

Trabalho de Implementação - Relatório

Ana Carolina Erthal

13 de novembro de 2022

1 Introdução

Para a realização desse trabalho tive grande interesse em estudar o uso da Otimização na mobilidade urbana, já que o uso de dados para avanços da modelagem e previsão de tráfego me pareceram um uso muito relevante da Ciência de Dados.

Encontrei o artigo [Graph Analytics and Optimization Methods for Insights from the Uber Movement Data](#), de autoria de Arun V. Sathanur, Vinay Amatya, Arif Khan, Robert Rallo e Kelsey Maass, todos do Pacific Northwest National Laboratory, de Washington.

O objetivo principal do artigo é discorrer sobre a aplicação de métodos de Otimização em dados do aplicativo Uber, obtidos na página [Uber Movement](#), que disponibiliza dados para estimular pesquisas sobre transporte urbano. Além de desenvolver um método para estimar o tempo de viagem na cidade de Los Angeles, o artigo discorre sobre diversas outras aplicações na área, reforçando a quantidade de aplicações possíveis para os dados em questão na área da Otimização.

O método utilizado no artigo é o de Mínimos Quadrados não negativos, empregando matrizes que correlacionam Origem e Destino medindo o tempo de viagem a partir dos dados disponibilizados. No entanto, ao avançar no assunto, me deparei com algumas situações que me fizeram reavaliar a situação, principalmente em termos da implementação. Diversos passos anteriores à implementação - relativos ao tratamento dos dados espaciais de formas bem específicas - eram apenas mencionados, além da falta de implementações na área.

Segui, portanto, o caminho de pesquisar mais sobre outras aplicações dos dados da Uber através da otimização, uma vez que o objetivo do arquivo era promover essa prática, citando diversos outros trabalhos na área. Me deparei com uma proposta muito interessante, em que o objetivo era determinar, dada a agenda flexível de um motorista, quais os melhores horários e regiões para trabalhar, assim como o lucro esperado dada essa configuração. O trabalho, [Uber Driver Schedule Optimization](#), pertence a Ivan Zhou, University of California, Berkeley.

2 O estudo

2.1 Proposta

A ideia da implementação é, levando em conta a flexibilidade ofertada aos motoristas pela Uber, utilizar esse fator para otimizar o lucro total obtido. É sempre uma dúvida se valeria a pena, por exemplo, atender regiões em seu horário de maior movimento - dado que haverá mais demanda, no entanto mais tempo de trânsito - ou se o melhor é atender regiões com menor demanda, mas em que cada viagem será mais rápida.

Dessa forma, temos interesse em otimizar o lucro levando em conta o tempo de viagem e demanda.

2.2 Dados

O em destaque, Uber Movement, não aborda a questão da demanda, então utilizaremos o DC Taxicab Data, fazendo a análise em Washington e unindo os datasets para reunir todas as informações que nos interessam. O processo definido pelo autor é o de considerar as 100 regiões de maior demanda, clusterizar em 5 regiões principais e organizar os dados pivotando por dia da semana, horário (em intervalos de 2h) e região. É importante considerar que só levamos em conta viagens que ocorrem inteiramente dentro da mesma região, uma vez que nosso objetivo é avaliar relações com o tráfego interno.

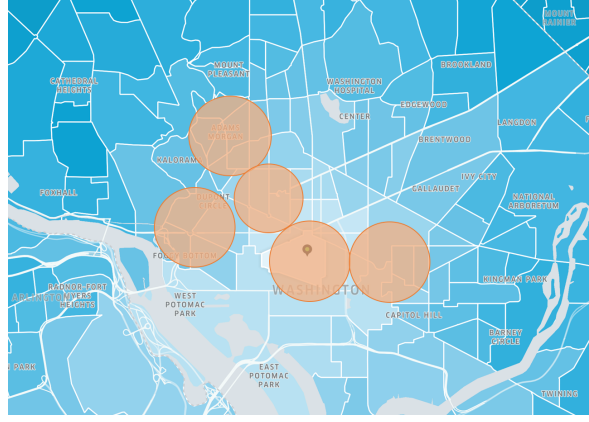


Figura 1: Regiões determinadas sobre as zonas do Uber Movement Data (só foram utilizados dados de zonas inteiramente dentro das regiões circulares).

2.3 Abordagem

É importante observar que nos deparamos com uma situação em que nosso modelo terá que seguir os conceitos da Programação Inteira, isto é, uma forma de Programação Linear em que algumas variáveis são obrigatoriamente inteiras. Note que esse problema é considerado NP-difícil.

Por isso, existe a abordagem de fazer uma busca "relaxada" e arredondar resultados (que em geral produz respostas menos exatas), a de enumerar soluções de forma inteligente e testar, e também o método de cortes, quando temos restrições que são inequações. O MATLAB, que usei para implementar, utiliza a abordagem de seguir um [passo a passo para simplificar o problema](#), que passa por redução do problema por Pré-processamento de Programação Linear, solução relaxada, cortes para restringir mais, etc..

2.4 Modelo

Na prática, o problema é modelado da seguinte forma:

$$\begin{aligned}
 &\text{Maximizar } \sum_{ij} R_{ij} x_{ij} \\
 &\text{s. a } \quad x_{ij} \leq A_i, \forall i, j; \\
 &\quad \quad x_{ij} \leq B_j, \forall i, j; \\
 &\quad \quad \sum_{ij} x_{ij} \leq T_{max}, \forall i, j; \\
 &\quad \quad \sum_j x_{ij} \leq 1, \forall i; \\
 &\quad \quad x_{ij} \in \{0, 1\}, A_i \in \{0, 1\}, B_j \in \{0, 1\}, i \in I, j \in J
 \end{aligned}$$

Onde as variáveis são tais que:

- i : horário em questão;
- j : região em questão;
- x_{ij} : booleano que indica a seleção (1) ou não (0) da região e horário em questão;
- R_{ij} : função de lucro em uma semana

$$R_{ij} = P(\text{nova viagem}) * \text{Número mínimo de viagens}$$

$$\text{Número mínimo de viagens} = \frac{7200s}{\text{Duração máxima da viagem}}$$

Observe que usamos 7200s pois os horários têm intervalo de 2h;

- A_i : booleano que indica disponibilidade do motorista no horário i ;
- B_j : booleano que indica disponibilidade do motorista na região j ;
- T_{max} : total de horas disponíveis na semana.

Cada restrição tem seu papel essencial. $x_{ij} \leq A_i, \forall i, j$ e $x_{ij} \leq B_j, \forall i, j$ garantem que o motorista está disponível no horário i e região j

$\sum_{ij} x_{ij} \leq T_{max}, \forall i, j$; garante que o total de horas semanais não é excedido, e $\sum_j x_{ij} \leq 1, \forall i$; que o motorista só estará em uma região por horário. As demais restrições definem as binariedades e pertencimento a conjuntos necessários.

2.5 Implementação

A implementação foi feita em MATLAB e adaptada do código produzido pelo autor do projeto.

Iniciamos importando os dados e definindo vetores a partir das colunas de interesse:

```
1 T = readtable('Data_Demand_Variability.csv');
2
3 Weekday = T.Weekday;
4 Timeslot = T.Timeslot;
5 Max_Duration = T.AverageMax;
6 Avg_Traffic = T.Avg_Traffic;
7 Region = T.Region;
```

Definimos também os parâmetros que utilizaremos para a situação em questão, isto é, os horários e regiões de interesse do motorista, etc.:

```
1 % Parametros
2 max_hours_per_week = 8;
3 % Horarios disponiveis
4 time_slot_available = ones(7,24);
5 time_slot_available(2,:) = 0;
6 time_slot_available(4,:) = 0;
7 time_slot_available(6:7,:) = 0;
8 time_slot_available(:,1:18) = 0;
9 % Regioes de interesse
10 region_available = ones(5,1);
11 avg_revenue_trip = 12; % Lucro medio por viagem
12
13 p_max = 1;
14 p_min = 0.5;
```

Definimos uma função que partirá de nossos dados de demanda para obter a probabilidade de uma nova viagem:

```
1 function P_new_trip = Cvt_Traffic_Probability1(Avg_Traffic,p_max,p_min)
2 multiplier = (p_max-p_min)/(max(Avg_Traffic)-min(Avg_Traffic));
3 P_new_trip = p_min+ multiplier.*(Avg_Traffic-min(Avg_Traffic));
4 end
```

Agora podemos elaborar a função que de fato realiza o modelo descrito:

```
1 function [x,obj_ip,time_ip] = Solve_IP(max_hours_per_week,time_slot_available,
2     region_available,avg_revenue_trip,p_max,p_min)
3
4 P_new_trip = Cvt_Traffic_Probability1(Avg_Traffic,p_max,p_min);
5 Min_Trips = 7200./Max_Duration;
6
7 weekday_choices = 1:7;
8 time_choices = [0 1 7 8 16 17 18 19 20 21 22 23];
9 timeslot_choices = [0 1 2 3 4 5];
10 n_weekdays_choices = size(weekday_choices,2);
11 n_timeslot_choices = size(timeslot_choices,2);
12 n_weekdays = 7;
13 % marcacao e remocao de horarios e regioes nao disponiveis do motorista omitidas aqui
14 % devido ao codigo extenso.
15 % na pratica realizamos loops para marcar e depois remover os indices em questao
```

```

16 % aplicacao das restricoes e programacao de inteiros
17 n_x = size(Min_Trips,1);
18 f = (P_new_trip.*Min_Trips)'.*avg_revenue_trip;
19 intcon = 1:n_x;
20
21 % Restricao 1: numero max de horas por semana
22 A1 = ones(1,n_x);
23 b1 = max_hours_per_week/2;
24 % Restricao 2: esta em uma regioao por vez
25 A2 = zeros(n_weekdays_choices*n_timeslot_choices,n_x);
26 b2 = ones(n_weekdays_choices*n_timeslot_choices,1);
27 row_count = 0;
28
29 for d = weekday_choices
30     for t = timeslot_choices
31         row_count = row_count+1;
32         ind_marked = (Weekday == d & Timeslot == t);
33         A2(row_count,ind_marked) = 1;
34     end
35 end
36
37 A = [A1;A2];
38 b = [b1;b2];
39 lb = zeros(n_x,1);
40 ub = ones(n_x,1);
41
42 % solucao
43 [x,fval] = intlinprog(-f,intcon,A,b,[],[],lb,ub);
44 obj_ip = -1*fval;

```

Foram omitidas aqui questões quanto à seleção apenas das regiões e horários determinados pelo motorista, assim como do output desejado (com o fval, mas também saída com interpretação do vetor x para sabermos o melhor roteiro possível para o motorista dadas as restrições). Mas nesse ponto a otimização já foi realizada.

Veja a seguir as saídas dadas as restrições e horários que impusemos:

		Lucro total otimizado: 501.0025		
Horas por semana	8	Rec	Horário	Região
Dias	Seg, Qua, Sex	1	Qua, 20h às 22h	2
Horários	18h a 00h	2	Sex, 20h às 22h	4
Regiões	1, 2, 3, 4, 5	3	Sex, 22h às 00h	4
		4	Seg, 20h às 22h	4

Figura 2: À esquerda as escolhas do motorista X. À direita seu lucro total otimizado e cronograma correspondente.

O resultado foi bem interessante em termos de aplicações práticas. Isto é, dentro da área de Ciência de Dados busca-se frequentemente situações deste tipo, em que podemos aproveitar dados existentes para otimizar situações.

Para um motorista de aplicativo certamente é interessante, considerando seus horários disponíveis para trabalhar (ou mesmo regiões que prefere evitar por questões de segurança, por exemplo), predefinir qual as condições ideais para maximizar seu lucro. É importante atentar-se, porém, que nesse trabalho não consideramos o fato de que o valor da viagem se altera bastante, consideramos apenas um valor médio.

Assim, pegar viagens mais longas no trânsito podem ter recompensas financeiras também, o que deve ser levado em conta para aplicação prática. De qualquer forma, o resultado teórico é bastante interessante.