

# CS5785 Homework 3

Irene Font Peradejordi - if76, Antonio Mojena - am3238

November 14, 2019

# 1 Sentiment Analysis of Online Reviews

## 1.1 Loading Data

The data consists of 3000 online reviews from Yelp, Amazon and IMDB. Each website has 1000 reviews and a balance of 500 positive and negative reviews.

## 1.2 Preprocessing Data

Each review passed through several layers of processing to facilitate the sentiment analysis. The different layers are the following:

- Removal of stop words
  - Words such as "the", "and", etc. are neutral and do not provide any sentiment to the review
- Making every word lowercase
  - The same word can be written in either uppercase or lowercase and could cause the same word to be analyzed multiple times in different ways. This prohibits that from happening.
- Removing punctuation
  - Punctuation only add noise and intensity to the sentiment. It is not a determining factor to categorize the review as positive or negative.
- Lemmatization
  - Lemmatization allows that any word written in different conjugations or with different prefixes to be analyzed the same way (i.e. "running" and "run" would both be analyzed as run").

## 1.3 Bag of Words

In order to make a Bag of Words (BoW), the data was split into training and testing data. The first 400 positive and negative reviews of each website

was used as training data (2,400 total) and the remaining 100 positive and negative reviews (600 total) were used as testing data.

An initial iteration of the preprocessed training data was used to find all the unique words and their frequencies. You cannot do this operation on the testing data because at that point you are overfitting the data to work almost exclusively with the testing data. However, the testing data was used to count the frequencies of the unique words that were originally found in the training data. At this point, we have 2 BoWs with 3683 feature vectors representing all the unique words with their respective frequencies.

## 1.4 Postprocessing Data

Having now a BoW for both sets of data, it is necessary to carry out another processing operation to normalize the feature vectors for each review. L1 normalization was performed on these BoWs as this normalization works best with sparse data and data that contains multiple 0s. This normalizes the weight of each frequency which will provide a better sentiment analysis.

## 1.5 Sentiment Prediction

After performing all the processes previously mentioned. The data is now in shape to be analyzed in order to predict the sentiment (positive/negative) of each review. To do so, two different methods were used: Logistic Regression and Naive Bayes Classification.

### 1.5.1 Logistic Regression

Logistic Regression predicted 82.33% of all reviews correctly, producing the following confusion matrix:  $\begin{bmatrix} 260 & 40 \\ 66 & 234 \end{bmatrix}$ .

### 1.5.2 Naive Bayes Classification

Logistic Regression predicted 64.67% of all reviews correctly, producing the following confusion matrix:  $\begin{bmatrix} 263 & 37 \\ 175 & 125 \end{bmatrix}$ .

### 1.5.3 Most important feature vector per classification

Moreover, when a review was classified as positive, the word which appeared most frequently in the review was "great". In contrast, when a review was classified as negative, the word which appeared most frequently in the review was "poor".

## 1.6 N-gram Modeling

N-gram modeling, in this context, is creating a BoW but each feature vector is a combination of n contiguous words. For example, a 2-gram model would split up the sentence "Alice fell down the rabbit hole" in the following manner: "Alice fell", "fell down", "down the", "the rabbit", "rabbit hole".

A new BoW was made using this 2-gram model and the prediction was done once again using Logistic Regression and Naive Bayes.

### 1.6.1 Logistic Regression

Logistic Regression predicted 65.83% of all reviews correctly, producing the following confusion matrix:  $\begin{bmatrix} 254 & 46 \\ 159 & 141 \end{bmatrix}$ .

### 1.6.2 Naive Bayes Classification

Logistic Regression predicted 61.33% of all reviews correctly, producing the following confusion matrix:  $\begin{bmatrix} 102 & 198 \\ 34 & 266 \end{bmatrix}$ .

### 1.6.3 Most important feature vector per classification

Moreover, when a review was classified as positive, the word which appeared most frequently in the review was "work great". In contrast, when a review was classified as negative, the word which appeared most frequently in the review was "veri disappoint".

## 1.7 PCA for Bag of Words

PCA was used to reduce the dimensions of the feature vectors since both models (1-gram, 2-gram) have redundant features. The sentiment analysis

was run three times, each with different amounts of dimensions:  $r = 10, 50, 100$ . Below are the results from Logistic Regression and the Naive Bayes Classifier with the different dimensions.

### 1.7.1 $r = 10$ dimensions, BoW

Performance Comparison		
	Logistic Regression	Naive Bayes
Score	54.33 %	50.83 %
Confusion Matrix	$\begin{bmatrix} 90 & 210 \\ 64 & 236 \end{bmatrix}$	$\begin{bmatrix} 7 & 293 \\ 2 & 298 \end{bmatrix}$

Most common n-gram in positive reviews: "great"

Most common n-gram in negative reviews: "like"

### 1.7.2 $r = 50$ dimensions, BoW

Performance Comparison		
	Logistic Regression	Naive Bayes
Score	64.33 %	50.33 %
Confusion Matrix	$\begin{bmatrix} 164 & 136 \\ 78 & 222 \end{bmatrix}$	$\begin{bmatrix} 113 & 187 \\ 111 & 189 \end{bmatrix}$

Most common n-gram in positive reviews: "great"

Most common n-gram in negative reviews: "bad"

### 1.7.3 $r = 100$ dimensions, BoW

Performance Comparison		
	Logistic Regression	Naive Bayes
Score	68.50 %	59.50 %
Confusion Matrix	$\begin{bmatrix} 186 & 114 \\ 75 & 225 \end{bmatrix}$	$\begin{bmatrix} 107 & 193 \\ 50 & 250 \end{bmatrix}$

Most common n-gram in positive reviews: "great"

Most common n-gram in negative reviews: "bad"

For the bag of words it is clear that logistic regression outperforms the Naive Bayes Classification for each of the dimensions tested.

#### 1.7.4 $r = 10$ dimensions, N-gram model

Performance Comparison		
	Logistic Regression	Naive Bayes
Score	49.33 %	50.67 %
Confusion Matrix	$\begin{bmatrix} 296 & 4 \\ 300 & 0 \end{bmatrix}$	$\begin{bmatrix} 276 & 24 \\ 272 & 28 \end{bmatrix}$

Most common n-gram in positive reviews: "work great"

Most common n-gram in negative reviews: "year old"

#### 1.7.5 $r = 50$ dimensions, N-gram model

Performance Comparison		
	Logistic Regression	Naive Bayes
Score	50.83 %	52.00 %
Confusion Matrix	$\begin{bmatrix} 285 & 15 \\ 280 & 20 \end{bmatrix}$	$\begin{bmatrix} 269 & 31 \\ 257 & 43 \end{bmatrix}$

Most common n-gram in positive reviews: "highly recommend"

Most common n-gram in negative reviews: "wast time"

#### 1.7.6 $r = 100$ dimensions, N-gram model

Performance Comparison		
	Logistic Regression	Naive Bayes
Score	53.83 %	54.17 %
Confusion Matrix	$\begin{bmatrix} 277 & 23 \\ 254 & 46 \end{bmatrix}$	$\begin{bmatrix} 267 & 33 \\ 242 & 58 \end{bmatrix}$

Most common n-gram in positive reviews: "work great"

Most common n-gram in negative reviews: "wast time"

For the n-gram models the Naive Bayes Classifier slightly outperformed the Logistic Regression for all dimensions. It is worth noting that in most dimensions, both performed slightly better than chance.

## 2 Clustering for Text Analysis

### 2.1 K-means

In order to cluster documents into  $k$  clusters we used K-means. In order to do this, obviously, a  $k$  is needed. To choose  $k$ , K-means was ran with 10 iterations for each  $k$  between 2 and 20 and the Sum of Squared Errors (SSE) was calculated and plotted.

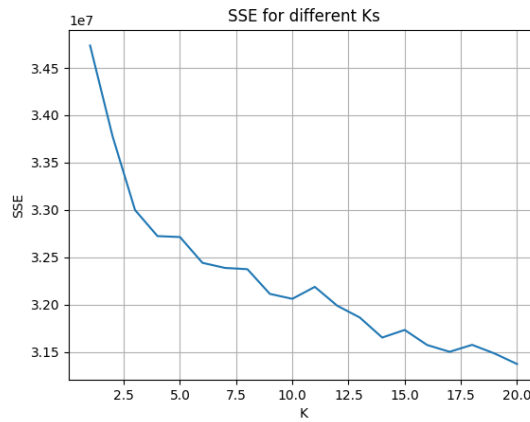


Figure 1: Sum of Square Errors for each K

By inspection of the graph,  $k = 10$  was chosen to do the clustering.

### 2.2 Results with $k = 10$

#### 2.2.1 Document Clustering

Cluster 1:

- Observation of X-ray Lines from a Gamma-Ray Burst (GRB991216): Evidence of Moving Ejecta from the Progenitor
- Upwelling Intensification as Part of the Pliocene-Pleistocene Climate Transition
- $^{92}\text{Nb} - ^{92}\text{Zr}$  and the Early Differentiation History of Planetary Bodies

- Multidecadal Changes in the Vertical Temperature Structure of the Tropical Troposphere
- Lunar Impact History from  $^{40}\text{Ar}/^{39}\text{Ar}$  Dating of Glass Spherules
- Support for the Lunar Cataclysm Hypothesis from Lunar Meteorite Impact Melt Ages
- Extinct  $^{129}\text{I}$  in Halite from a Primitive Meteorite: Evidence for Evaporite Formation in the Early Solar System
- Cool Glacial Temperatures and Changes in Moisture Source Recorded in Oman Groundwaters
- Suppression of Rain and Snow by Urban and Industrial Air Pollution
- Evidence That the Reactivity of the Martian Soil Is Due to Superoxide Ions

Cluster = 2:

- Dimer Preparation That Mimics the Transition State for the Adsorption of  $\text{H}_2$  on the Si(100)-2 x 1 Surface
- All-Optical Magnetic Resonance in Semiconductors
- Molybdenum Nanowires by Electrodeposition
- Imaging Coherent Electron Flow from a Quantum Point Contact
- Physics of Iron at Earth's Core Conditions
- Defect-Induced Phase Separation in Dipolar Fluids
- Generating Solitons by Phase Engineering of a Bose-Einstein Condensate
- Suppressing Wall Turbulence by Means of a Transverse Traveling Wave
- Carrier-Envelope Phase Control of Femtosecond Mode-Locked Lasers and Direct Optical Frequency Synthesis



- $O_2$  Activation by Nonheme Iron Complexes: A Monomeric Fe(III)-Oxo Complex Derived from  $O_2$

Cluster = 3:

- Activation of the DNA Replication Checkpoint through RNA Synthesis by Primase
- Survival for Immunity: The Price of Immune System Activation for Bumblebee Workers
- Regulation of Antigen-Specific  $CD8^+$  T Cell Homeostasis by Perforin and Interferon- $\gamma$
- Sex Determination in Malaria Parasites

Cluster = 4:

- Atomic Structure of PDE4: Insights into Phosphodiesterase Mechanism and Specificity
- Structure of Murine CTLA-4 and Its Role in Modulating T Cell Responsiveness
- The Complete Atomic Structure of the Large Ribosomal Subunit at 2.4 Å Resolution
- Adaptive Recognition by Nucleic Acid Aptamers
- Convergent Solutions to Binding at a Protein-Protein Interface
- Topologically Linked Protein Rings in the Bacteriophage HK97 Capsid
- The Structural Basis of Ribosome Activity in Peptide Bond Synthesis
- Twists in Catalysis: Alternating Conformations of Escherichia coli Thioredoxin Reductase
- Structure of the Cytoplasmic  $\beta$  Subunit: T1 Assembly of Voltage-Dependent  $K^+$  Channels

- Structure of the S15,S6,S18-rRNA Complex: Assembly of the 30S Ribosome Central Domain

Cluster = 5:

- Global Biodiversity Scenarios for the Year 2100
- Rapid Extinction of the Moas (Aves: Dinornithiformes): Model, Test, and Implications
- Parental Care and Clutch Sizes in North and South American Birds
- Harnessing the Power of the Genome in the Search for New Antibiotics
- U.S. Soil Erosion Rates: Myth and Reality
- Greenhouse Gases in Intensive Agriculture: Contributions of Individual Gases to the Radiative Forcing of the Atmosphere
- Contribution of Increasing  $CO_2$  and Climate to Carbon Storage by Ecosystems in the United States
- Coherence and Conservation
- Contributions of Land-Use History to Carbon Accumulation in U.S. Forests
- Spatial Patterns in the Distribution of Tropical Tree Species

Cluster = 6:

- Pattern of Marine Mass Extinction near the Permian-Triassic Boundary in South China
- Altered River Morphology in South Africa Related to the Permian-Triassic Extinction

Cluster = 7:

- Hurricane Disturbance and Tropical Tree Species Diversity

- Selectivity for 3D Shape That Reveals Distinct Areas within Macaque Inferior Temporal Cortex
- Quantitative Imaging of Lateral ErbB1 Receptor Signal Propagation in the Plasma Membrane
- Rapid Evolution of Reproductive Isolation in the Wild: Evidence from Introduced Salmon
- Monolithic Microfabricated Valves and Pumps by Multilayer Soft Lithography
- Evidence for a Low-Density Universe from the Relative Velocities of Galaxies
- A Simple Model for Complex Dynamical Transitions in Epidemics
- Molecular Architecture and Evolution of a Modular Spider Silk Protein Gene
- The Bacterial Flagellar Cap as the Rotary Promoter of Flagellin Self-Assembly
- A Quantum State-Resolved Insertion Reaction:  $O(^1D) + H_2(J=0) \rightarrow OH(^2\Pi, v, N) + H(^2S)$

Cluster = 8:

- Stalking the Wild Mustard
- Superbugs on the Hoof?
- Against All Odds, Victories from the Front Lines
- The Quiet Revolution: Biodiversity Informatics and the Internet
- Reaping the Plant Gene Harvest
- On the Hunt for a Wolf in Sheep's Clothing
- Something to Be Done: Treating HIV/AIDS

- Can Celera Do It Again?
- Does Science Drive the Productivity Train?
- Will Livermore Laser Ever Burn Brightly?

Cluster = 9:

- Dual Signaling Regulated by Calcyon, a D1 Dopamine Receptor Interacting Protein
- Fluorescent Timer: Protein That Changes Color with Time
- X-Chromosome Inactivation in Cloned Mouse Embryos
- Rapid Destruction of Human Cdc25A in Response to DNA Damage
- Regulation of JNK by Src during Drosophila Development
- Transport of Peptide: MHC Class II Complexes in Developing Dendritic Cells
- Genome-Wide Location and Function of DNA Binding Proteins
- The Ins and Outs of Body Surface Immunology
- Development of CD8a-Positive Dendritic Cells from a Common Myeloid Progenitor
- Gatekeepers of the Nucleus

Cluster = 10:

- Turning Attraction into Repulsion
- Corrections and Clarifications: Commercialization of Genetic Research and Public Policy
- Corrections and Clarifications: First-Principles Determination of Elastic Anisotropy and Wave Velocities of MgO at Lower Mantle Conditions

- Regulating Export of ER Cargo
- Susceptibility Locus for Alzheimer's Disease on Chromosome 10
- Science's Conversation with Jiang Zemin
- Biological Control of Invading Species: Risk and Reform
- National Academy of Sciences Elects New Members
- Glacial Climate Instability
- Genomic Medicine and the Future of Health Care

The algorithm seems to have clustered documents that are similar in topic or by topics that use similar vocabulary. This type of algorithm is useful if you are searching for literature by topic or tags and not necessarily by author, publisher, title, etc.

### **2.2.2 Vocabulary Clustering**

k = 0: started, recalls, hold, needs, die, raise, improve, scientist, goes, european

k = 1: reflectivity, kv, doxy, aptamers, lcts, rory, ist2p, noxa, snake, ultimate

k = 2: displacement, determination, charged, consisting, metals, clearly, walls, hydrogen, degrees, application

k = 3: biochem, three, specificity, proc, putative, residue, cooh, references, catalytic, complexes

k = 4: offers

k = 5: initial, number, size, high, two, indicates, surface, resolution, increased, changes

k = 6: presumably, biochemical, nine, chromosomes, developmental, transporter, subset, assess, rat, heterozygous

k = 7: cdna, assays, showed, stained, blot, deficient, mutant, intracellular, anti, type

k = 8: pacific, planetary, anthropogenic, lunar, documented, rates, natural, holocene, mixing, yields

k = 9: complex

This is a less useful algorithm than the document clustering. Vocabulary clustering would be useful if you knew what the topic was but some grouping could be ambiguous. For example, there is no clear topic in cluster 1 that you can determine from the combination of words.

Clustering terms is different from clustering documents because terms are not necessarily indicative of the context. Documents have a title and are composed of hundreds or thousands of words whereas a term is just a combination of letters. While grouping some terms might suggest a topic, it is not as clear as grouping documents together. Therefore, the application of this algorithm is important because you cannot conclude or interpret much of the results if it is applied on terms.

## 3 EM Algorithm and Implementation

### 3.1 E-Step and M-Step

K-means is a hard cluster method, whereas the Expectation Maximization (EM) algorithm is the soft version of K-means.

In K-means all points are assigned to belong to one cluster. In the case of  $k = 2$ , each point will belong to either 0 or 1. However, in the EM algorithm, we compute a responsibility for each point, which is the likelihood of this point belonging to a certain cluster. These lie in a continuum from 0 to 1.

In order to converge, the EM algorithm alternates between two steps. These are the E-step and the M-step. The E-step is in charge of computing each point's responsibilities. The M-step will compute the weighted means and the variance. Then, it calculates a new mean and new responsibilities will be extracted (E-step again). In short, both K-means and EM are clus-

tering methods which run iteratively to find the best congestion. However, while EM uses statistical methods to achieve convergence, the K-means only computes Euclidean distances between each point and the means.

### 3.2 2D plot of the Old Faithful Geyser Dataset

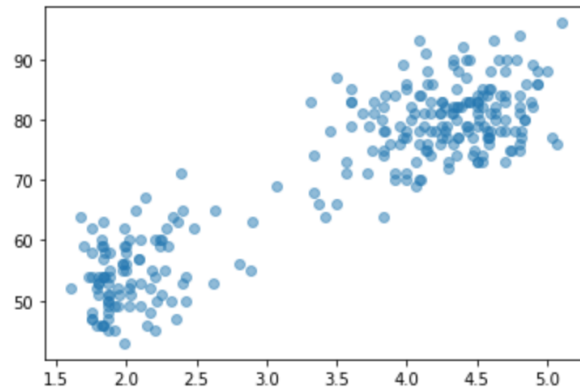


Figure 2: Old Faithful Geyser Dataset plot.

### 3.3 Implementation of a GMM model

#### 3.3.1 Plot the trajectories of two mean vectors

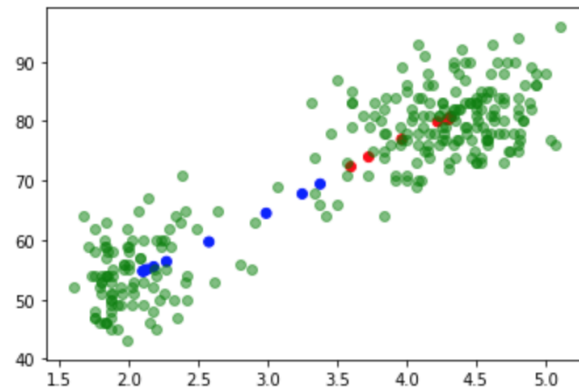


Figure 3: Trajectories of the two mean vectors. Coordinates vs. iteration.

#### 3.3.2 Distribution number of iterations needed for convergence

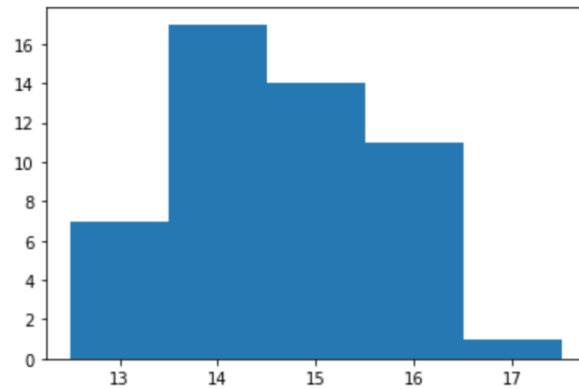


Figure 4: Distribution of the total number of iterations needed for algorithm for 50 runs.



### 3.4 GMM initialized with K-means

This time, the GMM was initialized with the parameters extracted from a K-means. As seen before, when the initial parameters are random, the GMM algorithm takes between 13 and 17 iterations to converge, whereas when the parameters are initialized using a K-means, it takes only 2 iterations each time to converge.

During these two iterations the means do not change much (Figure 5), as the initial value is already a good estimate.

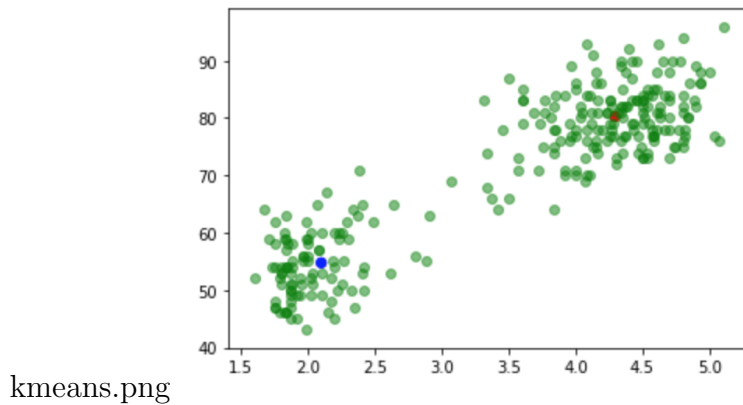


Figure 5: Trajectories of the two mean vectors. GMM initialized with K-means. Coordinates vs. iteration.

## 4 Multidimensional scaling for genetic population differences

### 4.1 MDS 2-dimensional vector

The matrix is giving us the Euclidean distance between each population. These populations are physical spaces situated on earth, which is round. Therefore, when trying to plot these points in a 2D space we will face an impossibility. The pairwise distances are not affected by the rotation, translation or reflection of the surface, thus we don't have that information available.

Determining how many dimensions are needed to capture most of the variance

To identify the optimal number of dimensions we need to capture most of the variance of the data we used SVD. The singular values of a matrix (extracted using SVD) tell us the relevance of each dimension in terms of the amount of variance they capture. Consequently, if we can find singular values = to 0, then, this dimension is irrelevant.

Following this logic, we did MDS to our initial matrix, reducing it to a certain number of dimensions, from 2 to 41. Then, we computed the SVD of each of the 40 resulting matrix and looked at the singular values and then we set a threshold of 0.01. We started seeing singular values smaller than 0.01 in our MDS reduction to 18 dimensions. Therefore, we can conclude that, given this set threshold, 17 dimensions is the optimal number to reduce our matrix to.

Dimensions    N° of singular values smaller than 0.01

2	0
3	0
4	0
5	0
6	0
7	0
8	0
9	0
10	0
11	0
12	0
13	0
14	0
15	0
16	0
17	0
18	1
19	1
20	3
21	4
22	5
23	5
24	6
25	7
26	8
27	8
28	9
29	11
30	11
31	13
32	14
33	14
34	16
35	17
36	18
37	19
38	19
39	20
40	22
41	23

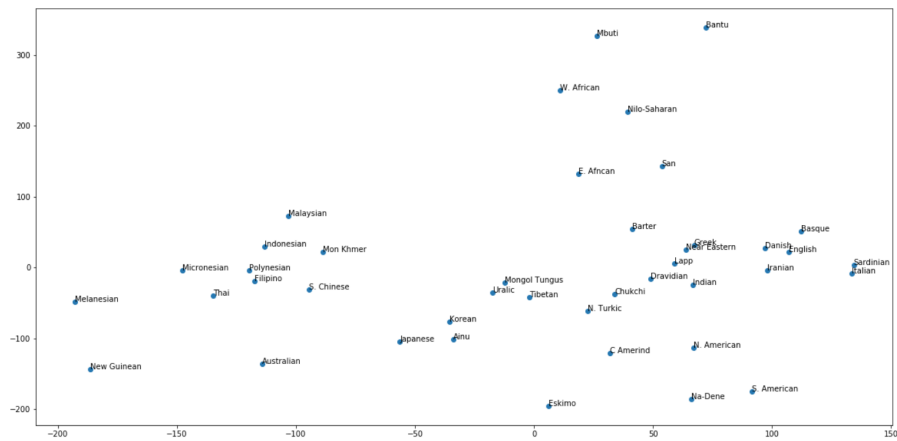


Figure 6: Plot of the distance matrix in two dimensions after MDS.

## 4.2 K-means on 2D embedding

K = 5 was selected using the Elbow method.

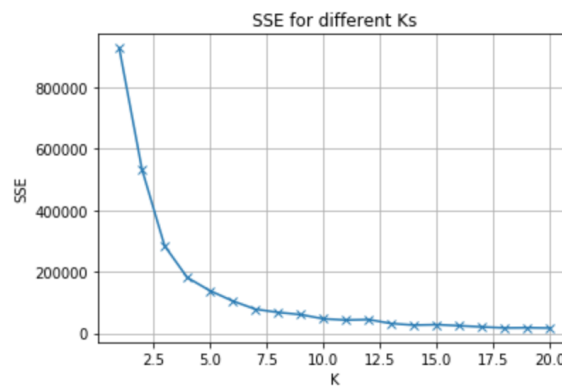


Figure 7: The Elbow method for K-means.

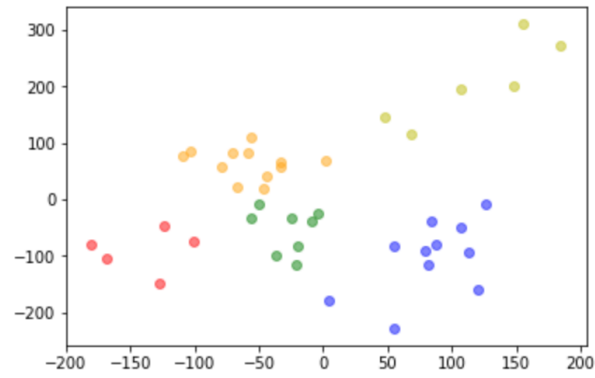


Figure 8: K-means with  $K = 5$

### 4.3 Comparing hierarchical clustering with K-Means

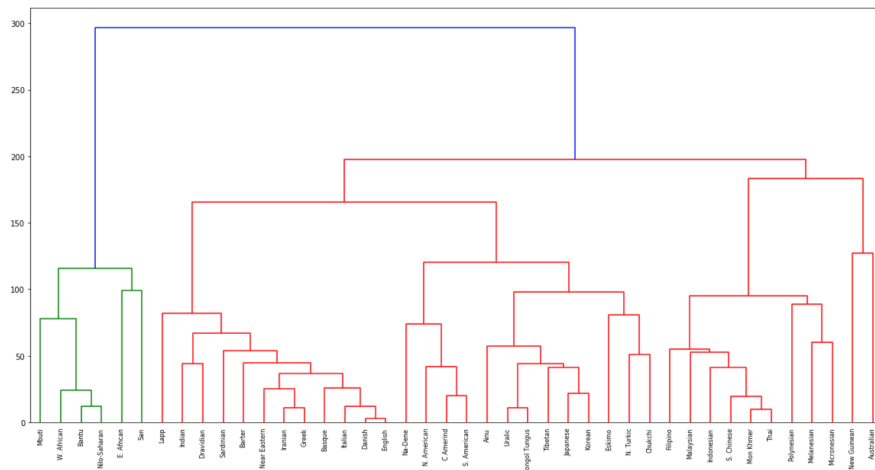


Figure 9: Hierarchical clustering - Dendrogram

Both of the clusters are very similar. The plot are inverted, however, this is not significant.

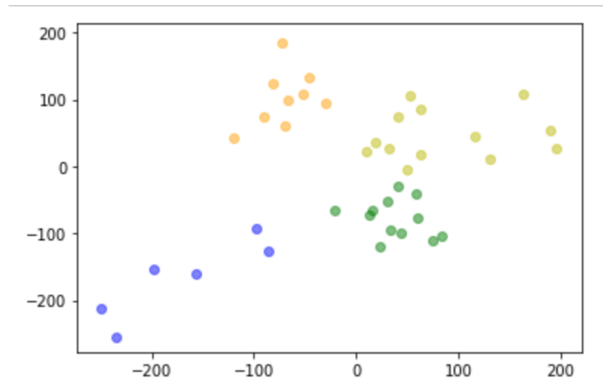


Figure 10: Visualizing the resulting clusters after cutting of the tree using 150 as threshold.

#### 4.4 Compare k-medoids with k-means

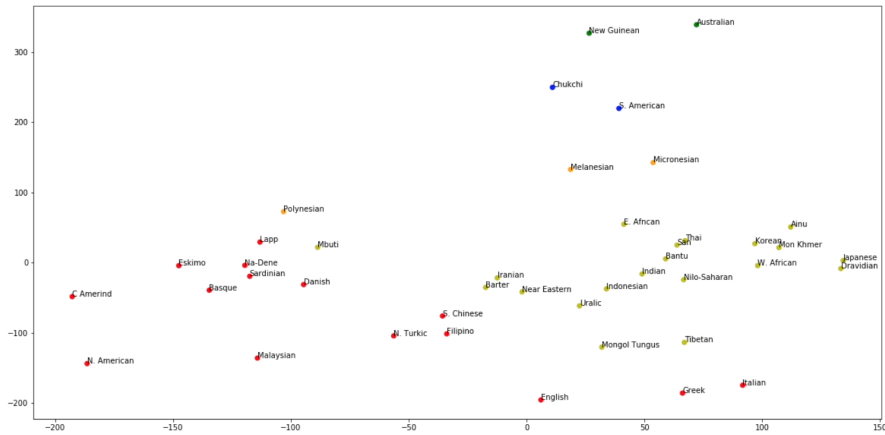


Figure 11: K-medoid clustering

Comparing both the K-means and the K-medoids clustering we can see that K-means does a much better job. K-medoids clusters all the data points in the bottom of the plot together even though they are closer in distance to other center means. The population groups created by K-means make much more sense.

## 5 Written Exercises

### 5.1 Decision Trees

#### 5.1.1 Weighted min-error impurity

Given the min-error impurity function:

$$I(r) = r, 1 - r$$

and the assumption that  $\max(p_1, n_1) = p_1$  and  $\max(p_2, n_2) = p_2$ . The weighted min-error impurity function is as follows:

$$\begin{aligned} & (p_1 + n_1) \cdot I\left(\frac{p_1}{p_1 + n_1}\right) + (p_2 + n_2) \cdot I\left(\frac{p_2}{p_2 + n_2}\right) \\ I(r) &= \min\{r, 1 - r\} \\ I\left(\frac{p_r}{p_r + n_r}\right) &= \min\left\{\frac{p_r}{p_r + n_r}, \frac{1 - p_r}{p_r + n_r}\right\} = \min\left\{\frac{p_r}{p_r + n_r}, \frac{n_r}{p_r + n_r}\right\} \\ &= \frac{n_r}{p_r + n_r} \\ (p_1 + n_1) \cdot I\left(\frac{p_1}{p_1 + n_1}\right) &+ (p_2 + n_2) \cdot I\left(\frac{p_2}{p_2 + n_2}\right) \rightarrow \cancel{(p_1 + n_1)} \left(\frac{n_1}{\cancel{p_1 + n_1}}\right) + \cancel{(p_2 + n_2)} \left(\frac{n_2}{\cancel{p_2 + n_2}}\right) \\ &\rightarrow n_1 + n_2 \end{aligned}$$

Figure 12: Weighted min-error impurity function

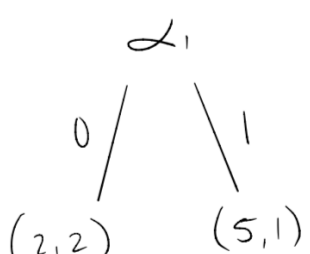
### 5.1.2 Splitting Trees

Given that Gini Index =  $\sum_{k=1}^K \hat{p}_{mk}(1 - \hat{p}_{mk})$ , the split of the root of the tree will be the index with the lowest value.

$$\text{Gini Index: } \sum_{k=1}^K \hat{p}_{mk}(1 - \hat{p}_{mk})$$

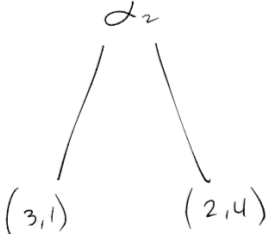
$$\mathcal{L}_1 = 2\left(\frac{2}{4}\right)\left(\frac{2}{4}\right) + 2\left(\frac{5}{6}\right)\left(\frac{1}{6}\right)$$

$$= .5 + .28$$

$$= .78$$


$$\mathcal{L}_2 = 2\left(\frac{3}{4}\right)\left(\frac{1}{4}\right) + 2\left(\frac{2}{6}\right)\left(\frac{4}{6}\right)$$

$$= .375 + .444$$

$$= .82$$


$$\mathcal{L}_3 = 2\left(\frac{3}{7}\right)\left(\frac{4}{7}\right) + 2\left(\frac{2}{3}\right)\left(\frac{1}{3}\right)$$

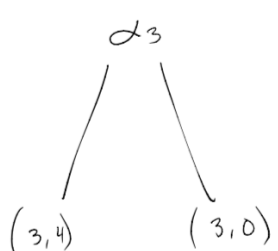
$$= .49$$


Figure 13: Tree splitting based on Gini Index

This means that  $\alpha_3$  would be chosen to make the split.



Using the min-error impurity function mentioned above, we would get the following errors:

$$\begin{array}{l} \mathcal{I}(r) = \min \{r, 1-r\} \\ \alpha_1 : 2+1 = 3 \\ \alpha_2 : 1+2 = 3 \\ \alpha_3 : 3+0 = 3 \end{array} \quad \Bigg|$$

Figure 14: Min-error impurity

On this metric, each split is equally likely to be chosen.

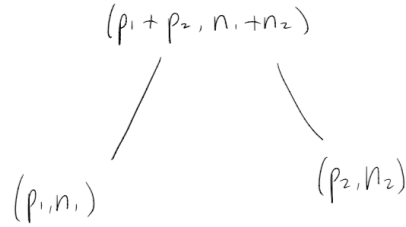
### 5.1.3 Weighted min-error impurity vs Min-error impurity

For the weighted min-error impurity of the split be strictly smaller than the min-error impurity before making the split we need to consider 2 cases:

Case 1:  $p_1 < n_1$  ,  $p_2 > n_2$

Case 2:  $p_1 > n_1$  ,  $p_2 < n_2$

The weighted min-error impurity is calculated in the image below:



2 cases to consider : 1)  $p_1 < n_1$  &  $p_2 > n_2$   
 2)  $p_1 > n_1$  &  $p_2 < n_2$

$$1) (p_1 + n_1) \cdot \left( \frac{p_1}{p_1 + n_1} \right) \cdot (p_2 + n_2) \cdot \left( \frac{n_2}{p_2 + n_2} \right) \\ = p_1 + n_2$$

$$2) (p_1 + n_1) \cdot \left( \frac{n_1}{p_1 + n_1} \right) \cdot (p_2 + n_2) \cdot \left( \frac{p_2}{p_2 + n_2} \right) \\ = n_1 + p_2$$

Figure 15: Minerror

In these two cases, the weighted min-error impurity of the split be strictly smaller than the min-error impurity before making the split.

### 5.1.4 Suitability of min-error impurity

As we saw when comparing the Gini Index with the min-error impurity, the latter did not give a clear indicator of what a good split for a tree should be. Therefore, min-error is not well suited since choosing a tree by chance increases the chance of misclassification.

## 5.2 Bootstrap aggregation

\* For  $n=2$   
① ②  $\rightarrow$  probability of excluding one is  $\frac{1}{2}$   
thus,  $\frac{1}{2} * \frac{1}{2} = \left(\frac{1}{2}\right)^2$

\* For  $n=4$   
① ② ③ ④  $\rightarrow \frac{3}{4} * \frac{3}{4} * \frac{3}{4} * \frac{3}{4} = \left(\frac{3}{4}\right)^4$

\* For  $n=n$   $\rightarrow \left(\frac{n-1}{n}\right)^n$

LIMITS

$$N \rightarrow \infty$$

$$\lim_{N \rightarrow \infty} \left(\frac{N-1}{N}\right)^N = \lim_{N \rightarrow \infty} e^{\ln \left(\frac{N-1}{N}\right)^N}$$

$$\lim_{N \rightarrow \infty} \ln \left(\frac{N-1}{N}\right)^N = -1$$

therefore  $\lim_{N \rightarrow \infty} \left(\frac{N-1}{N}\right)^N = e^{-1} = \frac{1}{e}$

$$\frac{1}{e} = 0.367 \rightarrow 36.7\%$$

36.7% of the training data will be excluded from the bootstrapping replicate if  $N$  is big enough.

## 6 Sources

Elements of Statistical Learning Andrew Tulloch <https://tullo.ch/static/ESL-Solutions.pdf> [https://waxworksmath.com/Authors/G\\_M/Hastie/WriteUp/Weatherwax\\_Epstein\\_H](https://waxworksmath.com/Authors/G_M/Hastie/WriteUp/Weatherwax_Epstein_H)