Optimization Methods in ML Spring 19 - HW 2

Lecturer: Dan Garber

Due date: 23.6.2020

Guidelines:

- Please make an effort to type your answers on a computer
- you may submit in pairs
- you may consult with your fellow classmates ("high-level" discussions) - but you may not copy answers!
- programming is allowed in whatever environment you prefer
- submit to me via email with title: "Optimization Methods in ML - HW 2" (expect my acknowledgment) (include names and IDs in email body)
- feel free to contact me with questions

Question 1. *Recall that in the Conditional Gradient (CG) method, the next iterate at the end of time t is taken to be $\mathbf{x}_{t+1} = \mathbf{x}_t + \eta_t(\mathbf{y}_t - \mathbf{x}_t)$, where $\mathbf{y}_t$ minimizes the inner product with the gradient direction and $\eta \in (0, 1)$. Recall that we proved that CG with step-size $\eta_t = 2/(t + 1)$ converges with rate $O(\beta D^2/t)$.*
*However, in practice it is very much desired that η_t is not predetermined, but best suites the current step. That is, ideally on each time t we will set η_t via **line-search** using the rule:*

$$\eta_t \leftarrow \arg\min_{\eta\in[0,1]} f\left(\mathbf{x}_t + \eta(\mathbf{y}_t - \mathbf{x}_t)\right),$$

1. *Prove that CG with line-search also converges with rate $O(\beta D^2/t)$*
2. *Give a closed-form expression for η chosen via line-search (as above) for objective of the form: $f(\mathbf{x}) := \frac{1}{2}\|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2$. Here $\mathbf{x} \in \mathbb{R}^n$, $\mathbf{A} \in \mathbb{R}^{m\times n}$.*

Question 2. *Recall we have shown that the Conditional Gradient (CG) method, when initialized at an extreme point of the feasible set, always maintains a sparse solution, i.e., current iterate is given by a convex combination of a few extreme points of the feasible set (provided number of steps taken is not to large).*
Consider the case in which the feasible set is

$$\mathcal{K} = \{\mathbf{X} \in \mathbb{R}^{d\times d} \mid \mathbf{X} \succeq 0,\ \ Tr(\mathbf{X}) = 1\} \equiv \textit{convex-hull}\{\mathbf{v}\mathbf{v}^\top \mid \mathbf{v} \in \mathbb{R}^d,\ \|\mathbf{v}\| = 1\}.$$

This set, also known as the "spectrahedron" underlies many important matrix optimization problems.
Formally, for this setting, given a convex and β-smooth function, CG can guarantee to find a solution with ϵ accuracy and of rank $O(\beta/\epsilon)$.
Prove that when d is sufficiently large, this trade-off between accuracy and rank is optimal. i.e., there exists a function $f : \mathcal{K} \to \mathbb{R}$ such that any $\mathbf{X} \in \mathcal{K}$ which is an ϵ-approximated minimizer of f over $\mathcal{K}$ satisfies: rank($\mathbf{X}$) $= \Omega(\beta/\epsilon)$. Hint: revisit the lower bound for conditional gradient (see last slide at end of lecture 4 presentation on the website).

Question 3 (Stochastic Conditional-Gradient?). *Let $\mathcal{X} \subset \mathbb{R}^d$ be a convex and compact set. Let $f(\mathbf{x})$ be a convex and β-smooth function over $\mathcal{X}$. Suppose $f(\mathbf{x})$ is given by a stochastic first-order oracle, i.e., given some $\mathbf{x} \in \mathcal{X}$, the oracle returns a vector $\mathbf{g}$ such that $\mathbb{E}[\mathbf{g} \mid \mathbf{x}] = \nabla f(\mathbf{x})$. Suppose also that $\|g\|_2 \leq G$ and that $\mathbb{E}[\|\mathbf{g} - \nabla f(\mathbf{x})\|_2^2 \mid \mathbf{x}] \leq \sigma^2$, for some $G > 0, \sigma > 0$.*

Consider now the following stochastic variant of the Conditional-Gradient method: replacing the step $\mathbf{y}_t \in \arg\min_{\mathbf{y} \in \mathcal{X}} \nabla f(\mathbf{x}_t)^\top \mathbf{y}$ with the new step $\mathbf{y}_t \in \arg\min_{\mathbf{y} \in \mathcal{X}} \widehat{\nabla} f(\mathbf{x}_t)^\top \mathbf{y}$, where $\widehat{\nabla} f(\mathbf{x}_t) := \frac{1}{k}\sum_{i=1}^k \mathbf{g}_i$, where $\mathbf{g}_1, \ldots, \mathbf{g}_k$ are produced by k different calls to the first-order stochastic oracle with the same point $\mathbf{x}_t$. k is a parameter to be tuned.

- *Observe that for $k = 1$ (i.e., using one sample per iteration) the above algorithm cannot be proven to converge (you don't need to prove this formally, just observe that the analysis fails).*
- *Suppose we want to find a point $\mathbf{x} \in \mathcal{X}$ such that $f(\mathbf{x}) - f(\mathbf{x}^*) \leq \epsilon$ for some given $\epsilon > 0$. Give a bound for k (the mini-batch size), note this bound can depend on ϵ and all other parameters, such that this modified algorithm converges in expectation to an ϵ-approximated solution. Find the value of k that minimizes the overall sample complexity (i.e., total number of calls to the stochastic first-order oracle) to reach ϵ-error in expectation.*

  Hint: the analysis of SGD with smoothness might be helpful here.

Question 4 (SVRG for sum of non-convex functions). *Recall that the SVRG method (Lecture 8) deals with the following problem: $\min_{\mathbf{x} \in \mathbb{R}^d}\{F(\mathbf{x}) := \frac{1}{n}\sum_{i=1}^n f_i(\mathbf{x})\}$, where each f_i is convex and β-smooth, and $F(\mathbf{x})$ is α-strongly convex. Recall that there exists a choice of step-size η such that for all $\epsilon > 0$, the SVRG method finds a point $\mathbf{y}$ such that $F(\mathbf{y}) - F(\mathbf{x}^*) \leq \epsilon$, after processing $O\left(\left(\frac{\beta}{\alpha} + n\right)\log(1/\epsilon)\right)$ gradients of any of the individual functions $f_1, \ldots, f_n$.*

However, in certain problems of interest, while $F(\mathbf{x})$ is convex, each individual function $f_i(\mathbf{x})$ need not be convex. Prove that in this setting, there exists a choice of step-size η such that the SVRG method find an ϵ-approximated solution (as above) after processing $O\left(\left(\left(\frac{\beta}{\alpha}\right)^2 + n\right)\log(1/\epsilon)\right)$ gradients of any of the individual functions $f_1, \ldots, f_n$.

Question 5 (programming question 2). *Repeat programming question from previous HW (only in the strongly convex case). This time include SGD, Mini-batch SGD (try a few options for size of mini batch), and SVRG. Plot the function value vs.*

number of effective passes (e.g.,, for $\mathbf{A} \in \mathbb{R}^{m\times d}$*, let SGD do* m *stochastic updates for each point in the graph, and for mini-batch SGD with batch-size* N*, do* m/N *updates). Submit same things as in the previous HW.*