Single Resource Capacity Allocation

Part 2

The Many-Class Problem

We next consider the general case of $n > 2$ classes and stages.

We again assume that the demand arrives one class at a time.

During the first stage we receive requests for seats at price p_1, during the second at price $p_2 > p_1$, and so on till stage n with the maximum price p_n.

Littlewood's Rule is not easily extendable to the general case.

Applying the same logic, we would build a huge decision tree.

In the 2-stage case, in the second stage we can either sell a seat with a certain probability or loose it. In the *n*-stage case, we can sell it with a certain probability or keep it for the third stage, where it can be sold with a certain probability or …

We have to take into account *conditional probability*, which leads us to complex and unmanageable decision trees. The next picture shows a 3-stage decision tree.

Luckily, we have at disposition a very powerful method to state this problem in an elegant and compact way.

With a few equations we represent all stages of the problem, summarizing what we could say with big trees.

This method is *Dynamic Programming*, a venerable 60-years old methodology applicable to a vast realm of optimization problems.

The name is somewhat old-fashioned: nowadays we would use the term *Planning* instead of *Programming*.

Let us formalize the problem.

We have *n* stages with *n* prices $p_1 < p_2 < \dots < p_{n-1} < p_n$ and *n* unknown demands *D1, …, Dn*, for which we have a forecasting methods (not explained here).

We want to maximize the revenue over the whole cycle of *n* stages.

Let s_t be the number of seats sold at stage *t*.

Our problem can be mathematically stated as

$$\max \sum_{t=1}^{n} s_t p_t$$

subject to

$$\sum_{t=1}^{n} s_t \leq C$$

$$\max \sum_{t=1}^{n} s_t p_t$$

subject to

$$\sum_{t=1}^{n} s_t \leq C$$

The first block denotes the objective: maximizing the sum of every daily revenue, each one expressed as product of quantity sold and price.

The second block denote a constraint: the overall quantity sold cannot exceed the available capacity (the number of seats).

In this formulation, s_t for $t = 1, \ldots, n$ are the *decision variables*, also known as *control variables*.

$$\max \sum_{t=1}^{n} s_t p_t$$

subject to

$$\sum_{t=1}^{n} s_t \leq C$$

To better understand the problem, let us describe the decision process. At each stage t the following sequence of events occurs:

1. We have number x_t of available seats, i.e. the remaining capacity. It is $x_t = C - \sum_{i=1}^{t-1} s_i$, capacity minus quantity already sold.
2. We observe a demand D_t.
3. We decide on a quantity u_t of this demand to accept. To be meaningful, it must be $0 \leq u_t$, $u_t \leq D_t$ and $u_t \leq x_t$.
4. We collect revenue $p_t u_t$
5. We proceed to stage $t + 1$ with a remaining capacity $x_{t+1} = x_t - u_t$.

Be careful: control variable u_t are really decided *before* observing demand. We are pretending we set them *after*. We can do it because the optimal choice of control variables does not actually require knowledge of demand. We are going to see this point.

This is a key point to understand the whole procedure.

$$\max \sum_{t=1}^{n} s_t p_t$$

subject to

$$\sum_{t=1}^{n} s_t \leq C$$

Control variables s_t are not very comfortable to manage, because what we really set are the u_t variables. So, we express them as

$$s_t = \min\{D_t, x_t, u_t\}$$

We are saying that the number of seats really sold has both demand and remaining capacity as upper bounds at each time.

$$\max \sum_{t=1}^{n} s_t p_t$$

subject to

$$\sum_{t=1}^{n} s_t \leq C$$

We rewrite the problem in a more explicit form:

$$\max \sum_{t=1}^{n} p_t u_t$$

subject to

$$\sum_{t=1}^{n} u_t \leq C$$

$$0 \leq u_t \leq \min\{D_t, x_t\}$$

The Dynamic Programming Formulation

The key idea of the Dynamic programming methodology is to reduce the *n*-stages problem to a 2-stages problem, where the new second stage aggregate all original stages but the first.

Imagine we have 4 stages.

The original problem is to maximize the revenue over 4 stages.

We solve a modified problem:

Maximize the sum of 1st stage revenue and all next revenue from 2nd to 4th.

The difference is subtle: the first stage has single and well defined demand D_1 but the "next" stages has no demand, at least not in the obvious meaning.

The revenue for stages 2-to-4 has to be computed solving a second problem which is of the same form, but covering 3 stages instead of 4.

The second problem is:

Maximize the sum of 2nd stage revenue and all next revenue from 3rd to 4th.

Then we have to solve a third problem:

Maximize the sum of 3rd stage revenue and all next revenue from 4th to 4th.

Then we have to solve a fourth problem:

Maximize the sum of 4th stage revenue and all next revenue, which do not exist.

The last problem is easy to solve: sell every possible seat, without denying any booking. Otherwise, the plane will depart with some seat empty which could have been sold.

Once the fourth problem is solved, the third is easy, because it is a 2-stages problem, and we con solve it with Littlewood's rule.

Once solved the third problem, it is easy to solve the second for the same reason, then the first problem.

The whole procedure is not intuitive, yet extremely powerful.

It is worth studying carefully what follows, because this is a formidable tool for optimization.

Let us introduce a new key concept, the *value function* $V_t(x)$.

It represents *the expected revenue of having x seats available at time t under the assumption we will use them in the optimal way*.

Let us have capacity $C = 100$. At stage 1 we want to manage these 100 seats at the best. If we will really succeed in doing it, we will get revenue $V_1(100)$. This is a definition: we do not know what $V_1(100)$ is. It is the revenue we are trying to get, choosing the right u_t at each stage.

If 5 is the best choice for the control variable u_t at time 1, then we can write:

$$V_1(100) = 5p_1 + V_2(95)$$

We know neither $V_1(100)$ nor $V_2(95)$, yet we know this is true, under the hypothesis that 5 is really the optimal number of seats to make available for sale at time 1.

The best action at time 1 is to sell 5 seats. The best course of action form time 2 on will collect revenue $V_2(95)$. Consequently, the best course of action at time 1 will collect $5p_1 + V_2(95)$.

Note that u_t is supposed to be less than the demand D_t at time t. So, we sell exactly u_t seats because enough customers ask for them. Number 5 is not the best choice in absolute, but that demand given.

We do not really know in advance what is the best value for u_t.

Let us rewrite the equation as relationship among unknown quantities:

$$V_t(x) = \underset{D_t}{\mathrm{E}}\left[\max_{0 \le u \le \min\{D_t, x\}} \{p_t u + V_{t+1}(x - u)\}\right]$$

For each possible value of demand at time *t*, we compute the expected revenue if we maximize the revenue under that scenario. Then we weight each scenario revenue with the scenario probability. The weighted sum is the expected value at time *t* if we have a capacity of *x* seats available.

Maximizing the revenue means to choose the best possible *u*, compatibly with capacity and demand at time *t*.

The recursive form of equation needs a final condition stopping the recursion.

We introduce a stage $n + 1$ when each seat has a zero value, because the plane has already departed:

$V_{n+1}(x) = 0$

whichever x is.

Now we have two equations capable of express the maximum revenue attainable with every possible capacity along a decision process of any possible length.

Note that the rigorous form of equation should be max E[·] instead of E[max{·}]. Yet we can use our form because it is simpler and *almost* equivalent for practical purposes.

The Insight Behind

When we decide whether to make a seat immediately available for sale we have to balance between

1. the present revenue we gain if it is available, i.e. the present price, and
2. the future revenue we can gain if it is not immediately available for sale and we will follow an optimal strategy in the future.

One remark is that optimality of our present decision depends on optimality of our future behavior.

Another fundamental remark is that the decision process goes backward: when planning, we start from the last decision step and move toward the first one, with a temporal inversion.

Optimal Policy

We define *expected marginal value of capacity* as

$$\Delta V_t(x) = V_t(x) - V_t(x-1)$$

the expected revenue of the *x-th* seat.

It has to properties:

1. $\Delta V_t(x+1) \leq \Delta V_t(x)$ (decreasing in capacity)
2. $\Delta V_{t+1}(x) \leq \Delta V_t(x)$ (decreasing in time)

Intuitive support for these properties:

1. the second seat is less worth than the first because it is more subject to risk of remaining unsold;
2. while time is passing, you have less options for your strategies, which becomes less effective.

Now we can write:

$$V_t(x) = V_{t+1}(x) + ED\left[\max_{0 \le u \le \min\{D_t, x\}} \left\{\sum_{z=1}^{u} (p_t - \Delta V_{t+1}(x+1-z))\right\}\right]$$

Because V is decreasing in capacity, the term $p_t - \Delta V_{t+1}(x+1-z)$ is decreasing in *z*.

Thus, it is optimal to increase *u* until that term is negative or the upper bound $min\{D_t, x\}$ is reached, whichever comes first.

Now we have the optimal protection level:

$$y_t^{*} = \max\{x | p_{t-1} < \Delta V_{t-1}(x)\}$$