Text Document Pre-processing and Classification Using Self-Organising Maps

Deepanshu
ED19D402

Department of Engineering Design
Indian Institute of Technology Madras, Chennai

ADOPT Advanced Design, Optimization and Probabilistic Techniques laboratory

Outline

- Motivation
- Introduction
- Text Documents Pre-processing
- Algorithms for Text Classification
- Classification of Vectorized Document using SOM
- Implementation of som-supervised function
- Examples
- Results
- Summary
- Way Forward

ADOPT Advanced Design, Optimization and Probabilistic Techniques laboratory

Motivation 1/3

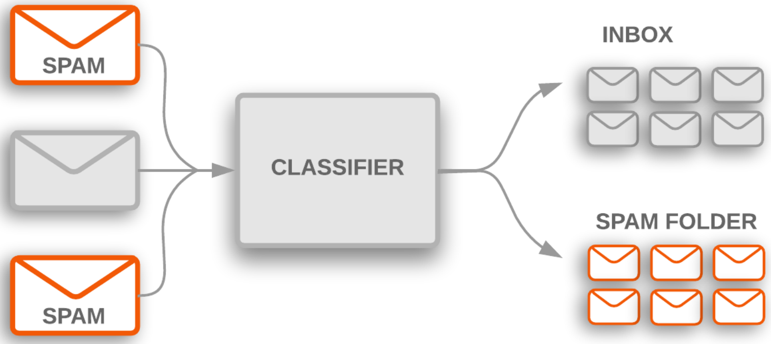

Figure 1: Spam Detection

ADOPT | Advanced Design, Optimization and Probabilistic Techniques laboratory

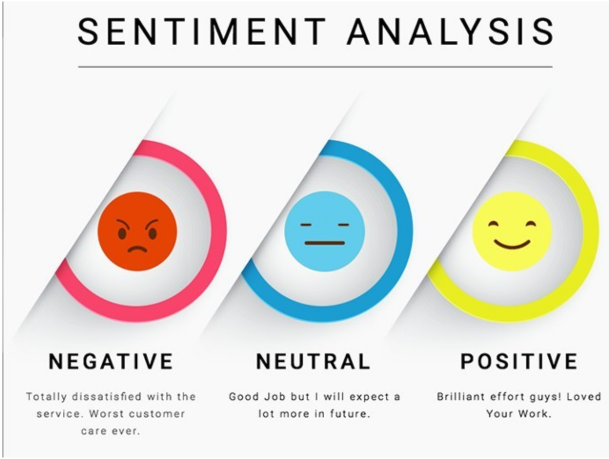

Figure 2: Sentiment Classification

ADOPT Advanced Design, Optimization and Probabilistic Techniques laboratory

Figure 3: Authorship Identification

SOM Working

Introduction

Input:

- A set of documents $[D = d_1, d_2, ..., d_n]$
- A fixed set of class $[C = c_1, c_2, ..., c_k]$
- A training set of **m** hand-labeled documents $[(d_1, c_1), ..., (d_m, c_m)]$

ADOPT | Advanced Design, Optimization and Probabilistic Techniques laboratory

Introduction

Input:

- A set of documents $[D = d_1, d_2, ..., d_n]$
- A fixed set of class $[C = c_1, c_2, ..., c_k]$
- A training set of **m** hand-labeled documents $[(d_1, c_1), ..., (d_m, c_m)]$

Output:

- A predicted class c ϵ C

Text Documents Pre-processing

Data Pre-processing

- Cleaning the text data
- Vectorizing the text data
- TF-IDF of the text data
- Characteristics of vectorized data
- Scope of the vectorized data

Cleaning the text data 1/3

- Tokenizing
- Remove Punctuation
- Remove Stop-words
- Lemmatization
- Stemming

ADOPT | Advanced Design, Optimization and Probabilistic Techniques laboratory

Cleaning the text data 1/3

- Tokenizing
- Remove Punctuation
- Remove Stop-words
- Lemmatization
- Stemming

Example:

Text Data: So there is no way for me to plug it in here in the US unless I go by a converter.

Label: 0

Cleaning the text data 2/3

Tokenizing:

Tokenizing a string denotes splitting a string with respect to a delimiter.

['So', 'there', 'was', 'no', 'way', 'for', 'me', 'to', 'plug', 'it', 'in', 'here', 'in', 'the', 'US', 'unless', 'I', 'went', 'by', 'a', 'converter,'.']

ADOPT Advanced Design, Optimization and Probabilistic Techniques laboratory

Cleaning the text data 2/3

Tokenizing:

Tokenizing a string denotes splitting a string with respect to a delimiter.
['So', 'there', 'was', 'no', 'way', 'for', 'me', 'to', 'plug', 'it', 'in', 'here', 'in', 'the', 'US', 'unless', 'I', 'went', 'by', 'a', 'converter,'.']

Remove Punctuation:

['So', 'there', 'was', 'no', 'way', 'for', 'me', 'to', 'plug', 'it', 'in', 'here', 'in', 'the', 'US', 'unless', 'I', 'went', 'by', 'a', 'converter']

ADOPT | Advanced Design, Optimization and Probabilistic Techniques laboratory

Cleaning the text data 2/3

Tokenizing:

Tokenizing a string denotes splitting a string with respect to a delimiter.

['So', 'there', 'was', 'no', 'way', 'for', 'me', 'to', 'plug', 'it', 'in', 'here', 'in', 'the', 'US', 'unless', 'I', 'went', 'by', 'a', 'converter,'.']

Remove Punctuation:

['So', 'there', 'was', 'no', 'way', 'for', 'me', 'to', 'plug', 'it', 'in', 'here', 'in', 'the', 'US', 'unless', 'I', 'went', 'by', 'a', 'converter']

Remove Stop-words:

Removes words like 'if', 'he', 'she', 'the', etc which never belongs to any topic.

['So', 'way', 'plug', 'US', 'unless', 'I', 'went', 'converter']

ADOPT | Probabilistic Techniques laboratory

Cleaning the text data 3/3

Lemmatizer:

Lemmatizer is a transformers which transforms the word to its singular, present-tense form
['So', 'way', 'plug', 'US', 'unless', 'I', 'go', 'converter']

Cleaning the text data 3/3

Lemmatizer:

Lemmatizer is a transformers which transforms the word to its singular, present-tense form
['So', 'way', 'plug', 'US', 'unless', 'I', 'go', 'converter']

Stemming:

Stemming and Lemmatization both generate the root form of the inflected words. The difference is that stem might not be an actual word whereas, lemma is an actual language word.

Raw Data vs Clean Data

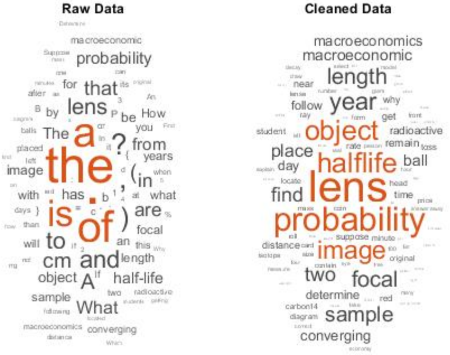

Figure 4: Bag of Words

ADOPT Advanced Design, Optimization and Probabilistic Techniques laboratory

Vectorizing the Text Documents 1/2

Vectorizing the Text Documents 1/2

Document Term Matrix (DTM)

- Each column represents a word
- Each row represents a document
- The value in each cell is typically a count of appearance of words but can be other values (e.g., does the word appear at all).

ADOPT Advanced Design, Optimization and Probabilistic Techniques laboratory

Vectorizing the Text Documents 1/2

Document Term Matrix (DTM)

- Each column represents a word
- Each row represents a document
- The value in each cell is typically a count of appearance of words but can be other values (e.g., does the word appear at all).

CountVectorizer

Creates a document term matrix in which the value in each cell represents the count of the appearance of that word in the document each row represents

ADOPT Advanced Design, Optimization and Probabilistic Techniques laboratory

Vectorizing the Text Documents 2/2

ADOPT | Advanced Design, Optimization and Probabilistic Techniques laboratory

Vectorizing the Text Documents 2/2

Doc1: The movie was good and I liked it's music, good movie. : C_1
Doc2: Movie has bad script and slow music. : C_2

Vectorizing the Text Documents 2/2

Doc1: The movie was good and I liked it's music, good movie. : C_1
Doc2: Movie has bad script and slow music. : C_2
Bag of Words: ['movie';'good';'like';'bad';'script';'slow';'music']

ADOPT | Advanced Design, Optimization and Probabilistic Techniques laboratory

Vectorizing the Text Documents 2/2

Doc1: The movie was good and I liked it's music, good movie. : C_1
Doc2: Movie has bad script and slow music. : C_2
Bag of Words: ['movie';'good';'like';'bad';'script';'slow';'music']

CountVectorizer							
Doc no	movie	good	like	bad	script	slow	music
Doc1	2	2	1	0	0	0	1
Doc2	1	0	0	1	1	1	1

Table 1: CountVectorizer

ADOPT | Advanced Design, Optimization and Probabilistic Techniques laboratory

Vectorizing the Text Documents 2/2

Doc1: The movie was good and I liked it's music, good movie. : C_1
Doc2: Movie has bad script and slow music. : C_2
Bag of Words: ['movie';'good';'like';'bad';'script';'slow';'music']

CountVectorizer							
Doc no	movie	good	like	bad	script	slow	music
Doc1	2	2	1	0	0	0	1
Doc2	1	0	0	1	1	1	1

Table 1: CountVectorizer

Drawback

- It only gives the frequency of a word appeared across the documents
- It is more important to know the frequency of the word in different class

TF-IDF 1/2

Term Frequency-Inverse Document Frequency (TF-IDF)

- The term frequency refers to how much a term (i.e. a word) appears in a document

$$TF(t) = \frac{\Sigma w_t}{\Sigma W}$$

- Inverse document frequency refers to how common or rare a term appears in a document

$$IDF(t) = log_e \frac{\Sigma D}{\Sigma D_{w_t}}$$

- **TF-IDF(t) = TF(t)×IDF(t)**

ADOPT | Advanced Design, Optimization and Probabilistic Techniques laboratory

TF-IDF 2/2

Significance of TF-IDF Value

In general higher the TF-IDF value more is it's significance/relevance in document

TF-IDF 2/2

Significance of TF-IDF Value

In general higher the TF-IDF value more is it's significance/relevance in document

Example

- Consider a document containing 100 words wherein the word cat appears 3 times. Now, assume we have 10 million documents and the word cat appears in one thousand of these.
- $\text{TF(cat)} = (3/100) = 0.03$
- $\text{IDF(cat)} = \log(10{,}000{,}000/1{,}000) = 4$
- Thus, $\text{TF-IDF(cat)} : 0.03 \times 4 = 0.12$

ADOPT | Advanced Design, Optimization and Probabilistic Techniques laboratory

Characteristics and Scope of Vectorized data

- **Characteristics**: It is generally large size sparse matrix
- **Scope**: This matix can be used as input in different supervised and un-supervised learning algorithms
- It's numerical analogous of text data, so it can be used to compare documents.(Cosine Similarity)

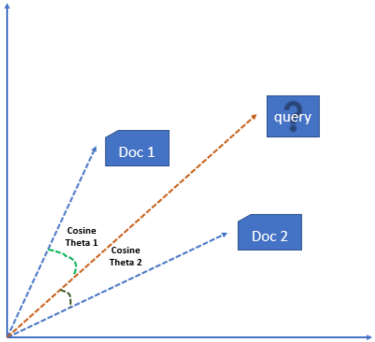

Figure 5: Cosine Similarity

ADOPT | Advanced Design, Optimization and Probabilistic Techniques laboratory

Algorithms for Text Classification

- Naïve-Bayes
- Logistic Regression
- Support vector machines
- k Nearest Neighbors
- Self-Organising Maps
- Recurrent Neural Networks (RNN)

ADOPT | Advanced Design, Optimization and Probabilistic Techniques laboratory

Algorithms for Text Classification

- Naïve-Bayes
- Logistic Regression
- Support vector machines
- k Nearest Neighbors
- Self-Organising Maps
- Recurrent Neural Networks (RNN)

Algorithm

$argmin_G$ **E[L(g(X),Y)]**

$g \epsilon G$, where G is function class

L is the learner

ADOPT | Advanced Design, Optimization and Probabilistic Techniques laboratory

Algorithms for Text Classification

- Naïve-Bayes
- Logistic Regression
- Support vector machines
- k Nearest Neighbors
- Self-Organising Maps
- Recurrent Neural Networks (RNN)

Algorithm

$argmin_G$ **E[L(g(X),Y)]**

$g \epsilon G$, where G is function class

L is the learner

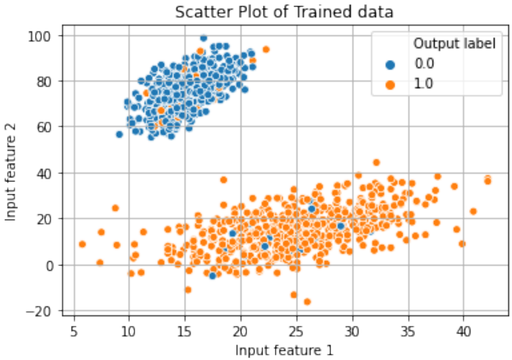

Figure 6: 2-D Dataset

Algorithms for Text Classification

- Naïve-Bayes
- Logistic Regression
- Support vector machines
- k Nearest Neighbors
- Self-Organising Maps
- Recurrent Neural Networks (RNN)

Algorithm

argmin$_G$ **E[L(g(X),Y)]**

$g \epsilon G$, where G is function class

L is the learner

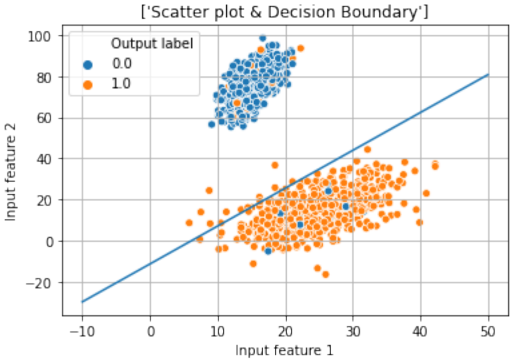

Figure 6: Linear Classifier

Algorithms for Text Classification

- Naïve-Bayes
- Logistic Regression
- Support vector machines
- k Nearest Neighbors
- Self-Organising Maps
- Recurrent Neural Networks (RNN)

Algorithm

$argmin_G$ **E[L(g(X),Y)]**

$g \epsilon G$, where G is function class

L is the learner

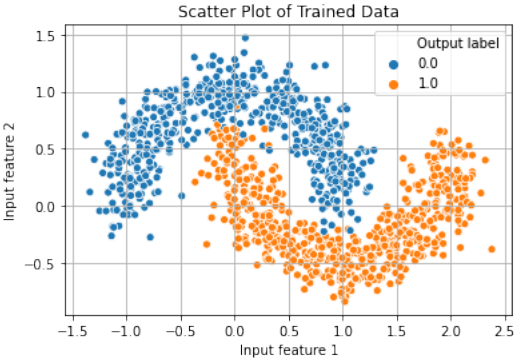

Figure 6: 2-D Dataset

ADOPT Advanced Design, Optimization and Probabilistic Techniques laboratory

Algorithms for Text Classification

- Naïve-Bayes
- Logistic Regression
- Support vector machines
- k Nearest Neighbors
- Self-Organising Maps
- Recurrent Neural Networks (RNN)

Algorithm

$argmin_G$ **E[L(g(X),Y)]**

$g \epsilon G$, where G is function class

L is the learner

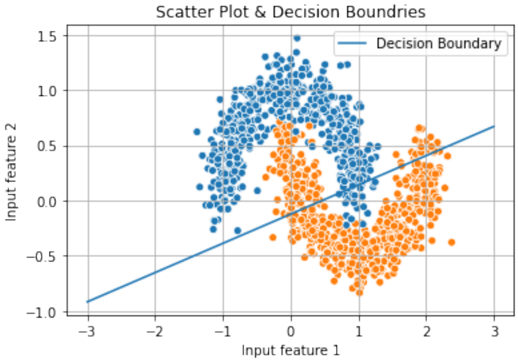

Figure 6: Linear Classifier

ADOPT Advanced Design, Optimization and Probabilistic Techniques laboratory

- Naïve-Bayes
- Logistic Regression
- Support vector machines
- k Nearest Neighbors
- Self-Organising Maps
- Recurrent Neural Networks (RNN)

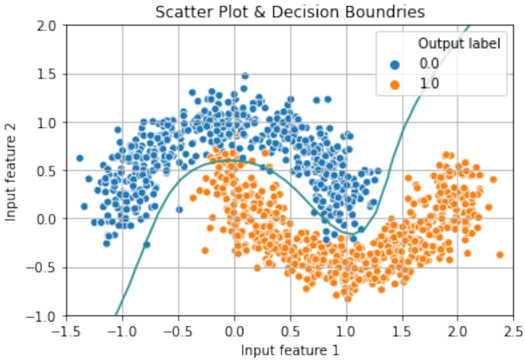

Figure 6: Non-linear Classifier

Algorithm

$argmin_G$ **E[L(g(X),Y)]**

$g \epsilon G$, where G is function class

L is the learner

Algorithms for Text Classification

- Naïve-Bayes
- Logistic Regression
- Support vector machines
- k Nearest Neighbors
- Self-Organising Maps
- Recurrent Neural Networks (RNN)

Algorithm

$argmin_G$ **E[L(g(X),Y)]**
$g \epsilon G$, where G is function class
L is the learner

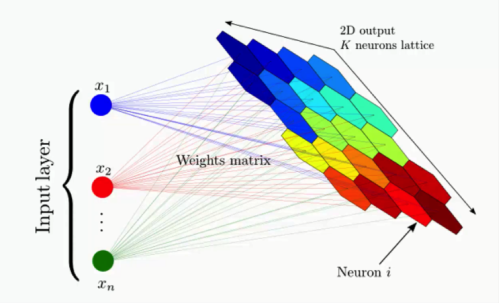

Figure 6: SOM

$$N_i = \Sigma_{i=1}^{n} w_i * x_i$$

ADOPT Advanced Design, Optimization and Probabilistic Techniques laboratory

Classification of Vectorized Document using SOM

Figure 7: Bag of Words

Classification of Vectorized Document using SOM

Figure 7: Bag of Words

Obs	_X1	_X2	_X3	_X4	_X5	_X6	_Z1	_Z2	_Z3	_Z4
1	1	0	1	0	1	0	0	0	1	0
2	1	1	0	1	0	0	0	0	0	1
3	1	0	1	1	0	0	1	0	0	0
4	1	1	0	0	0	1	0	1	0	0
5	1	1	0	0	1	0	0	0	1	0
6	1	1	0	1	0	0	0	0	0	1
7	1	1	0	0	1	0	1	0	0	0
8	1	1	0	1	0	0	0	1	0	0
9	1	1	0	0	1	0	0	0	1	0
10	1	0	1	0	1	0	0	0	0	1

Figure 8: Document Term Matrix

Classification of Vectorized Document using SOM

Figure 7: Bag of Words

Obs	_X1	_X2	_X3	_X4	_X5	_X6	_Z1	_Z2	_Z3	_Z4
1	1	0	1	0	1	0	0	0	1	0
2	1	1	0	1	0	0	0	0	0	1
3	1	0	1	1	0	0	1	0	0	0
4	1	1	0	0	0	1	0	1	0	0
5	1	1	0	0	1	0	0	0	1	0
6	1	1	0	1	0	0	0	0	0	1
7	1	1	0	0	1	0	1	0	0	0
8	1	1	0	1	0	0	0	1	0	0
9	1	1	0	0	1	0	0	0	1	0
10	1	0	1	0	1	0	0	0	0	1

Figure 8: Document Term Matrix

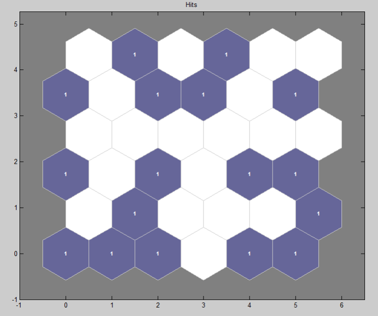

Figure 9: SOM implementation

ADOPT Advanced Design, Optimization and Probabilistic Techniques laboratory

Implementation of som-supervised function

- **STEP I**: Text Pre-processing
- **STEP II**: Train-Test Split
- **STEP III**: Document Term Matrix
- **STEP IV**: Implement SOM Supervised on labelled trained data
- **STEP V**: Get Struct Map of trained data
- **STEP VI**: Getting labels of test data by mapping it on trained map
- **STEPVII**: Cross validate it with actual labels

Binary classification 1/3

Figure 10: Raw data vs Cleaned data

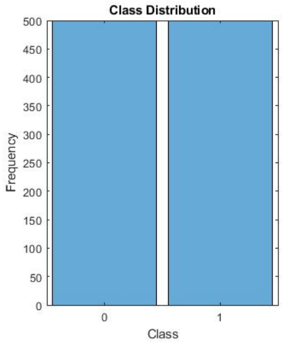

Figure 11: Class Distribution

ADOPT | Advanced Design, Optimization and Probabilistic Techniques laboratory

Binary Classification 2/3

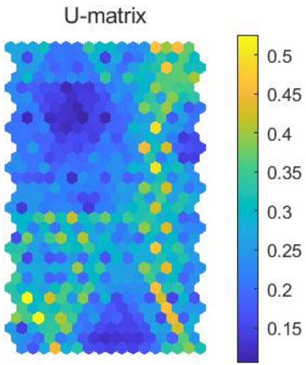

Figure 12: U-Matrix

Binary Classification 2/3

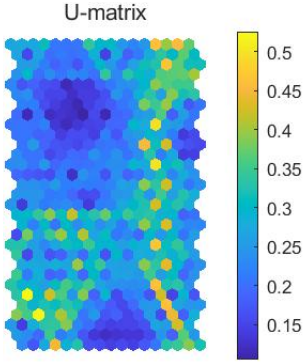

Figure 12: U-Matrix

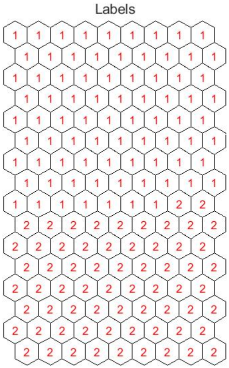

Figure 13: Labels on training dataset

Binary Classification 2/3

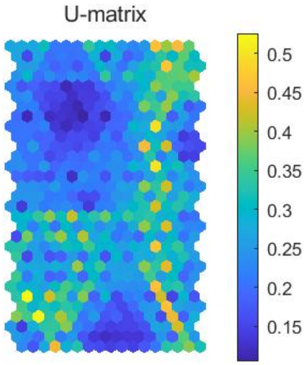

Figure 12: U-Matrix

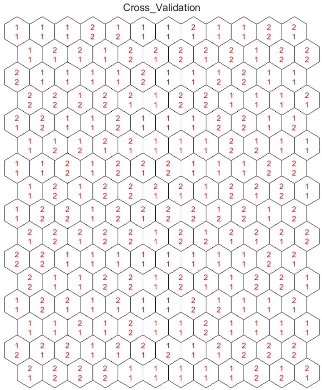

Figure 13: Cross-Val on test dataset

ADOPT Advanced Design, Optimization and Probabilistic Techniques laboratory

Binary Classification 3/3

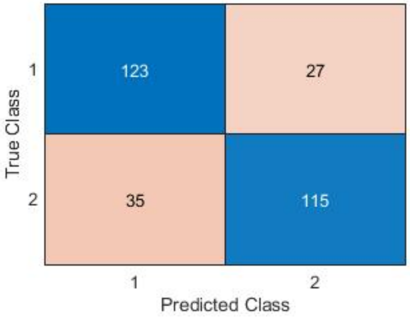

Figure 14: Confusion matrix: SVC (0.7933)

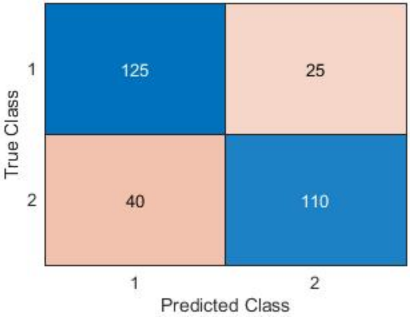

Figure 15: Confusion matrix: SOM (0.7833)

ADOPT | Advanced Design, Optimization and Probabilistic Techniques laboratory

Results of Binary Classification

Accuracy on Test data set					
Dataset	Size	Naïve-Bayes	SVM	Log-Reg	SOM
Amazon	1000	0.7933	0.7933	0.8200	0.7833
Yelp	1000	0.7666	0.7600	0.7466	0.7350
IMDB	1000	0.8200	0.8333	0.8000	0.7104
SMS Spam	10000	0.9546	0.9701	0.9558	0.9756

Table 2: Results of Binary Classification

ADOPT Advanced Design, Optimization and Probabilistic Techniques laboratory

Multi-class Classification 1/3

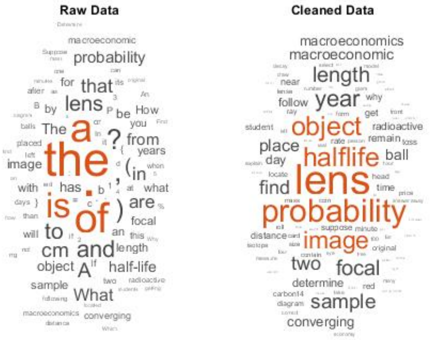

Figure 16: Raw data vs Cleaned data

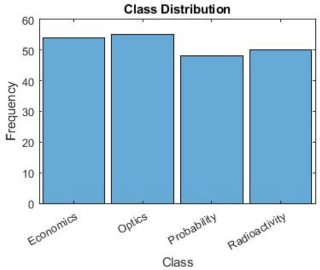

Figure 17: Class Distribution

ADOPT | Advanced Design, Optimization and Probabilistic Techniques laboratory

Multi-class Classification 2/3

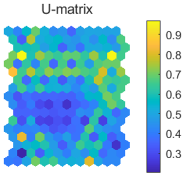

Figure 18: U-Matrix

Multi-class Classification 2/3

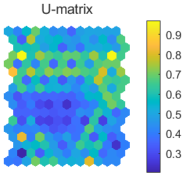

Figure 18: U-Matrix

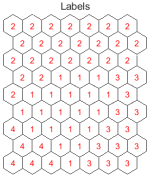

Figure 19: Labels

ADOPT Advanced Design, Optimization and Probabilistic Techniques laboratory

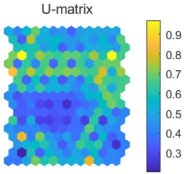

Figure 18: U-Matrix

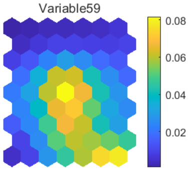

Figure 20: macroeconomics

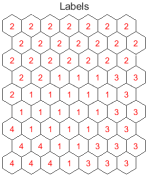

Figure 19: Labels

ADOPT Advanced Design, Optimization and Probabilistic Techniques laboratory

Multi-class Classification 2/3

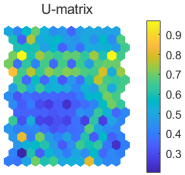

Figure 18: U-Matrix

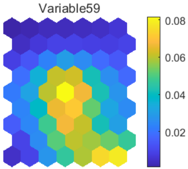

Figure 20: macroeconomics

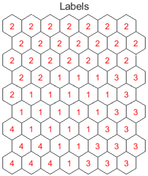

Figure 19: Labels

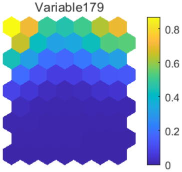

Figure 21: lenses

ADOPT Advanced Design, Optimization and Probabilistic Techniques laboratory

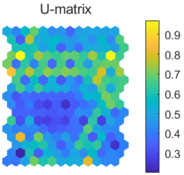

Figure 18: U-Matrix

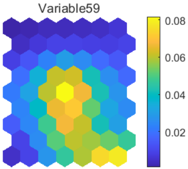

Figure 20: macroeconomics

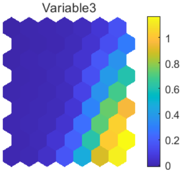

Figure 22: probability

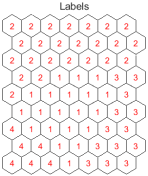

Figure 19: Labels

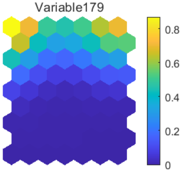

Figure 21: lenses

Multi-class Classification 2/3

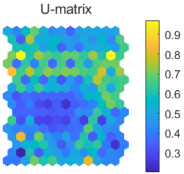

Figure 18: U-Matrix

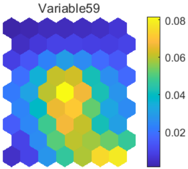

Figure 20: macroeconomics

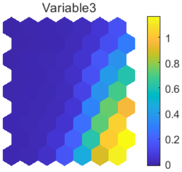

Figure 22: probability

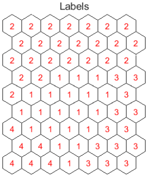

Figure 19: Labels

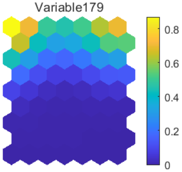

Figure 21: lenses

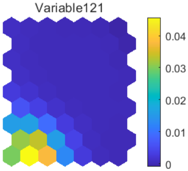

Figure 23: half life

ADOPT Advanced Design, Optimization and Probabilistic Techniques laboratory

Multi-class Classification 3/3

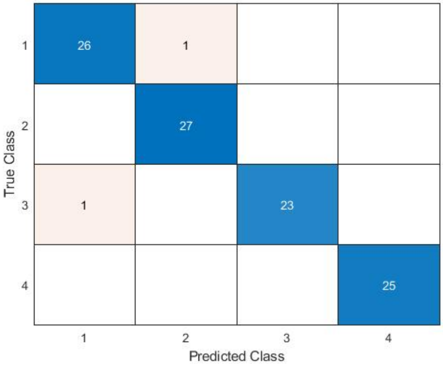

Figure 24: Confusion matrix using SOM(accuracy = 0.9806)

Multi-class Classification 3/3

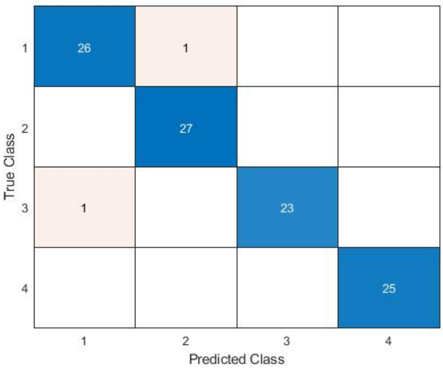

Figure 24: Confusion matrix using SOM(accuracy = 0.9806)

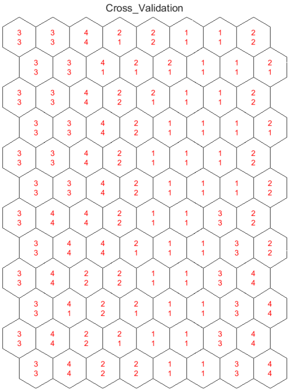

Figure 25: Cross Validation (accuracy = 0.9806)

Multi-class Classification 3/3

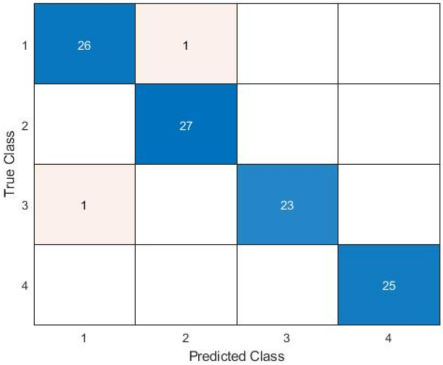

Figure 24: Confusion matrix using SOM(accuracy = 0.9806)

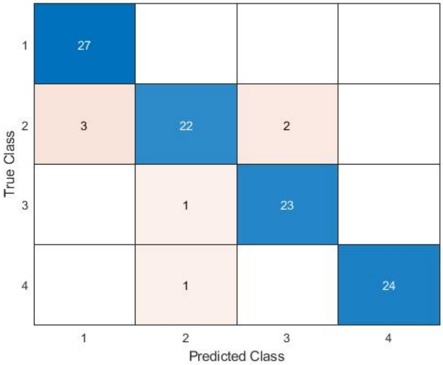

Figure 25: Confusion matrix using SVC(accuracy = 0.9320)

ADOPT | Advanced Design, Optimization and Probabilistic Techniques laboratory

Summary

- There are lots of classification problem which are in the form of text documents
- Text data pre-processing is done to extract relevant keywords
- Appearance or absence of these relevant keywords in particular document makes the classification task effective
- Document term matrix is numerical data analogous of text data
- Document term matrix can be used as input for the different classification algorithms
- SOM supervised function of SOM toolbox has been used to predict the given class/label of text data-sets
- The accuracy of SOM supervised is calculated and compared with that of achieved from other classification algorithms

Way Forward

- Using SOM for unsupervised learning
- Using hybrid classification method (Naïve-Bayes+SOM) to increase the accuracy
- Implementation of High Relevance Key Extraction (HRKE) for text preprocessing
- Implementation of tournament ranking methods for multi-class classification problems
- Improve the visualization of clusters through U-matrix

Thank You

ADOPT | Advanced Design, Optimization and Probabilistic Techniques laboratory