HW2 MPRI 1.24 Randomized Algorithms

Thu. Jan. 30, 2014 - **Due on Thu. Feb. 6, 2014**

You are asked to complete the exercises and send me your solutions by email at:
`nicolas.schabanel@liafa.fr`
(or drop it in my mail box on the 4th floor) on or before **Thu. Feb. 6, 2014 at noon**.

■ **Exercise 1 (A semi-definite program-based approximation algorithm for Maximum Cut).**

We consider the following problem (Max Cut): Given a complete undirected graph $G = (V, E, w)$ with edge non-negative weights $w : E \to \mathbb{Q}_+$, find a partition $(C, \overline{C})$ of V so as to maximize the total weight of edges in this cut, i.e., edges that have one endpoint in C and one endpoint in $\overline{C} = V \setminus C$. Let $n = |V|$. Let OPT denote the maximum weight of a cut in G and $\sigma = \sum_{1 \leqslant i < j \leqslant n} w_{ij}$ be the sum of all edge weights.

▶ ***Question 1.1)*** *Show by considering an uniform random cut $(C, \overline{C})$ that:* $\frac{\sigma}{2} \leqslant \mathrm{OPT} \leqslant \sigma$.

A *semi-definite program* is defined over n n-dimensional *vector variables* $\mathbf{v}_1, \ldots, \mathbf{v}_n \in \mathbb{R}^n$, and consists in optimizing (minimizing or maximizing) a linear function of their inner products $\langle \mathbf{v}_i | \mathbf{v}_j \rangle_{1 \leqslant i,j \leqslant n}$ subject to linear constraints on these inner products. Consider the following semi-definite program relaxation for Max-Cut.

$$\begin{cases} \text{Maximize} & \displaystyle\frac{1}{2} \sum_{1 \leqslant i < j \leqslant n} w_{ij} \cdot (1 - \langle \mathbf{v}_i | \mathbf{v}_j \rangle) \\ \text{subject to} & (\forall i) \quad \langle \mathbf{v}_i | \mathbf{v}_j \rangle = 1 \\ & (\forall i) \quad \mathbf{v}_i \in \mathbb{R}^n \end{cases}$$

▶ ***Question 1.2)*** *Show that it is indeed a relaxation of the Max Cut problem, i.e. that for all cut $(C, \overline{C})$ in G, there is a feasible vector solution $\mathbf{v}_1, \ldots, \mathbf{v}_n \in \mathbb{R}^n$ to the semi-definite program whose objective value is equal to the weight of the cut. How does the optimal value of this program compare to the weight* OPT *of an optimum cut in G?*
▷ Hint. $\langle \mathbf{v}_i | \mathbf{v}_j \rangle = \pm 1$ *whenever $\mathbf{v}_i$ and $\mathbf{v}_j$ point in the same or in opposite directions.*

Let us denote by OPT_{SDP} the optimal objective value of the semi-definite relaxation. Semi-definite programs can be solved in polynomial time up to an arbitrary small additive constant η using the ellipsoid algorithm, i.e. there is a poly$(n, \log(1/\eta))$-time algorithm that outputs a solution $\mathbf{v}$ with objective value at least $\mathrm{OPT}_{SDP} - \eta$ for all $\eta > 0$.

Intuitively, the objective value is constructed in such a way that the vectors associated with vertices linked by heavy edges should point in opposite direction, and thus should lie in opposite parts of the n-dimensional sphere. This motivates the following randomized rounding algorithm on the following page.

▶ ***Question 1.3)*** *Assume we have a procedure that picks a real value $x \in \mathbb{R}$ at random according to the normal probability density $d(x) = \frac{1}{\sqrt{\pi}} e^{-x^2}$. Design an algorithm that picks an uniform random unit vector* $\mathbf{u}$ *in $\mathbb{R}^n$. How would you proceed to sample a real x according the normal law?*
▷ Hint. *What about sampling each coordinate independently according to the normal law?*

Algorithm 1 SDP rouding algorithm for Max-Cut

1: Let $\mathbf{v}_1^*, \ldots, \mathbf{v}_n^*$ be an optimal solution to the semi-definite program
2: Pick a random hyperplane H in $\mathbb{R}^n$ passing through 0, i.e. pick an uniform random unit vector $\mathbf{u} \in S(0,1)$ and set $H = \mathbf{u}^\perp$
3: Let $C = \{i : \langle \mathbf{u} | \mathbf{v}_i^* \rangle \geqslant 0\}$ the set of vertices whose vertex lies in the positive side of the hyperplane H
4: **return** the cut $(C, \overline{C})$

Let us now analyze the performance of this rounding scheme in terms of approximation ratio. Let θ_{ij} denote the angle between each pair of vectors $\mathbf{v}_i^*$ and $\mathbf{v}_j^*$ (defined as the angle between them in the plane defined by themselves, i.e. $\theta_{ij} = \arccos(\langle \mathbf{v}_i^* | \mathbf{v}_j^* \rangle) \in [0, \pi]$).

► ***Question* 1.4*)*** *Show that the contribution of each edge ij to the optimal value of the semi-definite program is: $w_{ij} \dfrac{1 - \cos \theta_{ij}}{2}$.*

► ***Question* 1.5*)*** *Show that the probability for each edge ij to be cut in the randomized procedure is exactly: $\dfrac{\theta_{ij}}{\pi}$.*
▷ Hint. *Look at the footprint of the hyperplane in the plane defined by $\mathbf{v}_i^*$ and $\mathbf{v}_j^*$.*

Let $\alpha = \dfrac{2}{\pi} \min_{\theta \in [0,\pi]} \dfrac{\theta}{1 - \cos \theta} > 0.87856$. Let W be the random variable for the value of the solution output by the randomized rounding algorithm.

► ***Question* 1.6*)*** *Show that $\mathbb{E}[W] \geqslant \alpha \cdot \mathrm{OPT}$.*

Recall that σ denotes the sum of all edge weights.

► ***Question* 1.7*)*** *Using that $W \leqslant \sigma$ and $\mathrm{OPT} \geqslant \dfrac{\sigma}{2}$, show that $\mathbb{V}\mathrm{ar}(W) \leqslant \left(\frac{4}{\alpha^2} - 1\right) \mathbb{E}[W]^2$. Conclude that there is a α-approximate (ϵ, δ)-estimator for the weight of the maximum cut of G.*

► ***Question* 1.8*)*** *In the step 1. of the algorithm, one can get in polynomial time only a solution whose value is guarantee to be at least $\mathrm{OPT}_{SDP} - \eta$ for all $\eta > 0$. Explain how to absorb this inaccuracy overcost in the approximation ratio.*

■ **Exercise 2 (Randomized rounding for Set-Cover).** Consider the following problem (Set-Cover): Given an universe $\mathcal{U} = \{u_1, \ldots, u_m\}$ of m elements, a collection $\mathcal{S} = \{S_1, \ldots, S_n\}$ of n subsets of $\mathcal{U}$ such that $\bigcup_{i=1}^n S_i = \mathcal{U}$, and positive weights (w_i) on each S_i, find a subcollection $C \subset \mathcal{S}$ covering $\mathcal{U}$ (i.e. such that $\bigcup_{S_i \in C} S_i = \mathcal{U}$) with minimum weight $w(C) = \sum_{S_i \in C} w_i$.

We consider the following linear relaxation for Set-Cover:

$$\begin{cases} \text{Minimize} & \sum_{i=1}^n w_i x_i \\ \text{such that} & (1)\ \displaystyle\sum_{i: u_j \in S_i} x_i \geqslant 1 \quad (\forall j \in \{1, \ldots, m\}) \\ & (2)\ 1 \geqslant x_i \geqslant 0 \qquad (\forall i \in \{1, \ldots, n\}) \end{cases}$$

► ***Question* 2.1*)*** *Show that this linear program (which can be solved in polynomial time in n and m) is a relaxation of Set-Cover. Show that the constraints $1 \geqslant x_i$ can be waived (i.e. that removing them does not change the optimal solution of the linear program). What is the trivial relationship between OPT_{LP} (the optimal value of the linear program) and OPT (the weight of an optimal set cover)?*

Now assume that we have obtained an optimal fractional solution (x_i^*) to the linear program. We now want to transform it into a proper set cover. A natural approach is to interpret the x_i^* as the probability that S_i belongs to the aimed optimal subcollection. Let us denote by C a random subcollection obtained by putting each S_i in C independently with probability x_i^*.

▶ ***Question* 2.2*)*** *What is the expected weight of C?*

▶ ***Question* 2.3*)*** *Fix some $u_j \in \mathcal{U}$. Show that the probability that u_j is not covered by C (i.e. that $u_j \notin \bigcup_{S_i \in C} S_i$) is at most $1/e$.*
▷ Hint. *Use that $\prod_k (1 - a_k) \leqslant e^{-\sum_k a_k}$ for all $a_k \leqslant 1$.*

We now consider the following algorithm:

Algorithm 2 Randomized rounding for Set-Cover

1: Let x_i^* be an optimal solution to the linear program. Let $\ell := 0$.
2: **repeat**
3: Set $\ell := \ell + 1$ and Draw a random collection C_ℓ by selecting each S_i in C_ℓ independently with probability x_i^*.
4: **until** $\displaystyle\bigcup_{k=1}^{\ell} C_k$ covers $\mathcal{U}$.
5: Set $T := \ell$ and output the collection $\Gamma := \displaystyle\bigcup_{k=1}^{T} C_k$.

Note that the variable T (line 5) stands for the random variable for the number of random collections successively computed by the algorithm. Since all the C_k's are independently and identically distributed as C in question 2.2, and since T only depends on $C_1, \ldots, C_T$, Wald's equation (admitted) ensures that

$$\mathbb{E}[w(\Gamma)] \leqslant \mathbb{E}[w(C_1) + \cdots + w(C_T)] = \mathbb{E}[T] \cdot \mathbb{E}[C].$$

We are thus now left with estimating $\mathbb{E}[T]$.

▶ ***Question* 2.4*)*** *Show that: for all k,* Pr$\left\{\exists j, u_j \text{ is not covered by } C_1 \cup \cdots \cup C_k\right\} \leqslant \dfrac{m}{e^k}$.

▶ ***Question* 2.5*)*** *Show that:* Pr$\left\{T > \lceil \ln m \rceil + t\right\} \leqslant e^{-t}$, *for all $t \geqslant 0$. Conclude that: $\mathbb{E}[T] \leqslant \ln m + O(1)$.*
▷ Hint. *Use that $\mathbb{E}[X] = \sum_{n \geqslant 0}$ Pr$\{X > n\}$ for every non-negative integer-valued random variable X. And decompose $\mathbb{E}[T]$ according to the events $\{T \leqslant \ln m\}$ and $\{T > \ln m\}$.*

▶ ***Question* 2.6*)*** *What is the running time of our algorithm (excluding the resolution of the linear program)? Give an upper bound on the expected weight of its output. What is its approximation ratio?*