CS 798: Convexity and Optimization **Last Updated:** Mar 18, 2017
Homework 2 **Due Date: Mar 24, 2017 (in class)**

You are allowed to discuss with others but not allowed to use any references other than the course notes and the four primary references listed in the course page. If you discuss with others, please list your collaborators for each question. In any case, you must write your own solutions.

The full mark is 60. This homework is counted 15% of the course.

---

1. **Multiplicative Update**

   (20 marks) We analyze the multiplicative weights update method in solving special linear programs. The problem is $\min \sum_{i=1}^{n} x_i$ subject to $Ax \geq \vec{1}$ and $x \geq \vec{0}$, where A is an $m \times n$ matrix with every entry of A being either zero or one. Design an algorithm to solving this linear program using the multiplicative weights update method (including the design of the oracle), and analyze its total running time.

2. **Gradient Descent for Linear Programming**

   (20 marks) Your friend came up with an interesting idea to solve linear programs of the form $Ax \geq b$. Let A be an $m \times n$ matrix with its i-th row denoted by a_i. We assume that each entry of A and b has absolute value at most c. First, the problem is reformulated as an unconstrained problem

   $$\min_{x \in \mathbb{R}^n} \max_{1 \leq i \leq m} b_i - a_i x,$$

   so that the original problem is feasible if and only if the optimal value of the reformulated problem is non-positive. Then, the objective function is smoothened by using the soft-max function

   $$\min_{x \in \mathbb{R}^n} \frac{1}{K} \log \sum_{i=1}^{m} \exp(K(b_i - a_i x))$$

   so that gradient descent algorithms can be applied, where K is a parameter that we can tune for the accuracy of the approximation to the original problem. So far so good, but your friend does not know what is the best way to solve this problem using gradient descent algorithms. As a CS 798 student, your task is to come up with the fastest algorithm (that you can) to solve this problem and provide an analysis of its running time. More precisely, if the linear program is indeed feasible, provide an analysis of the time complexity of your algorithm to return a solution x such that $Ax \geq b - \epsilon \vec{1}$ where ϵ is the given accuracy requirement. You can use all the results in the lecture notes and the primary references without providing proofs. You will get some bonus marks if your solution surprises your TA.

3. **Black Box Reduction**

(20 marks) Suppose your friend gave you an "accelerated" algorithm $\mathcal{A}$ for minimizing a strongly convex function with the following guarantee: If f is α-strongly convex and β-smooth with x^* as its unique minimizer and x_0 an initial point, then $\mathcal{A}$ returns a point x' satisfying $f(x') - f(x^*) \leq \epsilon$ in time

$$O\left(\sqrt{\frac{\beta}{\alpha}} \log\left(\frac{f(x_0) - f(x^*)}{\epsilon}\right)\right).$$

You would like to use algorithm $\mathcal{A}$, but the function is not strongly convex in your problem, and it is not clear how to change the algorithm and the analysis for your purpose. Prove that you can still use algorithm $\mathcal{A}$ as a black box to give an "accelerated" algorithm with the following guarantee: If f is β-smooth and x^* is a minimizer of f and x_0 is an initial point, then your accelerated algorithm will return a point x' satisfying $f(x') - f(x^*) \leq \epsilon$ in time

$$O\left(\sqrt{\frac{\beta}{\epsilon}} \|x_0 - x^*\|_2 \log\left(\frac{f(x_0) - f(x^*)}{\epsilon}\right)\right).$$

You can assume (unrealistically) that you know the values $\|x_0 - x^*\|_2$ and $f(x_0) - f(x^*)$ in advance.

Bonus: You will get up to 20 bonus marks if you can improve the above running time in a black box manner.