Instructions: Portfolio optimization

Aaltoes Advicer

November 7, 2024

1 Introduction

Given some information about the projects, we would like to know which projects are advisable to invest to. About each project p_i, we have next information : Cost of the project (c_i), Expected benefit of the project (b_i), Risk of failure of the project (r_i), Set of projects that p_i depends on (D_i). B is our budget.

Since each project has associated risk of failure, the problem can be formulated as stochastic one. You can imagine it as some alternative universes, where some project fail and some don't. We call these universes - scenarios.

How can we simulate this process? It is quite straightforward. For each of the projects, we generate number between 0 and 1 (from uniform distribution). If this number is greater than risk (r_i), then project succeeded. For example, risk of failure of me accepting to board is 10%, then there is 90% chance that generated number will be more than 10%. Let $S = N \times M$ matrix, where (N is number of projects, M is number of scenarios). Then if $S[1, 203] = 1$, it means that project 1 in scenario 203 succeeded. Decision maker sets preferred number of scenarios, but he/she just needs to make sure that it is enough. We say that each scenario is equally probable. If there are M scenarios, then probability of each is $1/M$.

It is not possible to select such set of projects that would be optimal for each individual scenario, it is possible only if decision maker has a "perfect information" about universe he lives in. Thus, in some of the scenarios, we have to choose failing projects, but in general for all scenarios, the solution will be optimal. When we choose to invest to failing project, we don't get benefit and loose money. I don't like that problem formulation uses "benefit" as a parameter. At least for stochastic case, it is better to work with profits since they are on the same scale with costs. So, I will optimize revenue (profit - cost). I will assume that there is some function "benefit-to-profit" that turns b_i to pr_i, which a separate topic of discussion.

2 Formulation

2.1 Easy formulation

As additional constraint I will add U - upperbound for number of projects.

Now, we are ready to present it as a integer programming problem.

$$\begin{aligned} \textbf{max} \quad & \sum_{j=1}^{M}\sum_{i=1}^{N} pr_i x_i S_{ij} - \sum_{i=1}^{N} c_i x_i \\ \text{s.t} \quad & \sum_{i=1}^{N} x_i \leq U \\ & \sum_{j \in D_i} x_j \geq |D_i| x_i, \quad \forall i \in \{1, \dots, |P|\} \\ & \sum_{i=1}^{N} x_i c_i \leq B \\ & x \in \{0,1\}^N \end{aligned}$$

where x_i is a binary variable denoting that we are investing in project i.

2.2 Extended formulation

Suppose, you want to have some control over situation. Yes, returned solution would be optimal if you are interested in maximizing expected revenue over all universes. But output solution can be risky. In some of the "universes", you can loose money and it can be a lot... Suppose you want to escape that with some guaranteed probability, so in most of the scenarios, you want to get at least some value T, but is allowed to violate this rule in some small number of cases by some number which is at most R. So, for each scenario we need to add this additional constraint that will work in $P\%$ of chances. Let $v \in \{0,1\}^M$ is a binary variable that is 1, when the constraint is violated and 0 when it is not violated. Let $u \in \mathbb{R}^M$, be the variable that allows us some violation but at most R.

$$\begin{aligned} \textbf{max} \quad & \sum_{j=1}^{M}\sum_{i=1}^{N} pr_i x_i S_{ij} - \sum_{i=1}^{N} c_i x_i \\ \text{s.t} \quad & \sum_{i=1}^{N} x_i \leq U \\ & \sum_{j \in D_i} x_j \geq |D_i| x_i, \quad \forall i \in \{1, \dots, |P|\} \\ & \sum_{i=1}^{N} x_i c_i \leq B \\ & \sum_{i=1}^{N} pr_i x_i S_{ij} - \sum_{i=1}^{N} c_i x_i \geq T - u_j, \quad j \in \{1, \dots, M\} \\ & u_j \leq v_j R, \quad j \in \{1, \dots, M\} \\ & \frac{1}{M} \sum_{j=1}^{M} v_j \leq 1 - P \\ & x \in \{0,1\}^N, v \in \{0,1\}^M, u \geq 0 \in R^M \end{aligned}$$

Hm, we haven't discussed yet the way constraint about dependence of project on each other. It is relatively easy. If project x_i is selected, then sum of projects on which it depends one is forced to greater or equal to $|D_i|$, which is possible only in case all of these projects are selected. If x_i is 0, then sum of this projects is greater or equal to 0, which is always true, since projects are binary variable.

2.3 Possible ideas for formulation

The model can be further extended if we were provided not expected benefit, but some distribution of benefits. For example, profit of each project can be normally distributed with some mean value and standard deviation. Then, we could easily add it to our model by changing pr_i to pr_i^j (meaning profit in scenario/universe j).

Also, it is possible to reformulate it as a multistage optimization problem, since projects progree evolve in time. There is a good paper about it, which I can share.

3 Analysis

(a) Given, we optimized the model and got some solution vector x. We can have a look at revenues that we got in each of the universes and plot it as a histogram. By looking at the plot, we will be able to notice if there are some cases where are revenues go to 0. And if it happens, we can use extended formulation and play with parameters T, R (note if want to allow negative numbers R should be bigger than T). The meaning of this parameters would be as follows: "With $P\%$ chance we will have revenue greater than T, but in $1-P\%$" cases we can have revenue less than R but guaranteed to be at least $T - R$.

Be careful, the model may fail to find such solution.

(b) We can also, calculate expected value of perfect information. It is important for example when you deal with an expert in the field and he/she claims that knows whar future to expect. Then, expected value of perfect information would be amount of money that you can pay him/her at most and if his information is perfect. To analyze this number, we just need to solve original easy formulation for each of the scenario separately to get maximum revenue for each case:

$$\begin{aligned}
\textbf{max} \quad & \sum_{i=1}^{N} pr_i x_i S_{ij} + \sum_{i=1}^{N} c_i x_i \\
\text{s.t} \quad & \sum_{i=1}^{N} x_i \leq U \\
& \sum_{j \in D_i} x_j \geq |D_i| x_i, \quad \forall i \in 1, \ldots, |P| \\
& \sum_{i=1}^{N} x_i c_i \leq B \\
& x \in \{0, 1\}^N
\end{aligned}$$

Then, we need to subtract these revenues from our vector of revenues (that we got from original problem) and find expected value (take mean).

(c) There is also one more metric - price of stochastic solution: solve problem only for 1 one scenario ⇒ use this solution to calculate revenue in all scenarios and calculate difference in revenues with original vector.

(d) Once we got our solution vector, we can test it. We just need to generate more scenarios that model hasn't used in optimization and see if the revenues have the same distribution.

4 How to use?

You need to have .xslx' (multiples tabs are allowed) or '.csv' that have next column names: 'project', 'benefit', 'cost', 'risk', 'dependence'. Please make sure that 'dependence' column makes sense. Possible inputs are '{1,2,3}' (no white spaces), '{}' or empty field. 'project' - can be either name or index (doesn't matter).

Given files in the folder, it should turn all tab from '.xslx' to separate '.csv' files and for each '.csv' solve optimization problem. At the end if parameter 'analysis' was put to 'true', you will receive a full report automatically generated in LaTeXthat you can analyze separately (it has evaluation of the solution (on test scenarios), expected value of perfect information, value of stochastic solution). Everything that you put in the folder will be turned to 1 report.

5 User interface

Here is the interesting part. I implemented a telegram bot that handles the same process with you. It uses a lovely character as an avatar and really friendly. You can expect jokes, stickers and of course help with optimization of portfolio.

She provides a detailed assistance about the files you are submitting along with all required instructions and examples. She requests parameters for the model and in the end generates a lovely report about found solution.

The bot has a next link.

Source code