На правах рукописи

Трофимов Илья Егорович

Разработка и обоснование методов параллельного покоординатного спуска для обучения обобщенных линейных моделей с регуляризацией

05.13.17 - теоретические основы информатики

Автореферат диссертации на соискание ученой степени кандидата физико-математических наук

Москва - 2018

Работа выполнена в Федеральном исследовательском центре "Информатика и управление" Российской Академии Наук.

Научный руководитель	**Воронцов Константин Вячеславович** доктор физико-математических наук, профессор, заведующий лабораторией машинного интеллекта ФГАОУ ВО «Московский физико-технический институт (ГУ)»
Официальные оппоненты	**Аветисян Арутюн Ишханович** доктор физико-математических наук, член-корр. РАН, профессор, директор Института системного программирования им. В.П. Иванникова РАН
	Бурнаев Евгений Владимирович кандидат физико-математических наук, доцент «Центра по научным и инженерным вычислительным технологиям для задач с большими массивами данных» Сколковского института науки и технологий
Ведущая организация	Институт проблем управления им. В.А. Трапезникова РАН

Защита состоится 28 февраля 2019 г. в 15:00 на заседании диссертационного совета Д 002.073.05 при Федеральном исследовательском центре "Информатика и управление" Российской Академии Наук (ФИЦ ИУ РАН) по адресу: 11933, г. Москва, ул. Вавилова д.40.

С диссертацией можно ознакомиться в библиотеке ФИЦ ИУ РАН и на сайте `http://www.frccsc.ru`.

Автореферат разослан: _____

Ученый секретать
диссертационного
совета Д 002.073.05,
д.ф.-м.н., профессор  Рязанов В.В.

Общая характеристика работы

Работа посвящена разработке и обоснованию методов параллельного покоординатного спуска для обучения обобщенных линейных моделей с регуляризацией. Разработанные методы применимы для больших обучающих выборок и выполнения на вычислительном кластере.

Актуальность темы. В настоящее время во многих задачах машинного обучения и анализа данных возникают большие обучающие выборки. В качестве примера можно привести задачи поиска в интернете, онлайн рекламы, обработки текстов, анализа показателей датчиков, генетики и т.д. Такие задачи характеризуются большим числом обучающих примеров, высокой размерностью, или и тем и другим одновременно. Обучающие выборки, как правило, разреженные. Желательным свойством также является разреженность полученного решения. Если в этих задачах использовать для обучения только часть имеющихся данных, то качество прогноза, как правило, падает. Поэтому важным направлением исследований является разработка методов машинного обучения, специально предназначенных для больших выборок, а также разработка алгоритмов, позволяющих применять существующие методы на больших выборках.

Обобщенные линейные модели (generalized linear models, GLM) - это класс статистических моделей, в которых предполагается, что зависимая переменная y связана с вектором независимых переменных $\mathbf{x}$ через нелинейную функцию от скалярного произведения с вектором весов $\boldsymbol{\beta}^T\mathbf{x}$. В класс обобщенных линейных моделей входят: логистическая регрессия, пробит регрессия, пуассоновская регрессия, линейная регрессия с квадратичной функцией потерь и некоторые другие статистические модели. Для обеспечения устойчивости к переобучению и стабильной сходимости численных методов обычно используется регуляризация. Наиболее часто используется L_1 или L_2 регуляризация. Также иногда применяются одновременно оба вида регуляризации, данный метод получил название elastic net. Для случая категориальных переменных используется регуляризатор group lasso. Более редким является использование невыпуклых регуляризаторов SCAD или MCP.

Обучение GLM сводится к минимизации целевой функции - суммы эмпирического риска и регуляризации. Это является задачей

численной оптимизации. Использование L_2 регуляризации - более простое, т.к. соответствующая целевая функция выпуклая и гладкая. Однако случай негладкого L_1 регуляризатора является более сложным. Существует несколько подходов к обучению обобщенных линейных моделей на больших обучающих выборках с L_1 и L_2 регуляризацией.

Последовательные алгоритмы

Первый подход - онлайн обучение. В онлайн обучении элементы из обучающей выборки обрабатываются по одному. Поэтому необязательно хранить обучающую выборку в оперативной памяти, ее можно читать последовательно с диска или получать по сети. Примерами таких методов являются: стохастический градиентный спуск (SGD), RMMP, онлайн обучение с усеченным градиентом (online learning via truncated gradient)и метод FTRL-Proximal. С одной стороны, данные методы позволяют получить регрессию приемлемого качества за небольшое число проходов по обучающей выборке. С другой стороны, в них присутствует несколько гиперпараметров (темп обучения, скорость затухания темпа обучения, количество проходов и др.) от которых существенно зависит качество регрессии. На практике оптимальные гиперпараметры подбираются с использованием тестовой выборки или кросс-валидации. Эта задача может быть сложной, так как процедура обучения на большой обучающей выборке обычно занимает продолжительное время. Полезная особенность, отличающей онлайн обучения от остальных методов - это возможность обучения в реальном времени (по мере прихода свежих данных), что позволяет подстраиваться под изменяющееся распределение.

Второй подход - это методы покоординатного спуска. Методы покоординатного спуска по очереди обновляют одну или несколько переменных, стремясь минимизировать целевую функцию или приближение к ней. Методы покоординатного спуска универсальны и позволяют работать как с L_1, так и с L_2 регуляризацией. В статье [1] проведено сравнение большого числа методов для реше-

[1] A Comparison of Optimization Methods and Software for Large-scale L1-regularized Linear Classffication / Guo-Xun Yuan, Kai-Wei Chang, Cho-Jui Hsieh, Chih-Jen Lin // Journal of Machine Learning Research - 2010 - Vol. 11 - P. 3183-3234.

ния задачи линейной классификации с L_1 регуляризацией и сделан вывод, что методы покоординатного спуска работают лучше всего. На практике чаще всего используются следующие алгоритмы этого типа: BBR, GLMNET, newGLMNET. Все эти методы работают последовательно на одном сервере. Также их программные реализации требуют загрузки обучающей выборки в оперативную память (RAM).

Третий подход - квазиньютоновские методы Limited Memory BFGS (L-BFGS), TRON. Квазиньютоновские методы эффективно решают задачу оптимизации в случае выпуклой гладкой целевой функции, т.е. применимы только для L_2 регуляризации.

Параллельные алгоритмы

В последнее время все чаще встречаются задачи, в которых обучающие выборке настолько большие, что даже описанные методы (предназначенные для выполнения на одном сервере) работают слишком долго. Для решения этой проблемы необходимо модифицировать описанные выше методы, адаптировав их для параллельного выполнения на одном сервере с несколькими процессорами или в распределенной системе.

Универсальным способом параллельной оптимизации является метод "Alternating direction method of multipliers" (ADMM). Он применим для задач оптимизации, возникающих при обучении GLM с регуляризацией. Разные варианты ADMM обеспечивают параллелизм как по обучающим примерам, так и по переменным. Обратной стороной универсальности является медленная сходимость метода ADMM.

Алгоритмы онлайн обучения могут выполняться параллельно в распределенных системах. Обучающая выборка разделяется между серверами по примерам, на каждой подвыборке обучается независимо обучаются классификаторы и усредняются после каждой эпохи. Усредненный вектор весов используется как начальное приближение на следующей эпохе, и т.д. Данный подход имеет те же плюсы и минусы, что и последовательное онлайн обучение на одном сервере. Ускорение от количества узлов кластера - сублинейное, что является недостатком.

Квазиньютоновские методы Limited Memory BFGS (L-BFGS), TRON могут быть эффективно распараллелены для выполнения

на кластере при разбиении обучающей выборки по примерам.

Естественное обобщение методов покоординатного спуска - это выполнение параллельных шагов по нескольким переменным. Именно так работает алгоритм Shotgun. Он основан на параллельных шагах по случайно выбранным переменным. Недостатком этого алгоритма является то, что существует верхний предел по количеству параллельных обновлений, при котором алгоритм сходится. Этот предел зависит от обучающей выборки и его теоретические оценки тяжело вычислимы на практике. В алгоритме GRock множество переменных для обновления выбирается жадно, т.е. шаги выполняются по нескольким переменным, дающим наибольшее уменьшение целевой функции. Проведенные численные эксперименты показали, что алгоритм GRock сходится быстрее, чем параллельный вариант FISTA и ADMM. Алгоритм Shotgun может быть запущен в распределенной системе с помощью технологии Stale Synchronous Parallel Parameter Server (SSPPS). Также, как и стандартный Shotgun, данный вариант не всегда сходится.

Альтернативный подход к распараллеливанию покоординатного спуска состоит в том, чтобы, не меняя последовательности вычислений (с точки зрения математики), ускорить их, распределив операции между процессорами. Плюсом данного подхода является гарантированная сходимость и хорошее ускорение. Недостатком является необходимость загружать обучающую выборку в память GPU (которая меньше RAM) а также относительная дороговизна серверов с GPU.

Цель диссертационной работы - исследование методов покоординатного спуска для обучения обобщенных линейных моделей с регуляризацией. Разработка методов, применимых для больших обучающих выборок и выполнения на вычислительном кластере.

Методы исследования. В данной работе использованы, с одной стороны, теоретические методы исследования, опирающиеся на математический анализ и линейную алгебру; с другой стороны, используется метод численного эксперимента на вычислительном кластере.

Основные положения, выносимые на защиту:

1. Метод минимизации функций риска обобщенных линейных моделей с регуляризацией “elastic net” - `d-GLMNET`.

2. Достаточные результаты сходимости метода и линейной ско-

рости сходимости метода d-GLMNET.

3. Метод "асинхронной балансировки нагрузки" для обеспечения эффективного выполнения метода d-GLMNET при наличии медленных узлов кластера.

4. Численные эксперименты, доказывающие, что метод d-GLMNET получает разреженные решения при использовании L_1 регуляризации.

5. Численные эксперименты, доказывающие, что метод d-GLMNET более эффективен, чем общепринятые методы при работе с разреженными обучающими выборками с высокой размерностью признакового пространства.

6. Общедоступная программная реализация метода d-GLMNET: https://github.com/IlyaTrofimov/dlr

Научная новизна данной работы заключается в разработке нового метода минимизации функций риска обобщенных линейных моделей с регуляризацией "elastic net". Метод основан на параллельном покоординатном спуске. Метод эффективно работает с большими обучающими выборками (big data) с использованием вычислительного кластера. Научной новизной обладают теоретические результаты относительно сходимости метода, а также модификация метода (асинхронная балансировка нагрузки), обеспечивающая эффективное выполнение при неравномерности скорости работы узлов кластера.

Теоретическая значимость состоит в установлении достаточных условий сходимости и линейной скорости сходимости разработанного метода d-GLMNET, в том числе, для модификации метода d-GLMNET, использующей технику асинхронной балансировки нагрузки.

Практическая значимость определяется тем, что разработанный метод d-GLMNET позволяет проводить обучение обобщенных линейных моделей быстрее, чем при использовании общепринятых методов, что позволяет экономить вычислительные ресурсы и получать более точные решения при ограниченном бюджете вычислительных ресурсов.

Теоретические и экспериментальные результаты данной работы используются в курсах “Машинное обучение и большие данные”, которые автор читал в 2015-2018 гг. на факультете инноваций и высоких технологий (ФИВТ) МФТИ и Школе анализа данных (ШАД) Яндекса.

Степень достоверности. Достоверность результатов обеспечивается доказательствами теорем и описаниями выполненных экспериментов, допускающими их воспроизводимость. Исходный код программ и выборки, использовавшиеся для численных экспериментов общедоступны.

Апробация работы. Основные положения и результаты работы докладывались автором на конференциях:

- Sixth International Workshop on Data Mining for Online Advertising and Internet Economy, 2012, Пекин;
- 22nd International Conference on World Wide Web, 2013, Рио-де-Жанейро;
- Machine learning and Very Large Data Sets, 2013, Москва;
- Seventeenth International Conference on Artificial Intelligence and Statistics, 2014, Рейкьявик;
- The 4-th International Conference on Analysis of Images, Social Networks and Texts, 2015, Екатеринбург;
- Machine Learning: Prospects and Applications, 2015, Берлин.

Публикации. По тематике исследований опубликовано 5 статей, в том числе 2 статьи в изданиях, входящих в список ВАК и 2 статьи, входящие в индекс SCOPUS .

Структура работы. Диссертация состоит из введения, 5 глав, заключения, приложения и списка литературы. Материал изложен на 115 стр., содержит 19 рисунков, 5 таблиц, 18 алгоритмов и 103 наименований в списке литературы.

Личный вклад диссертанта является решающим во всех результатах, выносимых на защиту.

Основное содержание работы

Во **введении** обосновывается актуальность выбранной темы исследования - обучение на больших выборках (big data) обобщенных линейных моделей с регуляризацией, с использованием вычислительного кластера. Обзор литературы показывает, что методы покоординатного спуска для данного класса задач, предназначенные для выполнения на кластере недостаточно разработаны, что и становится темой диссертации.

В **главе 1** вводятся основные понятия и определения, которые будут использоваться в работе. Обобщенные линейные модели (GLM) - это класс статистических моделей, в которых предполагается, что зависимая переменная Y имеет распределение с мат. ожиданием μ

$$\mathbb{E}[Y] = \mu = g^{-1}(\boldsymbol{\beta}^T \mathbf{x})$$

и дисперсией, зависящей только от мат. ожидания

$$\mathbb{D}[Y] = \phi V(\mu)$$

Функция $g(\cdot)$ называется функцией связи (link function), а величина ϕ - параметром дисперсии. Наиболее часто используемые на практике распределения Y принадлежат к экспоненциальному семейству, т.е. функции плотности распределения имеет вид

$$P(y|\theta, \phi) = \exp\left(\frac{y\theta - b(\theta)}{\phi + c(y, \phi)}\right)$$

Можно показать, что для экспоненциального семейства

$$\mathbb{E}[Y] = b'(\theta), \quad \mathbb{D}[Y] = \phi b''(\theta)$$

Функция связи называется *канонической*, если $g(\cdot) = (b')^{-1}(\cdot)$. Все рассматриваемые в этом разделе GLM, кроме пробит-регрессии, имеют каноническую функцию связи. В этом случае имеем

$$g(\mu) = g(\mathbb{E}[Y]) = g(b'(\theta)) = \theta = \boldsymbol{\beta}^T \mathbf{x}$$

Таким образом, распределение Y зависит от $\boldsymbol{\beta}$ только через линейную комбинацию $\boldsymbol{\beta}^T \mathbf{x}$.

$$P(y|\mathbf{x}) = \exp\left(\frac{y\boldsymbol{\beta}^T\mathbf{x} - b(\boldsymbol{\beta}^T\mathbf{x})}{\phi + c(y, \phi)}\right)$$

Наиболее часто используемые GLM:

- Линейная регрессия, $y \in \mathbb{R}$:

$$P(y|\mathbf{x}) = \frac{1}{\sqrt{2\pi}} \exp\left(\frac{(y - \boldsymbol{\beta}^T\mathbf{x})^2}{2}\right)$$

- Логистическая регрессия, $y \in \{-1, +1\}$:

$$P(y|\mathbf{x}) = \frac{1}{1 + \exp(-y\boldsymbol{\beta}^T\mathbf{x})}$$

- Пробит регрессия, $y \in \{-1, +1\}$:

$$P(y|\mathbf{x}) = \Phi(y\boldsymbol{\beta}^T\mathbf{x})$$

  где $\Phi(\cdot)$ - это функция распределения стандартного нормального распределения

- Пуассоновская регрессия, $y \in \mathbb{N}$:

$$P(y|\boldsymbol{\beta}^T\mathbf{x}) = \frac{\exp(-\lambda)\lambda^n}{n!}, \text{ где } \lambda = \exp(\boldsymbol{\beta}^T\mathbf{x})$$

Обучение GLM выполняется через максимизацию правдоподобия на обучающей выборке $\{\mathbf{x}_i, y_i\}_{i=1}^n$

$$\max_{\boldsymbol{\beta}} \prod_{i=1}^{n} P(y_i|\boldsymbol{\beta}^T\mathbf{x}_i)$$

что эквивалентно минимизации минус лог-правдоподобия

$$\min_{\boldsymbol{\beta}} \left\{ -\sum_{i=1}^{n} \log P(y_i|\boldsymbol{\beta}^T\mathbf{x}_i) \right\}$$

Обозначим

$$\ell(y_i, \boldsymbol{\beta}^T\mathbf{x}) = -\log P(y_i|\boldsymbol{\beta}^T\mathbf{x}_i)$$

Функция $\ell(y, \hat{y})$ называется *функцией потерь* и может быть интерпретирована как штраф за отличие истинного y от предсказанного $\hat{y}$. Таким образом, задачи классификации и регрессии свелись к задаче оптимизации

$$\min_{\boldsymbol{\beta}} L(\boldsymbol{\beta}), \qquad L(\boldsymbol{\beta}) = \sum_{i=1}^{n} \ell(y_i, \boldsymbol{\beta}^T\mathbf{x}_i). \tag{1}$$

Данная задача может быть рассмотрена напрямую для заданной функции $\ell(y, \hat{y})$, без статистической интерпретации через обобщенные линейные модели. Функция $L(\boldsymbol{\beta})$ называется также *функцией эмпирического риска*.

Задача (1) обычно решается с помощью численных методов оптимизации. Качество решения может быть неудовлетворительным по следующим причинам:

- Решение может значительно изменяться при сколь угодно малых изменениях обучающей выборки (некорректно поставленная задача);
- Численные методы могут работать нестабильно;
- Решение может обладать плохой обобщающей способностью.

Для устранения этих проблем к функции риска добавляют регуляризацию $R(\boldsymbol{\beta})$ и решается задача минимизации *регуляризованного эмпирического риска*

$$\boldsymbol{\beta}^* = \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \left(L(\boldsymbol{\beta}) + R(\boldsymbol{\beta}) \right) \tag{2}$$

Перечислим наиболее часто используемые виды регуляризации:

1. L_1 регуляризация

$$R(\boldsymbol{\beta}) = \lambda_1 \|\boldsymbol{\beta}\|_1 = \lambda_1 \sum_{k=1}^{p} |\beta_k|$$

Обладает тем свойством, что у решения часть компонент будут нулевыми. Чем больше λ_1 - тем больше нулевых компонент. Таким образом, L_1 регуляризация выполняет также отбор признаков. Кроме того, L_1 регуляризация имеет тенденцию отбирать по одному признаку из группы сильно коррелированных. Функция $R(\boldsymbol{\beta})$ - не гладкая в точке $\boldsymbol{\beta} = 0$.

2. L_2 регуляризация

$$R(\boldsymbol{\beta}) = \lambda_2 \|\boldsymbol{\beta}\|_2^2 = \frac{\lambda_2}{2} \sum_{k=1}^{p} \beta_k^2$$

Функция $R(\boldsymbol{\beta})$ - гладкая и не приводит к занулению компонент $\boldsymbol{\beta}$. Имеет тенденцию уменьшать по абсолютному значению коэффициенты $\boldsymbol{\beta}$, значения которых не могут быть достоверно определены по обучающей выборке.

3. Комбинация L_1 и L_2 регуляризации носит название "elastic net"

$$R(\boldsymbol{\beta}) = \lambda_1 \|\boldsymbol{\beta}\|_1 + \frac{\lambda_2}{2}\|\boldsymbol{\beta}\|_2^2$$

Выполняет отбор признаков также, как и L_1 регуляризация. В тоже время, elastic net регуляризация имеет тенденцию оставлять группу коррелированных коэффициентов целиком. Elastic net регуляризация не гладкая в точке $\boldsymbol{\beta} = 0$.

Более редкие типы регуляризации: групповое лассо (group lasso), Smoothly Clipped Absolute Deviation (SCAD), Minimax Convex Penalty (MCP), Sparse Laplacian Shrinhage (SLS).

Регуляризаторы L_1, L_2, elastic net, group lasso являются выпуклыми. Поэтому задача (2) является задачей выпуклой оптимизации. Если регуляризатор гладкий (L_2), то решение задачи упрощается. Случай негладкого регуляризатора более сложный с точки зрения оптимизации.

В конце главы 1 приведен краткий обзор современных методов оптимизации для решения задачи (2), которые применимы к большим обучающим выборкам : метод онлайн обучения с усеченным градиентом, методы Shooting, GLMNET, newGLMNET, L-BFGS.

В **главе 2** сначала описываются наиболее популярные архитектуры вычислительных систем для распределенного машинного обучения (Map/Reduce, MPI, сервера параметров, Spark, GraphLab). После этого описываются модификации методов из главы 1 для исполнения на вычислительном кластере, анализируются их достоинства и недостатки.

В **главе 3** описывается предложенный автором метод параллельного покоординатного спуска - `d-GLMNET`. В методе `d-GLMNET` выполняются параллельные шаги по блокам переменных. Предположим, что множество из p признаков разбито на M непересекающихся подмножеств S^m

$$\bigcup_{m=1}^{M} S^m = \{1, \ldots, p\}, \quad S_m \cap S_k = \varnothing, k \neq m.$$

Тогда каждый из узлов кластера $m = 1 \dots M$ выполняет шаг по подмножеству переменных $\Delta\boldsymbol{\beta}^m$, решая оптимизационную задачу

$$\underset{\Delta\boldsymbol{\beta}^m}{\operatorname{argmin}} \left\{ L_q(\boldsymbol{\beta}, \Delta\boldsymbol{\beta}^m) + R(\boldsymbol{\beta} + \Delta\boldsymbol{\beta}^m) \right\}, \tag{3}$$

$$L_q(\boldsymbol{\beta}, \Delta\boldsymbol{\beta}) \stackrel{\text{def}}{=} L(\boldsymbol{\beta}) + \nabla L(\boldsymbol{\beta})^T \Delta\boldsymbol{\beta} + \frac{1}{2} \Delta\boldsymbol{\beta}^T H(\boldsymbol{\beta}) \Delta\boldsymbol{\beta},$$

при ограничении $\Delta\beta_j^m = 0$ если $j \notin S^m$.

Далее доказывается

Теорема 1. *Независимая оптимизация (3) по блокам переменных $\Delta\boldsymbol{\beta}^m$ эквивалентна оптимизации квадратичного приближения к целевой функции*

$$\underset{\Delta\boldsymbol{\beta}}{\operatorname{argmin}} \left\{ L(\boldsymbol{\beta}) + \nabla L(\boldsymbol{\beta})^T \Delta\boldsymbol{\beta} + \frac{1}{2} \Delta\boldsymbol{\beta}^T \widetilde{H}(\boldsymbol{\beta}) \Delta\boldsymbol{\beta} + R(\boldsymbol{\beta} + \Delta\boldsymbol{\beta}) \right\} \tag{4}$$

с блочно-диагональным $\widetilde{H}(\boldsymbol{\beta})$ приближением Гессиана

$$(\widetilde{H}(\boldsymbol{\beta}))_{jl} = \begin{cases} (\nabla^2 L(\boldsymbol{\beta}))_{jl}, \text{ если } \exists m : \ j, l \in S^m, \\ 0, \quad \text{иначе.} \end{cases} \tag{5}$$

Подробный вывод шага метода `d-GLMNET` приведен в Приложении 1 к диссертации. Шаг метода вычисляется по формуле

$$\Delta\boldsymbol{\beta} = \alpha \sum_{m=1}^{M} \Delta\boldsymbol{\beta}^m,$$

где α - мультипликатор, подбираемый по правилу Армихо. Нужно выбрать α равное наибольшему элементу последовательности $\{\alpha_{init} b^j\}_{j=0,1,\dots}$ удовлетворяющему критерию Армихо

$$f(\boldsymbol{\beta} + \alpha\Delta\boldsymbol{\beta}) \leq f(\boldsymbol{\beta}) + \alpha\sigma D, \tag{6}$$

где

$$f(\boldsymbol{\beta}) = L(\boldsymbol{\beta}) + R(\boldsymbol{\beta}),$$
$$D = \nabla L(\boldsymbol{\beta})^T \Delta\boldsymbol{\beta} + \gamma \Delta\boldsymbol{\beta}^T (\mu(\widetilde{H}(\boldsymbol{\beta}) + \nu I)) \Delta\boldsymbol{\beta} + R(\boldsymbol{\beta} + \Delta\boldsymbol{\beta}) - R(\boldsymbol{\beta}),$$
$$\delta > 0, 0 < b < 1, 0 < \sigma < 1, 0 \leq \gamma < 1.$$

Имеет место

Теорема 2. *Линейный поиск по правилу Армихо завершится за конечное число шагов, если на любом отрезке* $\hat{y} \in [a, b]$, $\partial^2 \ell(y, \hat{y})/\partial \hat{y}^2$ *ограничена сверху.*

Для обеспечения разреженности решения на каждом шаге минимизируется *модифицированное* локальное квадратичное приближение

$$L_q^{gen}(\boldsymbol{\beta}, \Delta\boldsymbol{\beta}) \stackrel{\text{def}}{=} L(\boldsymbol{\beta}) + \nabla L(\boldsymbol{\beta})^T \Delta\boldsymbol{\beta} + \frac{1}{2}\Delta\boldsymbol{\beta}^T(\mu(\widetilde{H}(\boldsymbol{\beta}) + \nu I))\Delta\boldsymbol{\beta},$$

и на каждой итерации решается задача

$$\underset{\Delta\boldsymbol{\beta}}{\operatorname{argmin}} \left\{ L_q^{gen}(\boldsymbol{\beta}, \Delta\boldsymbol{\beta}) + R(\boldsymbol{\beta} + \Delta\boldsymbol{\beta}) \right\}. \tag{7}$$

Решение задачи одномерной оптимизации по $\Delta\beta_j$ имеет вид:

$$\Delta\beta_j^* = \frac{S\left(\sum_{i=1}^n w_i x_{ij} r_i + \nu\beta_j, \lambda_1\right)}{\mu \sum_{i=1}^n w_i x_{ij}^2 + \lambda_2 + \nu} - \beta_j, \tag{8}$$

$$r_i = z_i - \mu(\Delta\boldsymbol{\beta}^T \mathbf{x}_i + (\beta_j + \Delta\beta_j)x_{ij}),$$

$$w_i = \frac{\partial^2 \ell(y_i, \boldsymbol{\beta}^T x_i)}{\partial \hat{y}^2},\ z_i = -\frac{\partial \ell(y_i, \boldsymbol{\beta}^T x_i)/\partial \hat{y}}{\partial^2 \ell(y_i, \boldsymbol{\beta}^T x_i)/\partial \hat{y}^2}.$$

Здесь $S(\cdot)$ - это функция soft-threshold

$$S(x, a) = \operatorname{sgn}(x) \max(|x| - a, 0). \tag{9}$$

Параметр $\mu \geq 1$ подбирается адаптивно по ходу алгоритма. Доказывается

Теорема 3. *Пусть* $\Lambda_{max}, \lambda_{min}$ *- максимальное и минимальное собственные значения* $H(\boldsymbol{\beta})$ *и* $\widetilde{H}(\boldsymbol{\beta})$ *соответственно. Тогда если* $\mu \geq \frac{\Lambda_{max}}{(1-\sigma)\lambda_{min}}$, *по критерий Армихо (6) с* $\gamma = 0$ *будет выполнен для* $\alpha = 1$.

В **разделе 3.3** получены достаточные условия сходимости метода `d-GLMNET`. Доказательства опираются на результаты, полученные для семейства методов CGD (цикличный блочно-координатный спуск).

Теорема 4. *Достаточные условия сходимости метода* `d-GLMNET`

- *Существование и единственность минимума функции* $L(\boldsymbol{\beta})+R(\boldsymbol{\beta})$*;*
- *На любом отрезке* $\hat{y} \in [a,b]$, $\partial^2\ell(y,\hat{y})/\partial\hat{y}^2$ *ограничена сверху.*

Функция $\partial^2\ell(y,\hat{y})/\partial\hat{y}^2$ будет ограничена сверху на любом отрезке если она непрерывна, данное условие проще проверяется на практике. Это будет выполняться для линейной и логистической регрессии с любой регуляризацией, а также для Пуассоновской регрессии при наличии L_2 регуляризатора (возможно, одновременно с L_1 регуляризатором).

В **разделе 3.4** получены достаточные условия *линейной скорости* сходимости метода `d-GLMNET`.

Теорема 5. *Линейная скорость сходимости метода* `d-GLMNET` *будет иметь место в случае строгой выпуклости* $L(\boldsymbol{\beta})$ *или при наличии* L_2 *регуляризатора.*

Это будет иметь место для линейной, логистической и пробит-регрессии при условии строгой выпуклости $L(\boldsymbol{\beta})$ (т.е. если нет линейно зависимых признаков $\mathbf{x}_i$). Кроме того, линейная скорость сходимости будет иметь место для всех обобщенных линейных моделей с L_2 регуляризацией, в частном случае для Пуассоновской регрессии с L_2 регуляризацией.

Теоремы о сходимости опираются на результаты [2], полученные для минимизации суммы двух функций

$$\min_{\boldsymbol{\beta}} F_c(\boldsymbol{\beta}) \stackrel{\text{def}}{=} L(\boldsymbol{\beta}) + cR(\boldsymbol{\beta}), \tag{10}$$

где $L(\boldsymbol{\beta})$ - гладкая (непрерывно дифференцируемая функция) на открытом множестве в $\mathbb{R}^n$ содержащем $dom\ R = \{\boldsymbol{\beta} \mid R(\boldsymbol{\beta}) < \infty\}$, а $R(\boldsymbol{\beta})$ - собственная выпуклая и полунепрерывная снизу функция, константа $c > 0$.

В **разделе 3.5** высокоуровнево описывается программная реализация метода `d-GLMNET`. Программа запускается на кластере

[2] Tseng Paul, Yun Sangwoon. A coordinate gradient descent method for nonsmooth separable minimization // Mathematical Programming. - 2009. - Aug. - Vol. 117, no. 1-2. - P. 387-423.

Алгоритм 1: Метод `d-GLMNET`

Вход : обучающая выборка $\{\mathbf{x}_i, y_i\}_{i=1}^n$, $\lambda_1 \geq 0$, $\lambda_2 \geq 0$, $\eta_1 \geq 1, \eta_2 \geq 1$, разбиение признаков $S^1, \ldots, S^M$

$\boldsymbol{\beta} \leftarrow 0$, $\mu \leftarrow 1$
Пока *не выполнено условие останова*
  Выполнить параллельно на M узлах:
    Выбрать $P^m \subseteq S^m$
    Минимизировать $L_q^{gen}(\boldsymbol{\beta}, \Delta\boldsymbol{\beta}^m) + R(\boldsymbol{\beta} + \Delta\boldsymbol{\beta}^m)$ по $\Delta\boldsymbol{\beta}^m$
    $\Delta\boldsymbol{\beta} \leftarrow \sum_{m=1}^M \Delta\boldsymbol{\beta}^m$
    Найти $\alpha \in (0, 1]$ с помощью линейного поиска
    $\boldsymbol{\beta} \leftarrow \boldsymbol{\beta} + \alpha\Delta\boldsymbol{\beta}$
  Если $\alpha < 1$ **тогда**
    $\mu \leftarrow \eta_1 \mu$
  Иначе
    $\mu \leftarrow \max(1, \mu/\eta_2)$
Возврат: $\boldsymbol{\beta}$

Map/Reduce, дополнительно используя стандарт MPI для передачи данных между узлами кластера. Данная связка удобна, так как, с одной стороны, кластера Map/Reduce широко распространены, с другой стороны, стандарт MPI обеспечивает высокую производительность.

В **разделе 3.6** описываются алгоритм программной реализации метода `d-GLMNET`, см. Алгоритм 1.

В **разделе 3.7** описываются модификация метода `d-GLMNET` - "асинхронная балансировка нагрузки" (ALB). Данная модификация ускоряет метод в случае, если узлы кластера имеют различающуюся производительность. Такая неравномерность часто встречается на практике и может быть связана как с неравномерностью разбиения обучающей выборки, так и с неравномерностью загрузки узлов кластера другими задачами. Проблема состоит в том, что так как в алгоритме необходима синхронизация между узлами кластера, то скорость алгоритма ограничена сверху скоростью *самого медленного узла.*

Идея асинхронной балансировки нагрузки состоит в том, что на каждой итерации делаются шаги только по части переменных $P_k^m \subseteq$

Алгоритм 2: Реализация метода `d-GLMNET` на вычислительном кластере

Вход : Обучающая выборка, λ_1, λ_2, разбиение признаков $S^1, \ldots, S^M$

Пока *не выполнено условие останова*
- Выполнить параллельно на M машинах:
  - Прочитать последовательно часть обучающей выборки X^m
  - Вычислить $\Delta\boldsymbol{\beta}^m$ и $X^m\Delta\boldsymbol{\beta}^m$ для весов из $P^m \subseteq S^m$
  - Суммировать вектора $X^m\Delta\boldsymbol{\beta}^m$ с помощью MPI_AllReduce:
    - $X\Delta\boldsymbol{\beta} \leftarrow \sum_{m=1}^{M} X^m\Delta\boldsymbol{\beta}^m$
  - Вычислить размер шага α используя линейный поиск по правилу Армихо
  - $\boldsymbol{\beta}^m \leftarrow \boldsymbol{\beta}^m + \alpha\Delta\boldsymbol{\beta}^m$
  - $X\boldsymbol{\beta} \leftarrow X\boldsymbol{\beta} + \alpha X\Delta\boldsymbol{\beta}$

Возврат: $\boldsymbol{\beta}$

Таблица 1: Характеристики использованных датасетов

dataset	size	#examples	#features	nnz (train)	avg nonzeros
epsilon	12 Gb	0.5×10^6	2000	8.0×10^8	2000
webspam	21 Gb	0.35×10^6	16.6×10^6	1.2×10^9	3727
yandex_ad	56 Gb	61.7×10^6	35×10^6	5.7×10^9	100

S^m. Узлы кластера выполняют шаги циклически по подмножествам переменных S^m (см. рис. 1). Итерация прекращается, если более κM узлов выполнили итерацию. По умолчанию $\kappa = 0.75$. Таким образом, быстрее узлы могут выполнить более одного шага по своим переменным.

Приводятся результаты численных экспериментов, свидетельствующих о том, что что метод ALB снижает время простоя узлов кластера и ускоряет сходимость. Теорема с сходимости метода `d-GLMNET` применима также при асинхронной балансировке нагрузки. В тоже время, линейная скорость сходимости не гарантируется.

В **главе 4** описываются численные эксперименты, проведенные для сравнения предложенного метода `d-GLMNET` с общепринятыми

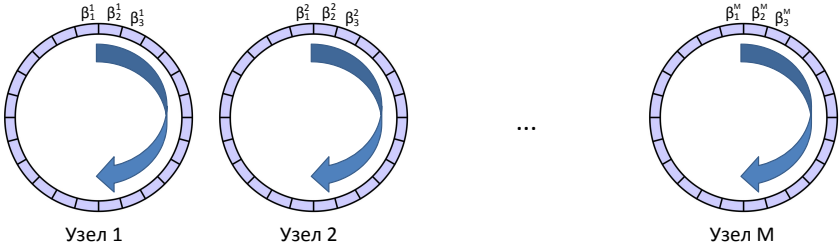

Рис. 1: Асинхронная балансировка нагрузки.

методами для обучения обобщенных линейных моделей с регуляризацией на вычислительном кластере - параллельное онлайн обучение, L-BFGS, ADMM. Характеристики использовавшихся для численных экспериментов датасетов приведены в таблице 1.

Датасеты *epsilon*, *webspam* - публичные, они использовались в Pascal Large Scale Learning Challenge 2008 [3]. Датасет *yandex_ad* - не публичный, содержит коммерческие данные компании Яндекс.

Эксперименты выполнялись на кластере, состоящем из серверов Intel(R) Xeon(R) CPU E5-2660 2.20GHz, 32 GB RAM, соединенных гигабитным Ethernet. Для всех датасетов вычисления проводились на 16 серверах, на каждом сервере запускался один экземпляр сравниваемой программы.

В **разделе 4.3** описаны численные эксперименты, доказывающие необходимость использования адаптивного алгоритма изменения параметра μ для получения разреженного решения в случае L_1 регуляризации.

В **разделе 4.4** описываются эксперименты с L_1 регуляризацией, в **разделе 4.5** - с L_2 регуляризацией.

Методы сравнивались по следующим характеристикам:

- Скорость достижения качества классификации на тестовой выборке.
- Скорость минимизации целевой функции оптимизации (вычислялась по обучающей выборке).
- Разреженность решения.

[3] `http://largescale.ml.tu-berlin.de/`

Таблица 2: Ускорение/замедление относительно лучшего из конкурирующих методов

dataset	**L1**		**L2**	
	ALB		ALB	
webspam	**6.0**	**3.1**	**5.5**	**2.9**
yandex_ad	**13.3**	**10.0**	**1.9**	**1.4**
epsilon	0.7	0.55	0.4	0.3

Выводы из численных экспериментов. Для датасетов *webspam*, *yandex_ ad*, метод `d-GLMNET` быстрее оптимизирует целевую функцию и достигает качества на тестовой выборке, чем конкурирующие алгоритмы (см. таблицу 2). Отличительной особенностью этих датасетов является большое число признаков и высокая разреженность. Для датасете *epsilon* метод `d-GLMNET` медленнее, чем ADMM (случай L_1 регуляризации) и комбинирование онлайн-обучения и L-BFGS (случай L_2 регуляризации).

Что касается метода асинхронной балансировки нагрузки `d-GLMNET-ALB`, то он сходится быстрее или, как минимум, с такой же скоростью, как исходный метод `d-GLMNET`.

Отдельно было исследовано ускорения метода `d-GLMNET` в зависимости от числа используемых узлов кластера (степень параллелизма). Ускорение - сублинейное и достигает максимума для некоторого предельного числе узлов, зависящего от датасета.

В **главе 5** описывается применение методов обучения обобщенных линейных моделей с регуляризацией на больших выборках для задачи прогнозирования вероятности клика в онлайн рекламе. Один из подходов к решению этой задачи - это комбинирование бустинга деревьев решений и логистической регрессии.

Прогнозирование вероятности клика по онлайн рекламе - важная задача для оптимизации работы онлайн аукциона рекламы. Рекламодатели платят за клик по объявлению и поэтому для показа пользователя отбираются рекламные объявления с максимальным значением $P(click) * bid$, где bid - ставка рекламодателя за клик. Таким образом, более точное прогнозирование вероятности клика приводит к росту прибыли рекламной площадки (поиск Яндекса).

В **разделе 5.1** описывается алгоритм отбора объявлений в поисковой рекламе Яндекса, а также используемый метод прогноза

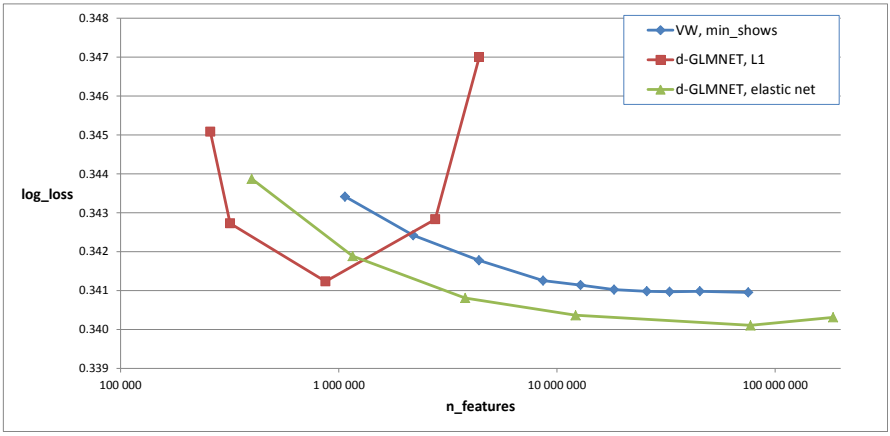

Рис. 2: Сравнение разреженности и ошибки на тесте методов обучения логистической регрессии

вероятности клика. Далее, в **разделе 5.2** приведены численные эксперименты, показывающие превосходство метода `d-GLMNET` с elastic-net регуляризацией над используемым в настоящий момент в Яндексе методом.

Как следует из рис. 2, метод `d-GLMNET` показывает лучшие результаты, в том смысле, что для фиксированной степени разреженности решения, ошибка на тестовой выборке минимальна. Вариант обучения, использовавшийся в Яндексе на момент численного эксперимента - фильтрация с $n_{min} = 5$. Таким образом, выбирая параметры elastic net регуляризации можно найти варианты, в которых относительно использующегося в Яндексе метода:

- Точность прогноза такая же, но решение в 7 раз более разреженное: 26×10^6 против 3.7×10^6 ненулевых компонент $\boldsymbol{\beta}$.
- Точность прогноза лучше на 0.26% (уменьшение log_loss), что является хорошим улучшением.

Разреженность решения важна на практике из-за ограниченности памяти серверов, на которых модель выполняет прогнозы в реальном времени. Время обучения составляло 2-3 ч. при различных параметрах для обеих программ.

Заключение

Основные результаты исследования:

1. Предложен новый метод минимизации функций риска обобщенных линейных с регуляризацией “elastic net” - `d-GLMNET`.
2. Получены достаточные условия сходимости и *линейной скорости* сходимости метода `d-GLMNET`.
3. Доказана возможность получать разреженные решения с помощью метода `d-GLMNET` при использовании L_1-регуляризации.
4. Предложен метод “асинхронной балансировки нагрузки” для обеспечения эффективного выполнения при неравномерной производительности узлов кластера.
5. Проведены численные эксперименты, доказывающие, что метод `d-GLMNET` более эффективен, чем общепринятые методы при работе с разреженными обучающими выборками с высокой размерностью признакового пространства.
6. Разработана программная реализация метода `d-GLMNET`: `https://github.com/IlyaTrofimov/dlr`

Список работ, опубликованных автором по теме диссертации

Публикации из перечня ВАК

1. Trofimov I., Genkin A. Distributed coordinate descent for generalized linear models with regularization // Pattern Recognition and Image Analysis. – 2017. – Т. 27. – №. 2. – С. 349-364.

2. Трофимов И. Е. Распределенные вычислительные системы для машинного обучения // Информационные технологии и вычислительные системы. – 2017. – №. 3. – С. 56-69.

3. Trofimov I., Genkin A. Distributed coordinate descent for l1-regularized logistic regression // International Conference on Analysis of Images, Social Networks and Texts. – Communications in Computer and Information Science, vol 542. - Springer, Cham, 2015. – С. 243-254.

Другие публикации

1. Trofimov I., Kornetova A., Topinskiy V. Using boosted trees for click-through rate prediction for sponsored search // Proceedings of the Sixth International Workshop on Data Mining for Online Advertising and Internet Economy. – ACM, 2012. – С. 2.

2. Trofimov I. New features for query dependent sponsored search click prediction // Proceedings of the 22nd International Conference on World Wide Web. – ACM, 2013. – С. 117-118.