Chapter 3

Techniques for Segmentation, Classification and Indexing

This chapter describes the feature extraction and modeling techniques used in this thesis. Section 3.2 describes two pattern classification techniques: Support vector machine and autoassociative neural network. Section 3.3 explains the method of extracting acoustic and visual features.

3.1 Introduction

Systems that are designed for audio and video segmentation, classification and indexing for audio-video retrieval usually take segmented audio and video rather than raw audio and video data as input. Changes in audio and video signal characteristics help in detecting the category change point between different categories of broadcast audio and video. In general, considerable work has been done in Audio/Video segmentation [3], [2]. Speech is further classified into pure-speech and non-pure speech [16]. An audio stream is segmented into speech, music, environment sound, and silence [15]. Various acoustic features are short-term energy, zero crossing rates, band periodicity, and noise-frame ratio [26].
3.2 Modeling Techniques

3.2.1 Support Vector Machine (SVM)

Support vector machine (SVM) \cite{17}, \cite{30} is based on the principle of structural risk minimization (SRM). Like RBFNN, support vector machines can be used for pattern classification and nonlinear regression. SVM constructs a linear model to estimate the decision function using non-linear class boundaries based on support vectors. If the data are linearly separated, SVM trains linear machines for an optimal hyperplane that separates the data without error and into the maximum distance between the hyperplane and the closest training points. The training points that are closest to the optimal separating hyperplane are called support vectors. Fig. 3.1 shows the architecture of the SVM. SVM maps the input patterns into a higher dimensional

![Architecture of the SVM](image)

**Fig. 3.1:** Architecture of the SVM ($N_s$ is the number of support vectors).
feature space through some non-linear mapping chosen a priori. A linear decision surface is then constructed in this high dimensional feature space. Thus, SVM is a linear classifier in the parameter space, but it becomes a non-linear classifier as a result of the non-linear mapping of the space of the input patterns into the high dimensional feature space.

**SVM Principle:** Support vector machine (SVM) can be used for classifying the obtained data (Burges, 1998). SVM are a set of related supervised learning methods used for classification and regression. They belong to a family of generalized linear classifiers. Let us denote a feature vector (termed as pattern) by \( \mathbf{x} = (x_1, x_2, \cdots, x_n) \) and its class label by \( y \) such that \( y = \{+1, -1\} \). Therefore, consider the problem of separating the set of \( n \)-training patterns belonging to two classes,

\[
(x_i, y_i), \quad x_i \in \mathbb{R}^n, \quad y = \{+1, -1\}, \quad i = 1, 2, \cdots, n
\]

A decision function \( g(x) \) that can correctly classify an input pattern \( x \) that is not necessarily from the training set.

**SVM for Linearly Separable Data:** A linear SVM is used to classify data sets which are linearly separable. The SVM linear classifier tries to maximize the margin between the separating hyperplane. The patterns lying on the maximal margins are called support vectors. Such a hyperplane with maximum margin is called maximum margin hyperplane [17]. In case of linear SVM, the discriminant function is of the form:

\[
g(x) = w^T x + b
\]

such that \( g(x_i) \geq 0 \) for \( y_i = +1 \) and \( g(x_i) < 0 \) for \( y_i = -1 \). In other words, training samples from the two different classes are separated by the hyperplane \( g(x) = w^T x + b = 0 \). SVM finds the hyperplane that causes the largest separation between the decision function values from the two classes. Now the total width between two margins is
\[ \frac{2}{w^tw}, \text{ which is to be maximized. Mathematically, this hyperplane can be found by minimizing the following cost function:} \]

\[ J(w) = \frac{1}{2} w^t w \]  \hspace{1cm} (3.2)

Subject to separability constraints

\[ g(x_i) \geq +1 \text{ for } y_i = +1 \]

or

\[ g(x_i) \leq -1 \text{ for } y_i = -1 \]

Equivalently, these constraints can be re-written more compactly as

\[ y_i (w^t x_i + b) \geq 1; \quad i = 1, 2, \ldots, n \]  \hspace{1cm} (3.3)

For the linearly separable case, the decision rules defined by an optimal hyperplane separating the binary decision classes are given in the following equation in terms of the support vectors:

\[ Y = \text{sign} \left( \sum_{i=1}^{N_s} y_i \alpha_i (x x_i) + b \right) \]  \hspace{1cm} (3.4)

where \( Y \) is the outcome, \( y_i \) is the class value of the training example \( x_i \), and represents the inner product. The vector corresponds to an input and the vectors \( x_i, i = 1, \ldots, N_s \), are the support vectors. In Eq. 3.4, \( b \) and \( \alpha_i \) are parameters that determine the hyperplane.

**SVM for Linearly Non-separable Data**: For non-linearly separable data, it maps the data in the input space into a high dimension space \( x \in \mathbb{R}^I \mapsto \Phi(x) \in \mathbb{R}^H \) with kernel function \( \Phi(x) \), to find the separating hyperplane. A high-dimensional version of Eq. 3.4 is given as follows:

\[ Y = \text{sign} \left( \sum_{i=1}^{N} y_i \alpha_i K(x, x_i) + b \right) \]  \hspace{1cm} (3.5)
Determining Support Vectors: The support vectors are the (transformed) training patterns. The support vectors are (equally) close to hyperplane. The support vector are training samples that define the optimal separating hyperplane and are the most difficult patterns to classify. Informally speaking, they are the patterns most informative for the classification task. Fig. 3.2 shows a SVM example to classify a person into two classes: overweighted, not overweighted; two features are pre-defined: weight and height. Each point represents a person. Dark circle point (●) : overweighted; star point (∗) : not overweighted.

Inner Product Kernels: SVM generally applies to linear boundaries. In the case where a linear boundary is inappropriate SVM can map the input vector into a high dimensional feature space. By choosing a non-linear mapping, the SVM constructs an optimal separating hyperplane in this higher dimensional space, as shown in Fig. 3.3. The function K is defined as the kernel function for generating the inner products to
construct machines with different types of non-linear decision surfaces in the input space.

**Table 3.1: Types of SVM inner product kernels**

<table>
<thead>
<tr>
<th>Types of kernels</th>
<th>Inner Product Kernel $K(x^T, x_i)$</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Polynomial</td>
<td>$(x^T x_i + 1)^p$</td>
<td>Where $x$ is input patterns, $x_i$ is support vectors, $p$ is degree of the polynomial</td>
</tr>
<tr>
<td>Gaussian</td>
<td>$exp \left[- \frac{|x^T - x_i|^2}{2\sigma^2}\right]$</td>
<td>$\sigma^2$ is variance, $1 \leq i \leq N_s$, $N_s$ is number of support vectors,</td>
</tr>
<tr>
<td>Sigmoidal</td>
<td>$tanh(\beta_0 x^T x_i + \beta_1)$</td>
<td>$\beta_0, \beta_1$ are constant values.</td>
</tr>
</tbody>
</table>

The kernel function may be any of the symmetric functions that satisfy the Mercer’s conditions (Courant and Hilbert, 1953). There are several SVM kernel functions as given in Table 3.1.

### 3.2.2 Autoassociative Neural Network (AANN)

Neural network models such as backpropagation neural network (BPNN) and radial basis function neural network (RBFNN) are used for pattern classification because of their ability to capture the hyperspace separating the classes in the feature space. A special kind of backpropagation neural network called autoassociative neural network (AANN) can be used to capture the distribution of feature vectors in the feature space. Autoassociative neural network models are feed-forward neural networks performing an identity mapping of the input space, and are used to capture the distribution of the
Fig. 3.3: An example for SVM kernel function $\Phi(x)$ maps two dimensional input space to higher three dimensional feature space. (a) Non-linear problem. (b) Linear problem.

input data [19]. The distribution capturing ability of the AANN model is described in this section. Let us consider the five layer AANN model shown in Fig.3.4, which has three hidden layers.

\[ Z = \Phi(X) = \{ x_1^2, x_2^2, \sqrt{2}x_1x_2 \} \]

Fig. 3.4: A five layer AANN model.

In this network, the second and fourth layers have more units than the input layer. The third layer has fewer units than the first or fifth. The processing units in the first
and third hidden layer are non-linear, and the units in the second compression/hidden layer can be linear or non-linear. As the error between the actual and the desired output vectors is minimized, the cluster of points in the input space determines the shape of the hyper-surface obtained by the projection onto the lower dimensional space.

Fig. 3.5(b) shows the space spanned by the one dimensional compression layer for the 2 dimensional data shown in Fig. 3.5(a) for the network structure $2L\ 10N\ 1N\ 10N\ 2L$, where $L$ denotes a linear unit and $N$ denotes a non-linear unit. The integer value indicates the number of units used in that layer. The non-linear output function for each unit is $\tanh(s)$, where $s$ is the activation value of the unit.

The network is trained using backpropagation training algorithm [14], [20]. The solid lines shown in Fig. 3.5(b) indicate mapping of the given input points due to the one dimensional compression layer. Thus, one can say that the AANN captures the distribution of the input data depending on the constraints imposed by the structure of the network, just as the number of mixtures and Gaussian functions do in the case of Gaussian mixture models (GMM).

In order to visualize the distribution better, one can plot the error for each input data point in the form of some probability surface as shown in Fig. 3.5(c). The error $e_i$ for the data point $i$ in the input space is plotted as $p_i = \exp(-e_i/\alpha)$, where $\alpha$ is a constant. Note that $p_i$ is not strictly a probability density function, but we call the resulting surface as probability surface. The plot of the probability surface shows a large amplitude for smaller error $e_i$, indicating better match of the network for that data point.

The constraints imposed by the network can be seen by the shape the error surface takes in both the cases. One can use the probability surface to study the characteristics of the distribution of the input data captured by the network. Ideally, one would
Fig. 3.5: Distribution capturing ability of AANN model. From [14]. (a) Artificial 2 dimensional data. (b) 2 dimensional output of AANN model with the structure $2L \ 10N \ 1N \ 10N \ 2L$. (c) Probability surfaces realized by the network structure $2L \ 10N \ 1N \ 10N \ 2L$.

Like to achieve the best probability surface, best defined in terms of some measure corresponding to a low average error.

### 3.3 Feature Extraction Methods

In the design of audio and video features, one often has to deal with the problem of satisfying two conflicting goals at the same time: robustness to admissible variations on the one hand and accuracy with respect to the relevant characteristics on the other hand. Furthermore, the features should support an efficient algorithmic solution of the problem they are designed for. Acoustic and visual feature extraction plays an important role in constructing an audio and video classification system. The aim is to select features which have large between-class and small within-class discriminative power. Discriminative power of features or feature sets tells how well they can discriminate different classes. Feature selection is usually done by examining the discriminative capability of the features.
Audio and video feature extraction plays an important role in analysing and characterizing the audio and video content. Acoustic and visual features representing the audio and video information can be extracted from the audio and video signal. Feature extraction is the process of converting an audio and video signal into a sequence of feature vectors carrying characteristic information about the signal. These vectors are used as basis for various types of audio and video classification algorithms. It is typical for audio and video classification algorithms to be based on features computed on a window basis. These window based features can be considered as short time description of the signal for that particular moment in time.

The performance of a set of features depends on the application. The design of descriptive features for a specific application is the main challenge in building audio classification systems. A wide range of audio features exist for classification tasks. In this work, acoustic features MFCC and visual features color histogram are extracted for constructing an audio and video segmentation, classification and indexing system.

### 3.3.1 Mel Frequency Cepstral Coefficients

MFCCs are short-term spectral features and are widely used in the area of audio and speech processing. The mel frequency cepstrum has proven to be highly effective in recognizing the structure of music signals and in modelling the subjective pitch and frequency content of audio signals. The MFCCs have been applied in a range of audio mining tasks, and have shown good performance compared to other features. MFCC is computed by various authors in different methods. [23] computes the cepstral coefficients along with delta cepstral energy and power spectrum deviation while [24] results in 26 dimensional features. The low order MFCCs contain information of the slowly changing spectral envelope while the higher order MFCCs explain the fast variations of the envelope. Several authors report success using only the first 6 - 10
MFCCs are based on the known variation of the human ears critical bandwidths with frequency, filters spaced linearly at low frequencies and logarithmically at high frequencies to capture the phonetically important characteristics of speech and audio. To obtain MFCCs, the audio signals are segmented and windowed into short frames of 20 ms. Fig. 3.6 describes the procedure for extracting the MFCC features.

![Diagram](image)

**Fig. 3.6: Extraction of MFCC from audio signal.**

- **Mel frequency wrapping:** Magnitude spectrum is computed for each of these frames using fast Fourier transform (FFT) and converted into a set of mel scale filter bank outputs. The human ear resolves frequencies non-linearly across the audio spectrum and empirical evidence suggests that designing a front-end to operate in a similar non-linear manner improves performance. A popular solution is therefore filter-bank analysis since this provides a much more straightforward route to obtain the desired non-linear frequency resolution. However, filter-bank amplitudes are highly correlated and hence, the use of a cepstral transformation in this case is virtually mandatory.

A simple Fourier transform based filter-bank is designed to give approximately equal resolution on a mel-scale. Fig. 3.7 illustrates the general form of this
filter-bank. As can be seen, the filters used are triangular and they are equally spaced along the mel-scale which is defined by

\[ \text{Mel}(f) = 2595 \log_{10}(1 + \frac{f}{700}) \]  

To implement this filter-bank, the window of audio data is transformed using a Fourier transform and the magnitude is taken. The magnitude coefficients are then binned by correlating them with each triangular filter. Here binning means that each FFT magnitude coefficient is multiplied by the corresponding filter gain and the results are accumulated. Thus, each bin holds a weighted sum representing the spectral magnitude in that filter-bank channel.

Normally the triangular filters are spread over the whole frequency range from zero up to the Nyquist frequency. However, band-limiting is often useful to reject unwanted frequencies or avoid allocating filters to frequency regions in which there is no useful signal energy. For filter-bank analysis, lower and upper frequency cut-offs can be set. When low and high pass cut-offs are set in this way, the specified number of filter-bank channels are distributed equally on the mel-scale across the resulting pass-band.
• **Cepstrum:** Logarithm is then applied to the filter bank outputs followed by discrete cosine transformation to obtain the MFCCs. Because the mel spectrum coefficients are real numbers (and so are their logarithms), they may be converted to the time domain using the Discrete Cosine Transform (DCT). In practice the last step of taking inverse DFT is replaced by taking discrete cosine transform (DCT) for computational efficiency. The cepstral representation of the speech spectrum provides a good representation of the local spectral properties of the signal for the given frame analysis. Typically, the first 13 MFCCs are used as features.

In this work, the 12 MFCC coefficients \( c_1, c_2, c_3, \cdots, c_{12} \) for a frame are extracted from the audio signal at the segmental level. The ‘null’ MFCC coefficient \( c_0 \) is excluded from the DCT, since it represents the mean value of the input signal which carries little information. The dynamic parameters derived from 13th order static cepstral coefficients \( c_0, c_1, c_2, c_3, \cdots, c_{12} \) have been suggested and shown to improve the performance in audio classification systems [3]. These dynamic features include the delta-cepstrum (the first-order difference of the short-time static cepstrum), the delta-delta-cepstrum (the second-order difference of the static cepstrum), delta- and delta-delta-energy. Especially, the dynamic features are verified to be more robust than the static features in noisy conditions. A 39th order MFCC is used to capture the static and dynamic features of an audio signal spectrum which contains 13th order static coefficients, 13th order delta coefficients and 13th order acceleration (delta-delta) coefficients. This results in a 39 dimensional MFCC feature vector for each frame.
3.3.2 Color Histogram

A color histogram is a representation about distribution of colors in an image, derived by counting the number of pixels in each of the given set of color ranges in a typically two dimensional (2D) color space. A histogram of an image is produced first by discretization of the colors in the image into a number of bins, and counting the number of image pixels in each bin.

The histogram provides a compact summarization of the distribution of data in an image. The color histogram of an image is relatively invariant with translation and rotation about the viewing axis, and may vary very slowly with the view angle. Further, they are computationally trivial to compute. Moreover, small changes in camera viewpoint has no change in the color histograms. Hence, they are used to compare images in many applications. This work uses color histogram as visual feature. The RGB color space is quantized into 64 bins. 64 bin histogram extracted from an image is shown Fig. 3.8

In this work, RGB (888) color space is quantized into 64 dimensional feature vector. The image/video histogram is a simply bar graph of pixel intensities. The pixels are
plotted along the x-axis and the number of occurrences for each intensity represent the y-axis.

\[ p_{rk} = \frac{n_k}{n}, \quad 0 < k < L - 1 \]  

(3.8)

Where \( r_k \) kth gray level

\( n_k \) Number of pixels in the image with that gray level

\( L \) Number of levels

\( n \) Total number of pixels in the image

\( p_{rk} \) gives the probability of occurrence of gray level \( r_k \)

### 3.4 Summary

This chapter described the two pattern classification techniques and method for extracting features from the audio and video data. MFCC and color histogram features are extracted to characterize audio and video content.