Mini-project : Deploying a 5G Network in a country

In telecommunications, 5G networks are the next generation of broadband cellular networks. Telecommunication companies are actively testing and starting to roll them out in different parts of the world.

As part of a deployment team, your task is to deliver a roadmap to test this new network while optimizing the cost of the maintenance of the new installations. More precisely, there are two adversarial factors that you will need to take into account:

- You obviously want to place antennas in cities where the market is significant, and you will assume this is proportional to the population of the cities as an approximation. Formally, you are given a set of n cities with their given (normalized) populations $v_1, \dots, v_n$ and you have to choose a subset $\mathcal{S} \in \mathcal{P}(\{1, \dots, n\})$ of these cities.
- As the technology is new, some fine tuning and adjustments are expected to take place after the installation of the antennas. In other words, some impending technical costs are expected. And because there are only a few 5G technical teams available for now, deploying the system at once in a whole given country is out of question. Therefore, as a good approximation, you will have to consider the mobility of the team within the deployment area, which will be approximated as the area of the smallest circle enclosing all cities in $\mathcal{S}$, given the n locations $x_1, \dots, x_n$.

In summary, your task is to maximize the following objective function:

$$f(\lambda, \mathcal{S}) = \sum_{i \in \mathcal{S}} v_i - \lambda \cdot n \cdot \max_{(i,j) \in \mathcal{S}^2} \pi \left(d(x_i, x_j)/2 \right)^2$$

where $\mathcal{S}$ denotes the set of chosen cities, $d(x, y)$ denotes the Euclidean distance between cities at positions x and y, and λ is a parameter indicating the importance attributed to the deployment cost. Likewise, we will denote by

$$\mathcal{S}^\star(\lambda) = \underset{\mathcal{S} \in \mathcal{P}(\{1,\dots,n\})}{\arg\max} f(\lambda, \mathcal{S})$$

the corresponding optimizing set[1].

At the beginning, a few cities are chosen for testing purposes to set up the initial configuration of the antennas. Then, as the company acquires more feedback and knowledge throughout the testing, the technical cost is expected to evolve until it is worth deploying the network in the entire country. To be more specific, $\lambda = \lambda_0$ is expected to be large at the beginning, leading to a small cardinality of the set $\mathcal{S}_0 = \mathcal{S}^\star(\lambda_0)$. Conversely, $\lambda = \lambda_1$ is expected to be small at the end, in which case most or even all cities will be selected.

The aim of the project is to obtain, for each *fixed* value of the parameter λ, an approximate solution of $\mathcal{S}^\star(\lambda)$ using the Metropolis-Hastings algorithm. Then your objective is to understand how the number of selected cities (that is, the cardinality $|\mathcal{S}^\star(\lambda)|$) and the objective function (that is, $f(\lambda, \mathcal{S}^\star(\lambda))$) evolve with respect to the parameter λ, so as to be able to plan carefully the deployment.

The following questions are provided as a starting point for your exploratory analysis:

1. Implement an approximate solution of $\mathcal{S}^\star(\lambda)$ using Metropolis-Hastings in the core of your algorithm as seen in the course. You are free to add any optimizations which may help converge towards the desired approximate solution.
2. Test your implementation on the following generative model:[2]

$$\mathcal{G}_1 : \quad \left\{ n = 100, \quad (v_i)_{1 \le i \le n} \sim \mathcal{U}([0,1])(\text{iid}), \quad (x_i)_{1 \le i \le n} \sim \mathcal{U}([0,1]^2)(\text{iid}) \right\}$$

with varying $\lambda \in [0, 1]$.

[1] Note that $\mathcal{S}^\star(\lambda)$ need not be unique.

[2] You can use the skeleton Python notebook `https://github.com/antoinexp/markov-chains-COM-516/blob/main/model.ipynb`

(a) Explain how your algorithm evolves towards the solution for a given instance of $\mathcal{G}_1$. You may want to visualize how many cities are being selected at various steps of the algorithm and plot how the objective function evolves depending on your own algorithmic optimizations.

(b) Explain how $\mathbb{E}_{\mathcal{G}_1}[|\mathcal{S}^\star(\lambda)|]$ and $\mathbb{E}_{\mathcal{G}_1}[f(\lambda, \mathcal{S}^\star(\lambda))]$ evolve with respect to λ using a plot (the average being taken over multiple instances of $\mathcal{G}_1$).

3. Same questions for the following generative model (pay attention to ln):

$$\mathcal{G}_2 : \quad \{n = 100, \quad (\ln v_i)_{1 \leq i \leq n} \sim \mathcal{N}(-0.85, 1.3)(\text{iid}), \quad (x_i)_{1 \leq i \leq n} \sim \mathcal{U}([0,1]^2)(\text{iid})\}$$

with varying $\lambda \in [0, 2]$.

You are expected to write a report of maximum 4 pages tackling the previous questions, and provide next to that the source code of your solution (preferably in Python).

The 4 page report should contain explanations about the details of the Metropolis-Hasting implementation (e.g., specifics of the proposed moves and any extra optimization method that is used). You should select your most meaningful plots and write informative comments about them.

Submission deadline: 14/12/2020 at 23:59