Hardness of Approximation - Problem Set 1

Due: 25 April 2019

1. In **3-coloring** problem, given an undirected graph the task is to color its vertices with three colors such that no adjacent vertices have the same color.

   In **Max-3-coloring**, the task is to color the vertices of a graph with 3 colors such that the fraction of edges which are properly colored (i.e. fraction of edges such that its endpoints get different colors) is maximized.

   Show that there exists $s < 1$, such that given a graph which is 3-colorable, it is NP-hard to find a 3-coloring which properly colors more s fraction of the edges. In other words, gap-Max-3-coloring$(1, s)$ is NP-hard for some constant $0 < s < 1$.

   Hint: Start with the NP-hardness of gap-3SAT$(1, s)$ *and recall the NP-completeness reduction from 3SAT to 3-coloring*

2. In the first lecture, we defined g-approximation for maximization problems. We can similarly define g-approximation for any minimization problems.

   g-approximation: A minimization problem P is said to have a g-approximation algorithm if there exists a polynomial time algorithm such that given an instance x of P, it always returns a solution with value at most $g\cdot$ OPT, where OPT is the optimal value of x. Note that in this case, g must be at least 1.

   - **Vertex Cover:** Given a graph $G(V, E)$, find a smallest subset $S \subseteq V$ such that for any edge $(u, v) \in E$, either $u \in S$ or $v \in S$.
   - **Independent set:** Given a graph $G(V, E)$, find a largest subset $I \subseteq V$ such that no edge is fully contained in I.

   A simple observation: if $S^\star$ is the *minimum* vertex cover in G, then $V \setminus S^\star$ is the *maximum* independent set in G.

   Suppose you have given a 2-approximation algorithm for Vertex Cover. Consider the following approximation algorithm for Independent Set problem: Given a graph $G(V, E)$, use the 2-approximation algorithm for Vertex Cover on G to get a vertex cover S and output $V \setminus S$.

   How good an approximation algorithm is this for the Independent Set problem?

3. In the first lecture, we defined a problem called *Label Cover* and showed a reduction from gap-3SAT$(1, s)$ to gap-LC$(1, s')$ for some $0 < s, s' < 1$.

   A 3SAT instance is a CSP over Boolean variables $x_1, x_2, \ldots, x_n$ and consists of collection of constraints $C_1, C_2, \ldots, C_m$ where each C_i is the OR of 3 literals eg. $C_i = x_3 \vee \neg x_2 \vee x_7$. Recall how we constructed a Label Cover instance from a 3SAT instance: One side contains a vertex for every clause, the other side contains a vertex for every variable. A clause vertex is connected to a variable vertex if and only if the variable is contained in the clause. A set of labels for any variable vertex is $\{0, 1\}$, whereas a set of labels for any clause vertex is all strings $\{0, 1\}^3$ which satisfies the clause. The projection constraint between the clause vertex and a variable vertex is just the consistency constraint.

   Show that the reduction has the following completeness and the soundness guarantees:

   - **Completeness:** If the 3SAT instance is satisfiable then the Label Cover instance is also satisfiable.
   - **Soundness:** If the value of 3SAT instance is $s < 1$, then the value of the Label Cover instance is s', for some constant $s' < 1$.

4. **Parallel Repetition and Label Cover:** We showed above that gap-LC$(1, s')$ is NP-hard for some $0 < s' < 1$.

   Consider a following way of *amplifying* the gap i.e showing gap-LC$(1, s)$ is NP-hard for small s : Fix $t \geq 1$. Start with a Label Cover instance instance $\mathcal{G}_1(L_1, R_1)$ and construct a Label Cover instance $\mathcal{G}_2(L_2, R_2)$ as follows:

   On the left side of $\mathcal{G}_2$, we have a vertex for every ordered tuple of t vertices from L_1. On the right, we have a vertex for every ordered tuple of t vertices from R_1 i.e. $L_2 = \underbrace{L_1 \times L_1 \times \ldots, \times L_1}_{t \text{ times}}$ and $R_2 = \underbrace{R_1 \times R_1 \times \ldots, \times R_1}_{t \text{ times}}$.

   A vertex $(u_1, u_2, \ldots, u_t) \in L_2$ from the left is connected to a vertex $(v_1, v_2, \ldots, v_t) \in R_2$ from the right if (u_i, v_i) are connected in $\mathcal{G}_1$, for every $1 \leq i \leq t$. A valid label set to a vertex in $(x_1, x_2, \ldots, x_t) \in L_2 \cup R_2$ is just a tuple of t label set such that the i^{th} label is a valid label to x_i. A label $(\alpha_1, \alpha_2, \ldots, \alpha_t)$ to $(u_1, u_2, \ldots, u_t) \in L_2$ and and a label $(\beta_1, \beta_2, \ldots, \beta_t)$ to $(v_1, v_2, \ldots, v_t) \in R_2$ satisfy the edge $((u_1, u_2, \ldots, u_t), (v_1, v_2, \ldots, v_t))$ if and only if (α_i, β_i) satisfy the projection constraint between (u_i, v_i) from $\mathcal{G}_1$. Ideally, we would like the following completeness and soundness guarantees:

   - **Completeness:** If the $\mathcal{G}_1$ is satisfiable then $\mathcal{G}_2$ is also satisfiable.
   - **Soundness:** If the value of $\mathcal{G}_1$ is $s' < 1$, then the value of $\mathcal{G}_2$ is s, where $s \to 0$ as t increases.

   (a) Show that the completeness guarantee holds.

(b) An assignment to the Label Cover instance $\mathcal{G}_2$ is called a **direct product** assignment if it is coming from a fixed assignment to the vertices of $\mathcal{G}_1$ i.e. fix an assignment σ to $L_1 \cup R_1$ and this naturally gives an assignment to $L_2 \cup R_2$, where a vertex $(x_1, x_2, \ldots, x_t) \in L_2 \cup R_2$ gets a labeling $(\sigma(x_1), \sigma(x_2), \ldots, \sigma(x_t))$. Show that for any assignment to the Label Cover instance $\mathcal{G}_2$ which is a direct product assignment, the value goes to 0 as t increases, if the value of $\mathcal{G}_1$ is at most s' for some constant $s' < 1$.