Optimizing Guided Counts for Maximum Lost Sales Reduction Using Linear Programming

Problem Formulation

We address the challenge of selecting top candidates for Guided Counts lists in such a way that we maximize the benefit (lost sales reduction) that we can achieve from recounting and correcting an item's inventory level.

Decision Variables

Define $x_{i,d}$ as a binary decision variable where:

$$x_{i,d} = \begin{cases} 1 & \text{if item } i \text{ is recounted on day } d, \\ 0 & \text{otherwise.} \end{cases}$$

These variables indicate whether an item i is recounted on day d.

Objective Function

The objective function is formulated to maximize the total benefit:

$$\text{Maximize} \sum_{i=1}^{n} \sum_{d=1}^{4} B_{i,d} \cdot x_{i,d}$$

where $B_{i,d}$ represents the benefit of recounting item i on day d.

Constraints

The model includes the following constraints:

1. Each item is recounted at most once over the four days:

$$\sum_{d=1}^{4} x_{i,d} \leq 1 \quad \forall i$$

2. No more than L items are recounted on any given day:

$$\sum_{i=1}^{n} x_{i,d} \leq L \quad \forall d$$

Additionally, the decision variables $x_{i,d}$ are binary.

Data Representation and Constraint Matrices

Benefit Matrix

We define the benefit $B_{i,d}$ of recounting an item x_i on a particular day d_i, as the difference between the lost sales if the item was not recounted at all and the lost sales if the item recounted on d_i. The benefits associated with recounting each item on each day are represented in a matrix B of size $n \times 4$. For instance, for 3 items and 4 days, B might look like:

$$B = \begin{pmatrix} B_{1,1} & B_{1,2} & B_{1,3} & B_{1,4} \\ B_{2,1} & B_{2,2} & B_{2,3} & B_{2,4} \\ B_{3,1} & B_{3,2} & B_{3,3} & B_{3,4} \end{pmatrix}$$

Constraint Matrix Representation

Constraints are represented in a structured matrix form.

Single Counting Constraint Matrix

For the single counting constraint, the matrix has n rows (one for each item), with each row having a '1' in the columns corresponding to the four days for that item, and '0' elsewhere. This enforces that the sum of these variables is at most 1. The matrix for 3 items looks like:

$$\begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix}$$

Each row represents an item, and the '1's in a row correspond to that item being recounted on different days.

Daily Limit Constraint Matrix

For the daily limit constraint, there are 4 additional rows (one for each day). Each of these rows has '1's in the columns corresponding to all items for that specific day, ensuring the sum does not exceed L. The matrix part for a daily limit L and 3 items is:

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \end{pmatrix}$$

Each row corresponds to a day, and the '1's in a row correspond to different items being recounted on that day.

Understanding the HiGHS Package in R

The HiGHS package in R is an interface to the High-Performance Software for Linear Optimization (HiGHS) solvers. It efficiently solves large-scale linear programming (LP) and mixed-integer programming (MIP) problems.

Working Mechanism

HiGHS utilizes advanced algorithmic strategies to solve optimization problems. For LP problems, it implements dual and primal simplex methods, and for MIP problems, it uses a branch-and-cut algorithm. These methods are designed to handle the intricacies of linear and integer constraints efficiently.

Inputs Required by the Solver

To solve an optimization problem using HiGHS, the following inputs are necessary:

1. **Coefficient Matrix (A)**: A matrix representing the coefficients of the constraints. Each row corresponds to a constraint, and each column corresponds to a decision variable.
2. **Objective Function Coefficients**: A vector representing the coefficients in the objective function.
3. **Right-Hand Side (RHS) Vector**: A vector specifying the RHS values of each constraint.
4. **Constraint Types**: A vector indicating the type of each constraint (e.g., '$\leq$', '$=$', '$\geq$').
5. **Variable Types**: For MIP problems, a vector indicating whether each decision variable is continuous, integer, or binary.

Mapping Problem Inputs to HiGHS Inputs

In the context of our inventory recount optimization problem, the inputs to the HiGHS solver can be mapped as follows:

1. **Coefficient Matrix (A)**: This matrix is constructed based on the constraints of our problem. It consists of two parts:
   (a) Rows for single counting constraints, where each row has '1's in columns corresponding to the same item on different days, ensuring an item is recounted at most once.
   (b) Rows for daily limit constraints, where each row has '1's in columns corresponding to all items for a specific day, ensuring no more than L items are recounted each day.
2. **Objective Function Coefficients**: This is a flattened version of the benefit matrix B, which represents the benefit of recounting each item on each day.
3. **Right-Hand Side (RHS) Vector**: This vector combines the limits from both sets of constraints:
   (a) A sequence of '1's for the single counting constraints (one for each item).
   (b) The value L repeated for the number of days, representing the daily limit on recounted items.
4. **Constraint Types**: All constraints in this problem are of the '$\leq$' type.
5. **Variable Types**: Since the decision variables $x_{i,d}$ are binary (representing whether an item is recounted on a certain day), they are defined as binary in the HiGHS model.

Extension to the Model: Departmental Constraints

To ensure a diverse recount strategy, we introduce an additional constraint that items from a single department cannot make up more than 10% of the entire list L on any given day.

Formulation of Departmental Constraint

Let's assume there are m departments. The constraint for each department k on day d can be expressed as:

$$\sum_{i=1}^{n} d_{i,k} \cdot x_{i,d} \leq 0.1 \cdot L$$

where $d_{i,k}$ is a binary parameter that is 1 if item i belongs to department k, and 0 otherwise. This constraint ensures that no more than 10% of recounted items on any day come from the same department.

Impact on Constraint Matrix

The constraint matrix A will have additional rows to represent these departmental constraints. For each department and each day, a new row will be added. These rows will have '1's in columns corresponding to items from the relevant department and '0's elsewhere.

Extended Constraint Matrix for Departmental Constraints

In the scenario with 3 items and 4 days, we now include departmental constraints. Assuming two departments, where the first item belongs to department 1 and the other two items belong to department 2, the constraint matrix A is extended as follows:

Single Counting Constraints:

$$\begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix}$$

Daily Limit Constraints:

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \end{pmatrix}$$

Departmental Constraints:

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \end{pmatrix}$$

Right-Hand Side (RHS) and Signs

The RHS for these constraints is set to 10% of L, ensuring that the sum of items from a single department does not exceed this limit. If L is 150, the RHS value for each departmental constraint will be $0.1 \times 150 = 15$. The sign for these constraints is '$\leq$'.