данных, использованных в вычислительном эксперименте

Название набора данных	*Объем обучающей выборки*	*Размерность пространства признаков*	*Количество классов*
Letter Recognition	16000	16	26
База журнала «Вестник ННГУ»	1855	181822	18

Таблица 3. Результаты вычислительного эксперимента

Название набора данных	*Время работы последовательной версии, сек.*	*Кол-во процессов для параллельной версии*	*Время работы параллельной версии, сек.*	*Ускорение*
Letter Recognition	2743	26	170	~16.1
База журнала «Вестник ННГУ»	149114	18	9888	~15

Параллельным алгоритмом обучения для каждого набора данных использовалось число процессов, равное числу классов K в решаемой задаче. Каждый из создаваемых процессов строил последовательность из 1000 деревьев. Время работы сравнивалось с версией программы, работающей на одном узле и строящей все деревья последовательно. Результаты вычислительного эксперимента указаны в табл. 3.

Как можно видеть из приведенных результатов, эффективность параллельной версии зависит от характеристик конкретной задачи. Во-первых, малая размерность пространства признаков для задачи распознавания букв при относительно небольшом объеме обучающей выборки приводит к тому, что построение отдельного дерева происходит достаточно быстро, и, следовательно, время, затрачиваемое на пересылки данных, дает больший вклад в общее время работы алгоритма обучения. Во-вторых, больший объем выборки и количество классов соответствуют большему объему пересылок на каждой итерации алгоритма, что также негативно сказывается на эффективности параллельного подхода.

Таким образом, можно сделать вывод о том, что задачи (наборы данных) с относительно небольшим объемом обучающей выборки и большим числом предикативных переменных являются наиболее подходящими для текущей реализации кластерной версии GBT.

4. Заключение

В статье рассмотрены аспекты параллельной реализации одного из наиболее перспективных алгоритмов обучения с учителем – градиентного бустинга деревьев решений. Предложена и реализована параллельная схема алгоритма обучения модели для задачи классификации объектов с большим числом классов для систем с распределенной памятью. Эффективность предложенного подхода подтверждена результатами вычислительных экспериментов.

Литература

1. Breiman L. Random Forests. // Machine Learning. 2001. V. 45, №. 1, P. 5-32.
2. Breiman L., Friedman J., Olshen R., Stone C. Classification and Regression Trees. Wadsworth, 1983.
3. Breiman L. Bagging predictors // Machine Learning. 1996. V. 26, № 2, P. 123-140.
4. Druzhkov P. N., Eruhimov V. L., Kozinov E. A., Kustikova V. D., Meyerov I. B., Polovinkin A. N., Zolotykh N. Yu. On some new object detection features in OpenCV Library // Pattern Recognition and Image Analysis. 2011. V. 21, № 3. P. 384–386.
5. Enzweiler M., Gavrila D. M . Monocular Pedestrian Detection: Survey and Experiments // IEEE Transactions on Pattern Analysis and Machine Intelligence. 2009. V.31, № 12. P. 2179–2195.
6. Freund Y., Schapire R. Experiments with a New Boosting Algorithm // Machine Learning: Proceedings of the Thirteenth International Conference. 1996.
7. Friedman J. H. Greedy Function Approximation: a Gradient Boosting Machine. Technical report. Dept. of Statistics, Stanford University, 1999.
8. Friedman J. H. Stochastic Gradient Boosting. Technical report. Dept. of Statistics, Stanford University, 1999.
9. Geurts P., Ernst D., Wehenkel L. Extremely Randomized Trees // Machine Learning. 2006. V. 36, № 1. P. 3–42.
10. Hastie T., Tibshirani R., Friedman J. The Elements of Statistical Learning. Springer, 2008.
11. OpenCV Wiki.
URL: http://opencv.willowgarage.com/wiki (дата обращения: 02.12.2011).
12. Panda B., Herbach J.S., Basu S., Bayardo R.J. PLANET: massively parallel learning of tree ensembles with MapReduce // VLDB Endow. 2009. V. 2, №. 2, P. 1426–1437.
13. Tyree S., Weinberger K.Q., Agrawal K., Paykin J. Parallel boosted regression trees for web search ranking // WWW. 2011. P. 387-396.
14. UCI Machine Learning Repository.
URL: http://archive.ics.uci.edu/ml (дата обращения: 02.12.2011).
15. Вапник В. Н. Восстановление зависимостей по эмпирическим данным. М.: Наука, 1979. 448 с.
16. Дружков П.Н., Золотых Н.Ю., Половинкин А.Н. Программная реализация алгоритма градиентного бустинга деревьев решений // Вестник Нижегородского государственного университета им. Н. И. Лобачевского. 2011. № 1, С. 193–200.
17. Дружков П.Н., Золотых Н.Ю., Половинкин А.Н. Реализация параллельного алгоритма предсказания в методе градиентного бустинга деревьев решений // Вестник ЮУрГУ. Сер. «Математическое моделирование и программирование». 2011. Вып. 10, №37(254), С. 82 – 89.