CSCI3160: Regular Exercise Set 12

Prepared by Yufei Tao

Problem 1. Consider the set cover algorithm discussed in the lecture. Prove: it achieves an approximation ratio of $h = \max_{S \in \mathcal{S}} |S|$, where $\mathcal{S}$ is the input family of sets.

Remark: This means if all the sets in $\mathcal{S}$ have constant sizes, then the approximation ratio is $O(1)$.

Solution. Suppose that our algorithm picks t sets. Every time the algorithm picks a set, at least one new element is covered. For each $i \in [1, t]$, denote by e_i an arbitrary element that is *newly* covered when the i-th set is picked.

Let $\mathcal{C}^*$ be an optimal universe cover. Because each e_i ($i \in [1, t]$) exists in at least one set of $\mathcal{C}^*$, we have:

$$t = \sum_{i=1}^{t} 1 \le \sum_{i=1}^{t} |\{S \in \mathcal{C}^* : e_i \in S\}| \le \sum_{S \in \mathcal{C}^*} |S|.$$

The right hand side of the above is bounded by $|\mathcal{C}^*| \cdot h$. This completes the proof.

Remark: Our algorithm actually enjoys an approximation ratio of $1 + \ln h$, but the proof is much more sophisticated.

Problem 2. Let $\mathcal{C}^*$ be an optimal universe cover for the set cover problem. Consider running the set cover algorithm discussed in the lecture. In particular, consider the moment right before the algorithm is to choose the i-th set S_i, having chosen already $S_1, S_2, ..., S_{i-1}$. Let z_{i-1} be the number of elements in the universe that have not been covered by $S_1 \cup S_2 \cup ... \cup S_{i-1}$. Let $s = |\{S_1, S_2, ..., S_{i-1}\} \cap \mathcal{C}^*|$, i.e., s of the $i-1$ sets chosen by the algorithm are from $\mathcal{C}^*$. Prove: S_i has benefit at least $z_{i-1}/(|\mathcal{C}^*| - s)$ (namely, the i-th set picked by the algorithm covers at least $z_{i-1}/(|\mathcal{C}^*| - s)$ new elements).

Solution. Let $\mathcal{C}^*_{out} = \mathcal{C}^* \setminus \{S_1, S_2, ..., S_{i-1}\}$; note that $|\mathcal{C}^*_{out}| = |\mathcal{C}^*| - s$. The z_{i-1} elements not yet covered by $\{S_1, S_2, ..., S_{i-1}\}$ must be covered by $\mathcal{C}^*_{out}$. Therefore, at least one of the sets in $\mathcal{C}^*_{out}$ has benefit at least $z_{i-1}/|\mathcal{C}^*_{out}| = z_{i-1}/(|\mathcal{C}^*| - s)$. The claim thus follows from the algorithm's greedy nature.

Problem 3. Let R be a set of n red points in 2D space, and B be a set of n black points in 2D space. Fix an integer $\epsilon > 0$. A subset $S \subseteq R$ is a *B-guarding set* if, for every black point $b \in B$, there is at least one point $r \in S$ with $dist(r, b) \le \epsilon$, where $dist(r, b)$ is the Euclidean distance between r and b. Let OPT be the smallest size of all B-guarding sets. Design a poly(n)-time (i.e., polynomial in n) algorithm that returns a B-guarding set with size OPT $\cdot\, O(\log n)$; if no B-guarding sets exist, your algorithm must correctly declare so.

Solution. We will convert the problem to set cover. For each red point $r \in R$, obtain the set $B(r)$ of black points p satisfying $dist(r, p) \le \epsilon$. This can be easily implemented in poly(n) time. If $\bigcup_{r \in R} B(r) \ne B$, we declare that no B-guarding sets exist. Otherwise, run our greedy set-cover algorithm on $\{B(r) \mid r \in R\}$.

Problem 4. Let S be a set of n axis-parallel rectangles in 2D space (i.e., each rectangle has the form $[x_1, x_2] \times [y_1, y_2]$; you can assume that the x_1, x_2, y_1, y_2 values of the n rectangles are all distinct). A set P of points is an S-*pinning set* if every rectangle of S covers at least one point in P. Let OPT be the smallest size of all S-pinning sets. Design a poly(n)-time algorithm that returns an S-pinning set with size $\text{OPT} \cdot O(\log n)$.

Solution. We say that a point p is *important* if

- it is a corner of some rectangle in S, or
- it lies on the boundary of at least two rectangles in S.

If P is an S-pinning set, we can always find an S-pinning set of the same size that contains *only* important points (if a point in P is not important, keep pushing it towards a corner or the intersection of two edges). Equipped with this observation, we can convert the problem to hitting set.

First, obtain the set I of all the important points. The size of I is $O(n^2)$ and can be easily computed in poly(n) time. For each point $p \in I$, define $S(p)$ to be the set of rectangles in S that contain p. Run our greedy set-cover algorithm on $\{S(p) \mid p \in I\}$.

Problem 5 (Conversion from Set Cover to Hitting Set). Suppose that we have an algorithm $\mathcal{A}$ for the hitting set problem that achieves an approximation ratio ρ. Use $\mathcal{A}$ to design a ρ-approximate algorithm for the set cover problem.

Solution. Let $(U_{sc}, \mathcal{S}_{sc})$ be an input to the set cover problem. W.l.o.g., assume $\mathcal{S}_{sc} = \{S_1, S_2, ..., S_t\}$ for some integer $t \geq 1$. We create an input $(U_{hs}, \mathcal{S}_{hs})$ to the hitting set problem as follows.

- $U_{hs} = \{1, 2, ..., t\}$.
- For each $e \in U_{sc}$, define $OriginS_e = \{i \in [1, t] \mid e \in S_i\}$.
- $\mathcal{S}_{hs} = \{OriginS_e \mid e \in U_{sc}\}$.

Now, run $\mathcal{A}$ on $(U_{hs}, \mathcal{S}_{hs})$ and let H be its output. Create a universe cover $\mathcal{C}$ for $(U_{sc}, \mathcal{S}_{sc})$ as follows:

$$\mathcal{C} = \{S_i \mid i \in H\}.$$

By an argument similar to what was discussed in the lecture, we can prove that $\mathcal{C}$ is a universe cover whose size is at most ρ times the optimal size.