Office of Graduate Studies

Dissertation / Thesis Approval Form

This form is for use by all doctoral and master's students with a dissertation/thesis requirement. Please print clearly as the library will bind a copy of this form with each copy of the dissertation/thesis. All doctoral dissertations must conform to university format requirements, which is the responsibility of the student and supervising professor. Students should obtain a copy of the Thesis Manual located on the library website.

Dissertation/Thesis Title: Multimodal Information Retrieval and Classification

Author: Kamelia Aryafar

This dissertation/thesis is hereby accepted and approved.

Signatures:

Examining Committee

Chair _____

Members _____

_____

_____

_____

_____

Academic Advisor _____

Department Head _____

Office of Graduate Studies • 3141 Chestnut St. • Randell Hall 240 • Philadelphia, PA 19104
Tel.: 215-895-0366 • **Fax**: 215-895-0495 • **Email**: graduate@drexel.edu • **Web**: www.drexel.edu/provost/graduatestudies

Multimodal Information Retrieval and Classification

A Thesis

Submitted to the Faculty

of

Drexel University

by

Kamelia Aryafar

in partial fulfillment of the

requirements for the degree

of

Doctor of Philosophy in Computer Science

December 2015

Dedications

This thesis is dedicated to my parents

whose love and support made this work possible.

Acknowledgments

It is my great pleasure to thank all the people who inspired and helped me during my years as a graduate student. My foremost gratitude goes to my fantastic adviser, Ali Shokoufandeh, for his help, inspiration, and guidance during this journey. Dr.Shokoufandeh is not only an amazing advisor, but he is also a true mentor, and most certainly the most influential person in my life, after my parents. I am extremely grateful to him for his patient guidance during the course of my Ph.D. program, teaching me how to do constructive research and think critically. Dr.Shokoufandeh taught me to *pay it forward* and I'm extremely thankful to him for this.

I owe my deepest gratitude to my committee members: Dario Salvucci, Jeremy Johnson, Santiago Ontanon and Xiaohua Tony Hu, for their time and continuous input on my ideas. I would also like to extend my thanks to my collaborator in Etsy for encouraging me along the way, notably Josh Attenberg, Diane Hu, Corey Lynch and Robert Hall. Josh has been a great mentor to me, inspiring me to do research and provided me with freedom to try new ideas everyday. I'm grateful for his support, inspiration and feedback on my document and presentations and stimulating conversations on various research topics.

I am deeply indebted to my family and friends who supported and encouraged my decision to work on my Ph.D. even when that involved travelling to another country. I would especially like to thank my mother and father, Afsaneh and Majid, who spent many years educating and inspiring me and sacrificed so much for me to able to accomplish my dreams. I would like to thank my brother and sister, Ehsan and Parisa, for always supporting me. I'd also like to thank the many friends I've made at Drexel, including Linge Bai, Yuan Duan, Dmitriy Bespalov, Tom Plick, Ehsan Khosroshahi, Walt Mankowski, Evan Sultanik, Yusuf Osmanlioglu, Lou Kratz, Aylin Caliskan-Islam, Sadia Afroz, Laura Beck, Tuyet Sithiphavong, Brenna Martin and Evy Vega.

Table of Contents

List of Tables . . . vii

List of Figures . . . viii

List of algorithms . . . x

Abstract . . . xi

1. Introduction . . . 1

1.1 Information Retrieval . . . 3

1.2 Classification . . . 5

1.3 Thesis Overview . . . 9

1.3.1 Unimodal Approaches . . . 9

1.3.2 Cross-Modal Methods . . . 9

1.3.3 Multimodal Models . . . 10

2. Classification Methods . . . 11

2.1 Background . . . 12

2.2 An Overview of Feature Representations . . . 16

2.3 An Overview of SVM . . . 20

2.4 Conclusions . . . 24

3. Explicit Semantic Analysis . . . 25

3.1 Background . . . 26

3.2 Vector Space Model . . . 28

3.3 The tf-idf Model . . . 31

3.4 Explicit semantic analysis . . . 32

3.5 ESA Feature Encoding . . . 34

3.6 Empirical Evaluation . . . 37

3.7 Conclusions . . . 41

4. Sparsity-Eager Support Vector Machines 43
4.1 Background 43
4.2 Deep Neural Networks 44
4.2.1 Manifold Learning 44
4.2.2 DNN Architecture 45
4.3 ℓ_1-SVM 47
4.3.1 Imposing Sparsity 49
4.4 Empirical Evaluation 52
4.4.1 Dataset 52
4.4.2 Data Preprocessing 53
4.4.3 Results 55
4.5 Conclusions 60
5. Cross-Modal Information Retrieval 62
5.1 Background 63
5.2 Cross-Modal Methods 66
5.3 Cross-Modal CCA 67
5.4 ESA Retrieval 69
5.5 Empirical Evaluation 71
5.5.1 Dataset 71
5.5.2 Results 72
5.6 Conclusions 75
6. Multimodal Approaches to Classification 76
6.1 Background 78
6.2 Multimodal ℓ_1-SVM 82
6.3 Multimodal SLIM 85
6.4 Challenges 86
6.5 Empirical Evaluation 88

6.5.1 Genre Classification 88
6.5.2 Artist Identification 90
6.6 Conclusions 92
7. Concluding Remarks 94
7.1 Summary and Conclusions 95
7.1.1 ESA Feature Encoding 95
7.1.2 ℓ_1-SVM 96
7.1.3 Cross-Modal Retrieval 97
7.1.4 Multimodal Classification 99
7.2 Future Work 100
Bibliography 101
Vita 115

List of Tables

2.1 Selected audio features and dimensionality of the feature space is shown above. The short-time audio features are represented using the dimensionality of the mean feature vector. 18

3.1 Number of songs per genre. 38

3.2 Genre classification accuracy on the benchmark dataset: aggregation of MFCC features(AM) [127] and Temporal, spectral and phase (TSPS) features [126] are compared to ESA representation of MFCC features using three different supervised methods: k-NN, ℓ_2-SVM, and random classifier as a baseline. We considered two variations of ESA-model one based using 1000 and 5000 code-words. 40

4.1 Number of songs per genre. 52

4.2 Classification accuracy (%) on sample dataset is compared across various methods. . . . 55

5.1 **Lyrics retrieval mean average precision for the Million Song Dataset (MSD) are compared using variations of CCA and ESA representation method of audio MFCC features and their corresponding lyrics. The subsets are correlated with textual tf-idf in the canonical subspaces where the retrieval is performed to obtain lyrics metadata associated with each concept. CCA on quantized feature vectors are compared to modified ESA representation for music information retrieval.** 73

6.1 Classification accuracy (%) on sample dataset is compared across various methods on the MSD dataset using multimodal and single modality data. 89

6.2 Artist Identification accuracy (%) on sample dataset is compared across various methods on the MSD dataset using mutimodal and single modality data. Top k nearest neighbors is fixed as the classification algorithm using single modality ESA (audio only) and multimodal ESA fusion (audio and lyrics). 92

List of Figures

1.1 A term-document indices matrix is illustrated. Matrix element (t_i, d_j) is 1 if the j^{th} dissertation d_j contains the term t_i. 4

1.2 Examples of RGB fundus images for diabetic retinopathy eye conditions is illustrated. 6

1.3 Input images from a sample dataset in five labeled classes: (a) Normal , (b) Mild, (c) Moderate , (d) Severe and (e) Proliferative diabetic retinopathy are shown. 7

1.4 Examples of preprocessing and feature representation on fundus images dataset for two classes: (a) Original image , (b) Green channel (Red-free) and (c) Green channel histogram equalization for an example of class 3 (severe condition) are illustrated. (d) Original image , (e) Green channel (Red-free) and (f) Green channel histogram equalization for an example of class 4 (proliferative diabetic retinopathy) are illustrated. 8

2.1 A sample song is represented: (a) time domain (b) frequency spectrum, (c) MFCCs and (d) Chroma features. 19

3.1 A query audio from the test set is illustrated. 38

3.2 True positive genre classification rate is illustrated for each music genre. temporal, spectral and phase (TSPS) features [126] are compared to ESA representation of MFCC features using k-NN learning schema. 41

3.3 Classification accuracy results are demonstrated using varying number of code-book elements, k. In all classification experiments SVM classifier is used as the learning schema and the only variable is the number of clusters used to form the audio code-book. 42

4.1 The Rectified Linear units as the activation function is illustrated. 45

4.2 The music classification method is illustrated. 53

4.3 A sample audio input from class I, II and III(*alternative, pop and rock*) genre is illustrated as (a,d,g) original input signal in the frequency domain by magnitude and phase, (b,e,h) frequency spectrum and (c,f,i) power spectrogram. 56

4.4 A sample audio input from *blues, electronic, folk/country, jazz, rap/hiphop* genre is illustrated as (left) original input signal in the frequency domain by magnitude and phase, (middle) frequency spectrum and (right) power spectrogram. 57

4.5 True positive genre classification rate is illustrated for each music genre. 58

4.6 ℓ_1-regression [7], logistic regression [8], ℓ_2-SVM optimization [9] and ℓ_1-SVM classification accuracy rates are illustrated using different number of training samples. The number of training samples have been limited to 20 samples to reduce the convergence time of the classifier. There is no deviation in the classification accuracy rate for a larger training set. 59

4.7 Average classification accuracy rate is reported for different average training time. . . . 60

5.1 The observed variables in the audio dataset are represented using **MFCC** feature vectors. These feature vectors are quantized and represented in the shared concept (shared by textual metadata formed from tf-idf encoding of song lyrics). Canonical correlation analysis is then used to project the audio and text representation onto canonical variates P_A and P_T respectively, where the projections are maximally correlated. 71

5.2 A query audio set Q is first represented using its short term windowed MFCC feature vectors. These feature vectors are then quantized and interpreted in terms of the audio words in $\mathcal{D}$. The quantized feature vectors are then projected onto canonical variates P_A and P_T where the corresponding textual information is retrieved as $\mathcal{T}$ 72

6.1 The steps involved in multimodal fusion of signals from different modalities are illustrated. 79

6.2 Multimodal fusion at four different levels are illustrated: fusion at data level, feature level, score level and decision level. 80

6.3 Audio signals and lyrics data are projected into ESA space before estimating a fusion model for artist identification on the MSD dataset. 89

6.4 Classification accuracy rate is reported for different classification algorithms using different average trainig time on the MSD dataset using multimodal and single modality data. 90

6.5 Audio signals and lyrics are used to create a fusion classifier for music artist identification. Audio signals and lyrics are represented as ESA matrices $\mathcal{E}(A)$ and $\mathcal{E}(T)$, respectively. A fusion classifier matrix X in combination with k-NN is then utilized for artist identification. 91

List of Algorithms

1 The retrieval operation in a Boolean standard model of data retrieval. 29
2 Construction of the ESA vector of an audio sequence. 37

3 The retrieval operation in a manifold learning model of cross-modal retrieval. 66
4 Canonical correlation retrieval based on ESA representations of both modalities for input queries Q. 70

Abstract

Multimodal Information Retrieval and Classification

Kamelia Aryafar

Advisor: Ali Shokoufandeh, Ph.D.

This thesis deals with classification and retrieval models from two different perspectives: (1) the single modality classification optimizations where only one data channel can be used; (2) the multi-modal classification methods where more than one data channel is available;

A classification system is composed of two main steps: extraction of meaningful features to represent the dataset in a feature space and the classification optimization. The first contribution of this thesis is based on sparse approximation techniques introduced by Donoho, namely the ℓ_1-regression. We introduce a sparsity-eager support vector machine optimization that combines the ideas behind ℓ_1-regression and SVM to boost the classification performance. We show that the optimization of sparsity-eager SVM can be relaxed and formulated as a linear program. This linear program is then solved by fast gradient descent techniques, yielding an optimal set of classifier coefficients. We compare the performance of this classifier with state-of-the-art deep neural networks and baseline models on various public datasets.

The second contribution of this thesis is a vector space model of feature vectors to boost the classification performance. This representation is similar to the explicit semantic analysis modeling of text documents introduced by Evgeniy Gabrilovich and Shaul Markovitch. In essence, the explicit semantic analysis representation is an extension of term frequency inverse document frequency modeling to multi-dimensional feature vectors. Through a set of experiments, we show that this representation can boost the classification accuracy. The explicit semantic analysis can also provide an efficient cross-domain information retrieval framework. We combine this representation with a canonical correlation optimization to achieve this.

The third contribution of this thesis is based on multimodal approaches to classification: sparse linear integration model and ℓ_1-SVM. We extend the sparsity-eager support vector machine optimization to deal with more than one data modality. Again we formulate this optimization as a linear combination of the training samples in multimodal settings. We then relax this optimization by replacing the ℓ_0-norm. The final optimization is solvable using convex optimization methods. We show that combining all available data channels can boost the classification accuracy of the sparsity-eager support vector machine classifier in comparison with baseline classifiers.

Chapter 1: Introduction

This dissertation explores the problem of information retrieval (IR) and classification from different perspectives: (i) single modality classification and feature representation, (ii) cross-modal retrieval models and (iii) multimodal classification techniques. In the context of machine learning, a *modality* is the classification of a single independent channel of sensory input/output examples. A machine learning system is designated *unimodal* if it has only one modality implemented, and *multimodal* if it has more than one. Cross-modal retrieval models are systems that correlate a query from one modality to an output from another modality. When multiple modalities are available for a machine learning task, the system is said to have overlapping modalities. Multiple modalities can be used in combination to provide complementary data that may be redundant but convey information more effectively. Our main goal in this thesis is to advance the single modality learning methods and introduce an extension of them to multimodal settings.

IR is the activity of obtaining data relevant to an information need or a *query* from a collection of information resources. IR is the foundation of organization and access to information items. Perhaps the most well-known example of an IR system is a web search engine. A web search engine is an IR system that is designed to retrieve the most relevant information from the World Wide Web. The retrieved information can be a mix of web pages, images, text documents, etc. An IR system is unimodal when it retrieves information in the same space as the query and multimodal when the query and retrieved information belong to different modalities. Cross-modal IR systems are a subset of multimodal approaches that require the query and information sources to be of non-overlapping modalities.

Most sensory devices such as cell phones are now equipped with more than one data acquisition channel. Cameras, audio recorders, GPS sensors, radars, LiDARs, etc. are easily accessible to the public. This accessibility results in huge collections of multimedia data available in different contexts like text, image, audio and video. Multimodal IR models are required for representation, storage,

organization of, and access to these multimodal information items.

An emerging challenge for large scale IR systems is to analyze unstructured data produced by various sensors or *classification*. The goal of a classification system is to uncover hidden patterns in the dataset (*cluster analysis*) or to identify to which of a set of categories (sub-populations) a new observation belongs (*statistical classification*). Cluster analysis is considered an unsupervised learning while classification is an instance of a supervised learning problem. Classification is often a fundamental first step in numerous learning systems that deal with large scale datasets. An example of a classification in real world can be the categorization of user accounts on numerous websites as valid or fraudulent accounts. In this sense the classifier models the training user accounts data in a feature space and then *learns* a categorization of data into two classes, valid and fraudulent. This learnt function can then be applied to new user accounts in production to identify fraudulent accounts.

Similar to IR methods, classification techniques can also be extended to multimodal settings. The input to a classifier can be a combination of complementary sensory information. The provision of multiple modalities in classification systems is often motivated by a performance boost while using the complementary information in other data channels, reduction of noise sensitivity in one channel and, the universality of the classification method. Consider our previous example to classify valid and fraudulent user accounts. The user account in unimodal classification can be a text-based representation of the user where it includes tokens from user name, location, account age and other factors. A multimodal extension of the classification can be employed if we consider a user's profile image. In this sense, a fraudulent user that registers new accounts using valid text-based information but reuses a profile image (or stock photos) can now be categorized as fraudulent. The image data in each user profile is providing complementary information that was previously unrecognized in text only classifier.

The work presented in this dissertation considers both the IR and classification problems in unimodal and multimodal settings. The unimodal classification is tackled by introducing a novel feature embedding in a latent space and a novel classification method. The feature embedding

is then used to correlate queries in one modality to information sources in other modalities, i.e., cross-modal IR. The classifiers are extended to multimodal settings where we take advantage of all available data sources in training to boost the classification performance. Our methods can be used as an important complement to existing classification techniques to deal with overfitting.

1.1 Information Retrieval

The problems considered in this dissertation are a subset of IR methods. An IR method has three basic components: a user query or information request which describes what information we are looking for, a repository or database of information sources and, a retrieval algorithm which assigns a score [1] which describes the degree of relevance between the query and each object in the database. We discuss the details of IR methods and in specific *vector space* models in Chapter 3, but in this section we present a few examples to illustrate the inner workings of unimodal and multimodal IR systems.

To begin, consider a unimodal retrieval model (text-based) where we have a repository of all Drexel university's computer science dissertations. Suppose we are interested to determine which dissertations contain the words "classification" and "retrieval" and not "multimodal". A simple retrieval model can be one that scans through all the dissertations text line by line and matches the query pattern against the database. A simple *grep* can do this efficiently with modern computers for simple queries on modest collections. Two main questions remain with this approach: *What happens when we scale the dataset to millions of terabytes? How can we define a ranking function that decides which documents are more important for our issued query?* We need more flexible matching operations and ranked retrieval.

The way to avoid linearly scanning the texts for each query is to index the documents in advance. Here we consider an illustration of the *Boolean retrieval model* as a scalable alternate to scanning. Later in Chapter 3 we introduce this model in more details. The Boolean retrieval model creates a binary term-document incidence matrix for each dissertation in our repository, as illustrated in the matrix of Figure 1.1. Terms are the indexed units here which are often words, but can be any

[1] In most IR systems the relevance score is a numerical value.

	classification	*retrieval*	*unimodal*	*unicorn*	*multimodal*	...
*dissertation*1	1	0	0	1	0	
*dissertation*2	0	0	0	1	0	
*dissertation*3	1	0	0	0	1	
*dissertation*4	1	1	1	0	1	
*dissertation*5	1	1	0	1	0	
⋮						

Figure 1.1: A term-document indices matrix is illustrated. Matrix element (t_i, d_j) is 1 if the j^{th} dissertation d_j contains the term t_i.

token that characterizes a unit of information. The matrix rows and columns can be represented as document or term vectors. Each row describes a dissertation by the terms that appear in it and each column represents the term distribution among all documents.

To find the answer to our original query q containing the words "classification" and "retrieval" and not "multimodal" we can form the term vectors for "classification", "retrieval" and complement of "multimodal" and perform a bitwise AND operation as:

$$q \;=\; 10111 \quad \text{AND} \quad 00011 \quad \text{AND} \quad 11001 \;=\; 00001 \tag{1.1}$$

Equation 1.1 shows that the dissertation d_5 (which has a 1 at index 5) is the only document in our database that matches our desired query. A crucial observation to ensure the scalability of this model is the inherent sparsity in the term-document matrix as the number of documents and the number of terms increase. In a large corpus a substantial set of documents do not contain a large subset of the terms and this results in *zero* elements in the term-document matrix. A functional alternative to indexing is an *inverted indexing* schema that stores the index of all documents that contain a term and skips the rest of the documents. The inverted index schema in combination with the Boolean retrieval model presents a scalable solution for IR systems. It is however limiting to always assume a boolean expression of all queries. Moreover, this model performs an exact match between the query and the document terms and equally weights the terms. In Chapter 3 we review the vector space model as a suitable alternative to this method.

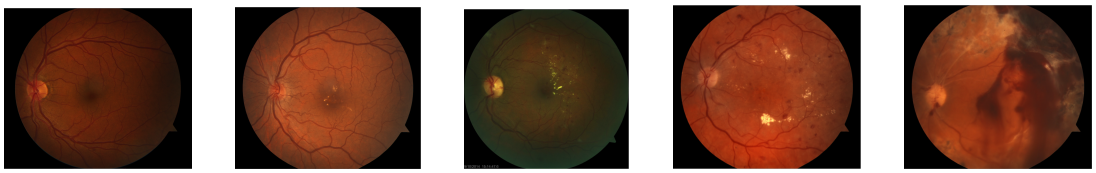
The above example is a unimodal retrieval task, that is both the query and the documents in the database belong in the same modality. The question remains whether the single modality IR methods can be extended to cross-modal and multimodal settings. The cross-modal retrieval considers an extension of unimodal IR methods where the query and documents each belong to a different data modality. Consider our previous example with a slight modification. This time the query stays the same, but each dissertation document is replaced by the *audio recording* of the defense oral presentation. Our goal is once again to retrieve all the recordings that contain "classification" and "retrieval" and not "multimodal". The query in this sense belongs to a text modality (or a Boolean expression of terms) and the database objects are in the audio format. Cross-modal retrieval models can be designated to present both query and database objects in a latent space and allow for retrieval between the two channels. In Chapter 5 we introduce a cross-modal retrieval framework that takes advantage of the latent vector space.

A final aspect of IR in multimodal settings is the combination of all data channels as a single identity for retrieval between multimodal datasets. Our goal here is to close the gap between all different data modalities. The query in the previous example can be a multimodal query that contains both audio recordings and dissertations where we are interested in finding all relevant dissertations and talks that are relevant to this query. In Chapter 6 we consider multimodal classification as an example of this latter problem in more details.

1.2 Classification

The problem of classification is closely related to IR methods. Classification is an emerging challenge for large scale information systems and is used to analyze unstructured data produced by various sources. In this thesis we explore single modality and multimodal classification methods. In Chapter 2 we explore the details of different classification methods in terms of the feature representation and learning models. In this section we present a few examples of classification in unimodal and multimodal settings to emphasize the challenging aspects of this problem.

To begin, consider the previous example of Section 1.1 and its formulation as a classification problem. This time we are interested in categorizing the dissertations by research areas such as



(a) **(b)** **(c)** **(d)** **(e)**

Figure 1.2: Examples of RGB fundus images for diabetic retinopathy eye conditions is illustrated.

machine learning, *computer vision*, *graph theory*, *human-computer interaction* and so on. Each research area is considered a class and we are provided with a set of labeled dissertations as the training documents. Our goal is to classify the new dissertations as belonging to a research area. This example is an instance of a multi-class classification. Most classification methods are described in binary settings and are later extended to multi-class problems by using the output coding techniques such as the models proposed by[1;2].

A classification model is concerned with two main components: (i) feature representation and (ii) the choice of a learning model or classifier. Feature representation is the preprocessing of sensory data and the representation of this data in an informative feature space. In Chapter 2 we dive deeper into the inner workings of both components in state-of-the-art classification methods. We introduce an extension of the vector space model for time-series signals in a latent space as an alternate feature representation for classification tasks in Chapter 3. We later introduce a new classifier as well in Chapter 4.

The choice of a proper preprocessing and feature selection can often considerably impact the performance of the classification model. In our previous example the feature space can be the document vector of each dissertation. An excellent example of the impact of a feature space on classification performance is captured when dealing with special data formats such as RGB fundus images of the eyes. Figure 1.2 illustrates an example of such images that is used for diabetic retinopathy detection. These images are often used by medical professionals to detect stages of diabetic retinopathy as illustrated in Figure 1.3. The automatic detection of this eye condition can be formulated as a classification problem. In this sense our goal is to label each instance of

(a) (b) (c) (d) (e)

Figure 1.3: Input images from a sample dataset in five labeled classes: (a) Normal , (b) Mild, (c) Moderate , (d) Severe and (e) Proliferative diabetic retinopathy are shown.

unseen fundus images with a disease stage class label as shown in Figure 1.3. The fundus eye images are often preprocessed as a green channel (red-free) where the abnormalities are best exaggerated. The green channel is then presented as an equalized channel histogram which is the input feature representation of the input fundus images. The process of preprocessing and feature extraction is illustrated in Figure 1.4 for this problem.

The second main component of a classification method is the choice of a learning model or classifier. A typical classifier utilizes a majority of training examples in the classification decision for an unseen (test) example. This could potentially introduce *overfitting* in classification, that is when we model the noise in data channels rather than the actual underlying relationship between observations and labels. Principles of learning theory [3] suggest that simpler models in which only a small, well-chosen subset of training examples is involved in making the final classification decisions is statistically more preferable. This motivates a sparse approximation framework, which is the essence of our proposed classifier in Chapter 4.

Similar to IR models, methods for classification can be divided into single modality and multimodal approaches. Single modality classification methods consider the problem of classification in the presence of one information channel such as audio signal, text and images. Both examples introduced in this section are instances of *unimodal* classification. Single modality architectures are only useful when rich datasets are available for a data modality and suffer from non-universality and single channel dependency. In the absence of one modality or presence of a noisy channel, a *multimodal* fusion model becomes a necessity. As a result, there has been an increasing emphasis on the use of all available data modalities for both characterization and retrieval of unstructured data.

A multimodal classifier deals with datasets that can be presented using more than one channel,

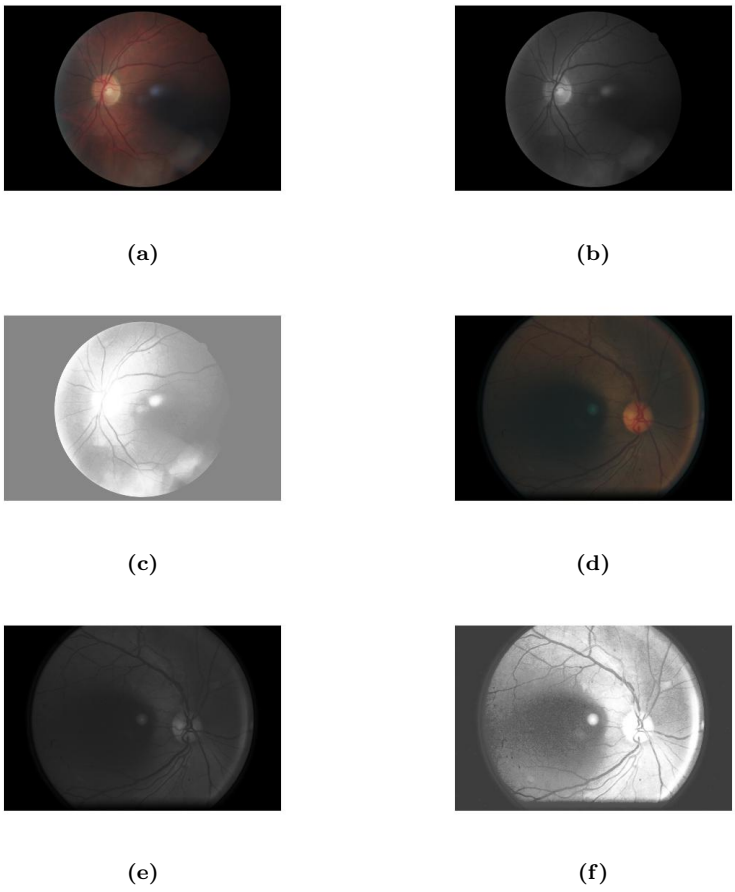


Figure 1.4: Examples of preprocessing and feature representation on fundus images dataset for two classes: (a) Original image , (b) Green channel (Red-free) and (c) Green channel histogram equalization for an example of class 3 (severe condition) are illustrated. (d) Original image , (e) Green channel (Red-free) and (f) Green channel histogram equalization for an example of class 4 (proliferative diabetic retinopathy) are illustrated.

fully or partially. Imagine our previous example when the goal is still the classification of stages of diabetic retinopathy, but the dataset is composed of both fundus images and the text description provided by a medical professional. The classifier can benefit from both modalities to predict more accurate class labels. Moreover, the classifier can make predictions based on images or text descriptions only and is more resilient to loss of information in any of the two information channels. This example highlights the foundations of multimodal classification. In Chapter 6 we explore state-of-the-art multimodal classifiers and introduce an extension of our models to multimodal settings.

1.3 Thesis Overview

The contributions of this dissertation include the following extensions to unimodal, multimodal and cross-modal IR and classification methods. First, we introduce an extension of the vector space models to time-series signals such as audio sequences. Second, we use this semantic representation of time-series signals in combination with canonical correlation analysis for cross-modal retrieval . Third, we introduce the sparsity-eager support vector machines to combine the ideas behind support vector machines with sparse approximation techniques. Fourth, we introduce extensions of this classifier to multimodal settings where the dataset is presented using more than one data channel. The following sections outline these four components.

1.3.1 Unimodal Approaches

Our first set of contributions are related to single modality classification and retrieval. We examine the classification problem from two different perspectives, the feature space and the learning methods. We impose the following questions: (i) can feature embeddings based on explicit semantic analysis (ESA) boost classification accuracy? (ii) how can we enhance the classic support vector machines classifier to improve accuracy while dealing with overfitting on training examples? (iii) how do the hand-crafted features and developed classification techniques compare with state-of-the-art neural networks in terms of classification accuracy on public datasets?

Our contributions in single modality models are presented in chapters 3 and 4. First in chapter 3 we introduce a feature encoding schema, namely the ESA as an extension of the vector space model to time-series signals. Next in chapter 4, we introduce our sparsity-eager SVM to address the inherent shortcomings of classic SVM. In both chapters our hypotheses are verified through an extensive set of experiments.

1.3.2 Cross-Modal Methods

The second set of contributions of this thesis are focused on cross-modal retrieval models. Our motivation here is to extend the existing retrieval methods to deal with multimodal datasets. Specifically, we are interested in creating a shared semantic space between different data modalities that allow

for retrieving information from a database for a query that belongs in a different modality. Our key hypothesis here is that the ESA embeddings can be used in combination with canonical correlation analysis (CCA) methods to allow for cross-modal retrieval. In chapter 5 we provide a framework that can achieve this retrieval in an efficient manner. We evaluate our hypothesis through a set of retrieval experiments and compare the results with baseline retrieval models.

1.3.3 Multimodal Models

The third set of contributions of this thesis are concentrated on multimodal classification and retrieval. These contributions are described in chapter 6 in details. Our contributions are motivated by hidden information in complementary data channels that can boost the accuracy of classification methods. Our hypothesis is that a multimodal classifier which can take advantage of available data modalities can boost the performance of classification on multimodal datasets. To this end, we evaluate the classification accuracy of two novel classification methods and baseline unimodal classifiers on public datasets.

Chapter 2: Classification Methods

Data classification or statistical classification is a fundamental task in pattern recognition and machine learning[4;5]. Classification is often described as the problem of organizing unstructured data into similar groups based on various similarity metrics or assigning labels to unobserved data based on training examples. With the recent information and data explosion[6], the world's technological capacity to store information grew from 2.6 (optimally compressed) exabytes in 1986 to 15.8 in 1993, over 54.5 in 2000, and to 295 (optimally compressed) exabytes in 2007[6]. This rapid increase in the amount of published information or data and the effects of this abundance call for machine learning models that are able to analyze and structure new observations based on previous examples or are able to identify patterns within a set of sensory data.

The machine learning community often refers to the classification problem as a supervised learning model where a set of labeled training data is used to assign class labels to new unseen test examples. In this sense, the datasets for supervised classification are often composed of the training (labeled) dataset and the test (unlabeled) dataset[5]. Various classification algorithms (classifiers) have been proposed in the literature of pattern recognition and machine learning community[4]. The ℓ_1-regression[7], logistic regression[8], support vector machines (SVM)[9], k-nearest neighbor classifiers[10] and various neural networks[11] are among the most popular classification techniques and are used in many different applications such as computer vision[12], medical diagnosis[13], speech recognition[14], music classification[15], hand-writing detection[16] and text classification[17].

In the absence of the training set or a set of labeled examples, the categorization problem becomes an unsupervised learning problem where the task at hand is to divide the population into sub-populations that share the same characteristics[18]. The unsupervised learning problem is also known as the cluster analysis or clustering task[19]. Various clustering algorithms have been studied in the literature[19] and are widely used in different applications such as segmentation of brain MRI scans[20], image segmentation[21] and object retrieval[22].

In this chapter, we review the main components of a classification task: (i) the feature representation and, (ii) the learning models. This chapter is organized as follows; in Section 2.1, we explore the various classification techniques and their applications in machine learning and pattern recognition. In Section 2.2 we explore various feature representations for classification. In Section 2.3 we examine the theoretical foundations of support vector machines (SVM) classifiers. Later in Chapter 4, we introduce the ℓ_1-SVM classifier to address the inherent issues of classic SVM. Finally, we conclude this Chapter in Section 2.4.

2.1 Background

Classification and clustering are examples of the pattern recognition problem in which we assign an output value to an input value. In machine learning and statistics, classification is the problem of identifying to which of a set of categories a new test example belongs, on the basis of a training set of data containing labeled instances. An example of a classification process is labeling input email messages as *spam* or *non-spam* [23].

The classification problem can often be separated into two problems: binary and multi-class classification. In binary classification, only two classes are involved, whereas multi-class classification involves assigning class labels to k classes where $k \geq 3$. In a binary classification settings, every single instance x has a corresponding label $y \in \{-1, 1\}$ indicating whether it belongs to a specific class or not. Multi-class classification can be achieved by combining multiple binary classifiers or output coding techniques [1;2;24]. In this thesis, we use the binary classification as a proof of concept and extend the classifiers to multi-class classifiers using the output coding.

Statistical classification was initially introduced by Fisher *et. al* [25;26] in the context of binary classification which led to the introduction of Fisher's linear discriminant function [27] and later the linear discriminant analysis (LDA) [28]. This model assumes that each data and labels within each class has a multivariate normal distribution. The extension of this classifier to multi-class classification has also been considered with a linear [27;29] and non-linear classification rules [30].

A typical classifier is composed of two main steps: the representation of input examples in a feature space and the selection of the classification algorithm. In a classification settings, individual

examples are often represented via a set of feature vectors which are measurable properties of each example and vary depending on the input modality. In Section 2.2 we examine various feature representations for different data modalities such as text, audio and images. The vector space associated with these vectors is often called the feature space and is often of high dimensionality. Various dimensionality reduction techniques have been employed in the literature to boost the classification performance in high-dimensional feature spaces. In Chapter 4 we examine the dimensionality reduction techniques that are used in our experiments.

The second step in the data classification problem is the selection of a suitable classification algorithm. A large set of classification algorithms are based on a linear function that assigns a score to each possible class, k. These classifiers assign the classification score by computing the dot product between a feature vector and a weight vector for each instance. The linear classifier then sorts the scores for each instance and assigns the class label with the highest score to unseen examples. The linear score function can be represented as:

Definition 2.1.1. Linear score function.

$$\text{score}(\mathbf{X}_i, k) = \boldsymbol{\beta}_k \cdot \mathbf{X}_i,$$

where $\mathbf{X}_i$ is the feature vector representation of example i and $\boldsymbol{\beta}_k$ is the vector of weights corresponding to class k for input example i.

The $\text{score}(\mathbf{X}_i, k)$ is then the score associated with assigning input example i to class k. This set of classification models are often known as the linear classifiers [31] and are often distinguished by the training algorithm that produces the optimal weights or coefficients vector for each class. Logistic regression [8], Probit regression [32], the perceptron algorithm [33], support vector machines (SVM) [9] and linear discriminant analysis [28] are among the most popular linear classification methods. Most linear classifiers can be converted into non-linear algorithms by applying a *kernel trick* [34;35] where the input is projected into a higher dimensional feature space to find non-linear correlations.

Intuitively kernel methods can be explained as learning the correlation between the training

example $\mathbf{x}_i$ with label y_i and the corresponding weight, w_i instead of the fixed parameters corresponding to each input example. In this sense the class labels for unseen or test examples, $\hat{y}$, are predicted by the application of a similarity function or *kernel function* between the unlabeled input $\mathbf{x}'$ and each of the training inputs $\mathbf{x}_i$. For instance, a kerneled binary classifier typically computes a weighted sum of similarities as:

$$\hat{y} = \operatorname{sign}\left(\sum_{i=1}^{n} w_i y_i k(\mathbf{x}_i, \mathbf{x}')\right), \tag{2.1}$$

where $\hat{y} \in \{-1, +1\}$ is the kerneled binary classifier's predicted label for the unlabeled input $\mathbf{x}'$ whose hidden true label y is being predicted. In this sense $k\colon \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}$ is the kernel function that measures similarity between any pair of inputs $\mathbf{x}, \mathbf{x}' \in \mathcal{X}$. The $w_i \in \mathbb{R}$ are then the weights for the training examples, as determined by the learning algorithm and the sign function sign() determines whether the predicted classification $\hat{y}$ comes out positive or negative. Kernel methods have been a powerful method in machine learning and have been widely used since the introduction of SVMs when the SVM was found to be competitive with neural networks on tasks such as handwriting recognition[33–36].

Data classification is an experimental problem. Both linear and non-linear classifiers are often compared through a set of empirical tests to compare classifier performance and to find the characteristics of data that determine a suitable classification model. Classifier performance depends greatly on the characteristics of the data to be classified. Specific classifiers can outperform other classifiers in special problems and through different measures. Different classification metrics can also be measured to find a suitable classifier for the task in hand[37].

The classification performance, convergence time, robustness, simplicity, interpretability, scalability and domain-dependent quality indicators are often the most important evaluation metrics for a classifier[37;38]. Predictive or classification accuracy refers to the ability of the model to correctly predict the class label of new or previously unseen examples. The predictive classification accuracy can be estimated by estimating the loss function, analysis of the confusion matrix and cost function analysis[38]. Convergence time or the classifier speed is the computational costs involved in

model generation and application. Robustness is the ability of the classifier to make correct predictions given noisy or sparse data. Scalability refers to the ability to construct the model efficiently given large amount of data. Interpretability defines the level of understanding and insight that is provided by the model. Finally the domain-dependence indicates the prime modality utilized for classification[37].

The classification performance is often measured through misclassification cost, lift, Brier score[39], information score, margin class probabilities and receiver operating characteristic (ROC) curves. Precision and recall and F-measure[40] are also widely used to measure a classifiers performance. Regression algorithms are often measured through mean squared error and mean absolute error[41]. The classification evaluation approaches can often be characterized as theoretical evaluations and experimental or empirical evaluations. The prior method addresses general laws that may govern training examples from unseen test data. Sample complexity, computational complexity and mistake bound are examples of theoretical approaches to classifier evaluation. The experimental and empirical assess the predictiveness of the classifier through a set of experiments.

Experimental estimation of classification accuracy is often achieved by a random data partitioning into train and test data. Hold-out, k-fold cross validation and leave-one-out are widely used as data partitioning techniques[42]. Hold-out utilizes two independent datasets for training and testing the classifier. It should be noted that the test data is not used in creating the classifier in a hold-out partitioning model. A random split for hold-out is used for large scale datasets while repeated hold-out can be used to estimate classification accuracy on medium-sized datasets. In this model, the error rate or classification accuracy on all iterations are averaged to yield an overall accuracy rate. In a k-fold cross validation model the dataset is randomly divided into k subsamples. The $k-1$ subsamples are treated as the training data and one sub-sample as test data. This process is often repeated k times and average classification accuracy rate is reported. Leave-one-out cross validation is a special case of cross-validation that is used on small scale datasets due to computational costs. In this approach the number of folds is equal to the number of training examples. It is important to note that the dataset scale and evaluation metrics decide the best evaluation method for a specific

classification task[43].

2.2 An Overview of Feature Representations

Choosing informative, discriminating and independent features is a crucial step for effective algorithms in pattern recognition, classification and regression. A feature vector is an n-dimensional vector of numerical features that represent some object such as text, images or audio. In Chapter 3 we discuss a set of feature vectors to represent text information, namely the *tf-idf* feature vectors in details. The choice of features in a particular system may be highly dependent on the specific problem at hand. The input to a classifier is a numerical representation of objects, since such representations facilitate processing and statistical analysis.

Feature vectors can be defined from raw sensory data to characterize the individual measurable properties in the signal or can be automatically learned. Feature learning can be divided into supervised and unsupervised feature learning[44]. In supervised feature learning, features are learned with labeled input data[45]. Neural networks[46], perceptron[47], and (supervised) dictionary learning[48] are examples of supervised feature learning. Unsupervised feature learning[44] refers to learning features from unlabeled data. Dictionary learning[49], independent component analysis[50], auto-encoders[51], matrix factorization[52], and various forms of clustering are examples of unsupervised feature learning[44].

Feature vectors can be defined by individual measurable properties of the input signal. Different features have been proposed in the literature to represent various data modalities such as text, audio and images. In this Section, we briefly introduce some of the most popular feature representations for text, audio and images. The tfidf, short for term frequency-inverse document frequency of text information is perhaps the most well-known feature vector representation of input data[53]. The tf-idf function weights each vector component (each of them relating to a word of the vocabulary) of each document on the following basis. First, it incorporates the word frequency in the document. Thus, the more a word appears in a document (e.g., its tf, term frequency is high) the more it is estimated to be significant in this document. In addition, idf measures how infrequent a word is in the collection. This value is estimated using the whole training text collection at hand. Accordingly,

if a word is very frequent in the text collection, it is not considered to be particularly representative of this document (since it occurs in most documents; for instance, stop words). In contrast, if the word is infrequent in the text collection, it is believed to be very relevant for the document. The tf-idf feature vectors are commonly used in text information retrieval community to compare a query vector with a document vector using a similarity or distance function such as the cosine similarity function. In Chapter 3 we discuss the formulation and structure of the tf-idf feature vectors and an extension of this representation to continuous vector spaces.

Similar to tf-idf representation of text data, computer vision and signal processing communities have defined a set of feature vectors describing images[54]. Features may be specific structures in the image such as key-points, corners, edges, blobs, etc. Features may also be the result of a general neighborhood operation or feature detection applied to the image (e.g. the scale invariant feature vectors (SIFT)[55]). Other examples of features are related to motion in image sequences or videos, in terms of curves or boundaries between different image regions, or to properties of such a region[56].

An image feature can often be represented in different ways. For example, an edge can be represented as a Boolean variable in each image pixel that describes whether an edge is present at that pixel or not. Alternatively, we can define a representation which provides a measure instead of a Boolean statement of the edge's existence and combine this with information about the orientation of the edge. Similarly, the color of a specific region can either be represented in terms of the average color (three scalars) or a color histogram (three functions). The choice of a feature representation can often drastically impact the performance of computer vision systems[57]. While a more descriptive feature representation can boost the performance of the vision system, the cost of solving problems in higher-dimensions becomes a hurdle in rich data modalities such as images and video frames.

Similar to vision systems various feature vectors have been developed in the machine learning community for audio processing, music information retrieval and speech recognition. Audio signals are often characterized by signal-based features[58–60] including timbre, harmony, and rhythm. The short-time audio features are mainly derived from short segments of the signal spectrum and include spectral centroids, Mel-frequency cepstral coefficients (MFCC) [61], and octave based spectral contrast

(OSC)[62]. The long-time audio features are mainly based on variation of spectral or beat information over a long segment of the audio signal. Typical examples of long-time audio features include Daubechies wavelet coefficients histogram (DWCH) [63], octave-based modulation spectral contrast (OMSC), low-energy and beat histogram[61]. Low-level descriptors often lack high-level abstractions that are critical for audio classification tasks[64]. Several high-level features such as distance-based[65] and statistical[66] features are proposed to represent audio signals. There has been several attempts in combining low- and high-level audio features for classification tasks[66–69] or using complementary information channels such as lyrics (for songs), web and social tags[70–72].

In this thesis we mainly evaluate the classification performance of proposed classifiers on audio datasets. We formulate a set of classification problems such as music genre classification and artist identification to illustrate the efficiency of our sparse approximation framework. Throughout this thesis, we use a subset of audio features. The selected audio features including both short-time and long-time audio features. MFCCs[61] and chroma features[61] are our main short-time features that are extracted using a sliding texture window. Spectral centroid, entropy, spectral irregularity, brightness, roll off, spread, skewness, kurtosis and flatness are also extracted as the representation of long-time audio characteristics. Table 2.1 illustrates the dimensionality of each feature vector.

Table 2.1: Selected audio features and dimensionality of the feature space is shown above. The short-time audio features are represented using the dimensionality of the mean feature vector.

Audio Feature	Dimensionality
MFCCs	13
spectral centroid	1
entropy	1
spectral irregularity	1
brightness	1
roll off	1
spread	1
skewness	1
kurtosis	1
flatness	1
chroma	12

The MFCC feature vectors have been shown to be effective for audio classification[73]. They represent short-duration musical textures by encoding a sound's timbral information[74;75]. The

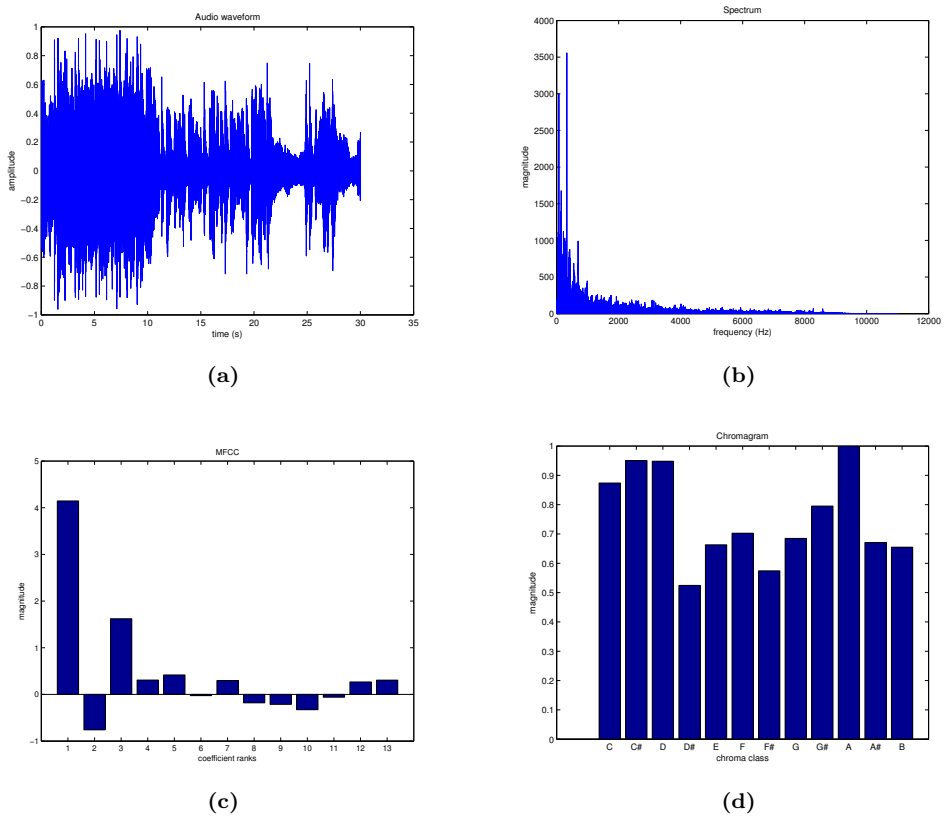


Figure 2.1: A sample song is represented: (a) time domain (b) frequency spectrum, (c) MFCCs and (d) Chroma features.

process of constructing the MFCCs begins by applying a Fourier transform to fixed size overlapping windows. A series of transformations combining an auditory filter bank with a cosine transform will result in a discrete representation of each audio window in terms of MFCC descriptors. The filter bank is constructed using 13 linearly-spaced filters followed by 27 log-spaced filters [74;75]. Each short-time audio window is then represented as an MFCC vector composed of 13 cepstral coefficients.

We also utilize the chroma features [76]. This feature is a 12-dimensional vector, where each element represents the spectral energy corresponding to one pitch class. To compute a chroma vector from a magnitude spectrum, we assign each bin of the fast Fourier transform (FFT) to the pitch class of the nearest step in the chromatic equal-tempered scale. Then, given a pitch class, we average the magnitude of the corresponding bins. This results in a 12-value chroma vector. Figure 2.1 illustrates a sample music from the dataset. The original audio is shown in Figure 2.1a.

The frequency spectrum and the MFCCs are illustrated in Figure 2.1b and Figure 2.1c respectively. Figure 2.1d illustrates the chroma features histogram for the input audio signal. Both short-time feature vectors in combination with signal properties are aggregated into a single feature vector characterizing audio information.

The concept of feature is generic and the choice of features in a particular classification task is highly dependent on the specific problem at hand. In this thesis, we fix a subset of feature vectors to compare the classification accuracy without bias for additional information in different features. We note that this feature vector selection might not be optimal for all classification problems and is mainly used as a proof of concept. Feature learning models such as convolutional neural networks is introduced later on as an alternative to feature engineering in classification problems.

2.3 An Overview of SVM

In Section 2.1 we introduced an overview of the classification problem and well-studied approaches used in the literature for this problem. In this section we examine the theoretical foundations of SVMs as one of the most popular classification techniques. SVM as shown by [77;78] is a linear threshold classifier in some feature space, with maximum margin and consistent with the training examples. Throughout this section, we only consider the problem of using SVM for binary classification as a proof of concept. A multi-class SVM is directly obtainable from a binary SVM using the output coding techniques such as the models proposed by [1;2]. In the binary classification setting, every instance (the feature space representation of training examples) x has a corresponding label $y \in \{-1, 1\}$ indicating whether it belongs to a specific class or not.

Any linear threshold classifier $w(x)$ corresponds to a vector $w \in \mathbb{R}^n$ such that the prediction of w for the instance x is $w(x) = \text{sign}\left(w^\top x\right)$; as a result, we identify the linear threshold classifiers with their corresponding vectors. For simplicity we only focus on classifiers passing through the origin. The results can be simply extended to the general case. Observe that since the prediction $w(x)$ does not change by rescaling, without loss of generality we can always assume that every instance x is normalized to have unit ℓ_2 norm, i.e., $\|x\|_2 = 1$.

Whenever the training examples are not linearly separable soft margin SVM is used as proposed

by [77]. Soft margin SVM allows for mislabeled examples. If there exists no hyperplane that can split the labeled examples, the soft margin SVM will choose a hyperplane that splits the examples as cleanly as possible, while still maximizing the distance to the nearest cleanly split examples. The idea is to simultaneously maximize the margin and minimize the empirical hinge loss. Specifically, let

$$H(x) \doteq (1+x)_+ = \max\{0, 1+x\} \tag{2.2}$$

denote the *hinge function*, and let $S \doteq \langle (x_1, y_1), \cdots, (x_M, y_M) \rangle$ be a set of M labeled training data. For any linear classifier $w \in \mathbb{R}^n$ we define its empirical hinge loss[1]

$$\hat{H}_S(w) \doteq \mathbb{E}_{(x_i,y_i)\sim S}\left[\left(1 - y_i w^\top x_i\right)_+\right]. \tag{2.3}$$

The empirical (ℓ_2) regularization loss is similarly defined to be

$$\hat{L}(w) \doteq \hat{H}_S(w) + \frac{1}{2C}\|w\|^2, \tag{2.4}$$

where C is the regularization constant. Soft margin SVM then minimizes the empirical regularization loss which is a convex optimization program.

The following theorem is a direct consequence of the convex duality.

Theorem 1. *Let $\langle (x_1, y_1), \cdots, (x_M, y_M) \rangle$ be a set of M labeled training examples, and let w be the SVM classifier obtained by minimizing Equation 2.4. Then the SVM classifier can be represented as a linear combination of the training examples[3], i.e., $w = \sum_{i=1}^{M} \alpha_i y_i x_i$. Moreover,*

$$\textit{for all } i \in \{1, \cdots, M\}: \qquad 0 \le \alpha_i \le \frac{C}{M}. \tag{2.5}$$

Proof. The optimization problem of Equation 2.5 can be written as

[1] The defined empirical hinge loss of equation 2.2 is an upper bound for the classification error.

$$\begin{aligned} &\text{minimize } \frac{1}{2}\|w\|^2 + \frac{C}{M}\sum_{i=1}^{M}\xi_i \\ &\text{subject to :} \\ &\qquad \forall\ i:\ \xi_i \geq 0 \\ &\qquad \forall\ i\ :\ \xi_i \geq 1 - y_i w^\top x_i \end{aligned} \tag{2.6}$$

We form the Lagrangian of the above optimization problem by linearly combining the objective function and the constraints we have $\mathcal{L}(w, \xi, s, \eta) =$

$$\frac{1}{2}\|w\|^2 + \frac{C}{M}\sum_{i=1}^{M}\xi_i + \sum_{i=1}^{M}\alpha_i(1 - y_i w^\top x_i - \xi_i) - \sum_{i=1}^{M}\eta_i\xi_i. \tag{2.7}$$

By writing the KKT [2] conditions, we obtain the conditions for the saddle point of the Lagrangian function. The optimal classifier is a linear combination of the training examples:

$$w - \sum_{i=1}^{M}\alpha_i y_i x_i = 0. \tag{2.8}$$

The optimal dual variables α_i and η_i are all non-negative and for every index i we have $\frac{C}{M} - \alpha_i - \eta_i = 0$, which implies that $\alpha_i \leq \frac{C}{M}$. Therefore, the optimization problem of equation 2.6 can be written

[2] The KarushKuhnTucker (KKT) conditions are first order necessary conditions for a solution in nonlinear programming to be optimal, provided that some regularity conditions are satisfied.

as

$$\text{minimize } \frac{1}{2}\sum_{i,j=1}^{M}\alpha_i\alpha_j y_i y_j {x_i}^\top x_j + \frac{C}{M}\sum_{i=1}^{M}\xi_i$$

subject to:

$$\begin{aligned} &\forall\ i:\ \xi_i \geq 0 \\ &\forall\ i\ :\ \xi_i \geq 1 - y_i\sum_{j=1}^{M}\alpha_j y_j {x_j}^\top x_i \\ &\forall\ i:\ 0 \leq \alpha_i \leq \frac{C}{M}. \end{aligned} \tag{2.9}$$

Furthermore, its dual program can be stated as:

$$\text{maximize } \sum_{i=1}^{M}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{M}\alpha_i\alpha_j y_i y_j {x_i}^\top x_j$$

subject to:

$$\forall\ i \in \{1,\cdots,M\}:\ 0 \leq \alpha_i \leq \frac{C}{M}. \tag{2.10}$$

This optimization is a quadratic program and can be solved efficiently using convex optimization methods[79].

SVM provides an efficient classification scheme based on maximizing the margin between training examples belonging to different classes. However, a majority of the training examples are involved in making the final decisions for the unseen (test) examples. The final decision for any new instance x is sign $\left(\sum_{\mathrm{i}=1}^{\mathrm{M}}\alpha_{\mathrm{i}}\mathrm{y}_{\mathrm{i}}{\mathrm{x}_{\mathrm{i}}}^\top\mathrm{x}\right)$. On the other hand, principles of learning theory[3] suggest that simpler models in which a small, well-chosen subset of all training examples is only involved in making the final classification decisions is statistically more preferable. In Chapter 4, we introduce a sparse approximation model that addresses the inherent issues of classic SVM, namely the sparsity-eager SVM. The sparse approximation framework , can be viewed as an alternative classification framework which aims to base the classification decision on a small number of well-chosen training examples.

2.4 Conclusions

In this chapter we explored the classification problem as a fundamental step in IR. We reviewed the main components of a classification task, the feature representation and, the choice of a learning model. In Section 2.1, we examined the various classification techniques and their applications in machine learning and pattern recognition. In Section 2.2 we explored various feature representations for classification. In Section 2.3 we described the theoretical foundations of support vector machines (SVM) classifiers as one of the most popular learning models. Later in Chapter 4, we introduce the ℓ_1-SVM classifier to address the inherent issues of classic SVM.

Chapter 3: Explicit Semantic Analysis

In this chapter we propose a model that explores the semantic space representation of time-series signals by presenting the signal-based features (code-words) in terms of a set of high-level concepts (documents). Our work is closely related to Explicit Semantic Analysis (ESA) [80] that utilizes term-frequency (tf) and inverse document frequency (idf) weighting schema to represent text information in a higher-dimensional concept space. We show the efficiency of this representation through a unimodal classification task on a public time-series dataset. Later, in Chapter 5 we combine this representation with correlation analysis to introduce a cross-modal retrieval framework.

Computers understand very little of the text information and foundations of natural language. For humans to be able to effectively communicate with computers and for computers to be able to organize and process text documents, a semantic model of text information is required. Over the past decades, the information retrieval community has proposed numerous models that can effectively represent text information in a semantic space. The vector space model or the term vector model is one of the most widely used models to represent text information. This algebraic model represents text information [1] as vectors of identifiers such as index terms. This model was first introduced by Salton *et. al* [81;82] in 1971 and has been used ever since for indexing, search and organizing documents in large scales.

ESA [80] is an instance of vector space model where the relationship between a code-word and a concept pair is captured through the so-called term frequency-inverse document frequency (tf-idf) value of the (word, concept) pair. The tf-idf values will be obtained by multiplying tf (the frequency of a given word in the document), and the idf values (the inverse document frequency that quantifies the importance of a word over a given set of concepts). The set of tf-idf scores associated with a code-word will form a vector representation of it in the concept space. This model can be used in combination with classification methods for categorization of labeled time-series data.

[1] The vector space model is not limited to text information and can be used for objects in general.

This chapter presents an extension of the ESA model for time-series signals such as audio sequences. We evaluate the efficiency of the proposed model through a set of experiments for a classification task on a public audio dataset. First, in Section 3.1 we review the background work on vector space models and significance of these family of models for information retrieval. In Section 3.2 we dive deepr into foundations of vector space models. In Section 3.3 we discuss a tf-idf weighting as a classic vector space model for text document representation. Section 3.4 explores the application of tf-idf model in explicit semantic analysis. In Section 3.5 we introduce the ESA encoding of time-series signals. In Section 3.6 we evaluate the proposed vector space model of time-series on a public dataset for a classification task. Finally, Section 3.7 summarizes the chapter.

3.1 Background

Information retrieval is the process of gathering information resources that are most relevant to an information need (query) from a collection of data. The search process can be based on metadata, text information or any other content-based indexing schema. Information retrieval models are widely used to collect relevant data and reduce the *information overload* with the availability of many different information sources[83]. Perhaps the most visible application of information retrieval systems is the web search engines.

An information retrieval system ranks the database information according to relevance to a given user query. Queries represent information needs such as text strings in search engines and can be any object such as images, audio, text or metadata. The information retrieval process then identifies the objects in the database that match the issued query by different degree of relevance. The majority of information retrieval systems rank the database objects according to relevance to query by a numerical score and report the ranked list of the top relevant objects[84].

One of the biggest obstacles to a practical information search and retrieval system is an effective communication system between computer and natural human language. Search engines have emerged around understanding queries, indexing and matching documents that are highly correlated by the issued query. Semantic [2] technologies such as vector space models have been developed over

[2]In this thesis, the *semantics* refers to general meaning of a text phrase or document.

the past decades to facilitate data organization and information retrieval[85].

The SMART information retrieval system introduce by Salton *et. al* in 1971[82] pioneered the basics of modern search engines[86] such as a vector space model for text documents. The vector space model introduced in SMART projects text documents and user queries into a latent semantic space where the similarity between a query and a specific document can be measured by a distance metric such as cosine similarity. The retrieval system then returns a sorted list of documents ordered by semantic similarity to the user query[85].

The successful application of vector space models in information retrieval has inspired several applications in natural language processing[87–89]. Rapp *et. al* used a vector space model of word meanings for Test of English as a Foreign Language (TOEFL) and Turney *et. al* applied a similar framework for SAT college entrance tests with impressive results. The vector space models are also used in search engines to measure the relevance or *similarity* between queries and corpus of documents[86].

The vector space models are also used in the machine learning community for various tasks such as classification[90–92] and building recommender systems[93–95]. The input examples for a given machine learning task are often characterized as a feature vector. These feature vectors are sometimes derived from event frequencies such as the occurrence rate of a term in an input document. The vector space models employ frequencies in a corpus as a clue for semantic information[92]. Similar to classification tasks, collaborative filtering methods and recommender systems also use feature vectors. In a recommender system, the input is often a matrix where each row represents a person, each column represents an item or object and the value at each index corresponds to a rating of that item for that person. The algebraic models that are built for vector space models on text documents can then be applied in a similar manner to build a recommender system[93–95]. In Section 3.2 we discuss vector space models in details and explore a tf-idf weighting in Section 3.3. We then introduce an extension of the vector space model in Section 3.5. In Section 3.6 we evaluate the proposed vector space model of time-series on a public dataset for a classification task and discuss our findings in Section 3.7.

3.2 Vector Space Model

A vector space model is an algebraic method to represent raw information such as text documents, audio signals, images and metadata as a vector of identifiers such as index terms or frequencies of occurrence. The vector space models are sometimes referred to as the term vector model. Vector space models are frequently used in information retrieval[82], information filtering[96], indexing[81], relevancy ranking[82;97] and machine learning[90–95]. The vector space models were first introduced by Salton *et. al* in the SMART information retrieval system[82]. The novelty of this model lies in the utilization of frequencies in a corpus of a text document as a clue for discovering semantic information[85]. While the vector space model was originally developed on text information, they can be extended to any input information modality such as audio signals, images, videos and metadata.

Vector space models provide an attractive alternative to the classic standard Boolean model in information retrieval[98]. The standard Boolean model[99–101] is the first and most adopted information retrieval model. This model is based on the classic set theory and Boolean logic. In a standard Boolean model the corpus of documents and the queries are both represented as sets of terms. The retrieval model is then simply based on the document containing the query terms.

In a standard Boolean model, we often assume a finite set of index terms which are often the stemmed words in the corpus of the documents. These index terms are used to describe or characterize *all* documents in the corpus of our dataset.

Definition 3.2.1. We define T as the superset of index terms with $T = t_1, \ldots, t_j, \ldots, t_m$.

We then represent each document of the corpus in terms of the index terms as follows:

Definition 3.2.2. We define documents as index terms by $D = d_1, \ldots, d_i, \ldots, d_n$ where each d_i is an element of T.

A user query, Q, can then be translated as a Boolean expression in a normal form as follows:

$Q = \{(w_i \text{ OR } w_k \text{ OR } \ldots)\text{AND } \ldots \text{AND } (w_j \text{ OR } w_s \text{ OR } \ldots)$ where $w_i = t_i$ or 0. The query can simply be formed as a disjunctive normal form as well. The standard Boolean retrieval algorithm the identifies the set S of documents obtained that contain the term t_i. These documents are then

retrieved in response to Q as a result of set operations described in Algorithm 1.

Algorithm: BooleanRetrieval(T,D,Q)

Input: T: index sets term, D: documents set, Q: user query
Result: S: set of retrieved documents for Q
$S_j = \{d_i | w_j \in d_i\}$;
$S = \{\cup(\cap S_j)\}\forall j$;
return S

Algorithm 1: The retrieval operation in a Boolean standard model of data retrieval.

The standard Boolean model provides an intuitive information retrieval model that is easy to implement and comprehend. This model, however, does an exact matching on query terms and the corpus of documents. This can result in retrieving too many or too few documents that match a query. Moreover, in the standard Boolean model all terms are wighted equally and the translation of a query to a Boolean expression can present many hurdles. In this sense the standard Boolean model is more suitable for data retrieval rather than information retrieval where the retrieval model is mainly concerned with determining which documents from a collection contain the keywords in the user query [101].

In the vector space model both documents and queries are represented as vectors. This vector representation can be extended to other data modalities besides text as discussed in Section 3.5. Let $d_j = (w_{1,j}, w_{2,j}, \ldots, w_{t,j})$ and $q = (w_{1,q}, w_{2,q}, \ldots, w_{n,q})$ denote the document and query vectors where each dimension corresponds to a separate term. In this context the term definitions vary by the application and can be stemmed words, keywords, longer phrases or objects. The number of terms in the vocabulary [3] is the dimensionality of the vector representation. If a term occurs in the document, its value in the vector also known as *the term weights* is assigned as a positive number. The vector space models take advantage of different methods of assigning term weights. The vector space model then utilizes vector and tensor operations to retrieve the top relevant documents for a specific query. In Section 3.3 we discuss the tf-idf model as one of the most well-known term-weight

[3] Vocabulary can refer to a set of words but it can also refer to a set of visual words such as image patches or audio words such as clustered feature vectors.

assignments in information retrieval[102] and discuss vector operations that are used for information retrieval.

The vector space model provides a simple algebraic alternative for information retrieval to standard Boolean model. In this model the term wights are not binary and hence allow for a computation of a continuous degree of similarity between user queries and documents in the corpus of the dataset. The vector space model also provides a framework for the ranking of *all* documents in the corpus by the relevancy score. In this sense, the vector space model can be used for partial matching of user queries and the documents in the corpus and is well-suited for information retrieval rather than data retrieval which is primarily achieved through standard Boolean models of information retrieval.

The vector space model performs an exact match on search and document terms. In this sense the substrings can lead to a *false positive match*. The model is also semantically sensitive, i.e. documents with similar context but different vocabulary terms would be missing from the relevancy results. This can result in *false negative matches*. The co-occurrence or ordering of the terms is also neglected and the terms are assumed to be statistically independent in the classic vector space models. Various tools and mathematical models such as singular value decomposition[103] and lexical databases[104;105] have then been integrated into the classic vector space models to address these limitations.

Several models have been proposed in the literature of the information retrieval community to extend the vector space models and deal with its limitations. A generalized vector space model for information retrieval has been proposed by Wong *et. al*[106?] to introduce a term to term correlation, which deprecate the pairwise orthogonality assumption in the classic vector space model. Term Discrimination is also introduced by Salton *et. al*[102] as a way to rank keywords in how useful they are for information retrieval[107]. The latent semantic analysis (LSA[108]) is developed to analyze the relationships between a set of documents and the terms they contain, by producing a set of concepts related to the documents and terms[108]. Text classifiers such as the Rocchio classifier has been built as an extension of the vector space models[86]. Finally random indexing models for dimensionality reduction of distributional semantics have been developed as a vector space model[109;110].

3.3 The tf-idf Model

The term frequency inverse document frequency (tf-idf) was first introduced by Salton *et. al*[81] as the term weights in the classic vector space model. The tf-idf model is a numerical statistic that is intended to reflect how important a word is to a documents of a corpus. In essence the tf-idf weights increase proportionally to the frequency of a word in the document, and is offset by the frequency of the word in the corpus of all documents. This results in adjusting for words that appear frequently in most documents and is then can be used to extract semantic information from documents which is used in search, ranking and information retrieval. The tf-idf wighting schema is widely used by search engines as a central tool in scoring and ranking a document's relevance given a user query[111;112]. The machine learning community also utilizes the tfidf model for stop-words filtering in various subject fields including text summarization and classification[113;114].

In the classic vector space model proposed by Salton *et. al*[81], the term weights in the document vectors are the tf-idf weights which are the products of local and global parameters. The local parameters are defined as the factors that are extracted per document (term frequency) while the global parameters denote the factors extracted from all documents in the corpus (inverse document frequency).

If we denote the documents set or the corpus by $D = \{d_1, \ldots, d_{|D|}\}$ we can define the term frequency and inverse document frequency as follows:

Definition 3.3.1. Term frequency of term t in document d is defined as $\mathrm{tf}_{t,d}$ and denotes the frequency of occurance for term t in a document d (local parameter).

Definition 3.3.2. Inverse document frequency is defined as:

$$\mathrm{idf}_{t,d} = \log \frac{|D|}{|\{d' \in D \,|\, t \in d'\}|} \tag{3.1}$$

where $|D|$ is the total number of documents in the document set D and $|\{d' \in D \,|\, t \in d'\}|$ is the number of documents containing the term t (global parameter).

Using the above definitions, the term weight vector for document d, $\mathbf{v}_d = [w_{1,d}, w_{2,d}, \ldots, w_{N,d}]^T$, is obtained as:

$$w_{t,d} = \mathrm{tf}_{t,d} \cdot \mathrm{idf}_{t,d} = \mathrm{tf}_{t,d} \cdot \log \frac{|D|}{|\{d' \in D \,|\, t \in d'\}|} \tag{3.2}$$

The cosine similarity is often used in the literature of information retrieval community as a similarity measure between queries and documents and is defined as:

Definition 3.3.3. Cosine similarity distance: Given a pair of feature vectors x and y the cosine similarity distance $d(x, y)$ estimates the angle between the two vectors and is defined as $d(x, y) = \frac{x^t.y}{||x||.||y||}$, where $||.||$ denotes the length function.

Using the cosine the similarity between document d_j and a user query q can be calculated as:

$$\mathrm{sim}(d_j, q) = \frac{\mathbf{d_j} \cdot \mathbf{q}}{\|\mathbf{d_j}\| \, \|\mathbf{q}\|} = \frac{\sum_{i=1}^{N} w_{i,j} w_{i,q}}{\sqrt{\sum_{i=1}^{N} w_{i,j}^2} \sqrt{\sum_{i=1}^{N} w_{i,q}^2}} \tag{3.3}$$

Intuitively the tf-idf weighting schema assigns a higher score to the words that appear frequently in a document and not frequently across all the documents in the corpus. In the same manner, the tf-idf model assigns lower scores to words that appear frequently among all documents in the corpus. The tf-idf model provides a simple and efficient weighting schema. This model can be enhanced by modifications in term-frequency definition [115] and using stemming [116] and stop-words filtering in different applications [117].

3.4 Explicit semantic analysis

Explicit semantic analysis (ESA) [53] is a vector representation of text documents that uses a document corpus as a knowledge base. The ESA model has originally been proposed by Gabrilovich *et. al* [53] as an extension to the vector space model for the enhancement of text categorization [53]. The ESA model was later on adopted for *semantic relatedness* computation. In this sense, the cosine similarity between the vector representation of documents are computed and collectively interpreted as a

semantic space of *concepts explicitly defined by humans*, where Wikipedia articles or otherwise titles of documents in the knowledge base are equated with concepts[118]. The explicit semantic analysis then contrasts with latent semantic analysis (LSA)[108] since the concepts from the knowledge base in this vector space model are readable by humans (semantics)[118;119].

The ESA model utilizes the tf-idf weighting schemata of Section 3.3 to represent low-level textual information in terms of concepts in higher-dimensional space. Specifically, in the ESA model, the relationship between a code-word and a concept (document) pair will be captured through the tf-idf value of the word-concept pair. The tf-idf values will be obtained by multiplying tf (the frequency of a given codeword in the document/concept) and the idf values (the inverse document frequency that quantifies the importance of a word over a given set of concepts). The set of tf-idf scores associated with a codeword will form a vector representation of the code-word in the concept space.

The original ESA model assumes a large and diverse knowledge base (all Wikipedia articles), D where $|D| = N$ denotes the number of documents in the knowledge base. The documents in the knowledge base are transformed into a *bag of words* model[120] which essentially is a term frequency histogram. The bag of words are stored in an inverted index which can be used to find corresponding documents in the knowledge base for each word. In this sense the concepts of the semantic space are the titles of documents in the knowledge base and each word correlates to the concepts in the corresponding inverted index[119].

Once a single word query is issued, the inverted index is used to find a list of indexed documents from the knowledge base each with a tf-idf score. This results in an N-dimensional vector of scores per word. A zero score in this vector corresponds to documents not containing the query word. The same model can be used to compute the similarity between two words by comparing the two word-document vectors by a cosine similarity measure as defined in Section 3.3. Intuitively, the ESA model first projects both query words into a semantic space (presented by word-document vectors) where a similarity measure such as cosine similarity can be obtained as a score. Let u and v denote the projection of two query words in the semantic space. The similarity of query words, $word_1$ and

$word_2$ can then be described as:

$$\mathsf{sim}(\mathbf{u}, \mathbf{v}) = \frac{\mathbf{u} \cdot \mathbf{v}}{\|\mathbf{u}\|^2 \|\mathbf{v}\|^2} = \frac{\sum_{i=1}^{N} u_i v_i}{\left[\sum_{i=1}^{N} u_i^2\right] \left[\sum_{i=1}^{N} v_i^2\right]} \tag{3.4}$$

and this gives numeric estimate of the semantic relatedness of the two words. The same scheme can then be extended from single words to multi-word text documents by simply summing the vectors of all words in the text[53]. This model is then used to compute the similarity of query documents by a projection into a semantic space.

The original ESA model[53] makes the assumption that the knowledge base contains topically orthogonal concepts. It has later been shown in the literature of information retrieval and machine learning that this assumption is not necessary[121;122]. Various extensions of the ESA model have also been proposed for cross-language explicit semantic analysis (CL-ESA) where the relatedness of two documents in different languages is assessed by the cosine similarity between the corresponding vector representations in the semantic space[123;124].

In this chapter we present our generalization of the ESA model that will allow for the encoding of low-level time-series features (code-words) into a higher-dimensional concept space using a predefined code-book of signal-based features [4]. To this end, we extend the idf model to handle vector-based features such as low-level signal-based features. We also provide a suitable extension of the tf-idf model that expands the low-level signal representation into a semantic vector space. We use this concept-based signal representation to build a supervised learning model for a classification task as described in Section 3.5.

3.5 ESA Feature Encoding

In this section we introduce a generalization of the ESA model that will allow for the encoding of low-level time-series features. **Audio signals** are adopted as the time-series signals in our experiments. Hence from now on we denote the time-series features with audio features and the code-book

[4]This extension has been primarily developed for audio signals. Hence from now on we denote the signal-based code-words as audio words.

consists of audio-words. The signal-based features of the audio signals are defined as MFCC features as described in Chapter 1. In formation of the ESA vectors we will assume a diverse set $\mathcal{C} = \{C_1, ..., C_M\}$ of audio signals representing M concepts associated with the concept space for the ESA model. In essence our knowledge base in a repository of audio signals. We will also assume that the set $\mathcal{D} = \{\delta_1, ..., \delta_k\}$ denotes the code-book of k non-redundant audio words for concepts in $\mathcal{C}$.

The ESA matrix of concept/code-book pair $(\mathcal{C}, \mathcal{D})$ is an $M \times k$ matrix whose (i, j)-th entry represents the association between audio concept $C_i \in \mathcal{C}$ and code-word $\delta_j \in \mathcal{D}$. Specifically, for each concept $C \in \mathcal{C}$, we let $C = \langle (F_1, w_1), ..., (F_\ell, w_\ell) \rangle$, denote its feature-based representation. Each of the pairs $(F, w) \in C$ corresponds to the MFCC feature F that appears in concept C with a frequency w. Given a pair of features x and y, we use $d(x, y)$ to denote their (cosine) similarity.

Definition 3.5.1. The *fractional term frequency* for a feature x with respect to a concept C indicates the overall similarity of feature x with concept C and is defined as follows:

$$tf(C, x) = \frac{\sum_{i=1}^{\ell} w_i \times d(F_i, x)}{\sum_{i=1}^{\ell} w_i}, \tag{3.5}$$

where $C = \langle (F_1, w_1), ..., (F_\ell, w_\ell) \rangle$.

Definition 3.5.2. Indicator function $\chi(x, C)$ is defined for feature x with respect to concept C as:

$$\chi(x, C) = \max_{F \in C} d(x, F). \tag{3.6}$$

The *indicator function* estimates the closest audio word from the code-book to represent an observed feature x in the concept space. For a given audio word, $\delta \in \mathcal{D}$, we will define its *fractional inverse document frequency* as:

Definition 3.5.3. Fractional inverse document frequency

$$idf_\delta = \log\left(\frac{M}{\sum_{i=1}^{M}\chi(\delta, C_i)}\right). \tag{3.7}$$

Using these notations, the *fractional tf-idf* between feature δ and concept C will be estimated as:

Definition 3.5.4. Fractional tf-idf

$$tfidf(C, \delta) = tf(C, \delta) \times idf_\delta. \tag{3.8}$$

Finally, we will define the ESA matrix $\mathcal{E}_{\mathcal{C},\mathcal{D}}$ to be an $M \times N$ matrix with its $(i, j) - \text{th}$ entry defined as:

$$\mathcal{E}_{\mathcal{C},\mathcal{D}}[i, j] = tfidf(C_i, \delta_j). \tag{3.9}$$

We note that the higher the value $\mathcal{E}_{\mathcal{C},\mathcal{D}}[i, j]$ is the more important the code-word δ_j is in describing of the audio concept C_i. Each audio concept is then presented in terms of tf-idf weight vectors, where the classification is performed in the concept space rather than the original feature space.

For a codeword $\delta \in \mathcal{D}$ we use $\mathcal{E}(\delta)$ to denotes its ESA encoding, i.e., represents the column of ESA matrix associated with codeword δ. For a given audio sequence A, we assume the function **MFCC**(A) computes the set $\{f_1, ..., f_\ell\}$ of unique MFCC features. In what follows, we will use the procedure the ESAEncoding(.) in Algorithm 2 for computing the MFCC features of an input audio sequence A and aggregates the ESA vectors of best matching codeword in D to compute the ESA vector for A.

In order to build a supervised classifier based on ESA encodings, we will assume that a set of t ordered pairs $\mathcal{T} = \{(A_1, L_1), ..., (A_t, L_t)\}$ of audio sequences, A_i, and their corresponding labels, L_i, are provided as the training data. In practice, this set can be the same as the audio concepts $\mathcal{C}$ utilized in the construction of matrix $\mathcal{E}$. Our hypotheses is that one can build a classifier with a reasonable performance simply based on the ESA vectors of audios and their labels in $\mathcal{T}$. To this

Algorithm: ESAEncoding$(A, \mathcal{D}, \mathcal{E})$

Input: A: input audio, $\mathcal{D}$: code-book, $\mathcal{E}$: ESA matrix
Result: $\mathcal{E}(A)$: ESA-representation of A

$\{f_1, ..., f_\ell\} \leftarrow \mathbf{MFCC}(A)$;
$\mathcal{E}(A) \leftarrow \mathbf{0}$;
foreach $f \in \{f_1, ..., f_\ell\}$ **do**
    $\delta^* = \min_{\delta \in \mathcal{D}} d(f, \delta)$;
    $\mathcal{E}(A) = \mathcal{E}(A) + \mathcal{E}(\delta^*)$;
end
return $\mathcal{E}(A)$

Algorithm 2: Construction of the ESA vector of an audio sequence.

end, we will form the set

$$\mathcal{E}(\mathcal{T}) = \{(\mathcal{E}(A_1), L_1), ..., (\mathcal{E}(A_t), L_t)\}, \tag{3.10}$$

with $\mathcal{E}(A_i)$= ESAEncoding$(A_i, \mathcal{D}, \mathcal{E})$, $i = 1, ..., t$. The set $\mathcal{E}(\mathcal{T})$ of (ESA vector, label) pairs will be provided as the training data to a supervised classifier algorithm such SVM, building a model that assigns samples (points in the SVM feature-space) into their classes or categories. Note that, the set of hyperplanes that define these gaps between classes, i.e., decision planes, are the outcome of SVM training on $\mathcal{E}(\mathcal{T})$. Finally, to identify the final label of an audio query element q, we first form its ESA vector $\mathcal{E}(q)$ by applying ESAEncoding$(q, \mathcal{D}, \mathcal{E})$. Next, using the aforementioned classifier trained on set $\mathcal{E}(\mathcal{T})$, we can estimate the class label L_q by simply determining which side of the cell (defined by the set of decision planes and gaps) they will belong to. In Section 3.6 we use this ESA extension for a classification task on a public dataset.

3.6 Empirical Evaluation

In this section we evaluate the performance of the proposed ESA extension of feature vectors for a classification task. In specific, we perform a music genre classification on a public dataset and compare the results with a set of well-known feature representations and classifiers.

The proposed model is evaluated on the benchmark dataset for audio classification and clustering

proposed by Homburg et al. [125]. The dataset contains samples of 1886 songs obtained from the Garageband site and is comprised of 9 music genres including Pop, Rock, Folk/Country, Alternative, Jazz, Electronic, Blues, Rap/HipHop, and Funk/Soul. As illustrated in Table 3.1, the number of samples vary by the genre. Each song is associated with a 10 seconds audio sample drawn from a random position of the corresponding song. All audio samples are encoded using mp3 format with a sampling rate of 44100Hz and a bit-rate of 128 mbit/s. A total of 49 audio features including temporal features, spectral features, and phase space features (TSPS) are also provided as part of the benchmark [126] and are utilized as a benchmark.

Table 3.1: Number of songs per genre.

Genre	Number of Samples
alternative	145
blues	120
electronic	113
folkcountry	222
funksoulrnb	47
jazz	319
pop	116
raphiphop	300
rock	504

In our comparative experiments we will use two classification schemes: k-nearest neighbors and SVM [9]. We will also present the results of a random classifier as a baseline. We will perform a

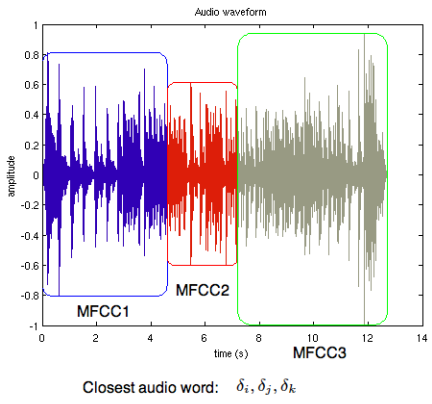


Figure 3.1: A query audio from the test set is illustrated.

10-fold cross validation to evaluate the classification accuracy. In this approach 90% of the dataset serves as the training set while a randomly selected 10% is used as the test data. We will use the *classification accuracy* as our performance measure. It corresponds to the number of audio samples correctly classified divided by the total number of audio samples in the benchmark dataset.

We process the benchmark dataset for constructing their ESA-based features. First for each audio sample, its MFCC windows are extracted using overlapping hamming windows. The MFCC features will be extracted using Auditory toolbox [74]. Each audio signal is divided into 40ms windows with 20% overlap. To reduce the complexity, only $n = 500$ non-redundant feature vectors were be randomly selected to represent each audio sample in the feature space. Next, for the selected audio descriptors, we construct the code-book by applying k-means clustering algorithm. The value of k determines the number of code-words in the dictionary. We conducted two set of classification experiments for values when $k = 1000$ and $k = 5000$. The genre classification results using ESA-based representation for the above classification algorithms are shown in Table 3.2.

The TSPS features [125] are used as a benchmark representation for supervised music genre classification on this dataset. We will also compare our results to a simple Aggregation of MFCC (AM) features by first forming constructing each a set of MFCC vectors for each audio sequence followed are their aggregation to form an average MFCC vector describing the audio sequence. MIRtoolbox [127] will be frequently used to represent each audio sample as a single vector of MFCC features. The classifiers in these experiments are k-nearest neighbors with an adaptive distance metric [128;129], ℓ_2-SVM, and, a random classifier. The classification accuracy based on these features ranged between 27% to 53%. The results of this experiments are presented in Table 3.2.

In the next experiment we evaluate the proper classification ratios of audio sequences based on each genre (true positives). In this experiment, we compute true positive ratios obtained under both TSPS and ESA models in each genre using both k-nearest neighbors and SVM learning schema. Figure 3.2 shows the true positive rate using k-NN for TSPS and ESA-based features. The results of SVM classifier are comparable to TSPS and ESA features in terms of true positive genre labels. We note that TSPS features show an overall boost to genre classification when more samples are

Table 3.2: Genre classification accuracy on the benchmark dataset: aggregation of MFCC features(AM) [127] and Temporal, spectral and phase (TSPS) features [126] are compared to ESA representation of MFCC features using three different supervised methods: k-NN, ℓ_2-SVM, and random classifier as a baseline. We considered two variations of ESA-model one based using 1000 and 5000 code-words.

Classification Method	Feature	Code-book size($\|\mathcal{D}\|$)	Average accuracy rate(%)
Random	AM	NA	22.4
Random	TSPS	NA	**21.7**
Random	ESA	$k = 1000$	29.5
Random	ESA	$k = 5000$	25.4
k-NN	AM	NA	35.8
k-NN	TSPS	NA	47.4
k-NN	ESA	$k = 1000$	48.6
k-NN	ESA	$k = 5000$	51.9
ℓ_2-SVM	AM	NA	40.8
ℓ_2-SVM	TSPS	NA	51.8
ℓ_2-SVM	ESA	$k = 1000$	53.8
ℓ_2-SVM	ESA	$k = 5000$	**57.8**

available (for example see Folk/country, Rock, Jazz and Rap/hiphop). However the ESA features seem to be less affected by number of samples available in each category resulting in higher overall true positive accuracy in most genres.

We also examine the effect of code-book size on the performance of general classification for ESA-based representation. In these experiments the number of audio-words comprising the audio code-book was varied between $k = 1000$ and $k = 5000$ while the number of descriptors for each sample was fixed, i.e., $n = 500$ per sample. The results of this experiment is illustrated in Figure 3.3. Note that expanding the vocabulary of audio keywords can slightly improve the classification accuracy. This can be explained by the fact that increasing the number of audio keywords increases the dimensionality of the feature space, which makes the representation of each audio sample sparse. This results in making the data points further apart and increases the classification accuracy.

The ESA encoding of audio (signal-based) features provide an effective semantic space for classification and categorization of audio signals. It should be noted that while the experimental results have been validated using a public music dataset, the same model can be extended to other data modalities such as images.

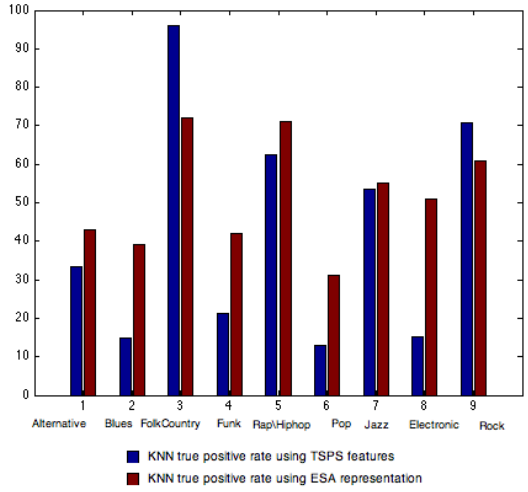


Figure 3.2: True positive genre classification rate is illustrated for each music genre. temporal, spectral and phase (TSPS) features [126] are compared to ESA representation of MFCC features using k-NN learning schema.

3.7 Conclusions

In this chapter we extend the unimodal IR models to include time-series signals. Our main hypothesis was that the ESA embeddings of audio signals can provide a better feature encoding that can boost the classification accuracy. We presented an extension of the explicit semantic model (ESA) to time-series such as audio signals by introducing a set of *fractional* term frequency (tf) and inverse document frequency (idf) models. Our hypothesis was validated through a set of experiments in applying the ESA model for a classification task on a public dataset. Specifically, the ESA model was adopted for a music genre classification task.

First, in Section 3.1 we reviewed background work on vector space models and significance of these family of models for information retrieval. In Section 3.2 we dived deeper into foundations of vector space models and the extensions of these models in the information retrieval and machine learning literature. In Section 3.3 we discussed a tf-idf wighting as a classic vector space model for text documents which has been widely used in search engines. Section 3.4 explored the explicit semantic analysis (ESA) model as proposed by Gabrilovich *et. al* [53] for text categorization in a concept space using Wikipedia articles as a knowledge base. In Section 3.5 we introduce our extension of the ESA

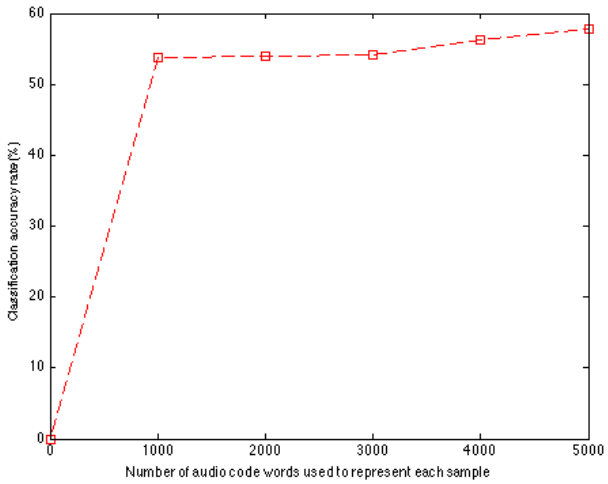


Figure 3.3: Classification accuracy results are demonstrated using varying number of code-book elements, k. In all classification experiments SVM classifier is used as the learning schema and the only variable is the number of clusters used to form the audio code-book.

encoding for time-series signals such as audio sequences. In Section 3.6 we evaluated the proposed vector space model of time-series on a public dataset for a classification task and provided empirical evidence that the ESA embeddings can boost the classification accuracy and provide a suitable representation of time-series features.

Chapter 4: Sparsity-Eager Support Vector Machines

In this chapter, we introduce a new classification algorithm, namely the ℓ_1-SVM which is closely related to the classic support vector machines (SVM) of Vapnik *et. al* [130]. Our sparsity-eager approach is motivated by inherent shortcomings of SVM. We combine the ideas behind SVM and ℓ_1-regression methods to come up with the more robust ℓ_1-SVM algorithm with the goal of (i) obtaining higher generalization accuracy on test data, (ii) increasing the robustness against overfitting to the training examples, and (iii) providing scalability in terms of the classification complexity. We provide evidences on the advantages of the new ℓ_1-SVM classifier over the baseline classification methods. We also introduce a new deep neural network architecture as a comparison to the sparsity-eager SVM. Through a comprehensive set of empirical results we evaluate the performance of the proposed classifier.

This chapter is organized as follows; in Section 4.1, we briefly describe the motivations behind sparse approximation techniques in classification. In Section 4.2 we develop a deep neural network architecture for classification as a comparison to the proposed ℓ_1-SVM. We describe the manifold learning techniques as an accuracy boosting method for classifiers in Section 4.2.1. We introduce the ℓ_1-SVM classifier in Section 4.3 to address the inherent issues of classic SVM. We compare the classification performance of the proposed classifier with state-of-the-art classifiers in Section 4.4 on a public dataset. Finally, we summarize our findings in Section 4.5 and discuss further research directions.

4.1 Background

SVMs are a set of supervised learning methods used for classification, regression, anomaly and outlier detection. In Chapter 2 we reviewed the theoretical foundations of this models. SVMs provide different advantages, most notably: (i) they are highly effective in high-dimensional spaces, (ii) the use of a subset of training examples in the decision function or the so-called support vectors

make SVMs a memory efficient paradigm, (iii) they are versatile in the sense that different kernel functions can be specified for the decision function.

SVMs however, utilize a large number of training examples in making the final classification decision. This complexity can result in an overfitting. If the number of features is much greater than the number of samples, the method is likely to give poor performances[131]. The basics of learning theory suggest that a simpler model in which only a smaller subset of training examples are involved in making the classification decision is more scalable and can boost the performance. This motivates the utilization of sparse approximation techniques in combination with SVMs.

4.2 Deep Neural Networks

In recent years, deep learning architectures [132] such as deep neural networks (DNN)[133], convolutional deep neural networks[134], deep belief networks[135] and recurrent neural networks[136] have been applied to fields like automatic speech recognition, natural language processing, audio recognition and bioinformatics where they have been shown to produce state-of-the-art results on various tasks and have won numerous contests in pattern recognition and machine learning. The success with deep network architectures on large-scale datasets has inspired classification efforts using deep networks[137;138]. Deep networks have recently been used to improve feature extraction on audio signals and boost the music classification accuracy[139]. These efforts pose an interesting question: *Can neural networks outperform hand-crafted features in combination with well-known classifiers such as SVMs?* In this Chapter, we explore the answers to this question in an experimental settings on a public dataset. We first define a network architecture in combination with various dimensionality reduction techniques for a classification task. Next we compare the performance of the DNNs with classic SVM, the proposed ℓ_1-SVM and a set of baseline classifiers such as logistic regression and ℓ_1-regression in Section 4.4.

4.2.1 Manifold Learning

Manifold learning is a non-linear dimensionality reduction method and is often applied in large scale datasets to avoid the *curse of dimensionality*[140] and reduce the time complexity of the learning

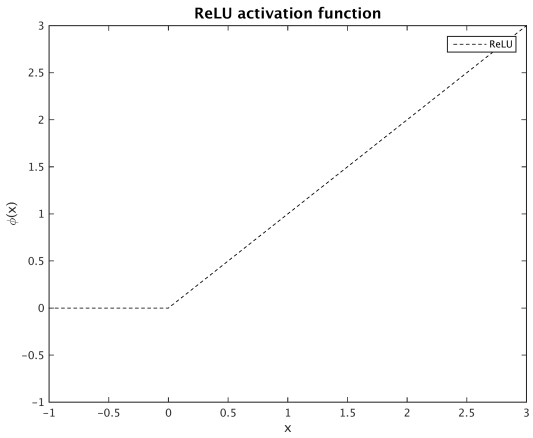


Figure 4.1: The Rectified Linear units as the activation function is illustrated.

models on high dimensional input. In this approach, it is often assumed that the original high-dimensional data lies on an embedded non-linear manifold within the higher-dimensional space.

Let matrix $M_1 \in R^{N \times L}$ represent the input matrix of all observed examples in a selected feature space, where each row represents an input example. N represents the total number of instances and L is the dimensionality of the unrolled feature representation of each input instance. If the attribute or representation dimensionality L is large, then the space of unique possible rows is exponentially large and sampling the space becomes difficult and the machine learning models can suffer from very high time complexity (curse of dimensionality [140]). This motivates a set of dimensionality reduction and manifold learning techniques to map the original data points in a lower-dimensional space. In this thesis, we experiment with principal component analysis (PCA) [141], supervised locality preserving projections (SLPP) [142] and spectral regression (SR) [143] as dimensionality reduction and manifold learning techniques. The $M_{DR} \in R^{N \times j}$ represents the reduced dataset with different models and serves as the input to DNNs.

4.2.2 DNN Architecture

Our proposed DNN architecture is based on a two hidden layered feed forward neural network [144]. The initial network weights are first randomly initialized using the *Xavier initialization* [145]. In our model we select the number of neurons, $n = 400$, in each network layer with an empirical approach

to boost the classification accuracy rate and minimize the final cost function of our model. The input and output relationship in the network is learned with a back-propagation where the weights W_{ij} approximate a global minima. Rectified Linear Units (ReLUs) [146] have been chosen as the activation function for the DNNs and can be described as:

$$\phi(x) = \max(0, x) \tag{4.1}$$

This activation function provides invariance to scaling, thresholds above zero, influences faster learning and is non-saturating in large scale datasets [146]. We also utilize the cross entropy function as a similarity metric. The cross-entropy function forces a penalty for wrong labels l_i during the learning process and avoids the local minimum convergence for the neural network. The cross entropy cost function is defined as:

$$Cost_{entropy} = -\sum_i \log\left(\frac{e^x}{\sum_i e^x}\right) \times t_i \tag{4.2}$$

for input x and target output t_i.

The initialization and estimation of weight values are an important part of designing a DNN. The weight regularization of learnt weights in an essential part of estimating the final cost function. We penalize the weights with a standard ℓ_2 weight regularization as follows:

$$Cost_{weighted} = C\lambda \sum |w|^2 \tag{4.3}$$

where C is the regularization constant and λ is the weight penalty. The final cost function can then be described as:

$$Cost_{final} = -\sum_i \log\left(\frac{e^x}{\sum_i e^x}\right) \times t_i + C\lambda \sum |w|^2 \tag{4.4}$$

The output of the final $Cost_{final}$ is then back-propagated throughout the network to adjust the weights. We use the well-known limited memory BFGS (L-BFGS) [147] algorithm for parameter

estimation in the DNN. Similar to the classic BFGS, L-BFGS uses an estimation to the inverse hessian matrix to steer its search through large-scale variable space. The L-BFGS however stores only a few vectors that represent the approximation implicitly rather than the dense inverse hessian approximation of the BFGS. L-BFGS requires linear memory and is particularly well suited for optimization problems with a large number of variables [147]. The memory benefits of L-BFGS makes it a suitable candidate for parameter estimation in our model and hence is selected as the optimization algorithm.

4.3 ℓ_1-SVM

In Chapter 2 we reviewed the theoretical foundations of SVM as a linear threshold classifier with maximum margin. In binary classification settings, every single instance x has a corresponding label $y \in \{-1, 1\}$ indicating whether it belongs to a specific class or not. Any linear threshold classifier $w(x)$ corresponds to a vector $w \in \mathbb{R}^n$ such that the prediction of w for the instance x is $w(x) = \text{sign}\left(w^\top x\right)$. As discussed in Chapter 2 for the new examples x, the SVM classification is decided as $\text{sign}\left(\sum_{\mathrm{i}=1}^{\mathrm{M}} \alpha_{\mathrm{i}} \mathrm{y_i} \mathrm{x_i}^\top \mathrm{x}\right)$. SVM provides an efficient classification scheme based on maximizing the margin between training examples belonging to different classes. However, a majority of the training examples are involved in making the final decisions for the unseen (test) examples. On the other hand, principles of learning theory [3] suggest that simpler models in which a small, well-chosen subset of all training examples is only involved in making the final classification decisions is statistically more preferable. In this section we introduce a sparse approximation model that addresses the inherent issues of classic SVM, namely the sparsity-eager SVM. The sparse approximation framework , can be viewed as an alternative classification framework which aims to base the classification decision on a small number of well-chosen training examples. First, we discuss the sparse approximation techniques and then impose a sparsity assumption in SVM optimization.

The sparse approximation approach for classification is based on recent advances in using compressed sensing [7;148;149]. This approach relies on the assumption that for a sufficiently large number of training instances per class, any new instance lies on the subspace spanned by all training examples belonging to that class, and is therefore representable by a *sparse* linear combination of all

training examples.

Let k denote the number of classes, r denote the number of training examples per class, and let n show the dimension of the extracted feature space. Denote the extracted feature vector of the j^{th} training example in the i^{th} class as $x_{i,j} \in \mathbb{R}^n$. Again, without loss of generality, we assume that each feature vector $x_{i,j}$ is normalized, that is $\|x_{i,j}\|_2 = 1$. Then, assuming that there are sufficient training samples for each class, the i^{th} class has a corresponding sample repository X_i which is an $n \times r$ matrix, obtained from all r training examples belonging to class i. The *training matrix* is then the $n \times M$ matrix

$$X \doteq [X_1, \cdots, X_k], \tag{4.5}$$

with $M = kr$ denotes the total number of training examples. A vector x is s-sparse if it has at most s non-zero entries. The support of a s-sparse vector indicates the positions of its non-zero entries. The ℓ_0 pseudo-norm of a vector counts the number of its non-zero entries. In other words, a vector x is s-sparse if and only if $\|x\|_0 \leq s$.

It follows from the subspace model assumption that if X contains a sufficiently rich set of training examples for each class, then every new test example can be represented by a *sparse linear* combination of *all* training examples in the training matrix. More formally, let $f \in \mathbb{R}^n$ be a test example. The sparse approximation classifier first finds the solution $\hat{\alpha}$ of

$$\begin{aligned} &\text{minimize } \|\alpha'\|_0 \\ &\text{s.t. } \|X\alpha' - f\|_2 \leq \epsilon \end{aligned} \tag{4.6}$$

for a sufficiently small parameter ϵ. For each class i, let δ_i be the indicator function that selects the coefficients associated with the i^{th} class. The sparse-approximation classifier then outputs its prediction $\hat{y}$ as

$$\hat{y} \doteq \arg \min_{i \in \{1, \cdots, k\}} \|f - X_i \delta_i(\hat{\alpha})\|_2. \tag{4.7}$$

Unfortunately, solving the optimization problem of Equation 4.6 is non-convex, and in general

NP-hard [150]. The emerging theory of compressive sensing [151;152], states that in many practical applications, the solution of the convex optimization problem

$$\begin{aligned} &\text{minimize } \|\alpha'\|_1 \\ &\text{s.t. } \|X\alpha' - f\|_2 \leq \epsilon, \end{aligned} \tag{4.8}$$

known as the *Basis Pursuit* optimization coincides with the solution of Equation 4.6. Nevertheless, the optimization problem of Equation 4.8 is a second order cone optimization and requires $O(M^3)$ running time. Moreover, the ℓ_1-regression classifier is a lazy classifier, that is, the optimization of Equation 4.8 must be solved for *each* test example f independently. As a result, scalability is a fundamental issue with respect to this approach.

4.3.1 Imposing Sparsity

The ℓ_1-SVM classifier combines the ideas of the classical SVM with the sparse approximation techniques with the goal of (i) obtaining higher generalization accuracy on new (test) examples, (ii) increasing the robustness against overfitting to the training examples, and (iii) providing scalability in terms of the classification complexity. Given a set $\langle (x_1, y_1), \cdots, (x_M, y_M) \rangle$ of M training examples, we aim to find a vector $\alpha \in \mathbb{R}^M$ such that (i) α is sufficiently sparse, and (ii) the classifier $w \doteq \sum_{i=1}^{M} \alpha_i y_i x_i$ has a sufficiently low empirical loss and therefore sufficiently large separating margin.

Recall that the objective function of a regular (ℓ_2)-SVM can be rewritten as:

$$\hat{L}(w) \doteq \hat{H}_S(w) + \frac{1}{2C}\|w\|^2, \tag{4.9}$$

where C is the regularization constant. On the other hand, to avoid the curse of dimensionality and overfitting to the training examples, we wish to find a solution α which is as sparse as possible. The sparsity condition guarantees that only a small subset of training examples have non-zero weights and are involved in making a final classification decision. We can enforce this sparsity assumption

by replacing the maximal margin minimization with a new sparsity ℓ_0 condition on α. In this sense, we are replacing the requirement of minimizing $\sum_{i,j=1}^{M} \alpha_i \alpha_j y_i y_j {x_i}^\top x_j$, with the new objective of minimizing $\|\alpha\|_0$. Theorem 2 is then a direct consequence of this replacement.

Theorem 2. *Let $\langle (x_1, y_1), \cdots, (x_M, y_M) \rangle$ be a set of M labeled training examples, and α is the sparse classifier. The objective function for training a sparse linear threshold classifier can be obtained as a linear combination of training examples with a sparsity assumption as:*

$$\begin{aligned} &\textit{minimize } \|\alpha\|_0 + \frac{C}{M} \sum_{i=1}^{M} \xi_i \\ &\textit{subject to:} \\ &\qquad \forall\ i:\ \xi_i \geq 0 \\ &\qquad \forall\ i\ :\ \xi_i \geq 1 - y_i \sum_{j=1}^{M} \alpha_j y_j {x_j}^\top x_i \\ &\qquad \forall\ i:\ 0 \leq \alpha_i \leq \frac{C}{M}, \end{aligned} \tag{4.10}$$

which can be approximated and solved using fast gradient descent methods.

Proof. Similar to the optimization problem of Equation 4.6, the optimization problem of Theorem 2 is also intractable. To overcome this intractability issue, we replace the non-convex pseudo-norm $\|\alpha\|_0$ by the convex ℓ_1 norm $\|\alpha\|_1$ which is the closest convex norm to the ℓ_0 pseudo-norm [1]. Therefore, the optimization objective of the ℓ_1-SVM classifier of Theorem 2 now becomes:

[1] Relaxing the ℓ_0 with an ℓ_1 is a convex relaxation technique in sparse approximation techniques.

$$\text{minimize } \sum_{i=1}^{M} \alpha_i + \frac{C}{M} \sum_{i=1}^{M} \xi_i$$

subject to:

$$\begin{aligned} &\forall\ i:\ \xi_i \geq 0 \\ &\forall\ i\ :\xi_i \geq 1 - y_i \sum_{j=1}^{M} \alpha_j y_j {x_j}^\top x_i \\ &\forall\ i \in \{1, \cdots, M\}:\ 0 \leq \alpha_i \leq \frac{C}{M}. \end{aligned} \tag{4.11}$$

The optimization program of this equation 4.11 is a linear program with the dual objective:

$$\text{maximize } \sum_{i=1}^{M} \lambda_i + \frac{C}{M} \sum_{i=1}^{M} \theta_i$$

sbject to:

$$\begin{aligned} &\forall\ i:\ \theta_i \leq 0 \\ &\forall\ i\ :\ \theta_i \leq 1 - y_i \sum_{j=1}^{M} \lambda_j y_j {x_j}^\top x_i \\ &\forall\ i:\ 0 \leq \lambda_i \leq \frac{C}{M}. \end{aligned} \tag{4.12}$$

This linear program can be efficiently solved using fast gradient descent techniques [153]. Moreover, in contrast to the ℓ_1-regression framework, here we solve this optimization only once to learn the optimal α_i. The classification decision on a new test example x is:

$$\hat{y} \doteq \text{sign}\left(\sum_{i:\alpha_i \neq 0} \alpha_i y_i {x_i}^\top x\right). \tag{4.13}$$

The ℓ_1-SVM is now operating with a sparsity assumption that enforces only a small subset of training examples to be involved in making a final classification decision for unseen examples. In Section 4.4 shows that this classifier can outperform the classic ℓ_2-SVM and is comparable to the state-of-the-art DNNs in terms of average classification accuracy rate on public datasets.

Table 4.1: Number of songs per genre.

Genre	Number of Samples
alternative	145
blues	120
electronic	113
folkcountry	222
funksoul/rnb	47
jazz	319
pop	116
raphiphop	300
rock	504

4.4 Empirical Evaluation

In this section we present an extensive evaluation of the classification methods of Chapter 4 on a public dataset for a music classification task. Specifically, we compare the results of a two hidden layered deep neural network architecture of Section 4.2 for genre classification with the ℓ_1-SVM and classic SVM introduced in Section 4.3 and Chapter 2, respectively. We will report the classification accuracy rate(%) to draw a comparison between DNNs, SVMs and baseline classifiers for music genre classification. Our efforts are focused on two main questions: (i) can hand-crafted (engineered) features in combination with ℓ_1-SVM outperform the classic ℓ_2-SVM and state-of-the-art DNNs? and, (ii) is the ℓ_1-SVM superior to classic baseline classifiers in terms of time complexity? In our experiments we perform a 10-fold cross validation and report the average classification accuracy rate with repeated trials across all experiments.

4.4.1 Dataset

We evaluate our experimental results on the benchmark dataset for audio classification and clustering proposed by Homburg et al. [125] as described in Chapter 3. This dataset contains samples of 1886 songs obtained from the Garageband site and is comprised of 9 music genres including Pop, Rock, Folk/Country, Alternative, Jazz, Electronic, Blues, Rap/HipHop, and Funk/Soul. Table 4.1 shows the number of examples in each genre. Once again we utilize the 10 seconds recordings of audio samples drawn in random from each song, in mp3 format, sampled with a sampling rate of 44100Hz

and a bit-rate of 128 mbit/s. The 49-dimensional TSPS features of the dataset [126] and MFCCs are extracted as the hand-crafted feature representation of the dataset. The input to the ℓ_1-SVM and baseline classifier is fixed as MFCC feature vectors to measure the impact of the learning method on classification accuracy. The DNNs are mainly designed as a comparison with these hand-crafted features in combination with novel learning methods. The raw audio signals are sampled and used as the input for DNNs. In the following experiments we report the average classification accuracy rate across all experiments and provide evidence to support our hypotheses. First we explain the preprocessing of input data for all classifiers and then present the results.

4.4.2 Data Preprocessing

The input data to DNNs is a set of unprocessed (raw) audio signals in wav format. Figure 4.3 illustrates an example of the input signal for the *alternative, pop and rock* genre classes in frequency domain and the power spectrogram from the sample dataset and Figure 4.4 shows the examples in *blues, electronic, folk/country, jazz, rap/hiphop* genres. Our proposed method with DNNs involves five successive sequences: preprocessing of audio signals to normalize and augment input data [2],

[2]Data augmentation to address data imbalances or removing small classes is part of preprocessing.

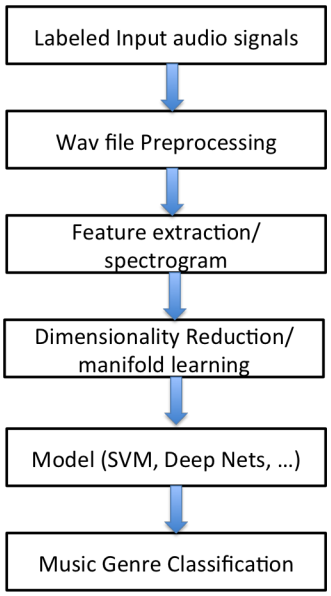


Figure 4.2: The music classification method is illustrated.

feature extraction or spectrogram representation, manifold learning and dimensionality reduction techniques, classification model such as SVMs or DNNs and genre label prediction of unseen (test) examples.

The preprocessing and representation of raw input audio signals vary among different experiments. The preprocessing steps on raw wav files can be divided into three main steps: (i) data manipulation (augmentation, normalization, subsampling, etc.), (ii) power spectrogram representation and, (iii) feature extraction. The first step in preprocessing includes subsampling all input signals by resizing to the minimum sample size across all datasets [3]. In these experiments, we have avoided zero-padding for noise reduction and better accuracy rate for neural networks. The funksoul/rnb has been excluded from genre classes due to small sample size without data augmentation. The DNNs first utilize the raw audio features with resizing and the power spectrogram as a set of input. We have also experimented with wavelet filtering of resized audio signals along with DNNs to boost the classification accuracy. The SVMs and regression based classifiers use hand-crafted features (MFCCs) to enhance the classification accuracy rate.

The process of constructing the MFCCs begins by applying a FFT to fixed size overlapping windows (Hann or Hamming windows). A series of transformations combining an auditory filter bank with a cosine transform will result in a discrete representation of each audio window in terms of MFCC descriptors. The filter bank is constructed using 13 linearly-spaced filters followed by 27 log-spaced filters[74]. A short-time audio window is then represented as an MFCC feature vector composed of 13 cepstral coefficients. The audio signal is then represented as a $k \times 13$ feature vector where k is the number of feature vectors for each song and has been set experimentally at $k = 500$. For N total examples from the dataset, the matrix $M_2 \in R^N \times (k \times 13)$ is the input for SVMs and regression based classifiers.

A spectrogram is a visual representation of the spectrum of the input signal frequencies as they vary with time or some other variable and can be used to identify spoken words phonetically, and to analyze the various calls of animals. Spectrograms are often created in one of two ways:

[3]The sampling and normalization of sample sizes is required in data preprocessing when different examples have different dimensionality. In this sense all input examples are often subsampled to a uniform size or up-sampled via zero-padding and then normalized.

Table 4.2: Classification accuracy (%) on sample dataset is compared across various methods.

Preprocessing	Augmentation	DR-method	Model	Average accuracy rate(%)
Raw wav	No	PCA (270d)	2-DNNs(400 neurons)	18.55%
Raw wav	No	SLPP	2-DNNs(400 neurons)	**17.03**%
Raw wav	No	SR	2-DNNs(400 neurons)	17.46%
Spectrogram	No	None	2-DNNs(400 neurons)	36.24%
Spectrogram	No	PCA (270d)	2-DNNs(400 neurons)	37.11%
Spectrogram	No	PCA (400d)	2-DNNs(400 neurons)	37.33%
Wavelet	No	PCA(400d)	2-DNNs(400 neurons)	**38.84**%
MFCC	RM-Uniform	None	ℓ_2-SVM	32.90%
MFCC	RM-Uniform	None	logistic regression	34.43%
MFCC	RM-Uniform	None	ℓ_1-SVM	37.43%
MFCC	RM-Uniform	None	ℓ_1-regression	30.45%

approximated as a filter-bank resulting from a series of bandpass filters (this was the only way before the advent of modern digital signal processing), or calculated from the time signal using the fast Fourier transform (FFT, DFT, STFT). We use the FFT to generate the power spectrogram of input audio signal $x(t)$. The spectrogram of $x(t)$ can be estimated by computing the squared magnitude of the STFT of the signal $x(t)$, as follows:

$$spectrogram(w,t) = \|STFT(t,w)\|^2$$

where t denotes the time axis of $x(t)$. The resulting spectrogram on resized audio files has been unrolled as an input matrix $M_1 \in R^{N\times L}$ for the DNNs where N is the number of total examples in the dataset and L is the dimensionality of unrolled power spectrogram for each signal.

4.4.3 Results

In this section we explain the results of our comparative experiments for music genre classification. Throughout our experiments we exclude the funksoul/rnb genre due to small sample size without data augmentation and duplication. The first round of experiments are performed on raw audio signals using a two hidden layer neural network. All audio samples have been normalized to the

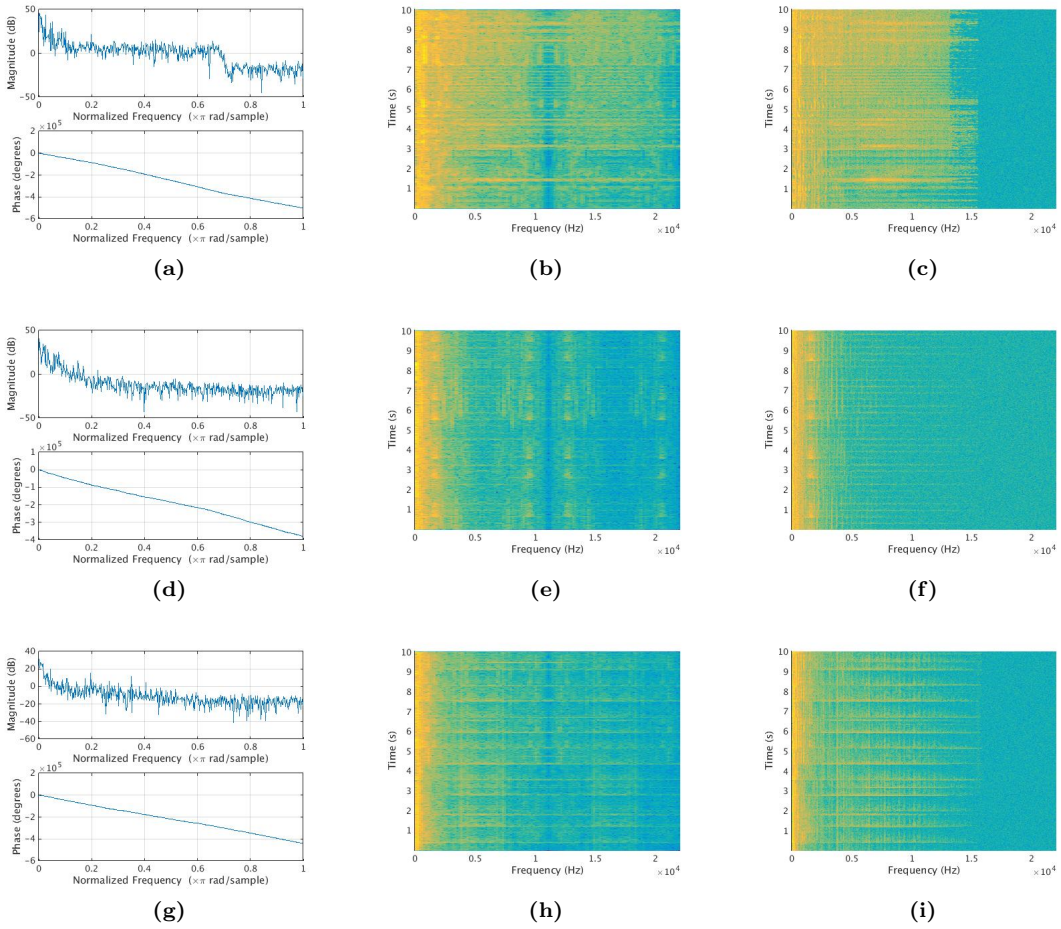


Figure 4.3: A sample audio input from class I, II and III(*alternative, pop and rock*) genre is illustrated as (a,d,g) original input signal in the frequency domain by magnitude and phase, (b,e,h) frequency spectrum and (c,f,i) power spectrogram.

least example size in the dataset (denoted by RM-resize) [4]. Rows 2 to 5 in Table 4.2 shows the genre classification results on this dataset via a two-layer deep neural network (2-DNN) with 400 neurons. The activation function for the DNNs is the ReLU function. A round of manifold learning has been applied as the dimensionality reduction step with PCA, SR and SLPP. We can observe that DNNs on the raw input audio signal perform poorly even with manifold learning. This motivates our next round of experiments with audio preprocessing to boost the performance of the DNNs [5].

[4]Zero padding can introduce additional noise and consequently decrease the accuracy of the classifiers. Hence the normalization step only resizes all samples to minimum size across all dataset points.

[5]MFCC trials with DNN did not boost the accuracy and hence are not selected as an input to this classifier. Instead, we focus on an automatic feature extraction in DNNs as a comparison to hand-crafted MFCCs.

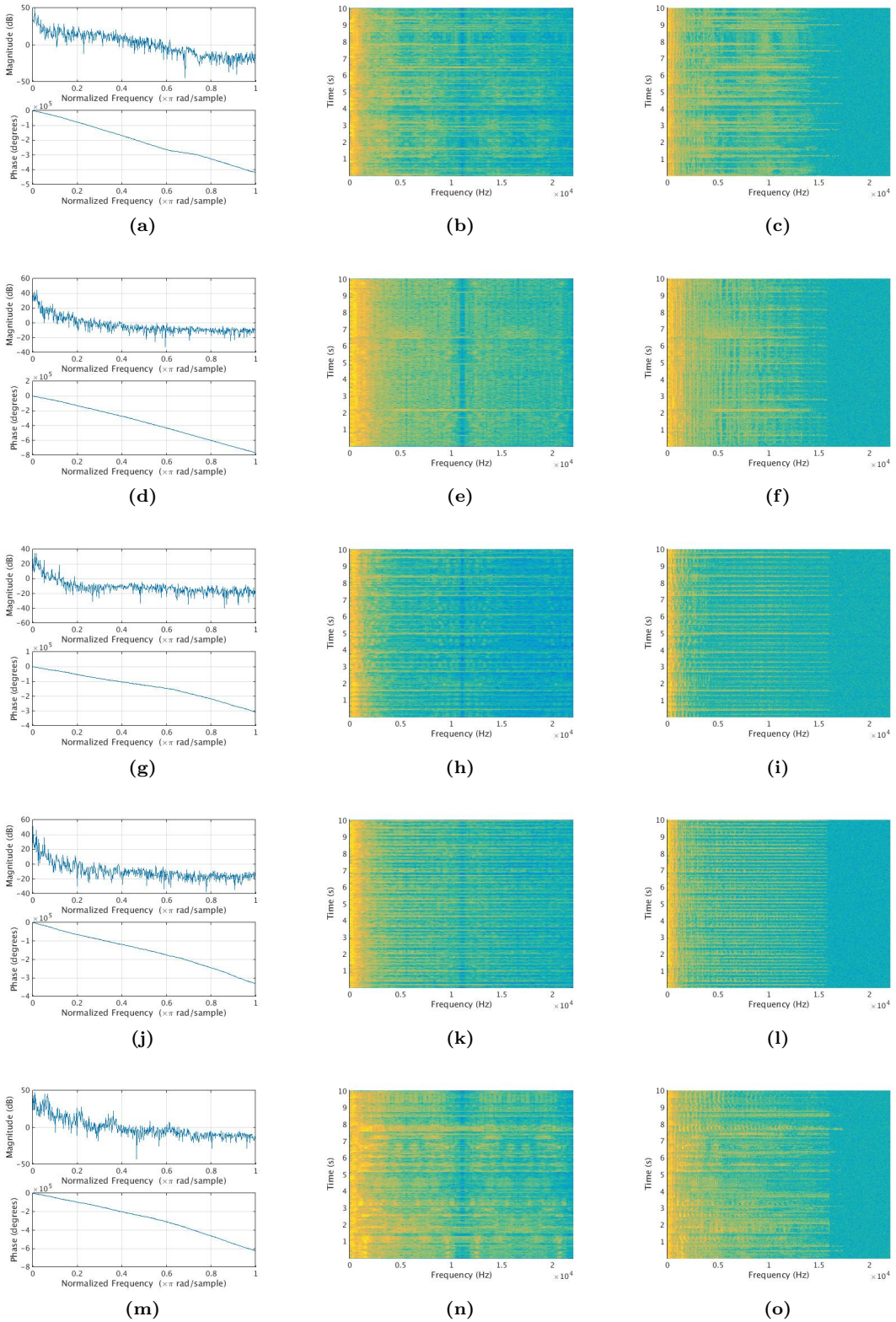


Figure 4.4: A sample audio input from *blues, electronic, folk/country, jazz, rap/hiphop* genre is illustrated as (left) original input signal in the frequency domain by magnitude and phase, (middle) frequency spectrum and (right) power spectrogram.

The next round of experiments explore the preprocessing of input audio signals to boost the genre classification accuracy on the dataset. The results of this set of experiments are illustrated in Table 4.2, rows 5 to 8. The input wav files have again been resized by keeping the minimum number of columns across all examples in the dataset. The resized columns are then represented as the audio power spectrogram. The spectrogram has been derived by a 1024 DFT using a Hanning window at 44100 Hz. Rows 5, 6 and 7 show the genre classification accuracy rate on the preprocessed dataset using a two-layer neural network with 400 neurons. PCA with 270 and 400 principal components have been selected as the dimensionality reduction method. We can observe that the power spectrogram representation of raw input audio files can boost the classification accuracy in combination with dimensionality reduction methods. We also experiment with the power spectrogram on wavelet representation of input signals using a kaiser window as shown in row 8 of Table 4.2. The wavelet power spectrogram representation with PCA and a 2-DNN can marginally boost the accuracy rate for genre classification on the dataset.

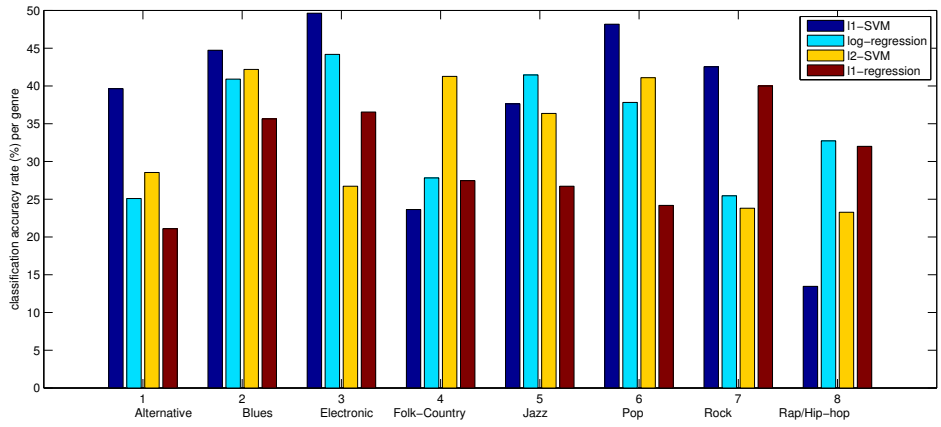


Figure 4.5: True positive genre classification rate is illustrated for each music genre.

Next, we compare the classification accuracy rates to hand-crafted feature extraction in combination with ℓ_1-regression[7], logistic regression[8], SVM optimization[9] and the proposed ℓ_1-SVM. The MFCC feature vectors has been selected as the audio representation where each MFCC vector is 13 dimensional. Each audio sample has been represented by 500 MFCC feature vectors selected at random. Rows 9 to 12 from Table 4.2 show the results of this set of experiments on the dataset.

Figure 4.5 illustrates the true positive genre classification rate per each genre for ℓ_1-regression[7], logistic regression[8], SVM optimization[9] and the proposed ℓ_1-SVM.

The suitability or performance of classifiers using different learning schemes are often tested using different number of training samples and varying training time. Figure 4.6 illustrates the classification accuracy rates of 50 rounds of independent experiments using different number of training samples. We note that the ℓ_1-SVM outperforms ℓ_2-SVM, logistic regression and ℓ_1-regression given the same number of training samples.

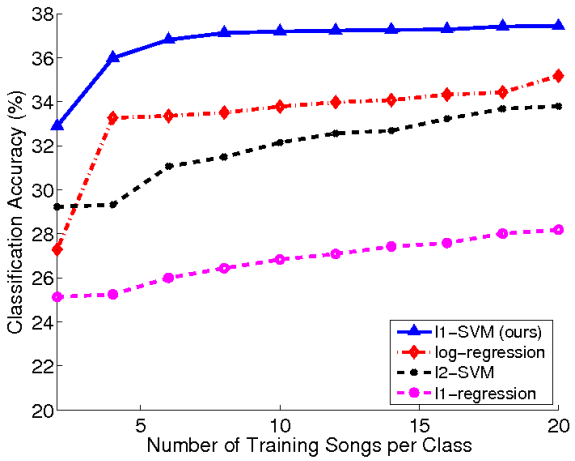


Figure 4.6: ℓ_1-regression[7], logistic regression[8], ℓ_2-SVM optimization[9] and ℓ_1-SVM classification accuracy rates are illustrated using different number of training samples. The number of training samples have been limited to 20 samples to reduce the convergence time of the classifier. There is no deviation in the classification accuracy rate for a larger training set.

Besides the classification accuracy rate, the average training time is a second important factor that affects the suitability of a classification algorithm. An algorithm with a slightly lower accuracy might be preferred if its training time is significantly lower. Additionally, an estimation of the required training time for a genre classification task is very useful if the result has to be available in a certain amount of time. In this experiment we use 20 random training songs to train our classifiers. The ℓ_1-regression method is a lazy classifier without a training phase and thus is excluded from this experiment. The DNNs are not in the same classification methods and are excluded in these experiments as well. Figure 4.7 shows the average classification accuracy rate of ℓ_2-SVM, logistic regression and ℓ_1-SVM on the corpus of our data set against the average training time

for each classifier. We note that the ℓ_1-SVM outperforms ℓ_2-SVM and logistic regression once the convergence threshold is set properly.

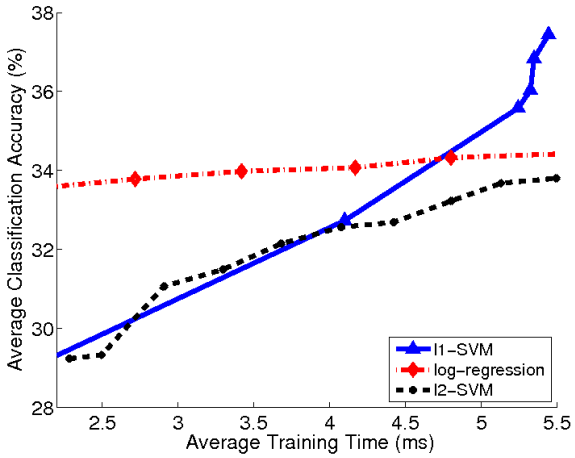


Figure 4.7: Average classification accuracy rate is reported for different average training time.

Our experiments indicate that the ℓ_1-SVM achieves comparable accuracy rate with DNNs on genre classification. It should be noted that the input features to SVMs are set as the MFCCs which are sub-optimal for classification. We anticipate that a more comprehensive feature vector in combination with the proposed ℓ_1-SVM can outperform the DNNs for this classification task while providing lower time-complexity.

4.5 Conclusions

In this Chapter we presented a set of classifiers. Our contributions have been threefold. First, we introduced a sparsity-eager SVM that combines the ideas of ℓ_1-regression with SVM to address the inherent shortcomings of the original SVM formulation. Second, we proposed a DNN architecture for a classification task as a baseline. Third, through a set experiments we demonstrated the utility of proposed models in classification. Our efforts have been focused on exploring the efficiency of hand-crafted features in combination with a supervised learning model and its comparison with neural networks as a widely popular classifier.In Section 4.2 we explored a deep neural network architecture for classification. We examined manifold learning techniques as an accuracy boosting method for classifiers in Section 4.2.1. We introduced the ℓ_1-SVM classifier in Section 4.3 to address

the inherent issues of classic SVM.

Finally, we compared the classification performance of the proposed classifier with state-of-the-art classifiers in Section 4.4 on a public dataset for a music classification task. We compared the results using raw audio data and power spectrogram of input signals with a two-layer neural network. We explored different manifold learning techniques such as PCA, SLPP and SR as dimensionality reduction methods. We then compared the classification results using hand-crafted audio feature vectors such as MFCCs. Our experimental results indicate that MFCCs in combination with ℓ_1-SVM classifiers are comparable with deep neural networks on power spectrograms for music genre classification.

In the future, we intend to study other network architectures such as convolutional neural networks and stacked auto-encoders for music classification. We will explore different signal preprocessing and representations to measure the network sensitivity for classification. This can provide more insight into our original question for this chapter: Can hand-crafted features in combination with SVMs outperform neural networks for classification?

Chapter 5: Cross-Modal Information Retrieval

In this chapter we verify two main hypotheses: (i) can we develop a multimodal concept space (using ESA embeddings) for cross-modal information retrieval? (ii) can we employ *canonical correlation analysis* (CCA) to allow for communication between the two input modalities? The main computational hypothesis is that the developed cross-modal retrieval system can significantly boost the accuracy compared to baseline retrieval models.

Information Retrieval (IR) was originally popularized in the context of enabling computers to find relevant information using the text modality[81;82]. In this sense, the IR systems were first used to find the top relevant text documents that match a user query for example search strings in web search engines. The IR system computes a numeric score on how well each text document matches the user issued query and ranks the database documents according to this relevancy score. The top ranking documents are then returned to the user.

In Chapter 3 we reviewed the basics of IR and it's applications. IR models often transform the documents into a suitable representation such as a mathematical model or a vector of features. In Section 3.2 we explored the *standard Boolean* and the *vector space* models as the two widely used mathematical and algebraic models for IR. We also examined the tf-idf as the term weights in the classic vector space model of Salton *et. al*[81]. The tf-idf term weights were originally proposed for text modality. In Section 3.5 we introduced an extension of the vector space model and tf-idf to time-series signals such as audio data.

The rapid increasing of the multimedia data such as images, audio and video data brings new challenges to IR systems[154]. Accessibility of various sensors for the crowds results in massive multimodal datasets. Users are frequently equipped with cameras, cell phones, voice recording devices and various other means of multimodal data collection. This multimodal data explosion creates two major challenges for traditional IR systems; first text based IR systems are required to handle other data modalities besides text information where the user query and the database documents

can be any object such as images. Second, the IR systems need to be extended to cross-modal data retrieval platforms where a qury from one modality can be retrieved in another modality. A good example of the latter problem is an image retrieval model where the user queries are strings while the database objects are images. These challenges call for multimodal and cross-modal IR systems that can explain all data modalities in a shared semantic space. This semantic space is used to discover the underlying shared *concepts* among different data modalities. The development of the concept space enables IR systems to navigate between multiple modalities.

In this chapter and Chapter 6, we dive deeper into the inner workings of cross-modal and multimodal IR and classification systems respectively. In Section 5.1 we review the popular approaches to build joint models for fusing various data modalities in IR. In Section 5.2 we explore the cross-modal methods that are used to retrieve objects in different modality than the query space. In Section 5.3 we review *canonical correlation analysis* (CCA) as way of measuring the linear relationship between two multidimensional variables. In Section 5.4 we introduce ESA embeddings in combination with CCA as a cross-modal IR framework. We evaluate the performance of the proposed model on a public dataset in Section 5.5. Finally, we conclude this chapter in Section 5.6 and propose future research directions in this active research area.

5.1 Background

The problem of cross-modal retrieval is becoming more pronounced as multiple sensory sources of data are becoming accessible for the crowds. Three main challenges are introduced to IR systems; first, unimodal IR systems are required to span other modalities beyond text. Second, multimodal IR models are required to handle multimodal datasets. Third cross-modal methods are required. Cross-modal IR systems often use a query from one data modality to retrieve dataset objects from a different modality. Finding text captions for images can be a good example of such systems.

In unimodal IR methods the text databases are searched with text queries, image databases with image queries and so on[155]. The first challenge in classical IR systems is the extension of the existing models to other data modalities besides text information[156] such as audio, images and videos. Various models have been proposed in the literature of IR to address such extension[156–159]. Content-

based retrieval [157–161], annotations[162–166], labeling[162;167–171] and semantic space modeling[172–174] are among the most popular methods to extend the text-based retrieval models.

Annotation retrieval models[162–166] are based on augmented datasets. A successful retrieval in this context requires the datasets to be augmented with text metadata provided by human annotators[155]. The annotations are often provided in terms of tags, keywords or a small text description of the data source. The retrieval system then reduces the problem to matching the input text metadata with database metadata of the non-text objects. These retrieval models often ignore the data sources besides text metadata. Annotation-based retrieval is labor-intensive and expensive. Moreover, these retrieval models are not scalable and expendable to raw datasets.

Content-based retrieval models[157–161] search and analyze the contents of the data sources rather than the associated metadata such as keywords, tags, labels and descriptions. The *content* in this sense refers to any information that can be derived from the data sources. Color, shapes, texture descriptors for images and signal-based properties such as harmony and timbre from audio data are good examples of *content* information that can be used in these models. In Chapter 3 we explored a set of feature vectors for various data sources that can be utilized in IR systems. Content-based retrieval does not rely on metadata and human annotations. This makes the IR systems scalable and more desirable in comparison with annotation retrieval methods.

Labeling retrieval models often assume that the dataset objects can be segmented and described by a small word vocabulary. In this sense, learning a joint probability model that correlates these data segments and vocabulary words becomes the focus of the retrieval model. In Chapter 3 we saw an example of developing an audio vocabulary to characterize a music dataset using the ESA model. Latent Dirichlet allocation (LDA) models[175], probabilistic latent semantic analysis (LSA)[167] ,histograms[162] and a combination of Bernoulli distributions[155;168;169] have been proposed in the literature for labeling-based IR. Gaussian mixtures and hidden Markov models[170;171] have also been used to train a word vocabulary or a concept space in a weakly supervised manner. This word vocabulary is later utilized for image retrieval.

In a semantic space retrieval model, the data sources are presented as a weighted sum of *concepts*.

The concepts are a set of predefined vocabulary words. These algorithms first learn a distribution of low-level features for each concept. Query and database objects are then projected into this semantic space by the similarity between their low-level distribution to the concept representations [173]. Once the query and the database objects are presented in the semantic space the IR and classification tasks can be performed using the classical IR models. In Chapter 3, we examined an extension of the vector space model to project signals in a latent semantic space for a classification task. The similar representation can be used for IR as shown in this Chapter. Semantic spaces provide a higher level of abstraction than the low-level signal representation. This abstraction allows for similarity assessment through a shared concept space which can increase the quality of retrieval [174] and boost the classification accuracy [172].

The second challenge in IR models is the extension of existing unimodal approaches to multimodal retrieval. The unimodal IR paradigm is of limited use in the modern information landscape, where multimedia content is ubiquitous. Due to this, multimodal modeling, representation, and retrieval have been extensively studied in the multimedia literature [158;176–181]. The key idea behind these models is the integration of information from all data channels (modalities) into a single representation. this representation can then be combined with the unimodal IR methods to achieve a multimodal retrieval model. The fusion of data from various channels can be in early and late stages. Early fusion models blend the feature representations from different modalities into a single vector [182;183]. Late fusion models build an IR system by learning different models for different modalities and fusing their predictions [179;180;184]. In Chapter 6 we dive deeper into multimodal IR and classification models and propose two new models for multimodal classification.

The third challenge of IR on modern datasets is the cross-modal retrieval. In cross-modal settings information from one data modality is used to retrieve relevant documents in the database which are presented as a different modality than the query. A good example of such systems are image retrieval using text queries. While multimodal IR models can theoretically be used for cross-modal retrieval, the inability to access each data modality individually (after the fusion of modalities) presents some new challenges. In Section 5.2 we take a closer look at cross-modal retrieval methods.

We then propose a new framework for retrieval in cross-modal settings in Section 5.4.

5.2 Cross-Modal Methods

Cross-modal retrieval models have become increasingly important with the explosion of multimodal datasets. These models are focused on using a query from one modality to retrieve a ranked set of objects from another modality in the database. Several models have been proposed in the literature of IR community for images and text[177;185;186], images and audio[187;188], audio, images, text and fMRI data sources[189–196]. Nevertheless, all these models are focused on learning a manifold, correlation or a shared concept space between different information modalities.

Manifold learning techniques are one of the most popular approaches for cross-modal retrieval[192–194]. In the manifold learning approaches[197], the model learns a matrix $M(X)$ of distances between multiple training modalities $X = X_1, \ldots, X_i$ where $dim(M(X)) \ll dim(X)$. Once a query Q is issued the retrieval system simply projects the query into the learnt manifold. The projected query is then mapped to the closet database object on the manifold $M(X)$. The database object can be used to retrieve the top ranked set of documents in the other modality besides the original query modality. Algorithm 3 shows the retrieval process once a manifold $M(X)$ is learnt.

Definition 5.2.1. Cross-modal Retrieval: Given an input query Q with a modality j and a set of training examples $X = X_1, \ldots, X_i, \ldots, X_N$ with modalities $i \in 1, \ldots, N$, the cross-modal retrieval algorithm finds a ranked subset of documents $\hat{X} \subset X$ that are in other modalities besides the original query space.

Algorithm: MANIFOLDRETRIEVAL$(X, M(X), Q)$

Input: X: TRAINING INPUT , $M(X)$: MANIFOLD, Q: SET OF QUERIES
Result: $\hat{X}$: RETRIEVED CROSS-MODAL DATA FOR QUERY Q

$PROJECTED_Q \leftarrow NN(Q) \in X$;

$\hat{X} \leftarrow RANKED(PROJECTED_Q) \in X$;

return $\hat{X}$

Algorithm 3: The retrieval operation in a manifold learning model of cross-modal retrieval.

The main limitations of the manifold learning models is the lack of generalization. Intuitively, the only efficient way to project queries into the learnt manifold is by using the training dataset. The unseen queries are first required to be approximated by the nearest neighbors from the training set and then projected into the learnt manifold. In Section 5.5.2 we explore a manifold learning technique as a baseline for a retrieval task on a public dataset.

Another popular model for cross-modal retrieval is learning the correlations between the input modalities through canonical correlation analysis(CCA) [186;187] or cross-modal factor analysis (CFA) [195]. Factor analysis is a statistical method used to describe variability between a set of correlated observations in terms of lower dimensional factors. The observed variables are modeled as linear combinations of the factors and an error term. The information gained about the learnt factors can be used later to reduce the set of variables in a dataset. Similar to CFA, CCA can be used as a way to learn the correlations between multimodal observed variables. In these cross-modal retrieval models CCA and CFA are utilized as joint dimensionality reduction methods that extract highly correlated features across input modalities. The correlation weights are then used to project a query into the other modality where retrieval is possible.

5.3 Cross-Modal CCA

Canonical correlation analysis (CCA) is a way of measuring the linear relationship between two multidimensional variables and making sense of cross-covariance matrices [198]. CCA can be seen as the problem of finding basis vectors for two sets of variables such that the correlation between the projections of the observed variables onto these basis vectors is mutually maximized [198]. Correlation analysis is dependent on the coordinate system in which the variables are described, so even if there is a very strong linear relationship between two sets of multidimensional variables, depending on the coordinate system used, this relationship might not be visible as a correlation. In this case, CCA seeks a pair of linear transformations, one for each of the sets of observed variables, such that when the set of variables is transformed, the corresponding coordinates are maximally correlated.

Definition 5.3.1. CCA Formulation: Given the two input vectors $X = (x_1, \ldots, x_n)$ and $Y = (y_1, \ldots, y_m)$ of random correlated variables CCA will find linear combinations of the x_i and y_j which are maximally correlated [198].

Definition 5.3.2. Cross-covariance: Given two column vectors $X = (x_1, \ldots, x_n)'$ and $Y = (y_1, \ldots, y_m)'$ of random variables with finite second moments, the cross-covariance $\Sigma_{XY} = \mathrm{cov}(X, Y)$ is defined as the $n \times m$ matrix whose (i, j) entry is the covariance $\mathrm{cov}(x_i, y_j)$.

CCA seeks vectors a and b such that the random variables $w_x = a'X$ and $w_y = b'Y$ maximize the correlation $\rho = \mathrm{corr}(a'X, b'Y)$. The random variables $w_x = a'X$ and $w_y = b'Y$ are the first pair of canonical variables or canonical variates. This procedure may be continued up to $\min\{m, n\}$ times to find independent pairs of canonical variates.

Theorem 3. [198] *Let $\Sigma_{XX} = \mathrm{cov}(X, X)$ and $\Sigma_{YY} = \mathrm{cov}(Y, Y)$ be the covariance matrices. The canonical correlation is then formulated as:*

$$\rho = \frac{a'\Sigma_{XY}b}{\sqrt{a'\Sigma_{XX}a}\sqrt{b'\Sigma_{YY}b}} \tag{5.1}$$

The canonical variates are then derived as $w_x = c'\Sigma_{XX}^{-1/2}X = a'X$ and $w_y = d'\Sigma_{YY}^{-1/2}Y = b'Y$.

Proof. We define $c = \Sigma_{XX}^{1/2}a$ and $d = \Sigma_{YY}^{1/2}b$ as a change of basis. Replacing this change in Equation 5.1, the optimization of CCA becomes:

$$\rho = \frac{c'\Sigma_{XX}^{-1/2}\Sigma_{XY}\Sigma_{YY}^{-1/2}d}{\sqrt{c'c}\sqrt{d'd}}. \tag{5.2}$$

By using the Cauchy-Schwarz inequality [199], we have:

$$\left(c'\Sigma_{XX}^{-1/2}\Sigma_{XY}\Sigma_{YY}^{-1/2}\right)d \leq \left(c'\Sigma_{XX}^{-1/2}\Sigma_{XY}\Sigma_{YY}^{-1/2}\Sigma_{YY}^{-1/2}\Sigma_{YX}\Sigma_{XX}^{-1/2}c\right)^{1/2}(d'd)^{1/2} \tag{5.3}$$

and

$$\rho \leq \frac{\left(c'\Sigma_{XX}^{-1/2}\Sigma_{XY}\Sigma_{YY}^{-1}\Sigma_{YX}\Sigma_{XX}^{-1/2}c\right)^{1/2}}{(c'c)^{1/2}}. \tag{5.4}$$

If vectors $d = \Sigma_{YY}^{1/2} b$ and $\Sigma_{YY}^{-1/2}\Sigma_{YX}\Sigma_{XX}^{-1/2} c$ are collinear, the equality in Eqation 5.3 is obtained. In addition, the maximum of correlation is attained if c is an eigenvector of $\Sigma_{XX}^{-1/2}\Sigma_{XY}\Sigma_{YY}^{-1}\Sigma_{YX}\Sigma_{XX}^{-1/2}$ and d is proportional to $\Sigma_{YY}^{-1/2}\Sigma_{YX}\Sigma_{XX}^{-1/2} c$ [1]. If we reverse the change of coordinates, we have that a is an eigenvector of $\Sigma_{XX}^{-1}\Sigma_{XY}\Sigma_{YY}^{-1}\Sigma_{YX}$, b is an eigenvector of $\Sigma_{YY}^{-1}\Sigma_{YX}\Sigma_{XX}^{-1}\Sigma_{XY}$. Moreover, a is proportional to $\Sigma_{XX}^{-1}\Sigma_{XY} b$ and b is proportional to $\Sigma_{YY}^{-1}\Sigma_{YX} a$. The canonical variables are then defined by:

$$w_x = c'\Sigma_{XX}^{-1/2} X = a'X, \quad w_y = d'\Sigma_{YY}^{-1/2} Y = b'Y. \tag{5.5}$$

The subsequent pairs are found by using eigenvalues of decreasing magnitudes. Orthogonality is guaranteed by the symmetry of the correlation matrices [198].

CCA can be used to produce a model which relates two sets of variables. Constraint restrictions can be imposed on such a model to ensure it reflects theoretical requirements or intuitively obvious conditions. This type of model is known as a *maximum correlation model*. In this Chapter, we combine the maximum correlation model with the ESA embeddings of Chapter 3 to introduce a cross-modal retrieval algorithm.

5.4 ESA Retrieval

In Section 3.5 we introduced a generalization of the ESA model that will allow for the encoding of low-level time-series features. In this Chapter we introduce a CCA-based retrieval using the ESA embeddings. Once again, $\mathcal{D} = \{\delta_1, ..., \delta_k\}$ denotes our code-book of keywords, $\mathcal{C} = \{C_1, ..., C_M\}$ denotes the concept space, and $\mathcal{E} = \mathcal{E}_{\mathcal{C},\mathcal{D}}$ is the $M \times k$ matrix as defined in Section 3.5. We write $\mathbf{0}$ for the vector of all zeroes of the appropriate dimension. For a code-book word $\delta \in \mathcal{D}$ we use $\mathcal{E}(\delta)$ to denote its ESA encoding, i.e., to represent the column of the ESA matrix associated with word δ. For a given observed instance X belonging to the first modality, we use the notation $F(X)$ to denote the set $\{f_1, ..., f_\ell\}$ of its unique feature vectors. We use the ESAEncoding in Algorithm 2 to compute the ESA representation for observed instance X.

[1] By changing the order of c and d we can derive the similar equation where d is an eigenvector of $\Sigma_{YY}^{-1/2}\Sigma_{YX}\Sigma_{XX}^{-1}\Sigma_{XY}\Sigma_{YY}^{-1/2}$.

Algorithm: CCARETRIEVAL($\mathcal{E}(X), \mathcal{E}(Y), Q$)

Input: $\mathcal{E}(X)$, $\mathcal{E}(Y)$: ESA MATRICES, Q: SET OF QUERIES **Result**: $\hat{Y}$: RETRIEVED DOCUMENTS FOR QUERY Q

$\hat{Y} \leftarrow \mathbf{0}$;

$w_Y, w_X \leftarrow max_{w_X, w_Y} \frac{w_X^T C_{XT} w_Y}{\sqrt{w_X^T C_{XX} w_X w_Y^T C_{YY} w_Y}}$;

$P(Y) \leftarrow \mathcal{E}(Y) \times w_Y$;

$P(X) \leftarrow \mathcal{E}(X) \times w_X$;

$\hat{Y} \leftarrow w_Y \times P_Y^T \times P_X \times w_X^T \times Q^T$;

return T

Algorithm 4: Canonical correlation retrieval based on ESA representations of both modalities for input queries Q.

Similar to the representation of instances in the first modality, the instances from the second modality can also be described as tf-idf weight vectors in the shared concept space. Let $\mathcal{E}(X)$ and $\mathcal{E}(Y)$ denote a pair of training multimodal ESA matrices where each row corresponds to a shared concept between the two sets. X and Y denote the presentations of the training set in each data modality such as audio and text or images and audio. CCA can be applied to learn the shared semantic space between these two matrices. The original training matrices are projected into shared semantic concept space. Each query from the query set Q represented in the feature space is also projected into the same space using the estimated canonical variates from the training set. The Euclidean distance between the query and the projected $\mathcal{E}(Y)$ is then measured to obtain the closest matching database objects that best represent the query in the other modality. Let w_Y and w_X denote the projection matrices for $\mathcal{E}(Y)$ and $\mathcal{E}(X)$, respectively. The canonical variates, P_X and P_Y, are the projection of the training matrices into the shared semantic space where the two sets of variables are maximally correlated. Algorithm 4 illustrates the process of retrieving the top correlated database objects for a set queries using CCA. In Section 5.5 we use the proposed model for a retrieval task in a multimodal public dataset.

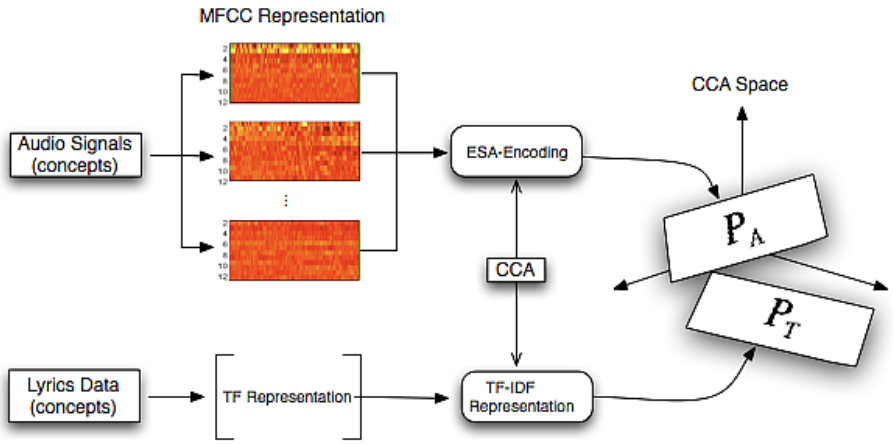


Figure 5.1: The observed variables in the audio dataset are represented using **MFCC** feature vectors. These feature vectors are quantized and represented in the shared concept (shared by textual metadata formed from tf-idf encoding of song lyrics). Canonical correlation analysis is then used to project the audio and text representation onto canonical variates P_A and P_T respectively, where the projections are maximally correlated.

5.5 Empirical Evaluation

We compared the performance of the modified ESA in cross-modal retrieval in combination with CCA on a public dataset. In these experiments our main goal is to retrieve corresponding textual representation for an audio query. As a baseline we compare the CCA retrieval to ESA-based retrieval of related database objects. Our experimental results show that ESA in combination with CCA can be used for cross-modal retrieval and outperforms simple ESA-based retrieval models. Section 5.5.1 explains the details of the dataset. We present the experimental results in Section 5.5.2.

5.5.1 Dataset

In our experiments we utilize the *Million Song Dataset (MSD)*[200] containing audio features and metadata for a million contemporary popular music tracks. The MSD dataset is a large scale freely-available dataset of multimodal nature. The MSD dataset contains 280 GB of data, $1,000,000$ songs, $44,745$ unique artists,$7,643$ unique terms (Echo Nest tags [2]), $2,321$ unique musicbrainz [3] tags, $43,943$ artists with at least one term, $2,201,916$ asymmetric similarity relationships between songs and $515,576$ dated tracks starting from 1922.

[2] `http://the.echonest.com/`
[3] `https://musicbrainz.org/`

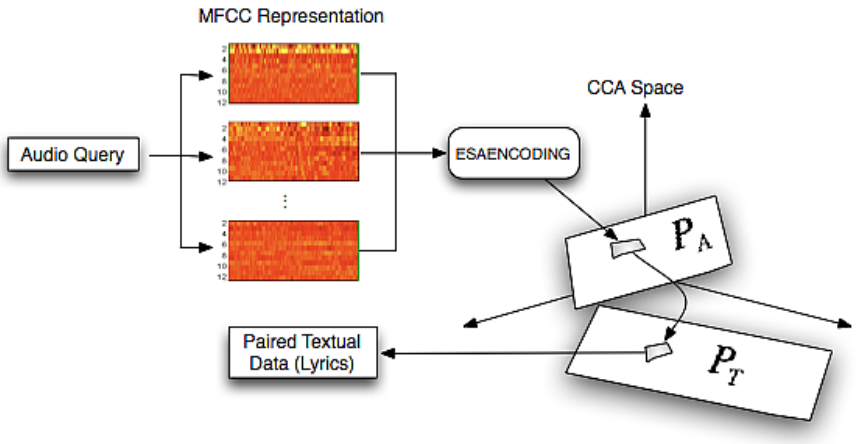


Figure 5.2: A query audio set Q is first represented using its short term windowed MFCC feature vectors. These feature vectors are then quantized and interpreted in terms of the audio words in $\mathcal{D}$. The quantized feature vectors are then projected onto canonical variates P_A and P_T where the corresponding textual information is retrieved as $\mathcal{T}$.

The MSD audio features are accompanied with a collection of music lyrics for 237, 662 tracks [4] that are release by musiXmatch, making the dataset specifically suitable for music information retrieval tasks in multimodal and cross-modal settings. We conducted two sets of cross-modal experiments on two subsets of the MSD dataset. The datasets of these experiments were of sizes $10,000$ and 1000 audio tracks and a 10-fold cross validation is performed across all experiments. We report the retrieval mean average precision across all experiments on this dataset.

5.5.2 Results

In this section we report the results of two rounds of experiments using CCA and a baseline method on the MSD dataset. The observed variables in the audio dataset are represented using **MFCC** feature vectors. These feature vectors are quantized and represented in the shared concept (shared by textual metadata formed from tf-idf encoding of song lyrics). Canonical correlation analysis is then used to project the audio and text representation onto canonical variates P_A and P_T respectively, where the projections are maximally correlated. Figure 5.1 illustrates this process.

Next, a query audio set Q is represented using its short term windowed MFCC feature vectors.

[4]These numbers are reported as the time of writing this thesis according to `http://labrosa.ee.columbia.edu/millionsong/`.

Table 5.1: Lyrics retrieval mean average precision for the Million Song Dataset (MSD) are compared using variations of CCA and ESA representation method of audio MFCC features and their corresponding lyrics. The subsets are correlated with textual tf-idf in the canonical subspaces where the retrieval is performed to obtain lyrics metadata associated with each concept. CCA on quantized feature vectors are compared to modified ESA representation for music information retrieval.

Retrieval Method	**Million Song Data set (MSD)**	
	Small Set (1000 samples)	Large Set (10,000 samples)
Base-line ESA	41%	48.11%
ESA + CCA	46.53%	51.67%

These feature vectors are then quantized and interpreted in terms of the audio words in $\mathcal{D}$. The quantized feature vectors are then projected onto canonical variates P_A and P_T where the corresponding textual information is retrieved as $\mathcal{T}$ as illustrated in Figure 5.2.

For the a baseline, we designed a simple and efficient retrieval method to estimate a set of textual data that best represent an audio query. Let $\mathcal{E}(T)$ and $\mathcal{E}(A)$ denote the paired text and audio training ESA matrices, respectively, and Q represent the audio test query. We will use $k = 1000$ tf-idf values to represent textual data while $N = 1000$ audio keywords represent the corresponding song in the audio feature space. The query matrix Q of the audio signal will be projected into the shared concept space of the paired text and audio training datasets using $P_A = \mathcal{E}_A \times Q^T$. Next, the projection matrix P_A will be transformed to the text space, forming a vector of relevant text data for each audio query using $P_T = P_A^T \times \mathcal{E}_T$.

The resulting inner-products are then sorted by value, from most-relevant to least-relevant, where the most-relevant textual results present the lyrics associated with the audio query. To evaluate the accuracy of retrieval, the Euclidean distance between the audio query and the projected $\mathcal{E}(T)$ is compared to the ground truth labels matrix.

Table 5.1 reports the lyrics retrieval mean average precision (MAP) for MSD using variations of CCA and ESA representation of audio features and lyrics. The subsets are correlated with textual tf-idf in the canonical subspaces where the retrieval is performed to obtain lyrics metadata associated with each concept. Finally, CCA on quantized feature vectors are compared to modified

ESA representation for music information retrieval. A two-proportion Z-test is calculated to show the significance of the results using the ESA encoding method. On the small dataset including 1000 samples, the Z-score is calculated as -2.4925. The p-value is 0.00639, showing the result is significant at $p < 0.05$ with a two-tailed hypothesis. The second dataset with $10,000$ samples, the Z-Score is -5.0346 showing the result is significant at $p < 0.01$. The results indicate that representing audio samples using ESA encoding of their MFCCs improves the accuracy of cross-domain music information retrieval.

5.6 Conclusions

In this chapter, we took a closer look into the inner workings of multimodal and cross-modal IR systems. Our main contributions have been three-fold: (i) we proved that a shared concept space between modalities can be achieved through our ESA embeddings, (ii) we proved that CCA in combination with ESA can create a cross-modal IR system, and, (iii) through a set of experiments we validated that our proposed models is effective and outperforms the baseline retrieval models on a public dataset.

In Section 5.1 we reviewed the popular approaches in building joint models for fusion of various data modalities. In Section 5.2 we explored the cross-modal methods that are used to retrieve objects in different modalities other than the query space. In Section 5.3 we reviewed *canonical correlation analysis* (CCA) as way of measuring the linear relationship between two multidimensional variables. In Section 5.4 we introduced ESA embeddings in combination with CCA as a cross-modal IR framework. We evaluated the performance of the proposed model on a public dataset in Section 5.5.

In our empirical evaluations we demonstrated the utility of the proposed method for lyrics retrieval on a multimodal dataset. The results indicated that representing audio samples using ESA encoding of their MFCCs in combination with CCA improves the accuracy of cross-modal music information retrieval. In the future, we intend to study the use of high-level audio features such as distance-based features and statistical features to enhance classification and retrieval accuracy in multimodal and cross-modal settings. We anticipate that a combination of other audio features will enhance the retrieval performance enabling users with multimodal music retrieval methods.

Chapter 6: Multimodal Approaches to Classification

In this chapter we investigate the multimodal classification and retrieval. Our main hypothesis is that a multimodal classification schema can uncover hidden information in complementary data channels and boost the accuracy in comparison to unimodal methods. To verify this hypothesis, we present two classifiers that operate on multimodal datasets; first we introduce an extension of the ℓ_1-SVM classifier to multimodal settings and second, we present an extension of sparse linear integration models for classification using ESA embeddings. Through a set of experiments we evaluate the performance of the both classifiers on multimodal datasets and provide insight into our hypothesis.

Humans are natural multimedia processing machines, they can combine information from different data sources as an individual identity. A simple example of this is binding the characteristics of one person as an individual through their voice, image and other physical attributes. Humans can detect the same individual through a subset of the physical attributes such as their voice in the absence of one modality or presence of noise in other modalities. The problem of multimodal fusion deals with the coherent integration of media from different sources or multi-modalities into a combined identity for computers. In this chapter, we explore this problem and examine fusion methods that can be used for modeling the multifaceted information sources. In specific we will explore the answer to a fundamental question in multimodal processing; *can we teach computers to learn an individual identity from multiple sensory sources of information?*

The convergence of web, mobile access and digital television in the past decade has boosted the production of images, audio and video contents, making the web a truly multimedia platform [201]. Multimedia is the domain of multi-facets including audio, images, texts, video and other sources of sensory information. As the demand for multimedia grows, the development of information retrieval systems utilizing all available data modalities becomes of paramount importance. Multimodal integration is the study of how information from various sensory modalities can be fused as a single identity. The provision of multiple modalities is motivated by usability, presence of noise in

one modality and non-universality of a single modality. In Chapter 5 we discussed cross-modal IR models that can be used to retrieve relevant information from one modality for a query in another modality. In this chapter, we focus on multimodal approaches that fuse multiple data modalities as a single identity for classification and IR.

Multimodal retrieval systems analyze multiple data sources to eliminate the limitations of single modality and can lead to accurate systems that are resilient to presence of noise and non-universality of a single modality. Identifying dissimilar characteristics between different modalities and optimal selection of fusion algorithms can present challenges in designing such systems [202]. This motivates research efforts addressing variations in information fusion models [202] at four different scales: data, feature, score, and decision levels. Data level fusion is the combination of unprocessed data from available sensors. Since the data formats are often not compatible, this model is not widely used in the literature. Feature level fusion refers to integration of different feature representations into a single combined feature representation for all modalities. The representation of features in a shared space becomes a challenge once using a feature level fusion model. The score level fusion is the aggregation of scores obtained from each individual modality into a final score. Finally, the decision level fusion refers to the combination of decisions from multiple classifiers which allows for flexible choice of classifier for each modality [202].

This chapter is organized as follows; In Section 6.1 we explore various state-of-the-art multimodal approaches to classification and retrieval. In Section 6.2 we introduce the multimodal ℓ_1-SVM classifier which provides an efficient classification scheme utilizing sparse methods to deal with over-fitting for classification of multimodal data. We explore the multimodal sparse linear integration models in Section 6.3. In Section 6.4 we examine the problem of classification in partially multimodal settings and explore state-of-the-art classifiers that can utilize partially multimodal data to achieve higher generalization accuracy. In Section 6.5 we compare the performance of the proposed multimodal ℓ_1-SVM classifier and sparse linear integration models to single modality and baseline multimodal classifiers on a public dataset. Finally we conclude this chapter in Section 6.6 and propose future research directions on multimodal learning.

6.1 Background

The extension of classic IR systems to multimodal settings can be divided into three main tasks; extension of text-based IR models to new modalities such as audio and images, introduction of cross-modal IR systems that can retrieve corresponding objects from one modality for a query in another modality and the utilization of multiple modalities as a single identity or the MIR problem. In Chapter 3 and Chapter 5 we introduced methods of addressing the first and second tasks. In this Chapter, we focus on the final step in MIR systems.

Classic method for IR in text are based on keywords, categories or vector space modeling of textual documents[203–205]. Image retrieval is often achieved through descriptors, texture or pattern recognition, feature based retrieval and retrieval through objects[206–211]. Audio retrieval models are mostly based on metadata (artist, song title, album title, etc.), conversion of audio signals into text words, measuring similarity by rhythm and tempo and annotation-based approaches[212]. MIR methods use a combination of two or more retrieval models to find relevant information in all modalities for a multimodal or single modal query.

MIR often deals with high-dimensional data. The complexity of the datasets paired with the multimodal nature of it presents unique challenges in retrieval. The main challenge in MIR is bridging the *semantic gap* between low-level signal-based features and high-level concepts. The key idea behind the issue is how to find correlations in various modalities that describe the same concept. If a user is searching for a drum playing, how can we find sounds, images and videos that all correspond to the drum with or without captions and metadata? The generalization, scalability, time complexity and handling large sparse feature spaces are also among the most common challenges in designing a MIR system. Numerous models have been proposed in the literature of IR and machine learning communities to address these challenges[213–221].

The development of the MIR system is dependent on the input data modalities and the specific application of the model. The critical concerns when designing the MIR system are: (i) the proper selection of the input data modalities, (ii) the feature extraction method for each modality and (iii) the fusion level and strategy of the modalities. The selection of the appropriate data modalities is

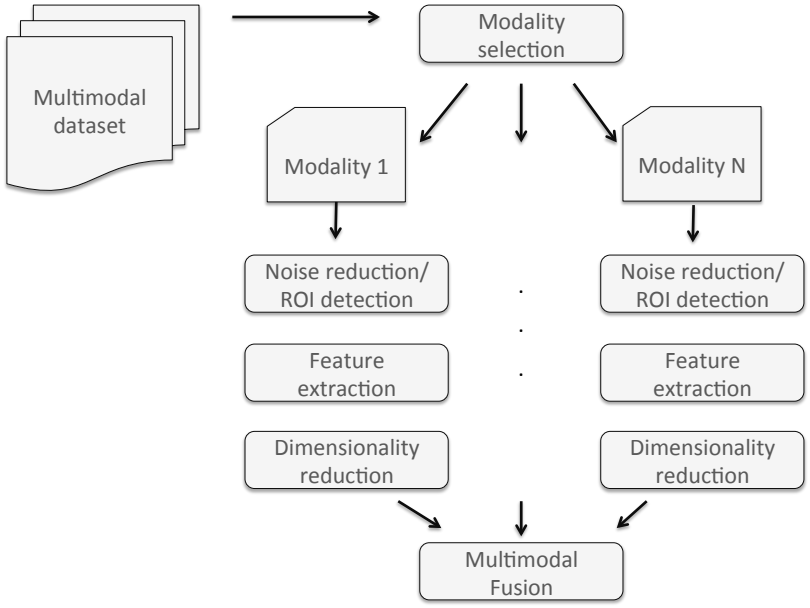


Figure 6.1: The steps involved in multimodal fusion of signals from different modalities are illustrated.

determined by the requirements of the application. These modalities should be capable of providing discriminatory patterns for classification and IR, can be collected quantitatively, and provide the MIR system with low cost and computationally efficient features. As an example consider the MIR system in an autonomous driving vehicle where the input modalities can be collected through various sensors such as Radar, LiDAR, GPS, ultrasonic sensors and cameras. The selection of a combination of these modalities or each one of them can affect the efficiency, cost and scalability of the final MIR system[222]. Camera and GPS might be used in applications such as urban driving while other sensors might be utilized in low visibility road conditions.

The second concern in designing an efficient MIR system is the identification and extraction of complementary and discriminatory features of each selected data modality. The feature space for each data modality is required to capture the characteristics of that modality while providing

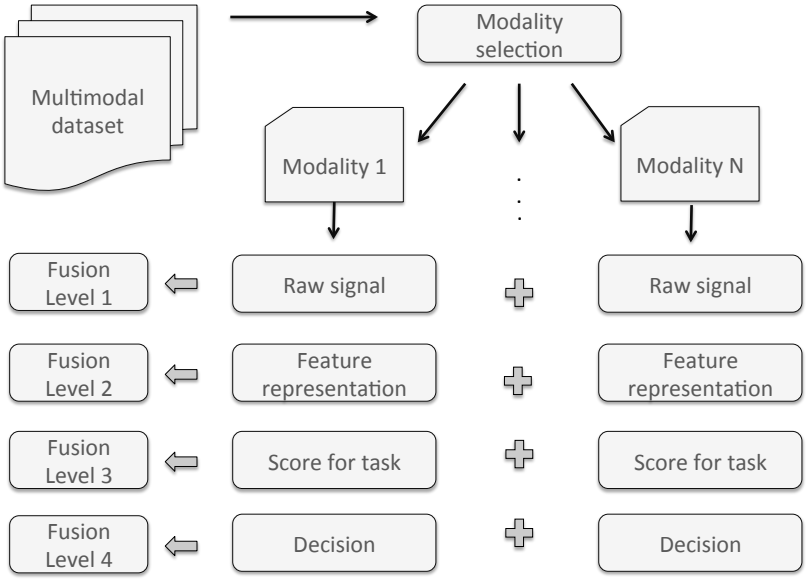


Figure 6.2: Multimodal fusion at four different levels are illustrated: fusion at data level, feature level, score level and decision level.

a distinctive representation against other modalities. The process of feature extraction for each modality often begins with a preprocessing step to reduce noise or detect the region of interest (ROI) [1]. For instance, a Gaussian blur or Gaussian smoothing which is the result of blurring an image by a Gaussian function is often applied to reduce noise in images before feature extraction [223] and ROI detection is used in autonomous driving to limit the feature extraction to the area in front of the vehicle [222]. Once the preprocessing step is completed, feature extraction is performed to represent each data modality in an appropriate feature space. The application and data modality are the main factors that indicate the proper feature space for each signal. As we have seen in previous chapters, MFCC feature vectors are a proper representation of audio signals for classification tasks while SIFT features might be applied to images to extract a compact representation. If the dimensionality of

[1] The noise reduction step is often applied in audio signals while the region of interest (ROI) detection and denoising are both usually applied in images.

the feature space is high, dimensionality reduction steps such as PCA can be applied to project the original feature into an uncorrelated lower-dimensional orthogonal basis set.

The final step in designing a MIR system is the fusion of multiple modalities into a single identity. Figure 6.1 summarizes the steps involved in fusion of multiple data modalities. The information fusion can often occur at four different levels: data, feature, score and decision levels [202]. Figure 6.2 illustrates the fusion at these four different levels. Data and feature level fusion combines the original raw data or extracted features through certain fusion strategies. The disparate characteristics of original data formats or the extracted features often make the fusion at these levels complicated. Moreover fusion at data or feature level results in a high-dimensional data or feature space which can increase the complexity of the MIR system. The *curse of dimensionality* and incomparability of the representation then makes data and feature level fusion a less adopted model in the literature.

Fusion at score level aggregates the scores generated from multiple classifiers or IR systems into a final score using multiple modalities through a rule based scheme [202]. A score normalization process is often applied to scale the scores generated by different modalities in the same range, so that no single modality will over power the others, and the significance of each individual modality is leveraged in the final decision of the classifier or retrieval system. Alternatively, the scores from each modality can be used as the input features to a final classifier or retrieval system where the final decision is a weighted combination of these scores. Finally, fusion at the decision level is the combination of decisions for each modality into a final decision using methods such as majority voting.

The efficiency of the fusion model for the MIR system depends on discriminatory feature extraction and selection, redundancy identification and elimination, information preserving fusion and computational complexity of the model. The main idea behind the multimodal fusion model is to take full advantage of the information collected from multiple sources and bear a better description of the intended perception. A proper multimodal fusion in this sense will outperform a single modality model, is salient to presence of noise in information channels and is universal. An ill-designed multimodal system will possibly produce degraded performance and lowered feasibility in comparison

with some single modality models. In the rest of this chapter we focus on feature level fusion models that utilize all available channels to boost the classification accuracy. We compare the performance of these models with single modality classifiers on public datasets in Section 6.5.

6.2 Multimodal ℓ_1-SVM

The ℓ_1-SVM is a sparse approximation model that addresses the inherent issues of classic SVM. The sparse approximation framework combines the ideas behind classical SVM and sparse approximation techniques and aims to base the classification decision on a small number of well-chosen training examples. Given a set $\langle (x_1, y_1), \cdots, (x_M, y_M) \rangle$ of M training examples, the ℓ_1-SVM classifier finds a vector $\alpha \in \mathbb{R}^M$ such that (i) α is sufficiently sparse, and (ii) the classifier $w \doteq \sum_{i=1}^{M} \alpha_i y_i x_i$ has a sufficiently low empirical loss and therefore sufficiently large separating margin. In Section 4.3 we proved that the classification decision on a new test example x is

$$\hat{y} \doteq \text{sign}\left(\sum_{i:\alpha_i \neq 0} \alpha_i y_i {x_i}^\top x \right). \tag{6.1}$$

In this section we extend the ℓ_1-SVM classifier to multimodal settings. In specific, we introduce a feature level fusion model based on ℓ_1-SVM , the MM-ℓ_1-SVM, that can take advantage of all available modalities while addressing the inherent issues of classic SVM.

Given a set of multimodal examples, we use $S^1 = [s_1^1 s_2^1 \dots s_M^1]$ to denote an $M \times K$ matrix of M feature vectors for the first modality. The s_i^1 is then a $1 \times K$ vectors that denotes the feature representation corresponding to the i-th training example. Similar to the feature representation of the first modality, we use $S^2 = [s_1^2 s_2^2 \dots s_M^2]$ to denote the $M \times K'$ matrix of feature representation for M training examples of the second modality. The s_i^2 vector is then a $1 \times K'$ feature representation of the i-th training example in the second modality. While the multimodal representation of examples in this chapter is limited to two modalities, we can extend all the calculations to more than two modality without loss of generality.

Theorem 4. *MM-ℓ_1-SVM Given a set $\langle (s_1^i, y_1), \cdots, (s_M^i, y_M) \rangle$ of M multimodal training examples, a sparse vector $\alpha = < \alpha^1, \alpha^2 > \in \mathbb{R}^{2 \times M}$ is estimated where α is sufficiently sparse, and the*

multimodal classifier

$$w \doteq \sum_{j=1}^{M} y_j \left(\sum_{i \in \{1,2\}} \alpha_j^i s_j^i \right)$$

has a sufficiently low empirical loss and therefore sufficiently large separating margin. Moreover the classification decision on a new multimodal test example s^i can be formulated as:

$$\hat{y} \doteq \text{sign} \left(\sum_{j:\alpha_j \neq 0} \alpha_j y_j {s_j^i}^\top s^i \right).$$

Proof. To avoid the curse of dimensionality and over-fitting the training examples, we wish to find the sparsest possible solution α. Therefore, by replacing the maximal margin requirement of minimizing $\sum_{i,j=1}^{M} \alpha_i^1 \alpha_j^1 y_i y_j {s_i^1}^\top s_j^1 + \sum_{i,j=1}^{M} \alpha_i^2 \alpha_j^2 y_i y_j {s_i^2}^\top s_j^2$, with the new objective of minimizing $\|\alpha\|_0$, we obtain the following objective function for training a *sparse* linear threshold classifier:

$$\begin{aligned} &\text{minimize } \|\alpha\|_0 + \frac{C}{M} \sum_{i=1}^{M} \xi_i \\ &\text{subject to :} \\ &\qquad \xi_i \geq 0 \qquad\qquad 1 \leq i \leq M \\ &\qquad 1 - \sum_{j=1}^{M} \sum_{k \in \{1,2\}} \alpha_j^k {s_j^k}^\top {s^k}_i y_i y_j \leq \xi_i \quad 1 \leq i \leq M \\ &\qquad 0 \leq \alpha_i \leq \frac{C}{M} \qquad\qquad 1 \leq i \leq M. \end{aligned} \tag{6.2}$$

The above optimization problem is known to be NP-Hard. To overcome the intractability issue, we replace the non-convex pseudo-norm $\|\alpha\|_0$ by the convex ℓ_1 norm $\|\alpha\|_1$ which is the closest convex

norm to the ℓ_0. Therefore, the optimization objective of the ℓ_1-SVM can be stated as

$$\text{minimize} \sum_{j=1}^{M} \sum_{k\in\{1,2\}} \alpha_j^k + \frac{C}{M} \sum_{i=1}^{M} \xi_i$$

subject to :

$$\begin{aligned} &\xi_i \geq 0 && 1 \leq i \leq M \\ &1 - \sum_{j=1}^{M} \sum_{k\in\{1,2\}} \alpha_j^k {s_j^k}^\top {s^k}_i y_i y_j \leq \xi_i && 1 \leq i \leq M \\ &0 \leq \alpha_i \leq \frac{C}{M} && 1 \leq i \leq M. \end{aligned} \tag{6.3}$$

The latter optimization problem is a linear program with the dual objective:

$$\text{maximize} \sum_{j=1}^{M} \gamma_j + \frac{C}{M} \sum_{j=1}^{M} \Phi_j$$

subject to :

$$\begin{aligned} &\Phi_j \leq 0 && 1 \leq j \leq M \\ &1 - \sum_{k=1}^{M} \sum_{i=1,2} \lambda_k y_j y_k {s_k^i}^\top {s^i}_j \geq \Phi_j && 1 \leq j \leq M \\ &0 \leq \gamma_j \leq \frac{C}{M} && 1 \leq j \leq M. \end{aligned} \tag{6.4}$$

This linear program can be efficiently solved using fast gradient descent techniques [153] similar to the ℓ_1-SVM. Finally, the classification decision on a new multimodal test example s^i can now be formulated as $\hat{y} \doteq \text{sign}(\sum_{j:\alpha_j \neq 0} \alpha_j y_j {s_j^i}^\top s^i)$.

The MM-ℓ_1-SVM classifier takes advantage of all available data modalities and combines the ideas behind the classical SVM with the sparse approximation techniques. Similar to ℓ_1-SVM classifier the MM-ℓ_1-SVM classifier aims to achieve higher generalization accuracy, increase the robustness against over-fitting to the multimodal training examples, and provide scalability in terms of the classification complexity. In Section 6.5 we compare the performance of the MM-ℓ_1-SVM with baseline multimodal and single modality classifiers on a public dataset.

6.3 Multimodal SLIM

The sparse linear integration model is an intermediate fusion approach and was originally proposed for retrieval of top recommendation, which generated the results by aggregating user purchase or rating profiles[224]. In the classic sparse linear integration (SLIM), the features are first extracted from each modality to form the content and the context representations. An optimization problem is then formulated to approximate an unseen example (test instance) using a sparse linear combination of training examples. Given a test example, a set of sparse coefficients are learned to reconstruct it using the positive training examples belonging to each class. The difference between the reconstruction error and the unseen test example then represents the error of assigning this unseen example to this class. The smaller the reconstruction error, the higher probability that the test instance belongs to this class[224]. In this section we utilize the ESA representation of each modality in a concept space as the input to the fusion model.

Let $\mathcal{E}(A)$ and $\mathcal{E}(T)$ denote the $M \times K$ and $M \times K'$ multimodal ESA matrices as described in Chapter 3. Our goal is to estimate an $M \times M$ sparse coefficient (classifier) matrix X which is learned by integrating feature information from both modalities. This cross aggregation matrix can be computed by minimizing the estimation error for each modality according to the other one. This latter requirement can be stated as the following optimization problem:

$$\begin{aligned} \text{Minimize} \quad & \alpha_1 ||\mathcal{E}(A) - X\mathcal{E}(A)||_2^2 + \alpha_2 ||\mathcal{E}(T) - X\mathcal{E}(T)||_2^2 \\ & + \beta ||X||_2^2 + \lambda ||X||_1 \\ \text{Subject to:} \quad & X \geq 0, \\ & \text{diag}(X) = 0; \end{aligned} \tag{6.5}$$

Equation 6.5 finds the aggregation matrix X while minimizing the estimation error with respect to the ESA matrices associated with each modality. Here $||.||_2^2$ and $||.||_1$ represent the ℓ_2 and ℓ_1 norms, respectively. The terms $\alpha_1 ||\mathcal{E}(A) - X\mathcal{E}(A)||_2^2$ and $\alpha_2 ||\mathcal{E}(T) - X\mathcal{E}(T)||_2^2$ measure how well the

update fits for both modalities. The β and λ are the regularization parameters which are imposed to enforce sparsity and prevent overfitting [224]. The $X \geq 0$ constraint guarantees a positive correlation between the two modalities and $\text{diag}(X) = 0$ prevents the trivial identity solution where modality is maximally correlated with itself.

The optimization problem of Equation 6.5 can be solved using coordinate gradient descent [225;226] using partial derivatives with respect to columns of matrix X. Since X is a sparse aggregation of the examples in both modalities, it can be used in combination with a k-NN classifier to perform classification for single modal queries.

Specifically, given a query q, we first compute its feature representation. Next, the query q will be projected into the ESA space using the procedure presented in Algorithm 2. The query, q is then represented as a set of M concepts. Let $\mathcal{E}(q)$ denote the ESA representation of the query. The top k nearest neighbors for this query can now be obtained as: $k\text{-NN}(X \times \mathcal{E}(q))$. These neighbors can decide the class label of the input query by aggregating the votes using a majority votes schema. Figure 6.5 and Figure 6.3 show the process for aggregation of both modalities into a fusion model where classification is performed. In specific these figures represent audio and lyrics as the two modalities for an artist identification (classification) task. The audio signals and lyrics are represented as ESA matrices $\mathcal{E}(A)$ and $\mathcal{E}(T)$, respectively. A fusion classifier matrix X in combination with k-NN is then utilized for the artist identification. In Section 6.5 we show the results of the MM-SLIM for artist identification on a public multimodal dataset.

6.4 Challenges

Learning from data has become the most dominant approach in the machine learning community. A natural solution to address the data bias and overfitting in learning models is to incorporate all data channels e.g., audio, lyrics and tags in training. However, it is often impractical to assume access to an additional modality both in training and testing. This motivates the need for learning models that address the classification with partially multimodal data. Given a partially multimodal dataset, we can use the additional modality to tune the classifiers in training and boost the performance in testing. In Chapter 5 we explored the extension of single modality classifiers to modalities beyond

text. We then explored the cross-modal retrieval in Chapter 5. In this chapter, we focused on multimodal classification models such as the proposed MM-ℓ_1-SVM and the MM-SLIM classifier. However, our journey through the multimodal data space will not be completed before we address the challenges and methods used in partially multimodal datasets.

SVM+ algorithm [227] forms the corner stone for most of the state-of-the-art models to address this problem and can be used to include partial information. This model is a generalization of the classical SVM formulation that takes advantage of available hidden information in the training set. In an SVM+ model, partial information is related to the slack variables in the SVM formulation. The slack variable is then modeled as a function of partial information in training. The additional information in this settings is used to parametrize the upper bound on the loss function and essentially build an optimal classifier on the training set. The SVM+ formulation of partially multimodal data, however, is only the generalization of an existing classifier to partially multimodal data. This paradigm does not address the partially multimodal data in a general settings and for an arbitrary classifier such as logistic regression.

Given a set $\langle (s_1^i, y_1), \cdots, (s_M^i, y_M) \rangle$ of M multimodal training examples, we can build a classifier that simultaneously minimizes the Hinge loss function of both modalities. We should notice that this classifier is then used in testing by only accounting for one data modality. To incorporate the additional modality, we build two models $y = f(S^1; w^1)$ and $y = f(S^2; w^2)$, where y denotes the training labels, S^1 and S^2 represents different training modalities. The w^1 and w^2 are the closely related parameters that are simultaneously learnt by minimizing the regularized empirical training loss of:

$$\text{minimize} \quad \sum_{j=1}^{M} \sum_{k \in \{1,2\}} l(y_i, (S_j)^k; w^k) + \alpha ||w^1||_2 + \beta ||w^2||_2 \tag{6.6}$$

where α and β are the regularization terms. The above optimization can be efficiently solved by

gradient descent methods. The partially multimodal classifier can be used to combine different data channels in training and boost the learning accuracy in testing.

6.5 Empirical Evaluation

In this section we present the results of our comparative study suing single and multimodal classifiers on a public dataset. In specific, we evaluate the performance of the classifiers for music genre classification and artist identification tasks on the dataset.

Experimental Setup:

Similar to experiments in Chapter 5 we use the publicly available million song dataset (MSD) [200] containing audio features and metadata for a million contemporary popular music tracks. We report the average classification accuracy rate for genre classification and artist identification on the this dataset. We consider different number of training examples and report the classification accuracy for each subset. A two-proportion Z-test will show the significance of the results for each set of experiments.

6.5.1 Genre Classification

In this section we present the results of a comparative study for music genre classification [228] using single modality and multimodal ℓ_1-SVM classifiers on MSD [200]. The audio tracks are accompanied with a collection of music lyrics for 237, 662 tracks that are release by musiXmatch, making the dataset specifically suitable for music information retrieval tasks. The dataset of these experiments contains $10,000$ audio tracks for which a 10-fold cross validation is performed (we will use 90% of the dataset as the training set and the remaining 10% as the test set). We report the classification accuracy rate for all experiments. To verify the significance of the results a two-proportion Z-test is performed.

For the baseline, we compare the multimodal ℓ_1-SVM (MM-ℓ_1-SVM) to a single modalityℓ_1-SVM, ℓ_2-SVM, ℓ_1-regression and logistic regression on audio features only. The experiments are repeated 10 times and the results are presented in Table 6.1. The performance values are reported based on *mean average classification accuracy rate* for MSD genre classification using single modality

Table 6.1: Classification accuracy (%) on sample dataset is compared across various methods on the MSD dataset using multimodal and single modality data.

Preprocessing	Augmentation	DR-method	Model	Average accuracy rate(%)
Raw wav	No	PCA (270d)	ℓ_2-SVM	18.55%
Raw wav	No	SLPP	ℓ_2-SVM	**17.03**%
Raw wav	No	SR	ℓ_2-SVM	17.46%
MFCC	RM-Uniform	None	MM-ℓ_2-SVM	**38.4**%
MFCC	RM-Uniform	None	ℓ_2-SVM	28.90%
MFCC	RM-Uniform	None	logistic regression	30.10%
MFCC	RM-Uniform	None	ℓ_1-SVM	36.30%
MFCC	RM-Uniform	None	ℓ_1-regression	31.50%

and multimodal representation of audio features and lyrics. Our preliminary results indicate an improvement in classification accuracy rate using a multimodal ℓ_1-SVM. A two-proportion Z-test is also performed to show the significance of the results using the multimodal classifier comparing the results with the best single modality classifier. The Z-score on this experiment is 2.9228. The p-value is 0.0035 indicating that the result is significant at $p < 0.05$. These experiments indicate that a multimodal classification of audio signals and lyrics will improve the classification accuracy rate for music genre classification.

Figure 6.4 shows the average classification accuracy rate of multimodal ℓ_1-SVM (MM-ℓ_1-SVM), ℓ_1-SVM, ℓ_2-SVM, and logistic regression on the corpus of our data set against the average training time for each classifier. The ℓ_1-regression method is a lazy classifier without a training phase and thus is excluded from this experiment. We note that the MM-ℓ_1-SVM outperforms ℓ_1-SVM, ℓ_2-SVM and logistic regression once the convergence threshold is set properly.

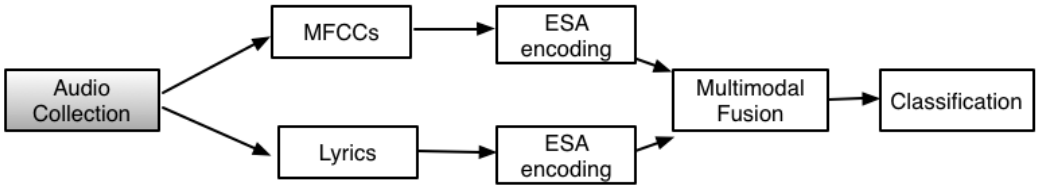


Figure 6.3: Audio signals and lyrics data are projected into ESA space before estimating a fusion model for artist identification on the MSD dataset.

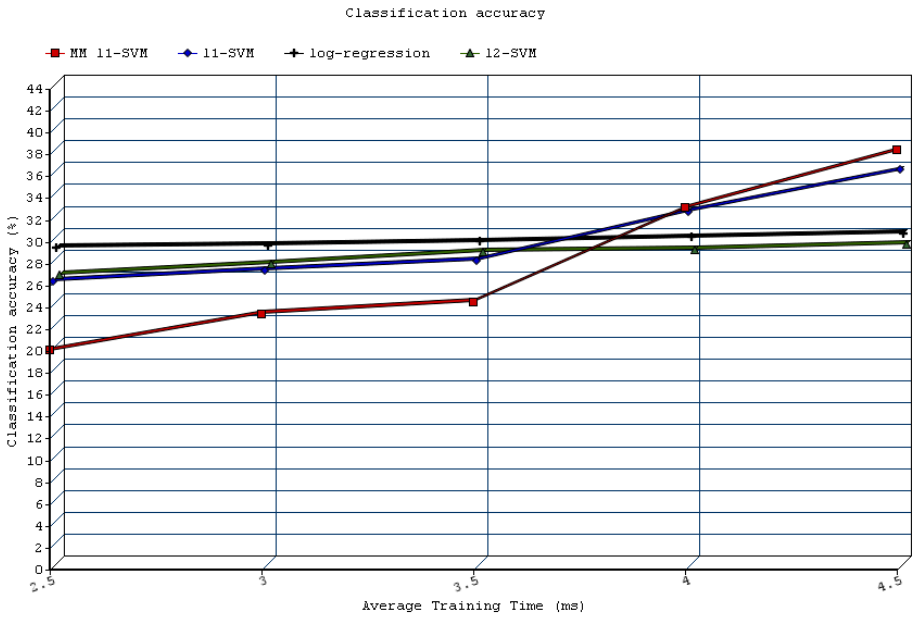


Figure 6.4: Classification accuracy rate is reported for different classification algorithms using different average trainig time on the MSD dataset using multimodal and single modality data.

6.5.2 Artist Identification

We evaluated the performance of the multimodal feature fusion, namely the multimodal sparse linear integration model introduced in Section 6.3 on the MSD dataset for an artist classification task[229]. Figure 6.3 presents an overview of the fusion for audio and lyrics. Figure 6.5 illustrates the process of artist classification in more details.

We report the classification accuracy rate for all experiments using a 10-fold cross validation schema. To verify the significance of our results a two-proportion Z-test is performed. For the baseline, we compare the multimodal fusion to a single modality k-NN on top of the ESA representation of audio signals. The experiments are repeated 10 times and the average results are presented in Table 6.2. The performance values are reported based on mean average classification accuracy rate for MSD artist identification using single modality and multimodal ESA representation of audio features and lyrics. We use a k-NN algorithms with $k = 5$ as the classification model in the fusion and single modality spaces in our experiments. Our preliminary results indicate an improvement in

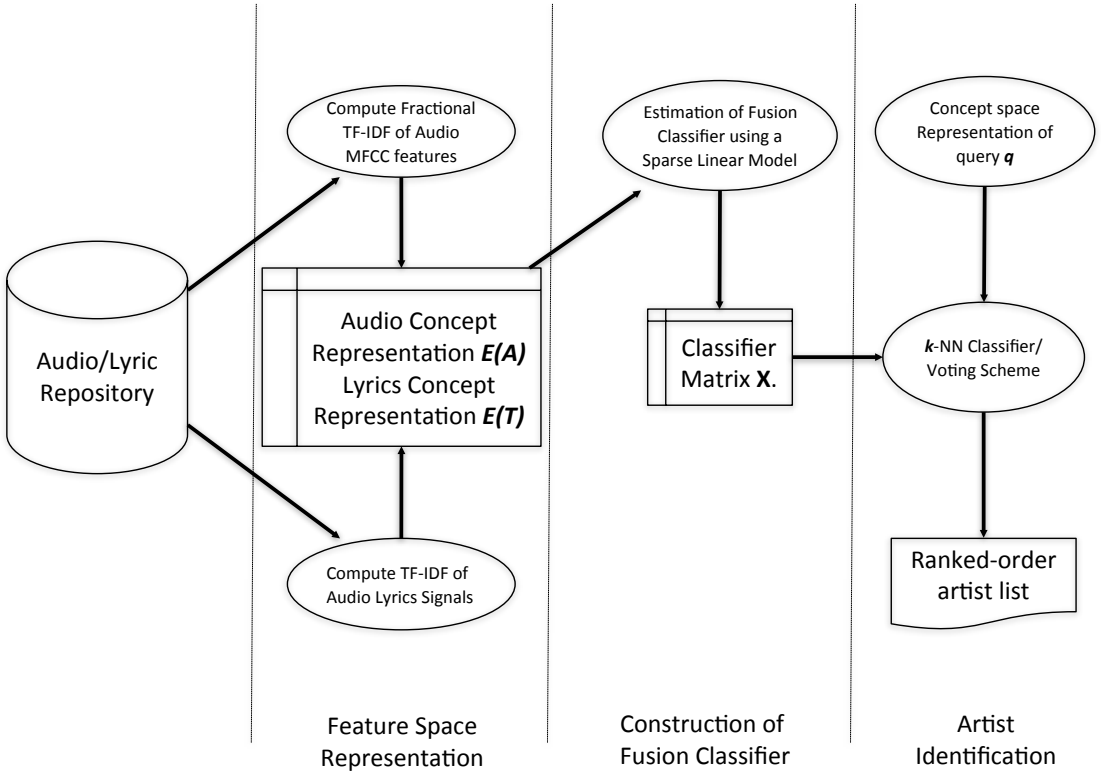


Figure 6.5: Audio signals and lyrics are used to create a fusion classifier for music artist identification. Audio signals and lyrics are represented as ESA matrices $\mathcal{E}(A)$ and $\mathcal{E}(T)$, respectively. A fusion classifier matrix X in combination with k-NN is then utilized for artist identification.

classification accuracy rate using a multimodal feature fusion model.

A two-proportion Z-test is also performed to show the significance of the results using the multimodal feature level fusion. On the small dataset of 1000 samples, the Z-score is -1.8881. The p-value is 0.05876, indicating that the result are statistically significant at $p < 0.10$ with a two-tailed hypothesis. For the second dataset of $10,000$ samples, the Z-Score is -8.7466 showing the result is significant at $p < 0.01$. These experiments indicate that a multimodal feature fusion of audio signals and lyrics will improve the classification accuracy rate for music artist identification.

Table 6.2: Artist Identification accuracy (%) on sample dataset is compared across various methods on the MSD dataset using mutimodal and single modality data. Top k nearest neighbors is fixed as the classification algorithm using single modality ESA (audio only) and multimodal ESA fusion (audio and lyrics).

Preprocessing	Sample Size	DR-method	Model	Average accuracy rate(%)
MFCC	1K	None	audio ESA +k-NN	32.07%
MFCC	10K	None	audio ESA +k-NN	28.91%
MFCC	10K	None	Fusion ESA + k-NN	**36.54**%
MFCC	10K	None	Fusion ESA + k-NN	34.67%

6.6 Conclusions

In this chapter we presented an extension of ℓ_1-SVM and SLIM to multimodal settings. Our hypotheses was that a multimodal classifier which can take advantage of available data modalities can boost the performance of classification on multimodal datasets. To this end, we evaluated the classification accuracy of ℓ_1-SVM, SLIM and baseline unimodal classifiers on public datasets. Our Experimental results indicated that a multimodal classifier can take advantage of all available channels and boost the accuracy on different classification tasks.

In Section 6.1 we explored various state-of-the-art multimodal approaches to classification and retrieval. In Section 6.2 we introduced the multimodal ℓ_1-SVM classifier which provides an efficient classification scheme utilizing sparse methods to deal with over-fitting for classification of multimodal data. The multimodal ℓ_1-SVM takes advantage of all available data modalities while avoiding the inherent problems of the classical SVM. Next, we explored the multimodal sparse linear integration models in Section 6.3.

In Section 6.4 we examined the problem of classification in partially multimodal settings and explored the state-of-the-art classifiers that can utilize partially multimodal data to achieve higher generalization accuracy. Finally, in Section 6.5 we compared the performance of the proposed multimodal ℓ_1-SVM classifier and sparse linear integration models to single modality and baseline multimodal classifiers on a public dataset. We evaluated the performance of these models in two different tasks on the *million song dataset (MSD)*; a music genre classification and an artist classification.

Our experimental results show that the multimodal approaches to classification can outperform single modality methods while the proper features are selected for each modality. Moreover, we observed that the MM-ℓ_1-SVM provides an efficient fusion model for integration of multiple modalities.

In the future we intend to study the relevance feedback techniques that allow users to associate their ranking of the returned results. This ranking is further used by the system to improve the retrieved results in the second round and boost the accuracy of the classification and multimodal information retrieval system. We also intend to explore the semantic gap problem that is present between different modalities. Overcoming the semantic gap between low level features and high level features can enhance the MIR systems.

Chapter 7: Concluding Remarks

This thesis has presented a number of contributions to the fields of computer science, machine learning, information retrieval, and related fields. Our contributions can be divided into three main categories: (i) single modality, (ii) cross-modality and (iii) multimodal contributions. Single modality refers to classification techniques were the query or test data and the training data are all of the same nature and hence can be presented in the same feature space. Cross-modal retrieval denotes a set of machine learning problems when the query is of one modality and the database objects are of another modality. Finally multimodal information retrieval and machine learning is about the search for information in any modality (text, image, audio and video), using combinations of two or more data modalities.

The single modality approaches focus on boosting the classification accuracy through more efficient feature representation and classifier selection. In specific we have demonstrated an extension of the vector space model, namely the explicit semantic analysis (ESA) for signals such as audio and images. We have also introduced a sparsity-eager classifier that involves a subset of training examples in the final classifier decision and thus achieves higher generalization accuracy on test examples while avoiding overfitting on training data. We have demonstrated the usefulness of both the ESA and this classification technique through a set of experimental work on multiple datasets.

The cross-modal contributions combine an extension of the vector space modeling of signals with canonical correlation analysis (CCA) to introduce a novel retrieval framework. Our multimodal contributions mainly focus on the extension of the sparsity-eager classifier to multimodal data and the introduction of a sparse linear integration model on these datasets. Similar to single modality approaches, the cross-modal and multimodal techniques have been tested through a set of experiments on multiple datasets. In this chapter we review the contributions of this dissertation, offer recommendations to researchers wishing to use the material in their own work, and suggest areas for future work on the topic.

7.1 Summary and Conclusions

This dissertation has explored the problem of classification and information retrieval from different perspectives in single and multimodal settings. There are four major contributions of this thesis. First, we have introduced a novel extension to the vector space modeling of signals. Second, we have introduced a sparsity-eager classifier. Third, we have introduced a cross-modal retrieval model that can be used to retrieve the database objects from one modality for a query in a different modality. Finally, we introduced an extension of the sparsity-eager classifier to multimodal settings and compared its performance with a sparse linear integration model of multiple modalities. Through this dissertation we have demonstrated the utility of all the introduced methods through experimental work on multiple single and multimodal datasets. In this section we review these contributions in more details.

7.1.1 ESA Feature Encoding

Efficient feature representation that can capture the essence of each data modality is a fundamental problem in machine learning. The standard Boolean model and the vector space model are two examples of such feature representation for text documents. In Chapter 3 we introduced these methods and the term frequency inverse document frequency (tf-idf) as a widely used weighting schemata in more details.

Explicit semantic analysis (ESA) [53] is an extension of the vector space model for the enhancement of text categorization [53]. The ESA model was later on adopted for *semantic relatedness* computation. In this sense, the cosine similarity between the vector representation of documents in the knowledge base are computed and collectively interpreted as a semantic space of *concepts explicitly defined by humans*. Wikipedia articles or otherwise titles of documents in the knowledge base are equated with concepts [118] in this model.

The ESA model utilizes the tf-idf weighting schemata to represent low-level textual information in terms of concepts in higher-dimensional space. Specifically, in the ESA model, the relationship between a code-word and a concept (document) pair will be captured through the tf-idf value of

the word-concept pair. The tf-idf values will be obtained by multiplying tf (the frequency of a given codeword in the document/concept) and the idf values (the inverse document frequency that quantifies the importance of a word over a given set of concepts). The set of tf-idf scores associated with a codeword will form a vector representation of the code-word in the concept space.

The ESA model presents a suitable representation of text documents in a concept space. Extensions of this model can be used to provide the same benefits for other data modalities such as audio and images. In Chapter 3 we presented an extension of the ESA model to encode time-series data such as audio signals by introducing a set of *fractional* term frequency (tf) and inverse document frequency (idf) models. The ESA encoding of audio (signal-based) features provide an effective semantic space for classification and categorization of signals. The proposed model is validated through a set of experiments in applying the ESA model for a classification task on a public dataset. In specific, the ESA model was adopted for a music genre classification task.

7.1.2 ℓ_1-SVM

Selection of a suitable classifier is an important step in a classification task. The choice of the classifier often depends on the dataset, sparsity of the feature representation and time or complexity constraints. Support vector machines (SVMs) of Vapnik *et. al* [130] are supervised learning models that are widely used for classification and regression analysis. SVM [77;78] provides an efficient classification scheme based on maximizing the margin between training examples belonging to different classes. However, a majority of the training examples are involved in making the final decisions for the test examples. As we discussed in Chapter 4, the final decision for any new instance x is $\text{sign}(\sum_{i=1}^{M} \alpha_i y_i x_i^\top x)$. This can result in overfitting on the training dataset. The principals of the learning theory suggest that a simpler classifier which only depends on a subset of well-chosen training examples is statistically more suitable.

The inherent issue in SVM motivates a classification schema that involves a small subset of training examples in the final decision of the classifier. This classifier selects the subset of training examples by imposing a sparsity condition in the classification optimization. In Section 4.3, we introduced this sparse approximation model that addresses the inherent issues of classic SVM, namely

the sparsity-eager SVM or the ℓ_1-SVM. We combine the ideas behind SVM and ℓ_1-regression methods to come up with the more robust ℓ_1-SVM algorithm with the goal of obtaining higher generalization accuracy on test data, increasing the robustness against overfitting to the training examples, and providing scalability in terms of the classification complexity. In Chapter 4, we provided evidences on the advantages of the new ℓ_1-SVM classifier over the baseline classification methods. We also introduced a new deep neural network architecture as a comparison to the sparsity-eager SVM. Through a set of empirical results we evaluate the performance of the proposed classifier.

In specific, in Section 4.4 we compared the classification performance of the proposed classifier with state-of-the-art classifiers on a public dataset for a music classification task. We compared the results using raw audio data and power spectrogram of input signals with a two-layer neural network. We explored different manifold learning techniques such as PCA, SLPP and SR as dimensionality reduction methods. We then compared the classification results using hand-crafted audio feature vectors such as MFCCs. Our experimental results indicate that MFCCs in combination with ℓ_1-SVM classifiers are comparable with deep neural networks on power spectrograms for music genre classification.

7.1.3 Cross-Modal Retrieval

Information retrieval (IR) methods have traditionally been developed around text information. Later, IR methods were extended to include other modalities such as images and audio. Each data modality often has specific set of feature representations and IR models that are calibrated for that modality. All these models of retrievals have their pros and cons, and are therefore used in different types of applications. There is a need of a generic system that can work on various techniques and modalities. This generic system can be used to retrieve information from one modality that correspond to a user query from another modality. This retrieval process between multiple modalities is the so-called cross-modal retrieval problem and is becoming increasingly important as more data sources are becoming readily available across different platforms. Users can issue queries on their cell phone, web and other platforms via speech, text and images to search and retrieve objects from all different modalities. This creates numerous challenging problems for classic machine learning and

retrieval models. First the IR models are required to extend beyond the traditional text modality. This problem was addressed in Chapters 3 and 4 by introduction of ESA model and classification techniques that are independent of data modality. Second, the IR models are required to span across more than one modality. In Chapter 6 we explored multimodal IR and classification approaches to address this problem. Third, the IR models are required to be able to understand, retrieve and rank queries and objects from different data sources. The cross-modal techniques introduced in Chapter 5 mainly deals with this cross-modal IR challenges.

The ESA model that was introduced in Chapter 3 is a semantic representation of any modality in a concept space. Intuitively this semantic space is designed based on the concepts in the knowledge base and is independent of a data modality. Text, audio, images and other signals can all be presented through the ESA embeddings. This shared semantic representation motivates IR methods that can encode more than one data modality at the same time and ease the burden of moving across multiple modalities in IR systems.

In Chapter 5 we introduced two different cross-modal retrieval models that benefit from the shared semantic space devised by ESA embeddings of data modalities. The first approach involves the *canonical correlation analysis* (CCA) as way of measuring the linear relationship between two multidimensional variables. Once the data modalities (both query and database objects) are presented in the shared semantic space by ESA, we can use the CCA to correlate the two representations. The second method is a direct transition between the two data modalities by using the ESA representation. In Chapter 5 we took a closer look into the inner workings of these cross-modal methods. We evaluated the performance of the proposed model on a public dataset in Section 5.5. In our empirical evaluations we demonstrated the utility of the proposed method for lyrics retrieval on a multimodal dataset. The results indicated that representing audio samples using ESA encoding of their low-level features in combination with CCA improves the accuracy of cross-modal music information retrieval.

7.1.4 Multimodal Classification

Multimodal information retrieval (MIR) is about the search for information in any modality such as text, image, audio and video on any platform, using combinations of two or more data modalities. Extension of existing retrieval models and introduction of new multimodal retrieval methods is an increasingly important challenge in machine learning and IR. A set of novel unified frameworks for multimodal search and retrieval have been proposed in the literature of IR and machine learning communities. The multimodal approaches enables users to issue queries that contains information across all modalities. Users can retrieve images, audio, video, news, text and metadata about a topic of interest. In order to support such challenging requests, researcher needs to work on several fronts; new semantic models for combining individual data modalities, new retrieval engines for crossing the single modality boundary during search, novel interfaces for input and output data presentation and new retrieval models for retrieving relative information within the same modality as well as in cross modal settings are among the top challenges in MIR.

The final section of this dissertation was dedicated to addressing the multimodal challenges in machine learning and IR. In specific, we introduced an extension of the novel sparsity-eager SVM or the ℓ_1-SVM of Chapter 4 to multimodal settings. Such multimodal extension enables the classifiers to benefit from information across all available data channels. This results in classifiers that are resilient to loss of information or noise in one data channel, are universal and achieve higher generalization accuracy on test examples. In Chapter 6 we also present a multimodal sparse linear integration model as a unifying method for multiple data sources. We evaluate the performance of both multimodal classifiers on a set of public datasets for genre classification and artist identification tasks. Our experimental results indicate that the multimodal methods can outperform the single modality methods in terms of classification accuracy while providing robustness in the presence of noise or loss of data in one channel.

7.2 Future Work

In this thesis we introduced a set of advancements in the machine learning and IR. We devised feature representation and classification methods that can boost the classification accuracy and address the inherent issues of some classic classifiers such as SVMs. Our primary focus have been on extension of these models to other modalities beyond text, cross-modal settings and multi-modalities. We introduced a number of methods and extensions that can be used to take advantage of multiple data modalities. Several avenues for interesting future work remain to be addressed. First, the problem of feature presentation requires careful studies. In our empirical results we have utilized a subset of feature vectors for audio and text that have been hand-crafted for specific tasks. The MFCC feature vectors for example are mainly designed for speaker identification tasks and we have chosen them based on their performance on a classification task via a fixed classifier as discussed in Chapter 1. It is however not clear how different features impact the accuracy of different classifiers.

A second interesting research direction is the comparison of hand-crafted features in combination with SVM based classifiers with neural networks. With recent advances in neural networks and their performance in many different fields it is an interesting to answer *under what conditions can SVMs outperform NNs?*. In this dissertation we compared our classifiers with a neural network but the question remains as different architectures and preprocessing could be used with both classifiers and input data respectively.

A third avenue of research is the extension of cross-modal and multimodal experiments to other data modalities beyond text and audio. The extension of the proposed models in this dissertation are yet to be applied outside of the music datasets with lyrics and metadata presented in this thesis. It is interesting to observe how the proposed models will extend to images, video and other sensory data. We believe that the application of devised methods in this thesis could prove fruitful on other data modalities.

Bibliography

[1] Koby Crammer, Yoram Singer, Nello Cristianini, John Shawe-taylor, and Bob Williamson. On the algorithmic implementation of multiclass kernel-based vector machines. *Journal of Machine Learning Research*, 2:2001, 2001.

[2] Ioannis Tsochantaridis, Thomas Hofmann, Thorsten Joachims, and Yasemin Altun. Support vector machine learning for interdependent and structured output spaces. In *Proceedings of the twenty-first international conference on Machine learning*, ICML '04, pages 104–. ACM, 2004.

[3] Vladimir N. Vapnik. *Statistical Learning Theory*. Wiley-Interscience, 1998.

[4] Donald Michie, David J Spiegelhalter, and Charles C Taylor. Machine learning, neural and statistical classification. 1994.

[5] Sotiris B Kotsiantis, I Zaharakis, and P Pintelas. Supervised machine learning: A review of classification techniques, 2007.

[6] Martin Hilbert and Priscila López. The worlds technological capacity to store, communicate, and compute information. *science*, 332(6025):60–65, 2011.

[7] Kaichun K. Chang, Jyh-Shing Roger Jang, and Costas S. Iliopoulos. Music genre classification via compressive sampling. In J. Stephen Downie and Remco C. Veltkamp, editors, *ISMIR*, pages 387–392. International Society for Music Information Retrieval, 2010. ISBN 978-90-393-53813.

[8] Pradeep Ravikumar, Martin J. Wainwright, and John D. Lafferty. High-dimensional ising model selection using ℓ_1-regularized logistic regression. *ANNALS OF STATISTICS*, 38:1287, 2010. URL `doi:10.1214/09-AOS691`.

[9] Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*, 9: 1871–1874, 2008.

[10] Padraig Cunningham and Sarah Jane Delany. k-nearest neighbour classifiers. *Multiple Classifier Systems*, pages 1–17, 2007.

[11] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366, 1989.

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

[13] Sridhar Ramaswamy, Pablo Tamayo, Ryan Rifkin, Sayan Mukherjee, Chen-Hsiang Yeang, Michael Angelo, Christine Ladd, Michael Reich, Eva Latulippe, Jill P Mesirov, et al. Multiclass cancer diagnosis using tumor gene expression signatures. *Proceedings of the National Academy of Sciences*, 98(26):15149–15154, 2001.

[14] Bishnu S Atal and Lawrence R Rabiner. A pattern recognition approach to voiced-unvoiced-silence classification with applications to speech recognition. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 24(3):201–212, 1976.

[15] Kamelia Aryafar and Ali Shokoufandeh. Music genre classification using explicit semantic analysis. In *Proceedings of the 1st international ACM workshop on Music information retrieval with user-centered and multimodal strategies*, pages 33–38. ACM, 2011.

[16] Lei Xu, Adam Krzyżak, and Ching Y Suen. Methods of combining multiple classifiers and their applications to handwriting recognition. *Systems, man and cybernetics, IEEE transactions on*, 22(3):418–435, 1992.

[17] Inderjit S Dhillon, Subramanyam Mallela, and Rahul Kumar. A divisive information theoretic feature clustering algorithm for text classification. *The Journal of Machine Learning Research*, 3:1265–1287, 2003.

[18] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *Unsupervised learning*. Springer, 2009.

[19] Anil K Jain, M Narasimha Murty, and Patrick J Flynn. Data clustering: a review. *ACM computing surveys (CSUR)*, 31(3):264–323, 1999.

[20] M Masroor Ahmed and Dzulkifli Bin Mohamad. Segmentation of brain mr images for tumor extraction by combining kmeans clustering and perona-malik anisotropic diffusion model. *International Journal of Image Processing*, 2(1):27–34, 2008.

[21] Zhenyu Wu and Richard Leahy. An optimal graph theoretic approach to data clustering: Theory and its application to image segmentation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 15(11):1101–1113, 1993.

[22] James Philbin, Ondřej Chum, Michael Isard, Josef Sivic, and Andrew Zisserman. Object retrieval with large vocabularies and fast spatial matching. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–8. IEEE, 2007.

[23] Tony A Meyer and Brendon Whateley. Spambayes: Effective open-source, bayesian based, email classification system. In *CEAS*. Citeseer, 2004.

[24] Erin L Allwein, Robert E Schapire, and Yoram Singer. Reducing multiclass to binary: A unifying approach for margin classifiers. *The Journal of Machine Learning Research*, 1:113–141, 2001.

[25] Ronald A Fisher. The statistical utilization of multiple measurements. *Annals of eugenics*, 8 (4):376–386, 1938.

[26] Ronald A Fisher. The use of multiple measurements in taxonomic problems. *Annals of eugenics*, 7(2):179–188, 1936.

[27] Ram Gnanadesikan. *Methods for statistical data analysis of multivariate observations*, volume 321. John Wiley & Sons, 2011.

[28] William N Venables and Brian D Ripley. *Modern applied statistics with S-PLUS*. Springer Science & Business Media, 2013.

[29] CR Rao. Advanced statistical methods in multivariate analysis, 1952.

[30] Theodore Wilbur Anderson, Theodore Wilbur Anderson, Theodore Wilbur Anderson, and Theodore Wilbur Anderson. *An introduction to multivariate statistical analysis*, volume 2. Wiley New York, 1958.

[31] Shayle R Searle. *Linear models*. Wiley, 2012.

[32] Lorenzo Cappellari and Stephen P Jenkins. Multivariate probit regression using simulated maximum likelihood. *The Stata Journal*, 3(3):278–294, 2003.

[33] Frank Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386, 1958.

[34] A Aizerman, Emmanuel M Braverman, and LI Rozoner. Theoretical foundations of the potential function method in pattern recognition learning. *Automation and remote control*, 25: 821–837, 1964.

[35] Isabelle Guyon, B Boser, and Vladimir Vapnik. Automatic capacity tuning of very large vc-dimension classifiers. *Advances in neural information processing systems*, pages 147–147, 1993.

[36] Thomas Hofmann, Bernhard Schölkopf, and Alexander J Smola. Kernel methods in machine learning. *The annals of statistics*, pages 1171–1220, 2008.

[37] George Forman. An extensive empirical study of feature selection metrics for text classification. *The Journal of machine learning research*, 3:1289–1305, 2003.

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

[39] Glenn W Brier. Verification of forecasts expressed in terms of probability. *Monthly weather review*, 78(1):1–3, 1950.

[40] George Hripcsak and Adam S Rothschild. Agreement, the f-measure, and reliability in information retrieval. *Journal of the American Medical Informatics Association*, 12(3):296–298, 2005.

[41] Bernd Girod. What's wrong with mean-squared error? In *Digital images and human vision*, pages 207–220. MIT press, 1993.

[42] Ron Kohavi et al. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Ijcai*, volume 14, pages 1137–1145, 1995.

[43] Ji-Hyun Kim. Estimating classification error rate: Repeated cross-validation, repeated hold-out and bootstrap. *Computational Statistics & Data Analysis*, 53(11):3735–3745, 2009.

[44] Adam Coates, Andrew Y Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In *International conference on artificial intelligence and statistics*, pages 215–223, 2011.

[45] Liefeng Bo, Xiaofeng Ren, and Dieter Fox. Unsupervised feature learning for rgb-d based object recognition. In *Experimental Robotics*, pages 387–402. Springer, 2013.

[46] Guoqiang Peter Zhang. Neural networks for classification: a survey. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 30(4):451–462, 2000.

[47] Yoav Freund and Robert E Schapire. Large margin classification using the perceptron algorithm. *Machine learning*, 37(3):277–296, 1999.

[48] Julien Mairal, Jean Ponce, Guillermo Sapiro, Andrew Zisserman, and Francis R Bach. Supervised dictionary learning. In *Advances in neural information processing systems*, pages 1033–1040, 2009.

[49] Ignacio Ramirez, Pablo Sprechmann, and Guillermo Sapiro. Classification and clustering via dictionary learning with structured incoherence and shared features. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 3501–3508. IEEE, 2010.

[50] Aapo Hyvärinen, Juha Karhunen, and Erkki Oja. *Independent component analysis*, volume 46. John Wiley & Sons, 2004.

[51] Quoc V Le. Building high-level features using large scale unsupervised learning. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pages 8595–8598. IEEE, 2013.

[52] Daniel D Lee and H Sebastian Seung. Algorithms for non-negative matrix factorization. In *Advances in neural information processing systems*, pages 556–562, 2001.

[53] Evgeniy Gabrilovich and Shaul Markovitch. Computing semantic relatedness using wikipedia-based explicit semantic analysis. In *IJCAI*, volume 7, pages 1606–1611, 2007.

[54] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. Orb: an efficient alternative to sift or surf. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 2564–2571. IEEE, 2011.

[55] Luo Juan and Oubong Gwun. A comparison of sift, pca-sift and surf. *International Journal of Image Processing (IJIP)*, 3(4):143–152, 2009.

[56] Jing Li and Nigel M Allinson. A comprehensive review of current local features for computer vision. *Neurocomputing*, 71(10):1771–1787, 2008.

[57] Anil Jain and Douglas Zongker. Feature selection: Evaluation, application, and small sample performance. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 19(2):153–158, 1997.

[58] N. Scaringella, G. Zoia, and D. Mlynek. Automatic genre classification of music content: a survey. *Signal Processing Magazine, IEEE*, 23(2):133–141, 2006. URL `http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1598089`.

[59] Wietse Balkema and Ferdi van der Heijden. Music playlist generation by assimilating gmms into soms. *Pattern Recognition Letters*, 31(11):1396 – 1402, 2010. ISSN 0167-8655. doi: DOI:10.1016/j.patrec.2010.02.001. URL `http://www.sciencedirect.com/science/article/pii/S0167865510000462`.

[60] Arthur Flexer, Dominik Schnitzer, Martin Gasser, and Gerhard Widmer. Playlist generation using start and end songs. In Juan Pablo Bello, Elaine Chew, and Douglas Turnbull, editors, *ISMIR*, pages 173–178, 2008. ISBN 978-0-615-24849-3. URL `http://dblp.uni-trier.de/db/conf/ismir/ismir2008.html#FlexerSGW08`.

[61] George Tzanetakis and Perry R. Cook. Musical genre classification of audio signals. *IEEE Transactions on Speech and Audio Processing*, 10(5):293–302, 2002.

[62] Dan-Ning Jiang Jiang, Lie Lu, Hong-Jiang Zhang, Jian-Hua Tao, and Lian-Hong Cai. Music type classification by spectral contrast feature. *Proceedings IEEE International Conference on Multimedia and Expo*, 1:113–116, 2002. URL `http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=1035731`.

[63] Tao Li, Mitsunori Ogihara, and Qi Li. A comparative study on content-based music genre classification. In *in Proc. SIGIR, 2003*, pages 282–289.

[64] Cory McKay and Ichiro Fujinaga. Musical genre classification: Is it worth pursuing and how can it be improved? In *ISMIR*, pages 101–106, 2006.

[65] Zehra Cataltepe, Yusuf Yaslan, and Abdullah Sonmez. Music genre classification using MIDI and audio features. *EURASIP J. Appl. Signal Process.*, 2007:150–150, January 2007. ISSN 1110-8657. doi: http://dx.doi.org/10.1155/2007/36409. URL `http://dx.doi.org/10.1155/2007/36409`.

[66] T. Lidy and A. Rauber. Improving genre classification by combination of audio and symbolic descriptors using a transcription system. In *Proc. ISMIR*, 2007.

[67] Amelie Anglade, Emmanouil Benetos, Matthias Mauch, and Simon Dixon. Improving music genre classification using automatically induced harmony rules. *Journal of New Music Research*, 39:349–361, 2010.

[68] Anders Meng, Peter Ahrendt, and Jan Larsen. Improving music genre classification by short-time feature integration. In *IEEE ICASSP*, pages 497–500, 2005.

[69] Cory McKay and Ichiro Fujinaga. Combining features extracted from audio, symbolic and cultural sources. In *ISMIR*, pages 597–602, 2008.

[70] Robert Neumayer and Andreas Rauber. Integration of text and audio features for genre classification in music information retrieval. In *Proceedings of the 29th European Conference on Information Retrieval*, pages 724–727, 2007.

[71] Brian Whitman. Combining musical and cultural features for intelligent style detection. In *Proc. Int. Conf. Music Information Retrieval (ISMIR)*, pages 47–52, 2002.

[72] Ling Chen, Phillip Wright, and Wolfgang Nejdl. Improving music genre classification using collaborative tagging data. In *Web Search and Data Mining*, pages 84–93, 2009. doi: 10.1145/1498759.1498812.

[73] Tom LH. Li and Antoni B. Chan. Genre classification and the invariance of mfcc features to key and tempo. In *Proceedings of the 17th international conference on Advances in multimedia modeling - Volume Part I*, MMM'11, pages 317–327, Berlin, Heidelberg, 2011. Springer-Verlag. ISBN 3-642-17831-6, 978-3-642-17831-3. URL `http://portal.acm.org/citation.cfm?id=1949994.1950029`.

[74] Malcolm Slaney. Auditory toolbox, version 2. Technical Report 1998-10, Interval Research Corporation, Palo Alto, California, USA, 1998. URL `http://cobweb.ecn.purdue.edu/~malcolm/interval/1998-010/`.

[75] M. J. Hunt, M. Lennig, and P. Mermelstein. Experiments in syllable-based recognition of continuous speech. In *International Conference on Acoustics, Speech, and Signal Processing*, 1980.

[76] Daniel P. W. Ellis. Classifying music audio with timbral and chroma features. In *Proc. Int. Conf. on Music Information Retrieval ISMIR-07, Vienna, Austria*, 2007.

[77] Christopher J. C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2:121–167, 1998.

[78] Bernhard Scholkopf and Alexander J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, Cambridge, MA, USA, 2001. ISBN 0262194759.

[79] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, New York, NY, USA, 2004. ISBN 0521833787.

[80] Evgeniy Gabrilovich and Shaul Markovitch. Computing semantic relatedness using Wikipedia-based explicit semantic analysis. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence*, pages 1606–1611, 2007.

[81] Gerard Salton, Anita Wong, and Chung-Shu Yang. A vector space model for automatic indexing. *Communications of the ACM*, 18(11):613–620, 1975.

[82] Gerard Salton. The smart retrieval systemexperiments in automatic document processing. 1971.

[83] Christopher C Yang, Hsinchun Chen, and Kay Hong. Visualization of large category map for internet browsing. *Decision Support Systems*, 35(1):89–102, 2003.

[84] Tie-Yan Liu. Learning to rank for information retrieval. *Foundations and Trends in Information Retrieval*, 3(3):225–331, 2009.

[85] Peter D Turney, Patrick Pantel, et al. From frequency to meaning: Vector space models of semantics. *Journal of artificial intelligence research*, 37(1):141–188, 2010.

[86] Christopher D Manning, Prabhakar Raghavan, Hinrich Schütze, et al. *Introduction to information retrieval*, volume 1. Cambridge university press Cambridge, 2008.

[87] Peter Turney, Michael L Littman, Jeffrey Bigham, and Victor Shnayder. Combining independent modules to solve multiple-choice synonym and analogy problems. 2003.

[88] Peter D Turney and Michael L Littman. Corpus-based learning of analogies and semantic relations. *Machine Learning*, 60(1-3):251–278, 2005.

[89] Reinhard Rapp. Word sense discovery based on sense descriptor dissimilarity. In *Proceedings of the Ninth Machine Translation Summit*, pages 315–322, 2003.

[90] Choon Yang Quek and T Mitchell. Classification of world wide web documents. *Master's thesis, School of Computer Science Carnegie Mellon University*, 1997.

[91] Ian H Witten and Eibe Frank. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, 2005.

[92] Fabrizio Sebastiani. Machine learning in automated text categorization. *ACM computing surveys (CSUR)*, 34(1):1–47, 2002.

[93] Paul Resnick, Neophytos Iacovou, Mitesh Suchak, Peter Bergstrom, and John Riedl. Grouplens: an open architecture for collaborative filtering of netnews. In *Proceedings of the 1994 ACM conference on Computer supported cooperative work*, pages 175–186. ACM, 1994.

[94] John S Breese, David Heckerman, and Carl Kadie. Empirical analysis of predictive algorithms for collaborative filtering. In *Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence*, pages 43–52. Morgan Kaufmann Publishers Inc., 1998.

[95] Greg Linden, Brent Smith, and Jeremy York. Amazon. com recommendations: Item-to-item collaborative filtering. *Internet Computing, IEEE*, 7(1):76–80, 2003.

[96] Tak W Yan and Hector Garcia-Molina. Index structures for information filtering under the vector space model. In *Data Engineering, 1994. Proceedings. 10th International Conference*, pages 337–347. IEEE, 1994.

[97] Dik L Lee, Huei Chuang, and Kent Seamons. Document ranking and the vector-space model. *Software, IEEE*, 14(2):67–75, 1997.

[98] Arash Habibi Lashkari, Fereshteh Mahdavi, and Vahid Ghomi. A boolean model in information retrieval for search engines. In *Information Management and Engineering, 2009. ICIME'09. International Conference on*, pages 385–389. IEEE, 2009.

[99] Frederick Wilfrid Lancaster and Emily Gallup. Information retrieval on-line. Technical report, 1973.

[100] Steven P Wartik. Boolean operations., 1992.

[101] WB Frakes. Introduction to information storage and retrieval systems. *Space*, 14:10, 1992.

[102] G. Salton, A. Wong, and C. S. Yang. *A vector space model for automatic indexing*, pages 273–280. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1997. ISBN 1-55860-454-5. URL `http://portal.acm.org/citation.cfm?id=275537.275700`.

[103] George W Furnas, Scott Deerwester, Susan T Dumais, Thomas K Landauer, Richard A Harshman, Lynn A Streeter, and Karen E Lochbaum. Information retrieval using a singular value decomposition model of latent semantic structure. In *Proceedings of the 11th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 465–480. ACM, 1988.

[104] Julio Gonzalo, Felisa Verdejo, Irina Chugur, and Juan Cigarran. Indexing with wordnet synsets can improve text retrieval. *arXiv preprint cmp-lg/9808002*, 1998.

[105] George A Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 38 (11):39–41, 1995.

[106] SK Michael Wong, Wojciech Ziarko, and Patrick CN Wong. Generalized vector spaces model in information retrieval. In *Proceedings of the 8th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 18–25. ACM, 1985.

[107] Fazli Can and Esen A Ozkarahan. Computation of term/document discrimination values by use of the cover coefficient concept. *Journal of the American Society for Information Science*, 38(3):171–183, 1987.

[108] Susan T Dumais. Latent semantic analysis. *Annual review of information science and technology*, 38(1):188–230, 2004.

[109] Gabriel Recchia, Michael Jones, Magnus Sahlgren, and Pentti Kanerva. Encoding sequential information in vector space models of semantics: Comparing holographic reduced representation and random permutation. 2010.

[110] Aditya Joshi, Johan Halseth, and Pentti Kanerva. Language recognition using random indexing. *arXiv preprint arXiv:1412.7026*, 2014.

[111] Alexandros Ntoulas, Junghoo Cho, and Christopher Olston. What's new on the web?: the evolution of the web from a search engine perspective. In *Proceedings of the 13th international conference on World Wide Web*, pages 1–12. ACM, 2004.

[112] Sara Cohen, Jonathan Mamou, Yaron Kanza, and Yehoshua Sagiv. Xsearch: A semantic search engine for xml. In *Proceedings of the 29th international conference on Very large data bases-Volume 29*, pages 45–56. VLDB Endowment, 2003.

[113] Thorsten Joachims. A probabilistic analysis of the rocchio algorithm with tfidf for text categorization. Technical report, DTIC Document, 1996.

[114] 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.

[115] Andreas Hotho, Steffen Staab, and Gerd Stumme. Ontologies improve text document clustering. In *Data Mining, 2003. ICDM 2003. Third IEEE International Conference on*, pages 541–544. IEEE, 2003.

[116] Mark Kantrowitz, Behrang Mohit, and Vibhu Mittal. Stemming and its effects on tfidf ranking (poster session). In *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 357–359. ACM, 2000.

[117] Ibrahim Abu El-Khair. Effects of stop words elimination for arabic information retrieval: a comparative study. *International Journal of Computing & Information Sciences*, 4(3):119–133, 2006.

[118] Evgeniy Gabrilovich and Shaul Markovitch. Overcoming the brittleness bottleneck using wikipedia: Enhancing text categorization with encyclopedic knowledge. In *AAAI*, volume 6, pages 1301–1306, 2006.

[119] Ofer Egozi, Shaul Markovitch, and Evgeniy Gabrilovich. Concept-based information retrieval using explicit semantic analysis. *ACM Transactions on Information Systems (TOIS)*, 29(2): 8, 2011.

[120] Yin Zhang, Rong Jin, and Zhi-Hua Zhou. Understanding bag-of-words model: a statistical framework. *International Journal of Machine Learning and Cybernetics*, 1(1-4):43–52, 2010.

[121] Thomas Gottron, Maik Anderka, and Benno Stein. Insights into explicit semantic analysis. In *Proceedings of the 20th ACM international conference on Information and knowledge management*, pages 1961–1964. ACM, 2011.

[122] Maik Anderka and Benno Stein. The esa retrieval model revisited. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 670–671. ACM, 2009.

[123] Martin Potthast, Benno Stein, and Maik Anderka. A wikipedia-based multilingual retrieval model. In *Advances in Information Retrieval*, pages 522–530. Springer, 2008.

[124] Maik Anderka, Benno Stein, and Martin Potthast. Cross-language high similarity search: why no sub-linear time bound can be expected. In *Advances in Information Retrieval*, pages 640–644. Springer, 2010.

[125] Helge Homburg, Ingo Mierswa, Bülent Möller, Katharina Morik, and Michael Wurst. A benchmark dataset for audio classification and clustering. In *ISMIR*, pages 528–531, 2005.

[126] Ingo Mierswa and Katharina Morik. Automatic feature extraction for classifying audio data. *Machine Learning Journal*, 58:127–149, 2005.

[127] Olivier Lartillot and Petri Toiviainen. MIR in Matlab (II): A toolbox for musical feature extraction from audio. *International Conference on Music Information Retrieval*, 2007. URL `http://en.scientificcommons.org/42610575`.

[128] Jigang Wang, Predrag Neskovic, and Leon N Cooper. Improving nearest neighbor rule with a simple adaptive distance measure. *Pattern Recognition Letters*, 28(2):207–213, 2007.

[129] Nayyar Zaidi, David Squire, and David Suter. Boostml: An adaptive metric learning for nearest neighbor classification. *Advances in Knowledge Discovery and Data Mining*, pages 142–149, 2010.

[130] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3): 273–297, 1995.

[131] Gavin C Cawley and Nicola LC Talbot. On over-fitting in model selection and subsequent selection bias in performance evaluation. *The Journal of Machine Learning Research*, 11: 2079–2107, 2010.

[132] Li Deng and Dong Yu. Deep learning: methods and applications. *Foundations and Trends in Signal Processing*, 7(3–4):197–387, 2014.

[133] Yoshua Bengio. Learning deep architectures for ai. *Foundations and trends® in Machine Learning*, 2(1):1–127, 2009.

[134] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[135] Geoffrey E Hinton, Simon Osindero, and Yee-Whye Teh. A fast learning algorithm for deep belief nets. *Neural computation*, 18(7):1527–1554, 2006.

[136] Alex Graves, Marcus Liwicki, Santiago Fernández, Roman Bertolami, Horst Bunke, and Jürgen Schmidhuber. A novel connectionist system for unconstrained handwriting recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 31(5):855–868, 2009.

[137] Philippe Hamel and Douglas Eck. Learning features from music audio with deep belief networks. In *ISMIR*, pages 339–344. Utrecht, The Netherlands, 2010.

[138] Honglak Lee, Peter Pham, Yan Largman, and Andrew Y Ng. Unsupervised feature learning for audio classification using convolutional deep belief networks. In *Advances in neural information processing systems*, pages 1096–1104, 2009.

[139] Siddharth Sigtia and Sam Dixon. Improved music feature learning with deep neural networks. In *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*, pages 6959–6963. IEEE, 2014.

[140] John A Lee and Michel Verleysen. *Nonlinear dimensionality reduction*. Springer Science & Business Media, 2007.

[141] Mark A Kramer. Nonlinear principal component analysis using autoassociative neural networks. *AIChE journal*, 37(2):233–243, 1991.

[142] Caifeng Shan, Shaogang Gong, and Peter W McOwan. Appearance manifold of facial expression. In *Computer Vision in Human-Computer Interaction*, pages 221–230. Springer, 2005.

[143] Deng Cai, Xiaofei He, and Jiawei Han. Spectral regression for efficient regularized subspace learning. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8. IEEE, 2007.

[144] Lars Kai Hansen and Peter Salamon. Neural network ensembles. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (10):993–1001, 1990.

[145] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *International conference on artificial intelligence and statistics*, pages 249–256, 2010.

[146] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 807–814, 2010.

[147] Robert Malouf. A comparison of algorithms for maximum entropy parameter estimation. In *proceedings of the 6th conference on Natural language learning-Volume 20*, pages 1–7. Association for Computational Linguistics, 2002.

[148] Dimitri Kanevsky, Tara N. Sainath, Bhuvana Ramabhadran, and David Nahamoo. An analysis of sparseness and regularization in exemplar-based methods for speech classification. In *INTERSPEECH*, pages 2842–2845, 2010.

[149] Tara N. Sainath, Bhuvana Ramabhadran, David Nahamoo, Dimitri Kanevsky, and Abhinav Sethy. Sparse representation features for speech recognition. In *INTERSPEECH*, pages 2254–2257. ISCA, 2010.

[150] B. K. Natarajan. Sparse approximate solutions to linear systems. *SIAM Journal of Computing*, 24:227–234, 1995.

[151] E. J. Candès, J. K. Romberg, and T. Tao. Stable signal recovery from in- complete and inaccurate measurements. *Communications on Pure and Applied Mathematics*, 59:1207–1223, 2006.

[152] D. L. Donoho. Compressed sensing. *IEEE Transactions on Information Theory*, 52:1289–1306, 2006.

[153] Yuri Nesterov. *Introductory lectures on convex optimization: A basic course*. Springer, 2004.

[154] Nikhil Rasiwasia, Jose Costa Pereira, Emanuele Coviello, Gabriel Doyle, Gert RG Lanckriet, Roger Levy, and Nuno Vasconcelos. A new approach to cross-modal multimedia retrieval. In *Proceedings of the international conference on Multimedia*, pages 251–260. ACM, 2010.

[155] Jose Costa Pereira, Emanuele Coviello, Gabriel Doyle, Nikhil Rasiwasia, Gert RG Lanckriet, Roger Levy, and Nuno Vasconcelos. On the role of correlation and abstraction in cross-modal multimedia retrieval. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 36 (3):521–535, 2014.

[156] G. Salton and M. J. Mcgill. *Introduction to Modern Information Retrieval*. McGraw-Hill, Inc., 1983.

[157] Nuno Vasconcelos. Minimum probability of error image retrieval. *Signal Processing, IEEE Transactions on*, 52(8):2322–2336, 2004.

[158] Ritendra Datta, Dhiraj Joshi, Jia Li, and James Z Wang. Image retrieval: Ideas, influences, and trends of the new age. *ACM Computing Surveys (CSUR)*, 40(2):5, 2008.

[159] Arnold WM Smeulders, Marcel Worring, Simone Santini, Amarnath Gupta, and Ramesh Jain. Content-based image retrieval at the end of the early years. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(12):1349–1380, 2000.

[160] Erling Wold, Thom Blum, Douglas Keislar, and James Wheaten. Content-based classification, search, and retrieval of audio. *MultiMedia, IEEE*, 3(3):27–36, 1996.

[161] Jonathan T Foote. Content-based retrieval of music and audio. In *Voice, Video, and Data Communications*, pages 138–147. International Society for Optics and Photonics, 1997.

[162] Jiwoon Jeon, Victor Lavrenko, and Raghavan Manmatha. Automatic image annotation and retrieval using cross-media relevance models. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 119–126. ACM, 2003.

[163] Rainer Typke, Frans Wiering, and Remco C Veltkamp. A survey of music information retrieval systems. In *ISMIR*, pages 153–160, 2005.

[164] Douglas Turnbull, Luke Barrington, David Torres, and Gert Lanckriet. Semantic annotation and retrieval of music and sound effects. *Audio, Speech, and Language Processing, IEEE Transactions on*, 16(2):467–476, 2008.

[165] Yong Rui, Thomas S Huang, and Shih-Fu Chang. Image retrieval: Current techniques, promising directions, and open issues. *Journal of visual communication and image representation*, 10 (1):39–62, 1999.

[166] Alan F Smeaton, Paul Over, and Wessel Kraaij. Evaluation campaigns and trecvid. In *Proceedings of the 8th ACM international workshop on Multimedia information retrieval*, pages 321–330. ACM, 2006.

[167] Florent Monay and Daniel Gatica-Perez. Modeling semantic aspects for cross-media image indexing. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 29(10):1802–1817, 2007.

[168] Victor Lavrenko, R Manmatha, and Jiwoon Jeon. A model for learning the semantics of pictures. In *Advances in neural information processing systems*, page None, 2003.

[169] SL Feng, Raghavan Manmatha, and Victor Lavrenko. Multiple bernoulli relevance models for image and video annotation. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 2, pages II–1002. IEEE, 2004.

[170] James Z Wang and Jia Li. Learning-based linguistic indexing of pictures with 2–d mhmms. In *Proceedings of the tenth ACM international conference on Multimedia*, pages 436–445. ACM, 2002.

[171] Malcolm Slaney. Mixtures of probability experts for audio retrieval and indexing. In *Multimedia and Expo, 2002. ICME'02. Proceedings. 2002 IEEE International Conference on*, volume 1, pages 345–348. IEEE, 2002.

[172] Kamelia Aryafar and Ali Shokoufandeh. Fusion of text and audio semantic representations through cca. In *Multimodal Pattern Recognition of Social Signals in Human-Computer-Interaction*. Springer, 2014.

[173] Nikhil Rasiwasia, Pedro J Moreno, and Nuno Vasconcelos. Bridging the gap: Query by semantic example. *Multimedia, IEEE Transactions on*, 9(5):923–938, 2007.

[174] Nuno Vasconcelos. From pixels to semantic spaces: Advances in content-based image retrieval. *Computer*, (7):20–26, 2007.

[175] Kobus Barnard, Pinar Duygulu, David Forsyth, Nando De Freitas, David M Blei, and Michael I Jordan. Matching words and pictures. *The Journal of Machine Learning Research*, 3:1107–1135, 2003.

[176] Cees GM Snoek and Marcel Worring. Multimodal video indexing: A review of the state-of-the-art. *Multimedia tools and applications*, 25(1):5–35, 2005.

[177] Ludovic Denoyer and Patrick Gallinari. Bayesian network model for semi-structured document classification. *Information processing & management*, 40(5):807–827, 2004.

[178] Satoshi Nakamura. Statistical multimodal integration for audio-visual speech processing. *Neural Networks, IEEE Transactions on*, 13(4):854–866, 2002.

[179] Tomas Kliegr, Krishna Chandramouli, Jan Nemrava, Vojtech Svatek, and Ebroul Izquierdo. Combining image captions and visual analysis for image concept classification. In *Proceedings of the 9th International Workshop on Multimedia Data Mining: held in conjunction with the ACM SIGKDD 2008*, pages 8–17. ACM, 2008.

[180] Gang Wang, Derek Hoiem, and David Forsyth. Building text features for object image classification. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 1367–1374. IEEE, 2009.

[181] John W Fisher III, Trevor Darrell, William T Freeman, and Paul A Viola. Learning joint statistical models for audio-visual fusion and segregation. In *NIPS*, pages 772–778, 2000.

[182] Thijs Westerveld. Image retrieval: Content versus context. In *RIAO*, pages 276–284. Citeseer, 2000.

[183] Trong-Ton Pham, Nicolas Eric Maillot, Joo-Hwee Lim, and Jean-Pierre Chevallet. Latent semantic fusion model for image retrieval and annotation. In *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, pages 439–444. ACM, 2007.

[184] Hugo Jair Escalante, Carlos A Hérnadez, Luis Enrique Sucar, and Manuel Montes. Late fusion of heterogeneous methods for multimedia image retrieval. In *Proceedings of the 1st ACM international conference on Multimedia information retrieval*, pages 172–179. ACM, 2008.

[185] Guo-Jun Qi, Charu Aggarwal, and Thomas Huang. Towards semantic knowledge propagation from text corpus to web images. In *Proceedings of the 20th international conference on World wide web*, pages 297–306. ACM, 2011.

[186] David R Hardoon, Sandor Szedmak, and John Shawe-Taylor. Canonical correlation analysis: An overview with application to learning methods. *Neural computation*, 16(12):2639–2664, 2004.

[187] Dongge Li, Nevenka Dimitrova, Mingkun Li, and Ishwar K Sethi. Multimedia content processing through cross-modal association. In *Proceedings of the eleventh ACM international conference on Multimedia*, pages 604–611. ACM, 2003.

[188] Hong Zhang, Yueting Zhuang, and Fei Wu. Cross-modal correlation learning for clustering on image-audio dataset. In *Proceedings of the 15th international conference on Multimedia*, pages 273–276. ACM, 2007.

[189] Malcolm Slaney. Semantic-audio retrieval. In *Acoustics, Speech, and Signal Processing (ICASSP), 2002 IEEE International Conference on*, volume 4, pages IV–4108. IEEE, 2002.

[190] Yi Yang, Dong Xu, Feiping Nie, Jiebo Luo, and Yueting Zhuang. Ranking with local regression and global alignment for cross media retrieval. In *Proceedings of the 17th ACM international conference on Multimedia*, pages 175–184. ACM, 2009.

[191] Yue-Ting Zhuang, Yi Yang, and Fei Wu. Mining semantic correlation of heterogeneous multimedia data for cross-media retrieval. *Multimedia, IEEE Transactions on*, 10(2):221–229, 2008.

[192] Yueting Zhuang, Yi Yang, Fei Wu, and Yunhe Pan. Manifold learning based cross-media retrieval: a solution to media object complementary nature. *The Journal of VLSI Signal Processing Systems for Signal, Image, and Video Technology*, 46(2-3):153–164, 2007.

[193] Yi Yang, Yue-Ting Zhuang, Fei Wu, and Yun-He Pan. Harmonizing hierarchical manifolds for multimedia document semantics understanding and cross-media retrieval. *Multimedia, IEEE Transactions on*, 10(3):437–446, 2008.

[194] Vijay Mahadevan, Chi W Wong, Jose C Pereira, Tom Liu, Nuno Vasconcelos, and Lawrence K Saul. Maximum covariance unfolding: Manifold learning for bimodal data. In *Advances in Neural Information Processing Systems*, pages 918–926, 2011.

[195] Alexei Vinokourov, Nello Cristianini, and John S Shawe-Taylor. Inferring a semantic representation of text via cross-language correlation analysis. In *Advances in neural information processing systems*, pages 1473–1480, 2002.

[196] Winston Hsu, Tao Mei, and Rong Yan. Knowledge discovery over community-sharing media: from signal to intelligence. In *Multimedia and Expo, 2009. ICME 2009. IEEE International Conference on*, pages 1448–1451. IEEE, 2009.

[197] Richard Socher, Milind Ganjoo, Christopher D Manning, and Andrew Ng. Zero-shot learning through cross-modal transfer. In *Advances in neural information processing systems*, pages 935–943, 2013.

[198] H. Hotelling. "Relations Between two Sets of Variates". *Biometrika*, 28:321–377, 1936.

[199] J Michael Steele. *The Cauchy-Schwarz master class: an introduction to the art of mathematical inequalities*. Cambridge University Press, 2004.

[200] Thierry Bertin-Mahieux, Daniel P.W. Ellis, Brian Whitman, and Paul Lamere. The million song dataset. In *Proceedings of the 12th International Conference on Music Information Retrieval (ISMIR 2011)*, 2011.

[201] HU Bokhari and Faraz Hasan. Multimodal information retrieval: Challenges and future trends. *International Journal of Computer Applications*, 74(14):9–12, 2013.

[202] Pradeep K Atrey, M Anwar Hossain, Abdulmotaleb El Saddik, and Mohan S Kankanhalli. Multimodal fusion for multimedia analysis: a survey. *Multimedia systems*, 16(6):345–379, 2010.

[203] Gilles Hubert and Josiane Mothe. An adaptable search engine for multimodal information retrieval. *Journal of the American Society for Information Science and Technology*, 60(8): 1625–1634, 2009.

[204] Edward Lim, James Liu, and Raymond Lee. *Knowledge Seeker–Ontology Modelling for Information Search and Management*. Springer, 2011.

[205] Victor Lavrenko and W Bruce Croft. Relevance models in information retrieval. In *Language modeling for information retrieval*, pages 11–56. Springer, 2003.

[206] Rajshree S Dubey, Rajnish Choubey, and Joy Bhattacharjee. Multi feature content based image retrieval. *International Journal on Computer Science and Engineering*, 2(6):2145–2149, 2010.

[207] Yixin Chen and James Z Wang. A region-based fuzzy feature matching approach to content-based image retrieval. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24 (9):1252–1267, 2002.

[208] 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.

[209] Florent Perronnin, Yan Liu, Jorge Sánchez, and Hervé Poirier. Large-scale image retrieval with compressed fisher vectors. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 3384–3391. IEEE, 2010.

[210] Faraj Alhwarin, Chao Wang, Danijela Ristic-Durrant, and Axel Gräser. Improved sift-features matching for object recognition. In *BCS Int. Acad. Conf.*, pages 178–190, 2008.

[211] Supheakmungkol Sarin and Wataru Kameyama. Joint equal contribution of global and local features for image annotation. *CLEF working notes*, 2009.

[212] Guodong Guo and Stan Z Li. Content-based audio classification and retrieval by support vector machines. *Neural Networks, IEEE Transactions on*, 14(1):209–215, 2003.

[213] Apostolos Axenopoulos, Petros Daras, Sotiris Malassiotis, Vincenzo Croce, Marilena Lazzaro, Jonas Etzold, Paul Grimm, Alberto Massari, Antonio Camurri, Thomas Steiner, et al. I-search: a unified framework for multimodal search and retrieval. In *The Future Internet*, pages 130–141. Springer, 2012.

[214] Jana Urban, Joemon M Jose, and Cornelis J Van Rijsbergen. An adaptive technique for content-based image retrieval. *Multimedia Tools and Applications*, 31(1):1–28, 2006.

[215] Panel Moderator. Multimedia access and retrieval: The state of the art and future directions. 1999.

[216] Alejandro Jaimes, Mike Christel, Sébastien Gilles, Ramesh Sarukkai, and Wei-Ying Ma. Multimedia information retrieval: what is it, and why isn't anyone using it? In *Proceedings of the 7th ACM SIGMM international workshop on Multimedia information retrieval*, pages 3–8. ACM, 2005.

[217] Marc Davis, Simon King, Nathan Good, and Risto Sarvas. From context to content: leveraging context to infer media metadata. In *Proceedings of the 12th annual ACM international conference on Multimedia*, pages 188–195. ACM, 2004.

[218] Alexander G Hauptmann and Michael G Christel. Successful approaches in the trec video retrieval evaluations. In *Proceedings of the 12th annual ACM international conference on Multimedia*, pages 668–675. ACM, 2004.

[219] Richard A Bolt. *Put-that-there: Voice and gesture at the graphics interface*, volume 14. ACM, 1980.

[220] Mieczyslaw M Kokar, Jerzy A Tomasik, and Jerzy Weyman. Formalizing classes of information fusion systems. *Information Fusion*, 5(3):189–202, 2004.

[221] Anil K Jain and Arun Ross. Multibiometric systems. *Communications of the ACM*, 47(1): 34–40, 2004.

[222] Yasamin Alkhorshid, Kamelia Aryafar, Gerd Wanielik, and Ali Shokoufandeh. Camera-based lane marking detection for adas and autonomous driving. In *Image Analysis and Recognition*, pages 514–519. Springer, 2015.

[223] Pietro Perona and Jitendra Malik. Scale-space and edge detection using anisotropic diffusion. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 12(7):629–639, 1990.

[224] Xia Ning and George Karypis. Slim: Sparse linear methods for top-n recommender systems. In *Data Mining (ICDM), 2011 IEEE 11th International Conference on*, pages 497–506. IEEE, 2011.

[225] Jerome Friedman, Trevor Hastie, and Rob Tibshirani. Regularization paths for generalized linear models via coordinate descent. *Journal of statistical software*, 33(1):1, 2010.

[226] Qiusha Zhu, Zhao Li, Haohong Wang, Yimin Yang, and Mei-Ling Shyu. Multimodal sparse linear integration for content-based item recommendation. In *Multimedia (ISM), 2013 IEEE International Symposium on*, pages 187–194. IEEE, 2013.

[227] Vladimir Vapnik. Learning hidden information: Svm+. In *Granular Computing, 2006 IEEE International Conference on*, pages 22–22. IEEE, 2006.

[228] Kamelia Aryafar and Ali Shokoufandeh. Multimodal sparsity-eager support vector machines for music classification. In *Machine Learning and Applications (ICMLA), 2014 13th International Conference on*, pages 405–408. IEEE, 2014.

[229] Kamelia Aryafar and Ali Shokoufandeh. Multimodal music and lyrics fusion classifier for artist identification. In *Machine Learning and Applications (ICMLA), 2014 13th International Conference on*, pages 506–509. IEEE, 2014.

Vita

Kamelia Aryafar

Email: karyafar@gmail.com

Git: https://github.com/karyafar

Homepage: https://www.cs.drexel.edu/ kca26/

Education:

2003–2007 B.Sc. Computer Engineering

Sharif University Of Technology, Tehran , Iran

2007–2010 M.Sc. Computer Science

Drexel University, Philadelphia, PA, USA

2010–2015 Ph.D. Computer Science

Drexel University, Philadelphia, PA, USA

Thesis:*Multimodal Information Retrieval and Classification*

Publications:

- Arjun Raj Rajanna, Kamelia Aryafar, Ali Shokoufandeh and Raymond Ptucha. *Deep Neural Networks: A Case Study for Music Genre Classification.* Machine Learning and Applications (ICMLA), 2015 14th International Conference on. IEEE, 2015.
- Arjun Raj Rajanna, Kamelia Aryafar, Rajeev Ramchandran, Christye Sisson, Ali Shokoufandeh and Raymond Ptucha. *Neural Networks with Manifold Learning for Diabetic Retinopathy Detection.*Machine Learning and Applications (ICMLA), 2015 14th International Conference on. IEEE, 2015.
- Yasamin Alkhorshid, Kamelia Aryafar, Gerd Wanielik, and Ali Shokoufandeh. *Camera-Based Lane Marking Detection for ADAS and Autonomous Driving.* In Image Analysis and Recognition, pp. 514-519. Springer International Publishing, 2015.

- Kamelia Aryafar and Jerry Soung. *Exploring Alternate Modalities for Tag Recommendation.* Multimodal Pattern Recognition of Social Signals in Human-Computer-Interaction. Springer International Publishing, 2015. 141-144.

- Kamelia Aryafar, Corey Lynch, and Josh Attenberg. *Exploring User Behaviour on Etsy through Dominant Colors.* Pattern Recognition (ICPR), 2014 22nd International Conference on. IEEE, 2014.

- Kamelia Aryafar and Ali Shokoufandeh. *Multimodal Music and Lyrics Fusion Classifier for Artist Identification.* Machine Learning and Applications (ICMLA), 2014 13th International Conference on. IEEE, 2014.

- Kamelia Aryafar and Ali Shokoufandeh. *Fusion of Text and Audio Semantic Representations Through CCA*. Multimodal Pattern Recognition of Social Signals in Human-Computer-Interaction. Springer International Publishing, 2015. 66-73.

- Kamelia Aryafar and Ali Shokoufandeh. *Multimodal Sparsity-Eager Support Vector Machines for Music Classification.* Machine Learning and Applications (ICMLA), 2014 13th International Conference on. IEEE, 2014.

- Kamelia Aryafar, Sina Jafarpour, and Ali Shokoufandeh. *Automatic musical genre classification using sparsity-eager support vector machines.* In Pattern Recognition (ICPR), 2012 21st International Conference on, pages 15261529. IEEE, 2012.

- Kamelia Aryafar and Ali Shokoufandeh. *Music genre classification using explicit semantic analysis.* In Proceedings of the 1st international ACM workshop on Music information retrieval with user-centered and multimodal strategies, pages 3338. ACM, 2011.

- Kamelia Aryafar and Patrice L. Jeppson. *A Semantic Analysis Approach to Thin-shell Ceramic Fragment Classification.* International Journal of Heritage in the Digital Era 2.2 (2013): 291-306.

- Kamelia Aryafar, Sina Jafarpour, and Ali Shokoufandeh. *Music genre classification using sparsity-eager support vector machines.* Technical report, Technical report, Drexel University, 2012.