Approximation Algorithms and Inapproximability Spring 2024: Assignment 1 problems

Due: August 8th

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. **Positive Semidefinite Matrices**

Consider any nonnegative scalars $f_1, \dots f_k \in \mathbb{R}$ and let $X \in \mathbb{R}^{n \times n}$ be a symmetric PSD matrix. Let $f(t) := \sum_k f_k t^k$. Prove that the following two matrices are PSD:

(a) $f[X] \in \mathbb{R}^{n \times n}$ which denotes entry-wise application of f to X.

(b) $\sum_{i \in [k]} f_k X^k$ where X^k denotes the usual matrix power.

A9-2. **Max-Cut and SDP**

Give a randomized approximation algorithm that given a graph G on m edges such that there is a cut of size at least $(1-\varepsilon)m$, the algorithm returns a cut of expected size at least $(1 - C\sqrt{\varepsilon})m$ (where C is a universal constant).

A9-3. **SDP Rounding II**

Consider the family of graphs $G = (V, E)$ on n vertices, of degree d that moreover admit the following embedding in space: there exist unit vectors $v_1, \dots v_n \in \mathbb{S}^{n-1}$ such that for any $(i, j) \in E$, we have $\langle v_i, v_j \rangle \leq -1/2$.

Give a randomized algorithm (and its analysis) that given the above vectors, produces an independent set in G of expected size $\Omega(n/\sqrt{d})$.

Define for $z \geq 0$, $\text{tail}(z) := \frac{1}{\sqrt{2\pi}} \int_z^\infty e^{-t^2/2} dt$, which also equals the probability that a standard gaussian is at least z. Use the following inequality in your analysis.

$$\forall t \geq 0, \quad e^{-t^2/2}(1/t - 1/t^3) \leq \sqrt{2\pi} \cdot \text{tail}(t) \leq e^{-t^2/2}/t.$$

Hint: Sample a random ice-cream cone of judiciously chosen aperture, and consider all vertices that are contained in it. Throw out any vertex for which at least one of its neighbors also happens to lie inside the cone. Analyze the expected number of vertices that fall in the cone and the expected number of vertices thrown out.