Multi-Armed Test Distribution

Joe McCall   Chris Zahuranec

03-23-2020

Abstract

When an infectious disease such as COVID-19 infects an area, a common problem is a shortage of test kits. The resulting lack of adequate information on the spread of the disease makes it difficult to mitigate. Furthermore it is reasonable to conclude that the novel nature of such diseases imply a limited supply of testing capability at first. This experiment runs on data pulled from a public data source on actual infection rates, and explores whether or not the classical artificial intelligence problem of the multi-armed bandit yields useful test distribuiton strategies. Specifically, the epsilon-greedy algorithm is tested against a uniform-distribution model to determine if, given a more limited test supply, it out-performs the naive approach.

1   Introduction

During the COVID-19 pandemic, test kits were distributed among counties to help assess the spread of the disease, and determine whether or not treatment was necessary. Ideally there are adequate tests for every person in a geographical area, but given the novel nature of the disease the test kits were in short supply.

While test kits are manufactured every day, the distributor of those test kits is left with a dilemma: should one distribute more to areas that appear to be experiencing more infections? If so, how much of the limited test kits should be allocated to highly-infectuous areas and how much should be allocated to areas with low levels of infection?

By combining positive and negative cases into a "score," one could model this problem as a classical multi-armed bandit problem, which illustrates the dilemma between exploration and exploitation. Specifically, the exploration of areas with low infection rates should be reconciled with the exploitation of areas with high infection rates. Given the multi-armed bandit problem is established, one can apply several known solutions to this problem and evaluate its performance against real publicly-provided infection data [1]. Since the data is given by counties, the simulation will be run with counties.

The goal is to use an artificial intelligent agent to smartly distribute limited test kits where they are needed most. Can an artificially intelligent agent out-perform a naive approach of equal test distribution? Such an agent would help mitigate the test kit shortage until enough test kits can be distributed for everyone.

2   Methods

2.1   Classes

- Simulation - the container that sets up the simulation, defining the start/end dates, agents to use, and number of test kits available per-day.
- Environment - the container for the environment state at a specific day. Simulates the passage of time and computes the score of the agent.
- Agent - an abstraction of the agent that defines the sensors (`consume_result`) and actuators (`distribute_test_kits`).
  - NaiveAgent - a specification of Agent that distributes test kits evenly among all the counties.
  - EpsilonGreedyAgent - a specification of Agent that uses an epsilon-greedy approach to distribute test kits while balancing the exploration/exploitation trade-off.
- County - a county under simulation
- TestKitEvaluator - an asbtraction of a means to evaluate a test kit

- – RandomTestKitEvaluator - a specification of TestKitEvaluator that evaluates the test based on the result of a coin-flip
- – PandasTestKitEvaluator - a specification of TestKitEvaluator that reads in data from a dataset (using the Pandas library) to determine the appropriate percent of positive cases detected, and evaluate the test kit based on that percent chance of positivity.
- ResultConsumer - an abstraction of a class that consumes the result of the test
  - – PrintResultConsumer - prints the results of all the tests for that day to STDOUT
  - – PandasResultConsumer - stores the results in a Pandas data frame for later use
- ScoringStrategy - defines the scoring strategy in use

2.2 Data Acquisition and Storing (Pandas)

Pandas is an open source python library that is heavily used in the data science community that allows for the easy acquistion, transformation, and storage of data. Pandas was used in the follow ways for this project:

- Obtaining actual infection data from NY times that contains data by county and provides dates, infection counts, and death counts.
- Transformed the data from the entire United States to the five counties in Florida used for this project
- Computing, retrieving, and storing actual infection results and test results from the test kits the agent distributed
- Creating the graphs that are both generated when the simulation runs and the graphs included in this report

2.3 Simulation Overview

- Simulation run for T days.
- Each day N test kits are distributed among M counties
- Test kits evaluated based on percent of population infected for that day
- Score for each "arm" (county) computed based on the scoring strategy
- Results are fed into the agent to help determine the distribution of test kits
- The agent distributes the test kit
- The score for the agent is computed and added to the county
- At the end of the simulation the scores for each county are summed to display the final score.

2.4 Naive Agent

The test kits for day i (N_i) is described in fig. 1. It is simply the distribution of test cases evenly among the number of counties being simulated.

$$N_i = \frac{N}{M} \tag{1}$$

2.5 Epsilon-Greedy Agent

The epilon-greedy agent is described by a constant defined as ε, which is used in the computation of test kits distributed to the highest scoring counties per-day. This is illustrated in fig. 2. If the number of detected cases in the county the previous day (given by $S(i-1)$) is the greatest number (given by $max(S)$), then the number of additional test kits is is $N(1-\varepsilon)$. These are considered the "exploitation" test kits. The remaining "exploration" test kits are distributed evenly among all the counties.

$$N_i = \begin{cases} N(1-\varepsilon) + \dfrac{N\varepsilon}{M} & S(i-1) = max(S) \\ \dfrac{N\varepsilon}{M} & S(i-1) \neq S(P_m) \end{cases} \tag{2}$$

In other words, the value of ε indicates the agent's preference of exploration to exploitation. When $\varepsilon = 1$, the behavior agent's behavior is identical to the naive agent. When $\varepsilon = 0$ the agent distributes all of the test kits to the first county

that reports a positive result.

2.6 Score Function

The agent's effectiveness is computed based on both its measure of positive and negative tests. The number of points awarded for a positive test result is V_p, and for a negative test result is V_n. See fig. 3, where S is the score for the agent being tested, D is the number of days, C is the number of counties, and P_m is the number of measured positive cases in the simulation.

$$S = \sum_{c=1}^{C} \sum_{d=1}^{D} (V_p P_m(c,d) + V_n N_m(c,d)) \tag{3}$$

2.7 Test Evaluation

A test kit returns positive at a percent equal to the actual percent of positive cases for the county as reported by [1] for that day. See fig. 4, where $R_m(i,j)$ is the positive rate for county c at date d.

$$R_m(c,d) = \frac{P_a(C,D)}{T_a(c,d)} \tag{4}$$

The test is positive if a random float between 0 and 1 is less than $R_m(c,d)$.

3 Experiment

3.1 Parameters

The experiment was run on the following counties in Florida:

- Miami-Dade
- Broward
- Palm Beach
- Monroe
- Collier

These counties were chosen due to their proximity to one another and their differences in case numbers. The test simulated test kits being distributed between 21 January 2020, and 18 April 2020. A positive test is rewarded ten times higher than a negative test ($V_p = 10$, $V_n = 1$).

The naive agent was run first, then the epsilon-greedy agent was run with ε values of 1, 0.5, 0.2, 0.1, and 0.

3.2 Assumptions

- Test kit capacity remains constant throughout the simulation
- The population of each county is estimated by the total number tested (as reported by the Florida Department of Health [2]).

4 Results

Agent	Average Score
NaiveAgent	60820
EpsilonGreedyAgent, $\varepsilon = 1$	59368
EpsilonGreedyAgent, $\varepsilon = 0.5$	65862
EpsilonGreedyAgent, $\varepsilon = 0.2$	69072

Agent	Average Score
EpsilonGreedyAgent, $\varepsilon = 0.1$	68288
EpsilonGreedyAgent, $\varepsilon = 0$	69120

The test was run with several values of ε, along with the `NaiveAgent`. The results are graphed in Figure 1.
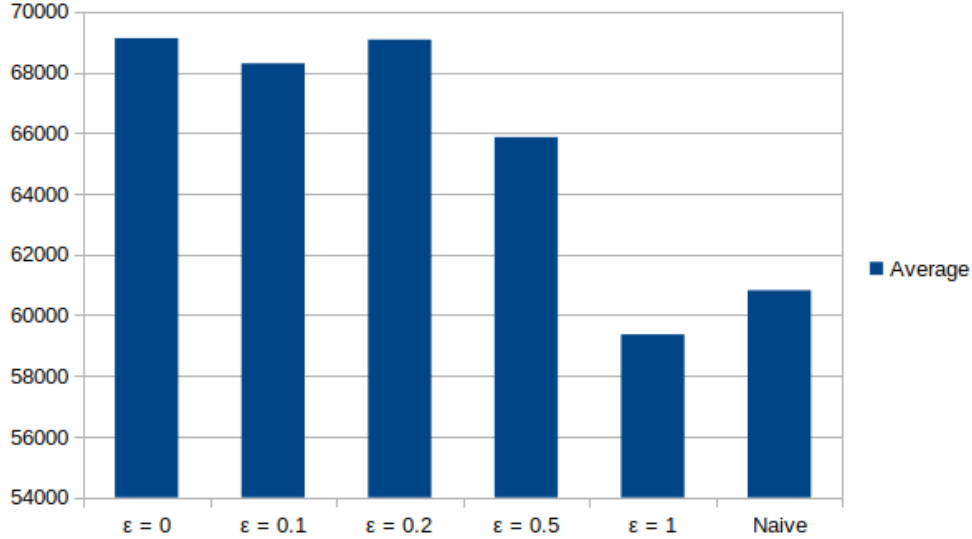

Figure 1: Score results for different agents (higher is better)

It is useful to view these results within the context of the data from the data-set to reveal how close they are to actually measured data. These are illustrated in figures 2 and 3. The top line represents the actual positive cases found during the reported time period, while the bottom line shows the positive test cases found by the agent during a simulated trial. The y-axis uses a logarithmic scale due to the large difference between the measured cases and the actual cases.

5 Conclusion

We found that our scoring algorithm rewarded exploitation heavily over exploration. The performance of the `NaiveAgent` performed slightly better than the performance of the `EpsilonGreedyAgent` with $\varepsilon = 1$.

As with many fields in AI the choice of a scoring function is very important when modelling real-world applications. Our simulation concluded that distributing the limited test kits to only the highest-infected county was the winning strategy. This conflicts with common knowledge, as at least some test kits must be distributed to uninfected counties in order to discover their infection rate. Put another way, our scoring function led to a local maxima, leaving the agent no choice but to continue distributing test kits to the highest county. The exploitation choice was favored heavily over the exploration choice.

There are further improvements that can be made. The local maxima can be alleviated with simulated annealing. Also, this experiment focused on five counties in Florida, but tests for more counties could prove useful. Further work could test the score of other solutions to the multi-armed bandit problem. A more sophisticated algorithm could take into account the county density, for example. Also the scoring function could be adjusted to reward negative test results, as those hold value as well as positive results. Another area worth exploring is training a neural network based on historical pandemic data to predict which areas should be prioritized before test kits are distributed.

Given the success of the multi-armed bandit problem when modelling other domains, there is high value in continuing this exercise with more advanced techniques. The problem of limited test kits will not be solved, given that novel
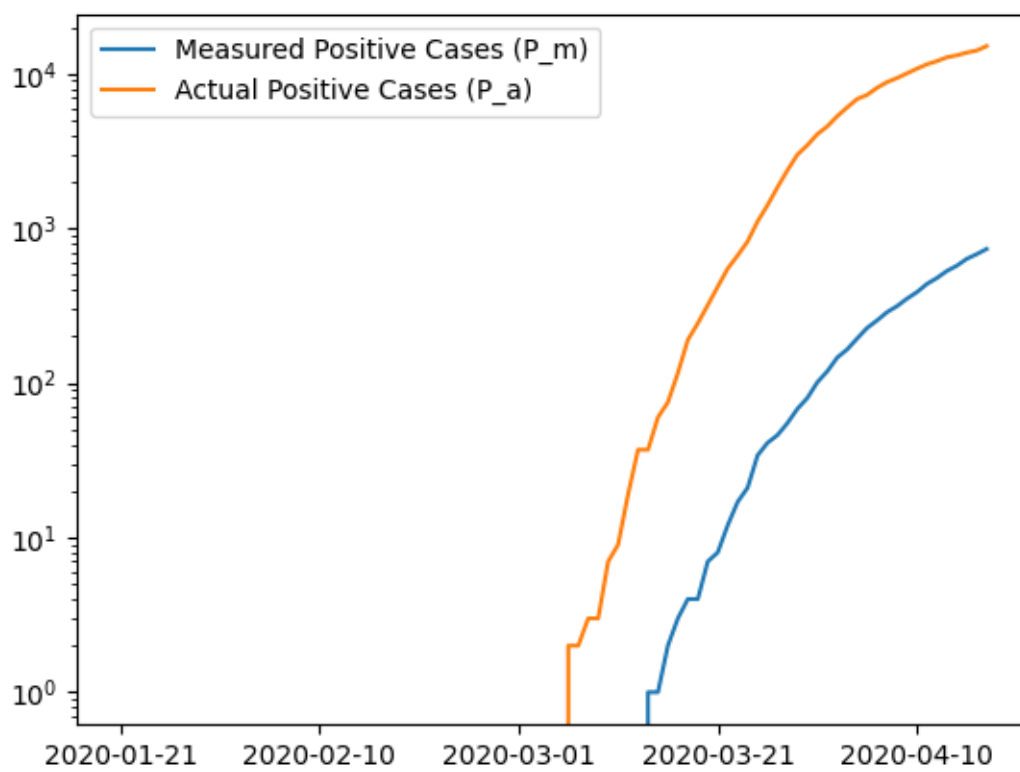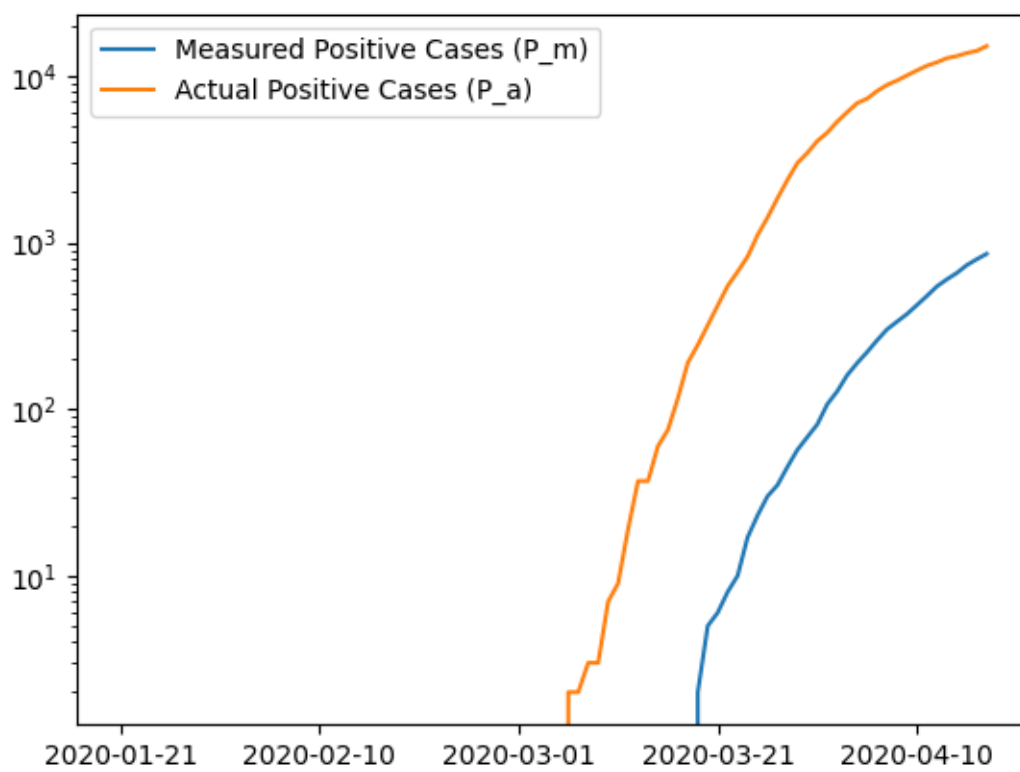
Figure 2: Naive Agent Results

Figure 3: Epsilon Greedy Agent Results ($\varepsilon = 0.5$)

diseases require novel test kits. While this experiment revealed a weakness in a specific choice of a scoring function, it still serves as a framework upon which new work can be built.

References

[1] "We're sharing coronavirus case data for every u.s. County," 2020. [Online]. Available: https://www.nytimes.com/article/coronavirus-county-data-us.html.

[2] "Florida covid19 cases by county," 2020. [Online]. Available: https://open-fdoh.hub.arcgis.com/datasets/florida-covid19-cases-by-county. [Accessed: 23-Apr-2020].