Learning from interactions

mail : marc.gelgon@univ-nantes.fr, office D111

Introduction

The applicative context related to clinical studies for investigating the efficiency of medical treatments. A new rare disease is discovered. Let us assume there exist a set of 10 different medical treatments, that might be interesting to fight this disease. However, initially we do not know which is (or which are) the most effective treatments among the 10.

The experimental protocol consists in 100 successive days. Every morning, you can freely select one (and one only) treatment among the 10 and apply it to a patient enrolled in the experiment. At the end of the day, a test shows whether the treatment has been effective or not (this is a binary value). This is repeated for 100 days. Hence, your input as the scientist is a sequence of 100 choices, starting from a "zero-knowledge" situation about the treatments.

While a reasonnable goal would be identify the most effective treatment(s) within the 10, the patients who are undergoing your experiments have volunteered for this and would very much like to be cured. Hence, your goal is to maximize the number of successful days among the 100 days involved in the experiment. You will be responsible for designing the experiment which tries to maximize the number of days (among 100) leading to a favourable outcome at the end of the day. What happens after the 100 days is irrelevant.

Your goal is to design a strategy (mathematical model/algorithm), implement it and evaluate it experimentally.

The goal of this short project is to explore on a small scale a typical problem and solutions in a specific branch of IA. However, the problem and techniques discussed below are truly commonly use in medical studies, as well as in recommender systems etc.

Two essential remarks :

1. the same treatment may sometimes lead to a positive result, sometimes a negative result, as if the outcome were random. However, some treatments are more often successful than others.
2. we assume the patients all behave the same way are they all behave the same way over the 100 days. As a result, no matter whether you consider this experiments involves a single patient or multiple patients. The patient can in fact be modelled as a set of 10 Bernoulli distributions which parameters are unknown to the medical team.

Work to do

You should implement both:

- the patient, which receives the number of the treatment selected on this morning and returns whether the treatment worked on that day. This can be achieved through sampling from the Bernoulli distribution corresponding to the selected treatment. You will need to assume a parameter value for each of the 10 Bernoulli distribution corresponding to the ten treatments. You can freely choose the 10 parameter values (hard-code them rather than sample them to make checking of results easier). Implementing the patient is only a few lines of code.

- the medical team, who selects every morning a treatment and gives it to the patient, then receives from the patient the binary result for that day. This is where your strategy is implemented.

Design a solution

Design a technique with unsophisticated maths to try to give a good result. Describe and argue for your choices in 8 to 12 lines. There are many possibilities, not a single good answer. In this type of problem, we face the need to do both exploration (gain some knowledge about the effectiveness of all possible treatments) and exploitation (using knowledge built so far, even if uncertain, to maximize our goal). For this, you should describe how your approach addresses exploration and exploitation and how it combines them.

Implement and evaluate results

Implement your technique and draw its result as a graph of the number of successes (as provided by the simulation that runs over the 100 days) as a function of the number of days. This graph provides richer information, compared to displaying a single number of successes over 100 days, as you can observe how your system improves over those 100 days. Your result should be non deterministic since patient simulation involves random sampling. Hence, you should evaluate the efficiency of your technique averaged over multiple runs.

Understanding some limitations

When you have built, implemented and evaluated your proposal, you should review the following questions :

a) what is different in the informations conveyed by the two following claims « getting 2 successes of out 4 trials » and « getting 20 successes of out 40 trials » ?
b) can this information be completely represented by the fraction of successful trials (e.g. 2 successful days amont 4= 1/2) ?
c) if we now try to represent the information about with a probability distribution over , rather than simply a number, what should be the shape of this distribution in the case we don't know anything about ? What should be its shape in the case we collected a clear majority of successes ?

An alternative method using Bayesian learning

Let's now implement the nice algorithm below for solving the above problem. It may or may not be more effective than yours, so let's compare them experimentally.

Representing knowledge as a probability distribution

The method represents the knowledge p_i (probability of success of treatment i) not by a single number corresponding for example as the fraction of successful days on past attempts, but by a probability density over p_i. We shall assume each p_i will follow a Beta distribution. Beta distributions are driven by two parameters often called a and b. We will consider the case where a and b are both strictly positive integers. Plot the distribution of Beta distribution using python for a=1 and b=1, then for a=1 and b=2, for a=1 and b=3, for a=2 and b=3, for a=5 and b=3, to see what they look like. Check that a=1 and b=1 corresponds to the distribution showing no knowledge about p_i, i.e. a uniform distribution. The python Beta distribution may be found in the scipy library. The `.pdf` function can help you plot the distribution.

A joint exploration-exploitation process

You should now implement the following algorithm as a strategy on the medical side. You can keep the patient side of your code as it is.

a) start with a=1 and b=1 for each of the 10 treatments, corresponding to the lack of initial knowledge.
b) then, for each of the 100 days, loop the following operations :
    a) for each of the 10 treatments, sample a random value from to its beta distribution. The `.rvs` function in scipy could help you.
    b) give the patient the treatment that gave the highest random value among the 10 in the previous step
    c) if the treatment is successful, increase parameter a by 1 for that treatment, otherwise increase parameter b by 1.

We suggest you plot how each of the 10 beta distributions evolve over successive days and check this maked sense with regard to the best and worse treatments.

Plot the number of successes over 100 days and compare with the technique you had previously proposed (overlay all graphs).

On the same graph, plot the number of successul days from 1 to 100 if every day, the most effective treatment is used. Obviously, this scenario cannot be a real case, since we don't know which is the most effective treatement, but is provides a nice upper bound against which the real-world solutions can be compared.

Considering the distributions of the beta distributions, show (text and/or pictures) that :
a) the algorithm selects treatments randomly but increases the probability of selecting a treatement as it performs well and confirms its efficiency
b) the algorithm progressively reduces the variance of each distribution as it learns about the degree of effectiveness of the corresponding treatment
c) the algorithm carries out both exploration and exploitation over the 100 days.