MGMTMSA 408 – Operations Analytics

Homework 2 – Constraint Generation and Assortment Optimization (Question Sheet)

Due May 6, 2024 (Section 2) / May 8, 2024 (Section 1) at 1:00pm PST

1 Learning Coffee Preferences

In this problem, we will be using column / constraint generation to learn about customer preferences through a *discrete choice model*. In a discrete choice model, there are n distinct products, indexed from 1 to n, as well as a dummy product 0, which represents the outside or no-purchase option (i.e., the customer chooses not to buy any of the products from the assortment). When a customer makes a decision, they choose a product from within the assortment or they choose the no-purchase option. The specific type of discrete choice model we will use is the *ranking-based choice model*, which will be explained in more detail in Part 1 and Part 2 of this problem.

The data set that we have consists of different assortments of coffee brands, which are listed below. For each assortment, we observe how many customers purchased each of those brands, as well as how many customers did not purchase anything when they considered the assortment. The data will be described in more detail in Part 3.

Product #	Brand
1	Peets
2	Folgers
3	Starbucks
4	Philz
5	Blue Bottle
6	Intelligentsia
7	Counter Culture
8	Stumptown
9	Groundworks

After introducing the data in Part 3, we will focus on estimating the ranking-based model from data by solving a linear program. Part 4 introduces this linear program and walks through solving it with a fixed set of rankings. Part 5 then involves applying the column randomization approach, while Part 6 involves using the column generation / constraint generation approach to solving the problem. Finally, Part 7 looks at understanding the ranking model obtained from Part 6. A Jupyter Notebook containing code snippets and skeleton code, `HW2 - Coffee Code.ipynb`, is provided to help you with this problem.

Part 1: Modeling preferences through rankings

In the ranking-based model, we assume that the customer population consists of K different customer types. We let k denote the index of the customer type, which ranges from 1 to K.

Each customer type k chooses according to a ranking σ_k of the products.

As an example, suppose that $n = 9$. Suppose that we have three customer types, which are given by the rankings

$$\sigma_1 = 4 \prec 3 \prec 0 \prec 1 \prec 5 \prec 6 \prec 8 \prec 7 \prec 9 \prec 2, \tag{1}$$
$$\sigma_2 = 3 \prec 2 \prec 4 \prec 0 \prec 1 \prec 5 \prec 6 \prec 8 \prec 7 \prec 9, \tag{2}$$
$$\sigma_3 = 2 \prec 5 \prec 0 \prec 1 \prec 4 \prec 3 \prec 6 \prec 8 \prec 7 \prec 9, \tag{3}$$

where the notation $i \prec j$ means that we prefer option i to option j.

The first ranking σ_1 represents customers who prefer product 4 the most, followed by product 3, followed by product 0 (the no-purchase option), then followed by the remaining products. The second ranking σ_2 represents customers who prefer product 3 the most, followed by product 2, followed by product 4, followed by the no-purchase option, then followed by the remaining products. The third ranking σ_3 represents customers who prefer product 2 the most, followed by product 5, followed by the no-purchase option, followed by all of the other products.

Each customer type k also has a probability λ_k associated with it; we can think of λ_k as the probability that a randomly drawn customer will be of type k, or equivalently, what fraction of customers in the population are of type k (i.e., they will choose according to the ranking σ_k).

Suppose that the probabilities of the three rankings are $\lambda_1 = 0.1$, $\lambda_2 = 0.3$, $\lambda_3 = 0.6$. Then the choice probabilities of the products are:

$$\begin{aligned}
P(1|\{1,4,5\}) &= 0 \\
P(4|\{1,4,5\}) &= \lambda_1 + \lambda_2 \\
&= 0.1 + 0.3 \\
&= 0.4 \\
P(5|\{1,4,5\}) &= \lambda_3 \\
&= 0.6 \\
P(0|\{1,4,5\}) &= 0
\end{aligned}$$

In this example, customer types 1 and 2 both choose product 4, because it is their most preferred option out of products 1, 4 and 5 and the no-purchase option 0. As a result, the probability that a random customer chooses product 4 is 0.4. Customer type 3 chooses product 5, because it is their most preferred option. Therefore, the probability that a random customer chooses product 5 is 0.6. None of the three customer types choose either product 1 or the no-purchase option, and hence their choice probabilities are 0.

a) For the same rankings given in the example, calculate the choice probabilities for the set of products $\{3, 4\}$.

b) For the same rankings given in the example, calculate the choice probabilities for the set of products $\{2, 3, 4\}$.

Part 2: Choice probabilities as a linear system of equations

Given K customer types with rankings $\sigma_1, \ldots, \sigma_K$ and their associated probabilities $\lambda_1, \ldots, \lambda_K$, the choice probability of a product i when a customer is offered the set S is given by

$$P(i \mid S) = \sum_{k=1}^{K} \mathbb{I}\{\text{option } i \text{ is most preferred out of } S \cup \{0\} \text{ according to } \sigma_k\} \cdot \lambda_k, \tag{4}$$

where $\mathbb{I}\{\cdot\}$ is the indicator function: $\mathbb{I}\{A\} = 1$ if condition A is true, and $\mathbb{I}\{A\} = 0$ if it is false.

Suppose now that we have M assortments $S_1, \ldots, S_M$. Let $\mathbf{A}$ be a matrix where each row corresponds to a pair (m, i), where $i \in \{0, 1, 2, \ldots, n\}$ and $m \in \{1, \ldots, M\}$, and each column corresponds to a customer type $k \in \{1, \ldots, K\}$. Each entry of this matrix is defined as

$$A_{(m,i),k} = \mathbb{I}\{\text{option } i \text{ is most preferred out of } S_m \cup \{0\} \text{ according to } \sigma_k\} \tag{5}$$

Let $\mathbf{v}$ be a vector where each entry corresponds to a pair (m, i), where $i \in \{0, 1, 2, \ldots, n\}$ and $m \in \{1, \ldots, M\}$. The vector $\mathbf{v}$ represents the vector of choice probabilities. Finally, let $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_K)$ denote the vector of customer type probabilities. Then the relationship between $\boldsymbol{\lambda}$, $\mathbf{A}$ and $\mathbf{v}$ is given by the following linear system of equations:

$$\mathbf{A}\boldsymbol{\lambda} = \mathbf{v}. \tag{6}$$

a) For the $M = 3$ assortments in Part 1 ($S_1 = \{1, 4, 5\}, S_2 = \{3, 4\}, S_3 = \{2, 3, 4\}$), and for the $K = 3$ customer types (rankings) given in Part 1, write down the $\mathbf{A}$ matrix. Your matrix should have 30 rows and 3 columns.

   Hint: if you are having difficulty with this problem, take a look at the function `permToA` provided in the skeleton code.

b) Using this matrix, compute $\mathbf{v} = \mathbf{A}\boldsymbol{\lambda}$, where $\boldsymbol{\lambda} = (0.1, 0.3, 0.6)$. What do you get for $\mathbf{v}$?

Part 3: Understanding the data

We will now introduce the two data sets provided for this problem:

- `coffee_assortments.csv`: This CSV contains a matrix with $M = 47$ rows and $n + 1 = 10$ columns. Each row corresponds to a distinct assortment of brands offered by the retailer. Each column corresponds to one of the $n = 9$ products or the no-purchase option. (Note that the no-purchase option is represented as the last / 10th column.) For example, the first row in the CSV is

  ```
  0, 1, 0, 0, 0, 0, 1, 0, 0, 1
  ```

  which implies that $S_1 = \{2, 7\}$, i.e., Folgers and Counter Culture were offered. (Note that the last column is always a 1, as the no-purchase option is always present.)

- `coffee_transaction_counts`: This CSV contains a matrix with $M = 47$ rows and $n + 1 = 10$ columns. The rows and columns have the same meaning as in `coffee_assortments.csv`. Each entry indicates the number of customers who purchased the corresponding option, when that assortment was offered. For example, for the first assortment (the first row), the value of 251 for column 2 indicates that when this assortment was offered, 251 customers bought product 2 (i.e., Folgers). Similarly, the value of 327 in the last column indicates that 327 customers considered the assortment but chose not to purchase anything.

a) Which assortment has the most transactions?

b) Which assortment has the least transactions?

c) How many assortments include the Starbucks brand?

Convert the matrix in `coffee_transaction_counts.csv` into an array of choice probabilities, i.e., divide each row by its row sum. If you performed this step correctly, then the choice probability of the no-purchase option for the first assortment should be 0.450413.

d) For assortment $m = 10$, what fraction of customers purchased the Starbucks brand?

e) What fraction of customers offered assortment $m = 3$ chose to not purchase anything?

Part 4: Estimating the ranking-based model

In this section, we will formulate a problem for estimating the probability distribution $\boldsymbol{\lambda}$ for a given set of customer types, using available data.

Suppose that we have assortments $S_1, \ldots, S_M$, and customer types $k \in \{1, \ldots, K\}$. Let $\mathbf{v}$ now denote the vector of choice probabilities from historical data. Consider the following abstract estimation problem:

$$\begin{aligned} \mathsf{EST}\text{-}\ell_1 : \text{minimize} \quad & \|\mathbf{A}\boldsymbol{\lambda} - \mathbf{v}\|_1 && \text{(7a)} \\ \text{subject to} \quad & \sum_{k=1}^{K} \lambda_k = 1, && \text{(7b)} \\ & \lambda_k \geq 0, \quad \forall k \in \{1, \ldots, K\}, && \text{(7c)} \end{aligned}$$

where $\|\cdot\|_1$ is the ℓ_1 norm, i.e., $\|\mathbf{x}\|_1 = \sum_i |x_i|$ for any vector $\mathbf{x}$.

This problem seeks to find a probability distribution $\boldsymbol{\lambda}$, such that $\mathbf{A}\boldsymbol{\lambda}$ is as close to possible to the choice probabilities $\mathbf{v}$ from historical data. Remember that $\mathbf{A}\boldsymbol{\lambda}$ is the vector of predicted choice probabilities. This problem is similar to the least squares problem that is solved when one uses linear regression; the difference is that errors are measured in terms of ℓ_1 norm (sum of absolute values), instead of ℓ_2 norm (square root of sum of squares).

This problem is technically not a linear program because of the presence of the ℓ_1 norm which involves absolute values. However, it can be turned into a linear program. Let $\boldsymbol{\epsilon}^+$ and $\boldsymbol{\epsilon}^-$ be vectors of nonnegative decision variables, where each variable in each vector corresponds to a (m, i) pair for $i \in \{0, 1, \ldots, n\}$ and $m \in \{1, \ldots, M\}$. Then the problem can be written as

$$\begin{aligned} \mathsf{EST} : \text{minimize} \quad & \sum_{m=1}^{M} \sum_{i=0}^{n} \epsilon^+_{m,i} + \sum_{m=1}^{M} \sum_{i=0}^{n} \epsilon^-_{m,i} && \text{(8a)} \\ \text{subject to} \quad & \sum_{k=1}^{K} A_{(m,i),k} \lambda_k - \epsilon^+_{m,i} + \epsilon^-_{m,i} = v_{m,i}, \quad \forall i \in \{0, \ldots, n\}, m \in \{1, \ldots, M\}, && \text{(8b)} \\ & \sum_{k=1}^{K} \lambda_k = 1, && \text{(8c)} \\ & \lambda_k \geq 0, \quad \forall k \in \{1, \ldots, K\}, && \text{(8d)} \\ & \epsilon^+_{m,i} \geq 0, \quad \forall i \in \{0, 1, \ldots, n\}, m \in \{1, \ldots, M\}, && \text{(8e)} \\ & \epsilon^-_{m,i} \geq 0, \quad \forall i \in \{0, 1, \ldots, n\}, m \in \{1, \ldots, M\}. && \text{(8f)} \end{aligned}$$

a) Explain why solving problem EST is equivalent to solving the nonlinear problem EST-ℓ_1. (*Hint*: suppose that we have an optimal solution $(\boldsymbol{\lambda}, \boldsymbol{\epsilon}^+, \boldsymbol{\epsilon}^-)$ of problem EST. At optimality, what can you say about what the values of $\epsilon^+_{m,i}$ and $\epsilon^-_{m,i}$ will be? Is it possible for both $\epsilon^+_{m,i}$ and $\epsilon^-_{m,i}$ to be non-zero?)

b) For the coffee data set, suppose that our rankings are:

$$\begin{aligned}
\sigma_1 &= 1 \prec 2 \prec 7 \prec 10 \prec 6 \prec 8 \prec 9 \prec 3 \prec 5 \prec 4 \\
\sigma_2 &= 8 \prec 9 \prec 5 \prec 7 \prec 3 \prec 6 \prec 4 \prec 1 \prec 2 \prec 10 \\
\sigma_3 &= 2 \prec 6 \prec 9 \prec 7 \prec 4 \prec 1 \prec 5 \prec 3 \prec 10 \prec 8 \\
\sigma_4 &= 6 \prec 3 \prec 2 \prec 1 \prec 10 \prec 5 \prec 4 \prec 7 \prec 9 \prec 8 \\
\sigma_5 &= 5 \prec 2 \prec 4 \prec 7 \prec 8 \prec 1 \prec 6 \prec 10 \prec 3 \prec 9 \\
\sigma_6 &= 9 \prec 2 \prec 3 \prec 7 \prec 5 \prec 1 \prec 8 \prec 4 \prec 6 \prec 10
\end{aligned}$$

Formulate and solve problem EST above for these six rankings. What is the optimal objective? What is the optimal solution $(\boldsymbol{\lambda}, \boldsymbol{\epsilon}^{+}, \epsilon^{-})$?

Hint: The skeleton code includes a function `permToA` which given an input ranking σ, returns the column $\mathbf{A}_\sigma$ of the $\mathbf{A}$ matrix corresponding to this ranking. (Note that $\mathbf{A}_\sigma$ is represented as a 2D array, rather than a 1D array.) You may find this function useful in formulating your LP.

Part 5: A random sampling approach

A limitation of the LP we solved in Part 4 (b) is that the rankings given there were chosen arbitrarily. In principle, we can choose any set of rankings to model our data. In this section, we will attempt to use the column randomization approach described in class to solve the problem.

a) Set your random seed to 200 and randomly generate 1000 rankings. Solve problem EST with these 1000 rankings. What is your optimal objective value?

Hint: You may find it helpful to use the numpy function `np.random.permutation()`, where `np` is the numpy module. When called as `np.random.permutation(n+1)`, this function returns the numbers 0,1,…,n in a random order / permutation. This random permutation can then serve as a ranking of the options. You can then use the `permToA` function to create the corresponding column of the $\mathbf{A}$ matrix.

Part 6: A column generation approach

Let's now consider how to apply column / constraint generation to problem EST. The advantage of doing so is that we will determine both the rankings and the probability distribution $\boldsymbol{\lambda}$ over these rankings.

First, let us take the dual of EST:

$$\begin{aligned}
\mathsf{EST\text{-}DUAL} : \text{maximize} \quad & \sum_{m=1}^{M} \sum_{i=0}^{n} v_{m,i} \cdot p_{m,i} && \text{(9a)} \\
\text{subject to} \quad & \sum_{m=1}^{M} \sum_{i=0}^{n} A_{(m,i),k} p_{m,i} + q \leq 0, \quad \forall k \in \{1, \ldots, K\}, && \text{(9b)} \\
& -p_{m,i} \leq 1, \quad \forall i \in \{0, \ldots, n\},\ m \in \{1, \ldots, M\}, && \text{(9c)} \\
& p_{m,i} \leq 1, \quad \forall i \in \{0, \ldots, n\},\ m \in \{1, \ldots, M\}. && \text{(9d)}
\end{aligned}$$

In this dual problem, constraint (9b) is a family of linear constraints, with one constraint for each ranking. Theoretically, we could solve this by generating all possible rankings, but this will be too cumbersome computationally.

a) How many possible rankings are there of the $n+1$ options? (*Note*: in your answer, you can ignore the equivalence of rankings beyond the no-purchase option. For example, for $n = 4$, the two rankings

$$\sigma_1 = 1 \prec 2 \prec 3 \prec 0 \prec 4 \prec 5 \tag{10}$$

$$\sigma_2 = 1 \prec 2 \prec 3 \prec 0 \prec 5 \prec 4 \tag{11}$$

can be counted as distinct rankings, even though the ordering of products 4 and 5 does not matter since the no-purchase option is always available for the customer to choose.)

Instead, we will generate rankings one at a time. The separation problem for this constraint generation approach involves optimizing over a ranking σ. Specifically, let $\mathbf{A}_\sigma$ denote the column of the $\mathbf{A}$ matrix corresponding to ranking σ. Then the separation problem involves solving

$$\max_{\sigma} \sum_{m=1}^{M} \sum_{i=0}^{n} A_{(m,i),\sigma} p_{m,i} + q \tag{12}$$

and comparing this value to zero. If the optimal objective value of this separation problem is greater than zero, we have found a violated constraint, and we can add that constraint to the dual problem EST-DUAL. Otherwise, we conclude that the dual solution $(\mathbf{p}, q)$ satisfies all of the dual constraints and that it is therefore an optimal solution to the full dual problem with all possible rankings.

This separation problem can be formulated as the following integer program, where $A_{m,i}$ and $z_{i,j}$ are decision variables. Recall that S_m is the mth assortment, where m ranges from 1 to M.

$$\begin{aligned}
\mathsf{SEP}: \text{maximize} \quad & \sum_{m=1}^{M} \sum_{i=0}^{n} p_{m,i} A_{m,i} + q && \text{(13a)}\\
\text{subject to} \quad & z_{i,j} + z_{j,i} = 1, \quad \forall i,j \in \{0,1,\ldots,n\}, i \neq j, && \text{(13b)}\\
& z_{i,i} = 0, \quad \forall i \in \{0,\ldots,n\}, && \text{(13c)}\\
& z_{i,j} + z_{j,k} + z_{k,i} \leq 2, \quad \forall i,j,k \in \{0,1,\ldots,n\}, i \neq j, i \neq k, j \neq k, && \text{(13d)}\\
& A_{m,i} = 0, \quad \forall i \in \{1,\ldots,n\} \text{ such that } i \notin S_m,\ m \in \{1,\ldots,M\}, && \text{(13e)}\\
& A_{m,i} \leq z_{i,j}, \quad \forall j \in S_m \cup \{0\}, j \neq i,\ m \in \{1,\ldots,M\}, && \text{(13f)}\\
& \sum_{i=0}^{n} A_{m,i} = 1, \quad \forall m \in \{1,\ldots,M\}, && \text{(13g)}\\
& z_{i,j} \in \{0,1\}, \quad \forall i,j \in \{0,\ldots,n\}, && \text{(13h)}\\
& A_{m,i} \in \{0,1\}, \quad \forall i \in \{0,\ldots,n\},\ m \in \{1,\ldots,M\}. && \text{(13i)}
\end{aligned}$$

a) Explain the meaning of the $z_{i,j}$ decision variables in problem SEP.

b) Explain the meaning of each constraint in problem SEP.

c) Explain how to determine the ranking σ from the values of the $z_{i,j}$ variables.

Implement problem SEP in Python, and create a function called `separation` that solves SEP for two inputs, `p_val` and `q_val`, where `p_val` is a numpy array corresponding to $\mathbf{p}$, and tt q_val is a scalar variable corresponding to q. To check that you have implemented this function correctly, the skeleton code provides example values of `p_val` and `q_val` together with what the optimal objective value of SEP should be for those inputs. (In addition, ensure that your constraint generation procedure saves the ranking corresponding to each solution of SEP, as these will be needed for Part 7.)

Once you have implemented the `separation` function, implement the constraint generation procedure. Skeleton code for the procedure is provided in the accompanying Jupyter Notebook. Solve problem EST-DUAL using the constraint generation procedure. Use the six rankings provided in Part 4 as the initial rankings that your constraint generation procedure starts from.

d) What is the optimal objective value?

e) How many rankings are generated by your constraint generation procedure?

Part 7: Understanding the column generation solution

Solve the primal problem EST using the rankings generated by your constraint generation procedure.

a) How many rankings have a non-zero value of λ_k associated with them?

b) What are the top three rankings in terms of λ_k? For these three rankings, which are the top three most preferred options?

c) What is the average rank of each of the ten options (the $n = 9$ products plus the no-purchase option), where the average is taken with respect to the rankings and the probability distribution $\boldsymbol{\lambda}$ in your constraint generation solution? (Lower rank values correspond to more preferred options.)

d) Suppose that we offer the assortment $S = \{2, 4, 5, 6\}$. This corresponds to offering Folgers, Philz, Blue Bottle and Intelligentsia. What are the predicted choice probabilities for the products in this assortment? What is the predicted probability of customers not purchasing anything?

2 Designing a Sushi Menu

A high-end sushi restaurant is trying to decide what types of sushi to offer to its customers. The restaurant surveys $K = 500$ of its customers to obtain ratings of $n = 100$ different types of sushi. The ratings of the 500 customers for the 100 types of sushis are stored in the file `sushi_utilities_mat.csv` (columns are sushis, rows are customers). In addition, the restaurant also has additional information on the sushis: the file `sushi_info.csv` contains the name of each of the 100 sushi types (column 1), a category code for each sushi (column 2) and a price for each sushi (column 3). You may also find it helpful to consult the file `sushi_description.txt` which includes the name and a short description of each sushi.

We will assume that customers choose according to a first-choice model of choice, and that the utility of each option is the rating provided in `sushi_utilities_mat.csv`. Namely, each customer considers the available options, evaluates the utility of each option, and then selects the option that provides the highest utility. We will assume that the utility of the no-purchase option of each customer type is 3. So for example, suppose we offer sushi #2, #7 and #12. Let's fix customer #7. Customer #7 evaluates the utility of each of those options:

$$\begin{aligned}
&u_{7,2} = 3.985 \quad \text{(utility of sushi \#2 (= anago / sea eel))}\\
&u_{7,7} = 3.976 \quad \text{(utility of sushi \#7 (= ikura / salmon roe))}\\
&u_{7,12} = 3.530 \quad \text{(utility of sushi \#12 (= hotategai / scallop))}\\
&u_{7,0} = 3 \quad \text{(utility of no-purchase / outside option)}
\end{aligned}$$

If these are the options, customer #7 will choose sushi #2 (sea eel) because this provides the highest utility.

In our model of the customer choice behavior, we assume each customer selects his/her highest utility option. We also assume that each customer has an equal weight/probability ($1/K$). Therefore, if we offer the set $S \subseteq \{1, \dots, n\}$, then the predicted revenue is the average of the revenue from the choice of each of the 500 customers: mathematically, this is

$$R(S) = \frac{1}{K} \sum_{k=1}^{K} \left[\sum_{i \in S} r_i \cdot \mathbb{I}\{i \text{ is the first choice of customer } k \text{ in } S \cup \{0\}\} \right]$$

where $\mathbb{I}\{A\}$ is the indicator function for the event A (it is 1 if A is true, and 0 if A is false) and the first choice is that choice which provides the highest value of $u_{k,i}$, i.e., it is $\arg\max_{i \in S \cup \{0\}} u_{k,i}$.

Part 1: Understanding the data

Before we start, let's understand some patterns in the data. Load the file `sushi_utilities_mat.csv`.

a) What are the five most preferred sushis for customer 1? *Hint*: use the command `argsort` from `numpy` on a particular row of the utility matrix.

b) What are the five least preferred sushis for customer 2?

c) For each customer, compute the *rank* of each of the 100 sushis according to their utilities. Which sushis are the top five, i.e., the five with the best average rank over the 500 customers? What do you notice about the sushi (specifically, the type of fish)?

   Hint: Consider the following code snippet:

```
import numpy as np
temp = np.array([4.8, 2.3, 5.3, 3.9, 1.4, 5.1])
# temp contains the (hypothetical) utilities of 6 products
# Apply argsort once:
temp2 = np.argsort(temp) # What does temp2 contain?
temp3 = np.argsort(temp2) # What does temp3 contain?
```

d) Which sushi has the worst average rank over the 500 customers?

e) Which sushi is the most controversial, as measured by the standard deviation of its rank over the 500 customers?

Part 2: Common-sense solutions

Load the data into Python. Create a function that computes the expected per-customer revenue of an assortment of sushi items S, where each customer is picking the option that gives them the highest utility. As a check, if we offer the set of sushis $S = \{1, 2, 3, 4, 5\}$, then the per-customer revenue should be 14.2396. (*Note*: for some customers, it will be the case that all of the sushis will have a utility $u_{k,i}$ lower than the no-purchase utility $u_{k,0}$; these customers will never choose any sushi we offer them. Customers #1, 2 and 5 are example of this. Despite these customers not choosing anything, please continue to include them in the K of 500 for your revenue calculations.)

a) Suppose that we simply offer all sushi products, that is, we set $S = \{1, 2, \ldots, n\}$. What is the expected revenue in this case?

b) Suppose that we offer the ten highest revenue sushis, that is, we set $S = \{i_1, \ldots, i_{10}\}$, where $r_{i_1} \geq r_{i_2} \geq \cdots \geq r_{i_n}$. Which are the ten highest revenue sushis? What is the expected revenue in this case?

c) Suppose that for every customer k, we determine his/her most preferred sushi, $i^*_k = \arg\max_{1 \leq i \leq n} u_{k,i}$. We then offer all of the most preferred sushis, that is, we set $S = \{i^*_1, i^*_2, \ldots, i^*_K\}$. (Note that some sushis may be the top choice of more than one customer.)
What is the expected revenue in this case? What do you notice about this value?

d) Explain why the solution in (a) may be suboptimal.

e) Explain why the solution in (b) may be suboptimal.

Part 3: An integer optimization model

Let's see how to formulate the problem of selecting an optimal set of sushis as an integer optimization problem.

Consider the following integer optimization model:

$$\underset{\mathbf{x},\mathbf{y}}{\text{maximize}} \quad \frac{1}{K} \sum_{k=1}^{K} \sum_{i=1}^{n} r_i y_{k,i} \tag{14a}$$

$$\text{subject to} \quad \sum_{i=0}^{n} y_{k,i} = 1, \quad \forall\, k \in \{1, \ldots, K\}, \tag{14b}$$

$$\sum_{j=0}^{n} u_{k,j} y_{k,j} \geq u_{k,i} x_i + u_{k,0}(1 - x_i), \quad \forall\, k \in \{1, \ldots, K\},\ i \in \{1, \ldots, n\}, \tag{14c}$$

$$\sum_{j=0}^{n} u_{k,j} y_{k,j} \geq u_{k,0}, \quad \forall\, k \in \{1, \ldots, K\}, \tag{14d}$$

$$y_{k,i} \leq x_i, \quad \forall\, k \in \{1, \ldots, K\},\ i \in \{1, \ldots, n\}, \tag{14e}$$

$$x_i \in \{0, 1\}, \quad \forall\, i \in \{1, \ldots, n\}, \tag{14f}$$

$$y_{k,i} \in \{0, 1\}, \quad \forall\, k \in \{1, \ldots, K\},\ i \in \{1, \ldots, n\}, \tag{14g}$$

a) Explain how constraints (14b) – (14e) correctly model the preferences of each customer.

b) Implement the linear optimization *relaxation* of the above problem in Python. The relaxation is obtained by relaxing $\mathbf{x}$ and $\mathbf{y}$ to be continuous decision variables as opposed to binary decision variables – that is, we replace constraints (14f) and (14g) with:

$$0 \leq x_i \leq 1, \quad \forall\, i \in \{1, \ldots, n\}, \tag{15}$$

$$0 \leq y_{k,i} \leq 1, \quad \forall\, k \in \{1, \ldots, K\},\ i \in \{1, \ldots, n\}. \tag{16}$$

What is the optimal objective value of the relaxation?

c) A manager claims that it should be possible to obtain an assortment with an expected per-customer revenue of \$32. Explain why this is impossible based on your answer to (b).

d) Now, implement the integer version of the above problem (i.e., the binary constraints (14f) and (14g) are enforced) in Python. Solve the problem. What is the expected per-customer revenue of the optimal assortment? How much does this improve over the assortments from Part 2?

e) What is the optimal set of sushis the restaurant should offer?