Approximation Algorithms and Inapproximability Spring 2024: Assignment 2 problems

Due: August 13th

Preamble.

- You are graded on both your accuracy and your presentation. A correct solution that is poorly presented may not receive full marks. You are expected to provide full proofs of all claims.
- Reproduction, sharing or online posting of this document is strictly forbidden.

A9-1. **PCP and Projection Label Cover**

Definition 1. *A instance of PCP consists of sets of nodes $X = \{x_1, ..., x_n\}$, an alphabet Σ_i for each node x_i, a collection of hyperedges E, and a constraint C_e for each hyperedge $e \in E$. For each hyperedge $e \in E$, writing $e = (x_{i_1}, \ldots, x_{i_s})$, the set of accepting strings for the constraint C_e can be any subset of $\Sigma_{i_1} \times \cdots \times \Sigma_{i_s}$. The alphabet size of an instance is $max_i|\Sigma_i|$, and the number of queries is the size of the largest hyperedge in the graph.*

Given an instance $\Psi = (X, \Sigma, E, (C_e)_{e\in E})$ of PCP (where $\Sigma = \Sigma_1 \times \cdots \times \Sigma_n$), we define its value, denoted $\mathrm{Val}(\Psi)$ as the maximum fraction of satisfied constraints for any assignment $X \in \Sigma$.

We use $\mathrm{PCP}[c, s]$ to denote the distinguishing task of determining whether an instance Ψ has value at least c or at most s.

In lecture we covered the ideas in the proof of the following result:

Theorem 1. *There are absolute constants $C, c > 0$ such that $\mathrm{PCP}[1, 1/\log^c n]$ with $\log^C n$ queries and $\log^C n$ alphabet size is NP-Hard.*

Use the above result to show 1 vs. $1 - 1/\log^{O(1)} n$ NP-Hardness of projection label cover, with alphabet size $\log^{O(1)} n$. (Note: this is a reduction from polylogarithmic queries to two queries!)

Hint: Define a bipartite graph with constraints C_e being the left vertices (of larger alphabet) and variables x_i representing the right vertices.

A9-2. **Composing Outer PCP with Inner PCP**

Recall in the first half of the course we gave a $\pi/2$-approximation algorithm for a generalization of Max-Cut, namely PSD quadratic maximization over the hypercube: $\max_{x \in \{\pm 1\}^n} \langle x, Ax \rangle$ where $A \succeq 0$. Mimic the max-cut reduction and prove that for any $\epsilon > 0$, it is UG-Hard to approximate PSD quadratic maximization over the hypercube better than $\pi/2 + \epsilon$.

To analyze your reduction, use the following dictatorship testing claim:

Theorem 2. *Let $f : \{\pm 1\}^k \to \{\pm 1\}$ satisfy $|\widehat{f_i}| \leq \epsilon$ for all $i \in [k]$, i.e. all linear (degree-1) fourier coefficients of f are bounded in magnitude.*

Then $\sum_{i \in [k]} \widehat{f_i}^2 \leq 2/\pi + C\epsilon$, where C is a universal constant.

Hint: Note that if P_1 denotes the projector of a boolean function to the span of the degree-1 fourier characters, we have $\langle f, P_1 f \rangle = \sum_{i \in [k]} \widehat{f_i}^2$.