A Reinforcement Learning Approach to Managing Association and Handovers in Cellular Networks: Project Proposal

Saadallah Kassir
Nithin S. Ramesan

October 24, 2019

1 Project Motivation

In current cellular networks, fairly simple network policies are adopted to solve the mobile user association problem, i.e. the decision of which base station (BS) a user (or UE - user equipment) should connect to. One typical policy would be for a UE to associate to the "closest" BS, or more formally, the BS from which the UE receives the best average Signal-to-Interference and Noise Ratio (SINR). The common thread tying these policies together is that they are constant over the entire network and hence blind to local network information that could potentially be used to improve association. Consider the following toy example: a UE associates to its closest BS, which happens to be serving a large number of other users at the same time. While this might be the best decision to make whilst only considering signal strength, the reduced bandwidth that the UE will receive from this BS might ideally prompt it to associate to a further away, but less loaded BS in order to receive better throughput.

In this project, we propose to explore the learning of a more *aware* set of association policies that could be followed by UEs in order to improve the throughput they receive from their associated BS.

2 Project Structure

In order to gain better intuition about and to work our way up to the final solution, we start by analyzing a simplified version of this problem, before increasing the complexity of our model to capture relevant network features that may impact the association decision in real deployments. In all environments described below, the decision making can be seen as an episodic task, where at the start of each episode, the user's position (and motion, in later stages of the project) is initialized randomly and the user needs to determine its association in a static environment over T time steps.

In all of the following environments, we will not be using any existing codebases. The simulation of cellular wireless networks is is well-established in literature, and hence, generating our environment and episodes will be straightforward. We will implement all algorithms we use.

2.1 Initial Model

The first configuration we will analyze is the setting where BSs are randomly deployed in space, according to a spatial 2D homogeneous Poisson Point Process, with density λ_{BS}. We look at the decision that needs to be taken by a UE, assuming it is the only active device in the network, where the UE seeks to maximize the Shannon capacity/rate C_i it receives from the serving BS i at time t:

$$C_i(t) = B \cdot \log_2(1 + \text{SINR}_i(t)) \tag{1}$$

where B models the bandwidth used by the BSs, and where the SINR at time t can be formally expressed as

$$\text{SINR}_i(t) = \frac{p_i \cdot D_i(t)^{-\alpha}}{\sigma^2 + \sum_{j:j \neq i} p_j \cdot D_j(t)^{\alpha}} \tag{2}$$

where p_b is the transmission power of BS b, $D_b(t)$ is the distance between BS b and the tagged UE at time t, α is the path loss coefficient, typically between 3 and 4, and σ^2 is the thermal noise power.

The reward that the user obtains will be the Shannon rate over the entire period, $\sum_{t=0}^{T} C_i(t)$. In such setting, the state-space can be the SINR received by the user from the k closest BSs. The action space will be the decision of which BS to associate to.

We know that in such a configuration, the optimal decision to take is to associate to the closest BS if they all transmit at the same power. Hence, our first result will be to verify that such a policy can be recovered using reinforcement learning methods. The overarching algorithm we intend to use in this project is SARSA(λ).

As this is the default association policy currently used in real network deployments, we will use the results of this section as a baseline to compare our final solution against.

2.2 Second Model Iteration

In our second iteration, we propose to consider the setting where the tagged UE sees a network that is already loaded, i.e., it will need to share the bandwidth resource with other UEs, placed in space with density λ_{UE}. In this setting, the state-space will need to be augmented to capture the load $N_b(t)$ at each BS b at time t, and the action-space will remain constant from the previous model.

In other words, we aim at maximizing the following reward R at each episode:

$$\sum_{t=0}^{T} \frac{C_i(t)}{N_i(t)} \tag{3}$$

$N_b(t)$ can either be constant over time, or be a random variable. The policies learned in this environment will optimize the scenario we describe in the introduction.

2.3 Third Model Iteration

In the third (and main) part of the project, we propose to capture the impact of mobility on the association policy. This is a critical feature to capture, as in practice, a session needs to be disconnected and then reconnected whenever a UE switches association, e.g., where a UE moves from one cell to another, wasting precious resources. We call this effect the *handover cost.* The long term throughput seen by a UE is therefore strongly impacted by the amount of handovers it needs to make. More precisely, we assume that each handover leads to an instantaneous throughput of 0 bits/s for τ time steps. It is therefore imperative to find a policy that captures this effect.

The new reward function now becomes

$$\sum_{t=0}^{T} \frac{C_i(t)}{N_i(t)} \mathbb{1}_{\{t \geq H(t)+\tau\}} \tag{4}$$

where $H(t)$ represents the time index of the last handover.

Naturally, the state space needs to be augmented to capture the user dynamics. We propose to include the last s SINR traces from the k closest BSs, allowing our algorithm to infer future SINR values based on this data. The action space, as before, will remain constant.

2.4 Further Analysis

Depending on time and progress, we intend to complement our work with side analysis, such as finding values for T, k and s that lead to satisfactory performance, while keeping the model as simple as possible. Further analysis can also include the examination of the impact of other parameters, such as τ, α, λ_{BS} and λ_{UE}.

3 General Comments

In all of our proposed environments, our state spaces are vectors of real numbers. It hence seems natural to use tile coding to approximate our state space. It is not immediately clear to us if using policy gradient methods will be useful, since we have a discrete, finite action space. Our choice of state space is motivated by two considerations: one, in real deployments, UEs only have access to SINR meaurements. Two, we'd like the policy that we learn to be invariant of the specific BS configuration - we'd like the policy to generalize beyond the deployments that we use for training, and to be applicable to any cellular network realization.