Revenue Management Problem Initial Thoughts

Dawson Ren

November 14th, 2022

1 Mathematics

Given m nurses and n shifts, we have:

- $P \in \mathbb{R}^{m \times n}$, the probability that nurse i will select shift j. Note that the probabilities are all independent.
- $Q \in \mathbb{R}^{m \times n}$, the probability that nurse i will show up to shift j.
- $r \in \mathbb{R}^n$, the revenue for covering shift j.
- $Y \in \{0,1\}^{m \times n}$, whether or not we will show nurse i shift j.

We express the expected revenue using a recurrence relation on i, which nurse we are currently scheduling, and Z, the availability matrix. Our goal is to find $R(1, Y)$. The base cases are when row i of Z is a vector of 0 or $i > m$. We will be using one-indexed values.

$$R(i, Z) = \sum_{c \in \{0,1\}^m} \left(\prod_{j=1}^{n} f(c_j, P_{ij}) \right) \cdot \left[\left(\sum_{j=1}^{n} Q_{ij} r_j c_j \right) + R(i+1, g(Z, c)) \right]$$

We define:

$$f(c_j, p) = \begin{cases} (1-p) & c_j = 0 \\ p & c_j = 1 \end{cases}$$

The function $g(Z, c)$ returns another matrix $\tilde{Z}$ such that the columns in Z where $c = 1$ are filled with zeros.

2 Programming

We use Numpy to perform efficient matrix computations. Many operations can be vectorized, which will be an important source of efficiency gains. We use dynamic programming to eliminate additional work. For example, consider a case where we have three nurses and three shifts. If nurse 1 schedules shift

1, and nurse 2 schedules shift 2, nurse 3 can only schedule shift 3. However, it's also possible that nurse 1 scheduled shift 2 and nurse 2 scheduled shift 1, leading to the same possible outcomes for nurse 3. Therefore, we use caching as we recurse downwards, storing the matrix as a string for efficient storage and comparisons.