CIS 800/002 The Algorithmic Foundations of Data Privacy November 8, 2011

Lecture 16

Lecturer: Aaron Roth *Scribe: Aaron Roth*

Private Combinatorial Optimization: Vertex Cover

In this class, we will see one more approximation algorithm, this time for the *vertex cover* problem. The vertex cover problem differs from the min-cut problem we saw last time in two ways. The first (and least important) is that the vertex cover problem is NP complete. That is, we couldn't hope to have an efficient algorithm to get an exact solution even if we didn't care about privacy. But since we are concerned about privacy, we have to settle for an approximation anyhow...

The more important difference from our perspective, is that the constraints defining the set of feasible solutions for the vertex cover problem are a function of the private data! This means we won't be able to output a solution in its standard representation at all, and will have to get creative.

Definition 1 *An undirected graph $G = (V, E)$ is defined by a* vertex set V *of size $|V| = n$ and an* edge set $E \subset V \times V$. *If $(u, v) \in E$ We say that there is an edge between vertex u and vertex v if $(u, v) \in E$. For undirected graphs, we consider (u, v) to be equivalent to (v, u).*

Definition 2 *A vertex cover of $G = (V, E)$, $S \subseteq V$ is a subset of vertices such that for all $(u, v) \in E$, $u \in S$ or $v \in S$. The size of the vertex cover is $|S|$. The objective of the minimum vertex cover problem is to return a vertex cover of minimum cardinality. Write $OPT(G)$ to denote a vertex cover of minimum cardinality.*

First, let us observe two things: the cardinality of the minimum vertex cover is a 1-sensitive function (with respect to changes in the edge set), and so we can release this with noise only $O(1/\epsilon)$. On the other hand, suppose we must release an explicit vertex cover $S \subseteq V$. If there is any pair u, v such that $u, v \notin S$, then if S is a vertex cover, it must be that $(u, v) \notin S$. Hence, we won't be able to explicitely release a vertex cover S of size less than $|S| < |V| - 1$, and so no non-trivial approximations are possible.

Instead we will output an implicit representation of a vertex cover. For each *potential* edge $e = (u, v) \in V \times V$, we will output an orientation $c(e) \in \{u, v\}$. Such a set of orientations will correspond to a vertex cover in any graph $G = (V, E)$ as follows: the vertex cover will be $S = \bigcup_{e \in E} \{c(e)\}$. Note that S must by definition be a valid vertex cover for every graph G.

We will represent our edge orientations by outputting a permutation π over the vertices V. Each edge $e = (u, v)$ will have orientation $c(e) = u$ if u appears earlier than v in π. Our algorithm will start with a graph $G_1 = (V_1, E_1)$ and repeatedly select a vertex $v \in V$ to output, with probability proportional to its degree, plus some extra "hallucinated" edges, added for privacy. If a vertex v is selected, it is output as the next vertex in the permutation, and the algorithm recurses on $G_i = (V_i, E_i)$ where $V_i = V_{i-1} \setminus \{v\}$, and $E_i = E_{i-1} \setminus \{(u, v) : u \in V\}$. We write $n_i = |V_i| = n - i + 1$ and $m_i = |E_i|$. The algorithm is as follows:

Algorithm 1 Unweighted Vertex Cover

```
let $n \leftarrow |V|$, $V_1 \leftarrow V, E_1 \leftarrow E$.
for $i = 1, 2, \ldots, n$ do
    let $w_i \leftarrow (4/\epsilon) \times \sqrt{n/(n - i + 1)}$.
    pick a vertex $v \in V_i$ with probability proportional to $d_{E_i}(v) + w_i$.
    output $v$. let $V_{i+1} \leftarrow V_i \setminus \{v\}$, $E_{i+1} \leftarrow E_i \setminus (\{v\} \times V_i)$.
end for
```

Theorem 3 (Privacy) *ALG's differential privacy guarantee is* $\max\{1/w_1, \sum_i 2/iw_i\} \leq \epsilon$ *for the settings of w_i above.*

Proof For any two sets of edges A and B, and any permutation π, let d_i be the degree of the i^{th} vertex in the permutation π and let m_i be the remaining edges, both ignoring edges incident to the first $i-1$ vertices in π.

$$\frac{\Pr[ALG(A) = \pi]}{\Pr[ALG(B) = \pi]} = \prod_{i=1}^{n} \frac{(w_i + d_i(A))/((n-i+1)w_i + 2m_i(A))}{(w_i + d_i(B))/((n-i+1)w_i + 2m_i(B))} .$$

When A and B differ in exactly one edge, $d_i(A) = d_i(B)$ for all i except the first endpoint incident to the edge in the difference. Until this term $m_i(A)$ and $m_i(B)$ differ by exactly one, and after this term $m_i(A) = m_i(B)$. The number of nodes is always equal, of course. Letting j be the index in π of the first endpoint of the edge in difference, we can cancel all terms after j and rewrite

$$\frac{\Pr[ALG(A) = \pi]}{\Pr[ALG(B) = \pi]} = \frac{w_j + d_j(A)}{w_j + d_j(B)} \times \prod_{i \le j} \frac{(n-i+1)w_i + 2m_i(B)}{(n-i+1)w_i + 2m_i(A)} .$$

An edge may have arrived in A, in which case $m_i(A) = m_i(B) + 1$ for all $i \le j$, and each term in the product is at most one; moreover, $d_j(A) = d_j(B) + 1$, and hence the leading term is at most $1 + 1/w_j < \exp(1/w_1)$, which is bounded by $\exp(\epsilon/2)$.

Alternately, an edge may have departed from A, in which case the lead term is no more than one, but each term in the product exceeds one and their product must now be bounded. Note that $m_i(A) + 1 = m_i(B)$ for all relevant i, and that by ignoring all other edges we only make the product larger. Simplifying, and using $1 + x \le \exp(x)$, we see

$$\prod_{i \le j} \frac{(n-i+1)w_i + 2m_i(B)}{(n-i+1)w_i + 2m_i(A)} \le \prod_{i \le j} \frac{(n-i+1)w_i + 2}{(n-i+1)w_i + 0} = \prod_{i \le j} \left(1 + \frac{2}{(n-i+1)w_i}\right) \le \exp\left(\sum_{i \le j} \frac{2}{(n-i+1)w_i}\right) .$$

The w_i are chosen so that $\sum_i 2/(n-i+1)w_i = (\epsilon/\sqrt{n}) \sum_i 1/2\sqrt{i}$ is at most ϵ.

■

Theorem 4 (Accuracy) *For all G,* $\mathrm{E}[ALG(G)] \le (2+2avg_{i \le n} w_i) \times |OPT(G)| \le (2+16/\epsilon)|OPT(G)|$.

Proof Let $OPT(G)$ denote an arbitrary optimal solution to the vertex cover problem on G. The proof is inductive, on the size n of G. For G with $|OPT(G)| > n/2$, the theorem holds. For G with $|OPT(G)| \le n/2$, the expected cost of the algorithm is the probability that the chosen vertex v is incident to an edge, plus the expected cost of $ALG(G \setminus v)$.

$$\mathrm{E}[ALG(G)] = \Pr[v \text{ incident on edge}] + \mathrm{E}_v[\mathrm{E}[ALG(G \setminus v)]] .$$

We will bound the second term using the inductive hypothesis. To bound the first term, the probability that v is chosen incident to an edge is at most $(2mw_n + 2m)/(nw_n + 2m)$, as there are at most $2m$ vertices incident to edges. On the other hand, the probability that we pick a vertex in $OPT(G)$ is at least $(|OPT(G)|w_n + m)/(nw_n + 2m)$. Since $|OPT(G)|$ is non-negative, we conclude that

$$\Pr[v \text{ incident on edge}] \le (2 + 2w_n)(m/(nw_n + 2m)) \le (2 + 2w_n) \Pr[v \in OPT(G)]$$

Since $\mathbf{1}[v \in OPT(G)] \le |OPT(G)| - |OPT(G \setminus v)|$, and using the inductive hypothesis, we get

$$\begin{aligned} \mathrm{E}[ALG(G)] &\le (2 + 2w_n) \times (|OPT(G)| - \mathrm{E}_v[|OPT(G \setminus v)|]) + (2 + 2\mathrm{avg}_{i<n} w_i) \times \mathrm{E}_v[|OPT(G \setminus v)|] \\ &= (2 + 2w_n) \times |OPT(G)| + (2\mathrm{avg}_{i<n} w_i - 2w_n) \times \mathrm{E}_v[|OPT(G \setminus v)|] \end{aligned}$$

The probability that v is from an optimal vertex cover is at least $(|OPT(G)|w_i + m)/(nw_i + 2m)$, as mentioned above, and (using $(a + b)/(c + d) \ge \min\{a/c, b/d\}$) is at least $\min\{|OPT(G)|/n, 1/2\} =$

$|OPT(G)|/n$, since $|OPT(G)| < n/2$ by assumption. Thus $\mathrm{E}[|OPT(G \setminus v)|]$ is bounded above by $(1 - 1/n) \times |OPT(G)|$, giving

$$\mathrm{E}[ALG(G)] \quad \leq \quad (2 + 2w_n) \times |OPT(G)| + (2\mathrm{avg}_{i<n} w_i - 2w_n) \times (1 - 1/n) \times |OPT(G)| \ .$$

Simplification yields the claimed results, and instantiating w_i completes the proof. ■

Bibliographic Information The vertex cover algorithm is from "Differentially Private Combinatorial Optimization", 2010 by Gupta, Ligett, McSherry, Roth, and Talwar.