Multi-modal & Multi-level Video Data Characterization & Comparison

Thesis · September 2019

DOI: 10.13140/RG.2.2.17442.45766

1 author:

Saeed Eldah
Lebanese University

2 PUBLICATIONS **0** CITATIONS

SEE PROFILE

MASTER’S THESIS

In order to obtain the

Research Master’s Degree in

In
Information System & Data Intelligence

Issued by:

The Faculty of Science – Lebanese University

Presented and Defended by:

Said El Dah

On Monday, September 30, 2019

Title

Multi-modal & Multi-level Video Data Characterization & Comparison

Supervisors

Dr. Zein Al Abidin Ibrahim, Dr. Isabelle Ferrané, Dr. Sandrine Mouysset

Reviewers

Dr. Ihab Sbeity, Dr. Abed El Safadi

Lebanese University – Faculty of Science – www.fsciences.ul.edu.lb

Master's Thesis

In order to obtain the

Research Master's Degree in

Information System & Data Intelligence

Issued by

Lebanese University - Faculty of Sciences

Presented and Defended by

Said El Dah

Multi-modal & Multi-level
Video Data Characterization & Comparison

ABSTRACT

Multimedia big data are considered the largest and the fastest-growing amount of big data which is considered nowadays the "biggest big data". Moreover, one type of very spread multimedia data is video. With a simple click on a cheap smartphone, you can produce a video and share it on various social media. However, most of the shared videos have no information that describes them unless the user tags the video with a specific type and some keywords that cannot be considered as reliable. The analysis of the content of audiovisual content is the main step for better storage solutions, video recommendation systems, profile construction, duplication detection (for rights issues) and many other systems. Even though the work in this domain is not new, proposals fall short to create an effective multimedia content analysis system, due to multiple issues such as the using of subjective given data such as metadata and tags, or miss using the extracted features, for instance simple sum up for features (ex. Audiovisual content analysis), with no synergy between them. This work is a part of unified series of studies about grouping videos based on semantic similarity induced from multiple modalities of audio, video, metadata, transcripts and social data cooperation between Lebanese University and Université Toulouse III - Paul Sabatier. In this report we provide full details of the work done during the internship period and the planned methodology, along with video content structuring definitions, similarity measure, clustering classification algorithms, and methods used. At the end a comparison of our results with other works done on the MediaEval corpus (MediaEval) is presented.

ACKNOWLEDGEMENTS

I would like to extend my gratitude and appreciation to my supervision Professors Dr. Zein Ibrahim, Dr. Isabelle Ferrané and Dr. Sandrine Mouysset upon whose experiences I leaned on heavily during my internship in Lebanese university cooperation with Université Toulouse III - Paul Sabatier.

Table of Content

Figures

Equations

Tables

Chapter 1: Introduction

In recent times, there has been tremendous growth in multimedia information available on the Web and elsewhere. For instance, hundreds of videos are produced every day across different sources, times, languages, and locations, and the number of videos that are generated in the digital world is increasing exponentially. Furthermore, the overwhelming volume of videos available today, it became a challenge to track the development of the video content, mine their dependencies and semantically organize them [1]. Also, video clustering is a fundamental step for video retrieval, topic tracking, and summarization, targeting for the tracking of videos, either supervised or unsupervised, into clusters of videos according to their contents and themes. However, Video content analysis will make differences in multiple fields such as video recommendation in video in demand services. as well in business, where summarization and topic-based organization of conference/meeting videos, or even the internet video for multimedia content networks such as YouTube and other video-on-demand services. Also, in the security and public safety, where smart cameras for video surveillance can detect suspicious behaviors. This has many implications for big data scientists and experts, a huge proliferation of multimedia content induced several necessities to implement and ensure the continuity of legal and efficient video content exploitation.

Our aim here is to provide a bottom-up approach for feature extractions to enhance videos grouping, organization, and retrieval. We will be focusing on the multimedia content analysis problem. Here, the term multimedia serves primarily to indicate a broader scope of data types to be dealt with, encompasses the entire scope of different modalities, like visual, audio and text, either taken individually or coexisting together in compound multimedia documents [2]. A good example of such a compound document is a video that we refer to in this thesis in general as an image sequence with an accompanying soundtrack.

1.1 Context and Motivation:

Multimedia documents such as images, audio, videos, text, graphics, animations, and other sensory multimodal information have risen at a phenomenal pace in recent years and are almost omnipresent. As a result, widespread attention has been given not only to the methods and tools for organizing, managing and searching such data but also to the methods and tools for discovering hidden knowledge from such data. The task of creating such techniques and instruments is confronted with the great challenge of overcoming the multimedia information semantic gap. But in some sense, the methods of data mining attempt to bridge this semantic gap in analytical instruments. This is because in many circumstances such instruments can promote decision-making. Data mining relates to the method of discovering interesting patterns in information that are not normally available through fundamental queries and related outcomes with the aim of improving decision making using found patterns [7]. For instance, using easy surveillance systems, it may not be feasible to readily identify suspect occurrences. But multimedia data mining instruments that mines trajectories captured from surveillance videos can possibly discover suspect behavior, suspects, and other helpful data. Multimedia data mining introduces both multimedia and data mining strengths and challenges to these areas. In terms of strength, more information is available in most domains. Knowledge can be generally understood of such multimedia data. Some circumstances can also occur where no efficient way is available other than multimodal representation. Moreover, data areas with an exact and unambiguous significance are not well-defined and the data needs to be processed to reach areas that provide information on content. Such processing often results in unusual outcomes with several possible interpretations. In reality, even by a human, multimedia data are often subject to diverse interpretations. For instance, it is not unusual for distinct individuals to have a different interpretation of an image. also, heterogeneous nature is another challenge in the mining of multimedia data. The data are usually the consequence of inputs from different types of sensor modalities requiring advanced preprocessing, synchronization and conversion processes for each modality. The sheer quantity is yet another distinctive element of multimedia data. The high dimensional feature spaces and the size of the multimedia data sets make extraction of features a challenging issue. Multimedia data mining work focuses on addressing these problems while following the typical data mining method.

There are several phases in the typical data mining process and the general method is inherently interactive and iterative.

The main stages of the data mining process are 6:

1. Domain understanding;
2. Data selection;
3. Data preprocessing, cleaning and transformation;
4. Discovering patterns;
5. Interpretation;
6. Reporting and using discovered knowledge.

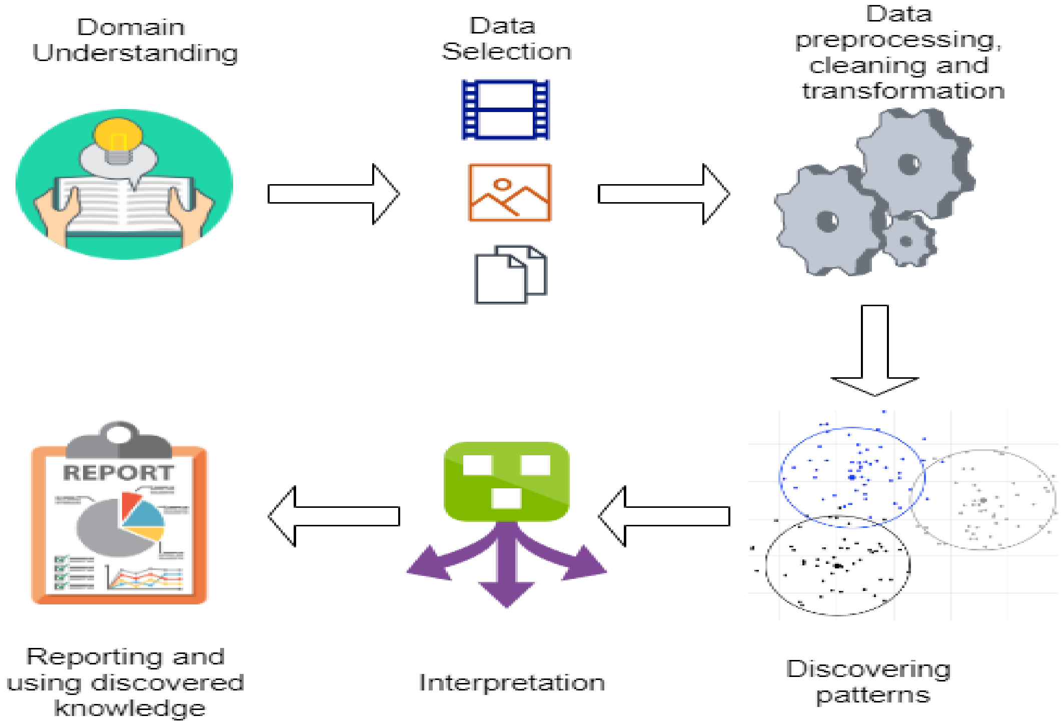

Domain Understanding
Data Selection
Data preprocessing, cleaning and transformation
REPORT
Reporting and using discovered knowledge
Interpretation
Discovering patterns

Figure 1.1: The main stages of the data mining process

Multimedia documents such as video, for example, typically require some pre-processing measures to either identify interesting parts of the video to be regarded as recovery units, or to disclose as much of the temporal content flow structure as possible to facilitate the extraction of features and to be used later in video categorization. An excellent instance of the former is the segmentation of a video into semantic sections or scenes defined by consistent material, and video segmentation into shots can illustrate the latter. A boundary of a shot is a transition from shot to shot. This shift may be abrupt (i.e. a cut), but it can also be accomplished by implementing an editing impact between two shots. Dissolves fade, and wipes are examples of such impacts. Shot boundary detection is referred to as the method of identifying shot limits and acquiring a shot-based video segmentation. However, the primary problem is to fill the gap between the extracted features and the semantic in the generic strategy which is not trivial as it appears. The combination of a feature-based approach and domain expertise is one of the methods.

For instance, to find a news reporter's appearance in TV news, news reporter shots are usually characterized by a fixed studio background, news reporter's faces are the only faces in the program that appear multiple times, and speech signal characteristics such as speed and clearance, noise rate is the same in all shots with the same news reporter. These production rules, however, needs the amount of previous understanding and depends on the application's context and nature. Other simple approaches, on the other side, such as text-based searching for non-textual information (e.g. Google picture search), which is the preferred option that is most used. Although a lot of jobs and amazing thoughts and a lot of effort has been suggested so far in the field of multimedia data mining studies, it is noteworthy that hardly any scheme that has been able to tackle our problem even in the business domain, for instance, YouTube used the name and description as the primary keys for indexing the video until now, but with multilingual information this becomes difficult to connect.

1.2 Scope and Objectives:

Technology's rapid evolution has made production, storage, and sharing of data very simple and at low cost. Besides, multimedia content is now regarded to be the largest and fastest-growing content in the big data ecosystem, deemed to be the "greatest large data." With a simple click on a cheap smartphone, you can produce a video and share it on various social media such as Facebook, YouTube, WhatsApp, …etc., and most of the shared videos over the internet have no information that describes them. However, the viewers of videos can add comments about the video that may contain some keywords about the content and which can be considered as complementary information for any analysis tool. Also, the way we handle and manage multimedia content changes drastically. This leads to the generation of a huge amount of data waiting to be analyzed, consulted or exploited. What all these problems provide is that they describe content analysis as an interpretation of metadata that is far from adequate to indicate the quality of the interactions observed in a video corpus. Also, this has developed some issues for stakeholders with the fundamental use and management of these videos. Efficiently managing and organizing videos has become tedious to deliver precise query outcomes that suit the requirements of customers who generally search for particular content in a video. In the multimedia sector, these issues play a significant part in complicating the lives of investors, video service suppliers, manufacturing firms,1 company owners and the customers, especially with the presence of online video streaming services such as YouTube, Twitch, and Netflix that accumulate an enormous number of customers, more than a billion of users on their own, "almost a third of the web" as they claim. While the above issues could be mitigated by covering easy or naive techniques, our research needs to implement multi-modal (audio-video-metadata-text) use of video modalities in the phase of unsupervised learning to produce a system capable of acknowledging video similarities. In other words, the main focus is on extracting the appropriate set of features that could best represent a video and then applying specific clustering and classification methods to group similar videos together. A preliminary and significant step for any technique of video assessment is to extract from the video a set of characteristics that well represent its content. In this job, we used a multi-dimensional set of three-dimensional characteristics.

- The first dimension handles the multimodality of the features.
- The second dimension takes into account the fact that some features are of a higher level than others. For instance, the presence of blue dominant color is not like a face appearing on the screen or a two-speaker debate.
- The third dimension is about the video segment granularity from which the characteristics are extracted.

1. **Multi-modal characterization**: We mentioned that feature space should include more than one data type to maximize the amount of information it represents. In our work, features are extracted from audio, video and text modalities in addition to the metadata associated with videos. Audio features focus on speaker conversational interactions [3] in which features describe situations involving a number of participants and their roles, the type of interactions they engage in, the type of interventions between them and the context or subject being handled. Video characteristics are obtained from face detection and the visual similarity of the content in the video. While text characteristics are obtained from speech transcripts.

2. **Multi-level characterization**: As a first step, video content was described by low-level features extracted but with a poor meaning on their own. Those features are extracted from the raw signal and then exploited to deduct higher-level information that describes video content more accurately, giving different insights that could be used as input for learning models. For example, data records describing the start and end of each speech segment in the video are utilized to infer a more understandable feature, this feature could describe the speech rate in that video segment, etc.
3. **Multi-granularity characterization**: Traditional ways to characterize a video is to extract some features that describe the content as a whole. They are called global features because they describe globally the content but do not take into account any temporal information. In previous work, instead of describing a video document just as one feature vector extracted over the whole video, they proposed to create a hierarchical set of features, in which they provide a tree-like structure, at each level of this tree, the video is segmented into several parts and the same set of features are recalculated for those individual parts separately. This tree-like representation is a top-down approach in which we start from the video as a whole then we start splitting it into equal parts until reaching parts below a fixed threshold. Using this representation, we would be able to compare segments of videos and extract parts that are similar to what we are looking for, rather than having to search the whole video to find them.

This research adds to a body of prior information, research, and attempts listed in section (1.2), which are required to offer a concrete response on how we could implement an automated process, smart enough to understand the measures or tasks necessary to achieve our objective in this internship. The aim of our work is two folds. In the first fold we re-engineer the way we create the tree-like structure to represent videos. One of the drawbacks of the previous work is that when video is split into equal parts we do not take into account the content when splitting. We propose here to use a bottom-up approach in which the video is already split into non-equal basic units (parts), then these parts are merged in a way to reconstruct the whole video. The basic parts that we start from are the shots detected in the video, where each shot is attributed a p-dimensional feature vector (p is the number of features). In the second fold, we define a distance measure between videos and video units, that acknowledges the suggested video granularity. We used clustering algorithms as an instance at the end of grouping comparable videos together to test the effectiveness of the characteristics suggested and the similarity measure defined. The proposed features may contain redundant features and/or irrelevant ones. therefore. In the next chapter, to carry out such a process, we present the original strategy taken in the past internship.

This report is organized into seven chapters. In chapter 2 we will summarize a literature review about multimedia data mining and we will talk about the previous internship's approaches. Chapter 3 talks about the dataset used. In chapter 4 we will talk about the video characterization, and how we extend the video characterization regarding multi-modality and multi-granularity. In chapter 5 we will introduce our approach and discusses the clustering algorithms that used and video similarity methods. In Chapter 6 we will provide the experiment results and finally in chapter 7 we will list the conclusion and future work.

Chapter 2: Literature Review

Multimedia data is more engaged than traditional data because of the unstructured nature of multimedia data. There are no well-defined data fields with accurate, non-ambiguous significance, and data must be processed to reach areas that can provide information about the content. Such processing often leads with several possible interpretations to non-unique outcomes. In reality, even by human, multimedia information is often subject to diverse interpretations. For instance, it is not unusual for distinct individuals to have a distinct interpretation of a picture. His heterogeneous nature is another challenge in the mining of multimedia information. The information is often the consequence of inputs from different types of sensor modalities requiring advanced preprocessing, synchronization and conversion processes for each modality. The elevated dimensionality of the function spaces and the size of the multimedia datasets make the extraction of features a challenging issue. Multimedia data mining works concentrate on addressing these problems while following the typical method of information mining.

In the following section, we will organize a literature review about the multimedia data mining and we will focus on the video data mining.

2.1 Data mining techniques:

Data mining methods are usually used to perform two types of methods on audio, video, text or picture information [13]

1. Descriptive mining characterizes information in the database's overall characteristics
2. To create predictions, predictive mining conducts inferences on present information.

The following sections are arranged for each of them according to the type of modality and the mining phases. Classification, clustering, association, time series or methods of visualization are used in each modality. In order to get a clearer knowledge of the material in the following parts, we provide an introduction to these fundamental data mining methods.

2.2 Classification:

Classification may be used to obtain models that describe significant classes of information or to predict categorical labels [20]. Such an assessment can assist us to better understand the information as a whole. Classification is a method of two stages. In the first step, a classifier is built describing a predetermined set of data classes called the learning step (or training phase). Here we learn a mapping or a function, y = f(X), that can predict the associated class label y of a given data X. This mapping is shown as classification rules, decision trees, or mathematical formulae. The classifier's precision is the proportion of sample set tuples that the classifier properly classifies. Neural network, Decision tree, Bayesian classifier, support vector machines, and k-nearest neighbors are the most common classification techniques. Bayesian belief networks, rule-based classifier, neural network method, genetic algorithms, rough sets, and fuzzy logic techniques, etc. are the other well-known methods.

The fundamental problems that need to be addressed during classification are

1. Removing or decreasing noisy data, irrelevant attributes and the impact on learning classifier of missing values.
2. It is also essential to select distance function and data transformation for appropriate representation.

A decision tree is a predictive model, which is a mapping of an item's findings to its target value findings. When applied to big databases, a naive Bayesian classifier based on Bayes theorem works well. The Bayesian belief network is used when needed to overcome the weak hypothesis of class conditional independence of the naive Bayes classifier. Support vector machines are probably one of the most commonly used classifiers. They can classify both linear and nonlinear data. The k-nearest neighbor classifier is another simple to introduce but slow classifier. However, these are the classifiers commonly used in the implementation, we can discover many other classifiers in the literature as well.

2.3 Clustering:

Clustering is an approach of grouping the objects into classes or clusters in such a way that objects within a cluster are highly similar to each other but very different from objects in other clusters [22]. It is possible to organize the clustering techniques as partitioning, hierarchical, density-based, grid-based and model-based methods. Sometimes clustering is biased, as only round-shaped clusters can be obtained, and scalability is also a problem. Using Euclidean or Manhattan distance measurements tends to find spherical clusters of similar size and density, but clusters may have any form. Some clustering techniques are susceptible to input information order and may not be able to integrate freshly inserted information at times. Interpretability and usability of clustering outcomes is a significant problem. The high dimensionality of data, noise and missing values are also problems for clustering. Based on the partitioning method, K-means [23] clustering is one of the common clustering techniques. Chameleon and BIRCH [25] are excellent hierarchical techniques for clustering. DBSCAN [24] is a clustering method based on density. Wavelet transform-based Wave Cluster clustering [26] is a technique based on a grid.

2.4 Video mining:

Video mining is a method that can not only automatically extract content and text structure, moving object characteristics, spatial or temporal correlations of these characteristics, but also find video structure patterns, object activity, video occurrences, etc. Moreover, video summary, classification, retrieval, abnormal event alarm, and other intelligent video applications can be achieved through the use of video mining techniques [27]. Video mining is not only a method based on content, but it is also aimed at obtaining semantic patterns. While pattern recognition focuses on the classification of unique samples using a current model

2.4.1 Feature Extraction:

One of the most significant measures is to convert video from non-relational information into a relational data set to apply current information mining methods to video information. Video as a whole is a huge amount of mine data. To obtain information in an appropriate format for mining, we need some preprocessing. Video information consists of spatial, temporal and optional audio characteristics. However, based on the application requirement, all these characteristics can be used to mine pattern and preform predictions. Video is usually built hierarchically from frames (keyframes), shots (segments), scenes, clips, and video in full length. Each hierarchical unit has its own characteristics that are helpful for mining patterns. We can get characteristics such as objects, their spatial positions, etc. from frames, for instance, whereas from shots we can get characteristics such as object trajectories and their movement, etc. It is also possible to use the characteristics among some hierarchical units for mining. Now, based on the application requirement and video structure, we can determine the pre-processing step to obtain either frames or shots or scenes or videos from the video. Spatiotemporal segmentation, for instance, may require breaking the video into consistent collections of frames that can be processed as a single unit to extract features. This is typically performed through an algorithm for shot detection in which the consecutive video frames are compared to determine discontinuity along the time axis. Video structure such as edited video sequences and raw video sequences affects the process of extraction of features. The first step for monitoring video such as raw video sequences is to group input frames into a collection of fundamental units call segment [28]. While shot identification for sports video such as edited video sequences are the first step [29]. in [28] they suggested raw video sequence multimedia information mining framework for segmentation using hierarchical clustering of movement characteristics. The popular preprocessing steps are to extract the background frame, quantize color space to decrease noise, calculate the distinction between the background frame and fresh frames, categorize frames based on the differential values acquired using certain threshold values to determine each category. These popular measures can be configured based on demands such as we want to use some other function instead of color, or we may decide to consider the distinction between two successive frames rather than the background frame, etc. We can use these category labels after categorizing frames.

Color, edges, shape, and texture are some of the low-level characteristics used to obtain higher-level characteristics from each frame or shot or segment, such as movement, items, etc. for video mining. In addition to these characteristics, object and camera movement characteristics can also be used for purposes of video mining. The suggested technique of qualitative camera movement extraction in [29] utilizes p-frame movement vectors to characterize camera movements. We can classify the video into three distinct kinds.

1. Raw video sequences, for example, video surveillance, are not scripted or restricted by regulations
2. Edited films, e.g. drama, news, etc. Are well organized but with intra-genre variations in manufacturing styles that differ from nation to nation or content creator to content creator.
3. Sports video is not scripted but regulated. In our study, we are not covering medical videos (ultrasound videos including echocardiogram).

2.4.2 Video mining techniques

Video mining has two levels. One is semantic data mining that is immediately at the stage of the function: e.g., the occurrence of primitive activities such as speaking about an individual X. Another is higher-level patterns and information mining. Patterns can represent cross-event relationships: for example, during office hours, individual X often speaks with individual Y. The following sections describe video mining methods.

2.4.3 Video Classification

Video classification mining approaches classify video items into predefined categories, depending on the shot histogram, movement characteristics of moving objects, or other semantic descriptions. The semantic descriptions or characteristics of each category can, therefore, be used to mine the implicit patterns between video objects in that category.

Video sequences comprising hand or head gestures are categorized in gesture (or human body movement) recognition according to the behavior they represent or the messages they try to express. The gestures or body motions may represent, for example, one of a fixed set of messages such as waving hello, goodbye, and so on [30], or they may be the various strokes in a tennis video [31], or in other cases they may belong to the dictionary of some sign language [31], etc. [32] created a system to enhance the vocabulary of learned ideas iteratively by using classifiers learned in previous iterations to learn composite, complicated ideas. The semantic descriptive capacity of their approach is therefore bound by the collective, multilingual vocabulary of hundreds of millions of web users who tagged and uploaded the videos, not by the intuition of a single designer. In [33], the Associative Classification (AC) approach produces classification rules based on the correlation between distinct value pairs and concept classes. In [44], the rough set theory-based method is used to obtain various learning definitions of the case. It overcomes the issue of mistakenly choosing interesting event as adverse examples for training and measures similarities in a high-dimensional feature space properly.

2.4.4 Video Clustering

The methods of clustering organize comparable video items into clusters through their characteristics. So, it is possible to use the characteristics of each cluster to assess the video shots containing those video items. It is possible to label or visualize these clusters. In the pre-processing phase, clustering is also very essential to remove noise and perform the conversion. As indicated in [34], the raw video sequences are segmented by clustering. The pioneering application in video clustering is a clustering of shots [35] [36], a clustering of still pictures (key frames) [38] [39] [40], and then effective indexing, searching and viewing the application. A hierarchical clustering of key frames based on various picture characteristics was suggested in [41] to improve search effectiveness.

2.5 Previous Work:

This job is part of a sequence of past internships, each focusing on particular issues, but none of them did the assessment on the video shot level. Even if we used the prior code, models, and outcomes that were helpful in removing some methods based on their outcomes or saving time in applying some scripts, we had to re-execute the code and validate the outcomes and code.

2.5.1 (2016) internship [4]:

In this internship the main focus was the re-engineering of the features used to represent videos in previous works in the same research team. The work’s focus was on two parts. The first part is aimed at exploiting the content of video files so that they can be described as a collection of appropriate characteristics. This step is achieved by implementing different duties and methods independently of the source of the information, which allows the research to be incorporated into any out their video corpus. Although, in this former study those features were either extracted from the set of metadata accompanied by the raw videos or obtained by applying automatic processing tools on the audio and video components. This information and tools were provided by an existing corpus we will describe later on. The main outcome of this part was, based on the previously proposed set of features, propose a new set of normalized features that can be calculated at any granularity and propose new other features.

The extracted features are structured into four distinct modalities, irrespective of the source, as it offers sufficient information to be able to build these modalities.

1. Audio: (sounds, music, and speech from which speech turn and speakers could be extracted, in addition to text data through transcription of what has been said, etc.),
2. Video: (identifying the presence of people, how many, etc.),
3. Metadata: (duration, title, tags, etc.)

4. Social data: (twitter feed of users talking about the video)

So typically, a video is a vector of values corresponding to each type of heterogeneous feature sets. This extracted information is depicted in two consecutive levels:

1. low-level descriptors: containing temporal sequences for each feature considered.
2. high-level descriptors: acting as an abstract depiction of the ratios derived from the past level, in addition to the different modalities.

For instance, in a video containing a dialog between two people, there would be a sequence on the low-level descriptors describing the exact timestamps that each person started or stopped speaking:

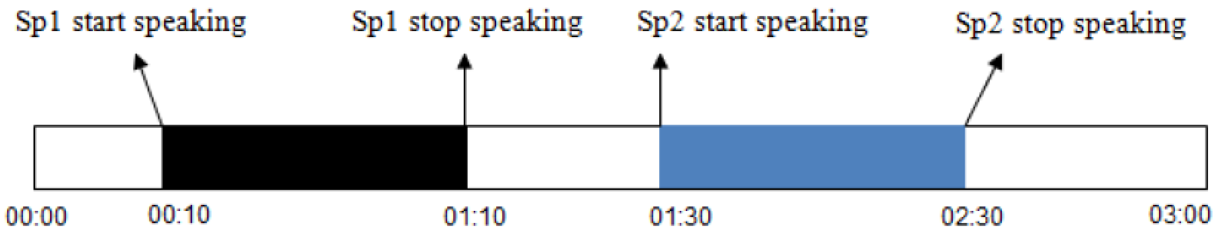
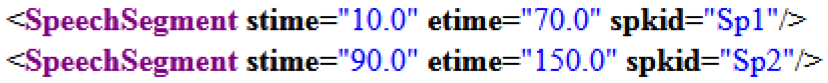


Figure 1.2. Low Level Speaker Turn Representation

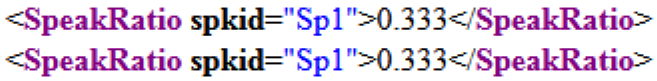
Figure 1.2. shows a simplistic representation of a video (3-minute duration) that contains a simple conversation between two people, which are considered as speakers in the study. The speaking segments for each speaker (Sp1 & Sp2) are represented with a starting and ending points in the video. This information could be conveyed by using different technologies, but for the sake of clear and robust representation we display how they could possibly represent this information as an xml format in Figure 1.2 and Figure 1.3:



```
<SpeechSegment stime="10.0" etime="70.0" spkid="Sp1"/>
<SpeechSegment stime="90.0" etime="150.0" spkid="Sp2"/>
```

Figure 1.3. Xml Representation of Low-Level Descriptors

Figure 1.3. is a deprecated view of a section of the xml file that could describe low level features. Then, on the higher level, descriptors show the ratio of speaking time for each person relative to the duration of the video:



```
<SpeakRatio spkid="Sp1">0.333</SpeakRatio>
<SpeakRatio spkid="Sp1">0.333</SpeakRatio>
```

Figure 1.4. Xml Representation of High-Level Descriptors

This sort of description allows them to acquire fundamental information, serving as the document's fundamental facts, so they can then adapt and deduce views to serve any sort of application they want. For example, from what is listed in Figure 1.3 they could also generate a new high-level feature describing the speaking time ratio relative to the duration of the video, regardless of who is speaking, which would be two-thirds of the total duration.

The second part of the work aimed to establish distance comparison methods that define the similarity between two documents based on their content description provided by a vector space of high-level descriptors. The similarity between videos is not constricted only to visual similarity, it is divided between other modalities, so it can be measured by a metric defined on a feature vector space for each separate modality on its own:

$$d(V_1,V_2) = \alpha \times d_{audio}(V_1,V_2) + \beta \times d_{video}(V_1, V_2)$$

Equation 2.1 Multi-Modal Distance Metric audio, video

The similarity value d (V1, V2) would be a weighted combination of the different modalities, where the result is non-negative, equal to 1 when videos V1 & V2 are identical, and close to zero when V1 & V2 are completely different. One could also depend on dissimilarity (or dissimilarity coefficients), likely so, they are non-negative numbers d (V1, V2) that are close to zero when V1 & V2 are near each other and this coefficient increases as V1 & V2 grow different. After deciding on the comparison methods, video documents are grouped together by applying unsupervised methods, k-means and spectral clustering [8]. Regardless of the interesting results obtained, the work done during previous internships [4] presents the following limitations:

- Some features spaces are different. For example, most of the features represent ratios (values between 0 and 100) while others represent real values (min and max number of faces).
- No tests have been done to recognize feature relevancy.
- Only two modalities were used (audio, video) to represent video documents.
- Features are extracted from the whole video even if the work has proposed to extract features at different granularities. That is when the whole video is divided into smaller parts based on a time threshold (1min, 5min, 10min, …) or on video duration ratio (1/2, 1/4, …) which is necessary for comparison between parts of videos together, instead of just comparing videos as a whole (This is referred to as multi granularity description).

- Experiment done on a small dataset, so the need arises for expanding the study to include a bigger dataset to incorporate more diversity and obtain more concrete results.

2.5.2 (2017) internship [5]:

In this internship, they had aimed to overcome these limitations and to increase the applications that can benefit from the video characterization and similarity measure. So, they added the text feature to the vector but they didn’t use them in the clustering algorithms, they were used in the video summarization.

Moreover, in [5] they achieved the following objectives:

- Introduce a representation of features derived from transcripts and social media.
- Propose Tree-Like representation of video documents, by dividing videos into several segments at each level of the tree structure and calculating the feature vector for each segment independently.
- Propose multi-modal-multi-level similarity measure that takes into consideration the different modalities in addition to the hierarchical granularity of video documents.
- Generate clustering results on the bigger dataset provided, using different clustering algorithms.
- Feature selection based on feature’s relevancy by studying the effect of each individual feature on the clustering results.

2.5.3 (2018) internship [6]:

In this internship they developed methods to measure video similarities, that can be classified into feature matching, text matching, ontology-based matching, and combination-based matching. The choice of method depends on the query type, they used the feature matching approach. It measures the average distance between the features of the corresponding frame. But the problem is semantic similarity cannot be represented because of the gap between sets of feature vectors and the semantic categories familiar to people.

in [6] they used feature matching method, they took a 24 vector to represent each video and compared it to another video, using the clustering algorithms, and then they reapply the algorithms for doing this using audio, and video features separately.

The main work done here is to re-engineer the pipeline done in previous internships and apply it on the whole dataset. The hard work was to extract the features on the 15000 videos and finding the bugs in the code and apply the clustering on the 15000 videos.

2.6 Discussion

The traditional strategy to video mining utilizes a label vocabulary to know and find characteristics and classifiers that can best differentiate these labels. This approach is inherently limited by the hand-selected label set. The classifiers by definition can only learn concepts within this vocabulary of labels. It cannot scale well for a diverse and enormous multimedia dataset of web-scale. Future study will, therefore, focus more on discovering methods for auto annotation for such large-scale multimedia information. Naturally, viable video mining algorithms are not restricted to specific applications within the above-mentioned kinds. On the one hand, a universal algorithm cannot be found for a particular sort of video. Depending on the video content, what characteristics are used to mine and what sort of mining algorithms are used to mine based on the application's requirements. On the other side, multiple methods are feasible to meet the requirements of the same implementation. It is therefore vital to have a video mining structure that can manage these differences. The literature covers many distinct applications of video mining. But still, in terms of the size of information set on which such mining can be performed, we see the greatest restriction. Since video information is very voluminous itself, there are no works considering many big videos. Video data mining utilizes audio and captioning characteristics.

In the next chapter we will talk about the user data set and the main elements in this dataset.

Chapter 3: The Dataset “Blib10000 – MediaEval”

In addition to providing a platform for developing and validating new algorithms, extensive, comprehensive and publicly available internet video datasets can also be a valuable resource that can be used to analyze user behavior. It is possible to define exciting and novel issues based on a big representative; a dataset made up of contributions from many users [9].

There are several elements to the demands for a well-designed Internet video dataset. First, the strategy for information collection should be well-intended to guarantee adequate scale and variety of content. In a way that is as unbiased as possible, the dataset can depict internet videos. Second, naturally, internet videos are linked to multi-modal data, this data should be comprehensively gathered and well-structured to evaluate the contribution of a specific information resource to a particular assignment. Specifically, social network development offers wealthy background data for internet videos. In many areas of information retrieval, social data has proved useful [10], demonstrate its interesting internet video analysis potential. Third, the definition of duties and ground-based truth generation should be based on real use-case scenarios and should also be suitable for metric-based assessment so that techniques extracted from the dataset can be similar to each other over time. The previous problems were the main motivation to create “Blip10000” dataset, consisting of 14,838 videos from blip TV for a total of 3,288 hours [9]. The “Blip10000” dataset includes blip.tv videos. Users who have gone beyond the point-and-shoot techniques of capturing video prevalent on platforms such as “YouTube” and “Flickr” have developed the blip.tv content. Moreover, blip.tv contributors demonstrate at least basic proficiency in filmmaking. Such content is usually referred to as user-generated semi-professional (SPUG) content that tends to be scripted or well thought out. It is generally intended to communicate a message or opinion or entertainment specifically. Users of blip.tv publish video content in a series on a specific subject, publish it regularly and target a wide audience. These shows cover a variety of subjects and styles [9]. In addition to a large number of videos available, this corpus is characterized by the heterogeneity of the video files, which cover a wide range of video types. 26 categories were taken into account to categorize each video among them. The category of each video published on blip.tv is given directly to the individual making the video and posting it on the site. This implies that this data is not necessarily accurate or subjective, and videos without a category are linked to the default category.

The dataset split into two subsets. Dev contains 5288 videos and test contains 9550 videos, Figure 3.1 was given by the dataset provider [9], which shows the distribution in genres of the videos between the development and testing sets. While figure 3.2 shows the distribution by duration [6].

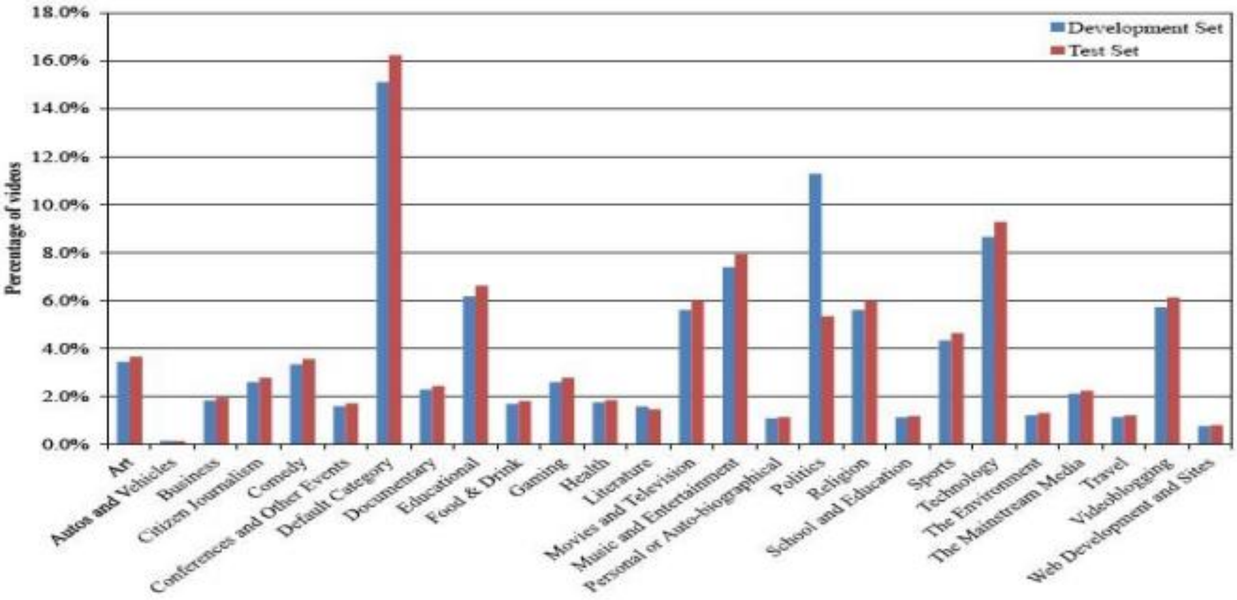


Figure 3.1 Distribution of videos between dev and test sets according to their categories

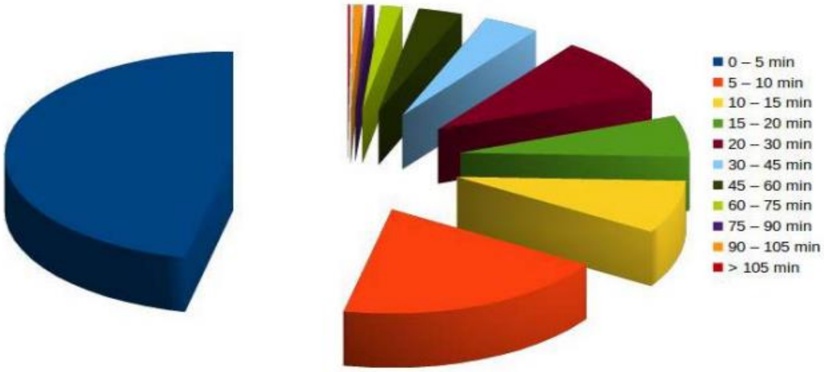
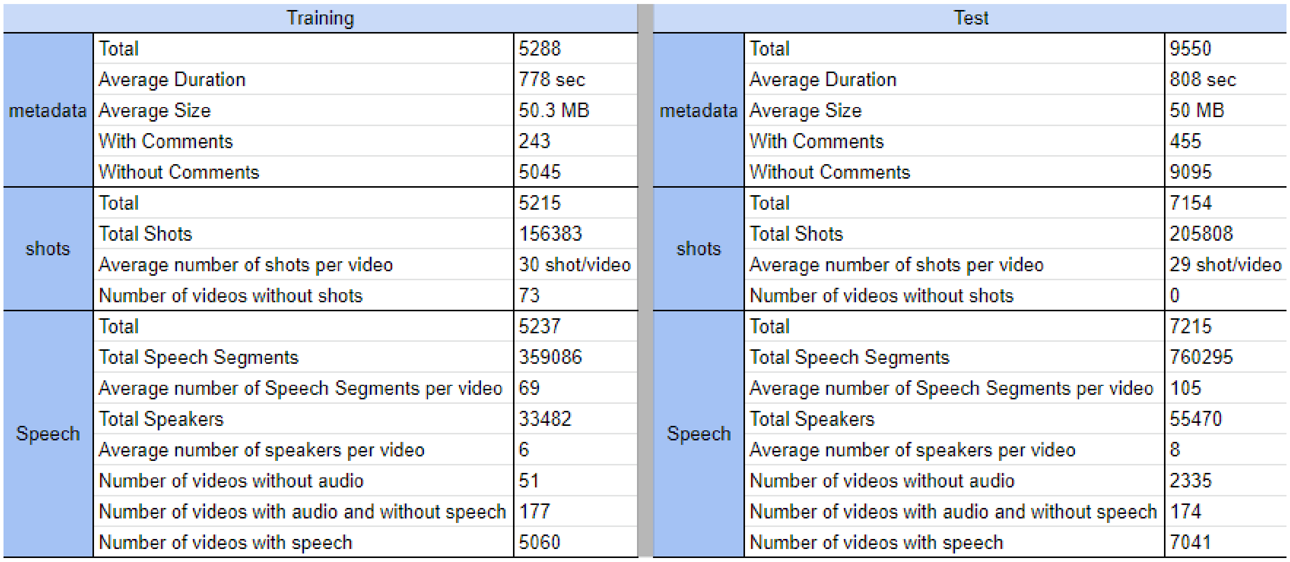


Figure 3.2 Video distribution by duration

Furthermore, we have applied some statistics on the dataset to display a wider image of the dataset, especially on the level of the shot. Table 3.1 shows the shot and speech segmentations distribution over the dataset in both Dev and Test data.



Training			Test		
metadata	Total	5288	metadata	Total	9550
	Average Duration	778 sec		Average Duration	808 sec
	Average Size	50.3 MB		Average Size	50 MB
	With Comments	243		With Comments	455
	Without Comments	5045		Without Comments	9095
shots	Total	5215	shots	Total	7154
	Total Shots	156383		Total Shots	205808
	Average number of shots per video	30 shot/video		Average number of shots per video	29 shot/video
	Number of videos without shots	73		Number of videos without shots	0
Speech	Total	5237	Speech	Total	7215
	Total Speech Segments	359086		Total Speech Segments	760295
	Average number of Speech Segments per video	69		Average number of Speech Segments per video	105
	Total Speakers	33482		Total Speakers	55470
	Average number of speakers per video	6		Average number of speakers per video	8
	Number of videos without audio	51		Number of videos without audio	2335
	Number of videos with audio and without speech	177		Number of videos with audio and without speech	174
	Number of videos with speech	5060		Number of videos with speech	7041

Table 3.1 shot and speech segmentations distribution

Some extra information acquired from other platforms, such as social media, is retrieved from social networks such as Twitter in the form of remarks and other annotations placed online by users and viewers.

We have analyzed the dataset before we started using it so we could have more information about its structure and content. The data was not cleaned, modalities especially the audio is missing for some videos, which leads to the absence of the provided features based on the missed modality such as speech transcripts etc. So, the first stage was to clean up the dataset before starting the processing and the validation of its structure.

Chapter 4: Video Characterization

In video data mining and retrieval systems, video representation methods are created as a preliminary phase. As with most approaches to data analysis, video information requires a specified framework, so one of the most significant functions concerned is transforming video information from unstructured information into a structured information collection. Moreover, the first step is to apply video track segmentation into smaller, more manageable tracks, thus allowing video indexing and extraction of characteristics (characteristics and descriptors are interchangeably used) for those larger sections known as video shots.

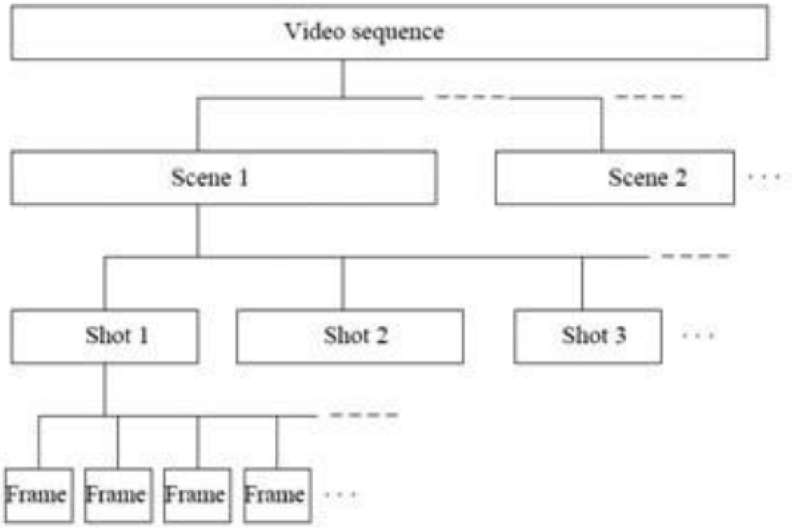


Figure 4.1 Common video segmentation

Scene: It is a compilation of semi-related and temporally adjacent organizations that depict and convey a high-level idea with a series of shots concentrating on the same point or place of concern.
Shot: is described as a series of frames taken by a single camera without significant visual content modifications.

Frame: is the images that compose a complete shot.

So, we can agree that the most basic step in video processing is to partition the video's length into a set of smaller sequences called shot and discover each shot's keyframe to allow users to annotate and extract features linked to each shot.

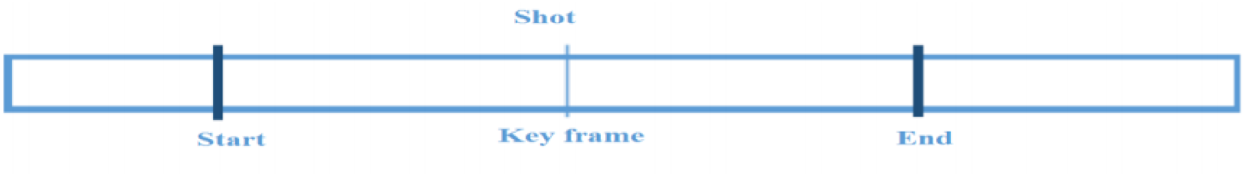


Figure 4.2 Diagram showing shot segmentation

Figure 4.2 demonstrates a video shot structure represented by the beginning and end limits and the keyframe representing this shot, obtained from the center frame, as stated above. Of course, the significance of incorporating multimodality into characterizing multimedia material can be inferred, hence linguistic relevance. As mentioned in the previous internships, [4] & [5] have already explored and defined audio and video features. In this chapter, we will present these features. Audio data could generally include all kinds of environmental and natural sounds, voices, and music that might be beneficial in identifying homogeneity. This research study, however, uses the data to enrich the audio modality with descriptors supplied through speech recognition instruments, a set of instruments devoted to speaker diarization, voice interaction/intervention, and speech/music segmentation. Next, video data also offers a significant investigative medium. Using face detection algorithms, spatial shot sequence recognition, shot transitions, and inter-intra-intensity differences between them, video features are defined.

4.1 Low Level Descriptors:

4.1.1 Audio Descriptors:

1. Speakers List: all speaker data that contains speakerId, gender, language, the additional length of speech and inactivity used in speaker identification.
   1.1. Extent is the range for each speaker which is equivalent to the difference in the period between the speaker's first intervention and his last intervention.
   1.2. Inactivity is the sum of all speech segments in which this speaker is not speaking.
2. Speech segmentation: is described by a temporal list of segments in addition to the total speech duration, each list element specifies the spoken words and their timestamps in the video, assigned with speaker identification and.

3. Music segmentation: temporary list of music segments specifying where music is present in the video and the assigned timestamps.
4. Interactions and Interventions: are higher-level descriptors obtained from low-level signals and speaker segmentation data [3] and then used as intermediate information to deduce higher-level descriptors:
    4.1. Interactions: temporal list of elements, each element describing the existence (start and end) of a sequence of interactions between speakers (sequence is denoted by a set of speakerIds).
        - Interaction is the communication between two or more speakers in a manner that the time gap between their speech segments is smaller than a threshold ε as shown in Figure 6, where [sp1, sp2, sp1] form an interaction sequence.

    4.2. Intervention: is the act of interfering in a discussion where each segment of speech that is not a component of communication is treated as an intervention. Short intervention occurs when the duration of the voice segment is less than a certain threshold, and long intervention occurs when its threshold is greater than that. In In Figure 6, sp3 is an intervention as it is isolated from the interaction list [sp1, sp2, sp1] (gap> threshold). Note that an intervention is an isolated speech turn that is segregated by a gap higher than the defined limit from other speech turns before and after.

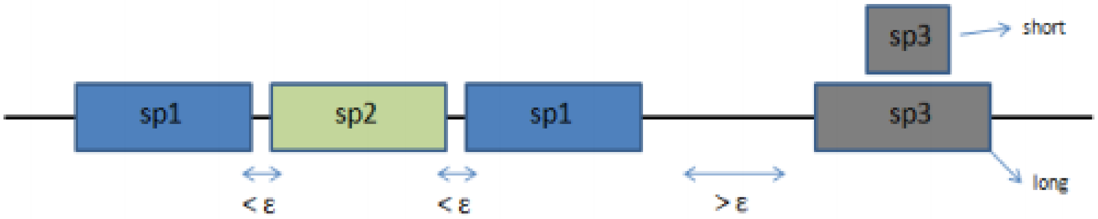


Figure 4.3 Diagram showing the notions of interaction and intervention between speakers

4.1.2 Video Descriptors:

In addition to video title, video file name, video duration, document size, frame rate, etc. There is a descriptor that lists video shots and their keyframes in temporal order:

Keyframe is one of the frames in a shot that is used to represent it. Most of the works in the literature use the middle frame as keyframe since the first keyframe or the last one may not represent well the shot especially if we have a problem of detection of the boundaries of the shot. However, some techniques use more than one keyframe to represent the shot such as the first, the last and the middle one. We should also mention that are some approaches that extract automatically several keyframes (number not fixed a priori) from the shot. In Figure 7, we show a shot from which the middle keyframe is used to represent it.

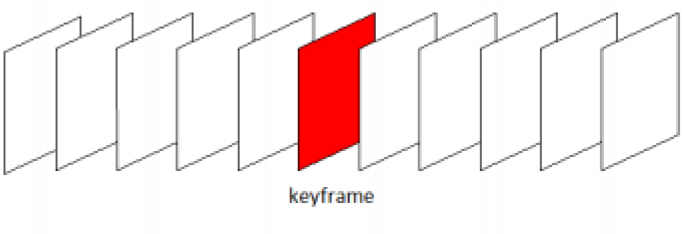


Figure 4.4 Schema representing the keyframe of a shot

each shot has start and end timestamps, keyframe index and its occurring time regarding video duration. On each keyframe, we have applied a face detection algorithm in order to detect the faces and calculated the sum of face areas in that keyframe we have calculated an 9x9 intensity matrix of that keyframe.

Intensity matrix, is obtained by dividing the keyframe into 9 blocks (3 x 3) and then calculating the average intensity of pixels in each block. The detected faces and intensity matrix will be used later to generate high level descriptors.

4.2 High Level Descriptors:

From the low-level descriptors, higher-level descriptors are computed as follows:

4.2.1 Audio Descriptors:

1. Speaker diarization: speakers are classified into five distinct types, based on two parameters, scope, and activity included in the descriptors at the low level. Five research-related descriptors were generated from these two parameters [3]:
    1.1. Punctual: are the speakers appearing only in one section, so their activity is equivalent to their scope. These speakers do not intervene in other places in the video, almost never.
    1.2. Regular: refers to speakers who are not very active but have a long-time span. They are speakers who appear on brief, remote sections. This sort of speakers typically corresponds to an advertiser's interventions or a sequence such as a frequently returning short data flash.
    1.3. Present: these speakers are the most numerous. These speakers are not very active and have a tiny amount of time. Usually, they are confined to a portion of the document. E.g. This category includes columnists, reporters or interviewees.

   1.4. Localized: speakers still have a less significant degree of activity. This kind is characteristic of speakers who speak a lot but are very dense and very localized. This is usually the case with individuals interviewed in a newsletter.
   1.5. Important: there is an important activity of speakers. Given that, they are very present in the document and used to locate presenters, key guests and more usually significant speakers.
2. Speaker distribution: this descriptor's aim is to understand how speakers are distributed in the video sections. This descriptor is equivalent to the proportion of the number of distinct speakers speaking in a portion of the video, multiplied by 100 over the complete number of speakers. For instance, if 10 speakers are recognized in a document and 5 of these speakers are involved in a section of this document, this value would be equivalent to 50%. This type of descriptor will allow us to know how the speakers intervene in the document.
3. Speaker change rate: this descriptor was suggested in the video or portion of it to obtain speaker change rate. It measures how often, during a video part, there is a shift from one speaker to another speaker. This amount is split by the video length and multiplied by 100 to guarantee values ranging from 0 to 100.
4. Interaction rate: this is the percentage value of speech that forms part of a sequence of interactions over the video's total duration. Several speakers can intervene in an interaction sequence, carrying their amount of data about the sort of this sequence. A series of interactions between two speakers, for instance, may constitute a sort of discussion or question/answer. On the other side, it may be a discussion to have a series of interactions between four speakers. The following four descriptors were deducted from the interaction level for this purpose:
   4.1. 2 speaker’s interaction rates: The proportion of speech that is composed of the series of interactions between two speakers.
   4.2. 3 speaker’s interaction rates: The proportion of speech segments which are part of a series of interactions between three speakers over the complete period between three speakers.
   4.3. 4 speaker’s interaction rates: The proportion of speech that is composed of the series of interactions between four speakers over the total duration of speech segments between four speakers.
   4.4. 5 or more speaker’s interaction rate: The proportion of speech segments forming a part of the series of interactions between five or more speakers.
5. Rate of intervention: this is the percentage of speech segments that do not form part of a sequence of interactions divided by the total duration of a video. There are two kinds of interventions:
   5.1. Short isolated intervention rate: The proportion of speech segments that are part of short-isolated interventions. It is recalled that a short-isolated intervention is any voice turn that is not part of an interaction sequence and its length is less than a fixed threshold.

5.2. Long-term isolated intervention rate: In contrast to short intervention, it is the proportion of speech segments that are parts of long-isolated video-duration intervention. A long-isolated intervention is any speech turn that is not a part of an interaction sequence and is longer than a fixed threshold.

6. Segmentation of speech/music: six descriptors are extracted from the segmentation supplied by the BMP tool created by [15]. All the descriptors below are divided by the duration of the video and multiplied by 100 to insure values between 0 and 100.

6.1. Speech Rate: Represents the segment's proportion of speech. It is the ratio of the amount of speech segments in the video.

6.2. Music Rate: Represents the segment's proportion of music. It is the ratio of the amount of music in the video.

6.3. Music and non-speech rate: Represent the percentage of segments in which we have music and no speech occurring simultaneously in the video. It is the ratio of the sum of the intersections between segments of music and segments of non-speech throughout the video.

6.4. Speech and non-music rate: Represent the proportion of speech segments that does not contain music. It is the ratio of the sum of intersections between segments of speech and segments of non-music throughout the video.

6.5. Music and speech rate: Represent the proportion of speech segments containing music. It is the ratio of the sum of intersections between segments of speech and segments of music.

6.6. Non-music and non-speech rate: Represent the proportion of non-speech segments occurring at the same time with non-music ones. It is the ratio of the sum of intersections between segments of non-speech and non-music segments over the video.

4.2.2 Video Descriptors:

Besides the video title, name, length and size of the video document. There is a descriptor listing video shot and their keyframes:

1. Face Detection in Keyframes: A keyframe is one of the frames used for representing a shot. Most literature works use the middle frame as a keyframe since the first or last keyframe may not depict the shot well, particularly if we have an issue detecting the shot's limits. On each keyframe, we apply a face detector to detect how many faces are present in the keyframe. However, a video or part of a video may be composed of one or more shots. If we adopt the one-keyframe-per-shot strategy, the video (or part of video) will be represented by the set of keyframes of all its composing shots. In this case, the video (or video part) will be represented by a vector of number of faces detected on all its keyframes. Moreover, we compute the two following descriptors to represent the visual part of the video by a one-value descriptor:

1.1. Minimum face number in a video portion: represents the minimum number of faces found in the series of keyframes considered.

1.2. Maximum number of faces in a part of the video: Represents the largest number of faces recognized in the considered keyframe sequence.

1.3. Mean average areas

1.4. Standard deviations

2. Shot Intensity Variation: this is the last descriptor form from which two descriptors are obtained from color intensity variation in a video. Then the variation in intensity can be calculated from keyframe to keyframe. As mentioned earlier, the image or keyframe is divided into 9 blocks and on each block, the average color intensity is calculated. Then, it is necessary to process and compare the blocks of the same position in the two successive keyframes. As a result of this comparison, two types of descriptors are proposed: inter-segment variation and intra-segment variation. The first reflects the variety of intensities from one video segment to the next video segment while the second measures the variety of intensities in the same (or video portion) video segment.

2.1. Inter-segment variation: The concept is to create an absolute difference between the 9 blocks of two keyframes, one from a video segment and the second from the next video segment. Since a video segment may contain more than one shot which means more than one keyframe, we have calculated a weighted average of the 9x9 patches of all the keyframes in a video segment. The weights are proportional to the coverage proportion of each in the video segment. For example, a video segment S1 consists of 3 shots, the first of which covers 20% of the segment, the second covers 70% of the segment and the third covers the remaining 10%. The average keyframe is then equal to $0.2\,KF_1 + 0.7\,KF_2 + 0.1\,KF_3$. The intensity variation between the 9 patches of the average keyframe in a video segment S_i and the 9 patches of the next average keyframe in a video segment S_{i+1} is calculated.

2.2. Intra-segment variation: Measures the median intensity variation in the same video (or video segment). The weighted average of the 9x9 patches of all keyframes in one segment is calculated as outlined in the inter-segment variation. The descriptor is then calculated as the temperature distribution over the weighted average of all the keyframes of the video (or the portion of the video). For instance, the descriptor is equal to given a part of video S having n shots and we calculated the weighted average KF keyframe.

Chapter 5: Our Proposed Approach

As mentioned before, the aim of work is to adopt a bottom-up approach instead of top-down to represent and compare videos. More specifically, our idea is to extract the features on the shot level (bottom-up) and apply a supervised and unsupervised learning on the extracted features in order to highlight the enhancement of our approach. The features extracted are done on the Dev and Test sets. Moreover, we have re-engineered the pipeline composed of the following steps:

1. Video Representation
2. Feature Extraction and Normalization
3. Similarity Calculation
4. Clustering and Classification
5. Result Statistics

5.1. Video Representation:

Video Representation is defined as the process of extracting important or useful information or what can be extracted because at this stage, we may not be aware of what might be useful and represent it in a clear format. As we mentioned before, each video consists of number of shots, in [6] internship they extracted the features for the whole video as single segment, however what we did is that we extracted the features on the shot level and then we calculated the video feature vector by taking the average vector of the shots:

5.1.1 Meta Data:

1. Duration: the duration of the shot

5.1.2 Audio:

1. Speakers Interactions: interaction between two, three, four or more speakers
2. Intervention: temporal list of elements, each element describing the existence (start and end) of a speaker intervention, in addition to its type (short, long).
3. Speakers type: the type of the speaker punctual, localized, present, regular and important
4. Speaker distribution:
5. Speech: for each speech we save the channel type start and end time, the languages. And the time sequence for the transcript
6. music: the start and end time of the musical parts in the shot
7. speech_with_music:
8. speech_with_non_music:
9. non_speech_with_music:
10. non_speech_with_non_music:

5.1.3 Visual:

1. mean_number_of_faces
2. std_number_of_faces (Standard Deviation)
3. inter_intensity_variation
4. intra_intensity_variation

5.1.4 Text:

1. Number of words

5.2. Feature Extraction and Normalization:

In this step, we extracted features for the videos using two different approaches: the bottom-up approach which is on the shot level and top-down approach which is on complete video level video for comparison issues. After extracting features on the shot level, we calculated the video features by taking the average of the shot feature vectors.

$$\boldsymbol{v}_{vid} = \frac{\sum_n \boldsymbol{v}_{sh}}{\mathrm{n}}$$

$Where:$
$n\ is\ number\ of\ shot\ in\ video$
$\boldsymbol{v}_{vid}\ feature\ vector\ of\ the\ video$
$\boldsymbol{v}_{sh}\ feature\ vector\ of\ the\ shot$

After the feature calculation we normalized the extracted features by making each value between 0 and 100.

5.3 Clustering and Classification:

In order to test the efficiency of our approach we have tested several clustering and classification algorithm and we compared the results in the experimentation section.

We have applied the following clustering and classification algorithms:

Clustering:

1. K-Means
2. DBSCAN
3. BIRCH
4. Mean Shift
5. K-Medoids

Classification:

1. K-Nearest Neighbor (kNN)
2. Support Vector Machine
3. Random Forest
4. Neural Network
5. Naïve Bayes
6. Logistic Regression
7. AdaBoost

To read more about the used algorithm please check Appendix A.

5.4 Similarity Measure:

One of the video content analysis steps is to compare a video to another one or to search for a shot into a video. Moreover, clustering videos into groups having the same content is also one of the analysis steps that is very used. Such algorithms rely on a defined distance measure between the objects to be compared. This measure could be the proximity of similarity, which measures how much objects resemble each other, or of a dissimilarity, which is the opposite, measuring how much the objects are far from each other. In the previous work, it had been established as a measure that takes into account the whole video described as one feature vector. However, it wouldn't fit into our representation of a video document, hence we had to define a new distance measure that respects the multi-granularity representation. So, in this chapter, we present our approach by defining a measure that takes a pair of videos as input, $S(v1, v2)$, and outputs the proximity measurement between them based on their inter-segment associations. The motivation for this approach lies in the fact that video documents are now divided into several segments that one could choose from to possibly obtain the highest level of relevancy between different parts of the videos, or to recognize which parts of a video hold the most significant amount of information (related to the feature set selected) compared to another video or even to itself. Now, if we want to use proximity measure to cluster videos as an application example, and given a dataset of n videos to be clustered, the distance proximities should be available for all the pairs of videos. These values can be arranged in an $\boldsymbol{n} \times \boldsymbol{n}$ matrix, which is called a proximity matrix, where the rows and columns are the same set of videos. Since in our video characterization chapter we have proposed to represent a video by a tree of feature vectors, we should define a similarity measure or distance between two trees associated with two videos. This chapter will focus on the definition of what we call it the multi granularity distance which we will validate it in the next chapter for clustering issues.

In this section we will list our approach methods to calculate similarity between two videos, we have developed three similarity approaches:

1. Shot-matching distance
2. Common clusters approach
3. Common shots approach

5.4.1 Shot-Matching Distance

In this approach we invented a new way to measure the similarity between two videos, this method depends on the distance between the shots of two videos.

The matching distance can be well understood through the following algorithm:

1. For each shot in V1 calculate the distance with V2 shots

   Where $d(S_1, S_2)$ is the Euclidean distance between $S_1\ and\ S_2$:

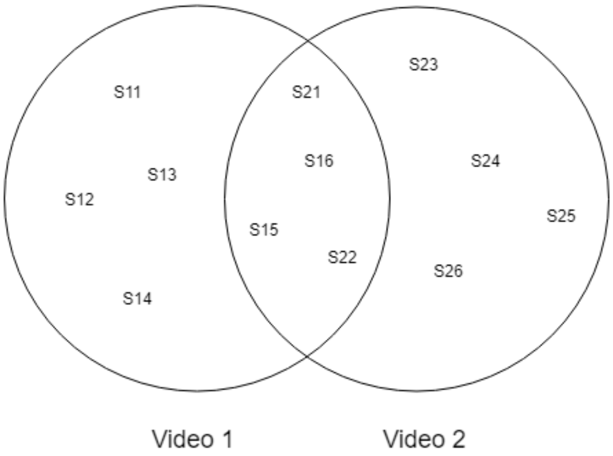
   $$d(S_1, S_2) = \sqrt{\sum_{i=1}^{n} (S_{1i} - S_{2i})^2}$$

2. Create the distance matrix between
3. Accumulate the minimum values as following
4. For each cell in the matrix

   If value = 0 then delete the cells that is in the same row and column
5. Search for minimum value that is not deleted and $dis = dis + minimum$ it, then delete the cells that is in the same row and column.
6. Repeat step 5 until delete all rows or columns
7. Here we have two cases
   a. the shot number in both videos are equal
   b. the shot number in both videos are not equal, in this case we will handle the extra shots as special case, so we will calculate the distance between those extra shots and each shot and then we will select the minimum distance
8. Repeat steps 6 until delete all rows or columns
9. $d(v1, v2) = dis$

$$where\ dis\ is\ the\ sum\ of\ minim\ shots\ distances\ for\ two\ videos$$

Example:



In the figure we have two videos:

$V_1\ = \{s_{11}, s_{12}, s_{13}, s_{14}, s_{15}, s_{16}\}$

$V_2\ = \{s_{21}, s_{22}, s_{23}, s_{24}, s_{25}, s_{26}\}$

By applying the previous algorithm, we will have the following matrix:

First, we will fill the distance matrix and delete the 0 cells rows and columns

V1 V2	S11	S12	S13	S14	S15	S16
S21	~~0~~	~~10~~	~~8~~	~~15~~	~~0~~	~~0~~
S22	~~20~~	~~0~~	16	11	~~3~~	~~30~~
S23	~~5~~	~~7~~	3	24	~~12~~	~~32~~
S24	~~3~~	~~20~~	21	1	~~3~~	~~9~~
S25	~~10~~	~~13~~	23	2	~~5~~	~~2~~
S26	~~6~~	~~12~~	22	25	~~4~~	~~25~~

Second, we will search for minimum value that is not deleted and store it, then delete the cells that is in the same row and column.

V1 V2	S11	S12	S13	S14	S15	S16
S21	0	10	8	15	0	0
S22	20	0	16	11	3	30
S23	5	7	3	24	12	32
S24	3	20	21	1	3	9
S25	10	13	23	2	5	2
S26	6	12	22	25	4	25

$d(v1, v2) = 1 + 3 = 4$

We used this distance definition as a distance between videos on k-medoid algorithm

5.4.2 Common Clusters Approach:

We define a similarity measure based on common clusters in two different videos, in this method, we calculated the similarity value by taking the number of common clusters in the videos and divide it by the total number of shots. It is similar to IOU (intersection over the union) measure. In this case, the shots of the whole dataset should be already clustered.

$$S(v1, v2) = \frac{C_{common}}{C_{total}}$$

$$C_{common} = the\ number\ of\ common\ clusters\ in\ the\ videos$$

$$C_{totla} = the\ number\ of\ total\ clusters\ in\ the\ videos$$

Equation 5.12 Common Clusters Approach

Example:

Let’s say we have the following Figure 5.17 two videos (video1 and video2)

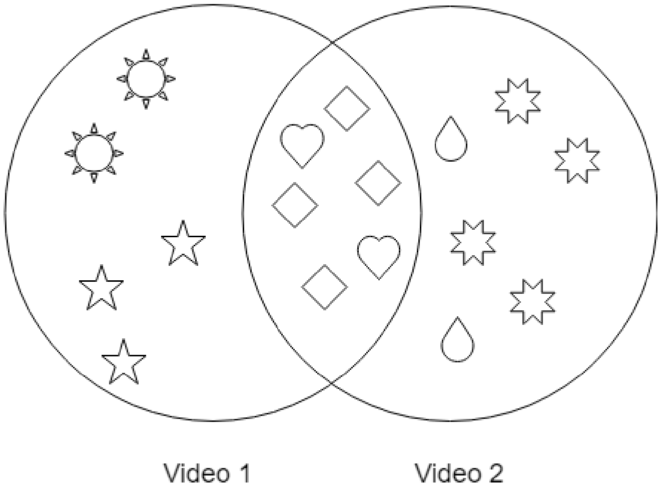


Figure 5.17 Common Clusters Approach Example

As you can see in the figure, suppose their shots are distributed over 6 clusters.

☼ ☆ ◇ ✷ ♡ 💧

Two of these clusters are common

◇ ♡

So, the similarity between these two videos is:

$$S(v1, v2) = \frac{C_{common}}{C_{total}} = \frac{2}{6} = 0.33$$

5.4.3 Common Shots Approach:

In this method, we calculated the similarity value by taking the number of the shots of the two videos that are in the same clusters and divide it by the total number of shots in both videos.

$$S(v1, v2) = \frac{Sh_{common}}{Sh_{total}}$$

$$Sh_{common} = the\ number\ of\ common\ shots\ in\ the\ videos$$

$$Sh_{totla} = the\ number\ of\ total\ shots\ in\ the\ videos$$

Equation 5.13 Common Clusters Approach

Example:

Let’s say we have the following Figure 5.18 two videos (video1 and video2)

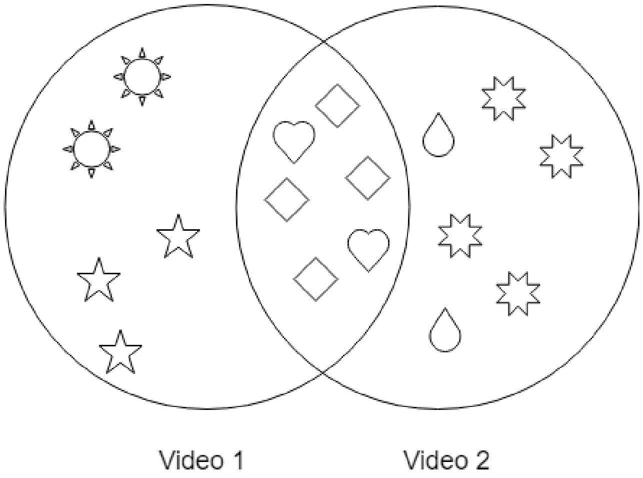


Figure 5.18 Common Shots Approach

As you can see in the figure, we have 17 shots as total in these two videos, 6 shots are in the same clusters. So, the similarity between these two videos is:

$$S(v1, v2) = \frac{Sh_{common}}{Sh_{total}} = \frac{6}{17} = 0.35$$

In the next chapter we will list the clustering and classification experiment and results that we did on out approach and last year approach.

Chapter 6: Experimentations and Results

In order to evaluate our proposal, we have applied a series of experimentations and compared the results with the results obtained in the 2018 internship which was based on a top-down approach. Several classification and clustering methods have been applied and the mAP and accuracy have been calculated. In the first section of the experimentations, we have applied several clustering methods in three different scenarios and compared the results. Since in clustering methods we should fix the value of some parameter such as the number of clusters in K-Means and the radius in DBSCAN, we have chosen to run the algorithm with different values and analyze the results. The three scenarios are as follows: (1) In the first scenario, we have extracted the features at the video level as it was used in the previous internship; (2) In the second scenarios, we have extracted the features at the shot level, then we have calculated the average feature vector over all the shots in the video; (3) In the third scenario, we have classified/clustered each shot separately.

6.1 Evaluation Metric:

Evaluating the machine learning algorithm is an essential part of any research. the model may give satisfying results when evaluated using a metric. Most of the times we use mAP and Accuracy to measure the performance of our model, however, it is not enough to truly judge our model. In this section, we will cover the different types of evaluation metrics that we used.

6.1.1 Mean Average Precision (mAP):

mAP is an evaluation measure for checking how accurate the results are. It is calculated as the average of the maximum precision at each recall level. Precision measures how accurate is your predictions. i.e. the percentage of your positive predictions are correct. While Recall measures how good you find all the positives. In each cluster we obtained we compare it with the ground truth and see the category of the majority of the correctly categorized and divide it by the total number of videos in this cluster, this gives us the average precision for the cluster. We apply this for all the clusters. To obtain the mAP, we sum all average precision and divide them by the number of the clusters. In the previous internships they mentioned they used the mAP but checking the code and the results, it was not the mAP, they used it without talking into consideration that we may have useless hidden results, which am going to explain.

In our case we assumed that the dominant category in cluster is the accurate result (TP), and the others are miss clustered (TN)

Precision measures how accurate is your predictions. i.e. the percentage of your predictions are correct.

$$C_{Prec} = \frac{C_{tp}}{C_{tp} + C_{fp}}$$

$\boldsymbol{C_{Prec}} = Cluster\ precision$

$\boldsymbol{C_{tp}} = Cluster\ true\ positive\ (retrieved\ and\ accurate)$

$\boldsymbol{C_{fp}} = Cluster\ flase\ positive\ (retrieved\ and\ not\ accurate)$

Equation 5.9 Precision

The following equation represent the mAP (mean average precision)

$$mAp = \frac{\sum_n C_{Prec}}{n}$$

$\boldsymbol{n} = number\ of\ clusters$

Equation 5.10 mAP

6.1.2 Accuracy:

Using clustering algorithms to identify common patterns in unsupervised data can be a difficult concept to apply in practice because there are no explicit methods to assess the accuracy of the results. Predicted data labels and real data labels are required to calculate precision, and there are no real data labels with unsupervised information. In this step we calculated the accuracy of the clustering results for each algorithm output for both bottom-up and top-down approach.

The following equation represent the accuracy of result

$$Acc = \frac{TP}{TP + FP}$$

$\boldsymbol{TP} = True\ positive\ (retrieved\ and\ accurate)$

$\boldsymbol{FP} = Flase\ positive\ (retrieved\ and\ not\ accurate)$

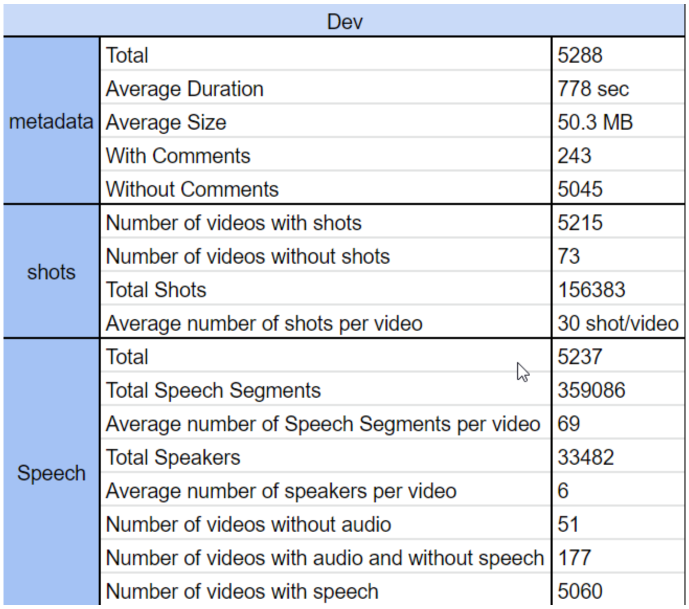
Equation 5.11 Accuracy

6.2 Dataset Statistics:

As a first step of our experiment was to apply statistics on the “Blib10000 – MediaEval” dataset in order to have a bird's eye view on the data nature.
For both ***Dev*** and ***Test*** data we calculated the following statistics:

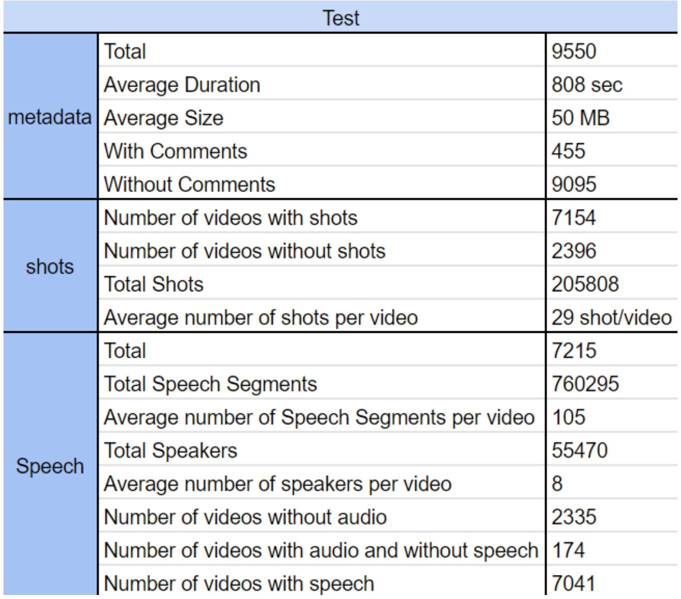
1. **Metadata**
    1. Total number of videos
    2. Video average duration
    3. Video average size
    4. Number of videos with comments
    5. Number of videos without comments
2. **Shots**:
    1. Number of videos with shots
    2. Number of videos without shots
    3. The average number of shots per video
3. **Speech**
    1. Total videos with speech
    2. Total Speech Segments
    3. The average number of Speech Segments per video
    4. Total Speakers
    5. The average number of speakers per video
    6. Number of videos without audio
    7. Number of videos with audio and without speech
    8. Number of videos with speech

The following table contains the statistics results for ***Dev*** and ***Test*** data:



Dev		
metadata	Total	5288
	Average Duration	778 sec
	Average Size	50.3 MB
	With Comments	243
	Without Comments	5045
shots	Number of videos with shots	5215
	Number of videos without shots	73
	Total Shots	156383
	Average number of shots per video	30 shot/video
Speech	Total	5237
	Total Speech Segments	359086
	Average number of Speech Segments per video	69
	Total Speakers	33482
	Average number of speakers per video	6
	Number of videos without audio	51
	Number of videos with audio and without speech	177
	Number of videos with speech	5060

Table 5.1 Dev Data Statistics



Test		
metadata	Total	9550
	Average Duration	808 sec
	Average Size	50 MB
	With Comments	455
	Without Comments	9095
shots	Number of videos with shots	7154
	Number of videos without shots	2396
	Total Shots	205808
	Average number of shots per video	29 shot/video
Speech	Total	7215
	Total Speech Segments	760295
	Average number of Speech Segments per video	105
	Total Speakers	55470
	Average number of speakers per video	8
	Number of videos without audio	2335
	Number of videos with audio and without speech	174
	Number of videos with speech	7041

Table 5.2 Test Data Statistics

6.3 Clustering Results:

We try to perform the experiments using the defined distance in the previous chapter using the same dataset, clustering algorithms and cluster evaluation method (mAP) and the accuracy. Figure 6.2 shows the mAP obtained by running the K-means algorithm in order to cluster videos in the three scenarios defined at the beginning of this chapter while Figure 6.3 shows the accuracy results.

As we can notice, clustering videos by taking the average of the feature vectors extracted from the shot level or clustering the shots themselves are more performant than clustering videos using features extracted from the video level.

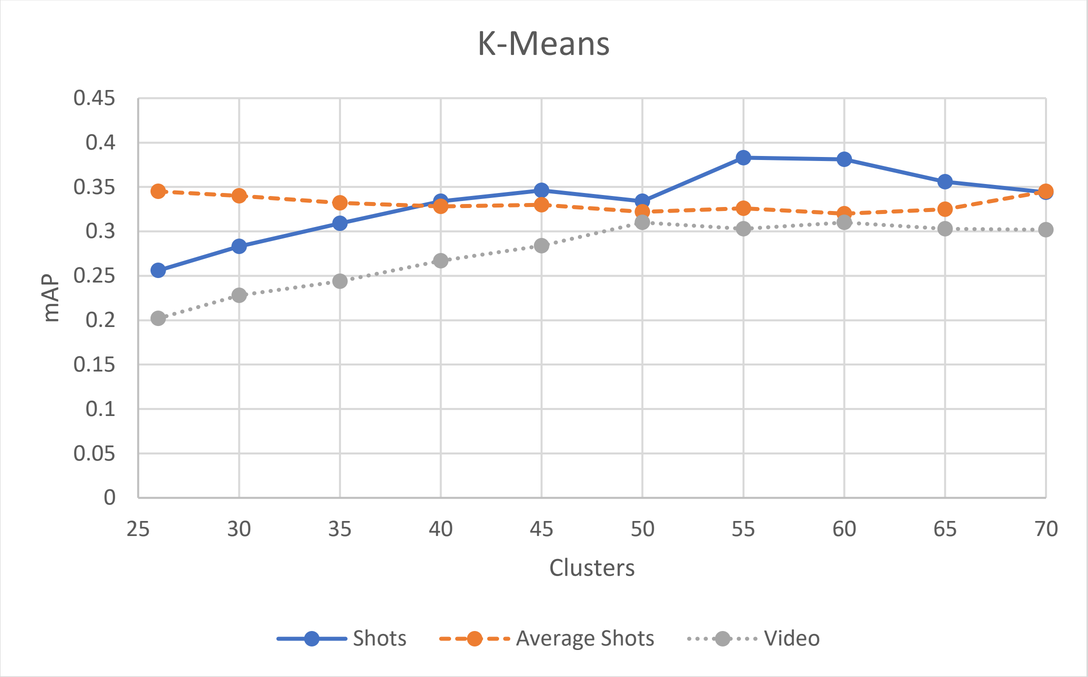


Figure 6.2 K-Means Mean Average Precision Results

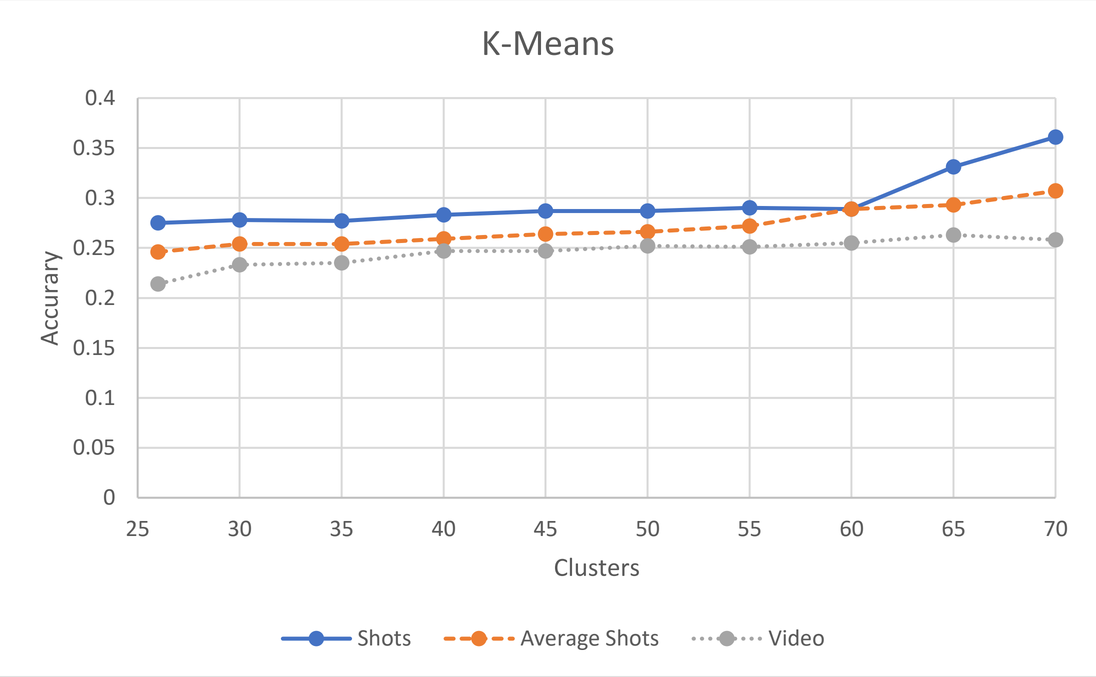


Figure 6.3 K-Means Accuracy Results

Figures 6.4 and 6.5 validate the same results obtained using the k-means algorithm but this time with the DBSCAN algorithm. Moreover, we notice that the DBSCAN algorithm is more performant than the k-means one.

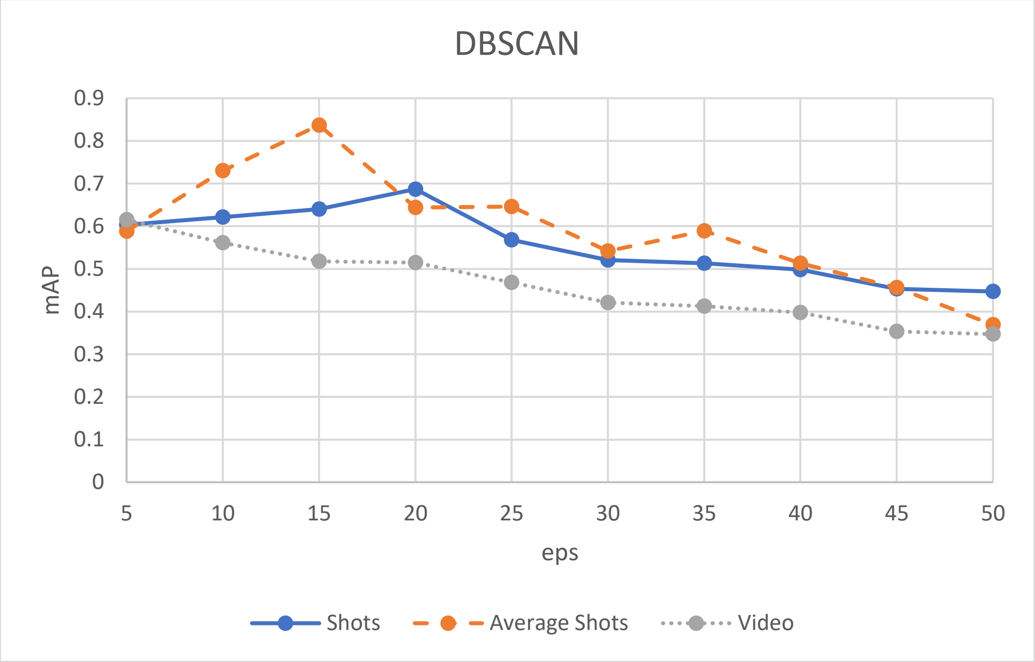


Figure 6.4 BDSCAN Mean Average Precision Results

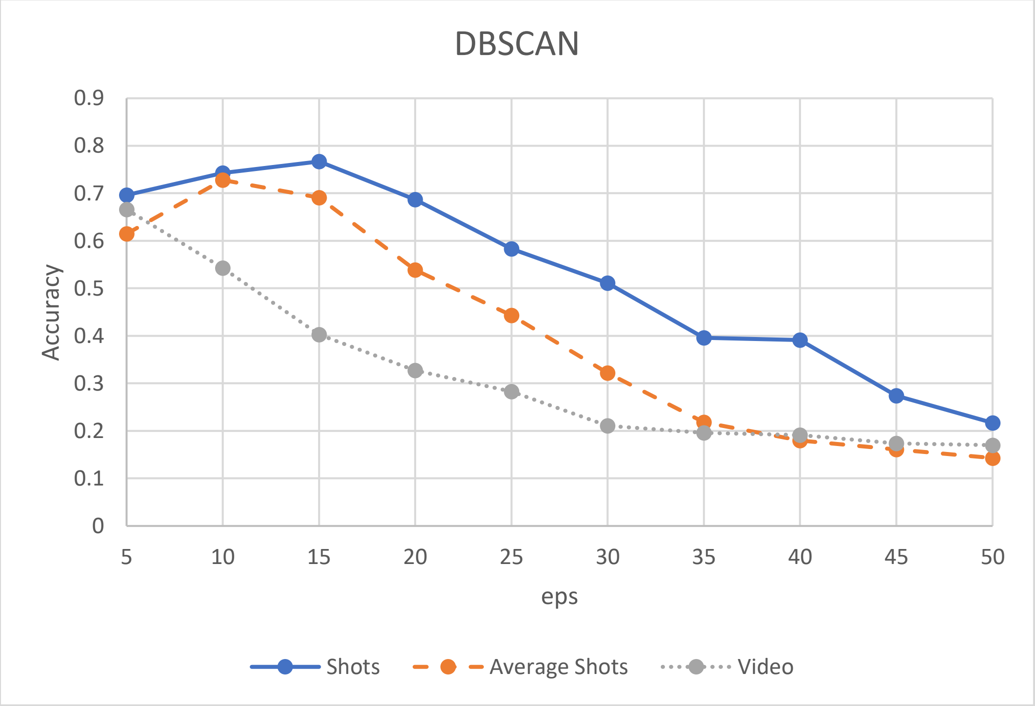


Figure 6.5 BDSCAN Accuracy Results

Figures 6.6 and 6.7 show the results of clustering obtained by running the BIRCH clustering algorithm while Figure 6.8 and 6.9 presents the results obtained by the Mean-Shift algorithm.

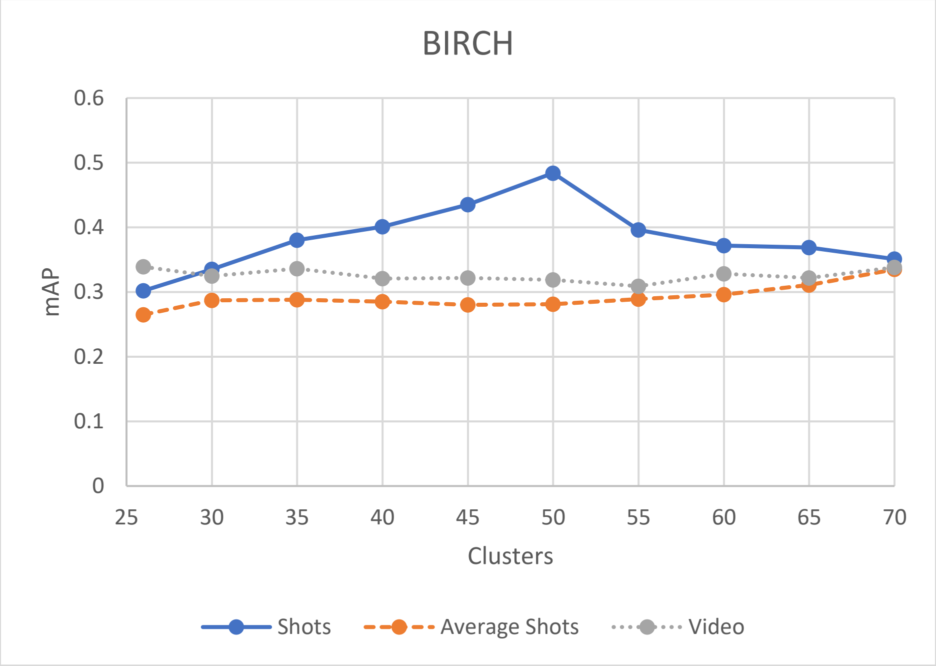


Figure 6.6 BIRCH Mean Average Precision Results

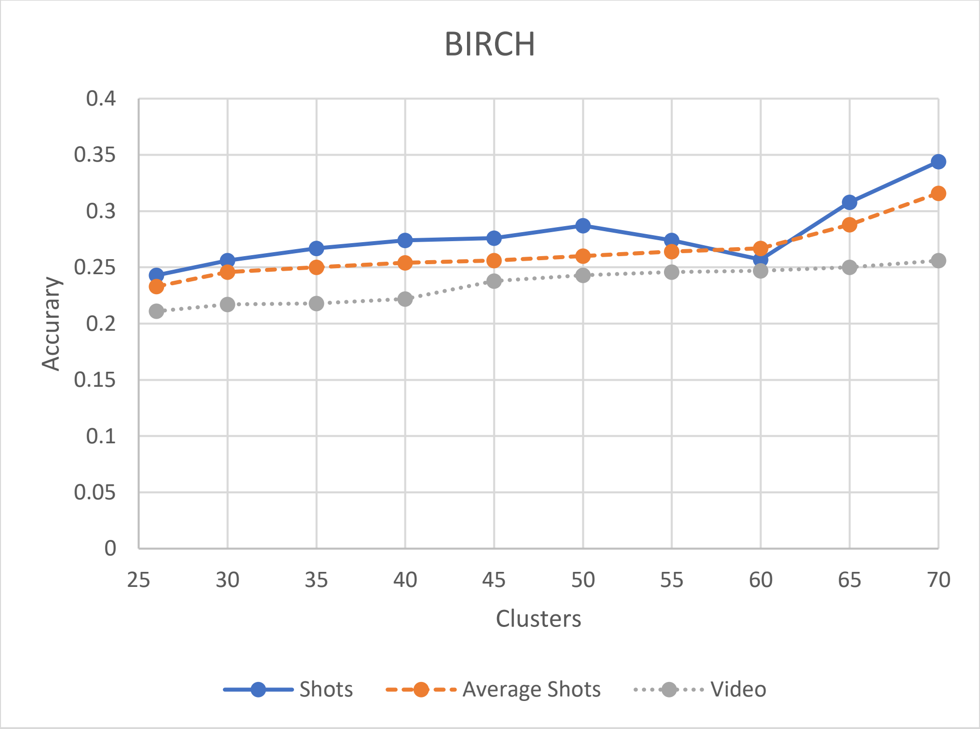


Figure 6.7 BDSCAN Accuracy Results

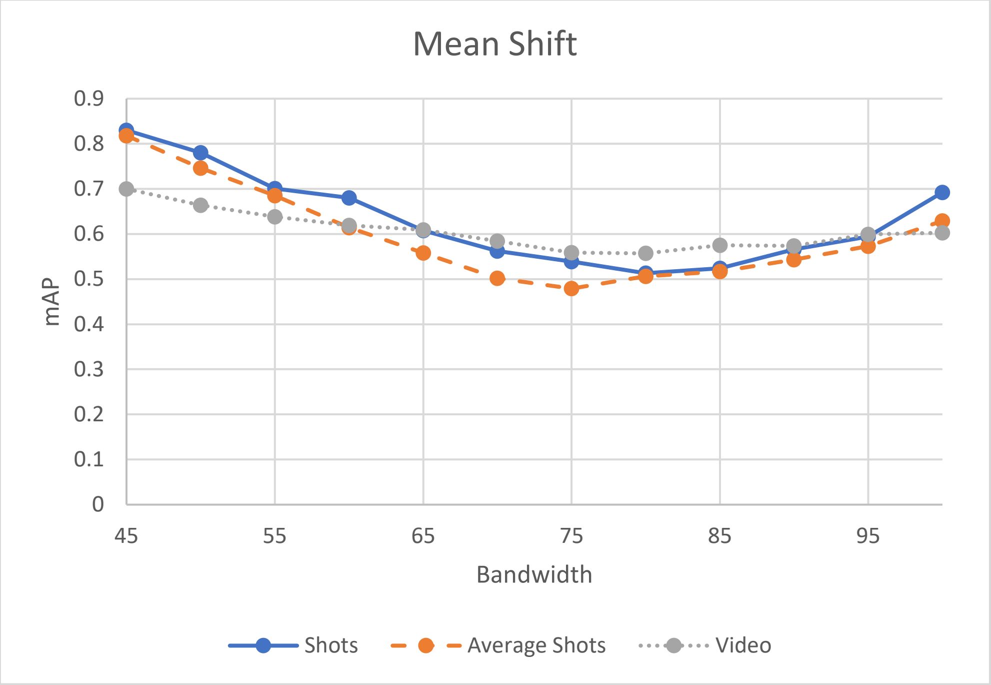


Figure 6.8 Mean Shift Mean Average Precision Results

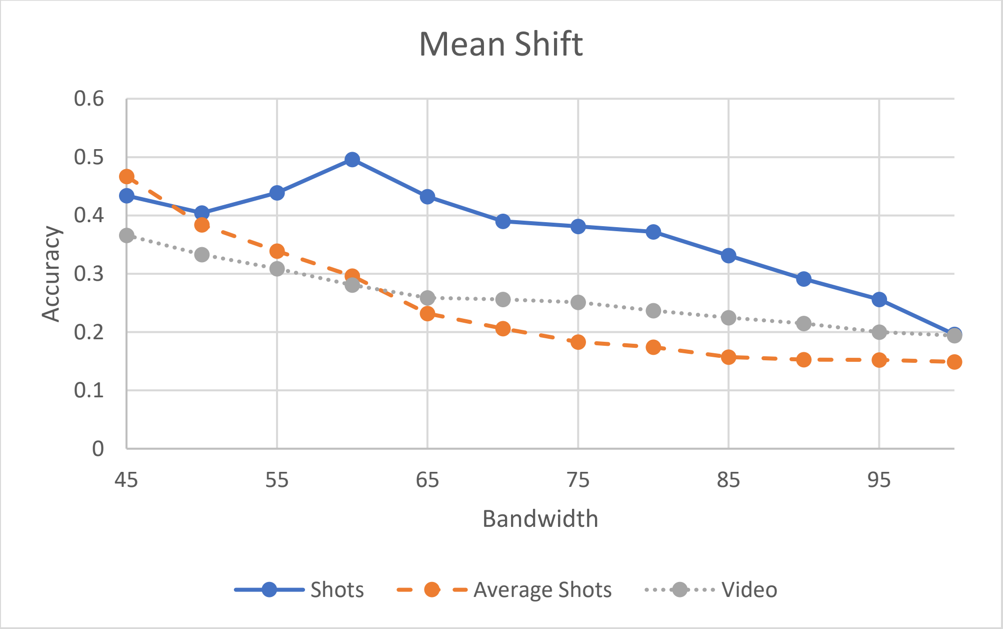


Figure 6.9 Mean Shift Accuracy Results

The last experiment done in the framework of clustering is applying the k-medoid clustering algorithm using the shot-matching distance defined in the chapter of proposed approach. The idea behind using the k-medoid and not the k-means is that the videos do not have the same number of shots which make hard the process of calculation of the average video of each cluster. As we can notice in Figures 6.10 and 6.11, the results are higher than simply cluster videos using features extracted at the vide level.

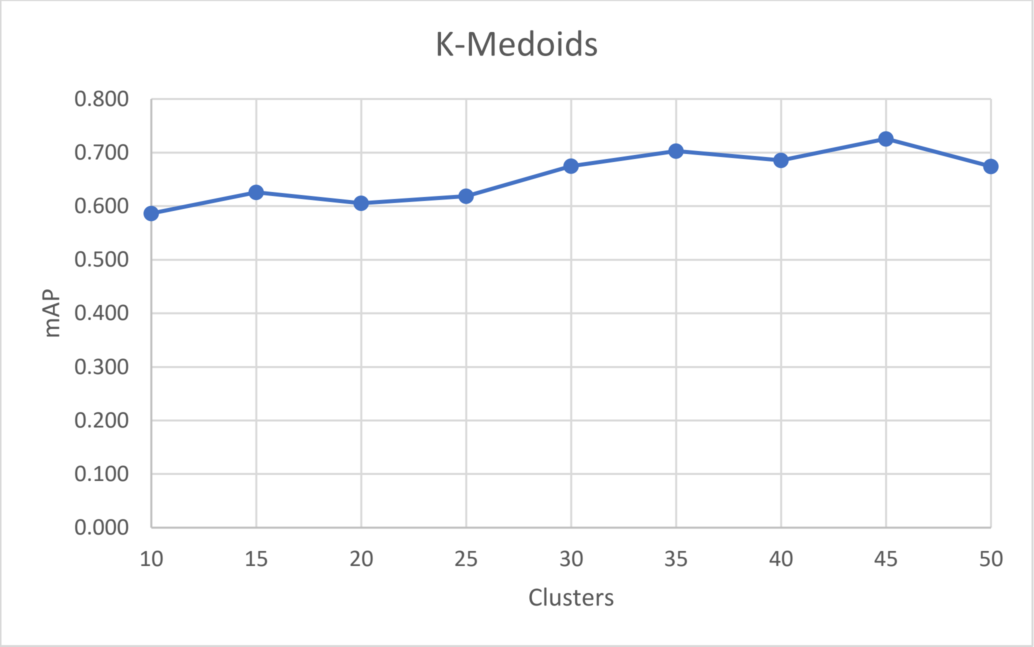


Table 6.10 K-Medoids Accuracy Results

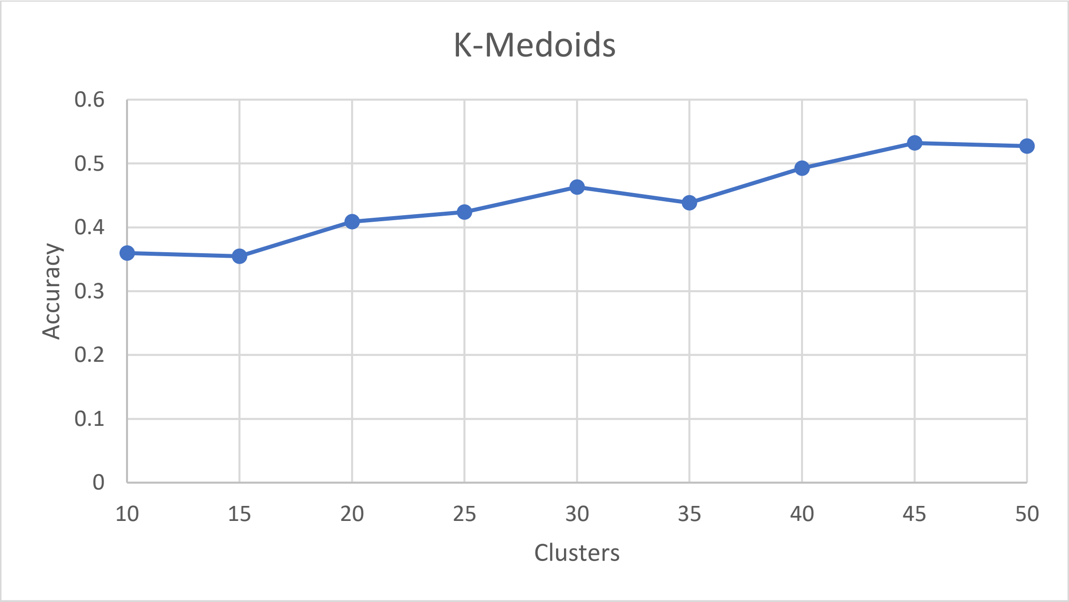


Figure 6.11 K-Medoids Accuracy Results

As we can highlight in the results above, clustering videos based on features extracted at the shot level that we have proposed is more performant than the one propose in the previous internships based on the video-level. The enhancement in some algorithms reached 30%.

6.4 Classification Results:

Several classification methods have been applied on the same scenarios mentioned above and the obtained results are shown in the tables below. We have run two type of experiments here. The first on the dataset including the default category while the second without the default category since this category may contain videos from the other categories. Two sampling methods are applied: (1) 10-folds cross-validation sampling method; (2) Random sampling method. Tables 6.13, 6.14, 6.15 and 6.16 shows the accuracy of the classification using different classification methods.

Accuracy	Shots	Average Shots	Video
kNN	0.467	0.303	0.281
SVM	0.061	0.308	0.320
Random Forest	0.618	0.330	0.375
Neural Network	0.432	0.334	0.361
Naïve Bayes	0.261	0.231	0.269
Logistic Regression	0.331	0.302	0.320
AdaBoost	0.809	0.251	0.288

Table 6.13 Cross Validation Results

Accuracy	Shots	Average Shots	Video
kNN	0.467	0.341	0.324
SVM	0.163	0.37	0.389
Random Forest	0.58	0.384	0.419
Neural Network	0.449	0.396	0.103
Naïve Bayes	0.287	0.267	0.314
Logistic Regression	0.367	0.34	0.361
AdaBoost	0.683	0.287	0.334

Table 6.14 Cross Validation Without Default Results

Accuracy	Shots	Average Shots	Video
kNN	0.415	0.272	0.243
SVM	0.141	0.304	0.322
Random Forest	0.535	0.305	0.333
Neural Network	0.406	0.306	0.328
Naïve Bayes	0.256	0.031	0.177
Logistic Regression	0.331	0.287	0.306
AdaBoost	0.697	0.224	0.249

Table 6.15 Random Sampling 30% Training and 70% Testing Results

Accuracy	Shots	Average Shots	Video
kNN	0.45	0.341	0.274
SVM	0.163	0.37	0.364
Random Forest	0.58	0.384	0.374
Neural Network	0.449	0.396	0.376
Naïve Bayes	0.287	0.267	0.083
Logistic Regression	0.367	0.34	0.345
AdaBoost	0.683	0.287	0.292

Table 6.16 Random Sampling 30% Training and 70% Testing Without Default Results

As we can notice here, classifying the shots is more accurate than classifying the videos. However, since our final aim is to classify videos, we have based on the classified shots to classify the videos. The idea is to label the video by the majority label of its composed shots. For example, suppose a video has 5 shots, 3 of them are classified as "education" shots and the other are "sport" ones. In this case, the video is labeled as "education". We run an experiment to test the accuracy and tables 6.17 and 6.18 shows the accuracy of the proposed method using only the cross-validation sampling method with a comparison with the other ones.

Accuracy	Shots	Average Shots	Video	Video from Shots
kNN	0.467	0.303	0.281	0.511
SVM	0.061	0.308	0.320	0.412
Random Forest	0.618	0.330	0.375	0.569
Neural Network	0.432	0.334	0.361	0.278
Naïve Bayes	0.261	0.231	0.269	0.246
Logistic Regression	0.331	0.302	0.320	0.045
AdaBoost	0.809	0.251	0.288	0.705

Table 6.17 Cross Validation Results

Accuracy	Shots	Average Shots	Video	Video from Shots
kNN	0.467	0.341	0.324	0.511
SVM	0.163	0.37	0.389	0.412
Random Forest	0.58	0.384	0.419	0.575
Neural Network	0.449	0.396	0.103	0.278
Naïve Bayes	0.287	0.267	0.314	0.246
Logistic Regression	0.367	0.34	0.361	0.045
AdaBoost	0.683	0.287	0.334	0.705

Table 6.18 Cross Validation Without Default Results

The above tables illustrate the enhancement that our approach added to the result, moreover, the enhancing in some algorithms reached the 250%.

7. Conclusion and Future Work:

7.1 Conclusion:

In this internship, we introduced new video characterization and comparison methods for performing unsupervised and supervised learning in order to explore and group video documents. It was based on an existing work, in which the feature set for describing a video was already established, in addition to the idea of considering a video at different granularity levels, where it is segmented into several shots. So, where we proposed a bottom-up representation of video documents, where the video is divided into several shots and feature vectors are extracted for each shot separately. Then, we provide an appropriate distance measure that takes into account the granular representation of video contents. Therefore, instead of measuring distance between videos based on one feature vector that is extracted from the whole video, we measure it based on the shots' comparisons.

Our proposed approach has added up to 200% enhancement to the classification and clustering results compared to the previous internship approach.

7.2 Future Work:

Considering what have been accomplished in this training, we would like to explore several other points and expand on some definitions provided in this training:

- Propose a method to compare a video to itself in order to structure it.
- Introduce additional modalities (text transcripts, social data) to extend the descriptors currently used.
- Explore different variations of clustering algorithms, evaluation methods and weight calculation methods regarding feature selection, a study in this field is required.

References:

[1] X. Wu, C.-W. Ngo, and Q. Li, “Threading and auto documenting news videos,” *IEEE Signal Process*. Mag., vol. 23, no. 2, pp. 59–68, Mar. 2006.

[2] K. PetridisIoannis, S. Bloehdorn, S. Handschuh, and S. Staab, "Knowledge Representation for Semantic Multimedia Content Analysis and Reasoning", *Informatics and Telematics Institute 1st Km Thermi-Panorama Rd*, 2004.

[3] B. Bigot, I. Ferrané, J. Pinquier, R. André-Obrecht, “Detecting Individual Role Using Features Extracted from Speaker Diarization Results”. *Springer Science and Business Media, LLC,* 2010

[4] M. Fakhoury, "Multimodal and multi-level characterization of audio-visual content for the search for similarity and the automatic creation of collections.," *Master's Thesis, Université Paul Sabatier, Lebanese University*, 2016.

[5] H. A. Jawad, "Multimodal and multi-level characterization of audio-visual content for the search for similarity and the automatic creation of collections," *Université Paul Sabatier Université Lebanaise*, 2017.

[6] A. Abbas, "Multimodal and multi-level characterization of audio-visual content for the search for similarity and the automatic creation of collections," *Université Paul Sabatier Université Lebanaise*, 2018.

[7] P. N. Sethi, “Multimedia data mining: an overview. In: Multimedia data mining and knowledge discovery”. *Springer*, 2007

[8] S. Mouysset., J. Noailles., D. Ruiz, R. Guivarch. “On a Strategy for Spectral Clustering with Parallel Computation”. *Lecture Notes in Computer Science High Performance Computing for Computational Science.* 2011

[9] S. Schmiedeke, P. Xu, I. Ferrané, M. Eskevich, C. Kofler, M. A. Larson, Y. Estève, L. Lamel, G. J. Jones, T. Sikora. “Blip10000: A social Video Dataset containing SPUG Content for Tagging and Retrieval”, *Technische Universität Berlin, Germany 2Delft University of Technology, The Netherlands 3University of Toulouse, France 4Dublin City University, Ireland, 5Language and Speech Technology (LST) team, LIUM, Le Mans, France 6Spoken Language Processing Group, LIMSI/Vocapia*, 2013

[10] M. Naaman. “Social multimedia: highlighting opportunities for search and mining of multimedia data in social media applications”. *Multimedia Tools Appl., 56(1):9–34*, Jan. 2012.

[11] LIMSI/Vocapia, *“https://www.vocapia.com”.* July 2000 [Online].

[12] LIUM Research team LST, *“https://lium.univ-lemans.fr”*, [Online].

[13] C. Amitkumar Bhatt· M. S. Kankanhalli, “Multimedia data mining: state of the art and challenges”*, Springer Science+Business Media*, 16 November 2010

[14] F. D. Blanchaud. “Extraction and combination of descriptors for the structuring of audiovisual documents”. *IEEE,* 2015

[15] Julien Pinquier, Jean-Luc Rouas, Regine André-Obrecht. “Merging parameters for robust speech / music classification”. *Computer Science and Technology (IST): Digital / Symbolic Fusion, Hermès, 8, quai du marche neuf,* 2003.

[16] . R. Agrawal, T. Imielinski, A. Swami, “Mining association rules between sets of items in large databases”. *ACM SIGMOD international conference on management of data,* 1993

[17] J. Hipp, U. Güntzer, G. Nakhaeizadeh, “Algorithms for association rule mining a general survey and comparison”. *SIGKDD Explorations 2(2):1–58*, 2000

[18] S. Kotsiantis, D. Kanellopoulos. “Association rules mining: a recent overview”. *Int Trans Comput Sci Eng 32(1):71–82*, 2006

[19] Han J, Pei J, “Mining frequent patterns by pattern-growth: methodology and implications”. *ACM SIGKDD Explorations Newsletter 2(2):14–20,* 2000

[20] R. Duda, P. Hart, D. Stork, “Pattern classification”. *Wiley, New York,* 2001

[21] J. Quinlan: “programs for machine learning”. *Morgan Kaufmann, San Mateo,* 2012

[22] J. Han, M. Kamber “Data mining concepts and techniques”. *Morgan Kaufmann, San Mateo,* 2006

[23] J. Artigan, “Clustering algorithms”. *Wiley, New York,* 1975

[24] M. Ester, H. Kriegel, J. Sander, X. Xu “A density-based algorithm for discovering clusters in large spatial databases with noise”. *International conference on knowledge discovery and data mining, pp 226–231,* 1996

[25] T. Zhang, R. Ramakrishnan, M. Livny, “Birch: an efficient data clustering method for very large databases”. *In: SIGMOD conference, pp 103–114,* 1996

[26] Sheikholeslami G, Chatterjee S, Zhang A, “Wavecluster: a multi-resolution clustering approach for very large spatial databases”. In: *International conference on very large data bases (VLDB), pp 428–439,* 1998

[27] Dai K, Zhang J, Li G, “Video mining: concepts, approaches and applications”. *In: Multimedia modelling,* 2006

[28] Hwan OJ, Lee JK, Kote S (2003) Real time video data mining for surveillance video streams. In: Pacific-Asia conference on knowledge discovery and data mining

[29] Zhu X, Wu X, Elmagarmid AK, Wu L, “Video data mining: semantic indexing and event detection from the association perspective”. *IEEE Trans Knowl Data Eng 17(5):665–677,* 2005

[30] Darrell T, Pentland A, “Space-time gestures”. *In: IEEE Computing Society conference on computer vision and pattern recognition, pp 335–340,* 1993

[31] Yamato J, Ohya J, Ishii K,”Recognizing human action in time-sequential images using hiddenmarkov model”. *In: IEEE Computing Society conference on computer vision and pattern recognition, pp 379–385*, 1992

[32] Aradhye H, Toderici G, Yagnik J,”Video2text: learning to annotate video content”. *In: International conference on data mining workshops, pp 144–151,* 2009

[33] Lin L, Shyu ML, Ravitz G, Chen SC, “Video semantic concept detection via associative classification.” *In: IEEE international conference on multimedia and expo, pp 418–421,* 2009

[34] Oh J, Bandi B, ”Multimedia data mining framework for raw video sequences”. In: International workshop on multimedia data mining (MDM/KDD), pp 1–10, 2002

[35] Yeung M, Yeo BL, Liu B,”Extracting story units from long programs for video browsing and navigation”. In: Readings in multimedia computing and networking. Morgan Kaufmann, San Mateo. 2001

[36] Yeung MM, Yeo BL, “Time-constrained clustering for segmentation of video into story unites”. *Int Conf Pattern Recognit 3:375–380*, 1996

[37] Shirahama K, Sugihara C, Matsumura K, Matsuoka Y, Uehara K, “Mining event definitions from queries for video retrieval on the internet”. In: International conference on data mining workshops, pp 176–183, 2009

[38] Faloutsos C, Equitz W, Flickner M, Niblack W, Petkovic D, Barber R, “Efficient and effective querying by image content”. Journal of Intelligent Information Systems 3:231–262, 1994

[39] Gorkani MM, Con R, Picard W, “Texture orientation for sorting photos at a glance." In: IEEE conference on pattern recognition. 1994

[40] Pentland A, Picard RW, Sclaroff S “Photobook: content-based manipulation of image databases”. Int J Comput Vis 18:233–254, 1996

[41] Zhang HJ, Zhong D, ”A scheme for visual feature-based image indexing”. *In: SPIE conference on storage and retrieval for image and video databases*, 1995

[42] Shirahama K, Iwamoto K, Uehara K, “Video data mining: rhythms in a movie”. In: International conference on multimedia and expo, 2004

[43] Lin L, Shyu ML, mining high-level features from video using associations and correlations. In: International conference on semantic computing, pp 137–144, 2009

[44] Shirahama K, Sugihara C, Matsumura K, Matsuoka Y, Uehara K, Mining event definitions from queries for video retrieval on the internet. In: International conference on data mining workshops, pp 176–183, 2009

[45] Chen SC, Shyu ML, Zhang C, Strickrott J, Multimedia data mining for traffic video sequenices. In: ACM SIGKDD, 2001

[46] Fu CS, Chen W, Jianhao MH, Sundaram H, Zhong D, A fully automated content based video search engine supporting spatio-temporal queries. IEEE Trans Circuits Syst Video Technol 8(5):602–615, 1998

[47] Sclaroff S, Kollios G, Betke M, Rosales R, Motion mining. In: International workshop on multimedia databases and image communication, 2001
[48] J. B. MacQueen, "Some Methods for classification and Analysis of Multivariate Observations, Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability," pp. 281-297, 1967.

[49] Mohammed J. Zaki Wagner Meira Jr, "Data Mining and Analysis: Fundamental Concepts and Algorithms", Cambridge University Press, 2013.

[50] Sci Kit Learn, " https://scikit-learn.org/stable/" , [Online].

[51] Cover, T., Hart, P.: Nearest neighbor pattern classification. IEEE Transactions on Information Theory 13(1), 21–27 1967

[52] Qin, Z., Wang, A.T., Zhang, C., Zhang, S.: Cost-sensitive classification with k-nearest neighbors. In: Wang, M. (ed.) KSEM 2013. LNCS, vol. 8041, pp. 112–131. Springer, Heidelberg 2013

[53] A New Model of Selection in Women's Handball Vatromir Srhoj1, Nenad Rogulj1, Neboj{a Zagorac2 and Ratko Kati}1 1 Faculty of Natural Sciences, Mathematics and Kinesiology, University of Split, Split, Croatia 2 Millennium Institute for Sports and Health, Auckland, New Zealand Coll. Antropol. 30 2006

Appendix A:

1. Clustering:

Clustering is a method of grouping the information into classes or clusters in such a way that objects within a cluster are highly similar to each other but very different from objects in another cluster [22]. It is possible to organize the clustering techniques as partitioning, hierarchical, density-based, grid-based and model-based methods. Sometimes clustering is biased, as only round-shaped clusters can be obtained, and scalability is also a problem. Using Euclidean or Manhattan distance measurements tends to find spherical clusters of similar size and density, but clusters may have any form. Some clustering techniques are susceptible to input information order and may not be able to integrate freshly inserted information at times. Interpretability and usability of clustering outcomes is a significant problem. The high dimensionality of data, noise and missing values are also problems for clustering. Based on the partitioning method, K-means [23] clustering is one of the common clustering techniques. Chameleon and BIRCH [25] are excellent hierarchical techniques for clustering. DBSCAN [24] is a clustering method based on density. Wavelet transform-based Wave Cluster clustering [26] is a technique based on a grid.

This chapter is intended to show the techniques used to group the information in our job. There is no widely applied clustering method, algorithms offer distinct results depending on the circumstances. For big information collection, some clustering methods are better and some offer excellent results to find a cluster with arbitrary forms. A grouping's quality depends on the similarity or distance measure used by the technique and its execution, the assessment measure we use is the key to excellent outcomes. We will use these algorithms to group our information, without using the specified category, and the latter to compare it to the category that is for us the ground truth, and we will also verify how remote our clutters are.

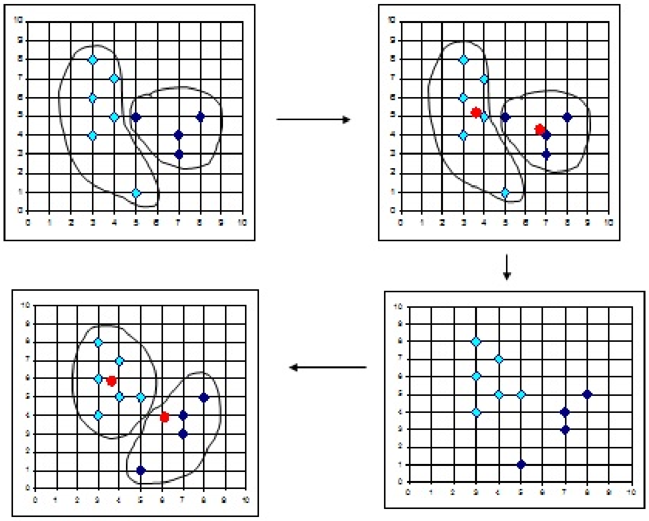
We used the K-means algorithm, DB scan, and the mean shift clustering because they were already implemented and are proven efficient enough and we don’t know the structure of the data.

1.1 K-Means Clustering [49] [50]:

K-means clustering method is one of the most adopted methods of clustering in healthcare industry due to its simplicity and performance in other fields of research [48]. The name was derived from representing each cluster with the mean (average weight) of the points in each cluster known as centroid. To determine the appropriate number of K value in K-means clustering is a common challenge in the clustering process. The algorithm performed as follows:

1. Randomly choose K data points from the dataset as the initial centroids based on the specified number of clusters.
2. Then assign each data point to the nearest centroid by calculating the minimum Euclidean distance of each data point to each centroid.
3. Set the position of each cluster to the mean of all data points belonging to that cluster.
4. Repeat steps b and c above until convergence is observed.

Figure 5.1 illustrates three iterations of k-means to arrive at the end to the desired result. In this case the cluster number is equal to 2. The two light blue and dark blue colors represent two different clusters and the red dots represent the average mean of the clusters.



5.1 three iterations of k-means example

Figure 5.1 Schema that graphically represents the different steps of k-means applied on 10 objects to classify them into two clusters. We applied this algorithm on the shot level and on the videos as well.

1.2 Density-based Clustering (DBSCAN) [49] [50]:

Representative-based clustering techniques such as K-means and Expectation-Maximization are appropriate for discovering clusters that are ellipsoid-shaped, or at best convex [49]. These techniques, however, have difficulty finding the real clusters for non-convex clusters, such as those shown in Figure 5.2, as two points from distinct clusters may be nearer than two points in the same cluster. These non-convex clusters can be mine by the density-based techniques we consider in this section.

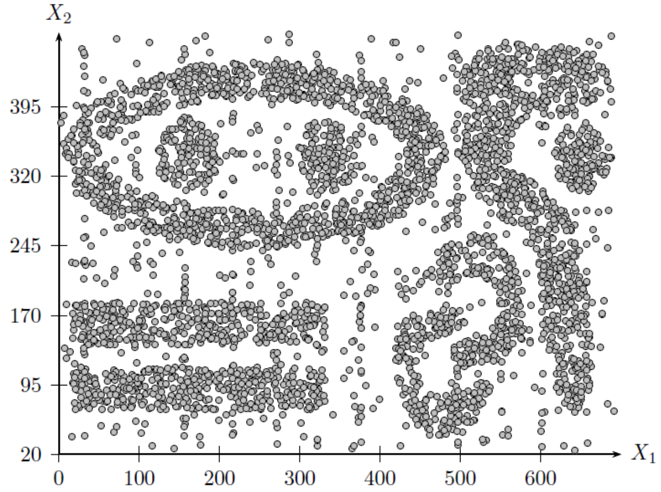
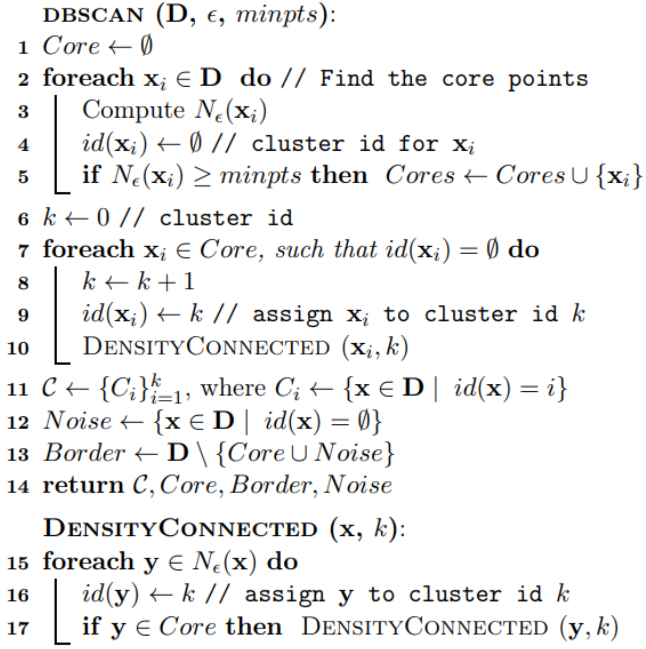


Figure 5.2 Density-based Dataset [49]

The DBSCAN algorithm considers clusters as high-density regions separated by low-density regions. Clusters discovered by DBSCAN can be any shape, as opposed to k-means, which assumes clusters are convex shaped because of this rather generic perspective. The core sample concept, which are samples in high-density areas, is the central component of the DBSCAN. Therefore, a cluster is a collection of key samples near each other (measured by some range measurement) and a set of non-core samples close to a core sample (but not core specimens themselves).

Density-based Clustering Algorithm [49]



```
DBSCAN (D, ϵ, minpts):
Core ← ∅
foreach $x_i \in D$ do // Find the core points
    Compute $N_\epsilon(x_i)$
    $id(x_i) \leftarrow \emptyset$ // cluster id for $x_i$
    if $N_\epsilon(x_i) \geq minpts$ then $Cores \leftarrow Cores \cup \{x_i\}$
$k \leftarrow 0$ // cluster id
foreach $x_i \in Core$, such that $id(x_i) = \emptyset$ do
    $k \leftarrow k + 1$
    $id(x_i) \leftarrow k$ // assign $x_i$ to cluster id $k$
    DensityConnected $(x_i, k)$
$\mathcal{C} \leftarrow \{C_i\}_{i=1}^{k}$, where $C_i \leftarrow \{x \in D \mid id(x) = i\}$
$Noise \leftarrow \{x \in D \mid id(x) = \emptyset\}$
$Border \leftarrow D \setminus \{Core \cup Noise\}$
return $\mathcal{C}, Core, Border, Noise$

DensityConnected (x, k):
foreach $y \in N_\epsilon(x)$ do
    $id(y) \leftarrow k$ // assign y to cluster id $k$
    if $y \in Core$ then DensityConnected $(y, k)$
```

1.3 Mean Shift Clustering [49] [50]:

Mean Shift clustering aims to discover blobs in a smooth density of samples. It is a centroid based algorithm, which works by updating candidates for centroids to be the mean of the points within a given region. These candidates are then filtered in a post-processing stage to eliminate near-duplicates to form the final set of centroids. [50]

Given a candidate centroid for iteration, the candidate is updated according to the following equation:

$$x_i^{t+1} = m(x_i^t)$$

Where $N(x_i)$ is the neighborhood of samples within a given distance around xi and m is the *mean shift* vector that is computed for each centroid that points towards a region of the maximum increase in the density of points? This is computed using the following equation, effectively updating a centroid to be the mean of the samples within its neighborhood:

$$m(x_i) = \frac{\sum_{x_j \in N(x_i)} K(x_j - x_i) x_j}{\sum_{x_j \in N(x_i)} K(x_j - x_i)}$$

The algorithm automatically sets the number of clusters, instead of relying on a parameter bandwidth, which dictates the size of the region to search through. This parameter can be set manually, but can be estimated using the provided estimate bandwidth function, which is called if the bandwidth is not set.

The algorithm is not highly scalable, as it requires multiple nearest neighbor searches during the execution of the algorithm. The algorithm is guaranteed to converge; however, the algorithm will stop iterating when the change in centroids is small.

Labelling a new sample is performed by finding the nearest centroid for a given sample.

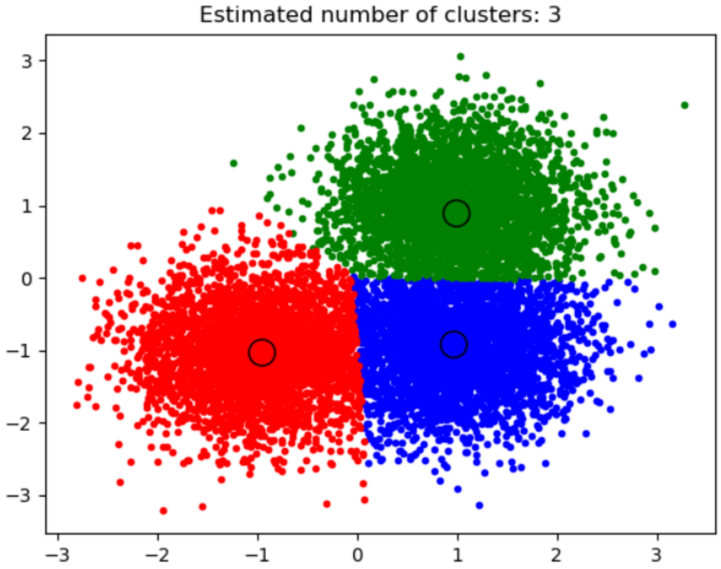


Figure 5.3 Mean Shift Clustering Example

1.4 BIRCH (balanced iterative reducing and clustering using hierarchies):

BIRCH is very fast. It clusters 100,000 points into 1000 clusters in 4 seconds [51]. BIRCH requires three parameters: the branching factor Br, the threshold T, and the cluster count k. While the data points are entered into BIRCH, a height-balanced tree, the cluster features tree, or CF tree, of hierarchical clusters is built. Each node represents a cluster in the cluster hierarchy, intermediate nodes are superclusters and the leaf nodes are the actual clusters. The branching factor Br is the maximum number of children a node can have. This is a global parameter. Every node contains the most important information of the belonging cluster, the cluster features (CF). From those, the cluster centers $Ci = \frac{1}{n_i}\sum_j^n x_{ij}$, where $\{x_{ij}\}_{j=1}^{n}$ are the elements of the ith cluster, and the cluster radii $R_i = \sqrt{\frac{1}{n_i}\sum_j^n (x_{ij} - C_i)^2}$ can be computed for each cluster. Every new point starts at the root and recursively walks down the tree, always entering the sub cluster with the nearest center until the walk ends at a certain leaf node. Once arrived at a leaf, the new point is added to this leaf cluster, provided this would not increase the radius of the cluster beyond the threshold T. Otherwise a new cluster is created with the new point as its only member. The threshold parameter thus regulates the cluster size. If creating a fresh cluster results in more than the parent's Br family nodes, the parent is divided. The nodes further above may need to be divided recursively to guarantee that the tree remains balanced. Once all points are presented to BIRCH, the leaf cluster centers are entered into a clustering algorithm in the worldwide clustering stage, such as agglomerative clustering or k-means, given as the cluster count k parameter. By combining adjacent clusters, this last step increases the performance of the cluster. In this document, if the BIRCH algorithm needs to be distinguished.

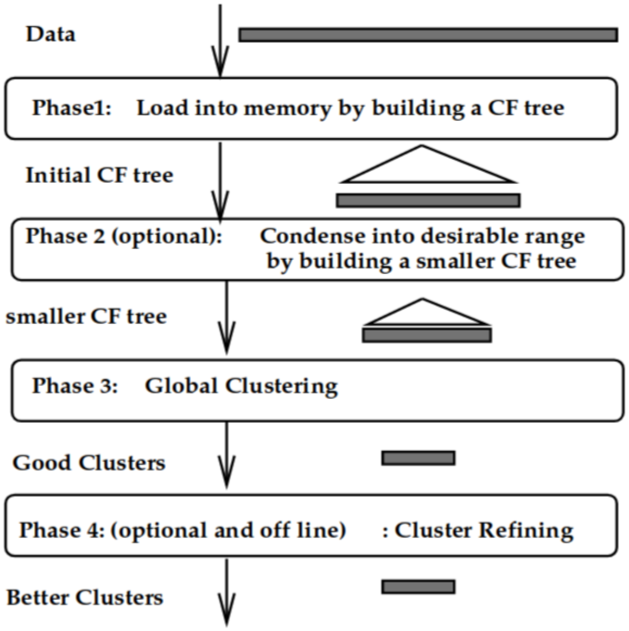

Data
Phase1: Load into memory by building a CF tree
Initial CF tree
Phase 2 (optional): Condense into desirable range by building a smaller CF tree
smaller CF tree
Phase 3: Global Clustering
Good Clusters
Phase 4: (optional and off line) : Cluster Refining
Better Clusters

Figure 5.4 BIRCH Algorithm

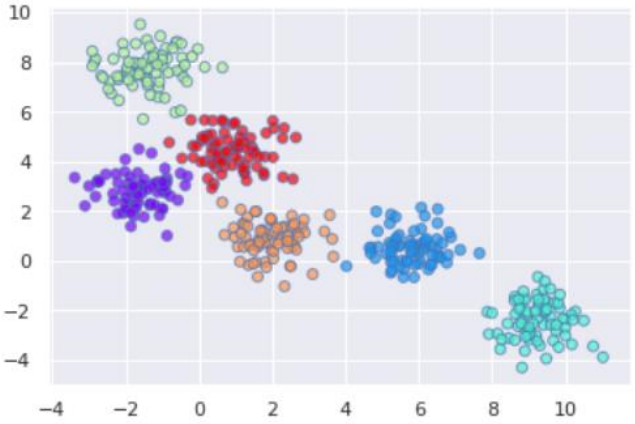


Figure 5.4 BIRCH Example

2. Classification:

2.1 K-Nearest Neighbors (kNN) [49] [50]:

The k Nearest Neighbors algorithm (kNN) is an instance-based or a lazy learning technique. It was considered one of the easiest algorithms of all machine learning [51] [52]. The rationale of kNN is that comparable samples belonging to the same class are highly likely, whereas the main concept of kNN algorithm is to first pick the closest k neighbors for each test sample, followed by using the closest k learned neighbors to estimate this test sample. The kNN algorithm has been considered as an algorithm in which no explicit training phase is needed.

A case is classified by its neighbors ' majority vote, with the case being assigned by a distance function to the most prevalent class among its closest K neighbors. If K= 1, the case will simply be assigned to its closest neighbor's class.

$$\sqrt{\sum_{i=1}^{k}(x_i - y_i)^2} \quad \sum_{i=1}^{k}|x_i - y_i| \quad \left(\sum_{i=1}^{k}(|x_i - y_i|)^q\right)^{1/q}$$

Equation 5.1 Distance

It should also be observed that for continuous variables only all three distance measurements are valid. The Hamming distance has to be used in the case of categorical factors. It also raises the problem of numerical variables standardization between 0 and 1 when there is a mix of numerical and categorical factors in the dataset.

$$D_H = \sum_{i=1}^{k}|x_i - y_i|$$

$$x = y \Rightarrow D = 0$$
$$x \neq y \Rightarrow D = 1$$

Equation 5.2 Hamming Distance

The best way to select the ideal value for K is to inspect the information first. A big K value is generally more accurate as it decreases the overall noise, but no guarantee exists. Cross-validation is another way to use an autonomous dataset to validate the K value to retrospectively determine a healthy K value. Historically, the optimal K was between 3-10 for most datasets. That's producing a lot.

2.2 Support Vector Machine (SVM) [49] [50]:

Support Vector Machines (SVM) is a classification method based on maximum linear discriminants, i.e. the goal is to find the optimum hyperplane that maximizes the class gap or margin [49]. The support vector machine algorithms objective is to discover a hyperplane in N-dimensional space(N) that classifies information points.

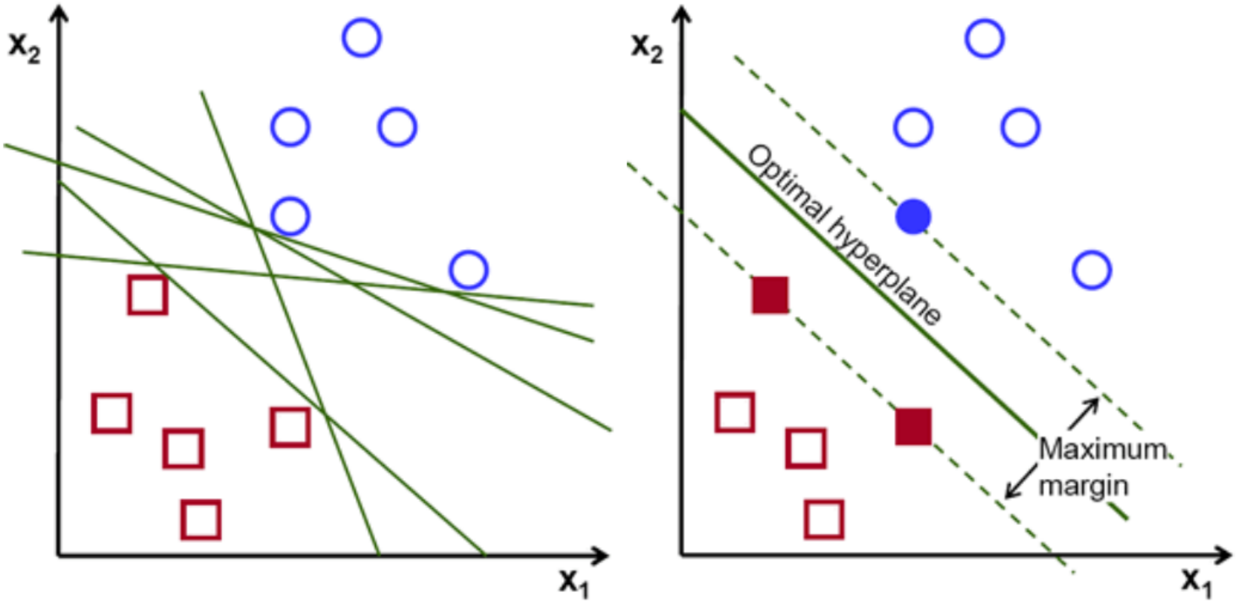


Figure 5.5 Possible hyperplanes

Many feasible hyperplanes could be selected to separate the two classes of information points. the goal is to discover a plane with the maximum margin, i.e. the highest between data points of both classes. Maximizing the margin distance offers some strengthening to be able to classify future data points with greater confidence.

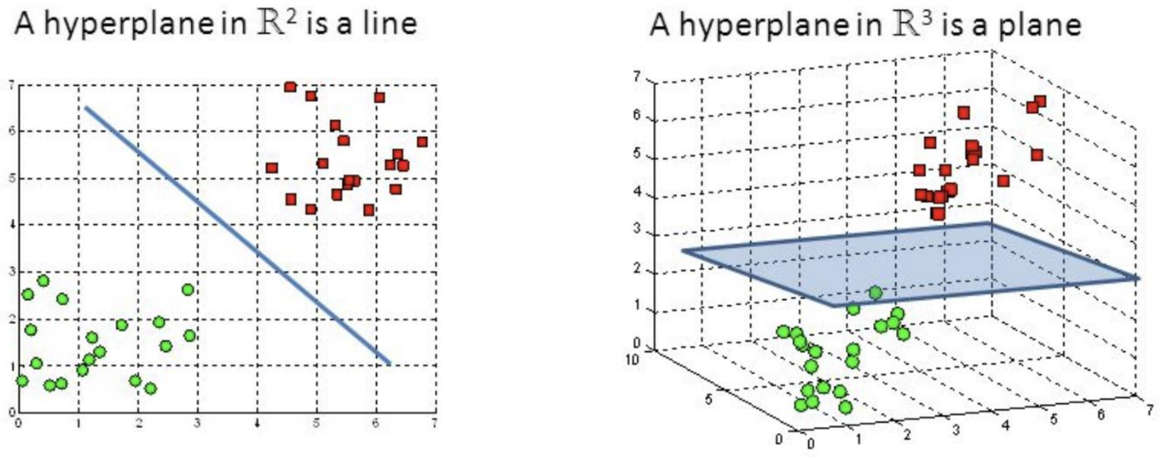


Figure 5.6 Hyperplanes in 2D and 3D feature space

Hyperplanes are limits of choice helping to classify data points. It is possible to attribute data points falling on either side of the hyperplane to distinct classes. Also, the hyperplane dimension relies on the number of features. If the number of features of the input is 2, the hyperplane is only a line. If the number of input features is 3, the hyperplane will become a two-dimensional.

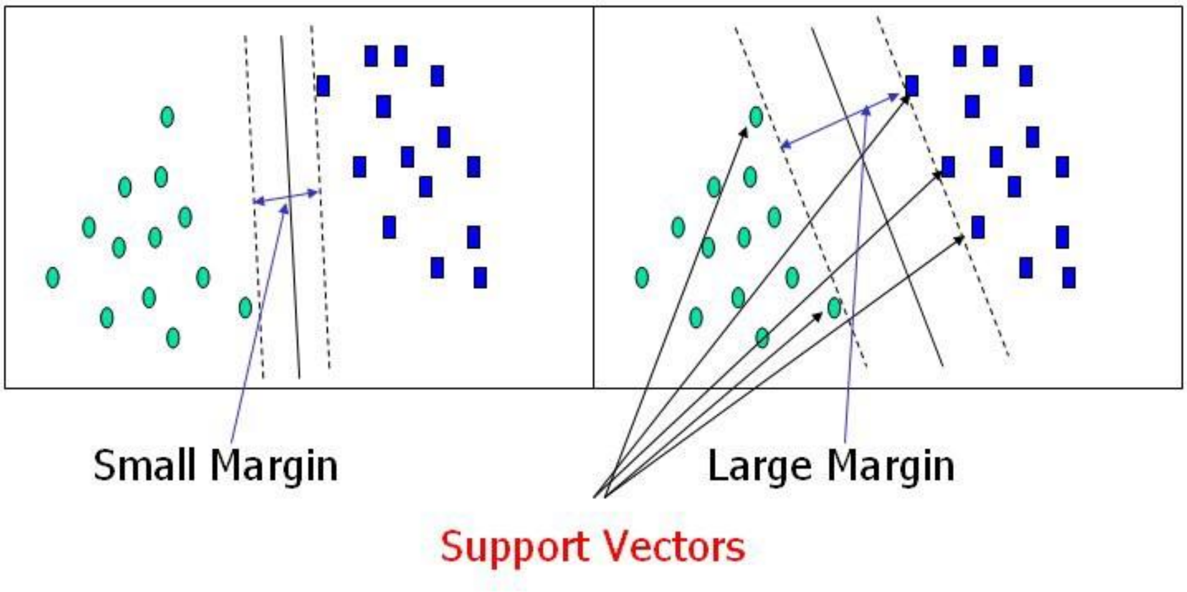


Figure 5.7 Support Vectors

Support vectors are the closest data points to the hyperplane and affect the hyperplane's position and orientation. We maximize the classifier's margin by using these support vectors. The removal of the support vectors will alter the hyperplane's position. These are the points that assist us to construct our SVM.

In logistical regression, we take the linear function output and use the sigmoid function to squash the value within the range of [0,1]. If the squashed value exceeds a limit value (0.5), a label 1 is assigned, otherwise, a label 0 is assigned. In SVM, we take the linear function output and if the yield exceeds 1, we identify it with one class and if the output is-1, we identify it with one class.

We seek to maximize the margin between the data points and the hyperplane in the SVM algorithm. The loss function helping to maximize the margin is the loss of the hinge.

$$c(x, y, f(x)) = \begin{cases} 0, & \text{if } y * f(x) \geq 1 \\ 1 - y * f(x), & \text{else} \end{cases}$$

$$c(x, y, f(x)) = (1 - y * f(x))_+$$

Equation 5.3 Hinge loss function

The cost is 0 if the predicted value is the same as the actual value. If they are not, then we will calculate the significance of the loss. We also add the cost function to a regularization parameter. The regularization parameter's goal is to balance the maximization and loss of the margin. The price functions appear as below after adding the regularization parameter.

$$min_w \lambda \parallel w \parallel^2 + \sum_{i=1}^{n} (1 - y_i \langle x_i, w \rangle)_+$$

Equation 5.4 Loss function for SVM

Now that we have the function of loss, we are taking partial weight derivatives to discover the gradients. We can update our weights using the gradients.

$$\frac{\delta}{\delta w_k} \lambda \parallel w \parallel^2 \quad = 2\lambda w_k$$

$$\frac{\delta}{\delta w_k} \big(1 - y_i \langle x_i, w \rangle\big)_+ \quad = \begin{cases} 0, & \text{if } y_i \langle x_i, w \rangle \geq 1 \\ -y_i x_{ik}, & \text{else} \end{cases}$$

Equation 5.5 Gradients

If there is no misclassification, i.e. our model properly predicts our data point class, we only need to update the gradient from the parameter of regularization.

$$w = w - \alpha \cdot (2\lambda w)$$

Equation 5.6 Gradient Update — No misclassification

When there is a misclassification, i.e. our model makes an error in our data point class forecast, we include the loss along with the regularization parameter for gradient updating.

$$w = w + \alpha \cdot (y_i \cdot x_i - 2\lambda w)$$

Equation 5.7 Gradient Update — Misclassification

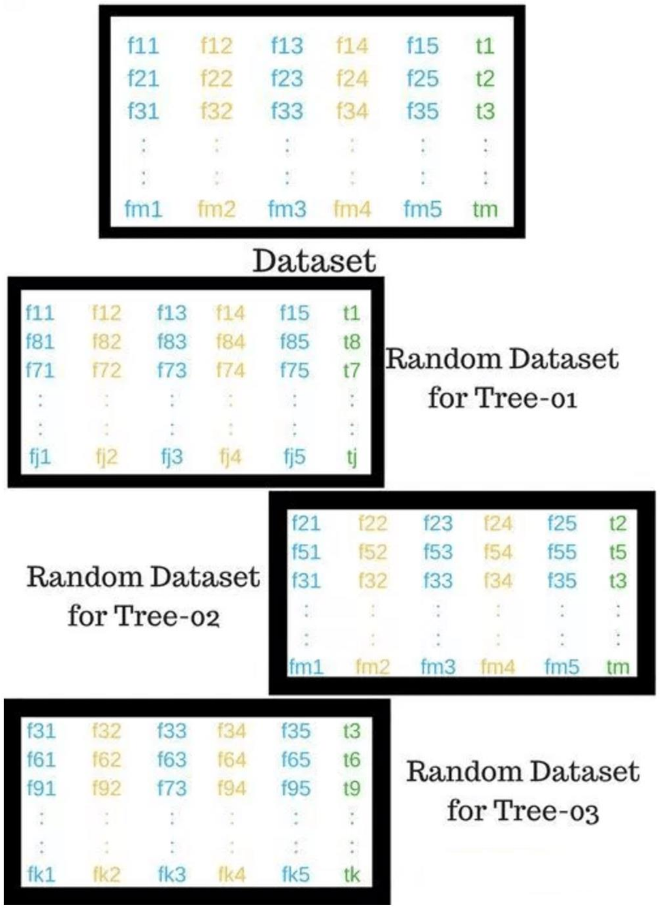
2.3 Random Forest [49] [50]:

As its name suggests, random forest comprises of a big number of individual decision trees that function as an ensemble. Each individual tree spouts a class forecast in the random forest and the class with the most votes becomes the target of our model (see figure below). The basic concept behind random forest is a simple but powerful one — the wisdom of crowds. The reason why the random forest model operates so well is that a big amount of comparatively uncorrelated models (trees) working as a panel will exceed each of the constituent models.

The Random Forest algorithm contains two stages, one is the creation of random forests, the other is the prediction of the random forest classification created in the first stage. The entire process is shown below, and using the figure is simple to comprehend.

Random Forest creation pseudocode:

1. Randomly select $\boldsymbol{k}$ features from total m features where $\boldsymbol{k} << \boldsymbol{m}$
2. Among the $\boldsymbol{k}$ features, calculate the node d using the best split point
3. Split the node into daughter nodes using the best split
4. Repeat the a to c steps until $\boldsymbol{l}$ number of nodes has been reached
5. Build forest by repeating steps 1 to 4 for $\boldsymbol{n}$ number times to create $\boldsymbol{n}$ number of trees



f11	f12	f13	f14	f15	t1
f21	f22	f23	f24	f25	t2
f31	f32	f33	f34	f35	t3
:	:	:	:	:	:
:	:	:	:	:	:
fm1	fm2	fm3	fm4	fm5	tm

Dataset

f11	f12	f13	f14	f15	t1
f81	f82	f83	f84	f85	t8
f71	f72	f73	f74	f75	t7
:	:	:	:	:	:
:	:	:	:	:	:
fj1	fj2	fj3	fj4	fj5	tj

Random Dataset for Tree-01

f21	f22	f23	f24	f25	t2
f51	f52	f53	f54	f55	t5
f31	f32	f33	f34	f35	t3
:	:	:	:	:	:
:	:	:	:	:	:
fm1	fm2	fm3	fm4	fm5	tm

Random Dataset for Tree-02

f31	f32	f33	f34	f35	t3
f61	f62	f63	f64	f65	t6
f91	f92	f73	f94	f95	t9
:	:	:	:	:	:
:	:	:	:	:	:
fk1	fk2	fk3	fk4	fk5	tk

Random Dataset for Tree-03

Figure 5.7 demonstrates the random selection process

We will create the forecast in the next phase with the creation of the random forest classifier. The pseudocode of the random forest forecast is shown below:

1. Takes the test features and use the rules of each decision tree that has been randomly generated to predict the result and stores the result (target)
2. Calculate the votes for each predicted target
3. Consider the highly voted target as the final forecast from the algorithm of random forests

2.4 Neural Network [49] [50]:

Neural networks are deep learning workhorses. And while they may look like black boxes, deep down they're attempting to do the same thing as any other model to predict things.

Neural networks are multi-layer networks of neurons that we use to classify things, make predictions, etc. Below is the diagram of a simple neural network with five inputs, 5 outputs, and two hidden layers of neurons.

Starting from the left:

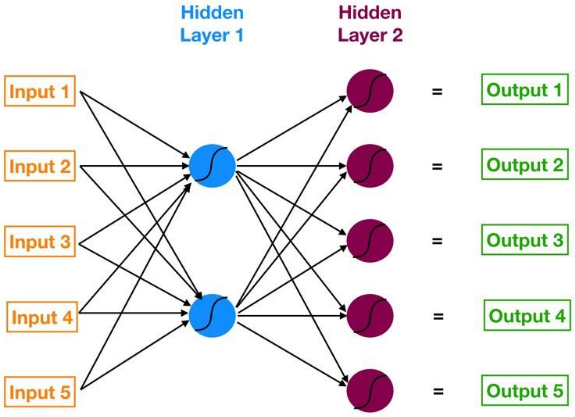


Figure 5.8 Neural network with two hidden layers

1. Input layer of our model in orange.
2. First hidden layer of neurons in blue.
3. Second hidden layer of neurons in magenta.
4. Output layer (a.k.a. the prediction) of our model in green.

The dot-connecting arrows show how all the neurons are interconnected and how data travels through the output layer from the input layer.

Later we calculate each output value step by step. We will also watch how the neural network uses a process known as backpropagation to learn from its error.

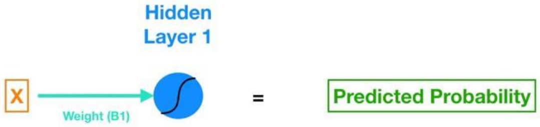


Figure 5.9 Logistic regression (with only one feature) implemented via a neural network

This is a single logistic regression characteristic expressed through a neural network. To see how they link, we can use our neural network color codes to rewrite the logistic regression equation.

$$S(x) = \frac{1}{1+e^{-x}} = \frac{e^x}{e^x+1}$$

Figure 5.10 sigmoid function

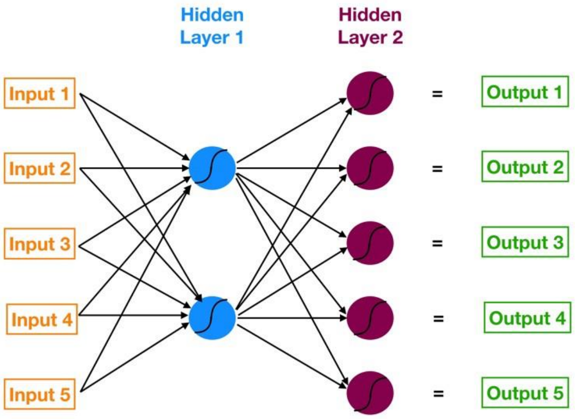


Figure 5.11 Neural network with two hidden layers

The first layer concealed is made up of two neurons. In Hidden Layer 1, we need ten links to connect all five inputs to the neurons. The following picture (below) only demonstrates the links between Input 1 and Hidden Layer 1.

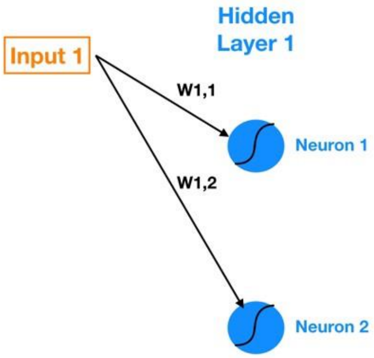


Figure 5.12 The connections between Input 1 and Hidden Layer 1

Note our notation about the weights in the connections — W1,1 refers to the weight in the connection between Input 1 and Neuron 1 and W1,2 refers to the weight in the connection between Input 1 and Neuron 2. So, the overall notation I'm going to follow is W, b indicates the weight on the link between Input a (or Neuron a) and Neuron b. Now let's calculate the outputs of each neuron in Hidden Layer 1 (known as the activations). We do so use the following formulas (W denotes weight, I_n denotes input).

Z1 = W1*In1 + W2*In2 + W3*In3 + W4*In4 + W5*In5 + Bias_Neuron1

Neuron 1 Activation = *Sigmoid*(Z1)

We can use matrix math to summarize this calculation:

$$\begin{bmatrix} W1,1 & W2,1 & W3,1 & W4,1 & W5,1 \\ W1,2 & W2,2 & W3,2 & W4,2 & W5,2 \end{bmatrix} \times \begin{bmatrix} X1 \\ X2 \\ X3 \\ X4 \\ X5 \end{bmatrix} + \begin{bmatrix} Bias1 \\ Bias2 \end{bmatrix} = \begin{bmatrix} Z1 \\ Z2 \end{bmatrix}$$

Figure 5.13 Matrix math makes our life easier

For any layer of a neural network where the prior layer is **m** elements deep and the current layer is **n** elements deep, this generalizes to:

$[W].[X] + [Bias] = [Z]$

Where $[W]$ is your n **by** m matrix of weights (the connections between the prior layer and the current layer), $[X]$ is your m **by** 1 matrix of either starting inputs or activations from the prior layer, [Bias] is your n **by** 1 matrix of neuron biases, and $[Z]$ is your n **by** 1 matrix of intermediate outputs. Once we have $[Z]$, we can apply the activation function (sigmoid in our case) to each element of $[Z]$ and that gives us our neuron outputs (activations) for the current layer.

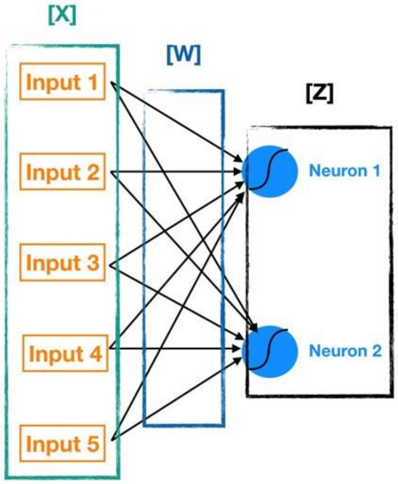


Figure 5.14 Visualizing [W], [X], and [Z]

By repeatedly calculating $[Z]$ and applying the activation function to it for each successive layer, we can move from input to output. This process is known as forward propagation. Now that we know how the outputs are calculated, it's time to start evaluating the quality of the outputs and training our neural network.

2.5 Naïve Bayes [49] [50]:

A Naive Bayes classifier is a model of probabilistic machine learning that is used for the task of classification. The classifier's crux is based on the theorem of the Bayes. Using Bayes theorem, given that B has happened, we can discover the probability of A happening. Here, the proof is B and the hypothesis is A. The hypothesis taken here is the independence of the predictors / features. That's one particular feature's presence doesn't influence the other. It is therefore called naïve.

$$P(y|X) = \frac{P(X|y)P(y)}{P(X)}$$

Equation 5.7 Bayes

The variable y is the class (play golf), which represents if it is suitable to play golf or not given the conditions. Variable X represent the parameters/features.

X is given as, $\boldsymbol{X} = (\boldsymbol{x_1}, \boldsymbol{x_2}, \boldsymbol{x_3} \ldots., \boldsymbol{x_n})$ Here $\boldsymbol{x_1}, \boldsymbol{x_2}, \boldsymbol{x_3} \ldots., \boldsymbol{x_n}$ represent the features, i.e. they can be mapped to outlook, temperature, humidity and windy. By substituting for $\boldsymbol{X}$ and expanding using the chain rule we get

$$P(y|x_1, ..., x_n) = \frac{P(x_1|y)P(x_2|y)...P(x_n|y)P(y)}{P(x_1)P(x_2)...P(x_n)}$$

Equation 5.7 Bayes

Now, by looking at the dataset, you can get the values for each and replace them in the equation. The denominator does not alter for all data entries in the dataset, it stays static. Therefore, it is possible to remove the denominator and introduce proportionality.

$$P(y|x_1, ..., x_n) \propto P(y) \prod_{i=1}^{n} P(x_i|y)$$

Equation 5.7 Bayes Equation

In our case, there are only two results in the class variable (y), yes or no. There may be instances where there may be multivariate classification. Therefore, with maximum probability, we need to discover the class y. Using the following function, we can obtain the class, given the predictors.

$$y = argmax_y P(y) \prod_{i=1}^{n} P(x_i|y)$$

Equation 5.8 obtain the class

2.6 Logistic Regression [49] [50]:

Logistic Regression is a method of' Statistical Learning,' classified as' Supervised' Machine Learning (ML) techniques dedicated to functions of classification. Over the past two centuries, it has acquired a tremendous reputation, particularly in the economic industry because of its prominent capacity to detect default. Below was given a general utilization scheme of Logistic Regression and other common Linear Classifiers.

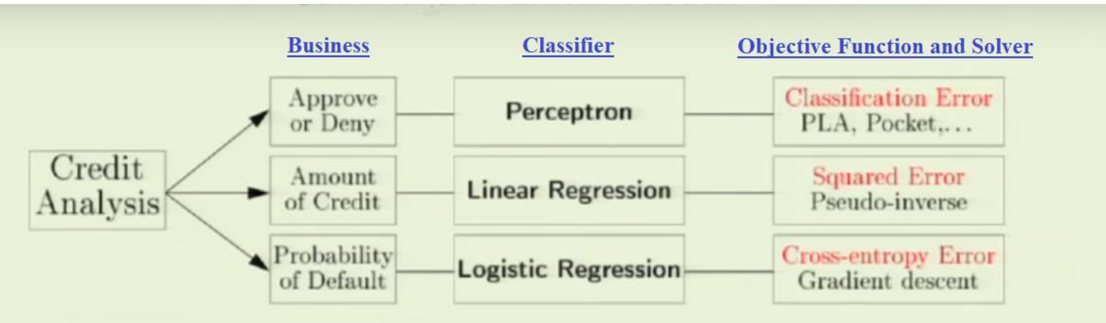


Figure 5.15 Linear Classifiers and their Usage

If we declare a classifier whose name includes the word' Regression' is used for classification, a contradiction appears, but that is why Logistic Regression is magical: using a linear regression equation to generate discrete binary outputs (Figure-2). And yes, it is also classified in the subgroup ' Discriminative Models'[53] of ML techniques such as Vector Machines Support and Perceptron where we are all concerned.

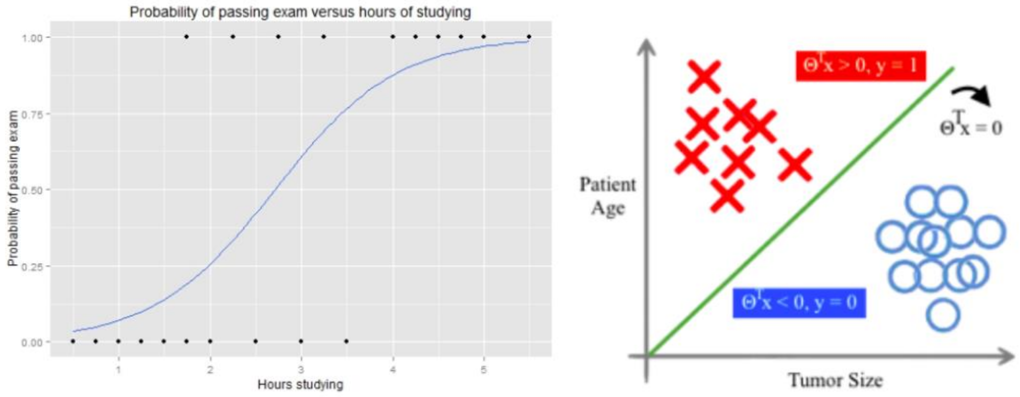


Figure 5.16 A Journey from Decision Function to Decision Boundary

Inputs x_{ij} are continuous feature-vectors (x_{i}’s) of length K, where $j = 1, \dots, k$ and $i = 1, \dots, n$. So, the input matrix is X which contains N number of inputs (data points) each contains K number of features. Inputs can be illustrated as a matrix X like below.

$$X = \begin{bmatrix} x_{11} & x_{12} & . & . & . & x_{1k} \\ x_{21} & . & & & & . \\ . & & . & & & . \\ . & & & . & & . \\ . & & & & . & . \\ x_{n1} & . & . & . & . & x_{nk} \end{bmatrix}_{n \times k}, \quad Y = \begin{bmatrix} y_1 \\ y_2 \\ . \\ . \\ . \\ y_n \end{bmatrix}_{n \times 1}$$

And output y_{i} is discrete and binary variable, such that $y \epsilon \{0,1\}$. So, we can assume that y_{i} *is Bernoulli distributed with probability parameter* p_{i}. Let’s say we have a ‘*flipping/tossing a coin*’ experiment. Supposing the coin is a fair one brings us ‘*equally likely*’ outcomes of ‘Head’ and ‘Tail’. That is the ‘*posterior*’ probabilities are:

$$P(Y = Head|X) = P(Y = Tail|X) = 0.5$$

where X is an input matrix and contains all trials/observations and their features. Since in this ‘flipping coin’ experiment does not include any independent variable (feature), our input matrix X includes only the trails we made, that is it will be a vector of ‘$n \times 1$’ where x_1 is just symbolizing the first trial rather than a concrete input.

$$X = \begin{bmatrix} x_1 \\ x_2 \\ . \\ . \\ . \\ x_n \end{bmatrix}_{n \times 1}$$

But if we replace the experiment with a ‘Credit Scoring’ one, our outcome universe will still be discrete and binary (‘Default’ and ‘Not’), however the input vector returns to a matrix again since some features has shown above!

Another radical change expecting us after shifting the experiment is the ‘*uncertainty*’ affecting the fairness that we assume for coins. Like unfair coins, credits are hosting different chances to be defaulted due to the different characteristics of obligators. So, our '*posteriors*' will not be '*equally likely*' anymore.

$$P(Y = Default|X) \neq P(Y = NotDefault|X)$$

2.7 AdaBoost [49] [50]:

Like the Random Forest Classifier, Ada-boost is another classifier of the ensemble. (Ensemble classifier consists of various classifier algorithms and the output of these classifier algorithms is the combined outcome). Ada-boost classifier combines to form a powerful classifier the weak classifier algorithm. A single algorithm could poorly classify the items. But if we combine various classifiers with a choice of practice set at each iteration and assign the correct quantity of weight in the final vote, the general classifier can get a decent precision rating. A random subset of the general training set is used to train each weak classifier. Ada-boost assigns a weight to each training product after training a classifier at any stage. Higher weight is allocated to the misclassified object so that it appears with greater probability in the next classifier's training subset. The weight is allocated to the classifier after each classifier is trained based on precision. Higher weight is allocated to more precise classifier so it will have more effect in the final results. A 50% precision classifier is provided a weight of zero, and adverse weight is provided to a classifier with less than 50% precision.

$$H(x) = sign\left(\sum_{t=1}^{T} \alpha_t h_t(x)\right)$$

$h_t(x)$: $is\ the\ output\ of\ weak\ classifier\ t\ for\ input\ x$

α_t: $is\ weight\ assigned\ to\ classifier.$

After weak classifier is trained, we update the weight of each training example with following formula:

$$D_{t+1}(i) = \frac{D_t(i)\exp(-\alpha_t y_i h_t(x_i))}{Z_t}$$

D_t: $is\ weight\ at\ previous\ level.$

We normalize the weights by dividing each of them by the sum of all the weights, Z_t. For example, if all of the calculated weights added up to 15.7, then we would divide each of the weights by 15.7 so that they sum up to 1.0 instead.

y_i is y part of training example (x_i, y_i) y coordinate for simplicity.