Homework for "Algorithms For Big Data Analysis"

Zaiwen Wen
Beijing International Center for Mathematical Research
Peking University

February 17, 2024

1 Theoretical exercises

1. 令$f_i(x) \in \mathbb{R}^n \to \mathbb{R}, \forall i = 1, \ldots, N$, 是闭凸函数，存在次梯度。假设随机次梯度方差是一致有界，即存在M，对任意$x \in \mathbb{R}^n$以及随机下标s_k，有

$$E_{s_k}[\|g\|^2] \le M^2 < +\infty, \quad \forall g \in \partial f_{s_k}(x).$$

考虑求解优化问题$\min_{x\in\mathbb{R}^n} \ f(x) = \frac{1}{N}\sum_{i=1}^N f_i(x)$ 的随机梯度下降算法(SGD)：

$$x^{k+1} = x^k - \alpha_k g_k, \quad g_k \in \partial f_{s_k}(x^k),$$

其中s_k是从$\{1, \cdots, N\}$中随机等可能地抽取的一个样本，$\alpha_k > 0$为步长。

(a) 令x^*是优化问题的最优解。证明对所有的$K \ge 1$，下面不等式成立：

$$\sum_{k=1}^K \alpha_k E[f(x^k) - f(x^*)] \le \frac{1}{2}E[\|x^1 - x^*\|^2] + \frac{1}{2}\sum_{k=1}^K \alpha_k^2 M^2.$$

(b) 令$A_K = \sum_{i=1}^K \alpha_i$，定义$\bar{x}_K = \frac{1}{A_K}\sum_{k=1}^K \alpha_k x^k$，证明存在常数$D$，使得下面不等式成立：

$$E[f(\bar{x}_K) - f(x^*)] \le \frac{D + \sum_{k=1}^K \alpha_k^2 M^2}{2\sum_{k=1}^K \alpha_k}.$$

2 Coding exercises

2.1 Submission Requirement

1. Prepare a report including
   - detailed answers to each question
   - numerical results and their iterpretation
2. The programming language can be either matlab, Python or c/c++.

3. Pack all of your codes named as "sto-ID-name.zip" and upload the file to send it to TA: pkuopt@163.com

   作业提交需要统一打包成压缩文件，命名格式为：sto-学号-姓名，文件类型随意。文件名中不要出现空格，最好不要出现中文。

4. 请勿大量将代码粘在报告中，涉及到实际结果需要打表或者作图，不要截图或者直接从命令行拷贝结果。

5. 提交word 的同学需要提供word 原文件并将其转换成pdf 文件。

6. If you get significant help from others on one routine, write down the source of references at the beginning of this routine.

2.2 Variants of Stochastic Gradients Algorithms

Consider the nonconvex problem

$$\min_{w\in\mathbb{R}^d} \frac{1}{n}\sum_{i=1}^{n} f_i(w) + \lambda\|w\|_2^2, \tag{2.1}$$

where $f_i(w) = 1 - \tanh(y_i w^\top x_i)$, $\lambda > 0$ and (x_i, y_i) is the i-th data pair.

1. Write down and implement two of the following algorithms: Adagrad, adam, SVRG

2. You are encouraged to read the implementation in Pytorch, tensorflow as well as other packages. However, you should implement the codes by yourself.

3. Download the datasets covtype and gisette from
   `https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/binary.html`.

   If the testing set is not available, please split the data set into a training set and testing set randomly according to a ratio $7:3$ (Also check references on cross validation).

4. Test a few choices of λ (for example, $10, 1, 0.1, 0.001$. This value probably depends on the data sets). Generate figures on training error versus epoch, testing error versus epoch, training error versus time, testing error versus time, etc.

5. Extra-credit: propose, implement and test one of the following algorithms

   (a) stochastic quasi-Newton method

   (b) any other better idea