Lecture Notes CS:5350 Introduction to Greedy approximation Algorithms
Lecture 6: Nov. 7, 2019
Scribe: Qi Qi

1 Minimum Vertex Cover

Greedy algorithm for MVC (Here is a **greedy (deterministic)** algorithm for MVC):

1. $S \leftarrow \emptyset$
2. while (S is not a vertex cover) do:
   // greedy step
3. pick a vertex with highest degree, v, in active graph and add to S
4. deactivate all activate edges incident on v
5. Output S

1.1 Counter Examples:

Figure 1 shows a simple counter example to show that the greedy algorithm will not always produce an optimal solution for MVC.

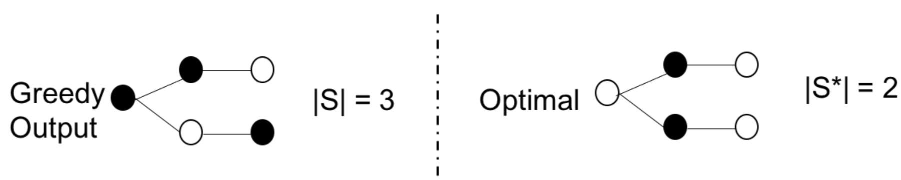

Figure 1: Simple counterexamp graph

In fact, the output of the greedy algorithm can be quite poor, as showing in the following claim.
Claim: There exists a graph $G = (V, E), |V| = n$ such that $|S| = \Omega(\log(n)) \cdot |S^*|$, where S is the output of greedy algorithms, and S^* is the optimal vertex cover.
Proof: Construct a bipartite graph (Figure 2), which contains $G = (L \cup R, E)$. Let k denote $|L|$, where $R = R_1 \cup R_2 \cup \cdots R_k$. For each set R_i:

1. $|R_i| = \lfloor \frac{k}{i} \rfloor$
2. Each vertex in R_i has degree i and no two vertexes in R_i have a common neighbour.

The vertex in R_k would be the first node to to be selected to join S by the greedy MVC algorithm since its degree is k. All other nodes in R have degree less than k and nodes in L also have degree less than k (the largest degree in the L is $k-1$). After deleting all the edges that connected with this node R_k, all the degree of nodes in the L will also decrease 1. Then in the next iterations, the nodes in R_{k-1} will be selected one by one, and on and on until all the nodes in R are selected to join S.

Then the final output $|S| = \sum_{i=2}^{k} |R_i| = \lfloor \frac{k}{i} \rfloor | \sim k \sum_{i=1}^{k} \frac{1}{i} = \Theta(k \log(k))$, but $|L| = k$. $\square$.

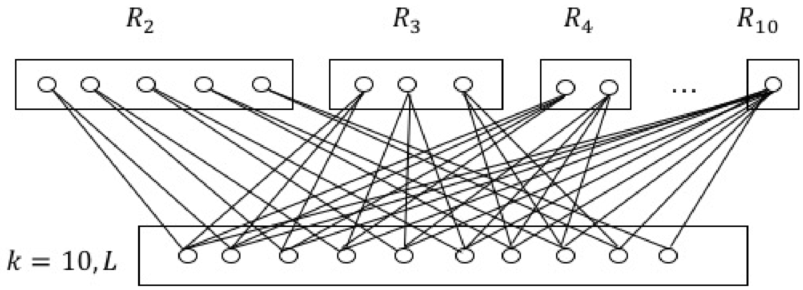

Figure 2: Constructed counterexample of bipartite graph

2 Landscape of problems approximation factors

Category of Approximation Factor	Best known Approximation Factor	Problem	Notes
PTAS	$(1+\epsilon)$, for any $\epsilon > 0$	Knapsack	Using data rounding and dynamic programming, there is an $O(\frac{n^3}{\epsilon})$ time complexity algorithm
constant	2	K-center	α-approximation, for $\alpha < 2$ is not possible unless P = NP
	2	MVC	Whether there is a better than 2-approximation is a long-standing open problem
logarithmic	$ln(n)$ n =size of ground set	SET COVER	Better approximation is not possible unless all problems in NP can be solved in sub-exponential time
	$ln(n)$ n =size of vertexes	Min. Dominating Set (MDS)	MDS is just a special case of SET COVER
Polynomial	$O(\frac{n}{poly(\log n)})$	Maximum Independent Set	No $O(n^{1-\epsilon})$-approximation exists unless P=NP

Table 1: Landscape of problems approximation factor

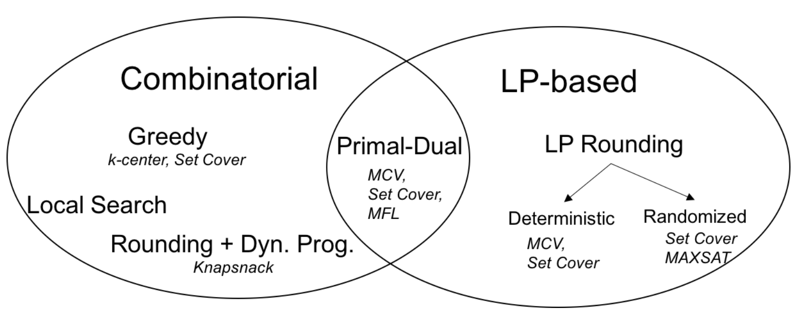

Figure 3: Landscape of techniques

Remark: Figure 3 illustrates the landscapes of techniques for solving the problems in Table 1.

3 K-Center

K-Center: well known clustering problems with many applications (e.g. unsupervised learning.)

Definition (Distance Metric): Let $P = \{p_1, p_2, \cdots, p_n\}$ be a set of points. Let $D : P \times P \to \mathbb{R}_{\geq 0}$ be a function. D is a called metric if:

1. $D(p_i, p_i) = 0$ for all $p_i \in P$ (reflexive)
2. $D(p_i, p_j) = D(p_j, p_i)$ for all $p_i, p = j \in P$ (symmetric)
3. $D(p_i, p_j) + D(p_j, p_k) \geq D(p_i, p_k)$ for all $p_i, p_j, p_k \in P$ (triangle inequality)

Examples (Aside from Euclidean Distance):

1. Let $G = (V, E)$ be an undirected graph. for any $u, v \in V$, $D(u, v)$ denotes the shortest path distance between (u, v). It is easy to check that D is a metric.
2. Let $G = (V, E)$ be an undirected graph. For any $u, v \in V$, let $D(u, v) = 1$, if $\{u, v\} \in E$ and $D(u, v) = 2$ if $\{u, v\} \notin E$. Also $D(v, v) = 0$ for all $v \in V$. D is a metric as $D(u, v) + D(v, k) \geq D(u, k)$.

Notation: Let $S \subseteq P$, $D(p, S) = \min_{s \in S} D(p, s)$:

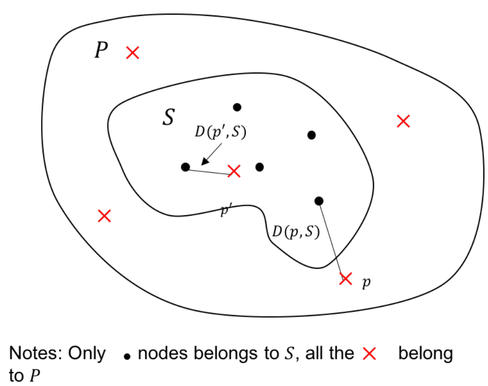

Figure 4: Illustrates of the distance between a point to a set i.e., $D(p, S)$.

K-Center:

1. **Input**: A metric $D : P \times P \to \mathbb{R}_{\geq 0}$.
2. **Output**: A subset $S \subseteq P, |S| = k$ such that $\max_{p \in P} D(p, S)$ is minimized.

Alternative view of K-Center:
Let $S \subseteq P$, $|S| = k$. For each $p \in P$, assign p to the nearest c̈enter; (i.e point in S). And for each $s \in S$, Ball(s) = set of points assigned to s. And the radius of s is radius(s) $= \max_{p' \in Ball(s)} D(p', s)$. And the radius of the set S Radius(S) $= \max_{s \in S}$ radius(s). We are looking for a subset $S \subseteq P$ with k points, whose radius is the smallest.