Homework #9
CS675 Fall 2023

Due Wednesday Nov 22, by midnight

General Instructions The following assignment is meant to be challenging. Feel free to discuss with fellow students, though please write up your solutions independently and acknowledge everyone you discussed the homework with on your writeup. I also expect that you will not attempt to consult outside sources, on the Internet or otherwise, for solutions to any of these homework problems — doing so would be considered cheating.

Several of these problems are drawn from the following texts, each of which is linked on the course website: Luenberger and Ye (4th edition), Korte and Vygen (5th edition), and Boyd and Vendenberghe. Please make sure you are using the correct edition of each of the books by using the links on the course website.

We request that you submit your homework as a pdf file, by email to the TA.

Finally, whenever a question asks you to "show" or "prove" a claim, please provide a formal mathematical proof.

Problem 1. (15 points)
In the lecture devoted to submodular function maximization subject to a matroid constraint, your instructor neglected the implementation details of the Pipage rounding procedure. At issue were steps (1) and (3). Step (1) requires computing the minimum-size *tight set* of a matroid $\mathcal{M} = (\mathcal{X}, \mathcal{I})$ with respect to a given point $x \in \mathcal{P}(\mathcal{M})$ subject to including a fractional element. Step (3) requires optimally trading off x_i and x_j while remaining in the matroid polytope. In this problem, you will show how to fill in these blanks.

(a) [2 points]. Let x be in the matroid base polytope $\mathcal{P}_{base}(\mathcal{M})$, and recall that $\mathcal{P}_{base}(\mathcal{M}) \subseteq \mathcal{P}(\mathcal{M})$. Show that the family of tight sets $\{S \subseteq \mathcal{X} : x(S) = rank_{\mathcal{M}}(S)\}$ forms a lattice. Conclude that, for each $i \in \mathcal{X}$, there is a tight set T_i with $i \in T_i$ such that every tight set containing i is a superset of T_i; we say T_i is the *minimum tight set* containing i.

(b) [8 points]. Show how to efficiently implement step (1) of the Pipage rounding procedure. Your algorithm should compute T in time polynomial in $|\mathcal{X}|$ and the bit complexity $\langle x \rangle$ of the solution x. You should assume that you have access to an independence oracle for the matroid $\mathcal{M}$. (**Hint**: Reduce this to computing T_i for each $i \in \mathcal{X}$. To compute T_i, use a greedy algorithm which starts with $R = \mathcal{X}$, and iteratively removes elements from R until it ends up with T_i. Your algorithm will use submodular minimization as a subroutine.)

(c) [5 points]. Show how to efficiently implement step (3) of the Pipage rounding procedure. Given

$x \in \mathcal{P}(\mathcal{M})$ and $i, j \in \mathcal{X}$, your algorithm should compute $\mu \in [0, 1]$ maximizing $F(x + \mu(e_i - e_j))$, where F is the multilinear extension of a submodular function, in time polynomial in $|\mathcal{X}|$ and $\langle x \rangle$. As usual, you may assume access to an independence oracle for $\mathcal{M}$. Moreover, for purposes of this problem you may assume that you can exactly evaluate F in constant time.

Problem 2. (10 points)
The following is a (simplification of) a task encountered in speech recognition. Here, we wish to select a small vocabulary (a set of words) which express many common phrases. Specifically, you are given a set P of phrases which, collectively, use only the words in the set W. Moreover, each phrase $p \in P$ is given a nonnegative weight $w(p)$ indicating its importance. For a vocabulary $V \subseteq W$, we use $P(V) \subseteq P$ to denote the phrases which use only words in V. Your goal is to choose a vocabulary optimizing a tradeoff between the total weight of expressed phrases and the size of the vocabulary; specifically, find $V \subseteq W$ maximizing $\left(\sum_{p \in P(V)} w(p)\right) - |V|$. Show that this problem is solvable in poly$(|P|, |W|)$ time. You may assume that you can test whether a particular word occurs in a particular phrase in constant time.

Problem 3. (15 points)
We showed in class that there is a polynomial-time algorithm for submodular minimization (and equivalently, supermodular maximization) when there are no constraints. In this problem, we will see how hardness kicks in as soon as even simple constraints are introduced.

(a) [6 points]. Show that it is NP-hard to minimize a submodular function subject to a cardinality lower-bound. Specifically, given an integer k and a submodular function $f : 2^{\mathcal{X}} \to \mathbb{R}$, your task is to minimize $f(S)$ over sets $S \subseteq \mathcal{X}$ with $|S| \geq k$. You may reduce from the NP-complete problem CLIQUE: given a graph G and an integer k, determine if G contains a clique with k nodes.

(b) [3 points]. Show that part (a) implies NP-hardness for minimizing a submodular function subject to a cardinality upper-bound: i.e., over sets S with $|S| \leq k$ for a given integer k. Note that we can now also conclude that supermodular maximization is NP-hard subject to a cardinality upperbound or cardinality lowerbound.

(c) [6 points]. We will now show that the problem is hard to approximate. In the *densest k-subgraph* problem, we are given an undirected graph G and an integer k, and we must find a k-node subgraph of G with the maximum number of edges. The densest k-subgraph problem is known to not admit a constant approximation algorithm if we assume the *exponential time hypothesis (ETH)*, which is a well-believed complexity-theoretic assumption stronger than $P \neq NP$. Show by a reduction from the densest k-subgraph problem that the problem of maximizing a supermodular function subject to a cardinality upperbound does not admit a constant approximation algorithm assuming ETH.