# Statistical Approaches for Binary and Categorical Data Modeling

Fahdah Abdullah Alalyan

A thesis

in

The Department

of

Concordia Institute for Information Systems Engineering

Presented in Partial Fulfillment of the Requirements
For the Degree of MASTER OF APPLIED SCIENCE
(INFORMATION SYSTEMS SECURITY)
Concordia University
Montréal, Québec, Canada

October 2020

# Concordia University
School of Graduate Studies

This is to certify that the thesis prepared

By: **Fahdah Abdullah Alalyan**

Entitled: **Statistical Approaches for Binary and Categorical Data Modeling**

and submitted in partial fulfillment of the requirements for the degree of

**Master of Applied Science**
**(Information Systems Security)**

complies with the regulations of this University and meets the accepted standards with respect to originality and quality.

Signed by the final examining commitee:

Dr. Ayda Basyouni _____ Chair
Dr. Nizar Bouguila _____ Supervisor
Dr. Jamal Bentahar _____ CIISE Examiner
Dr. Joonhee Lee _____ External Examiner

Approved _____
Dr. Mohammad Mannan, Graduate Program Director

_____ 20 _____ _____
Dr. Mourad Debbabi, Dean
Faculty of Engineering and Computer Science

# Abstract

## Statistical Approaches for Binary and Categorical Data Modeling

Fahdah Abdullah Alalyan

Nowadays a massive amount of data is generated as the development of technology and services has accelerated. Therefore, the demand for data clustering in order to gain knowledge has increased in many sectors such as medical sciences, risk assessment and product sales. Moreover, binary data has been widely used in various applications including market basket data and text documents analysis. While applying classic widely used k-means method is inappropriate to cluster binary data, we propose an improvement of K-medoids algorithm using binary similarity measures instead of Euclidean distance which is generally deployed in clustering algorithms. In addition to K-medoids clustering method, agglomerative hierarchical clustering methods based on Gaussian probability models have recently shown to be efficient in different applications. However, the emerging of pattern recognition applications where the features are binary or integer-valued demand extending research efforts to such data types. We propose a hierarchical clustering framework for clustering categorical data based on Multinomial and Bernoulli mixture models. We have compared two widely used density-based distances, namely; Bhattacharyya and Kullback-Leibler. The merits of our proposed clustering frameworks have been shown through extensive experiments on clustering text, binary images categorization and images categorization.

The development of generative/discriminative approaches for classifying different kinds of data has attracted scholars' attention. Considering the strengths and weaknesses of both approaches, several hybrid learning approaches which combined the desirable properties of both have been developed. Our contribution is to combine Support Vector Machines (SVMs) and Bernoulli mixture model in order to classify binary data. We propose using Bernoulli mixture model for generating probabilistic kernels for SVM based on information divergence. These kernels make intelligent

use of unlabeled binary data to achieve good data discrimination. We evaluate the proposed hybrid learning approach by classifying binary and texture images.

# Acknowledgments

First of all, I am grateful to God Almighty for his graces and sustenance for completing my graduate studies abroad. His benevolence has made me aspire and strive for success until I reached the stage of writing a master's thesis with complete satisfaction. Also, I would like to thank Minstry of Education in Saudi Arabia for awarding me a scholarship. I hope that I return back valuable knowledge and experience to my country. I would like to express my deepest appreciation for my supervisor Prof. Nizar Bouguila for his continuous support, help, and motivation. I was dreaming for publishing a conference paper in the beginning, but he encouraged and gave me a chance to publish more which leaves a huge impact in my entire life to starve for more achievements and believe in myself. My special thanks are extended to my parents Abdullah Alalyan and Fatmah Almulaifi and my mother in law Haya Alwarthan as well as my siblings Hanan, Abdulaziz, Nouf, and Fawaz for their continues support and love. I would like to express my deepest love and gratitude to my husband Abdullah Alolayan and my daughters Dona and Rola for their constant support and encouragement. My grateful thanks are also extended to my best friend Mona Albader for her motivation words, help and valuable gaudiness throughout my study. Finally, I would also like to thank all my lab research colleagues for sharing their vast knowledge and explain various concepts.

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

## 1.1 Background

A huge amount of data is generated every day due to today's technologies such as the Internet and professional cameras. Consequentially, automatic techniques have been required to be innovated in order to analyze data efficiently and extract hidden knowledge [15]. Learning techniques can be grouped into two families: supervised and unsupervised learning. While training data are required for supervised approaches, unsupervised models are needed when data are unlabeled. This makes unsupervised problems more challenging and difficult than supervised ones [50, 79]. Clustering is among the significant unsupervised tasks that have been discussed and caught scientists' attention [38]. The main goal of data clustering is to predict and find the groups/classes for each data object from unlabeled data. On the other hand, selecting appropriate representations for data is one of the central problems in machine learning and data mining [72]. Binary data, for instance, play an essential role in multiple fields such as computer vision, image processing, machine learning and data mining [16]. Binary data has been widely used in data analysis applications such as market basket data modeling and document clustering [59, 50, 22].

Although K-means algorithm has been developed over 50 years ago, it is still the most popular clustering algorithm because it is easy to implement, simple, and efficient [50]. Simplicity and speed are the major advantages of applying K-means in order to partition the number of objects to known classes. However, there are some limitations since classic K-means is not appropriate for all kinds of data because of

its dependency on using Euclidean distance [10]. This shortcoming is also valid for binary data which is the simplest form of categorical data [69]. In order to overcome the limitations of K-means, K-medoids has been proposed in the past as a potential solution [79]. In our work we propose a K-medoids algorithm for binary data clustering based on binary sequences similarity measures instead of Euclidean distance.

Moreover, in many domains, classes that are represented by different modes in the mixture are indeed generalizations of each other [75]. Examples include biological taxonomy, phylogenetic, internet newsgroups and emails where clusters may have one or more sub-clusters [36, 45]. In this case, the structure of clustering is hierarchical since each data point and cluster can be represented as leave and internal node of dendrogram (tree), respectively [45, 94]. Agglomerative algorithms have been primarily used for hierarchical structure solutions [43, 41, 55, 57, 81, 92]. Generally, Euclidean distance is used to measure the distance between any two clusters [45]. However, there are some limitations to the traditional hierarchical clustering. For instance, the difficulty of finding a theoretical justification for which distance measure to choose specifically for structured data such as images. Our proposed work has overcame the limitations of the traditional hierarchical clustering. We propose an agglomerative algorithm based on Multinomial and Bernoulli mixture models.

Learning approaches are generally divided into two groups, namely, generative (ex. mixture models) and discriminative approaches such as Support Vector Machines (SVMs) [12]. The advantages of using the two approaches are different, as well as the limitations vary from one approach to another. The idea of hybrid approaches is to combine the desirable capabilities of both approaches such that we can capture the substantial properties of the data to classify, taking into consideration prior knowledge of the problem domain [71]. A generative/discriminative approach for binary data has been proposed in [74], where the Bernoulli mixture model has been considered for generating probabilistic kernels based on the Kullback–Leibler divergence. The aim of our work is to propose and compare different probabilistic kernels based on the Bernoulli mixture model. We propose two generative kernels based on information divergence in probability density function (PDF) space, namely, Bhattacharyya and Rényi divergence and compare them with Kullback–Leibler divergence.

## 1.2 Contributions

The main objective of this thesis is to study the efficiency of categorical data clustering using hierarchical clustering and K-medoids for clustering binary data. In addition to clustering, binary vectors classification using a hybrid learning approach is included in our study. The contributions are listed as follows:

- **Binary Data Clustering Using K-medoids Clustering Method Based on Binary Sequences Similarity Measures**

  We propose a K-medoids algorithm for binary data clustering based on binary sequences similarity measures instead of Euclidean distance. We have considered two challenging applications, namely; text clustering and binary images categorization to show the merits of the proposed framework. This work has been published in IEEE International Conference on Control, Decision and Information Technologies, 2019 [2]

- **Categorical Data Clustering Using Hierarchical Clustering Approach Based on Multinomial and Bernoulli Mixture Models**

  An agglomerative algorithm based on Multinomial and Bernoulli mixture models is proposed. The distances between the clusters (components) are measured using Bhattacharyya and Kullback-Leibler distances. Efficiency of proposed model is tested for results on two real-world applications namely text clustering and images categorization using the bag of visual words model. This contribution has been published in IEEE International Symposium on Industrial Electronics, 2019 [3].

- **Binary Vectors Classification Using Hybrid Learning Approach Based on SVM and Bernoulli Mixture Model**

  We introduce and compare different probabilistic kernels based on the Bernoulli mixture model. We address the problem of classifying data that consist of bags of binary vectors by incorporating the Bhattacharyya and Rényi kernels based on the Bernoulli mixture model into SVMs. We validate the proposed hybrid model in classifying binary and texture images. This work has been accepted by IEEE International Conference on Systems, Man, and Cybernetics, 2020 [4].

## 1.3 Thesis Overview

- Chapter 1 introduces the concepts of data clustering and hybrid models for classification and a brief overview of several concepts related to our contributions. Moreover, we explain the motivation behind our research.

- Chapter 2 , we explain in detail K-medoids method to cluster binary data using different binary sequences similarity measures. The efficiency of our extension to the K-medoids algorithm is validated by two applications namely: text clustering and binary images categorization.

- Chapter 3, we discuss in details a hierarchical clustering framework in case of Bernoulli and Multinomial mixture models. The experiments with two applications, including text document clustering and images categorization are described and discussed.

- Chapter 4 describes and compares different probabilistic kernels based on the Bernoulli mixture model into SVMs. The model has been tested with the problem of classifying binary and texture images.

- Chapter 5 concludes and summarizes our contributions.

# Chapter 2

# An Improved K-medoids Algorithm Based on Binary Sequences Similarity Measures

In this chapter, we detail our proposed K-medoids algorithm for binary data clustering. We handle K-means' limitation, which is inappropriate to cluster binary data, with a K-medoids algorithm. We evaluate our model against two real-world applications like text document clustering and binary images categorization.

## 2.1 K-means

K-means algorithm is a clustering approach which has been widely used in various applications, such as image segmentation and information retrieval [44, 50, 87]. K-means is based on hard assignment which means each data object is assigned to one class [50]. Moreover, K-means has been used as an initialization step for other algorithms such as Expectation-Maximization (EM) [78], due to its simplicity and efficiency [69]. The classic K-means approach is based on Euclidean distance which is used to calculate the minimum distance between each data object and cluster centroids [44]. Each data object is assigned to the nearest mean which represents the class. Simplicity and speed are the major advantages of applying K-means in order to partition the number of objects to known classes. However, each run may provide different results depending on the initialization step [79]. In addition, the dependency

on using Euclidean distance is not appropriate for all kinds of data. For example, using K-means for mixed or categorical datasets would be inappropriate [10]. This limitation is also valid for binary data [69]. In addition, the sensitivity to outliers is another limit [85]. In order to overcome these problems, a potential solution, as K-medoids, has been proposed in the past [79]. K-medoids method is based on choosing the most central data object located in a given class as a medoid acting as representative of the class. Thus, the main difference between the K-means and the K-medoid algorithms is that the first is based on the mean as a cluster representative while the second is based on the median [86, 1].

## 2.2 The Extensions for K-means and K-medoid algorithms

Some extensions and modifications have been made in the past to improve the efficiency of K-means and K-medoids algorithms. The extensions for K-means are made in many different ways. For instance, Fuzzy c-means was proposed to handle soft assignment such that each data object can be assigned to more than one class with different posterior probabilities [50]. The goals for the improvement were different as well as the purposes. For instance, taking into account the kind of the data was the main motivation in [78], while the speed was the main purpose for the modification shown in [69]. Indeed, in [78] the authors have compared different distances integrated within the K-means algorithm to cluster proportional data. Their experiments showed that Aitchison distance gives the best performance among other distances [78]. In order to accelerate K-means algorithm, the authors in [69] have used sparse distance.

In addition, partitioning around medoids (PAM) is one extension of K-medoids algorithms [70]. PAM is a powerful clustering method, but it is not efficient for large datasets because of its complexity. As a result, Another work proposed a new algorithm which calculates the distance once for each iteration. This improvement reduces the computing time which eliminates the drawback of PAM [70]. Even though Euclidean distance is not appropriate for categorical data [10], such as binary data, an improvement of the K-means algorithm has been used to cluster large binary datasets by assuming that applying Euclidean distance to binary data is acceptable. Scalable

K-means and Incremental K-means have been used to accelerate the algorithm for clustering binary data streams [69].

An extension of K-medoids algorithm is proposed in this work in order to improve its performance using binary sequences similarity measures instead of Euclidean distance. Indeed, many researchers have taken elaborate efforts to deploy binary similarity and dissimilarity (distance) measures in pattern analysis problems such as classification, and clustering in various fields. For instance, Jaccard similarity measure, as one of the similarity measures, has been used to cluster ecological species [49]. Other applications for binary similarity measures have been applied to different fields such as biology [48, 67], image retrieval [80], geology [47], and chemistry [29]. Recently, binary similarity measures have been actively used to solve the identification problems in biometrics such as iris images [28], and handwritten character recognition [27, 26].

## 2.3 Proposed Method

This section will give a description of an extension of K-medoids algorithm to handle binary data clustering via applying different similarity measures. Applying K-medoids directly instead of K-means is not enough for improving binary data clustering. Thus, our objective is to find alternative similarity measures that are applicable to binary data instead of Euclidean distance. Consequently, we have applied and tested 21 similarity measures out of 76 binary sequences similarity measures which are mentioned in [29].

Let $\mathcal{X} = \{X_1, \ldots, X_N\}$, to be a dataset with $N$ instances where each is a $D$-dimensional binary vector representing a document, or a binary image. K-medoids method has been used in the proposed model in order to assign binary values to medoids instead of assigning non-binary values to centroids using K-means approach. The main idea of K-medoids is assigning a real data object, which is a binary vector, to be a medoid $M_j$ for each class $j$, $j$= {1,. . . ,$K$} while $K$ represents the number of classes which is given as input.

The main goal of K-medoids clustering method is increasing the similarity between data objects within the same class and dissimilarity between the data objects from different classes [70]. This goal is achieved by calculating a similarity measure between

Table 1: Binary similarity measures

| Similarity measures | Expression |
|---|---|
| Dice | $\frac{2C}{N_1+N_2}$ |
| 2nd Kulcz | $\frac{C(N_1+N_2)}{2(N_1N_2)}$ |
| Otsuka | $\frac{C}{\sqrt{(N_1N_2)}}$ |
| Sorgenfrei | $\frac{C^2}{N_1N_2}$ |
| No. Feat. Diff | $E_1 + E_2$ |
| Sokal Dist (a) | $\sqrt{\frac{E_1+E_2}{N_1+N_2-C+C_0}}$ |
| Sokal Dist (b) | $\sqrt{1 - \frac{C+C_0}{N_1+N_2-C+C_0}}$ |

each data object $X_i$ in $\mathcal{X}$ and each medoid $M_j$, as:

$$\mathcal{D}_{ij} = \mathcal{V}(X_i, M_j) \tag{1}$$

where $\mathcal{V}(X_i,M_j)$ is a similarity measure. As a result, similarity matrix $S$ which has size $N \times K$ is generated.

In our proposed model, we have tested 21 similarity measures which have been mentioned in [89]. Table 1 shows the best 7 similarity measures that we will consider in this contribution. The mathematical symbols in Table 1 are defined as follows:

- $N_1$ represents the number of (1)s in the first vector.
- $N_2$ represents the number of (1)s in the second vector.
- $C_0$ represents the number of (0)s which appears simultaneously in both vectors.
- $C$ represents the number of (1)s which appears simultaneously in both vectors.
- $E_1$ represents the number of (1)s in the first vector which corresponds to (0)s in the second vector.

**Algorithm 1** The proposed K-medoids clustering algorithm.

**INPUT:** A dataset $\mathcal{X}$ , number of clusters $K$.
Set Cost $\leftarrow$ inf.
**repeat**
Set the medoid $M_j$ for each class $j$ randomly.
**for** $j = 1 \rightarrow K$ **do**
  **for** $i = 1 \rightarrow N$ **do**
    Calculate the similarity measure between each data object $X_i$ and each medoid $M_j$.
  **end for**
**end for**
Assign all data points $X_i$ to $K$ clusters by applying Eq.(2).
Calculate the new cost by applying Eq.(3).
**if** new_cost $<$ cost **then**
  Update the medoid $M_j$ for cluster $j$.
  Assign each data point to the class with the nearest medoid.
**end if**
**until** No change in the cost.

- $E_2$ represents the number of (1)s in the second vector which corresponds to (0)s in the first vector.

We calculate the minimum distance for each data object $X_i$ to the nearest medoid $M_j$, such that:

$$I_i = \min \mathcal{D}_{ij} \tag{2}$$

Each data point $X_i$ is, thus, assigned to the class with the nearest representative $M_j$ in order to increase the similarity for data objects within the same class. Updating new medoids is the next step which requires calculating the cost based on the sum of distances from all objects to their medoids, as:

$$C = \sum_{i=1}^{N} I_i \tag{3}$$

Then, the current medoid in each cluster will be updated by replacing with the new medoid. Moreover, calculating the cost will be repeated until finding the minimum cost which represents the optimal selection of medoids. The complete clustering algorithm is summarized in (Algorithm 1).

## 2.4 Experimental Results

We have applied our framework to different text and binary image datasets and compared the performance with using the Euclidean distance. Moreover, the proposed framework compares the accuracies which are calculated from the confusion matrices. We have considered 20 runs of our algorithm and reported the average accuracy.

### 2.4.1 Text Documents Clustering

Two text datasets have been applied in order to validate the quality of our algorithm based on the accuracy of the resulting confusion matrices. DBWorld e-mails [1] is a text dataset which has 64 documents, characterized by 4702 words/vocabulary, and partitioned into 2 classes (spam and no-spam). In addition, we have evaluated the performance of our proposed approach on Classic 400 [2] has 400 documents, with a vocabulary size of 6205 words categorized in 3 classes (Medline, CISI, and Cranfield). The clustering results of the two datasets are shown in Table 2.

Each document in both datasets has been represented as a fixed length binary vector [58]. Moreover, the preprocessing has been applied using Rainbow package [65]. The first step in our preprocessing is removing all stop and rare words from the vocabulary. Afterward, we perform the feature selection, then each document is represented as a binary vector which contains binary values (0s or 1s) correspond to the presence of a given word [21, 63].

As shown in Table 2, our algorithm has been successfully used for clustering text documents using the binary similarity measures. In DBWorld e-mails dataset, Euclidean distance gives 59.38% accuracy. No. Feat. Diff, Sokal Dist (a), and Sokal Dist (b) distances give the same result. The best quality of clustering which is 87.50% has been achieved using 2nd Kulcz distance in our algorithm. Otsuka and Sorgenfrei

[1] `https://archive.ics.uci.edu/ml/datasets/dbworld+e-mails#`
[2] `http://www.dataminingresearch.com/index.php/tag/dataset-2`

Table 2: Clustering results (average accuracy %) for the two text datasets.

| Datasets | Euclidean | Dice | 2nd Kulcz | Otsuka | Sorgenfrei | No. Feat. | Sokal(a) | Sokal(b) |
|---|---|---|---|---|---|---|---|---|
| DBWorld | 59.38% | 62.50% | **87.50%** | 73.44% | 73.44% | 59.38% | 59.38% | 59.38% |
| Classic 400 | 40.75% | 59.50% | **61.25%** | 56.50% | 59.25% | 43.75% | 40.75% | 40.75% |

Table 3: Confusion Matrix using Euclidean Distance for DBWorld e-mails dataset

| | Yes | No |
|---|---|---|
| Yes | **3** | 26 |
| No | 0 | **35** |

Table 4: Confusion Matrix using 2nd Kulcz for DBWorld e-mails dataset.

| | Yes | No |
|---|---|---|
| Yes | **25** | 4 |
| No | 4 | **31** |

similarity measures show the same performance which is 73.44% while Dice shows 62.50%.

In the case of Classic 400, the clustering accuracy achieved using the Euclidean distance is 40.75%. Once again, both Sokal Dist (a) and Sokal Dist (b) have provided a similar performance to the one achieved using Euclidean distance. However, No. Feat. Diff distance gives a slight improvement in the accuracy which is 43.75%. The best accuracy achieved for this dataset is 61.25% using 2nd Kulcz distance. Moreover, Dice, Otsuka , and Sorgenfrei similarity measures show improved performance with accuracies equal to 59.50%, 56.50%, and 59.25%, respectively.

To summarize, 2nd Kulcz distance commonly gives the best accuracy for both text datasets while Euclidean, and No. Feat. Diff , Sokal Dist (a), and Sokal Dist (b) distances usually provide the lowest accuracy. The improvement of our algorithm is clearly expressed in text documents clustering result using 2nd Kulcz, Dice, Otsuka, and Sorgenfrei similarity measures.

Moreover, we measure the intra-class performance based on the confusion matrices when using Euclidean and the best performing similarity measures are shown in Table 3- 6. Each entry $(i, j)$ of the confusion matrix denotes the number of documents in class $i$ that are assigned to class $j$. According these tables, it is clear that the 2nd Kulcz perform better than Euclidean distance for both datasets.

Table 5: Confusion Matrix using Euclidean Distance for Classic 400 dataset.

| | Medline | CISI | Cranfield |
|---|---|---|---|
| Medline | **7** | 0 | 93 |
| CISI | 19 | **0** | 81 |
| Cranfield | 29 | 15 | **156** |

Table 6: Confusion Matrix using 2nd Kulcz for Classic 400 dataset.

| | Medline | CISI | Cranfield |
|---|---|---|---|
| Medline | **84** | 7 | 9 |
| CISI | 84 | **3** | 13 |
| Cranfield | 33 | 9 | **158** |

### 2.4.2 Binary Images Categorization

The proposed model has been applied to two binary image datasets namely; 99 Shape [76], and MPEG [51] datasets. Each binary image is represented as a binary vector that contains (0s or 1s) correspond to the two colors used in the image which are black and white. The 99 Shape is a binary image dataset which contains 9 classes. It consists of 99 binary images of size 128×128. We considered a subset which is composed of the 5 pixels classes considered (Dude, Fish, Fly, Hand, and Tool) as shown in Figure 1. The subset contains 11, 8, 7, 11, and 11 binary images, respectively. The second binary image dataset is MPEG-7 composed of 13 classes where each class contains 20 binary images, with a total of 1400 binary images of size 256×256 pixels. In our experiment, we consider a subset which is composed of the 4 classes (Frog, Hummer, Key, and Apple) as shown in Figure 2.

The binary images clustering results are summarized in Table 7. In 99 Shape database, Euclidean distance gives 60.42% accuracy. The best performance of clustering which is 83.33% has been achieved by applying our algorithm using both No. Feat. Diff and Sokal Dist (a) similarity measures. Sokal Dist (b) similarity measure shows 81.25% a quite lower than No. Feat. Diff and Sokal Dist (a) similarity measures. Concerning MPEG-7, an accuracy of 57.50% was achieved using Euclidean distance. Using the binary similarity measures, the performance of K-medoids has been improved to 91.25% in case of using No. Feat. Diff, Sokal Dist (a), and Sokal Dist (b) distances.

It is noteworthy that No. Feat. Diff, Sokal Dist (a) similarity measures commonly

Table 7: Clustering results (average accuracy %) for the two binary image datasets.

| Datasets | Euclidean | No. Feat. Diff | Sokal Dist (a) | Sokal Dist (b) |
|---|---|---|---|---|
| 99 Shape | 60.42% | **83.33%** | **83.33%** | 81.25% |
| MPEG-7 | 57.50% | **91.25%** | **91.25%** | **91.25%** |

Table 8: Confusion Matrix using Euclidean Distance for 99 Shape dataset.

| | Dude | Fish | Fly | Hand | Tool |
|---|---|---|---|---|---|
| Dude | **10** | 0 | 0 | 0 | 1 |
| Fish | 0 | **0** | 8 | 0 | 0 |
| Fly | 1 | 4 | **2** | 0 | 0 |
| Hand | 0 | 3 | 1 | **6** | 1 |
| Tool | 0 | 0 | 0 | 0 | **11** |

Table 9: Confusion Matrix using Sokal Dist (a) for 99 Shape dataset.

| | Dude | Fish | Fly | Hand | Tool |
|---|---|---|---|---|---|
| Dude | **10** | 0 | 0 | 0 | 1 |
| Fish | 0 | **8** | 0 | 0 | 0 |
| Fly | 0 | 2 | **5** | 0 | 0 |
| Hand | 0 | 1 | 3 | **6** | 1 |
| Tool | 0 | 0 | 0 | 0 | **11** |

reach the best accuracies for both binary images datasets, while Euclidean commonly gives the lowest accuracy. The improvement of our algorithm is clearly expressed in binary images categorization using No. Feat. Diff, Sokal Dist (a) similarity measures.

The confusion matrices have been used to compare the performance of the K-medoids algorithm. Table 10 and Table 11 show the results for clustering MPEG-7 dataset using Euclidean distance and No. Feat. Diff. Moreover, the algorithm performance in categorizing 99 Shape dataset using Euclidean distance and sokal dist (a) are shown in Table 8 and Table 9, respectively. We can notice that the intra-class accuracies have been improved using No. Feat. Diff and sokal dist (a) similarity measures.

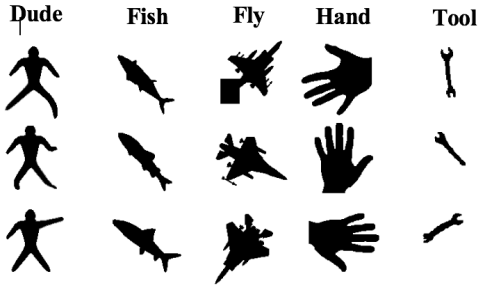


Figure 1: The considered subset of the 99 Shape database which is composed of the 5 classes.

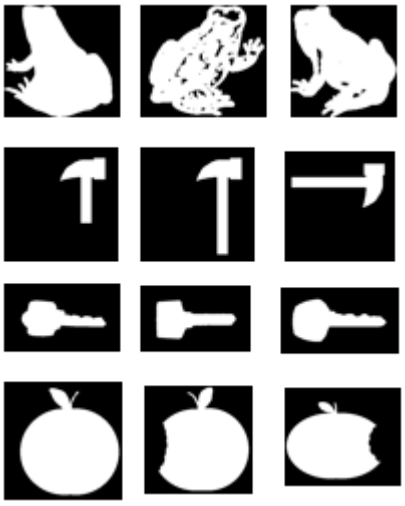

Figure 2: The considered subset of MPEG-7 Shape Database which is composed of 4 classes. The first row shows samples of the Frog class, the second row shows samples of the Hummer class, the third row shows samples of the Key class, and the fourth row shows samples of the Apple class.

Table 10: Confusion Matrix using Euclidean Distance for MPEG-7 dataset.

| | Frog | Hummer | Key | Apple |
|---|---|---|---|---|
| Frog | **6** | 0 | 0 | 14 |
| Hummer | 0 | **20** | 0 | 0 |
| Key | 0 | 0 | **20** | 0 |
| Apple | 19 | 0 | 1 | **0** |

Table 11: Confusion Matrix using No. Feat. Diff for MPEG-7 dataset.

| | Frog | Hummer | Key | Apple |
|---|---|---|---|---|
| Frog | **14** | 0 | 0 | 6 |
| Hummer | 0 | **20** | 0 | 0 |
| Key | 0 | 0 | **20** | 0 |
| Apple | 0 | 0 | 1 | **19** |

# Chapter 3

# Model-Based Hierarchical Clustering for Categorical Data

In this chapter, we introduce hierarchical clustering algorithm based on Multinomial and Bernoulli mixture models. We discuss the hierarchical clustering framework in the case of Bernoulli and Multinomial mixture models. Extensive experiments on clustering text and image as two real-world applications have shown our proposed model's merits.

## 3.1 Hierarchical Clustering

The hierarchical clustering can be addressed either in divisive (top-down) or agglomerative (bottom-up) approaches [92, 94]. Agglomerative algorithms have been primarily used for hierarchical structure solutions [43, 41, 55, 57, 81, 92]. The agglomerative tree algorithm using (top-down) mode is based on partitioning data points from a single cluster that contains all the data points to more than one cluster where each data point has its own cluster [94]. The traditional bottom-up agglomerative algorithm starts by treating each data point as a separate cluster then two clusters which are the closets will be repeatedly merged as we move up the hierarchy [45, 94]. In general, the distance between any two clusters is measured using Euclidean distance [45]. On the other hand, the traditional hierarchical clustering has some limitations. First, a theoretical justification is difficult to find for which distance measure to choose, particularly for structured data such as images. In addition, the failure of clustering data

points which are near the mean of each cluster. Moreover, the traditional hierarchical clustering approach does not provide any guidance for choosing the correct number of classes to prune the tree [55, 45].

Some previous works have been proposed to overcome these limitations including the probabilistic methods and Bayesian hierarchical clustering [45]. In [82], the authors proposed marginal likelihoods which are based on hidden Markov model structure to merge similar clusters. In Bayesian hierarchical clustering, marginal likelihood has been used in order to avoid overfitting and decide which clusters are suitable for merging [45]. In addition, Gaussian probability models have been used in [39], where the optimal clusters are chosen based on the maximum-likelihood pair which is merged at each stage. Gaussian hierarchical clustering has shown good results in different applications, such as clinical data clustering [7], social sciences, geophysical sciences, financial, and industrial data [39]. Moreover, measuring the similarity between components of mixture model based on discrete data densities has been applied for hierarchical image categorization [92]. The proposed work has overcame the limitations of the traditional hierarchical clustering.

## 3.2 Hierarchical Clustering Framework

Hierarchical clustering solutions have been primarily obtained using agglomerative algorithms [81, 42], in which objects are initially assigned to their own cluster and then pairs of clusters are repeatedly merged until the whole tree is formed. We are proposing a hierarchical clustering algorithm based on the probabilistic distance between the components of finite mixture models. Probabilistic distance measures between two probability distributions are significant metrics to evaluate the similarity for data of statistical nature. If the parameters of two Probability Density Functions (PDF) are known, or can reliably be estimated, a quantitative value can be calculated to assess how far or close the two distributions are from each other [25].

Let $\mathcal{X} = \{X_1, \ldots, X_N\}$, to be an observed dataset with $N$ instances, where each is a $\mathcal{D}$-dimensional random vector $\mathbf{X}_n = (x_1, \ldots, x_D)$ representing an image or document. Finite mixture models have been used to model a sample from a population which is composed of finite subpopulations, where the whole model is formed by a weighted sum of the densities [66]. A finite mixture model decomposes a probability

function $P(\mathcal{X}|\Theta)$, into the weighted sum of $K$ cluster probability functions where $P_j(\mathcal{X}|\theta_j)$ denotes the probability of the $j$th component. Thus, the joint probability of the finite mixture model is given by:

$$P(\mathcal{X}|\Theta) = \sum_{j=1}^{K} p_j P_j(\mathcal{X}|\theta_j) \tag{4}$$

where $\Theta = (\theta_1, \ldots, \theta_K, p_1, \ldots, p_K)$ denotes all parameters, each $p_j$ is the mixing proportion of cluster $j$ satisfying $(0 < p_j < 1), \sum_{j=1}^{K} p_j = 1$. The Maximum Likelihood Estimate (MLE) solution is a method to estimate the parameters $\Theta$ [32] by applying Expectation Maximization (EM) approach [35]. The EM algorithm has two steps which are the expectation (E-step) and the maximization (M-step). The two steps are iteratively processed until convergence. In the expectation step, the probability of a vector $X_n$ to be assigned to class $j$, called the posterior probability, is computed and given by:

$$\hat{z}_{nj} = P(j|X_n, \theta_j) = \frac{p_j P(X_n|\theta_j)}{\sum_{j=1}^{K} p_j P(X_n|\theta_j)} \tag{5}$$

In the M-step, we update the model parameter estimates according to:

$$\Theta = \arg\max \sum_{n=1}^{N} \sum_{j=1}^{K} \hat{z}_{nj} \log(p_j P(X_n|\theta_j)) \tag{6}$$

Each update to the parameters resulting from an E step followed by an M step is guaranteed to increase the log likelihood. When the change in the log likelihood function, or alternatively in the parameters, falls below some threshold, the algorithm meets the convergence condition and each data point then will be assigned to cluster which maximize its posterior probability.

Afterwards, the hierarchical approach will be applied by start merging the closest two clusters according to some probabilistic distances as we will show in the next subsections. In this work, the Battacharya and Kullback-Leibler distances have been used to measure the distance between two distributions. Bhattacharyya distance has been usually used to measure the similarity and separability of two distributions in classification problems [90]. Given that Bhattacharyya distance has the desirable properties of being computationally simple, it has been extended to measure the distance between two Gaussian distributions in Gaussian mixtures [54, 62]. In addition,

Kullback Leibler (KL) Divergence is a common similarity measure between two density distributions in statistics. In fact, Kullback Leibler (KL) distance is frequently used in Gaussian Mixture Models (GMMs), and it has been used in many pattern recognition applications such as speech and image clustering [46].

The input to the hierarchical algorithm is an $K \times K$ similarity matrix, where $K$ is prespecified number of clusters. The approach involves finding the least dissimilar pair of clusters in the current clustering, the shortest distance, and merge them into a single cluster to form the next clustering level with $K-1$ clusters. While all objects are in more than one cluster, the similarity will be updated and the closest pair of clusters will be merged repetitively. The complete hierarchical clustering algorithm is summarized in (Algorithm 2).

### 3.2.1 Hierarchical Mixture of Multinomial

The majority of the published work on unsupervised clustering has concentrated on continuous data, however, some research works considered modeling discrete data as an important component in many applications of data mining, machine learning, image processing, and computer vision [15, 14]. The Multinomial distribution is commonly used for modeling discrete variables [10]. Consider, for example, document clustering where each document is represented as a vector of counts (bag of words representation) [64]. Following a Multinomial distribution with parameters vectors $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_D)$, the probability of a random vector of counts $\mathbf{X} = (x_1, \ldots, x_D)$, is defined as:

$$p(\mathbf{X}|\theta) = \frac{(\sum_{d=1}^{D} x_d)!}{\prod_{d=1}^{D} x_d!} \prod_{d=1}^{D} \theta_d^{x_d} \tag{7}$$

The Battacharya distance and the Kullback-Leibler distance to measure the distance between two distributions/clusters. The Battacharya distance between two multinomial distributions $P(\mathcal{X}|\theta_{j_1})$ and $P(\mathcal{X}|\theta_{j_2})$ is computed as:

$$\mathcal{B}\Big(P(\mathcal{X}|\theta_{j_1}), P(\mathcal{X}|\theta_{j_2})\Big) = -\sum_{d=1}^{D} x_d \log \sum_{d=1}^{D} \sqrt{\theta_{dj_1}\theta_{dj_2}} \tag{8}$$

The Kullback-Leibler distance between two multinmoial distributions is given by:

$$\mathcal{KL}\Big(P(\mathcal{X}|\theta_{j_1}), P(\mathcal{X}|\theta_{j_2})\Big) = \sum_{d=1}^{D} \theta_{dj_1} \log \frac{\theta_{dj_1}}{\theta_{dj_2}} \tag{9}$$

### 3.2.2 Hierarchical Mixture of Bernoulli

Binary data has been used in various applications and fields such as machine learning, and computer vision where binary variables are used rather than continuous or discrete variables [16, 10]. Bernoulli mixture model is widely used for clustering binary vectors in many applications such as handwritten digit recognition, shape and visual scenes categorization [63, 22]. A multivariate Bernoulli distribution with parameters $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_D)$ for a binary vector $X = (x_1, \ldots, x_D)$, is:

$$P(X|\mu) = \prod_{d=1}^{D} \mu_d^{x_d}(1-\mu_d)^{1-x_d} \tag{10}$$

The Battacharya distance between two Bernoulli distributions $P(\mathcal{X}|\mu_{j_1})$ and $P(\mathcal{X}|\mu_{j_2})$, is given by:

$$\mathcal{B}\Big(P(\mathcal{X}|\mu_{j_1}), P(\mathcal{X}|\mu_{j_2})\Big) = -\sum_{d=1}^{D} \log\left(\sqrt{\mu_{dj_1}\mu_{dj_2}} + \sqrt{(1-\mu_{dj_1})(1-\mu_{dj_2})}\right) \tag{11}$$

The Kullback-Leibler distance between two Bernoulli distributions is calculated as:

$$\mathcal{KL}\Big(P(\mathcal{X}|\mu_{j_1}), P(\mathcal{X}|\mu_{j_2})\Big) = \sum_{d=1}^{D} \mu_{dj_1} \log \frac{\mu_{dj_1}}{\mu_{dj_2}} + (1-\mu_{dj_1}) \log \frac{(1-\mu_{dj_1})}{(1-\mu_{dj_2})} \tag{12}$$

## 3.3 Experimental Results

We have applied our framework to perform hierarchical clustering for different text and image datasets using the Bhattacharyya (BH) and Kullback-Leibler (KL) distances in case of Multinomial and Bernoulli mixture models.

### 3.3.1 Text document clustering

Three text datasets have been tested and the accuracies, which were computed based on the confusion matrices, were compared for validating the clustering quality of our algorithm. The first dataset is 7Sectors [1], which has 4,581 HTML articles partitioned

[1] `http://www.cs.cmu.edu/~webkb/`

**Algorithm 2** The proposed Hierarchical Clustering algorithm.

**INPUT:** A dataset $\mathcal{X}$ , prespecified number of clusters $K$.
**While** $K > 1$
Initialize all the parameters.
E-Step: compute the posterior probability $P(j|X_n, \theta_j)$ using Eq.(5).
M-Step: update the model parameter estimates $\Theta$ using Eq.(6).
Assign data points to clusters based on the maximum posterior probability.
**for** $i = 1 \rightarrow K$ **do**
  **for** $j = i + 1 \rightarrow K$ **do**
    Calculate the distance between cluster $i$ and cluster $j$.
  **end for**
**end for**
Find the minimum (non-zero) distance.
Update the labels to merge the closest clusters.
$K = K - 1$
**end**

into 7 classes (Materials, Energy, Financial, Health Care, Technology, Transportation, and Utilities). The second dataset is WebKb4 [2] which has 4,199 web pages partitioned into 4 clusters (Course, Faculty, Project, and Student). Lastly, NIPS [3] dataset which has 391 documents in 9 different clusters. The experiment results for clustering the three datasets using Multinomial and Bernoulli mixture models were compared in Table 12. The preprocessing has been applied to the three text datasets using the Rainbow package [65]. Each document has been represented, using the bag of words approach, as a fixed length counts vectors. Moreover, the documents have been represented as binary vectors to be used for evaluating the Bernoulli framework. As shown in Table 12, the Bhattacharyya (BH) and Kullback-Leibler (KL) distances have been successfully used for hierarchical clustering of text documents in Multinomial and Bernoulli distributions.

[2] `http://www.cs.cmu.edu/~webkb/`
[3] `https://cs.nyu.edu/~roweis/data.html`

Table 12: Hierarchical Clustering (HC) results (average accuracy %) for the three text datasets.

| | Multinomial | | | Bernoulli | | |
|---|---|---|---|---|---|---|
| Datasets | Accuracy | HC KL | HC BH | Accuracy | HC KL | HC BH |
| 7 Sectors | 49.18% | 70.35% | 70.35% | 73.56% | 97.00% | 97.00% |
| WebKb4 | 86.33% | 96.74% | 96.74% | 90.64% | 97.45% | 97.45% |
| NIPS | 79.03 % | 90.28% | 90.28% | 82.10% | 97.95% | 94.63% |

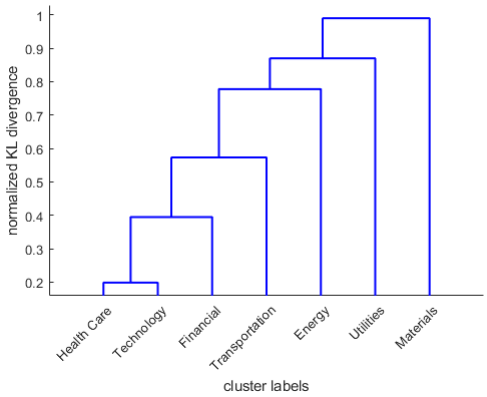


Figure 3: Dendrogram for hierarchical clustering using mixture of Multinomial distributions showing the closest clusters in 7 Sectors based on KL distance.

In Multinomial model, the clustering of 7Sectors dataset using mixture approach gives 49.18% accuracy, while the hierarchical clustering based on KL and BH gives 70.35% which shows an improvement of performance using our framework. In addition, the clustering of WebKb4 and NIPS datasets show higher accuracies using our framework compared to the accuracy of using mixture model clustering. For instance, the accuracy of the hierarchical clustering of WebKb4 is 96.74% based on KL and BH distances between two multinomial distributions.

Moreover, the hierarchical clustering based on Bernoulli distribution has achieved an excellent performance. Indeed, the clustering of 7Sectors and WebKb4 datasets using Bernoulli mixture approach gives accuracies of 73.56%, and 90.64%, respectively. On the other hand, applying hierarchical clustering using Kullback-Leibler and Bhattacharyya distances to 7Sectors and WebKb4 datasets gives around 97%, while it gives a quite lower performance for NIPS dataset by using Bhattacharyya distance. To summarize, the improvements have been achieved using hierarchical clustering based on KL and BH distances for the both distributions in text clustering.

We have demonstrated the tree result from the hierarchical clustering of 7Sectors using Multinomial model based on KL and BH distances in Fig. (3) and (4), respectively. We can see that different distances have considered the similarity between clusters differently. For instance, after merging Health Care and Technology clusters to Financial cluster using Bhattacharyya, the Financial cluster was the closest to Energy cluster. In contrast, the Financial cluster was first merged to Transportation cluster as the closest cluster, then Transportation was found to be the most similar

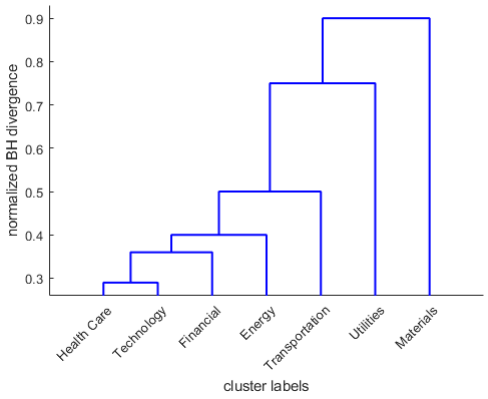


Figure 4: Dendrogram for hierarchical clustering using mixture of Multinomial distributions showing the closest clusters in 7 Sectors based on Bhattacharyya distance.

Table 13: Average accuracy of each level of the tree in NIPS dataset based on Bernoulli mixture.

| | KL | BH |
|---|---|---|
| 9 Clusters | 82.09% | 82.09% |
| 8 Clusters | 84.91% | 86.70% |
| 7 Clusters | 83.88% | 88.74% |
| 6 Clusters | 83.63% | 92.07% |
| 5 Clusters | 82.86% | 93.60% |
| 4 Clusters | 85.93% | 92.32% |
| 3 Clusters | 78.77% | 93.09% |
| 2 Clusters | 97.95% | 94.62% |

to them.

Furthermore, the accuracy of each level after merging the similar clusters in NIPS dataset using Bernoulli model is shown in Table 13. We can see that the performance has been improved from 82.09% (first level), which expressed the accuracy of mixture model clustering, to 97.95% and 94.62% (last level) based on KL and BH distances, respectively. The performance was slightly improved from the first level to the last level using BH distance while it was steadily improved using KL distance. According to previous results, it is clear that the hierarchical clustering performs better than mixture model clustering in both Multinomial and Bernoulli distributions.

### 3.3.2 Images Categorization

For this application, we have used the bag of features approach for representing the images as a vector of count, or binary, values as this approach has been successfully applied in computer vision applications. For instance, visual words have been analyzed as low-level image features in order to learn and categorize images in [21]. Two datasets were used as follows. The first dataset is a subset of the extensive Scene UNderstanding (SUN) database [88], that contains 899 categories and $130,519$ images. We have used $1,849$ natural scenes belonging to six categories (458 coasts, 228 river, 231 forests, 247 field, 518 mountains, and 167 sky/clouds). The average size of the images is $720 \times 480$ (landscape format) or 480 (portrait format). Example images from each class are shown in Fig. (5). The Second dataset is KTH-TIPS [4] which is called Textures under varying Illumination, Pose and Scale. KTH-TIPS contains 81 images which are divided into 10 classes (materials) as shown in Fig. (6). The size of images is 200 x 200 pixels.

Each dataset was splitted randomly into two halves; one for constructing the vocabulary and one for representation. We used Scale-Invariant Feature Transform (SIFT) [60] to detect the key points and compute the descriptors. After extracting the features from the first half, they are used to generate a codebook by quantizing the descriptors into a number of homogeneous clusters using a k-means algorithm, where the centroid of each cluster is treated as a visual word. Then, in each novel image, the extracted descriptors are assigned to the closest visual word (Euclidean distance) resulting in a histogram of frequencies that can be also binarized. The results of hierarchical clustering for SUN and KTH-TIPS datasets based on KL and BH distances in Multinomial and Bernoulli mixtures are shown in Table 14.

[4] `http://www.nada.kth.se/cvap/databases/kth-tips/documentation.html`

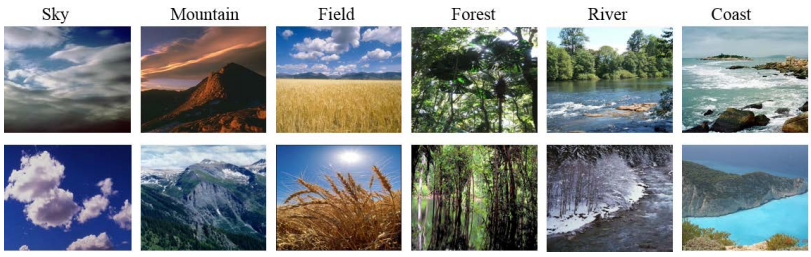


Figure 5: A Subset of The SUN which is composed of the 6 clusters.


Figure 6: Samples of each class in KTH texture dataset.

Table 14: Hierarchical Clustering (HC) results (average accuracy %) for the two image datasets.

| | Multinomial | | | Bernoulli | | |
|---|---|---|---|---|---|---|
| Datasets | Accuracy | HC KL | HC BH | Accuracy | HC KL | HC BH |
| KTH | 78.57% | 97.42% | 97.42% | 78.57% | 97.71% | 95.42% |
| SUN | 29.19% | 82.29% | 67.39% | 24.53% | 71.73% | 86.95% |

In Multinomial model, the accuracies of clustering KTH-TIPS and SUN datasets using mixture model are 78.57% and 29.19%, respectively. The accuracies of hierarchical clustering are based on both distances are 97.42% for KTH-TIPS. However, the proposed algorithm gives 82.29% and 67.39% for SUN data based on KL and BH distances, respectively. In Bernoulli model, the performance of hierarchical clustering for SUN dataset, for instance, is improved from 24.53% to around 71% and 86% using KL and BH, respectively.

Fig. (7) and (8) show the dendrograms of hierarchical clustering using KTH-TIPS

Table 15: Average accuracy of each level of the tree in SUN dataset based on Multinomial mixture.

| | KL | BH |
|---|---|---|
| 6 Clusters | 29.19% | 29.19% |
| 5 Clusters | 46.58% | 46.58% |
| 4 Clusters | 52.17% | 56.21% |
| 3 Clusters | 73.91% | 64.90% |
| 2 Clusters | 82.29% | 67.39% |

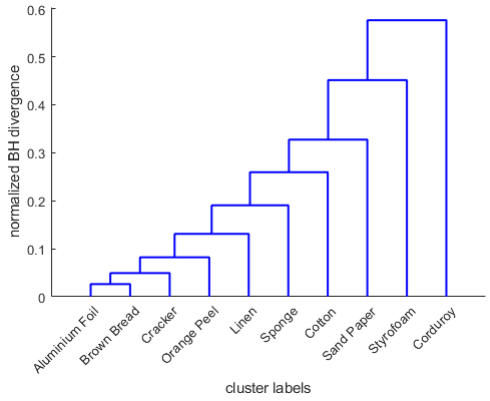


Figure 7: Dendrogram for hierarchical clustering using mixture of Bernoulli distributions showing the closest clusters in KTH based on Bhattacharyya distance.

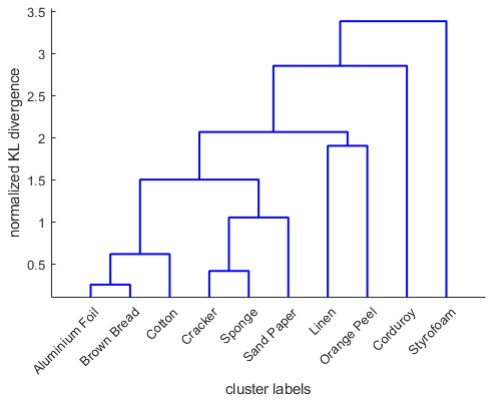


Figure 8: Dendrogram for hierarchical clustering using mixture of Bernoulli distributions showing the closest clusters in KTH based on KL distance.

dataset in Bernoulli distribution based on Bhattacharyya and Kullback-Leibler. The two dendrograms are different such as after Aluminum Foil and Brown Bread are merged to one cluster, Cracker is merged to Aluminum Foil based on Bhattacharyya while Sponge is the closest cluster to Cracker based on Kullback-Leibler distance.

On the other hand, the average accuracy of each level of the Dendrogram for SUN dataset in Multinomial mixture is shown in Table 15. For each level of merging, the accuracy is increased and expressed similar percentages for both distances except the last 2 levels where the percentages are quite different. The last level accuracy is 82% using KL, and 67% using BH, which are significant improvements compared to 29.19% in first level.

# Chapter 4

# A Hybrid Approach Based on SVM and Bernoulli Mixture Model for Binary Vectors Classification

In this chapter, we detail our findings when Support Vector Machines (SVMs), as a powerful classification tool, and Bernoulli mixture model are combined in order to classify binary data. First we discuss hybrid learning approach and the finite Bernoulli mixture model and then we present the proposed hybrid generative/discriminative learning approach and all related details about the proposed kernels. Finally, we demonstrate the merits of the proposed approach for the problem of classifying binary and texture images.

## 4.1 Hybrid Generative/Discriminative Learning Approach

Extracting classification rules from samples is an important procedure and a challenging task in various machine learning, pattern recognition and computer vision applications. Learning approaches are generally divided into two groups, namely, generative and discriminative approaches [12]. The objective of using generative learning approaches is the estimation of class conditional distributions $P(X|j)$ for $j = 1, \ldots, K$, where $K$ is the overall number of classes [11, 20]. In contrast, the goal of applying discriminative approaches is to estimate the classification function $j = f(X)$ instantly

from the data [73]. Discriminative models are based on two phases: training and classification phases. In the training phase, the boundaries between several categories as classes are defined through maximizing the margin between data. A new unseen data will be assigned to a class based on the side of the boundary, which mapped the data [9]. Even though there are different advantages of using the two approaches, the limitations vary from one approach to another. For example, constructing the decision boundaries leads to excellent classification performance when using discriminative approaches while missing and incomplete data can effectively be handled by generative approaches due to their efficiency in handling uncertainty. However, failing to provide an acceptable clustering or classification performance for new data is limiting traditional generative or discriminative approaches. A great comprehension of the strengths and weaknesses of each approach leads to an increasing interest in hybrid approaches. Combining the desirable capabilities of both approaches is the idea of hybrid approaches. For example, the data's substantial properties can be captured to classify while prior knowledge of the problem domain is taking into consideration [71].

Hybrid generative/discriminative learning approaches have been successfully implemented to improve the classification process by incorporating prior knowledge of the data involved in several applications. For instance, powerful models to classify proportional data have been proposed in [18, 17], where generative kernels for SVM were generated from Dirichlet, generalized Dirichlet and Beta-Liouville mixture models. Moreover, hybrid learning approach for mixed data has been developed based on a hidden Markov model [37]. Furthermore, the Langevin mixture model has been used to classify spherical data using a hybrid learning approach [6]. Recently, Zamzami and Bouguila proposed hybrid methods, based on a novel Multinomial Scaled Dirichlet [91], and based on mixtures of exponential family approximation to other powerful generative models for count data modeling [93].

## 4.2 Finite Bernoulli Mixture Model

Bernoulli mixture model is generally used for clustering binary data in many machine learning and computer vision applications [16, 10], such as visual scenes, shape categorization, and handwritten digit recognition [63, 22]. A multivariate Bernoulli

distribution with parameters $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_D)$ for a $D$-dimensional binary vector $X = (x_1, \ldots, x_D)$, is given by:

$$P(X|\theta) = \prod_{d=1}^{D} \theta_d^{x_d}(1-\theta_d)^{1-x_d} \tag{13}$$

Let $\mathcal{X} = \{X_1, \ldots, X_N\}$ be a dataset of binary vectors, where $N$ represents the number of instances or observations, e.g., images. Finite mixture models are applied to model samples collected from a finite number of sub-populations. The form of the whole model is formed by a weighted sum of the densities [66]. A probability function $P(\mathcal{X}|\Theta)$ is decomposed into the sum of $K$ components probability density functions which are weighted. Thus, the probability of the finite mixture model is presented by:

$$P(\mathcal{X}|\Theta) = \prod_{n=1}^{N} \sum_{j=1}^{K} \pi_j P(X_n|\theta_j) \tag{14}$$

where $\Theta = (\theta_1, \ldots, \theta_K, \pi_1, \ldots, \pi_K)$ indicates all the latent variables of the mixture model while the mixing proportion of a cluster $j$ is denoted by $\pi_j$ and satisfies $(0 < \pi_j < 1), \sum_{j=1}^{K} \pi_j = 1$. A widely used method for estimating the set of parameters $\Theta$ is the Maximum Likelihood Estimate (MLE) solution [32] where Expectation Maximization (EM) algorithm is applied [35]. The EM algorithm consists of two steps which are the expectation (E-step) and the maximization (M-step) that iteratively run until convergence. The posterior probability is calculated in the expectation step, where a vector $X_n$ is assigned to class $j$ that maximizes its posterior probability [3]. The computation of the posterior probability is given by:

$$\hat{z}_{nj} = P(j|X_n, \theta_j) = \frac{\pi_j P(X_n|\theta_j)}{\sum_{j=1}^{K} \pi_j P(X_n|\theta_j)} \tag{15}$$

For updating the model parameters, we maximize the log likelihood function in the M step according to:

$$\Theta = \arg\max \sum_{n=1}^{N} \sum_{j=1}^{K} \hat{z}_{nj} \log(p_j P(X_n|\theta_j)) \tag{16}$$

Maximizing the previous equation, results in the updated parameter $\theta_j$, and $\pi_j$, such that:

$$\pi_j = \frac{\sum_{n=1}^{N} \hat{z}_{nj}}{N} \tag{17}$$

$$\theta_j = \frac{\sum_{n=1}^{N} \hat{z}_{nj} x_n}{\sum_{n=1}^{N} \hat{z}_{nj}} \tag{18}$$

## 4.3 The Proposed Hybrid Learning Approach

Support Vector Machines (SVMs) are powerful tools widely used for supervised learning in classification and regression problems [8, 19]. The SVM classifier was originally introduced in [84], and significantly has an increased popularity because of its advantages such as good generalization, global solution, the number of tuning parameters and their solid theoretical foundation. Therefore, the development of efficient SVMs implementations leads to extend its application [33, 61, 68]. Even though SVMs are considered as powerful tools, the choice of the kernel function $\mathcal{K} : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ for non-separable data is a challenging task [24]. The kernel function is responsible for measuring the similarity between input vectors. When the data are not considered linearly separable, the kernel function can be used to map the data into a high dimensional feature space. Therefore, the computation of the inner product value of the transformed data in the feature space is simplified [9, 77]. In general, Radial Basis Function (RBF) and sigmoid are widely used kernel functions [56]. In most of the applications, it was indicated that the best choice is not the classic SVM kernels [5, 23, 13]. Generating the kernel function directly from data can achieve better results. Generating SVM kernels based on information divergence between distributions is one of the successful approaches. As a similarity measure between input vectors, a given kernel should capture the intrinsic properties of the data to classify, and take into account prior knowledge of the problem domain.

Let a considered dataset has a set of objects (e.g. images) $\mathcal{O} = \{O_1, \ldots, O_N\}$, where each image $O_i$ represents a sequence of feature vectors $\mathcal{X}_{O_i} = \{X_{O_{i1}}, ..., X_{O_{iT}}\}$. Each individual image $\mathcal{X}_{O_i}$ is represented by a bag of pixel vectors of a set of local descriptors [40, 52]. Therefore, each image has its own size T. Let $\mathcal{X}_{\mathbf{O_i}} = \{X_{O_{i1}}, ..., X_{O_{iT}}\}$ and $\mathcal{X'}_{\mathbf{O_i}} = \{X'_{O_{i1}}, \ldots, X'_{O_{iT}}\}$ to be two multimedia objects $O$ and $O'$, respectively

represented as sequences of feature vectors. The model of two objects are based on two probability density functions $P(X|\Theta)$ and $P'(X|\Theta')$, respectively. The idea to generate kernel functions to be computed in the PDF space instead of the original sequence space.

**The Kullback–Leibler Kernel:** measures the dissimilarity between $P(X|\Theta)$ and $P'(X|\Theta')$, as two probability distributions based on using Kullback–Leibler (KL) divergence, and it is given by:

$$K(P(X|\Theta), P'(X|\Theta')) = \exp\Big[\frac{-f_{\mathcal{KL}}(P(X|\Theta), P'(X|\Theta'))^2}{2\sigma^2}\Big] \tag{19}$$

where $\sigma$ is a constant value which can be changed based on the data while $f_{\mathcal{KL}}(P(X|\Theta), P'(X|\Theta'))$ is KL divergence proposed in [74], is given by:

$$f_{\mathcal{KL}}(P(X|\Theta), P'(X|\Theta')) = \sum_{d=1}^{D} x_d \log \frac{x_d}{x'_d} + x'_d \log \frac{x'_d}{x_d} \tag{20}$$

**The Bhattacharyya kernel**: has a main advantage of nonlinear flexibility. The Bhattacharyya kernel between $P(X|\Theta)$ and $P'(X|\Theta')$ has been originally proposed in [53]. In the case of Bernoulli distribution, it is given by:

$$K_{\mathcal{B}}(P(X|\Theta), P'(X|\Theta')) = \prod_{d=1}^{D}[(\theta_d\theta'_d)^\rho + (1-\theta_d)^\rho(1-\theta'_d)^\rho] \tag{21}$$

where $\theta$ is the parameters of the Bernoulli distribution, while $\rho$ is a constant which is equal to $\frac{1}{2}$.

**The Rényi divergence**: The Rényi divergence [83] between $P(X|\Theta)$ and $P'(X|\Theta')$ in the case of Bernoulli distribution is given by:

$$K_{\mathcal{R}}(P(X|\Theta), P'(X|\Theta')) = \frac{1}{\alpha-1}\ln \sum_{d=1}^{D} \frac{(x_d)^\alpha}{(x'_d)^{\alpha-1}} \tag{22}$$

where $\alpha$ is a constant value which can be changed based on a data satisfying $\alpha \neq 1$. When $\alpha = 1$ the Rényi divergence intends to be Kullback–Leibler divergence [83].

The proposed hybrid approach involves several steps to be performed in order to achieve an optimal performance. The first step is the initialization step; where

**Algorithm 3** The proposed hybrid learning approach algorithm.

**INPUT:** A dataset $\mathcal{X}$ , prespecified number of clusters $K$.
**Initialize all the parameters:**
Apply the K-Means to get the parameters.
Apply the Method-of-Moment for initializing the parameters of Bernoulli mixture model $\Theta = (\theta_1, \ldots, \theta_K, \pi_1, \ldots, \pi_K)$.
**E-Step:** compute the posterior probability $P(j|X_n, \theta_j)$ using Eq.(15).
**M-Step:** update the model parameter estimates $\Theta$ using Eq.(18) and Eq.(17).
Assign data points to clusters based on the maximum posterior probability.
**for** $i = 1 \rightarrow K$ **do**
  **for** $j = i + 1 \rightarrow K$ **do**
    Calculate the kernel between component $i$ and $j$ using Eq.(19), (21) or Eq.(22).
  **end for**
**end for**
Feed the SVM classifier by the Kernel matrix. **end**

a K-Means algorithm is used for initializing the mixing weight parameter $\pi$, and $\theta$ parameters are initialized using the method of moments where the number of classes $K$ is chosen based on the best experiment when $K = \{2, \ldots, 7\}$. The second step is learning the mixture model by using the Expectation-Maximization (EM) algorithm for estimating the parameters. The following step is based on computing the dissimilarity between each two mixture components, which generates the kernels. Finally, the kernel matrices are feed to the SVM classifier to discriminate the data vectors. The proposed hybrid learning approach is summarized in Algorithm (3).

## 4.4 Experimental Results

In this section, we have evaluated the hybrid approach based on SVM and Bernoulli mixture model using different generative kernels to classify binary vectors. The variable vectors have been given as feature vectors which contain (0s or 1s) correspond to black and white pixels in binary images, or bit-plane-level features under varying illumination, pose and scale for texture images. In our experiments, the 1-versus-all training approach has been used with performing 10-fold cross-validation. The accuracy of overall well-classified elements has been used as a measure of the model performance based on averaging the results of 10 independent runs. Moreover, we have compared the proposed hybrid learning approach with a pure discriminative approach, such as SVM with classic kernels and pure generative approach, i.e., the

Bernoulli mixture model (BMM).

### 4.4.1 Binary Images Categorization

The proposed framework has been applied to subsets of two binary image datasets, namely, 99 Shape [76], and MPEG [51]. The considered subsets are shown in Figure (1) and Figure (2).

The results for classifying the binary images using the proposed hybrid approach are summarized in Table 16. In our experiments, the constant values used are $\sigma$=6 and $\alpha$=6 which were selected by experiments for Kullback–Leibler (KL) and Rényi kernels, respectively. In the 99 Shape dataset experiment, the best classification performance with an accuracy of 73.50% has been achieved in the case of applying the Kullback–Leibler (KL) kernel. Then, Bhattacharyya and Rényi kernels show average accuracies of 68.00% and 62.50%, respectively, which are quite lower than the once achieved using Kullback–Leibler (KL) kernel. On the other hand, the classification results using SVM with standard kernels such as Radial Basis Function (RBF) and sigmoid give 27.00% and 54.50% accuracy, respectively, and with the Bernoulli mixture model, we get 22.91%.

Concerning MPEG-7, the achieved accuracies of SVM with RBF and sigmoid kernel, and with the Bernoulli mixture model are 20.75%, 42.25%, and 35.00%, respectively. Feeding the SVM with probabilistic kernels based on measuring the divergence between two distributions has improved the classification rate to 70.00% using Kullback–Leibler (KL) divergence, 63.75% and 52.50% using Bhattacharyya and Rényi kernels, respectively. It is remarkable that the best accuracies have been achieved using Kullback–Leibler (KL), Bhattacharyya, and Rényi kernels for both binary image

Table 16: Binary image categorization performance (average accuracy %) using different techniques.

| Approach | 99 Shape | MPEG-7 |
|---|---|---|
| BMM+ Kullback–Leibler (KL) | **73.50%** | **70.00%** |
| BMM+ Bhattacharyya | 68.00% | 63.75% |
| BMM+ Rényi | 62.50% | 52.50% |
| SVM (RBF kernel) | 27.00% | 20.75% |
| SVM (Sigmoid kernel) | 54.50% | 42.25% |
| BMM | 22.91% | 35.00% |

datasets.

### 4.4.2 Texture Images Categorization

For this application, we have used the bit-plane probability (BP), which is based on the product of Bernoulli distributions (PBD) for transforming texture image datasets as wavelet subband histograms, where each bit-plane contains binary bits (0 or 1), as proposed in [30]. Subsets of two texture image datasets have been used to validate the model. The first texture image dataset is KTH-TIPS [1]. The complete KTH-TIPS dataset contains 10 classes each has 81 images of size 64 $\times$ 64 pixels. In our experiment, we have selected a subset of 5 classes, which are Aluminium foil, Brown bread, Corduroy, Cotton, and Cracker, as shown in Fig. (9). The second texture image dataset is called DTD dataset [31], which has 47 classes. The total number of 120 texture images of size 64 $\times$ 64 pixels are assigned to each class. We have considered a subset with 4 classes, including Dotted, Fibrous, Flecked, and Freckled, as shown in Fig. (10).

In order to compare our framework with a pure generative approach, i.e., BMM and discriminative techniques, we need to represent each image as a single vector with binary variables. Hence, we considered the bag of visual words approach [34], then the histogram of frequencies has been binarized (each visual word has a value of 1 if it appears in the image and zeroes otherwise [21]). For creating the BovW representation, we have split each dataset into two halves; one for constructing the vocabulary, and the second for representation. The extracting features from the first half, the

[1] `http://www.nada.kth.se/cvap/databases/kth-tips/documentation.html`


Figure 9: Samples of 5 classes (Aluminium foil, Brown bread, Corduroy, Cotton, and Cracker) in KTH texture dataset.


Figure 10: Samples of 4 classes (Dotted, Fibrous, Flecked, and Freckled) in DTD texture dataset.

wavelet subband histograms in our case, have been clustered using a k-means algorithm where the centroids are considered as the visual words ($W_1 \dots W_D$). For each image in the second half, Euclidean distance is calculated to assign each extracted wavelet subband histogram feature to the closest visual word. Consequentially, each texture image will be presented as a histogram of frequencies, which can be binarized in order to train a Bernoulli mixture model. The results for texture images classification are summarized in Table 17.

In the KTH-TIPS dataset, the assigned constant values when using Kullback–Leibler (KL) is $\sigma$=2, and when using Rényi kernels $\alpha$= 17 which were selected experimentally. The accuracy for classifying the texture images using the Bernoulli mixture model is 20.00%, while in the case of using SVM with RBF kernel is 32.04%. The classification performance is improved using the proposed approach, where the accuracy has been greatly enhanced to 57.23% and 49.51% using the Bhattacharyya and Kullback–Leibler (KL), respectively. Moreover, the result of using Rényi kernel has significantly better performance than using the Bernoulli mixture model or SVM with the classic RBF kernel. The results of using Bhattacharyya and Kullback–Leibler

Table 17: Texture image categorization performance (average accuracy %) using different techniques.

| Approach | KTH-TIPS | DTD |
|---|---|---|
| BMM+ Kullback–Leibler (KL) | 49.51% | 40.20% |
| BMM+ Bhattacharyya | **57.23%** | **44.37%** |
| BMM+ Rényi | 34.27% | 37.29% |
| SVM (RBF kernel) | 32.04% | 35.83% |
| SVM (Sigmoid kernel) | 34.16% | 35.83% |
| BMM | 20.00% | 25.00% |

(KL) kernels in classifying DTD have shown a considerable improvement in the performance with an accuracy of 44.37% and 40.20%, respectively comparing to the accuracy achieved in the case of using Bernoulli mixture model which is 25.00%. Furthermore, the accuracy achieved using the Renyi kernel is 37.29%, which is slightly better than the accuracy in the case of using classic kernels such as RBF and Sigmoid.

The results of both applications suggest that the classification performance of the proposed hybrid approach using generative kernels based on information divergences has the ability to integrate prior knowledge regarding the nature of data involved in the problem and thus grants good data discrimination.

# Chapter 5

# Conclusion

This thesis has developed different clustering and classification approaches to improve the accuracy of clustering and classifying categorical data, specifically binary data.

In chapter 2, we have proposed an extension of the K-medoids algorithm to cluster binary data using different binary sequences similarity measures. Clustering binary data is important due to its existence in different applications. Two applications have been used to validate the proposed framework. The best performance has been achieved by using 2nd Kulcz and No. Feat. Diff similarity measures for text documents clustering and binary images categorization, respectively. The accuracy has been improved up to 91.25% in categorizing binary images and up to 87.50% in clustering text documents which is a significant improvement compared to the existing approach. We then conclude that using similarity measures instead of Euclidean leads to improve the performance of K-medoids for clustering binary data. There are some limitations of this work include handling data with large number of features and/or classes, as well as the randomly initialized medoids which give different results for each run. The proposed framework can be used in the parameter initialization for Expectation-Maximization (EM) algorithm with binary data instead of K-means which is usually used in the initialization step.

Then, in chapter 3, we have proposed a hierarchical clustering approach based on Bhattacharyya and Kullback-Leibler distances to cluster categorical data based on Multinomial and Bernoulli mixture models. The proposed framework has been applied to two real world applications, namely, text clustering and image categorization. The comprehensive performance analysis has shown that Bhattacharyya

and Kullback-Leibler distances can be efficiently used to measure similarity between two discrete distributions. The proposed approach can be applied to many other applications which involve hierarchy structure and count or binary data.

Finally, in chapter 4, we have proposed a hybrid learning approach based on two probabilistic kernels, namely, Bhattacharyya and Rényi kernels from the Bernoulli mixture model to classify binary vectors. The objective of developing the hybrid learning approach is to combine a generative model, i.e., the Bernoulli mixture model, with a supervised learning technique, namely Support Vector Machines (SVMs). The Bernoulli mixture model is used to generate probabilistic kernels which are computed to feed the SVM classifier. We validated the proposed learning approach via two different applications that involve binary and texture image categorization. The results demonstrate that the proposed algorithm is a powerful tool that provides better accuracie than either fully generative or discriminative techniques. The best classification performance has been achieved by using Kullback–Leibler (KL) and Bhattacharyya kernels for binary and texture images categorization, respectively.

The experiments with proposed frameworks are motivating and proves to be a better solution than classic K-means, traditional hierarchical clustering and fully generative or discriminative techniques for binary data. Future works might include making the fusion approach to handle mixed data that contains binary data. The efficient propagation and aggregation of Bernoulli mixture models (BMMs) in a decentralized fashion in a network is called gossip-based computation, which might be proposed in future work to improve estimates over time.

# Bibliography


[1] Charu C Aggarwal and Chandan K Reddy. *Data clustering: algorithms and applications*. CRC press, 2013.

[2] Fahdah Alalyan, Nuha Zamzami, Manar Amayri, and Nizar Bouguila. An improved k-medoids algorithm based on binary sequences similarity measures. In *2019 6th International Conference on Control, Decision and Information Technologies (CoDIT)*, pages 1723–1728. IEEE, 2019.

[3] Fahdah Alalyan, Nuha Zamzami, and Nizar Bouguila. Model-based hierarchical clustering for categorical data. In *2019 IEEE 28th International Symposium on Industrial Electronics (ISIE)*, pages 1424–1429. IEEE, 2019.

[4] Fahdah Alalyan, Nuha Zamzami, and Nizar Bouguila. A hybrid approach based on svm and bernoulli mixture model for binary vectors classification. In *2020 IEEE International Conference on Systems, Man and, Cybernetics (SMC)*. IEEE, 2020. (in press).

[5] Ola Amayri and Nizar Bouguila. Content-based spam filtering using hybrid generative discriminative learning of both textual and visual features. In *2012 IEEE International Symposium on Circuits and Systems, ISCAS 2012, Seoul, Korea (South), May 20-23, 2012*, pages 862–865. IEEE, 2012.

[6] Ola Amayri and Nizar Bouguila. Beyond hybrid generative discriminative learning: spherical data classification. *Pattern Analysis and Applications*, 18(1):113–133, 2015.

[7] Jeffrey D Banfield and Adrian E Raftery. Model-based gaussian and non-gaussian clustering. *Biometrics*, pages 803–821, 1993.

[8] Taoufik Bdiri and Nizar Bouguila. Bayesian learning of inverted dirichlet mixtures for SVM kernels generation. *Neural Computing and Applications*, 23(5):1443–1458, 2013.

[9] Christopher M Bishop. *Pattern recognition and machine learning*. springer, 2006.

[10] Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag, Berlin, Heidelberg, 2006.

[11] Christopher M Bishop et al. *Neural networks for pattern recognition*. Oxford university press, 1995.

[12] N. Bouguila. A data-driven mixture kernel for count data classification using support vector machines. In *2008 IEEE Workshop on Machine Learning for Signal Processing*, pages 26–31, 2008.

[13] N. Bouguila. A model-based discriminative framework for sets of positive vectors classification: Application to object categorization. In *2014 1st International Conference on Advanced Technologies for Signal and Image Processing (ATSIP)*, pages 277–282, 2014.

[14] N. Bouguila and D. Ziou. Improving content based image retrieval systems using finite multinomial dirichlet mixture. In *Proceedings of the 2004 14th IEEE Signal Processing Society Workshop Machine Learning for Signal Processing, 2004.*, pages 23–32, 2004.

[15] Nizar Bouguila. A model-based approach for discrete data clustering and feature weighting using MAP and stochastic complexity. *IEEE Trans. Knowl. Data Eng.*, 21(12):1649–1664, 2009.

[16] Nizar Bouguila. On multivariate binary data clustering and feature weighting. *Computational Statistics & Data Analysis*, 54(1):120–134, 2010.

[17] Nizar Bouguila. Bayesian hybrid generative discriminative learning based on finite liouville mixture models. *Pattern Recognition*, 44(6):1183–1200, 2011.

[18] Nizar Bouguila. Hybrid generative/discriminative approaches for proportional data modeling and classification. *IEEE Transactions on Knowledge and Data Engineering*, 24(12):2184–2202, 2011.

[19] Nizar Bouguila. Deriving kernels from generalized dirichlet mixture models and applications. *Inf. Process. Manag.*, 49(1):123–137, 2013.

[20] Nizar Bouguila and Ola Amayri. A discrete mixture-based kernel for svms: Application to spam and image categorization. *Inf. Process. Manag.*, 45(6):631–642, 2009.

[21] Nizar Bouguila and Khalid Daoudi. Learning concepts from visual scenes using a binary probabilistic model. In *2009 IEEE International Workshop on Multimedia Signal Processing, MMSP '09, Rio de Janeiro, Brazil, October 5-7, 2009*, pages 1–5, 2009.

[22] Nizar Bouguila and Khalid Daoudi. A statistical approach for binary vectors modeling and clustering. In *Advances in Knowledge Discovery and Data Mining, 13th Pacific-Asia Conference, PAKDD 2009, Bangkok, Thailand, April 27-30, 2009, Proceedings*, pages 184–195, 2009.

[23] Sami Bourouis, Atef Zaguia, and Nizar Bouguila. Hybrid statistical framework for diabetic retinopathy detection. In Aurélio Campilho, Fakhri Karray, and Bart M. ter Haar Romeny, editors, *Image Analysis and Recognition - 15th International Conference, ICIAR 2018, Póvoa de Varzim, Portugal, June 27-29, 2018, Proceedings*, volume 10882 of *Lecture Notes in Computer Science*, pages 687–694. Springer, 2018.

[24] Sami Bourouis, Atef Zaguia, Nizar Bouguila, and Roobaea Alroobaea. Deriving probabilistic SVM kernels from flexible statistical mixture models and its application to retinal images classification. *IEEE Access*, 7:1107–1117, 2019.

[25] Sung-Hyuk Cha. Comprehensive survey on distance/similarity measures between probability density functions. *City*, 1(2):1, 2007.

[26] Sung-Hyuk Cha and Sargur N Srihari. A fast nearest neighbor search algorithm by filtration. *Pattern Recognition*, 35(2):515–525, 2002.

[27] Sung-Hyuk Cha, Charles C Tappert, and Sargur N Srihari. Optimizing binary feature vector similarity measure using genetic algorithm and handwritten character recognition. In *null*, page 662. IEEE, 2003.

[28] Sung-Hyuk Cha, Sungsoon Yoon, and Charles C Tappert. Enhancing binary feature vector similarity measures. 2005.

[29] Seung-Seok Choi, Sung-Hyuk Cha, and Charles C Tappert. A survey of binary similarity and distance measures. *Journal of Systemics, Cybernetics and Informatics*, 8(1):43–48, 2010.

[30] Siu-Kai Choy and Chong-Sze Tong. Statistical properties of bit-plane probability model and its application in supervised texture classification. *IEEE Transactions on Image Processing*, 17(8):1399–1405, 2008.

[31] Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3606–3613, 2014.

[32] Gesteira Costa Filho et al. *Mixture Models for the Analysis of Gene Expression*. PhD thesis, 2008.

[33] Nello Cristianini, John Shawe-Taylor, et al. *An introduction to support vector machines and other kernel-based learning methods*. Cambridge university press, 2000.

[34] Gabriella Csurka, Christopher Dance, Lixin Fan, Jutta Willamowski, and Cédric Bray. Visual categorization with bags of keypoints. In *Workshop on statistical learning in computer vision, ECCV*, volume 1, pages 1–2. Prague, 2004.

[35] Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society. Series B (methodological)*, pages 1–38, 1977.

[36] Richard O Duda, Peter E Hart, and David G Stork. *Pattern classification*. John Wiley & Sons, 2012.

[37] Elise Epaillard and Nizar Bouguila. Hybrid hidden markov model for mixed continuous/continuous and discrete/continuous data modeling. In *2015 IEEE 17th International Workshop on Multimedia Signal Processing (MMSP)*, pages 1–6. IEEE, 2015.

[38] Jeffrey Erman, Martin Arlitt, and Anirban Mahanti. Traffic classification using clustering algorithms. In *Proceedings of the 2006 SIGCOMM workshop on Mining network data*, pages 281–286. ACM, 2006.

[39] Chris Fraley. Algorithms for model-based gaussian hierarchical clustering. *SIAM Journal on Scientific Computing*, 20(1):270–281, 1998.

[40] Kristen Grauman and Trevor Darrell. The pyramid match kernel: Discriminative classification with sets of image features. In *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, volume 2, pages 1458–1465. IEEE, 2005.

[41] Sudipto Guha, Rajeev Rastogi, and Kyuseok Shim. Rock: A robust clustering algorithm for categorical attributes. *Information systems*, 25(5):345–366, 2000.

[42] Sudipto Guha, Rajeev Rastogi, and Kyuseok Shim. Cure: an efficient clustering algorithm for large databases. *Information systems*, 26(1):35–58, 2001.

[43] Sudipto Guha, Rajeev Rostogi, and Kyuseok Shim. Cure: an efficient clustering algorithm for categorical attributes. In *Proc. Of 1998 ACM-SIGMOD Int. Conf. on Management of Data*, 1998.

[44] Jiawei Han, Jian Pei, and Micheline Kamber. *Data mining: concepts and techniques*. Elsevier, 2011.

[45] Katherine A Heller and Zoubin Ghahramani. Bayesian hierarchical clustering. In *Proceedings of the 22nd international conference on Machine learning*, pages 297–304. ACM, 2005.

[46] John R Hershey and Peder A Olsen. Approximating the kullback leibler divergence between gaussian mixture models. In *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*, volume 4, pages IV–317. IEEE, 2007.

[47] Michael Ed Hohn. Binary coefficients: A theoretical and empirical study. *Journal of the International Association for Mathematical Geology*, 8(2):137–150, 1976.

[48] Zdenek Hubalek. Coefficients of association and similarity, based on binary (presence-absence) data: an evaluation. *Biological Reviews*, 57(4):669–689, 1982.

[49] P Jaccard. Comparative study of the floral distribution in a portion of the alps and jura. *The Company Vaudoise Bulletin of Natural Sciences*, 37(5):547–579, 1901.

[50] Anil K Jain. Data clustering: 50 years beyond k-means. *Pattern recognition letters*, 31(8):651–666, 2010.

[51] Sylvie Jeannin and Miroslaw Bober. Description of core experiments for mpeg-7 motion/shape. *MPEG-7, ISO/IEC/JTC1/SC29/WG11/MPEG99 N*, 2690, 1999.

[52] Tony Jebara. Images as bags of pixels. In *ICCV*, pages 265–272, 2003.

[53] Tony Jebara and Risi Kondor. Bhattacharyya and expected likelihood kernels. In *Learning theory and kernel machines*, pages 57–71. Springer, 2003.

[54] Thomas Kailath. The divergence and bhattacharyya distance measures in signal selection. *IEEE transactions on communication technology*, 15(1):52–60, 1967.

[55] George Karypis, Eui-Hong Han, and Vipin Kumar. Chameleon: Hierarchical clustering using dynamic modeling. *Computer*, 32(8):68–75, 1999.

[56] S Sathiya Keerthi and Chih-Jen Lin. Asymptotic behaviors of support vector machines with gaussian kernel. *Neural computation*, 15(7):1667–1689, 2003.

[57] Benjamin King. Step-wise clustering procedures. *Journal of the American Statistical Association*, 62(317):86–101, 1967.

[58] Quoc Le and Tomas Mikolov. Distributed representations of sentences and documents. In *International Conference on Machine Learning*, pages 1188–1196, 2014.

[59] Tao Li. A unified view on clustering binary data. *Machine Learning*, 62(3):199–215, 2006.

[60] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[61] Yunqian Ma and Guodong Guo. *Support vector machines applications*. Springer, 2014.

[62] Brian Mak and Etienne Barnard. Phone clustering using the bhattacharyya distance. In *Fourth International Conference on Spoken Language Processing*, 1996.

[63] Mohamed Al Mashrgy, Nizar Bouguila, and Khalid Daoudi. A robust approach for multivariate binary vectors clustering and feature selection. In *Neural Information Processing - 18th International Conference, ICONIP 2011, Shanghai, China, November 13-17, 2011, Proceedings, Part II*, pages 125–132, 2011.

[64] Andrew McCallum, Kamal Nigam, et al. A comparison of event models for naive bayes text classification. In *AAAI-98 workshop on learning for text categorization*, volume 752, pages 41–48. Citeseer, 1998.

[65] Andrew Kachites McCallum. Bow: A toolkit for statistical language modeling, text retrieval, classification and clustering. *http://www. cs. cmu. edu/mccallum/bow/*, 1996.

[66] Geoffrey McLachlan and David Peel. *Finite mixture models*. John Wiley & Sons, 2004.

[67] Ellis L Michael. Marine ecology and the coefficient of association: a plea in behalf of quantitative biology. *Journal of Ecology*, 8(1):54–59, 1920.

[68] Javier M Moguerza, Alberto Muñoz, et al. Support vector machines with applications. *Statistical Science*, 21(3):322–336, 2006.

[69] Carlos Ordonez. Clustering binary data streams with k-means. In *Proceedings of the 8th ACM SIGMOD workshop on Research issues in data mining and knowledge discovery*, pages 12–19. ACM, 2003.

[70] Hae-Sang Park and Chi-Hyuck Jun. A simple and fast algorithm for k-medoids clustering. *Expert systems with applications*, 36(2):3336–3341, 2009.

[71] Rajat Raina, Yirong Shen, Andrew Mccallum, and Andrew Y Ng. Classification with hybrid generative/discriminative models. In *Advances in neural information processing systems*, pages 545–552, 2004.

[72] Brian D Ripley. *Pattern recognition and neural networks*. Cambridge university press, 2007.

[73] Y Dan Rubinstein, Trevor Hastie, et al. Discriminative vs informative learning. In *KDD*, volume 5, pages 49–53, 1997.

[74] Mehreen Saeed and Haroon Babri. Classifiers based on bernoulli mixture models for text mining and handwriting recognition tasks. In *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*, pages 2169–2175. IEEE, 2008.

[75] Mark Sandler. Hierarchical mixture models: a probabilistic analysis. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 580–589. ACM, 2007.

[76] Thomas B Sebastian, Philip N Klein, and Benjamin B Kimia. Recognition of shapes by editing shock graphs. In *ICCV*, volume 1, pages 755–762, 2001.

[77] Armin Shmilovici. Support vector machines. In *Data mining and knowledge discovery handbook*, pages 231–247. Springer, 2009.

[78] Jai Puneet Singh and Nizar Bouguila. Proportional data clustering using k-means algorithm: A comparison of different distances. In *Industrial Technology (ICIT), 2017 IEEE International Conference on*, pages 1048–1052. IEEE, 2017.

[79] Kehar Singh, Dimple Malik, and Naveen Sharma. Evolving limitations in k-means algorithm in data mining and their removal. *International Journal of Computational Engineering & Management*, 12:105–109, 2011.

[80] John R Smith, Shih-Fu Chang, et al. Automated binary texture feature sets for image retrieval. In *icassp*, pages 2239–2242. Citeseer, 1996.

[81] P Sneath and R Sokal. Numerical taxonomy. freeman, london, 1973.

[82] Andreas Stolcke and Stephen Omohundro. Hidden markov model induction by bayesian model merging. In *Advances in neural information processing systems*, pages 11–18, 1993.

[83] Tim Van Erven and Peter Harremos. Rényi divergence and kullback-leibler divergence. *IEEE Transactions on Information Theory*, 60(7):3797–3820, 2014.

[84] Vladimir N Vapnik. The nature of statistical learning theory. Technical report, 1995.

[85] T Velmurugan. Efficiency of k-means and k-medoids algorithms for clustering arbitrary data points. *Int. J. Computer Technology & Applications*, 3(5):1758–1764, 2012.

[86] T Velmurugan and T Santhanam. Computational complexity between k-means and k-medoids clustering algorithms for normal and uniform distributions of data points. *Journal of computer science*, 6(3):363, 2010.

[87] Kiri Wagstaff, Claire Cardie, Seth Rogers, Stefan Schrödl, et al. Constrained k-means clustering with background knowledge. In *ICML*, volume 1, pages 577–584, 2001.

[88] Jianxiong Xiao, James Hays, Krista A Ehinger, Aude Oliva, and Antonio Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *Computer vision and pattern recognition (CVPR), 2010 IEEE conference on*, pages 3485–3492. IEEE, 2010.

[89] Zijiang Yang, Jiandong Wang, and Tongwen Chen. Detection of correlated alarms based on similarity coefficients of binary data. *IEEE Transactions on Automation science and Engineering*, 10(4):1014–1025, 2013.

[90] Chang Huai You, Kong Aik Lee, and Haizhou Li. Gmm-svm kernel with a bhattacharyya-based distance for speaker recognition. *IEEE Transactions on Audio, Speech, and Language Processing*, 18(6):1300–1312, 2010.

[91] Nuha Zamzami and Nizar Bouguila. Hybrid generative discriminative approaches based on multinomial scaled dirichlet mixture models. *Applied Intelligence*, pages 1–18, 2019.

[92] Nuha Zamzami and Nizar Bouguila. Model selection and application to high-dimensional count data clustering - via finite EDCM mixture models. *Appl. Intell.*, 49(4):1467–1488, 2019.

[93] Nuha Zamzami and Nizar Bouguila. Deriving probabilistic svm kernels from exponential family approximations to multivariate distributions for count data. In *Mixture Models and Applications*, pages 125–153. Springer, 2020.

[94] Ying Zhao and George Karypis. Evaluation of hierarchical clustering algorithms for document datasets. In *Proceedings of the eleventh international conference on Information and knowledge management*, pages 515–524. ACM, 2002.