MGMTMSA 408 – Operations Analytics – Spring 2024

Final Exam

Due June 16, 2024 at 11:59pm PST on BruinLearn

Instructions

1. This exam consists of 4 questions, on 8 pages (including this one).
2. The take-home exam period begins on **Tuesday, June 11 at 12pm PST** and ends at **Sunday, June 16 at 11:59pm PST**. You may use as much of this time as you wish to solve the exam.
3. Save your answers to a separate PDF document, one for each question. Set the filename to `"FirstName LastName - Problem X.pdf"` where FirstName and LastName should be replaced by your first and last name, and X should be set to the question number. For example, `"John Doe - Problem 3.pdf"` would be the file name for John Doe's submission for Question 3. BruinLearn provides a separate PDF with blank boxes for each question.
4. Save the R commands used to solve a given problem to `"FirstName LastName - Problem X.R"` where X is the number of the problem to which the R script corresponds to. Similarly, save the Python/Jupyter Notebook used to solve a problem to `"FirstName LastName - Problem X.ipynb"`. Ensure that your Jupyter Notebooks include the output from all of your commands.
5. Failure to follow items #3 and #4 will result in a penalty of 10% on your total exam score.
6. Please show your work and include any calculations/commands needed to justify your answer. Answers without accompanying work will not receive any credit.
7. You may use any of the lecture slides, in-class examples or homework solutions to help you solve the problems.
8. Do not use packages or software that we have not used in class. We are not obligated to award partial credit to wrong answers obtained through packages or software not covered in class.
9. Submit your written solutions, R code and Python/Jupyter Notebook code through the submission link on BruinLearn (under the "Final Exam" page) by **11:59pm PST on Sunday, June 16**. **It is your responsibility to make sure that you have hit the submit button properly**. If you submit prematurely, you can modify your submission through BruinLearn; if this does not work, you may email your final submission to the instructor and the TA. **Late submissions will not be accepted.**
10. **You must complete the exam on your own. Discussing any aspect of the exam with a classmate or anyone else who is not the instructor or the teaching assistant is a violation of academic integrity and will result in disciplinary action.**
11. **Do not post any questions about the exam on BruinLearn.**
12. If you encounter technical issues with using any of the data sets, please email both the instructor and the teaching assistant. If any major issues emerge, we will send out announcements via BruinLearn.
13. The instructor and the teaching assistants are not allowed to answer any conceptual questions regarding the exam.

GOOD LUCK!

1 Medical waste disposal

Medical Waste Management Inc. (MWM) is a company that helps hospitals in the Greater Los Angeles area with disposal and transportation of medical waste. Each day, a truck is dispatched from MWM's depot to a set of hospitals to collect medical waste. Upon visiting all of these hospitals, the truck returns to MWM's depot. MWM is interested in determining an optimal route for their truck to visit all of the locations, so as to minimize the total transportation cost. The total transportation cost is given by

$$\$5000 + \$0.1 \times D$$

where D is the total distance traveled.

MWM serves a total of 117 hospitals. Two data files are provided. The first file, `MWM-hospitals.csv`, lists the hospitals, their addresses and their latitudes and longitudes. The second file, `MWM-dist.csv`, gives the driving distance (`dist`) from each hospital i (column `origin`) to hospital j (column `destination`). The values of `origin` and `destination` range between 0 and 117. The dummy value of zero corresponds to MWM's depot located in Van Nuys.

Part 1: Understanding the data

a) Which hospital is farthest from the depot?

b) Which hospital is closest to the depot?

c) Consider the following heuristic. Starting from MWM's depot, the next hospital in the sequence is the closest one in distance to the truck's current location. After all of the hospitals have been visited, the truck travels back to the depot.

   What is the cost of the route produced by this heuristic?

d) Consider the following alternate heuristic: create a random ordering of the 117 hospitals, and visit the hospitals in that sequence before returning to the depot. Simulate this heuristic 100 times, and save the cost of each simulation. (Set your random seed to 50 beforehand.)

   What is the average cost of this heuristic over the 100 simulations? What is the best (= lowest) cost that this heuristic attains over the 100 simulations?

Part 2: An optimization formulation

a) Write down an abstract formulation of the problem as an integer program. Define the decision variables, constraints and the objective function, and any parameters that your formulation requires. (Do not formulate any constraints with actual numbers).

b) Implement and solve your optimization problem. What is the optimal cost?

c) What are the first five hospitals that are visited in the optimal route?

2 Bike-share network design

A new bike-share company is interested in selecting docking locations for its bikes in Santa Monica. There are 42 candidate locations being considered by the bike-share company. The company has identified 700 potential customers who may be interested in using the service. The company would like to select exactly 10 locations, but does not know where these locations should be.

The file `bikeshare-time.csv` contains the walking time between each candidate location (1,...,42) and each potential customer (1, ..., 700). The file `bikeshare-customer-locations-v2.csv` contains the locations of the customers, as well as an additional column (`walktime`) which will be used in Part 4. The file `bikeshare-dock-locations-v2.csv` contains the locations of the dock locations, as well as an additional column (`capacity`) which will be used in Part 4.

Part 1: Minimizing average distance

The first question that the bike-share company asks is where to locate the 10 docking stations so as to minimize the average distance of each customer from their closest location.

a) Write down an abstract formulation of the problem as an integer programming problem. Define the decision variables, constraints and objective function, and any parameters that your formulation requires. Explain why your formulation is correct. (Do not formulate any constraints with actual numbers, with the exception of the requirement that 10 docking stations be selected.)

b) Formulate and solve your problem in Python using Gurobi. What is the optimal objective value?

c) Which stations are opened in the optimal solution? (Write down the indices from smallest to largest value.)

Part 2: Minimizing maximum distance

The bike-share company now considers an alternate strategy to determining the locations of the docking stations, based on minimizing the maximum walking distance of any customer from their closest docking station.

a) Write down an abstract formulation of the problem as an integer programming problem. Define the decision variables, constraints and objective function, and any parameters that your formulation requires. Explain why your formulation is correct. (Do not formulate any constraints with actual numbers, with the exception of the requirement that 10 docking stations be selected.)

b) Formulate and solve your problem in Python using Gurobi. What is the optimal objective value?

c) Which stations are opened in the optimal solution? (Write down the indices from smallest to largest value.)

Part 3: Maximizing coverage

From customer surveys, a major concern for many customers is having to walk too far to the closest docking station. Suppose that the company now wishes to locate its docking stations so as to maximize the number of customers who are within 10 minutes of a docking station.

a) Write down an abstract formulation of the problem as an integer programming problem. Define the decision variables, constraints and objective function, and any parameters that your formulation requires. Explain why your formulation is correct. (Do not formulate any constraints with actual numbers, with the exception of the requirement that 10 docking stations be selected.)

b) Formulate and solve your problem in Python using Gurobi. What is the optimal objective value?

c) Which stations are opened in the optimal solution? (Write down the indices from smallest to largest value.)

Part 4: Walking time preferences and capacity

From further customer surveys, the company has determined that each customer will in general use the dock location that is nearest to him/her (requiring the least amount of time to walk to). Each potential customer has indicated to the company the most amount of time that he/she would be willing to walk to reach a dock location. This time varies between 2 and 10 minutes for each customer. The file `bikeshare-customer-locations-v2.csv` contains the maximum amount of time that each customer has indicated that they would walk (`walktime`). If the closest dock location is farther in time than this maximum amount, the customer will give up and not retrieve a bike.

An additional concern that the company has recognized is that the candidate sites for the docking stations are limited in capacity. The file `bikeshare-dock-locations-v2.csv` includes how many bikes will be stored at each candidate location (`capacity`). If the number of customers who choose to walk to a particular location exceeds the capacity of that location, the customers in excess of the capacity will not be able to obtain a bike.

The company wishes to minimize an objective that captures the sum of two quantities:

- F_1: the number of customers who do not obtain a bike (either because a bike is unavailable at their closest dock location, or because the closest dock location was too far); and
- F_2: the number of bikes at an open docking station that are unused by any customer.

The company determines that F_1 is twice as important as F_2, and hence wishes to minimize $2F_1+F_2$.

a) Write down an abstract formulation of the problem as an integer programming problem. Define the decision variables, constraints and objective function, and any parameters that your formulation requires. Explain why your formulation is correct. (Do not formulate any constraints with actual numbers, with the exception of the requirement that 10 docking stations be selected.)

Formulate and solve your problem in Python using Gurobi.

b) How many customers do not walk to any of the opened locations?

c) How many customers walk to a location but find that a bike is unavailable?

d) How many bikes are not used by any customer?

3 Grocery store inventory optimization

A grocery store chain must decide on how many units to stock of a strawberry product (a 10.5 oz pack of fresh organic strawberries) each week. The product is sold for \$6 in the grocery store, and costs \$3 per unit to stock. Each unit that is unsold after a week is sold to a processing plant that uses the no longer fresh strawberries to produce strawberry-based products (such as jam). The price at which each unsold unit is sold to the processing plant is \$1.5. So as an example, if 5 units of the product are demanded and we have only stocked 3 units, then the profit is

$$\$6 \times 3 - \$3 \times 3 = \$9 \tag{1}$$

On the other hand, if 5 units of the product are demanded and we stocked 7 units, then the profit is

$$\$6 \times 5 - \$3 \times 7 + \$1.5 \times (7 - 5) = \$12. \tag{2}$$

The grocery store chain is interested in deciding how many units of this strawberry product to stock each week. To aid in this decision, the grocery store has provided a data set, `strawberry.csv`, containing 200 weeks of demand. For each week, the grocery store has data on the demand for the strawberry product (the column `demand`), as well as several other covariates. These include:

- `app.logins`: the number of logins to the grocery store's app, which provides information about special products and promotions;
- `temperature`: the average weekly temperature in degrees Fahrenheit;
- `prev.demand`: the demand for the strawberry product in the previous week; and
- `prev.demand.2`: the demand for the strawberry product two weeks ago.

Part 1: A constant order quantity

As a first step, the grocery store chain considers ordering the same quantity each week, and assuming that the time-varying factors that provided in the dataset do not affect demand. Before solving the data, split the data as follows: set the training set equal to the first 150 observations and the test set equal to the remaining 50 observations.

a) Formulate the problem of deciding how many units of the strawberry product to stock each week as a cost-based newsvendor problem. What is the demand D? What is the order quantity Q? What are the underage cost c_u and the overage cost c_o for this problem?

b) Calculate the optimal constant order quantity using the training set.

c) Calculate the average profit of this constant order quantity using the training set.

d) Calculate the average profit of this constant order quantity using the test set.

Part 2: A contextual ordering policy

The grocery store now decides to investigate whether it can achieve higher profits by varying the order quantity each week, based on the covariates provided in `strawberry.csv`. Build a regression tree using the training set from Part 1, with a maximum depth of 3 and a minimum number of observations per leaf of 15.

a) Which variables appear in the splits of your tree?

b) Use your regression tree to propose an ordering policy. What are the order quantities that your policy prescribes? Under what conditions (i.e., what values of the covariates in the data) does your policy prescribe those order quantities?

c) Calculate the average profit of this contextual ordering policy using the test set.

Part 3: A different contextual ordering policy

Using the training set, build a linear regression model to predict the demand using the covariates in your training set from Part 1.

a) Use your linear regression model to propose an ordering policy. Write down an equation for the form of your contextual ordering policy. (*Hint*: What assumption is being made about the conditional distribution of demand when one uses a linear regression model?)

b) Calculate the average profit of this contextual ordering policy using the test set.

Part 4: An idealized ordering policy

Finally, in this section we will understand the limits of performance for our approaches in Parts 1, 2 and 3. To do this, let us consider the following idealized policy: suppose that each week, we could know the demand that would be realized that week, and we could order exactly the demanded quantity of the product.

a) Calculate the average profit of this policy on the test set.

b) Consider the *coefficient of prescriptiveness*, which is defined as

$$\frac{R_{contextual} - R_{constant}}{R_{idealized} - R_{constant}} \tag{3}$$

where $R_{constant}$, $R_{contextual}$, $R_{idealized}$ are the average profits corresponding to the constant policy in Part 1, the contextual policy in either Part 2 or Part 3 and the idealized policy in Part 3.

Calculate the coefficient of prescriptiveness using the test set for the policies in Part 2 and Part 3.

c) The metric above is closely related to another metric that is used to measure the predictive performance of continuous prediction models. What is that metric? How is the coefficient of prescriptiveness similar to that metric?

4 Hotel revenue management

A luxury hotel is preparing to allow for reservations for a particular period in September. The hotel has three different room types, distinguished by the type of bed: Queen, King, and California King. There are 50 Queen (Q) rooms, 50 King (K) rooms and 20 California King (C) rooms. For convenience, let us use r to denote the room type, which satisfies $r \in \{Q, K, C\}$.

The hotel is interested in managing reservations over the period 9/11/2023 to 9/18/2023, which is a period of 8 days. For convenience, let us use the index d to denote the day, which can range from 1 (Monday 9/11/2023) to 8 (Monday 9/18/2023). We would like to model customers in the following way:

- Customers can check-in on any day $d_{in} \in \{1, \dots, 7\}$.
- Customers can check-out on any day $d_{out} \in \{d_{in} + 1, \dots, 8\}$.

A customer type is denoted by the tuple (r, d_{in}, d_{out}), where $r \in \{Q, K, C\}$. The data frame `hotel-probability.csv` is provided, which provides the likelihood of a customer request being of a particular type – i.e., a particular tuple (r, d_{in}, d_{out}) – over a 4 hour period. In each period, there is at most one reservation, if a reservation occurs at all. We will assume that a customer can request a reservation over the 90 days that precede 9/11/2023. Therefore, a customer request can occur in any period $t = 1, \dots, T$, where $T = 90 \times (24/4) = 540$.

In addition, the following is known about the pricing structure for this period in time:

- Queen rooms are charged at a base price of \$200 per night, King rooms at a base price of \$250 a night, and California King rooms at a base price of \$300 a night.
- For any type of room, the fee for Thursday night, Friday night, Saturday night ($d = 4, 5, 6$) incurs an additional 15% markup.

For example, suppose that we accept a reservation of type $(Q, 2, 6)$. This customer checks in on Tuesday ($d = 2$) and checks out on Saturday ($d = 6$). Therefore, the customer stays four nights. The total price charged to the customer for this reservation is

$$\begin{aligned} &\$200 + \$200 + \$200 \times 1.15 + \$200 \times 1.15 \\ &= \$860 \end{aligned}$$

Part 1: Understanding the data

a) What is the probability that the hotel receives a reservation for a Queen room (with any check-in / check-out dates) in a given (4 hour) period?

b) What is the probability that the hotel receives a reservation that checks in on Thursday 9/14/2023 and checks out on Sunday 9/17/2023 (with any kind of room) in a given (4 hour) period?

c) What is the probability that there is no request in a given (4 hour) period?

d) Over the 90 day horizon, what is the expected value of the number of requests for a California King room, with check-in on Friday 9/15/2023 ($d = 5$) and check-out on Saturday 9/16/2023 ($d = 6$)?

Part 2: Determining an optimal static allocation

a) Using the information given, formulate a linear programming problem to determine how many requests of each type – each tuple (r, d_1, d_2) – to accept over the 90 day horizon. Describe your decision variables, the constraints and your objective function. Describe any coefficients or right hand side values in your constraints. Do not formulate your problem with any actual numbers. (*Hint*: what is a "resource" in this problem? How many resources are there? Which resources are consumed by, for example, accepting the request in Part 1(d)?)

b) Implement and solve your linear program in Python using Gurobi. What is the optimal revenue?

c) What are the top five request types in terms of the number of accepted requests in the optimal solution?

d) What are the optimal values of the dual variables for the resource constraints?

e) An assistant manager at the hotel suggests the idea of converting 10 King rooms into Queen rooms. Without solving another linear program, what is the predicted change in the revenue that would result from this conversion?

Part 3: Determining an optimal dynamic allocation policy

In this part of the problem, we will consider different dynamic allocation policies. To do this, we will need to generate random sequences of requests. Set your random number seed to 50. Generate 100 random sequences of $T = 540$ requests.

a) Currently, the hotel accepts requests on a first-come first-serve basis. What is the average revenue of this policy over the 100 random requests sequences?

b) Suppose that a request of type $(K, 2, 5)$ arrives at some period $t \in \{1, \ldots, T\}$. By defining appropriate dual variables for your constraints in your formulation in Part 2, write down the algebraic expression for the opportunity cost of accepting this request.

c) Using the insight from (b), implement a dynamic allocation policy for accepting / rejecting requests that uses your linear program from Part 2 in Python. What is the average revenue of this policy over the 100 random request sequences? (*Note*: due to the large number of periods, it may take a couple of minutes for your simulation to run here.)

d) Explain why the dynamic allocation policy in (c) performs better than the hotel's current policy in (a). (*Hint*: which request types are at or near the top in terms of the arrival probability / expected number of arrivals over the 540 period horizon? How does that align with which resources are completely used up or almost completely used up by the policy in (a)? Based on this comparison, can you justify why might it be disadvantageous to accept the aforementioned request types?)