Homework 5 for "Algorithms For Big Data Analysis"

Zaiwen Wen
Beijing International Center for Mathematical Research
Peking University

April 26, 2017

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 "proj1-name-ID.zip" and send it to both me and TA:

   wendouble@gmail.com
   pkuopt@163.com
4. If you get significant help from others on one routine, write down the source of references at the beginning of this routine.

2 Variants of Stochastic Gradients Algorithms

Consider problem

$$\min_{w\in\mathbb{R}^d} \frac{1}{n}\sum_{i=1}^{n} f_i(w) + \lambda\|w\|_1, \tag{2.1}$$

where $f_i(w) = \log(1+\exp(-y^i w^\top x^i))$ and $\lambda > 0$.

1. Write down and implement two of the following algorithms: Adadelta, AdagradDA, Adagrad, ProximalAdagrad, Ftrl, Momentum, adam, Momentum, CenteredRMSProp, nesterov, rmsprop, SAG, SAGA, SVRG

   References: chapter 8 in: `http://www.deeplearningbook.org/`
2. You are encouraged to read the implementation in caffe, tensorflow as well as other packages. However, you should implement the codes by yourself. If you are going to test stochastic gradient methods in project 1, please choose different algorithms other than these in project 1.

3. Data sets: MNIST and Covertype. The set up is exactly the same as section 5 in the following paper, except that the ℓ_2-norm regularization term is replaced by ℓ_1-norm. Note that the MNIST Datset has been used for binary classfication of digits into even and odd.

   Exact and Inexact Subsampled Newton Methods for Optimization, Raghu Bollapragada, Richard Byrd, Jorge Nocedal, `https://arxiv.org/abs/1609.08502`

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 similar to Figure A.7 in the above paper.

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

   (a) stochastic gradient method using line search

   (b) stochastic gradient method using Barzilar-Borwein step sizes

   (c) any other better idea