

MFIN 706: Midterm Test

Dr. Oleksandr Romanko

June 24, 2016

Due: Sunday, June 26, 2016, not later than 10:00 p.m.

Use **MATLAB** for all MFIN 706 tests and assignments.

You should hand in:

- Your report (up to 1-page) and your MATLAB code.
 - Compress your report, all of your MATLAB code files and output files into a file **StudentID.zip** (which can be uncompressed by 7-zip (<http://www.7-zip.org>) software under Windows), and submit it via Avenue to Learn portal no later than 10:00 p.m. on June 26.
-

Question

Due to the retirement of the Canadian penny in 2013, **small businesses now round cash purchases to the nearest 5 cents**. However, Canadian businesses generally advertise prices before tax, making it incredibly difficult to put together a menu that ends in 0 or 5 without rounding so that pennies are not needed.

Your challenge is to put together a menu for the fictional MFIN Cafe (in Ontario, Canada) that **rounds cleanly to a 0 or a 5 for the most transactions**. Ontario sales tax (HST) is 13%. The current menu and a record of historical transactions can be found in the attached file. The **only restriction is that current prices cannot be changed by more than 2 cents** (up or down).

Feel free to make appropriate assumptions and/or simplifications if you think that they are in the spirit of the problem.

Tax Calculation

Thinking about your coffee, you realize that the \$1.24 you paid for it plus the 13% Ontario sales tax comes out to exactly \$1.40. However, had you purchased a donut with your coffee, the final price would be \$2.47.

HST is calculated for the whole customer's purchase (for a receipt or an invoice), e.g., if you bought the XS coffee and the donut then the price with tax is $(1.24 + 0.95) \cdot 1.13 = 2.19 \cdot 1.13 = 2.4747 = \2.47 . can be rounded to 2.45 or 2.50, since under both rounding methods, the rounded price is within the constraint of 2 cents (there are two types of products, so the total variation allowed is (+4, -4)). However, for

the company, they would prefer to round up to 2.50 to increase their revenue
Tax on the total is rounded itself to the nearest cent based on general rounding principles – round tax first fractions less than half a cent are disregarded, fractions greater or equal than half a cent are rounded to a cent. For instance, total purchase with tax of $2.19 \cdot 1.13 = 2.4747$ is rounded to \$2.47, while the total purchase of $2.20 \cdot 1.13 = 2.486$ is rounded to \$2.49.

Report

In your report (up to 1-page) describe how you solved the problem and **which computational techniques and algorithms you used**. If you utilize optimization techniques in your MATLAB

code, you can only use CPLEX, IPOPT or MATLAB optimization routines (e.g., `fmincon`).
 Include a table that shows current prices, new prices that you computed and price changes that you propose (those cannot exceed 2 cents, up or down). For the prices that you computed, report which percentage of the 200 purchases (historical transactions from the data file) cleanly rounds to a 0 or a 5. Discuss the performance of your algorithm, e.g., running time, solution precision, computational complexity, implementation challenges, etc. How can you improve your algorithm or its running time?

three parts, the price does not need rounding, price rounded to zero, and price rounded to 5

MATLAB Script for Testing Your Algorithm

```

clc;
clear all;
format long

% Input files
input_file_prices = 'prices.csv';
input_file_purchases = 'purchases.csv';

% Read transactions
if(exist(input_file_purchases,'file'))
    fprintf('\nReading transactions datafile - %s\n', input_file_purchases)
    fid1 = fopen(input_file_purchases);
    % Read item names
    hheader = textscan(fid1, '%s', 1, 'delimiter', '\n');
    headers = textscan(char(hheader{:}), '%q', 'delimiter', ',');
    items = headers{1}(2:end);
    fclose(fid1);
    purchases = dlmread(input_file_purchases, ',', 1, 1);
    % Remove empty transactions
    purchases = purchases(find(sum(purchases'),:));
else
    error('Transactions datafile does not exist')
end

% Read prices
if(exist(input_file_prices,'file'))
    fprintf('Reading prices datafile - %s\n', input_file_prices)
    cur_prices = dlmread(input_file_prices, ',', 1, 0);
    cur_prices = cur_prices';
else
    error('Prices datafile does not exist')
end

% Compute new prices
new_prices = tax_algorithm(cur_prices, purchases);

% Compute percentage of the historical transactions that cleanly rounds to a 0 or a 5
cur_purchases = 1.13*(purchases * cur_prices);
new_purchases = 1.13*(purchases * new_prices);
perc_round_cur = sum(mod(round(100*cur_purchases), 5)==0) / size(purchases,1);
perc_round_new = sum(mod(round(100*new_purchases), 5)==0) / size(purchases,1);

fprintf('\n ITEM      CUR PRICE    NEW PRICE    PRICE ADJ\n')
for(i=1:length(cur_prices))
    fprintf('%10s \t %6.2f \t %6.2f \t %6.2f\n', char(items(i)), cur_prices(i),
                                                new_prices(i), new_prices(i)-cur_prices(i))
end

fprintf('\n Current prices: percentage of transactions that
        cleanly round to 0 or 5 = %6.2f %%\n', 100*perc_round_cur);
fprintf('      New prices: percentage of transactions that
        cleanly round to 0 or 5 = %6.2f %%\n', 100*perc_round_new);
  
```

总的来说,我觉得这个作业的目的是找出每个商品最合适的rounding price (且单个商品正负波动不得超过2 cents),使得200个transaction* (1+0.13) 后的税后价能round成尾数为0 or 5 的价格,且rounding的波动是在正负2 cents*购买商品数量之间

MATLAB Function that Implements Your Algorithm

```
function [new_prices] = tax_algorithm(cur_prices, purchases)

    %%%%%%%%%%% Example algorithm, replace by yours %%%%%%%%%%%
    cur_prices(1) = cur_prices(1) + 0.01;
    cur_prices(2) = cur_prices(2) - 0.01;
    cur_prices(3) = cur_prices(3) + 0.02;
    cur_prices(4) = cur_prices(4) - 0.02;
    cur_prices(6) = cur_prices(6) - 0.01;
    cur_prices(7) = cur_prices(7) + 0.01;
    cur_prices(8) = cur_prices(8) + 0.01;
    cur_prices(9) = cur_prices(9) - 0.02;
    cur_prices(12) = cur_prices(12) + 0.01;
    new_prices = cur_prices;

    %%%%%%%%%%% Insert your code here %%%%%%%%%%%

end
```

Sample Output of the MATLAB Script

Reading transactions datafile - purchases.csv
Reading prices datafile - prices.csv

ITEM	CUR PRICE	NEW PRICE	PRICE ADJ
XS Coffee	1.24	1.25	0.01
S Coffee	1.33	1.32	-0.01
M Coffee	1.57	1.59	0.02
L Coffee	1.71	1.69	-0.02
XL Coffee	1.91	1.91	0.00
S Tea	1.33	1.32	-0.01
M Tea	1.52	1.53	0.01
L Tea	1.71	1.72	0.01
Donut	0.95	0.93	-0.02
Bagel	1.24	1.24	0.00
Muffin	1.30	1.30	0.00
Cookie	0.75	0.76	0.01

Current prices: percentage of transactions that cleanly round to 0 or 5 = 23.30 %
New prices: percentage of transactions that cleanly round to 0 or 5 = 30.11 %