MTH 420/520 Homework 7. **Yesh Godse**
Date: June 6, 2019
Prof. M. Peszynska

Problem II.14. See Problem II.14 in class notes for full problem statement

---

Solution II.15.

$$\begin{aligned}
M &= (1-\alpha)G + \frac{\alpha}{4}One \\
&= \begin{bmatrix} \frac{\alpha}{4} & \frac{\alpha}{4} & \frac{\alpha}{4} & \frac{\alpha}{4} + (1-\alpha)) \\ \frac{\alpha}{4} + 0.5(1-\alpha) & \frac{\alpha}{4} & \frac{\alpha}{4} & \frac{\alpha}{4} \\ \frac{\alpha}{4} + 0.3(1-\alpha) & \frac{\alpha}{4} & \frac{\alpha}{4} & \frac{\alpha}{4} \\ \frac{\alpha}{4} + 0.2(1-\alpha) & \frac{\alpha}{4} + (1-\alpha)) & \frac{\alpha}{4} & \frac{\alpha}{4} \end{bmatrix} \\
&= \begin{bmatrix} \frac{\alpha}{4} & \frac{\alpha}{4} & \frac{\alpha}{4} & 1 - \frac{3\alpha}{4} \\ \frac{1}{2} - \frac{\alpha}{4} & \frac{\alpha}{4} & \frac{\alpha}{4} & \frac{\alpha}{4} \\ \frac{3}{10} - \frac{\alpha}{20} & \frac{\alpha}{4} & \frac{\alpha}{4} & \frac{\alpha}{4} \\ \frac{1}{5} + \frac{\alpha}{20} & 1 - \frac{3\alpha}{4} & \frac{\alpha}{4} & \frac{\alpha}{4} \end{bmatrix}
\end{aligned}$$

M is not a stochastic matrix because column 3 does not sum to 1. If we were to use M as it is for PageRank, $\lim_{k\to\infty} x^{(k)}$ would not give us a vector of probabilities that add up to 1. To fix this, we alter the terms of the third column to be $\frac{1}{4}$ instead of $\frac{\alpha}{4}$.

$$M = \begin{bmatrix} \frac{\alpha}{4} & \frac{\alpha}{4} & \frac{1}{4} & 1 - \frac{3\alpha}{4} \\ \frac{1}{2} - \frac{\alpha}{4} & \frac{\alpha}{4} & \frac{1}{4} & \frac{\alpha}{4} \\ \frac{3}{10} - \frac{\alpha}{20} & \frac{\alpha}{4} & \frac{1}{4} & \frac{\alpha}{4} \\ \frac{1}{5} + \frac{\alpha}{20} & 1 - \frac{3\alpha}{4} & \frac{1}{4} & \frac{\alpha}{4} \end{bmatrix}$$

In the context of ranking pages, the adjustment made to M is equivalent to stating that a user on a page where there are no outgoing links (page 3 in the given example) will randomly land on any page.

Now that M is a stochastic matrix, we can analyze the boundary values of α. If $\alpha = 0$, we essentially ignore the chance of users going to any page by random chance, instead saying that users can only go to other pages by following links. The exception to this is column 3, which we have adjusted to always send users to a random page. Conversely, if $\alpha = 1$, we ignore links between pages and give each page an equal likelihood of being randomly accessed by the users. Thus, α is a parameter that dictates how much to weight the given transition matrix A over the chance of users randomly accessing pages. Obviously, accessing pages on the world wide web involves both a transition matrix and the chance of randomly appearing on a page, but an example of a model that could use a boundary value of α is a fixed population graph (such as the model of population flow constructed in assignment 2).

For the rest of this assignment, the value $\alpha = 0.15$ will be chosen. Computing M at this α value gives us:

$$M = \begin{bmatrix} 0.0375 & 0.0375 & 0.25 & 0.8875 \\ 0.4625 & 0.0375 & 0.25 & 0.0375 \\ 0.2925 & 0.0375 & 0.25 & 0.0375 \\ 0.2075 & 0.8875 & 0.25 & 0.0375 \end{bmatrix}$$

Problem II.15. See Problem II.15 in class notes for full problem statement

Solution II.15. The algorithm in example II.13 of class notes works because of two observations.

Firstly, the $\lim_{k\to\infty} x^{(k)} = \begin{cases} 0 \text{ if all } |\lambda| < 0 \\ x^* \text{ if all } |\lambda| \geq 0 \end{cases}$ Here, x^* is the eigenvector corresponding to the eigenvalue of largest magnitude. $\lim_{k\to\infty} x^{(k)}$ is computed by $M^k x^{(0)}$.

The next observation is that the dominant eigenvalue of a stochastic matrix M is 1.

Putting these two together, we can calculate the eigenvector corresponding to the leading eigenvalue of 1 for our stochastic matrix M. In context, this is equivalent to finding the relative importance of each page based on the links between the pages described in A.

Problem II.16. See Problem II.16 in class notes

Solution 3. As previously explained, $\alpha = 0$ corresponds to relying only on the given transition matrix G. In the solution to problem II.14, column 3 of M was modified so M could be classified as a stochastic matrix. Keeping this adjustment, we have

$$M = \begin{bmatrix} 0 & 0 & 0.25 & 1 \\ 0.5 & 0 & 0.25 & 0 \\ 0.3 & 0 & 0.25 & 0 \\ 0.2 & 1 & 0.25 & 0 \end{bmatrix}$$

Because this matrix is stochastic, its leading eigenvalue is 1 and the result of computing iterations $x^{(k+1)} = Mx^{(k)}$ won't converge to 0. If we use an unadjusted M, however, an initial vector r^0 can expose the third column and converge to 0 immediately:

$$M = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0.5 & 0 & 0 & 0 \\ 0.3 & 0 & 0 & 0 \\ 0.2 & 1 & 0 & 0 \end{bmatrix}$$

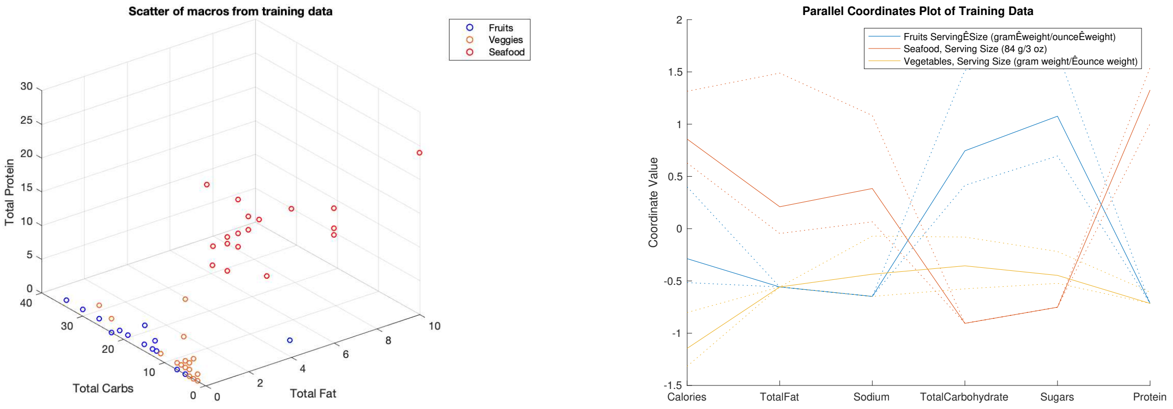
$$\begin{bmatrix} 0 & 0 & 0 & 1 \\ 0.5 & 0 & 0 & 0 \\ 0.3 & 0 & 0 & 0 \\ 0.2 & 1 & 0 & 0 \end{bmatrix} * \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Problem II.17. See Problem II.17 in class notes. dataset: https://fdc.nal.usda.gov

Solution 4. This solution utilized the foundation dataset from Food Data Central to see if food could be classified based on the metrics of Calories, Total Fat, Sodium, Total Carbohydrates, Sugars, and Protein. For simplicity, the dataset is processed to include 20 fruits, 20 vegetables, and 20 seafoods. A batch of testing data (2 each of fruits, vegetables, and seafoods) was also prepared.

Visualizing clusters of the data is simple with 3 or fewer variables, but for higher dimensional data more complicated visualization techniques are necessary. Below I've included the visualizations of the training data using two methods.

Firstly, there is a 3-D scatterplot of each entry in the training dataset based only on macronutrient values (Total Fats, Total Carbs, Total Protein). While it is not hard to find the labeled clusters from visual inspection, a different visualization method for higher-dimensional data could reveal more defined clusters. Alongside the scatterplot is a parallel coordinates plot of all variables in the training data set. The plot shows the median (solid line) and spread of data defined by the interquartile range (between the dashed lines).



(A) Scatterplot (3 out of 6 metrics)

(B) Parallel Coordinates Plot (all metrics)

FIGURE 1. Different visualizations of raw training data

From the second plot, we can see that a metric like sugar is great at distinguishing fruits from the other food groups. Because the 3D scatterplot from earlier did not include this metric, that information was missing from that visualization. The code for generating these plots is listed below.

```
groups = table2array(trainData(:,8));
varNames = {'Calories','TotalFat','Sodium','TotalCarbohydrate','Sugars','Protein'};

figure();
parallelcoords(A, 'group', groups,'standardize','on',...
                 'labels',varNames,'quantile',.25);
title('Parallel␣Coordinates␣Plot␣of␣Training␣Data');

A_vis = A(:,[2 4 6]);
```

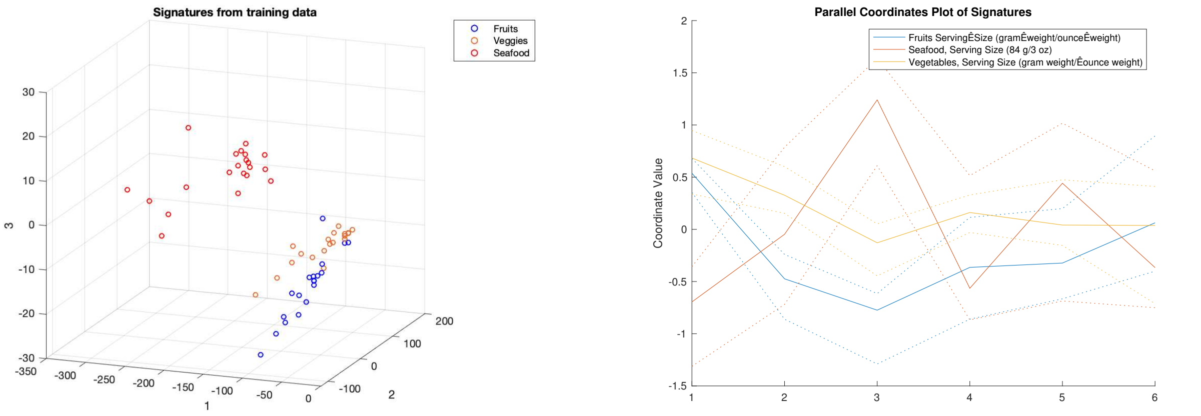
```
figure();
scatter3(A_vis(fruits,1),A_vis(fruits,2),A_vis(fruits,3),size,'blue'); hold on;
scatter3(A_vis(vegetables,1),A_vis(vegetables,2),A_vis(vegetables,3),...
        size,[0.9100 0.4100 0.1700]); hold on;
scatter3(A_vis(seafood,1),A_vis(seafood,2),A_vis(seafood,3),size,'red');
title('Scatter␣of␣macros␣from␣training␣data');legend('Fruits','Veggies','Seafood');
xlabel('Total␣Fat'); ylabel('Total␣Carbs'); zlabel('Total␣Protein');
```

Classificaiton by projections is done by projecting a new data point on the basis V, a product of taking the SVD of our training data stored in matrix A. The resulting projection is a particular data points "signature." We can average the signatures of each labeled datapoint in the training set that comes from the same group to obtain three generic signatures for each food group. Then, by finding the difference between the signature of some new datapoint and these 3 generic signatures, we can classify the new data.

The following code block first calculates the signatures of all points in our training dataset. Then it uses a 3d scatterplot to visualize the clusters of the signatures of the datapoints (using the first 3 elements of the signature) and a parallel coordinates plot to visualize the signatures with all elements of the signature. Finally, the average signature for each food group is calculated.

```
%% Take SVD and compute signatures
[U,S,V] = svd(A);
signatures = A*V;

%% visualize signatures of train data

groups = table2array(trainData(:,8));
varNames = {'1','2','3','4','5','6'};

figure();
parallelcoords(signatures, 'group', groups,'standardize','on',...
               'labels',varNames,'quantile',.25);
title('Parallel␣Coordinates␣Plot␣of␣Signatures');

% % plot scatter of every food group
signatures_vis = signatures(:,[1:3]);
figure();
scatter3(signatures_vis(fruits,1),signatures_vis(fruits,2),...
        signatures_vis(fruits,3),size,'blue'); hold on;
scatter3(signatures_vis(vegetables,1),signatures_vis(vegetables,2),...
        signatures_vis(vegetables,3),size,[0.9100 0.4100 0.1700]); hold on;
scatter3(signatures_vis(seafood,1),signatures_vis(seafood,2),...
        signatures_vis(seafood,3),size,'red');
title('Signatures␣from␣training␣data'); legend('Fruits','Veggies','Seafood');
xlabel('1'); ylabel('2'); zlabel('3');

%% average signature for fruits, veggies, seafood
veg_sig = mean(signatures(1:20,:));
figure(); histogram(veg_sig,10); title('veggies␣average␣signature');
fruit_sig = mean(signatures(21:40,:));
figure(); histogram(fruit_sig,10); title('fruit␣average␣signature');
seafood_sig = mean(signatures(41:61,:));
figure(); histogram(seafood_sig,10); title('seafood␣average␣signature');
```

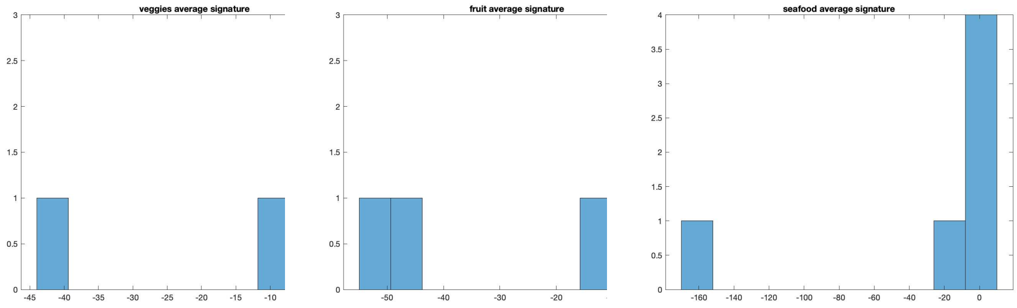
Below are the visualizations of the signatures



(A) Scatterplot (3 out of 6 elements) (B) Parallel Coordinates Plot (all elements)

Figure 2. Different visualizations of signatures

Compared to the plots of the raw training data, the signatures do appear to be more distinctly clustered. Note that the signatures of fruits and vegetables are very similar, while seafood is distinct. This makes sense given the differences in nutritional content of the three groups, but it may lead to inaccurate or low-confidence classification later on. Below are the average signatures for each food group in histograms.



(A) Vegetable signature (B) Fruit signature (C) Seafood signature

Figure 3. Histogram of each food group's signature

From these histograms we once again see that seafood is quite distinct from fruits and veggies. To test out the classification protocol of finding the difference between these average signatures and a new datapoint's signature, we will use 2 test data-points from each food group. Their signatures are below.

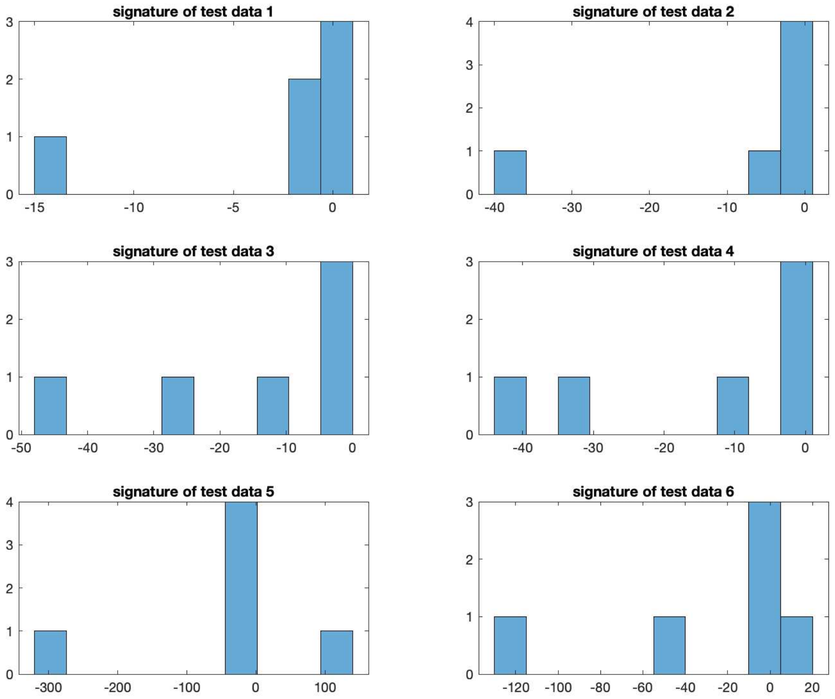


Figure 4. Signatures of the test data

From visual inspection, we can see that test data 1,2,3, and 4 are either fruits or vegetables, and that 5 and 6 must be seafood. To quantify the actual distance between the signatures of test data and the average for the food group, the following code calculates and visualized the L2 (euclidean) distances between both points:

```
distances = zeros([6,6]);

for i = 1:6
    distances(i,1) = norm(test_signatures(i,:)-veg_sig);
    distances(i,2) = norm(test_signatures(i,:)-fruit_sig);
    distances(i,3) = norm(test_signatures(i,:)-seafood_sig);
end

bar(distances); legend('veg','fruit','seafood');
title('L2 distances of training points from average signatures of each group');
```

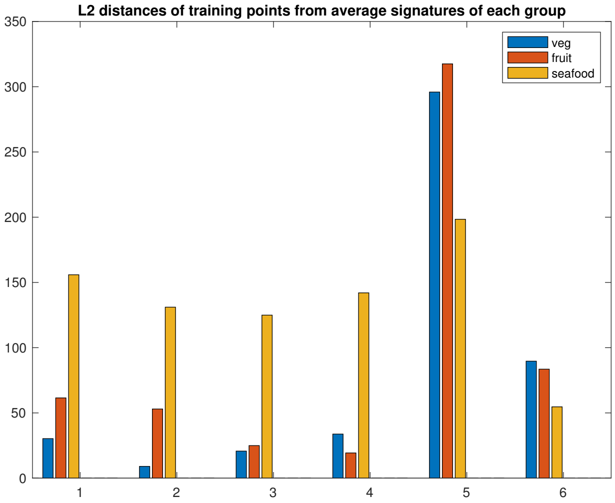


Figure 5. Differences between signatures of the test data points and each food group

The plot above shows that for a single test point, our classification of food groups based on six metrics is as simple as taking the minimum of three numbers. Test data 1 and 2 are veggies, and they are accurately classified as such using this protocol. Test data 3 and 4 are fruits, and only test data 4 is classified correctly. Test data 5 and 6 are both correctly classified as seafood. Overall the accuracy is 83.33%.