Sapienza University of Rome

Algorithm Design Second Homework

Andrea Fioraldi 1692419

January 15, 2019

Exercise 1

We redefine our problem as a problem on graphs. Let $G(M, F, E)$ a fully connected graph in which each friend is a node. Each edge (u, v) has a cost $w(u, v) \geq 0$. We define the following functions:

$deg(v, G) := \sum_{u \in G(M) \cup G(F)} w(v, u)$ $\qquad$ $density(G) := \frac{1}{|G(M) \cup G(F)|} \sum_{v,u \in G(E)} w(v, u)$

```
1  algorithm michele_party_approx(G)
2      n := |G(M) ∪ G(F)|
3      H_n := G
4      max_d := −1
5      s := −1
6      for i = n/2 to 1
7          d := density(H_i)
8          if d > max_d
9              max_d = d
10             s = i
11         m := v ∈ H_i(M) | deg(v, H_i) ≤ deg(u, H_i) ∀u ∈ H_i(M)
12         f := v ∈ H_i(F) | deg(v, H_i) ≤ deg(u, H_i) ∀u ∈ H_i(F)
13         H_{i−1} = H_i \ {m, f}
14     return H_s
```

Let OPT the optimal solution and $d_{OPT} = density(OPT) = \frac{\alpha}{\beta}$ its density (β is the number of nodes). Let $m \in OPT(M)$ and $f \in OPT(F)$.
Consider that $\frac{\alpha}{\beta} \geq \frac{\alpha - deg(m) - deg(f)}{\beta - 2} \Leftrightarrow \frac{deg(m) + deg(f)}{\beta - 2} \geq \frac{\alpha}{\beta - 2} - \frac{\alpha}{\beta} \Leftrightarrow deg(m) + deg(f) \geq \alpha - \frac{\beta - 2}{\beta} * \alpha = 2 * \frac{\alpha}{\beta}$. So in OPT the sum of the degree of a generic m and f is at least twice d_{OPT}.

Consider the iteration i of the algorithm in which for the first time two nodes $m \in OPT(M)$ and $f \in OPT(F)$ are removed. In the H_i graph all the pairs $m \in H_i(M)$ and $f \in H_i(F)F$ have $deg(m) + deg(f) \geq 2 * d_{OPT}$ thanks to the fact that we are going to remove nodes with minimual sum of degrees in which the previous observation holds. In H_i there are $\frac{|H_i(M) \cup H_i(F)|}{2}$ pairs that will be removed until $i = 1$. These couples does not share nodes. The total cost of the edges in H_i is greater than $\left(2 * d_{OPT} * \frac{|H_i(M) \cup H_i(F)|}{2}\right)/2$ (the division by 2 is due to the fact that we want to avoid to consider edges twice).
So we have that

$$density(H_i) \geq frac2 * d_{OPT} * \frac{|H_i(M) \cup H_i(F)|}{2} 2 * |H_i(M) \cup H_i(F)| = \frac{d_{OPT}}{2}$$

Since the algorithm returns the graph with the highest density over all the iterations we have a solution with density at least $\frac{d_{OPT}}{2}$. We proved that $michele\_party\_approx$ is a 2-approximation.
The cost is $O(n^3)$ because the most expensive operation in the body of the loop is the density that costs $|G(E)| = n^2$.

Exercise 2

Our solution was inspired by the Set Cover approximation proof described in [1] but follows the path of the proof shown during the class.
Let A is the set of required skills. S is the set of all the people available, each people is represented as a set of skills $S_j \subseteq A$. Let $n = |A|$.
We can express the Set Cover with Redundancies problem using the following ILP formulation:

$$\begin{aligned} &\min \sum_{S_j \in S} c_j * x_j \quad \text{s.t.} \\ &\sum_{S_j | A_i \in S_j} x_j \geq 3 \quad \forall A_i \in A \\ &x_j \in \{0,1\} \quad \forall S_j \in S \end{aligned}$$

In order to build a randomized approximation consider the associated LP relaxion where $x_j^* \in [0,1]$. The LP solution is a vector x^* of real values.
Consider the algorithm ALG in which each person S_j is chosen randomly with probability $p_j = min(d * log(n) * x_j^*, 1)$ with all choices that are indepedent. We denote the vector of these choices as x'.
Let $C_i = \sum_{S_j | A_i \in S_j} x'_j$ a random variable that represents the times that the skill A_i is covered. The expectation of C_i is $\mathbb{E}[C_i] = \mathbb{E}\left[\sum_{S_j | A_i \in S_j} x'_j\right] = \sum_{S_j | A_i \in S_j} p_j$. Then, $\mathbb{E}[C_i] = d * log(n) * \sum_{S_j | A_i \in S_j} x_j^* \geq d * log(n) * 3$ because the cover constraint in LP is satisfied.
From this, thanks to the Chernoff lower bound, follows that:

$$Pr\Big[C_i < 3\Big] = Pr\Big[C_i < \Big(1 - \Big(1 - \tfrac{1}{d*log(n)}\Big)\Big) * d * log(n) * 3\Big] \leq^1 Pr\Big[C_i < \Big(1 - \Big(1 - \tfrac{1}{d}\Big)\Big) * d * log(n) * 3\Big] \leq \exp\left(- \tfrac{1}{2} * 3 * d * log(n) * \Big(1 - \tfrac{1}{d}\Big)^2\right)$$

Note that $\left(1 - \frac{d*log(n)-1}{d*log(n)}\right) * d * log(n) = 1$. $0 < 1 - 1/d < 1$ must be valid in order apply Chernoff and so we have that $d > 1$.
Thanks to the Union bound we can get the probability that at least one A_i is not covered 3 times.

$$Pr\Big[\bigcup_{A_i \in A} C_i < 3\Big] \leq \sum_{A_i \in A} Pr\Big[C_i < 3\Big] = n * \exp\left(- \frac{1}{2} * 3 * d * log(n) * \Big(1 - \frac{1}{d}\Big)^2\right)$$

Note that increasing d give us a better probability that all skills are covered but also, on the other hand, decrease the probability to have a minimal solution because over a threshold we have that all the variables in x' are 1.
The expected cost is $\mathbb{E}\left[\sum_{S_j \in S} c_j * x'_j\right] = \sum_{S_j \in S} c_j * \mathbb{E}[x'_j] = \sum_{S_j \in S} c_j * d * log(n) * x_j^*$ and so it is $d * log(n)$ times the cost of the LP.

[1] This is done to get a result of the type $exp(W * log(n)) = n^W$

With the Markov inequality, we bound with a parameter k the probability to fail on having an average cost of less than $k * d * log(n)$ times the cost of LP.

$$Pr\Big[\sum_{S_j \in S} c_j * x'_j \geq k * d * log(n) * \sum_{S_j \in S} c_j * x^*_j \Big] \leq \frac{\sum_{S_j \in S} d*log(n)*c_j*x^*_j}{\sum_{S_j \in S} k*d*log(n)*c_j*x^*_j} = \frac{1}{k}$$

Obviously this is an event that we want to avoid and setting a big k help us in that but, on the contrary, give us also a worse approximation factor than $d * log(n)$.
We must choose d and k in a way that:

- minimizes the probability that our solution is not valid ($\cup_{A_i \in A} C_i < 3$);
- maximizes the probability that this solution is near to the optimal having the expected cost in a reasonable bound;

Choosing $d = 3$ we have an interesting result:

$$Pr\Big[\bigcup_{A_i \in A} C_i < 3 \Big] = n * \exp\big(-2 * log(n) \big) = \frac{1}{n}$$

The probability that all the skills are covered, so that x' is feasible, is $1 - \frac{1}{n}$ that is quite high and so we expect to need only few runs to get a feasible solution.
The cost of this approximation is at most $3 * k * log(n) * LP$.
The probability that the solution is not feasible or that the cost exceed the bound is

$$Pr\Big[\sum_{S_j \in S} c_j * x'_j \geq \sum_{S_j \in S} 3 * k * log(n) * x^*_j \cup \bigcup_{A_i \in A} C_i < 3 \Big] \leq \frac{1}{k} + \frac{1}{n}$$

Setting $k = 2$ we get that the solution is feasible and with cost less than $6 * log(n) * LP$ with probability greater than $\frac{1}{2} - \frac{1}{n}$ that is amost $\frac{1}{2}$ with a reasonable n.
We conclude that, as in the normal set cover approximation, the expected number of repetitions are 2 to get a feasible solution with a cost less the $6 * logn(n)$ times the cost of LP.

Exercise 3

We denote as F^* the optimal solution of the problem. F^* is the minimum cost set of edges that if removed creates k connected components with each target terminal s_i inside each component.
Let $F_i^* \subset F^*$ the set of edges that if removed separates the vertex s_i from the others s_j with $i \neq j$. We have that $\cup_{i=0}^{k} F_i^* = F^*$.
Each edge e in F^* is contained in two F_i^* because it is incident at two connected components. So we can say that:

$$\sum_{i=0}^{k} \sum_{e \in F_i^*} w(e) = 2 * \sum_{e \in F^*} w(e)$$

Our algorithm returns k min cuts F_i. By definition F_i is the minimum set of edges that separates s_i from the other s_j with $i \neq j$ so we have that $\sum_{e \in F_i} w(e) \leq \sum_{e \in F_i^*} w(e)$.

This implies that:

$$\sum_{i=0}^{k} \sum_{e \in F_i} w(e) \leq 2 * \sum_{e \in F^*} w(e)$$

And so, in the worst case, the cost of F is twice the cost of F^*. We proved that this is a 2-approximation algorithm.

Exercise 4

Let $n = |D|$. Let $sub(D, i, j)$ the substring of D from i to $j - 1$. As described in [2] we define a directed graph with $V = \{0, ..., n\}$ (the set of the indexes of the nucleobases in D) and $E = \{(i, j) \subset V^2 | i < j \wedge sub(D, i, j) \in G\}$. A path from node 0 to n in this graph is a concatenation of genes that results in D. $\delta^-(v)$ is the set of incoming edges to node v and $\delta^+(v)$ the outcoming edges.
The variables x_k represents if a gene $G_k \in G$ is in D. The variables $z_{i,j}$ represents if an edge (i, j) is in the optimal path.
The ILP formulation is the following:

$$
\begin{array}{lll}
 & \min \sum_{k=1}^{m} w_k * x_k & \text{s.t.} \\
(1) & x_{k|G_k = sub(D,i,j)} - z_{i,j} \geq 0 & \forall (i,j) \in E \\
(2) & \sum_{(i,j) \in \delta^+(0)} z_{i,j} = 1 & \\
(3) & \sum_{(i,j) \in \delta^-(v)} z_{i,j} - \sum_{(i,j) \in \delta^+(v)} z_{i,j} = 0 & \forall v \in V \setminus \{0, n\} \\
 & x_k, z_{i,j} \in \{0, 1\} &
\end{array}
$$

(1) is to force to consider a gene if a substring equal to the gene is taken. (2) and (3) are for the flow integrity.
Consider now the LP-relaxion with $x_k, z_{i,j} \geq 0$. Let's compute the dual of it. We associate varibales and constraints as follows: $(1) \to q_{i,j}, (2) \to p_0, (3) \to p_v$ with $v \in V \setminus \{0, n\}, x_k \to (4), z_{i,j} \to (5)$. The A matrix of the primal has dimension $(n + |E|) \times (m + |E|)$. Considering the fact that and edge (i, j) can be only in a $\delta^-(v)$ and only in a $\delta^+(v)$ we can easily compute A^T. Only constraint (2) contribute to the objective function and only q is sign constrained (cfr. [3]).

$$
\begin{array}{lll}
 & \max p_0 & \text{s.t.} \\
(4) & \sum_{i,j | sub(D,i,j) = G_k} q_{i,j} \leq w_k & \forall k \in \{1, ..., m\} \\
(5) & -q_{i,j} + p_i - p_j \leq 0 & \forall (i,j) \in E \\
 & q_{i,j} \geq 0 &
\end{array}
$$

Exercise 5

The proposed problem looks like a finite zero-sum game theory problem.

Comet, Dasher	Head	Tail
Head	4, -4	-1, 1
Tail	-2, 2	2, -2

We create the Comet's payoff matrix $A_{2,2}$ taking the first value of each table. The matrix $B_{2,2} = -A$, on the other hand, is the payoff matrix for Dasher. Santa must assign the probability to get head or tail for each coin in order to convince both Comet and Dasher to play. To do that the game the game must not be balanced against either player and so we equal the expected payoff of the two players in order to make the game fair.

$$Pr[\text{Comet coins is head}] = p_h \quad Pr[\text{Comet coins is tail}] = p_t \quad p = \begin{pmatrix} p_h \\ p_t \end{pmatrix} = \begin{pmatrix} p_h \\ 1 - p_h \end{pmatrix}$$
$$Pr[\text{Dasher coins is head}] = q_h \quad Pr[\text{Dasher coins is tail}] = q_t \quad q = \begin{pmatrix} q_h \\ q_t \end{pmatrix} = \begin{pmatrix} q_h \\ 1 - q_h \end{pmatrix}$$

The expected payoff of Comet is $p^T * A * q$ and the expected payoff of Dasher is $p^T * B * q = -p^T * A * q$. When they are equal we have that $2 * p^T * A * q = 0$ and so $p^T * A * q = 0$ is out constraint in order to have a fair game.

$$\begin{pmatrix} p_h & 1 - p_h \end{pmatrix} * \begin{pmatrix} 4 & -1 \\ -2 & 2 \end{pmatrix} * \begin{pmatrix} q_h \\ 1 - q_h \end{pmatrix} = \begin{pmatrix} p_h & 1 - p_h \end{pmatrix} * \begin{pmatrix} 5 * q_h - 1 \\ -4 * q_h + 2 \end{pmatrix} = p_h * (5 * q_h - 1) + (1 - p_h) * (-4 * q_h + 2) = 9 * p_h * q_h - 3 * p_h - 4 * q_h + 2$$

$9 * p_h * q_h - 3 * p_h - 4 * q_h + 2 = 0$ with $p_h, q_h \in [0, 1]$ is the constraint that give to santa a method how to select the probabilities in order to havea fair game.
Note that this equation is a piece of an hyperbola. A valid solution is, as instance, $p_h = 1 \wedge p_t = 0 \wedge q_h = \frac{1}{5} \wedge q_t = \frac{4}{5}$.

Exercise 6

Let h the position of Giorgio's home. Let $S(i) = Pr[Safe|x(t) = i]$ the probability that Giorgio goes to home safely when he is at position i. We know that $Pr[x(t+1) = x(t)+1] = p$ and $Pr[x(t+1) = x(t)-1] = 1-p$. Follows that:

$$S(i) = \begin{cases} 0 & \text{if } i = -1 \\ 1 & \text{if } i = h \\ p * S(i-1) + (1-p) * S(i+1) & \text{otherwise} \end{cases}$$

We have that in general $S(i) = p*S(i-1)+(1-p)*S(i+1)$ and $S(i) = p*S(i)+(1-p)*S(i)$ (from $1 = p-(1-p)$) and so $S(i+1) = \frac{p}{1-p} * \big(S(i) - S(i-1)\big) + S(i)$. Follows that $S(i+2) = \frac{p}{1-p} * \big(S(i+1) - S(i)\big) + S(i+1) = \frac{p}{1-p}\Big(\frac{p}{1-p}\big(S(i) - S(i-1)\big) + S(i) - S(i)\Big) + \frac{p}{1-p} * \big(S(i) - S(i-1)\big) + S(i)$.
With $i = 0$ we have $S(2) = \big(\frac{p}{1-p}\big)^2 * S(0) + \frac{p}{1-p} * S(0) + S(0)$ and so clearly:

$$S(i) = \sum_{j=0}^{i} \Big(\frac{p}{1-p}\Big)^j * S(0).$$

Consider now when $i = h$. The problem says that Giorgio makes an infinite number of steps so we can set $h = +\infty$.

$$1 = S(+\infty) = S(0) * \sum_{j=0}^{+\infty} \Big(\frac{p}{1-p}\Big)^j.$$

This is a geometric series [4]. By definition if $|\frac{p}{1-p}| < 1$ it converges to $\frac{1}{1-\frac{p}{1-p}}$.
The probability that Giorgio goes to the hospital from position 0 is $Pr[Hospital|x(t) = 0] = 1 + Pr[Safe|x(t) = 0] = 1 - S(0)$.
$|\frac{p}{1-p}| < 1 \Leftrightarrow p < 1/2$ and so, finally, we have that:

$$Pr[Hospital|x(t) = 0] = 1 - \frac{1}{\sum_{j=0}^{+\infty} \big(\frac{p}{1-p}\big)^j} = \begin{cases} \frac{p}{1-p} & p < 1/2 \\ 1 & p \geq 1/2 \end{cases}$$

Giorgio goes to hospital for sure when $p \geq \frac{1}{2}$ and goes to hospital with probability at most $\frac{1}{2}$ when $\frac{p}{1-p} \leq \frac{1}{2} \Rightarrow p \leq \frac{1}{3}$.

References

[1] "Randomized rounding - Wikipedia." https://en.wikipedia.org/wiki/Randomized_rounding#Proof. Accessed: 2019-1-3.

[2] S. Canzar, T. Marschall, S. Rahmann, and C. Schwiegelshohn, "Solving the minimum string cover problem," in *ALENEX*, pp. 75–83, SIAM / Omnipress, 2012.

[3] "Massimo Roma - Appunti dalla lezione di Ricerca Oprativa, chapter 8." www.dis.uniroma1.it/~roma/didattica/RO17-18/cap8.pdf. Accessed: 2019-1-3.

[4] "Geometric series - Wikipedia." https://en.wikipedia.org/wiki/Geometric_series#Geometric_power_series. Accessed: 2019-1-3.