Домашнее задание №3. Проксимальные методы
SGD

Дедлайн: 9 декабря, 23:59

Просьба присылать задания в виде **одного PDF-файла** (можно использовать LaTeX, а можно просто аккуратно сфотографировать рукописные решения и **собрать фотографии в один PDF-файл**) на мою почту: mikkhailenko@gmail.com. Кроме того, просьба **указывать следующую тему письма: «Оптимизация в ML. Домашнее задание 3.»**

Во всех задачах предполагается, что в задаче

$$F(x) = f(x) + R(x) \to \min_{x \in \mathbb{R}^n} \tag{1}$$

функция $f(x)$ является μ-сильно выпуклой, $R(x)$ — правильная замкнутая выпуклая функция, x^* — точка минимума функции $F(x)$.

1. (1 балл) Докажите, что $x^* = \mathrm{prox}_{\gamma R}(x^* - \gamma \nabla f(x^*))$, где $\gamma > 0$.

2. (2 балла) Пусть $\mathbb{R}^n_{++} = \{x \in \mathbb{R}^n \mid x_i > 0,\ i = \overline{1, n}\}$ и

$$R(x) = \begin{cases} -\gamma \sum\limits_{i=1}^{n} \ln x_i, & x \in \mathbb{R}^n_{++}, \\ +\infty, & \text{иначе}, \end{cases} \qquad \text{где } \gamma > 0.$$

   Найдите $\mathrm{prox}_R(x)$.

3. (4 балла) Рассмотрим задачу (1), в которой

$$f(x) = \frac{1}{m} \sum_{i=1}^{m} f_i(x), \tag{2}$$

   где $f_i(x)$ — выпуклые и L-гладкие функции для всех $i = \overline{1, m}$. Пусть есть m компьютеров (их называют *рабочими*), причём все они соединены с одним и тем же сервером (который называют *мастером*), но между собой не соединены. Такая архитектура сети в оптимизации называется *параллельной*. Предположим, что мы хотим решать задачу (1) в такой архитектуре обычным проксимальным градиентным спуском с шагом γ. В таком случае, на каждой итерации i-й рабочий должен вычислить $\nabla f_i(x^k)$ (для всех $i = \overline{1, m}$), затем отправить $\nabla f_i(x^k)$ мастеру, который в свою очередь вычисляет среднее арифметическое от полученных векторов, то есть вычисляет $\nabla f(x^k)$, вычисляет $x^{k+1} = \mathrm{prox}_{\gamma R}(x^k - \gamma \nabla f(x^k))$ и отправляет x^{k+1} всем рабочим. В описанной процедуре большая нагрузка ложится на мастера, поскольку m и n могут быть большими и мастеру придётся принять (а рабочим передать) за одну итерацию очень большой объём информации (битов).

Разумная попытка по устранению этого недостатка состоит в том, чтобы вместо $\nabla f_i(x^k)$ отправлять несмещённую оценку g_i^k, которая будет иметь меньше битов информации, чем $\nabla f_i(x^k)$. Один из таких способов — это кватизация/спарсификация.

Определение 1 (Квантизация/Спарсификация)**.** Будем называть стохастический оператор $Q(x)$ оператором квантизации или просто квантизацией, если для любого $x \in \mathbb{R}^n$ выполняется:

$$\mathbb{E}[Q(x)] = x, \quad \mathbb{E}[\|Q(x) - x\|_2^2] \leq \omega \|x\|_2^2, \tag{3}$$

где $\omega \geq 0$.

Легко заметить, что тождественный оператор $Q(x) = x$ удовлетворяет Определению 1 с константой $\omega = 0$.

(a) Рассмотрим стохастический оператор

$$\mathrm{rand}_t(x) = \frac{n}{t} \sum_{i \in S} x_i e_i,$$

где t — некоторое **фиксированное** число из множества $\{1, \ldots, n\}$ (количество компонент вектора x, которые мы передаём; например, можно выбрать $t = 1$), S — случайное подмножество множества $\{1, \ldots, n\}$ размера t (подмножество S выбирается случайно и равновероятно среди всех возможных подмножеств размера t), $(e_1, \ldots, e_n)$ — стандартный базис в $\mathbb{R}^n$. Иными словами, среди всех компонент вектора x выбираются t компонент случайно и равновероятно, а полученный вектор домножается на $\frac{n}{t}$, чтобы добиться несмещённости. Покажите, что данный оператор удовлетворяет Определению 1 с константой $\omega = \frac{n}{t} - 1$. *Подсказка:* для этого воспользуйтесь $\mathbb{E}[\|Q(x)\|_2^2] = \mathbb{E}[\|Q(x) - x\|_2^2] + \|x\|_2^2$. Если возникают трудности, рассмотрите для начала случай $t = 1$.