Методы второго порядка для дистрибутивной оптимизации

Исламов Рустем,
Научный руководитель: Питер Рихтарик

16 декабря 2020 г.

Введение

Проблема

Скорость роста размеров датасэтов очень большая, современные технологии не позволяют обрабатывать такие объемы данных на одном устройстве эффективно.

Возможное решение

Использование нескольких устройств, которые общаются через общий сервер.

Проблемы дистрибутивной оптимизации

Communication bottleneck

Наиболее узким местом в дистрибутивной оптимизации является передача данных на сервер и обратно. Обмен данными стоит намного дороже, чем локальные вычисления.

Возможное решение

Использование различных алгоритмов сжатия/кодирования информации, которые требует меньшего число бит при передаче информации на сервер.

Существующие методы дистрибутивной оптимизации

Методы первого порядка

- Вычисление градиента требует малых затрат;
- Существуют разнообразные эффективные методы сжатия градиента;
- Не теряется линейная скорость сходимости в сильно выпуклом случае.

Возможное улучшение

Использование методов второго порядка для ускорения.

Постановка задачи

Минимазиция эмпирического риска

$$\min_{w \in \mathbb{R}^d} \left[F(w) = \frac{1}{n} \sum_{i=1}^{n} f_i(w) \right], \tag{1}$$

где f_i имеет вид

$$f_i(w) = \frac{1}{m_i} \sum_{j=1}^{m_i} \ell_i(w, x_{ij}, y_{ij}). \tag{2}$$

ℓ_i — функция потерь, которую использует i-ый worker
(x_{ij}, y_{ij}) — элемент выборки i-го датасэта.

Предположения

Необходимые предположения

- Используются линейные модели $\ell_i(w, x, y) = \varphi(yw^\top x)$;
- Все функции $f_{ij}(w) = \ell_i(w, x_{ij}, y_{ij})$ являются сильными выпуклыми с константой μ;
- Все функции f_{ij} имеют липшицевый Гессиан с константой H

$$\left\|\nabla^2 f_{ij}(u) - \nabla^2 f_{ij}(v)\right\| \leq H\|u - v\|. \tag{3}$$

Фреймворк

- Считаем, что данные хранятся на сервере, доступ к которому имеют все worker-ы;
- Каждому worker-у соответствуют свои собственный элементы выборки.

Базовый алгоритм

MaxCoefficient Newton Method (MCNM)

Initialize: Choose starting iterates $x^0 \in \mathbb{R}^d$

for $k = 0, 1, 2, \ldots$ do in parallel

broadcast x^k to all workers ← master node

for $i = 0, 1, \ldots n$ do ← i-th node

compute $\alpha_{ij}^k = \varphi_{ij}''(y_{ij} w^\top x^k)$

compute $\beta_i^k = \max\limits_{j \in [m_i]} \alpha_{ij}^k$

broadcast β_i^k to master node

end for

$B_i^k = \frac{\beta_i^k}{m_i} \sum\limits_{j=1}^{m_i} x_{ij} x_{ij}^T$

$x^{k+1} = x^k - \left[\frac{1}{n} \sum\limits_{i=1}^{n} B_i^k\right]^{-1} \left[\frac{1}{n} \sum\limits_{i=1}^{n} \nabla f_i(x^k)\right]$

end for

Сходимость базового алгоритма

Теорема 1.

При введенных допущениях на функции алгоритм MCNM обладает линейной скоростью сходимости, при этом скорость сходимости задается формулой

$$\|x^{k+1} - x^*\| \leq \frac{C\|x^k - x^*\|}{\mu n} \sum_{i=1}^{n} \sum_{j=1}^{n} \|x_{ij}\|^2, \tag{4}$$

где $C = \max\limits_{i \in [n]} \sup\limits_{x \in \mathbb{R}^d} \left| \beta_i(x) - \int\limits_0^1 \alpha_i [x^* + \tau(x - x^*)] d\tau \right|$ определяет окрестность локальной сходимости.

Эксперименты с базовым алгоритмом

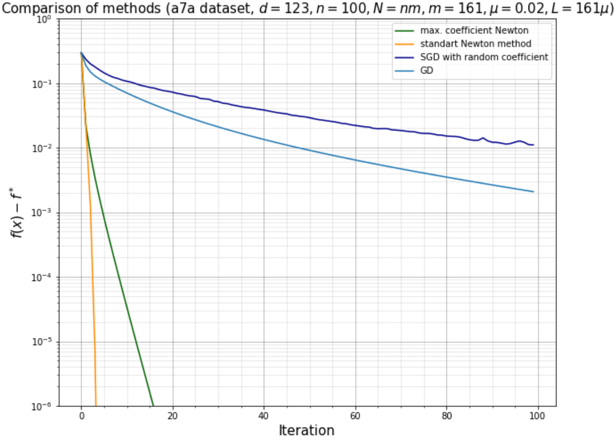

Рис.: Сравнение MCNM со стандартным NM, GD и CGD.

Модификация базового алгоритма

Дополнительное предположение

Временно считаем, что x^* — оптимальное решение — известно.

Scaled MaxCoefficient Newton Method (SMCNM)

Initialize: Choose starting iterates $x^0 \in \mathbb{R}^d$

for $k = 0, 1, 2, \dots$ do in parallel

broadcast x^k to all workers ← master node

for $i = 0, 1, \dots n$ do ← i-th node

compute $\alpha_{ij}^k = \varphi_{ij}''(y_{ij} w^\top x^k)$

compute $\beta_i^k = \max\limits_{j \in [m_i]} \frac{\alpha_{ij}^k}{\alpha_{ij}(x^*)}$

broadcast β_i^k to master node

end for

$B_i^k = \frac{\beta_i^k}{m_i} \sum\limits_{j=1}^{m_i} \alpha_{ij}(x^*) x_{ij} x_{ij}^T$

$$x^{k+1} = x^k - \left[\frac{1}{n}\sum_{i=1}^{n} B_i^k\right]^{-1} \left[\frac{1}{n}\sum_{i=1}^{n} \nabla f_i(x^k)\right]$$

end for

Сходимость модифицированного базового алгоритма

Теорема 2.

При введенных допущениях на функции алгоритм SMCNM обладает локальной квадратичной сходимостью, при этом скорость сходимости задается формулой

$$\|x^{k+1} - x^*\| \leq \frac{C}{\mu}\|x^k - x^*\|^2, \tag{5}$$

где

$$C = H\sum_{j=1}^{n}\left(\frac{1}{\min\limits_{i}\{\alpha_{ij}(x^*)\|a_{ij}\|^2\}} + \frac{1}{2\alpha_{ij}(x^*)\|a_{ij}\|^2}\right). \tag{6}$$

Сравнение SMCNM и MCNM

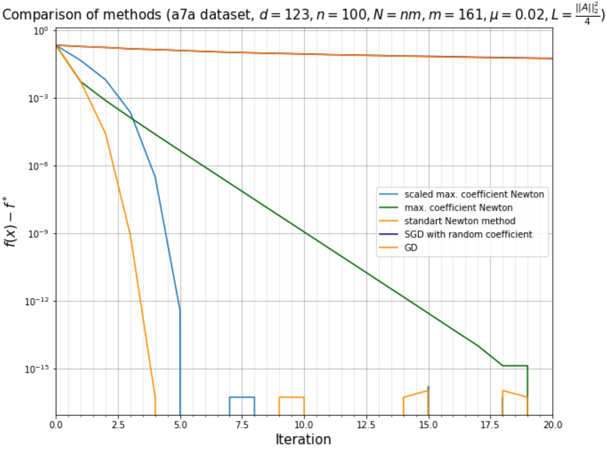

Рис.: Сравнение базового MNCM, Scaled MCNM и стандартного NM.

Дальнейшие исследования

- Убрать предположение знания w^* для SMCNM, построить последовательность γ_{ij}^k, которая будет сходится к $\alpha_{ij}(x^*)$ при $k \xrightarrow{\infty}$;
- Обобщить фреймворк на исходный случай: каждый worker имеет свой собственный датасэт.

Литература

1. A Distributed Second-Order Algorithm You Can Trust, C. Dünner, A. Lucchi, M. Gargiani, A. Bian, T. Hofmann, M. Jaggi, Proceedings of Machine Learning Research, 2018;
2. Quasi-Newton methods for deep learning: forget the past, just sample, A. S. Berahas, M. Jahani, P. Richtárik, and M Takáč, 2020;
3. Stochastic subspace cubic Newton method, F. Hanzely, N. Doikov, P. Richtárik and Yu. Nesterov, ICML 2020;
4. Tighter theory for local SGD on identical and heterogeneous data, A. Khaled, K. Mishchenko and P. Richtárik, AISTATS 2020;
5. Stochastic Newton and cubic Newton methods with simple local linear-quadratic rates, D. Kovalev, K. Mishchenko and P. Richtárik, NeurIPS 2019;
6. Unified analysis of stochastic gradient methods for composite convex and smooth optimization, A. Khaled, O. Sebbouh, N. Loizou, R. M. Gower, and P. Richtárik, 2020.