# FISHER LDA FOR MULTIMODAL INFORMATION RETRIEVAL

## SMAI PROJECT REPORT
## TEAM SMAILE

Sreenya Chitluri
Tejah S S
Sankeerthana Venugopal
M Krishna Praneet

SECTION 1

# INTRODUCTION

# INTRODUCTION - WHAT IS INFORMATION RETRIEVAL

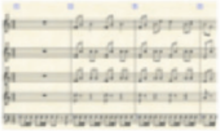
- With the advent of modern technology, there has been an unprecedented amount of multimedia data available (text, photos, videos, etc).
- In such a large ocean of data, searching for relevant data for a purpose is extremely difficult.
- This large amount of data creates a strong desire for efficient multimedia information retrieval systems able to search multimedia documents relevant to the information need.

# MULTIMODAL INFORMATION RETRIEVAL

- There exist many methods which can give relevant search results with a query of a single type (ex. text search)
- But it has been observed that multimodal approach gives better results in comparison.
- It is a system where the documents stored consist of multiple different types (modes) of data, and as such, the query also has multiple types of data.

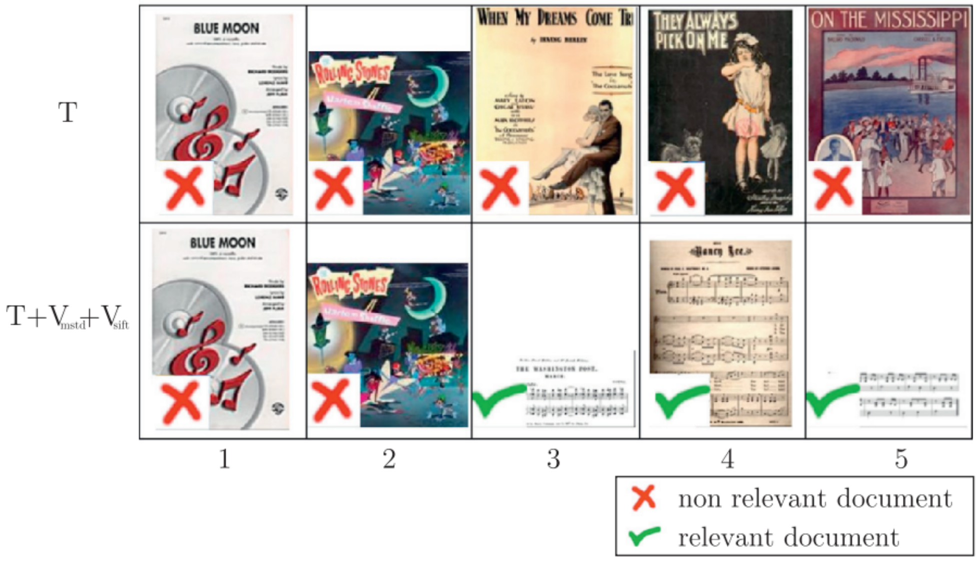
# EXAMPLES

query : red fruit + +

query : notes on music sheet + 

# GOALS

query : notes on music sheet +



# STRUCTURE

To implement what has been described above, we have divided the whole procedure into the following sections -

1. Bag of words
2. Feature extraction and representation
3. Fisher LDA

SECTION 2

# BAG OF WORDS

# BAG OF WORDS - INTRODUCTION

- The bag-of-words model is a way of representing text (or any other form of data) when modeling the data with machine learning algorithms.
- Essentially, it is a data pre-processing model, used for feature extraction.

# WHY USE IT?

Machine Learning algorithms prefer fixed-length inputs and outputs. Often, raw data comes in various shapes and sizes. Bag-of-words is just one of the ways to process the data to standardize the size of the datapoints and identify and select important features.

# MORE ON BAG OF WORDS

Taking the example of text data, we can say that bag-of-words is a representation that describes the occurrence of words within a document. It involves two things:

- Defining a **vocabulary** of known words
- A **measure of the presence** of these known words

It is called a 'bag', because we discard the information regarding the ordering of these words in this model. We only want to know if a certain word occurs in a document, not where in the document.

# DEFINING A VOCABULARY

- We call each datapoint a 'document' in this context.
- The vocabulary for a set of documents (which make up our dataset) is defined as the set of all unique 'words' present across all the documents.
- It is a common practice to ignore frequent words that do not contain much information ('the' for example).
- It is also common to merge very similar words into a single word.
- The size of the vocabulary defines the size of the input vector obtained from a document (datapoint).

# MEASURE OF PRESENCE

- Once we have defined a vocabulary, we need to know how frequently each of the words in the occur in each document.

- Since the aim of Information Retrieval is to decide how relevant a certain document is to the query, we need a metric that can give us a sense of what is contained in a document.

- A measure of the frequency of the words in the vocabulary is a decent indicator of the type of information contained in that document, helping us determine its relevancy to a query.

# SIMPLE FREQUENCY CONS

- While it is true that we need a measure of the frequency of vocabulary words in a document, simple frequency of these words is unfortunately of not much use in most cases.

- There might be words that occur very frequently across a large number of documents. Such words have less 'information content', and are less useful in determining the relevancy of a document with a query.

# OUR SYSTEM

- Now that we've seen that we can define a vocabulary for a set of documents, we have a basis for representing the data in a vector form.
- We use the Term frequency and Inverse document frequency we defined below to represent every document in the dataset as a vector.
- The number of components of the vector is the number of words in the vocabulary.
- Each component of the vector is a weight that is defined as the product of the tf and idf of the document.
- In our implementation, the dataset contains text and images.

SECTION 3

# FEATURE EXTRACTION AND REPRESENTATION

# FEATURE EXTRACTION AND REPRESENTATION

- Three modalities are considered to represent all the features present in the input data
- **Textual Representation**: Uses the literal bag of words approach to generate vocabularies
- **V_sift:** Contains the texture information denoted as words
- **V_mstd:** Contains the color information denoted as words

# TEXTUAL REPRESENTATION

•Let the vocabulary of the text in the documents dataset be $T = \{t_1, \dots, t_j, \dots, t_{|T|}\}$
•A document in the dataset is represented as a vector of weights

$$\vec{d_i} = (w_{i,1}, \dots, w_{i,j}, \dots, w_{i,|T|})$$

•Here, $w_{i,j} = (tf_{i,j}).(idf_j)$, where $tf_{i,j}$ is the term frequency of the jth word in the ith document and $idf_j$ is the inverse document frequency of the jth word in the vocabulary.

•We have now successfully represented the textual part of our multimedia document as a vector!

# VISUAL REPRESENTATION

$V_{mstd}$

•It is and indicator of the colour of the cell.
•It calculates a total of 6 features : mean and variance of each of these values

$V_{sift}$

- Represents the texture information of each cell.
- The sift description converts each cell into a 128-dimensional vector.

- Each image is sectioned into 16x16 subsections
- Vsift and Vmstd are calculated for each of these sections
- It is followed by k means clusterin to form k distinct visual "words" separately for both Vsift and Vmstd.
- They are then clustered to form visual "words" to form its own vocabulary

SECTION 4

# SCORING

# OUR IMPLEMENTATION

### Term Frequency

Term Frequency is a scoring of the frequency of the word in the current document.

$$tf_{i,j} = \frac{k_1 n_{i,j}}{n_{i,j} + k_1 \left( 1 - b + b \frac{|d_i|}{d_{avg}} \right)}$$

Here, *k1* and *b* are constants. *ni,j* is the number of occurrences of the term  in the document *di*. |*di*| is the size of the document, *davg* is the average size of all the documents

### Inverse Document Frequency

Inverse Document Frequency is a scoring of how rare the word is across documents.
As we discussed before, words that occur frequently in many documents are not very helpful. We therefore seek to penalize them when counting their frequency.

$$idf_j = \log \frac{|\mathcal{D}| + 1}{|\mathcal{D}_j| + 0.5}$$

Where |*D*| is the size of the corpus (number of all documents). |*Dj*| is the number of documents where *tj* occurs at least once.

SECTION 5

# FISHER LINEAR DISCRIMINANT ANALYSIS

# FISHER LDA

Say we have X as feature vectors and y as their labels (for now two classes). Let W be the direction of projection.

**AIM**: Find such that the separability of classes is maximized upon projection onto that direction.

$$z_i = W^T Xi$$

$$M_0 = \frac{1}{n_0} \sum_{X_i \in C_0} Xi \qquad M_1 = \frac{1}{n_1} \sum_{X_i \in C_1} Xi$$

$$m_0 = W^T M_0 \qquad m_1 = W^T M_1$$

$$s_0^2 = \sum_{X_i \ \ inC_0} (W^T X_i - m_0)^2$$

$$s_1^2 = \sum_{X_i \ \ inC_1} (W^T X_i - m_1)^2$$

To maximize the separability between classes, we define a function that needs to be maximized. Given below –

$$J(\boldsymbol{w}) = \frac{(m_1 - m_0)^2}{s_0^2 + s_1^2}$$

$$\frac{(m_1 - m_0)^2}{s_0^2 + s_1^2} = \frac{(W^T M_1 - W^T M_0)^2}{\sum_{X_i \in C_0} [W^T(X_i - M_0)]^2 + \sum_{X_i \in C_1} [W^T(X_i - M_1)]^2}$$

$$= \frac{W^T (M_1 - M_0)(M_1 - M_0)^T W}{W^T [\sum_{X_i \in C_0} (X_i - M_0)(X_i - M_0)^T + \sum_{X_i \in C_1} (X_i - M_1)(X_i - M_1)^T] W}$$

Defining SB and SW where SB is the between class scatter matrix and SW is the within class scatter matrix. Hence we are trying to maximize. For finding the optima, we differentiate and equate to zero

$$\frac{2S_B W}{W^T S_W W} - \frac{W^T S_B W}{(W^T S_W W)^2} 2S_W W = 0$$

$$S_W W = \lambda S_B W$$

$$W = S_W^{-1} S_B W$$

**From this we get our required result** $\longrightarrow W = S_W^{-1}(M_1 - M_0)$

# RESOLVING THE PROBLEM WITH F-LDA

The aim is to determine the fusion parameters for the linear combination of the scores corresponding to each vocabulary to best separate the two classes Given a score vector x and coefficient vector z, the linear discriminant function is given by Z

$$z = \boldsymbol{z}^T \boldsymbol{x}$$

$$\boldsymbol{z}^{\boldsymbol{T}} \boldsymbol{x} = \sum_{j=1}^{|M|} \alpha_j \chi_j$$

# SOME USEFUL METRICS TO PROCEED

$$\boldsymbol{T} = \frac{1}{|x|} \sum_{l=1}^{|x|} (\boldsymbol{x_l} - \boldsymbol{\mu})^T (\boldsymbol{x_l} - \boldsymbol{\mu}) \longrightarrow \text{Covariance matrix}$$

$$\boldsymbol{B} = \frac{1}{|x|} \Big( |\chi_R| (\boldsymbol{\mu_R} - \boldsymbol{\mu})^T (\boldsymbol{\mu_R} - \boldsymbol{\mu}) + \big( |\chi_{\bar{R}}| (\boldsymbol{\mu_{\bar{R}}} - \boldsymbol{\mu})^T (\boldsymbol{\mu_{\bar{R}}} - \boldsymbol{\mu}) \big) \Big) \longrightarrow \text{Between class scatter matrix}$$

$$\boldsymbol{W} = \textstyle\sum_{x_l \in x_R} (\boldsymbol{x_l} - \boldsymbol{\mu_R})^T (\boldsymbol{x_l} - \boldsymbol{\mu_R}) + \sum_{x_l \in x_{\bar{R}}} (\boldsymbol{x_l} - \boldsymbol{\mu_{\bar{R}}})^T (\boldsymbol{x_l} - \boldsymbol{\mu_{\bar{R}}}) \longrightarrow \text{Within class scatter matrix}$$

Here,

$$\boldsymbol{\mu} = \frac{1}{|x|} \sum_{l=1}^{|x|} \boldsymbol{x_l} \qquad \boldsymbol{\mu_R} = \frac{1}{|x_R|} \sum_{x_l \in x_R} \boldsymbol{x_l} \qquad \boldsymbol{\mu_{\bar{R}}} = \frac{1}{|x_{\bar{R}}|} \sum_{x_l \in x_{\bar{R}}} \boldsymbol{x_l}$$

According to Fisher-LDA, the optimal discriminant function z can be obtained by the maximization of the Fisher criterion

$$F(\mathbf{z}) = \frac{\mathbf{z}^T \mathbf{B} \mathbf{z}}{\mathbf{z}^T \mathbf{W} \mathbf{z}}$$

We want to have a large separation between the classes relative to their individual variance.

The numerator gives us an idea of the separation between samples of the two classes after projecting it onto z

The denominator captures the variances within the two classes

Also note that $F$( )is scale independent.

## FINAL RESULT USING FISHER LDA

For the case of a two class problem The optimal fusion vector can be given by,

$$\mathbf{z} = \mathbf{W}^{-1}(\boldsymbol{\mu}_{\mathbf{R}} - \boldsymbol{\mu}_{\overline{\mathbf{R}}})$$

SECTION 6

# PROPOSED ACCURACY MEASURES

# MAP

- Precision is a measure of correctness of predictions
- Average Precision(AP) is the average of precision values over varying parameters of the evaluation criterion (E.g., IOU)
- Mean average precision(MAP) is the average of AP values over all data points.
- Ranges from 0-1, higher is better.

# R

- This is called the recall.
- This is the ratio of number of relevant retrieved documents to number of relevant documents to retrieve.
- Ranges from 0-1, higher is better.

SECTION 7

# CHALLENGES FACED

- A major challenge faced on the way was that the dataset that we were initially working with - imageCLEF, didn't have the ground truth values that were needed.
- While doing the training of fisher LDA, we need the ground truth values of relevancy between queries and documents.
- This is not available in any of the datasets that we could find. The one used in the original implementation of the paper was not accessible to us.
- Hence we showed the working of the fisher LDA algorithm on the MNIST dataset after converting into a binary classification problem (described below) to replicate a similar problem to the one in our implementation.
- The same algorithm could be extended to our actual problem statement given the ground truth values.

SECTION 8

# RESULTS

# BAG OF WORDS - TEXTUAL

The vocabulary for this part is generated by going through all the textual documents available to us. It is a list of all the unique words found in the documents. For the dataset used, it results in 4155 different words. A link to the vocabulary generated can be found here.

# TERM FREQUENCY MATRIX

The code returns a tf_matrix through which we can access the frequency component of any word of any document, This eases the process of access for calculating the scores in the next stage. It is a DxV matrix, where D is the number of textual documents available and V is the length of the vocabulary. This representation was used so that the tf vector of any document can be accessed only using the document index as the input.

Eg - To get the vector of **4323.eng**, we can do **tf_matrix[4323]**.

While calculating the matrix, the value of the constants were taken as -

$$k1 = 1.5$$
$$b = 0.75$$

The link to the tf_matrix can be found here.

# INVERSE DOCUMENT FREQUENCY MATRIX

A single matrix of the inverse document frequency is returned. This is a matrix of the same size as the vocabulary.

# SCORING BETWEEN TWO DOCUMENTS

As mentioned earlier, each pair of documents have a score vector associated with it. This is the size of the number of modalities being considered. One component of it is the textual score between both the documents. This is calculated by the formula given below -

$$score_T(q_k, d_i) = \sum_{t_j \in q_k} tf_{k,j} idf_j \; tf_{i,j} idf_j$$

# BAG OF WORDS - SIFT

All the images in the dataset were used to find the k image words. Each image was divided into units of size 60x60. The key points and their SIFT descriptors were calculated for each of these cells. We pick the strongest key point and output that for each unit. If there are no key points, we output the center-most pixel and its descriptors. Using all these descriptors from all the input images, we then perform k-means clustering to get the SIFT image words. k was set to 10. The feature vectors for each of them was found using the above method of tf and idf. The link to the image words can be found here.

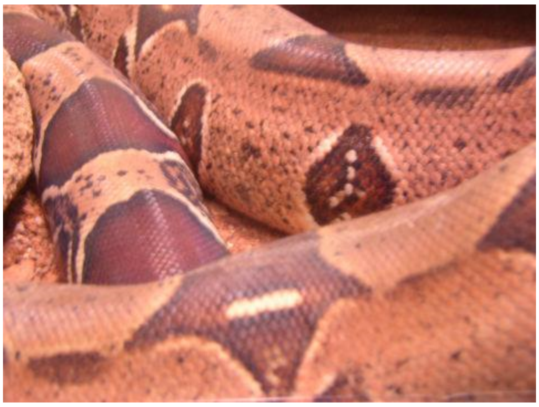
# BAG OF WORDS - MSTD

# SIFT SCORES

**Score: 1.63**

**Score: 2.57**

# SIFT SCORES (DISSIMAILAR IMAGES)



Score: 0.90

Score: 0.52

SECTION 9

# PROBLEM STATEMENT

*Given the scores for different modalities, we want to combine them using fusion parameters.*

**Final score = α1 (Text score) + α2 (SIFT score) + α3 (MSTD score)**

*We want to learn the α's such that the final score gives us the most relevant documents.*

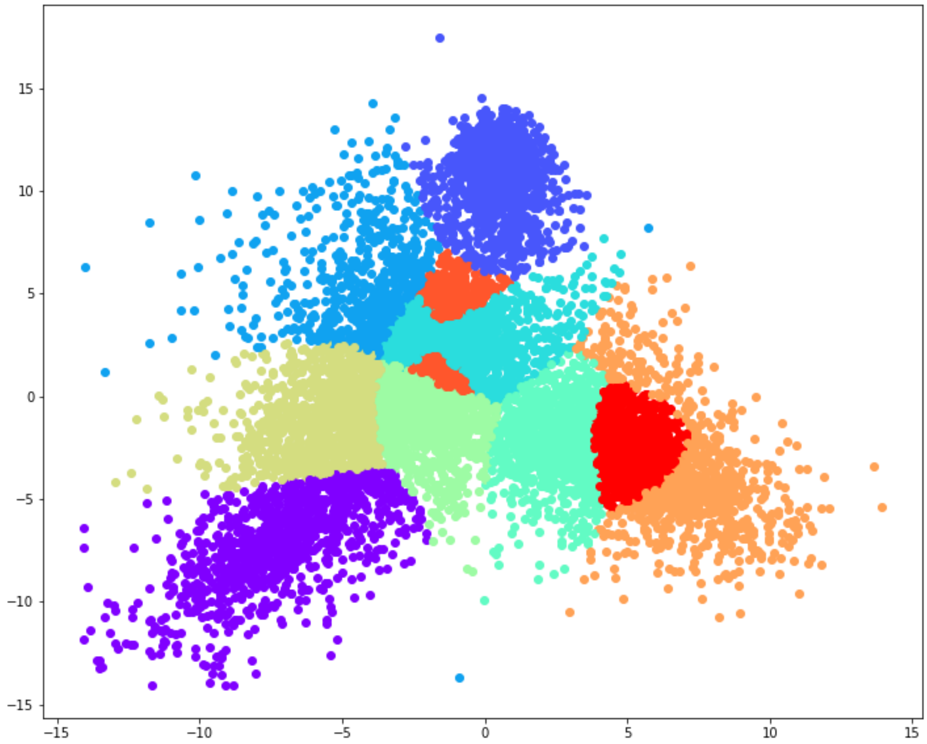
SECTION 10

# FISHER LDA

# DATASET USED - MNIST

To look at the working of the Fisher LDA algorithm, we used the MNIST dataset. This is a handwritten number dataset with 10 classes.

# REDUCTION FROM 784 TO 2 DIMENSIONS



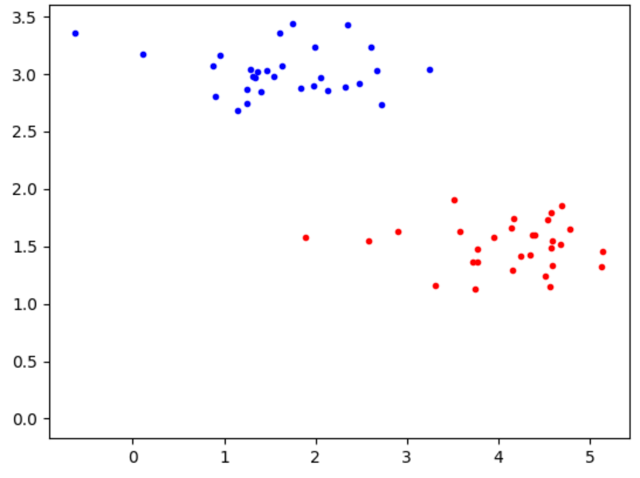
Accuracy: 56.61%

# REDUCTION FROM 784 TO 3 DIMENSIONS

Accuracy: 74.2%

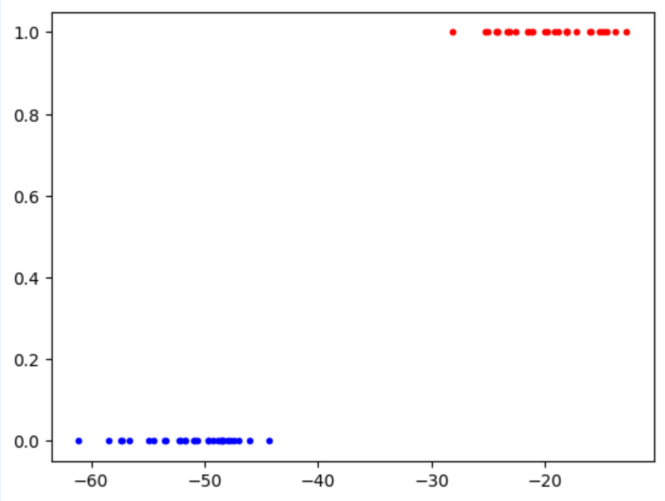
# ALGORITHM ANALYSIS FOR A 2 CLASS INPUT

We generated a 2 class input randomly to test the algorithm further. This was done because given the ground truth values of relevancy between query-document pairs, the problem would've been a 2 class problem, similar to this.



## INPUT VISUALISED. RED-CLASS 1. BLUE - CLASS 0.

## OUTPUT AFTER REDUCING FROM 2 TO 1 DIMENSIONS AND PROJECTING. LARGE SEPARATION.



SECTION 11

# WORK DIVISION

## SREENYA CHITLURI

- Worked on Fisher LDA
- Worked on generating image vocabulary and scoring
- Worked on report

## TEJAH S S

- Worked on Fisher LDA
- Worked on generating text vocabulary and scoring
- Worked on report

## SANKEERTHANA VENUGOPAL

- Worked on generating visual vocabularies - sift, mstd
- Worked on report

## M KRISHNA PRANEET

- Worked on generating textual vocabularies
- Worked on scoring
- Worked on report