A Report on the Project

Harmony Search for Feature Selection in Speech Emotion Recognition

Submitted in partial fulfillment for the award of degree

BACHELOR OF ENGINEERING

in

ELECTRONICS AND TELECOMMUNICATION

by

Pankaj Chauhan (Roll No. 18)

Gaurang Date (Roll No. 21)

Neha Nagarkoti (Roll No. 66)

Under the guidance of

Mr. Kevin D'souza

Department of Electronic & Telecommunication Engineering

St. Francis Institute of Technology, Mumbai

University of Mumbai

2017-2018

CERTIFICATE

This is to certify that the project entitled "**Harmony Search Algorithm for Feature Selection in Speech Emotion Recognition"** is a bonafide work of **Pankaj Chauhan (Roll No. 18), Gaurang Date (Roll No. 21) and Neha Nagarkoti (Roll No. 66)** submitted to the University of Mumbai in partial fulfillment of the requirement for the award of the degree of **Bachelor Of Engineering in Electronics and Telecommunication Engineering**.

(Mr. Kevin D'souza)

Project Guide

(Dr. Gautam Shah)

HOD (EXTC)

(Dr. Sincy George)

Principal

Project Report Approval for B.E.

This project entitled '***Harmony Search Algorithm for Feature Selection in Speech Emotion Recognition'*** by **Pankaj Chauhan, Gaurang Date and Neha Nagarkoti** is approved for the degree of **Bachelor of Engineering in Electronics and Telecommunication** from University of Mumbai.

Examiners:

1. ____________________

2. ____________________

Date: 03/05/2018

Place: Mumbai

ABSTRACT

Selecting significant features out of large dimensions of the original speech features is an integral part of accurate speech emotion recognition. In this paper, we proposed an automatic speech emotion classification system based on a harmony search algorithm as a feature selection strategy. First, an audio signal is divided into small frames of 20 ms and MFCC features are extracted from each frame to generate an original feature set. We employed Harmony search to derive local feature subsets for each pair of emotions. Selected subsets and original sets evaluated based on 10 fold cross-validation accuracy. Finally, each local feature subset is fed to corresponding one-against-one SVM classifier, and the majority voting method is used to classify each emotional recording. Experiments are conducted on the EMODB and IITKGP-SEHSC databases, demonstrating that size of each subset reduced to 50% of the size of original feature set, however, the accuracy remained almost same as original ones.

Keywords: Speech Emotion Recognition, Feature Selection, Harmony Search, IITKGP-SEHSC.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

ASER	Automatic Speech Emotion Recognition
SVM	Support Vector Machine
MFCCs	Mel-frequency Cepstral Cofficients
GA	Genetic Algorithm
HS	Harmony Search
HM	Harmony Memory
SAVEE	Surrey Audio-Visual Expresses Emotion
SUSAS	Speech Under Simulated and Actual Stress
LPCC	Linear Prediction Cepstral Coefficients
LFPC	Log Frequency Power Coefficients

Declaration

We declare that this written submission represents our ideas in our own words and where others' ideas or words have been included; we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in this submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Pankaj

(Pankaj Chauhan)

GRDate

(Gaurang Date)

Nagarkoti

(Neha Nagarkoti)

Date: 03/05/2018

Chapter 1: Introduction

Speech is a complex signal containing information about message, speaker, language, emotion and so on. Most existing speech systems process studio recorded, neutral speech effectively, however, their performance is poor in the case of emotional speech. This is due to the difficulty in modelling and characterization of emotions present in speech. Presence of emotions makes speech more natural. In a conversation, non-verbal communication carries important information like intention of the speaker and emotional state of the speaker. In addition to the message conveyed through text, the manner in which the words are spoken conveys essential non-linguistic information. The same textual message would be conveyed with different semantics (meaning) by incorporating appropriate emotions. Spoken text may have several interpretations, depending on how it is said. For example, the word *'Okay'* in English is used to express admiration, disbelief, consent, disinterest or an assertion. Therefore understanding the text alone is not sufficient to interpret the semantics of a spoken utterance. However, it is important that, speech systems should be able to process the non-linguistic information such as emotions, along with the message. Humans understand the intended message by perceiving the underlying emotions in addition to phonetic information by using multi-modal cues.

The applications of speech emotion recognition include humanoid robots, e-learning, automatic dialog systems and camera less mobile phones. As the volume of data grows rapidly, the traditional, manual knowledge discovery process becomes increasingly time-consuming and expensive. High dimensional data sets create problems even for automated systems. The computational complexity and search space are often increased exponentially due to high problem dimensionality (i.e., a large number of domain features). Thus reducing the dimensions of the input features and selecting significant features become integral part of speech emotion recognition.

Dimensionality reduction techniques [7] present a type of approach that attempts to reduce the overall dimensionality of the data. Which work by transforming the underlying meanings of the features, and preserving mechanisms maintain the original features. These methods search for and identify a subset of features that are necessary for emotion predication, using a dedicated evaluation measure, and are particularly beneficial for knowledge discovery tasks, as they preserve the human interpretability of the original data and the resultant, discovered knowledge.

1.1 Problem statement

Speech emotion recognition is particularly valuable for many real time applications. Although, many kinds of features have achieved good performance on recognition, dimensions of features are usually large and computation time is high.

Moreover, the naïve assumption of "more features = more knowledge" during data collection generally leads to a problem known as the *curse of dimensionality* [16]. This issue occurs when training data is not being collected at a desirable rate that is proportional to the increasing number of features. This is a frustrating issue for many machine learning methods for knowledge discovery. The abundance of features may also cause an induction algorithm to identify patterns that are in fact spurious, because of noise [6].

In this project, our approach is to select a small subset out of the thousands of speech Data which is important for accurate classification of speech emotion recognition using harmony search algorithm. The main aim of feature selection (FS) is to discover a minimal feature subset from a problem domain while retaining a suitably high accuracy (or information content) in representing the original data.

1.2 Applications

Speech emotion recognition is particularly useful for applications which require natural man machine interaction such as web movies and computer tutorial applications where the response of those systems to the user depends on the detected emotion [1, 3]. It is also useful for in-car board system where information of the mental state of the driver may be provided to the system to initiate his/her safety [2, 3]. It can be also employed as a diagnostic tool for therapists [1, 4]. It may be also useful in automatic translation systems in which the emotional state of the speaker plays an important role in communication between parties. In aircraft cockpits, it has been found that speech recognition systems trained to stressed-speech achieve better performance than those trained by normal speech [1, 5]. Speech emotion recognition has also been used in call center applications and mobile communication [5]. The main objective of employing speech emotion recognition is to adapt the system response upon detecting frustration or annoyance in the speaker's voice.

1.3 Motivation

Emotions play a major role in how a human being thinks and behaves. The emotional state of a human being influences the decision making, communication and understanding others. It is important to give clues about what we are feeling but at the same time interpreting and reacting appropriately to what others may feel.

The framework of speech emotion recognition system that consists of the following stages: selection or design of database, feature extraction and classification. The goal of feature extraction is to select an equivalent parametric representation of speech by enhancing those aspects of the speech signal that contributes significantly to the detection of emotions. The complexity of the system depends on the number of input dimensions for reduced memory and computation we consider less number of features and try to obtain high accuracy with these selected features.

1.4 Scope of the Project

Few challenges are faced while identifying emotions from audio and speech signals. These include the type of features which need to be extracted to recognize emotion, extracting features which are invariable to speaker, sentences and speaker style. The feature should also compensate for additional noise, which changes the energy distribution of the signal. Using harmony search can improve the accuracy of speech emotion recognition by reducing the number of selected feature sets thus reducing computational time [2].

1.5 Organization of the Project

Chapter 2 gives Theoretical Background of our Project and provides information about various existing methods used for feature selection.

Chapter 3 is focused on Harmony Search algorithm for feature selection and explains what is Harmony Search algorithm and Binary value representation of the same, steps to select features using harmony search in detail. We also discuss this concept using one optimization problem and also gave a pictorial representation of the same.

Chapter 4 evaluates the performance of system based on Accuracy and Cross Validation accuracy. This chapter also describes various datasets used in our project; results obtained for these datasets and provide an analysis of the same.

Chapter 5 concludes the discussion giving an insight of the future work and possible modifications in our project.

Chapter 2: Theoretical Background

2.1 Speech features

An important issue in speech emotion recognition is the extraction of speech features that efficiently characterize the emotional content of speech and at the same time do not depend on the speaker or the lexical content. Although many speech features have been explored in speech emotion recognition, researchers have not identified the best speech features for this task.

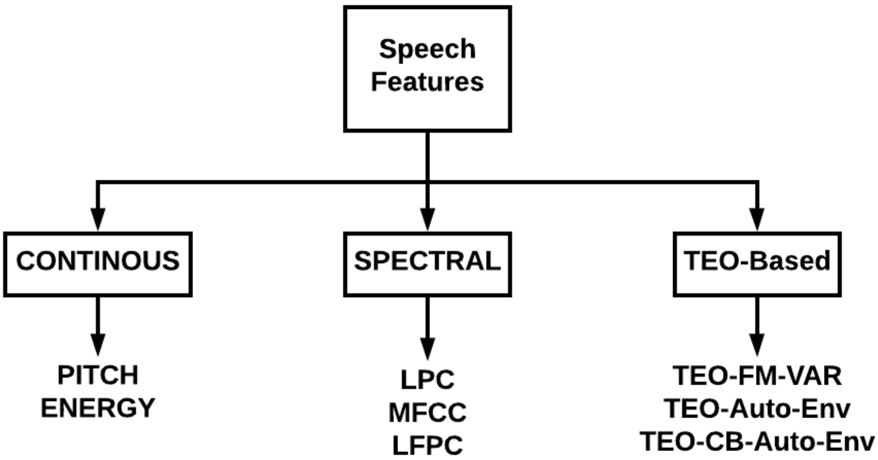


Fig 2.1: Types of feature [1]

(a) Vocal tract spectrum features

The vocal cords are the source of speech production in human beings. The sound that is created in the vocal tract gets shaped in the frequency domain by the frequency response of the vocal tract. Thus, the most important information is obtained in the spectral shape of the vocal tract.

In addition to time-dependent acoustic features such as pitch and energy, spectral features are often selected as a short-time representation for speech signal. It is recognized that the emotional content of an utterance has an impact on the distribution of the spectral energy across the speech range of frequency [10]. For example, it is reported that utterances with happiness emotion have high energy at high frequency range while utterances with the sadness emotion have small energy at the same range [9, 11]. While the energy level of happiness and anger are almost same as they have high initial energy.

(1) LPCC

LPCC is an example of vocal tract spectrum feature that equates the vocal tract to an all pole model. In [3], a comparison between LPC and another vocal tract spectrum feature i.e. Log frequency Power Coefficients (LFPC) shows LFPC to be a better choice, as using this feature the classification is more accurate. The purpose of using LPCCs is to consider vocal tract characteristics of the speaker, while performing automatic emotion recognition.

(2) LPFC

LFPC represents the energy distribution of a speech signal in different log frequency bands is a good indication of emotion as the distribution of energy is dependent on emotion type. Most emotion has different energy levels, while some have the same initial energy. Hence it is difficult to classify emotions accurately based on their energies.

(3) MFCC

The mel-frequency cepstrum (MFC) is a representation of the short-term power spectrum of a sound, based on a linear cosine transform of a log power spectrum on a nonlinear mel scale of frequency.

Mel-frequency cepstral coefficients (MFCCs) are coefficients that collectively make up an MFC. They are derived from a type of cepstral representation of the audio clip. The difference between the cepstrum and the mel-frequency cepstrum is that in the MFC, the frequency bands are equally spaced on the mel scale, which approximates the human auditory system's response more closely than the linearly-spaced frequency bands used in the normal cepstrum. There are 13 mel frequency cepstral coefficients. The Mel scale relates perceived frequency, or pitch, of a pure tone to its actual measured frequency. Humans are much better at discerning small changes in pitch at low frequencies than they are at high frequencies. Incorporating this scale makes our features match more closely what humans hear.

(b) Prosodic Features

(1) Jitter and shimmer

Jitter, is an indication of the variability or perturbations of fundamental frequency as a result of deviation from true periodicity of an assumed periodic signal, is different from shimmer. Shimmer is an indication of variability or perturbations of amplitude of a given speech signal.

(2) Pitch

Pitch is the perceived fundamental frequency of a speech signal; pitch is dependent on the tension created on the vocal fields due to the non-linear air flow during speech generation. It can be extracted using autocorrelation, cepstrum and the most popular RAPT (Robust Algorithm for Pitch Tracking).

(3) Energy

The significance of energy as a suitable feature for speech emotion recognition arises from the fact that it is related to the arousal level of emotions. Its value in emotion recognition is especially useful when its contours are considered. The emotion of anger has highest energy level while disgust has the least energy level.

(c) Non- linear features

Under Stressful condition Teager observed that there are additional excitation signals across the spectrum as harmonics because of nonlinear air flow in the vocal tract due to stressfully expressed or an angry speech. This nonlinear feature is called Teager-Energy operator. The feature is obtained by applying least square fit instead of applying DCT in TEO feature. The feature was tested on the publically available Berlin Emotion Database (EMO-DB) using a GMM classifier. TEO Slope feature based emotion recognition system shows a significant improvement over MFCC based baseline system by 2% and TEO feature based system by 6% in terms of overall accuracy. Also the feature set obtained by fusion of MFCC, its delta and TEO Slope was evaluated and 60% accuracy was obtained [12].

(d) Fourier Parameters

Fourier parameter model uses the perceptual content of voice quality and the first- and second-order differences for speaker-independent speech emotion recognition. Experimental results show that the proposed Fourier parameter (FP) features are effective in identifying various emotional states in speech signals. They improve the recognition rates over the methods using Mel frequency cepstral coefficient (MFCC) features by 16.2, 6.8 and 16.6 points on the German database (EMODB), Chinese language database (CASIA) and Chinese elderly emotion database (EESDB). In particular, when combining FP with MFCC, the recognition rates can be further improved on the aforementioned databases by 17.5, 10 and 10.5 points, respectively. [13]

2.2 Speech Emotion Databases

The databases in speech emotion recognition are broadly classified into acted, naturalistic and induced database. In acted database, professional speakers, who can emote given a specific situation, are considered.

Language	Name of the database	Type of database	Characteristics
German	Emo-DB	Acted	1. 535 Utterances of 5 female and 5 male speakers 2. The age of the speakers range from 21-35 years 3. Anger, boredom, disgust, fear, happiness, sadness and neutral explored.
English	SAVEE	Acted	1. 480 Utterances of 4 male actors 2. The age of the speaker range from 27-31 3. Anger, Disgust, Fear, Happiness, sadness, Surprise, Neutral.

English	SUSAS	Acted	1. 6,000 utterances, 32 actors (13 females + 19 males) 2. Four stress styles: Simulated Stress, Calibrated Workload Tracking Task, Acquisition and Compensatory Tracking Task, Amusement Park Roller-Coaster, Helicopter Cockpit Recordings
Hindi	IITKGP-SEHSC	Acted	1. The total number of utterances in the database is 12000 (15 sentences X 8 emotions X 10 artists X 10 sessions). Each emotion has 1500 utterances. 2. The eight emotions considered for recording this database are anger, disgust, fear, happy, neutral, sadness, sarcastic and surprise.

Table I: Characteristics of database in different languages

2.3 Feature Selection

Emotion Recognition as a part of Machine learning requires statistical, significant and reliable results, which may be difficult to achieve if similarity creeps in, given the dimensionality of the input data as its dimensionality grows. Thus reducing the dimensions of the input features become integral part of speech emotion recognition. It is broadly classified into two categories viz. feature selection and feature reduction. In feature selection, the useful subset of original feature is chosen.

2.3.1 Dimensionality reduction

(a) Principal component analysis (PCA)

The main linear technique for dimensionality reduction, principal component analysis, performs a linear mapping of the data to a lower-dimensional space in such a way that the variance of the data in the low-dimensional representation is maximized. In practice, the covariance (and sometimes the correlation) matrix of the data is constructed and the eigen vectors on this matrix are computed. The eigen vectors that correspond to the largest eigen values (the principal components) can now be used to reconstruct a large fraction of the variance of the original data. Moreover, the first few eigen vectors can often be interpreted in terms of the large-scale physical behavior of the system. The original space (with dimension of the number of points) has been reduced (with data loss, but hopefully retaining the most important variance) to the space spanned by a few eigenvectors.

Principal component analysis (PCA) involves a mathematical procedure that transforms a number of possibly correlated variables into a smaller number of uncorrelated variables called principal components. The first principal component accounts for as much of the variability in the data as possible, and each succeeding component accounts for as much of the remaining variability as possible. Depending on the field of application, it is also named the discrete Karhunen-Loève transform (KLT), the Hostelling transform or proper orthogonal decomposition (POD).PCA was invented in 1901 by Karl Pearson [14]. Now it is mostly used as a tool in exploratory data analysis and for making predictive models. PCA involves the calculation of the eigen value decomposition of a data covariance matrix or singular value decomposition of a data matrix, usually after mean centering the data for each attribute. The results of a PCA are usually discussed in terms of component scores and loadings. PCA is the simplest of the true eigenvector-based multivariate analyses. Often, its operation can be thought of as revealing the internal structure of the data in a way which best explains the variance in the data. If a multivariate dataset is visualized as a set of coordinates in a high-dimensional data space (1 axis per variable), PCA supplies the user with a lower-dimensional picture, a "shadow" of this object when viewed from its (in some sense) most informative viewpoint. PCA is closely related to factor analysis; indeed, some statistical packages deliberately conflate the two techniques. True factor analysis makes different assumptions about the underlying structure and solves eigenvectors of a slightly different matrix.

(b) Linear discriminant analysis (LDA)

Linear discriminant analysis (LDA) is a generalization of Fisher's linear discriminant, a method used in statistics, pattern recognition and machine learning to find a linear combination of features that characterizes or separates two or more classes of objects or events.

The goal of a LDA is often to project a feature space (a dataset n-dimensional samples) into a smaller subspace k (where $k \leq n-1$), while maintaining the class-discriminatory information. In general, dimensionality reduction does not only help to reduce computational costs for a given classification task, but it can also be helpful to avoid over fitting by minimizing the error in parameter estimation.

2.3.2 Feature selection vs. dimensionality reduction

Dimensionality reduction is typically choosing a basis or mathematical representation within which you can describe most but not all of the variance within your data, thereby retaining the relevant information, while reducing the amount of information necessary to represent it. There are a variety of techniques for doing this including but not limited to PCA, ICA, and Matrix Feature Factorization. These will take existing data and reduce it to the most discriminative components. These all allow you to represent most of the information in your dataset with fewer, more discriminative features [15].

Feature Selection is about choosing some features which are highly discriminative. This has a lot more to do with feature engineering than analysis, and requires significantly more work. It requires an understanding of what aspects of your dataset are important in whatever predictions you're making, and which aren't. Feature extraction usually involves generating new features which are composites of existing features.

2.3.2 Feature selection

The main aim of feature selection (FS) is to discover a minimal feature subset from a problem domain while retaining a suitably high accuracy (or information content) in representing the original data [16]. When analyzing data that has a very large number of features [24], it is difficult to identify and extract patterns or rules due to the high inter-dependency amongst individual features, or the complex behavior of combined features. Techniques to perform tasks such as text processing, data classification and systems control [20, 21, 22, 23] can benefit greatly from FS which directly addresses the problem of high dimensionality, since the noisy, irrelevant, redundant or misleading features may now be removed [18]. FS is pervasive, in the sense that it is not restricted as being purely a type of data mining technique. This characteristic is reflected by its use in a wide range of real-world applications [8, 19], a few example areas are illustrated in figure.

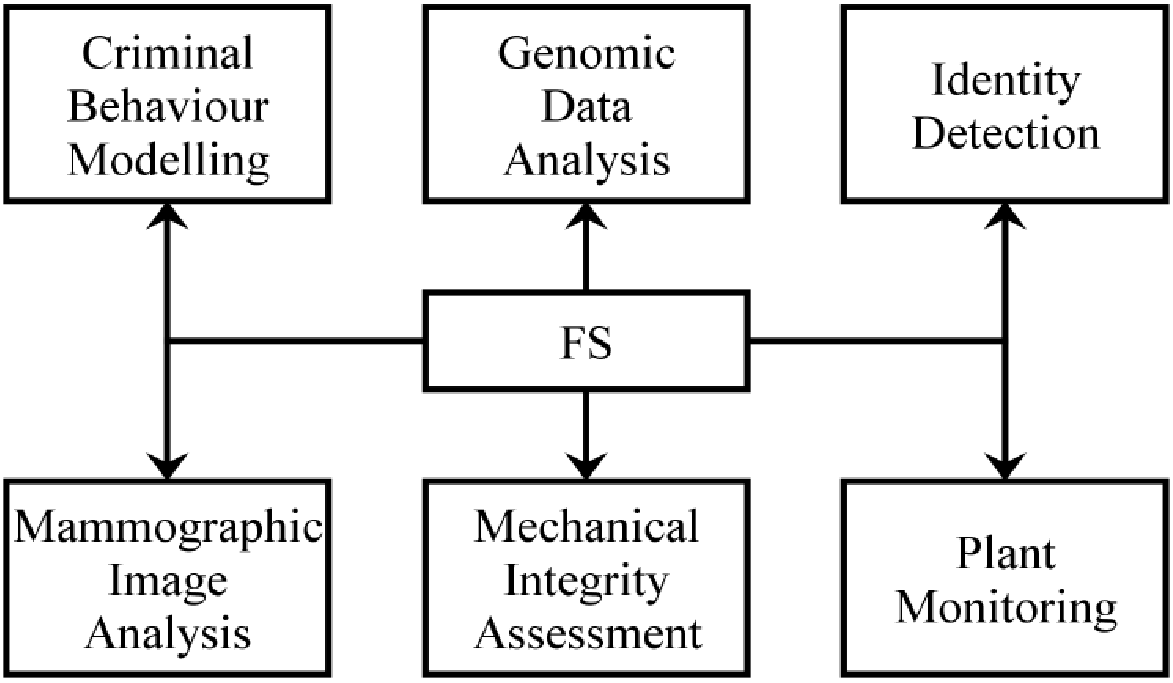


Figure 2.2: Selection of real-world applications of FS [53]

In the context of FS, an information system generally consists of a fixed number objects, and each object is described by a set of features. Features can be either qualitative (discrete-valued) or quantitative (real-valued). For a given data set with n features, the task of FS can be seen as a search for the "optimal" subset of features through the competing $2n$ candidates. In general, optimality is subjective depending on the problem at hand. The subset that is selected as optimal using one particular evaluation function, may not be equivalent to that selected by another. Various techniques [26] have been developed in the literature to judge the quality of the discovered feature subsets, several of which rank the features based on a certain importance measure, e.g., information gain [26], chi-square [36], rough set and fuzzy-rough set-based dependency [29, 30], and symmetrical uncertainty [35].

Recent trends in developing FS methods focus on evaluating a feature subset as a whole, forming an alternative type of approach to the aforementioned. Popular methods include the group-based fuzzy-rough FS (FRFS) [31, 34], probabilistic consistency-based FS (PCFS) [26], and correlation-based FS (CFS) [27]. These techniques (together with the individual feature-based methods) are often collectively classified as the filter-based techniques. They are typically used as a pre-processing step, and are independent of any learning algorithm that may be subsequently employed. In contrast, wrapper-based [28, 32] and also, hybrid algorithms [37] are used in conjunction with a learning or data mining algorithm, which is employed in place of an evaluation metric as used in the filter-based approach.

There are two main types of feature selection algorithms: wrapper type and filter type.

(a) Wrapper method

Wrappers tend to perform better in selecting features since they take the model hypothesis into account by training and testing in the feature space. This leads to the big disadvantage of wrappers, the computational inefficiency which is more apparent as the feature space grows. Unlike filters, they can detect feature dependencies. It is method in which the features are selected using the classifier. It is mainly used as post process method i.e. works on features after classification is done to optimize it.

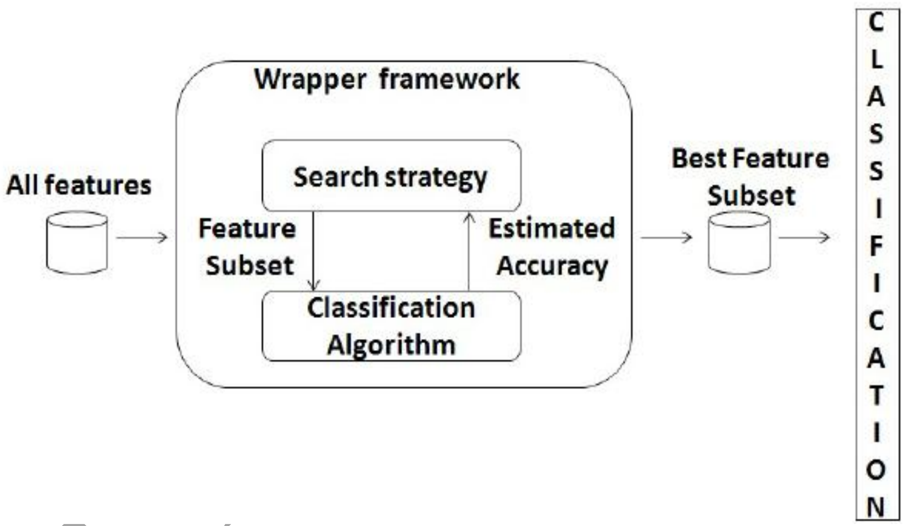


Fig 2.3: Wrapper type Feature Selection

(b) Filter method

Filters work without taking the classifier into consideration. This makes them very computationally efficient. They are divided into multivariate and univariate methods. Multivariate methods are able to find relationships among the features, while univariate methods consider each feature separately. Gene ranking is a popular statistical method. The following methods were proposed in order to rank the genes in a dataset based on their significance [38]. It is method in which the selection of features is independent of the classifier used. They are mainly used as a pre-process method.

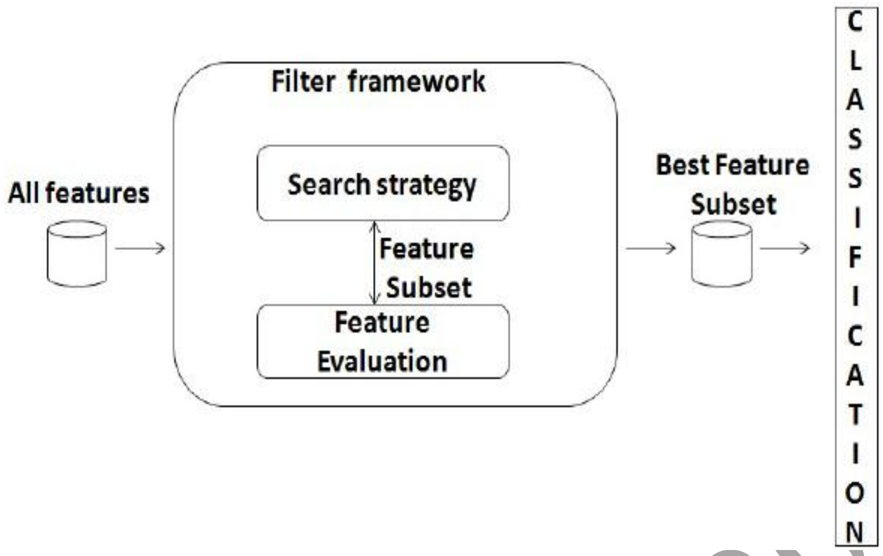


Fig 2.4: Filter type Feature Selection

2.3.3 Popular Metaheuristic Algorithms

We will now briefly introduce some popular metaheuristic algorithms, and we will try to see how they work and what the advantages of each over other are.

(a) Evolutionary algorithm

Evolutionary algorithms are the name for a subset of evolutionary computation [39]. They are the search methods that were inspired from Charles Darwin's natural selection and survival of the fittest. Evolutionary algorithms are population-based search algorithms and they use genetic operators to a certain degree. These operators typically include crossover or reproductive recombination, mutation, inheritance and selection based upon their fitness.

Evolutionary programming uses arbitrary data structures and representations tailored to suit a particular problem domain, and they are combined with the essence of genetic algorithms so as to solve generalized complex optimization problems. Evolutionary strategy is another class of nature-inspired evolutionary optimization techniques. It mainly uses mutation, selection and element-wise average for intermediate recombination as the genetic operators. It has been applied to a wide range of optimization problems [49].

EP is very similar to ES, but does not have the recombination operator. The main difference between EP and other methods is that EP does not use exchange of string segments, and thus there is no crossover or recombination between individuals. The primary genetic operator is mutation, often using Gaussian mutation for real-valued functions. However, its modern variants are more diverse [49].

(b) Simulated annealing

It was first developed by Kirkpatrick et al. in 1983 [40], inspired by the annealing process of metals during heat treatment and also by Metropolis algorithms for Monte Carlo simulations. The basic idea of the SA algorithm is similar to dropping a bouncing ball over a landscape. As the ball bounces and loses energy, it will settle down at certain local minimum. If the ball is allowed to bounce enough time and loose energy slowly enough, the ball may eventually fall into the globally lowest location, and hence the minimum will be reached [49].

The optimization process typically starts from an initial guess with higher energy. It then moves to another location randomly with slightly reduced energy. The move is accepted if the new state has lower energy and the solution improves with a better objective or lower value of the objective function for minimization [49].

(c) Ant Colony optimization

Another population-based metaheuristic algorithm is the ant colony optimization (ACO) [34-36] which was first formulated by Dorigo and further developed by other pioneers. This algorithm was based upon the characteristics of behaviours of social ants. For discrete routing and scheduling problems, multiple ants are often used. Each virtual ant will preferably choose a route covered with higher pheromone concentration, and it also deposits more pheromone at the same time. If there is no previously deposited pheromone, then each ant will move randomly. In addition, the pheromone concentration will decrease gradually due to the evaporation, often with a constant evaporation rate [49].

2.4 Harmony Search Algorithm

HS was first developed by Geem et al. in 2001 [42], and is based on natural musical performance processes that happen when a musician searches for a better state of harmony, such as that during jazz improvisation.

Recently music-inspired harmony search algorithm has been proposed and vigorously applied to various scientific and engineering applications such as music composition, Sudoku puzzle solving, tour planning, web page clustering, structural design, water network design, vehicle routing, dam scheduling, ground water modeling, soil stability analysis, ecological conservation, energy system dispatch, heat exchanger design, transportation energy modeling, pumping operation, model parameter calibration, satellite heat pipe design, medical physics, etc. [45,46].

In HS algorithm, the harmony memory (HM) stores the feasible vectors, which are all in the feasible space. When a musician improvises one pitch, usually one of three rules is used:

i. Generating any one pitch from his/her memory, i.e. choosing any one value from harmony memory, defined as memory consideration;

ii. Generating a nearby pitch of one pitch in his/her memory, i.e. choosing an adjacent value of one value from harmony memory, defined as pitch adjustments;

iii. Generating totally a random pitch from possible sound ranges, i.e. choosing totally random value from the possible value range, defined as randomization [47].

Geem et al. formalized these three options into Harmony Search as a Metaheuristic Algorithm 3 quantitative optimization process:

(1) Usage of harmony memory

(2) Pitch adjusting

(3) Randomization

(1) Usage of Harmony Memory

The usage of harmony memory (HM) is important because it ensures that good harmonies are considered as elements of new solution vectors. In order to use this memory effectively, the HS algorithm adopts a parameter r_{accept} is an element of $[0,1]$, called harmony memory considering (or accepting) rate. If this rate is too low, only few elite harmonies are selected and it may converge too slowly. If this rate is extremely high (near 1), the pitches in the harmony memory are mostly used, and other ones are not explored well, leading not into good solutions. Therefore, typically, we use $r_{accept} = 0.7 \sim 0.95$ [49].

(2) Pitch Adjustment

The second component is the pitch adjustment which has parameters such as pitch bandwidth b_{range} and pitch adjusting rate r_{pa}. As the pitch adjustment in music means changing the frequency, it means generating a slightly different value in the HS algorithm [47]. In theory, the pitch can be adjusted linearly or nonlinearly, but in practice, linear adjustment is used. So we have

$$x_{new} = x_{old} + b_{range} \times \varepsilon$$

Where x_{old} is the existing pitch stored in the harmony memory, and x_{new} is the new pitch after the pitch adjusting action. This action produces a new pitch by adding small random amount to the existing pitch [45]. Here ε is a random number from uniform distribution with the range of $[-1, 1]$. Pitch adjustment is similar to the mutation operator in genetic algorithms. We can assign a pitch-adjusting rate (r_{pa}) to control the degree of the adjustment. A low pitch adjusting rate with a narrow bandwidth can slow down the convergence of HS because of the limitation in the exploration of only a small subspace of the whole search space. On the other hand, a very high pitch-adjusting rate with a wide bandwidth may cause the solution to scatter around some potential optima as in a random search. Thus, we usually use $r_{pa} = 0.1{\sim}0.5$ in most applications [49].

(3) Randomization

It is to increase the diversity of the solutions. Although the pitch adjustment has a similar role, it is limited to certain area and thus corresponds to a local search. The use of randomization can drive the system further to explore various diverse solutions so as to attain the global optimality [49].

In the HS algorithm, diversification is essentially controlled by the pitch adjustment and randomization -- here there are two subcomponents for diversification, which might be an important factor for the high efficiency of the HS method. The first subcomponent of playing a new pitch (or generating a new value) via randomization would be at least at the same level of efficiency as in other algorithms that handle randomization. However, an additional subcomponent for HS diversification is the pitch adjustment operation performed with the probability of r_{pa} Pitch adjustment is carried out by tuning the pitch within a given bandwidth. A small random amount is added to or subtracted from an existing pitch (or solution) stored in

HM. Essentially, pitch adjustment is a refinement process of local solutions. Both memory consideration and pitch adjustment ensure that good local solutions are retained while the randomization makes the algorithm to explore global search space effectively. The subtlety is the fact that HS operates controlled diversification around good solutions, and intensification as well [49].

HS algorithm proved to be very successful in a wide range of optimization problems, such as water distribution and games, and showed better performance in comparison with other traditional optimization techniques.

2.4.1 Flowchart and Pseudo code

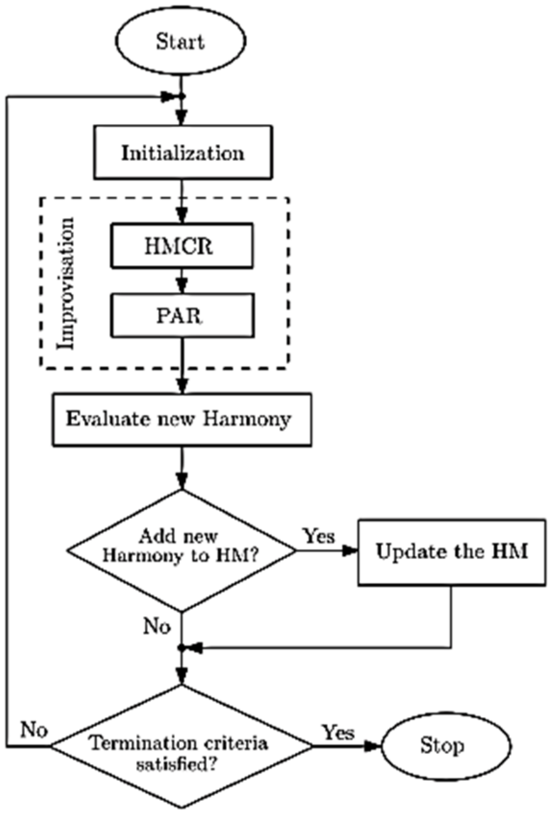


Fig 2.5: Flow chart of Harmony Search Algorithm [48]

```
Harmony Search
begin
  Define objective function f(x), x=(x1,x2, …,xd)^T
  Define harmony memory accepting rate (r_accept)
  Define pitch adjusting rate (r_pa) and other parameters
  Generate Harmony Memory with random harmonies
  while ( t<max number of iterations )
    while ( i<=number of variables)
      if (rand<r_accept), Choose a value from HM for the variable i
        if  (rand<r_pa), Adjust the value by adding certain amount
        end if
      else  Choose a random value
      end if
    end while
    Accept the new harmony (solution) if better
  end while
  Find the current best solution
end
```

Fig. 2.6: Pseudo code of the Harmony Search algorithm [49]

4.3 Support Vector Machine

Support vector machine (SVM) classifiers often have superior recognition rates in comparison to other classification methods. However, the SVM was originally developed for binary decision problems, and its extension to multi-class problems is not straightforward. How to effectively extend it for solving multiclass classification problem is still an on-going research issue. The popular methods for applying SVMs to multiclass classification problems usually decompose the multi-class problems into several two-class problems that can be addressed directly using several SVMs.

4.3.1 Binary SVM

The support vector machine is originally a binary classification method developed by Vapnik and colleagues at Bell laboratories, with further algorithm improvements by others. For a binary problem, we have training data points: $\{x_i, y_i\}$, $i=1,...,l$, y_i $\{-1, 1\}$, x_i R^d .

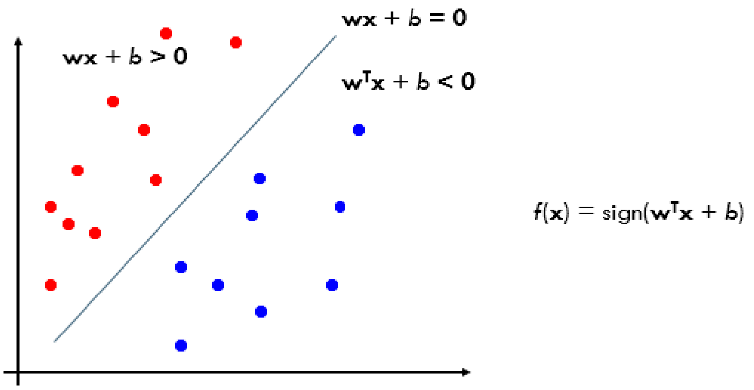
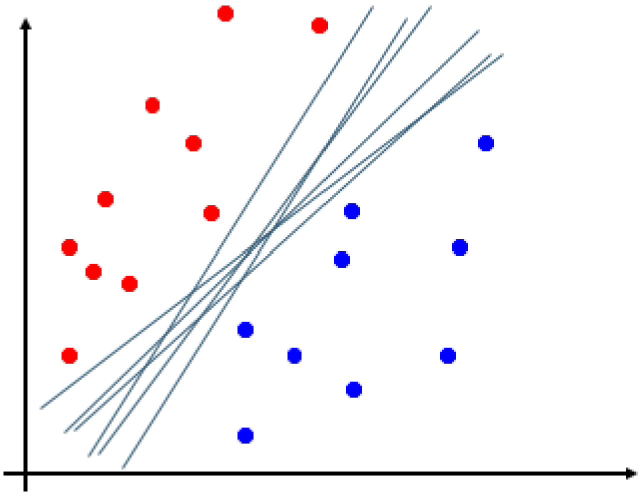


Fig. 2.7: Binary Classification

But then there are lots of hyper planes that can separate data. Exactly which one should be used?



Suppose we have some hyper plane which separates the positive from the negative examples (a "separating hyper plane"). The points x which lie on the hyper plane satisfy $w \cdot x + b = 0$, where w is normal to the hyper plane, $|b|/||w||$ is the perpendicular distance from the hyper plane to the origin, and $||w||$ is the Euclidean norm of w. Let d_+ (d_-) be the shortest distance from the separating hyper plane to the closest positive (negative) example. Define the "margin" of a separating hyper plane to be $d_+ + d_-$. For the linearly separable case, the support vector algorithm simply looks for the separating hyper plane with largest margin.

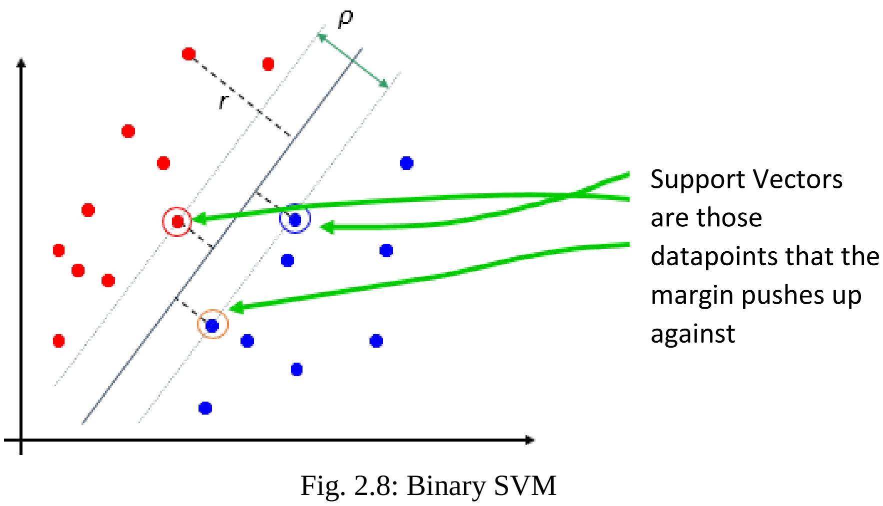


Fig. 2.8: Binary SVM

Distance from example $\mathbf{x}_i$ to the separator is:

$$r = \frac{\mathbf{w}^T \mathbf{x}_i + b}{\|\mathbf{w}\|}$$

Examples closest to the hyperplane are support vectors.

Margin ρ of the separator is the distance between support vectors.

Maximizing the margin is good. This implies that only support vectors matter; other training examples are ignorable.

Let training set $\{(\mathbf{x}_i, y_i)\}$ $i=1\ldots n$, $x_i \in \mathbf{R}^d$, $y_i \in \{-1, 1\}$ be separated by a hyperplane with margin ρ. Then for each training example $(\mathbf{x}_i, y_i)$:

$$\mathbf{w}\,\mathbf{x}_i + b \leq -\frac{\rho}{2} \quad \text{if } y_i = -1$$

$$\mathbf{w}\,\mathbf{x}_i + b \geq \frac{\rho}{2} \quad \text{if } y_i = 1$$

ie.

$$|y_i(\,w\,x_i + b\,)| \geq \frac{\rho}{2}$$

For every support vector $\mathbf{x}_s$ the above inequality is an equality. After rescaling $\mathbf{w}$ and b by $\rho/2$ in the equality, we obtain that distance between each $\mathbf{x}_s$ and the hyperplane is

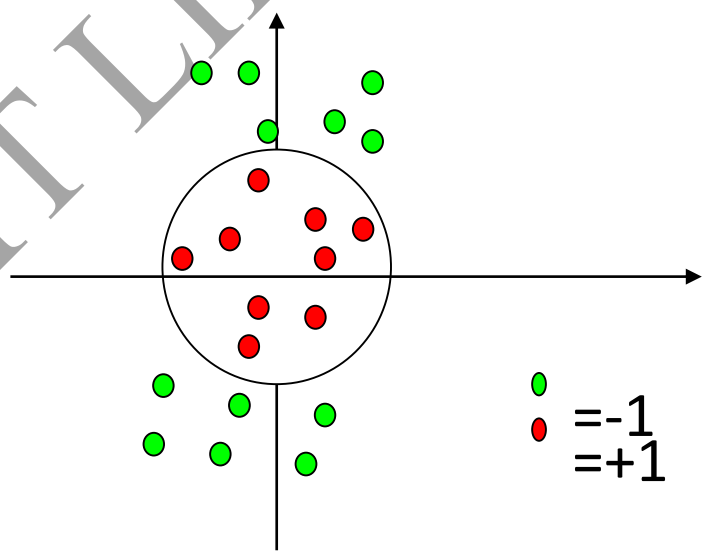
$$r = \frac{y_s(\mathbf{w}^T\mathbf{x}_s + b)}{\|\mathbf{w}\|} = \frac{1}{\|\mathbf{w}\|}$$

Then the margin can be expressed through (rescaled) $\mathbf{w}$ and b as:

$$\rho = 2r = \frac{2}{\|\mathbf{w}\|}$$

4.3.2 Kernalization

Problem with linear binary SVM arises when data is non-linearly separated like shown: The idea behind kernalization is that the original feature space can always be mapped to some



higher-dimensional feature space where the training set is separable as:

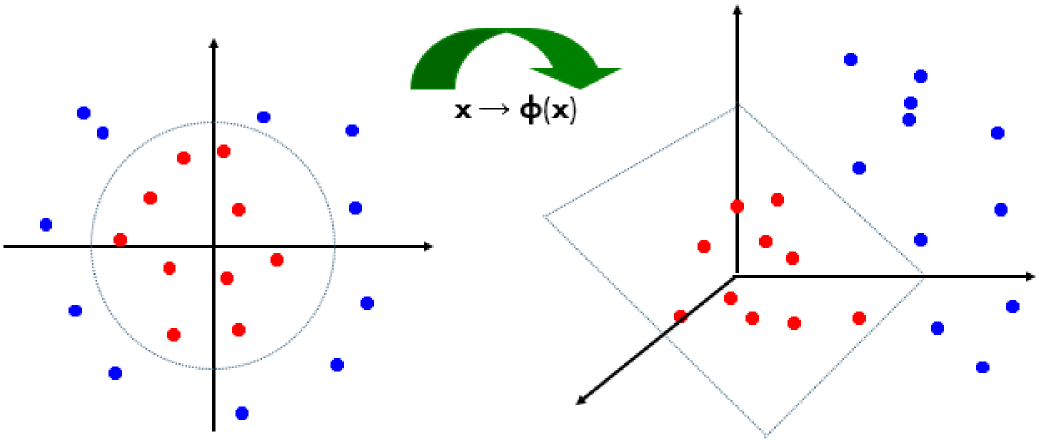


Fig. 2.9: Kernalization

For many mappings from a low-D space to a high-D space, there is a simple operation on two vectors in the low-D space that can be used to compute the scalar product of their two images in the high-D space.

Some commonly used kernels are:

Polynomial: $K(\mathbf{x},\mathbf{y}) = (\mathbf{x}.\mathbf{y} + 1)^p$

Parameters that the user must choose

Gaussian radial basis function: $K(\mathbf{x},\mathbf{y}) = e^{-\|\mathbf{x}-\mathbf{y}\|^2/2\sigma^2}$

Neural net: $K(\mathbf{x},\mathbf{y}) = \tanh(k\,\mathbf{x}.\mathbf{y} - \delta)$

4.3.3 Multi-class SVM

Although SVMs were originally designed as binary classifiers, approaches that address a multi-class problem as a single “all-together” optimization problem exist, but are computationally much more expensive than solving several binary problems. A variety of techniques for decomposition of the multi-class problem into several binary problems using Support Vector Machines as binary classifiers have been proposed. Some of them are mentioned below. [9]

1) One Vs One (OvO)

This algorithm constructs N(N-1)/2 two-class classifiers, using all the binary pair-wise combinations of the N classes. Each classifier is trained using the samples of the first class as positive examples and the samples of the second class as negative examples. To combine these classifiers, the Max Wins algorithm is adopted. It finds the resultant class by choosing the class voted by the majority of the classifiers [9].

2) One Vs All (OvA)

For the N-class problems (N>2), N two-class SVM classifiers are constructed [6]. The i th SVM is trained while labeling the samples in the i th class as positive examples and all the rest as negative examples. In the recognition phase, a test example is presented to all N SVMs and is labelled according to the maximum output among the N classifiers. The disadvantage of this method is its training complexity, as the number of training samples is large. Each of the N classifiers is trained using all available samples.

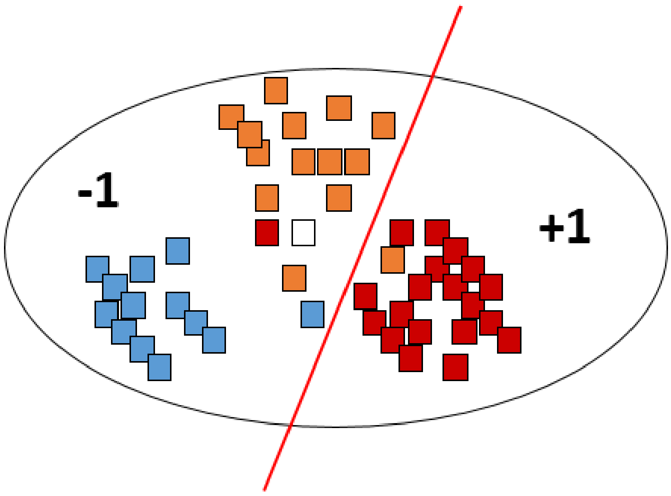


Fig. 2.10: Multi-class SVM (One *vs* All)

Here the samples taken are of various type but it is taken care that the sample of interest is considered to be '1' while rest all are termed at '-1' just to separate the data.
This is performed for all N classes.

Chapter 3: Feature Selection using Harmony Search Algorithm

Our work will also mainly focus on feature selection and experiments are based on EMODB and IITKGP-SEHSC. EMODB is a popular database, on which many works have been reported. IITKGP-SEHSC is a database consists of Hindi utterances collected from professional artists from Gyanavani FM radio station, Varanasi, India with 7 emotions. We extracted MFCC features from EMODB and IITKGP-SEHSC, respectively.

Harmony search (HS) algorithm is a heuristic algorithm which simulates the generation process of harmony, it has been used in gene selection and achieved good results because of its global search ability works on selecting features. Due to simplicity, flexibility, generality, and lower parameter sensitivity of HS, we consider using HS algorithm to select subsets and reduce the dimensions. In summary the proposed method has the following advantages

I. HS algorithm imposes fewer mathematical requirements and does not require initial value settings of the decision variables.

II. As the HS algorithm uses stochastic random searches, derivative information is also needless.

III. HS algorithm generates a new vector, after considering all of the existing vectors, while the genetic algorithm (GA) only considers the two parent vectors. These features increase the flexibility of the HS algorithm and produce better solutions.

IV. HS is good at identifying the high performance regions of the solution space at a reasonable time.

3.1 Harmony Search Algorithm as Feature Selection Technique

3.1.1 Principles of HS

The original HS algorithm is designed to solve numerical optimization problems, and most of its early applications [50] involve discrete-valued variables. When applied to such problems, musicians typically represent the decision variables of a given cost function, and HS acts as a meta-heuristic algorithm that attempts to find a solution vector that optimizes this function. In such a search process, each decision variable (musician) generates a value (musical note) for finding a global optimum (best harmony). The aim here is to provide a thorough explanation of the algorithm, including its key notions and iteration steps, so that the proposed HSFS algorithm may be better introduced thereafter [53].

3.1.2 Key Notions

The key notions of HS, as illustrated in Fig. 3.1, are musicians, notes, harmonies, fitness, and harmony memory. In most optimization problems solvable using HS, the musicians $P = \{P^i | i = 1, \dots, |P|\}$ represent the variables of the cost function being optimized, and the values of the variables are referred to as musical notes. A harmony $H,\ |H| = |P|$ is a candidate solution vector containing the values for each variable, where a collection of good quality solutions are stored in the harmony memory $\mathbb{H} = \{H^j | j = 1, \dots, |\mathbb{H}|\}$. Note that all of the above mentioned collections: $P,\ \mathbb{H},\ H$ are fixed-sized, ordered lists, rather than sets. In particular, $H_i^j, i = 1, \dots, |P|, j = 1, \dots, |\mathbb{H}|$, denotes the value selected by the i^{th} musician in the j^{th} harmony stored within the harmony memory.

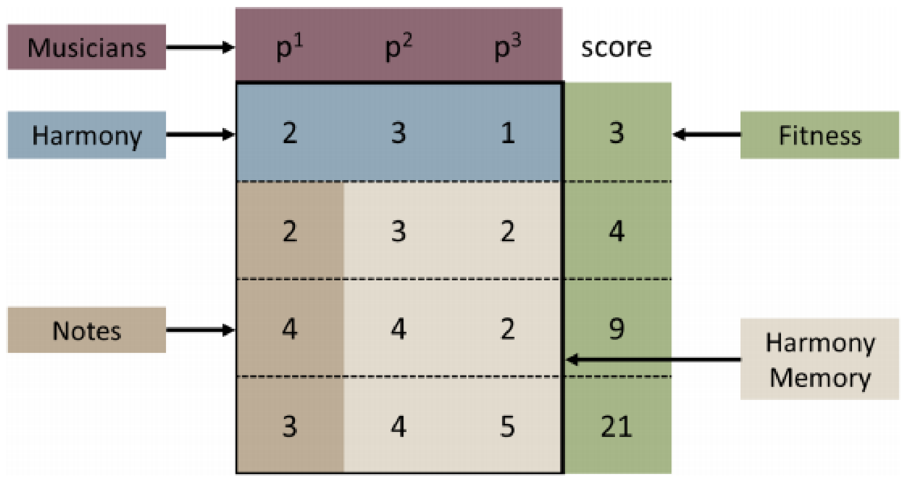


Fig 3.1: Key notions of HS

For a newly constructed (empty) harmony, all of the internal values are initialized as $-$, indicating that no musical notes have been assigned. A harmony memory $\mathbb{H}$ can be concretely represented as a two dimensional matrix. Without losing generality, the number of rows (harmonies) $|\mathbb{H}|$ is a predefined parameter that limits the maximum number of harmonies to be stored. Each column of the matrix is dedicated to one musician, which provides a pool of playable notes for future improvisations. In this thesis, such a pool is referred to as the note domain $\aleph^{th}$ of a musician p^i :

(3.1)

$$\aleph^i = \bigcup_{j=1}^{|\mathbb{H}|} H_i^j, H^j \in \mathbb{H}, i = 1, \dots, |P|$$

3.1.3 Iterative Process of HS

HS can be divided into two core phases: initialization and iteration, as shown in Fig. 3.2. A simple discrete numerical problem [51] given in Eqn. 3.2 is used here to illustrate the process of HS. Minimize

$$(a-2)2 + (b-3)4 + (c-1)2 = 3 \tag{3.2}$$

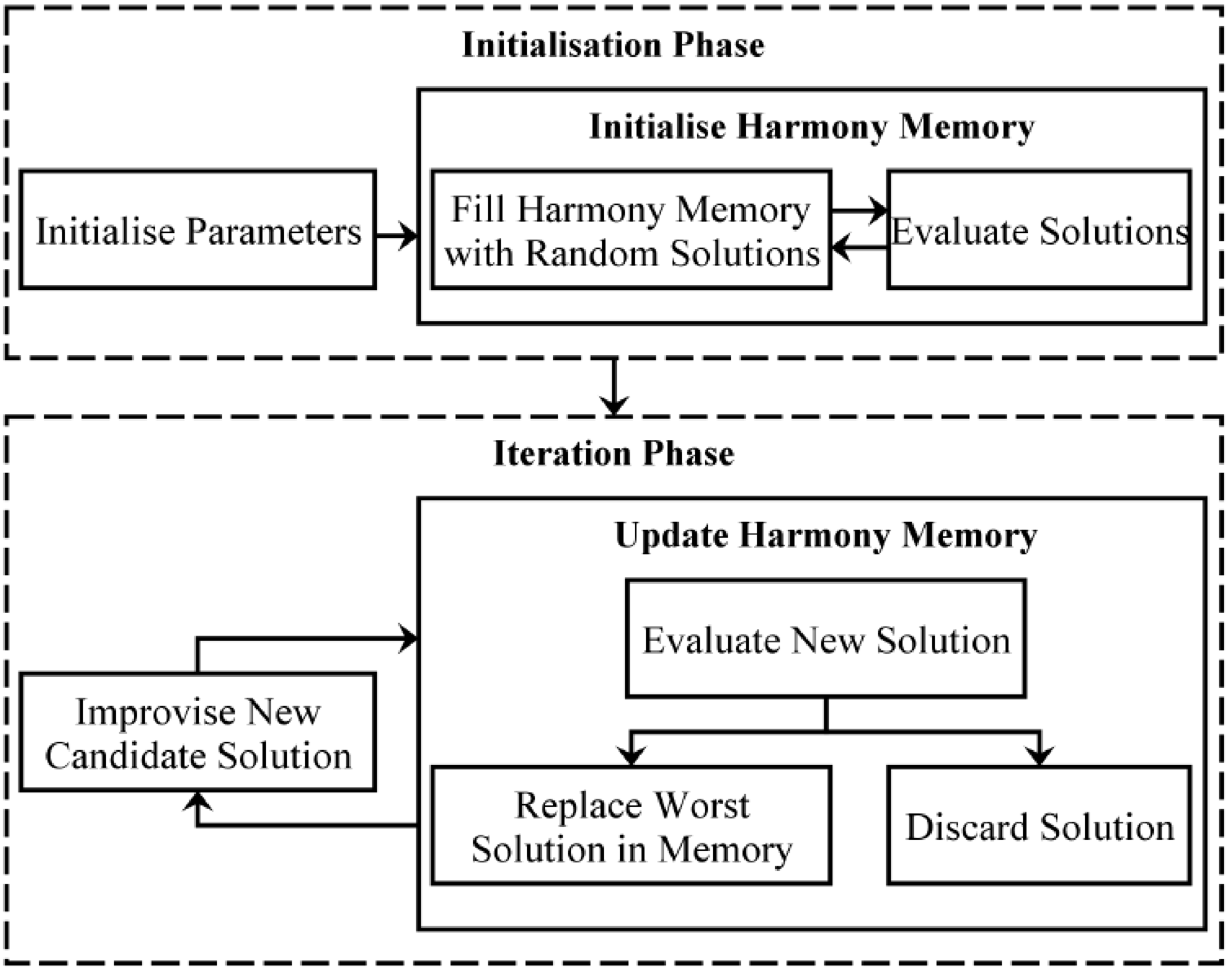


Fig 3.2: Iteration Steps of HS [53]

Where $a, b, c \in \{1,2,3,4,5\}$.

1. Initialise Problem Domain

In the beginning, the parameters used in the search need to be established. This includes $|\mathbb{H}|$, δ, g_{max}, ρ, τ, and $|P|$.

According to the problem at hand, the group of musicians $\{p^1, p^2, p^3\}$ is initialised with a size equal to the number of variables ($|P| = 3$), each corresponds to the function variables a, b and c. The harmony memory $\mathbb{H}$ is filled with randomly generated solution vectors. In the example problem, 3 randomly generated solution vectors may be $\{2,2,1\}$, $\{1,3,4\}$ and $\{5,3,3\}$.

2. Improve New Harmony

A new value is chosen randomly by each musician out of their note domain and together form a new harmony. During the improvisation process, the stochastic events controlled by δ and ρ will also occur, causing the value of the selected notes to change.

In the example, musician p^1 may randomly choose 1 out of $\aleph^1 = \{2,1,5\}$, p^2 chooses 2 out of $\aleph^2 = \{2,3,3\}$and p^3chooses 3 out of $\aleph^3 = \{5,3,3\}$, forming a new harmony $\{1,2,3\}$. Given the above example, with $\delta = 0.9$, and $\rho = 0.1$, musician p^1 will choose from within his note domain $\aleph^3 = \{2,1,5\}$with a probability of 0.9. After making a choice, say, 5, the musician will choose the left or right neighbours with 0.05 probability for each, and the left neighbouring value 4, may then be chosen in the end. Alternatively, the musician may choose from the range of all possible values, i.e., $\{1,2,3,4,5\}$ with a probability of 0.1, and the note 4 may again be chosen but without further pitch adjustment.

To further ease the understanding of HS, Algorithm 3.1.1 presents an outline of the improvisation procedure in pseudo code.

$p^i \in P, i = 1, \ldots, |P|$, group of musicians
$H^j \in \mathbb{H}, j = 1, \ldots, |\mathbb{H}|$, harmony memory
H_i^j, the value of the i^{th} variable in H^j
H_{new}, emerging harmony
$\aleph^i = \bigcup_{j=1}^{|\mathbb{H}|} H_i^j$, note domain of p^i
δ, harmony considering rate
ρ, pitch adjustment rate
τ, fret-width
min_i , $\max_i$, the value range of the i^{th} variable
for $i = 1$ to $|P|$ do
    random $r_\delta, 0 \le r_\delta \le 1$
    if $r_\delta < \delta$ then
        random $r_i, r_i \in \aleph_i$
        random r_ρ, $0 \le r_\rho \le 1$
        if $r_\rho < \rho$ then
            random $r_\tau, -1 \le r_\tau \le 1$
            $r_i = r_i + r_\tau \tau$
    else

random r_i, $\min_i \leq r \leq \max_i$
$H_i^{new} = r_i$
return H^{new}

Algorithm 3.1.1: Improvisation process of original HS [53]

3. Update Harmony Memory

If the new-harmony is better than the worst harmony in the harmony memory (judged by the objective function), the new-harmony is then included in the resulting harmony memory and the existing worst harmony is removed.

For example, assume the newly improvised harmony {1,2,3} has an evaluation score of 9, making it better than the worst harmony in the harmony memory {1,3,4} which has a score of 16. Therefore the harmony {1,3,4} is removed from harmony memory, and replaced by {1,2,3}. If {1,2,3} had a greater score than 16, then it would be the one being discarded.

4. Iteration

The algorithm continues to iterate until the maximum number of iterations g_{max} is reached. In the end, the highest quality solution present in the harmony memory is returned as the final output.

In the example, if the musicians later improvise a new harmony with values {2,3,1}, which is very likely as these numbers are already in their respective note domains, the problem will be solved (with a minimal fitness score of 3).

3.2 Initial Development

This section describes the preliminary investigations [52] carried out, which explores the feasibility of applying HS to the problem domain of FS. This initial HS-based FS approach, being a stand-alone and functional application of HS, helped substantially in obtaining a better understanding of the internal mechanisms of HS, and its application to the problem domain of FS. It also revealed a number of drawbacks that inspired further development, from which the current HSFS algorithm is derived.

3.3 Binary-Valued Representation

A binary-valued feature subset representation has been adopted in the initial approach, which is also the most commonly used representation in the literature. Recall from Section 3.1.1 that the key notions of HS are musicians, notes, harmonies and harmony memory. The binary-valued approach maps musicians directly onto the available features to be selected, i.e. $|P| = |A|$. The note domain $\aleph_i$ of a given musician p^i, contains only binary values: $\aleph_i = \{0,1\}$, which indicates whether corresponding feature is included (1), or not (0) in the emerging feature subset.

A harmony is represented as a series of bits that encodes the selected features. For example, as shown in below table, for a given data set with 6 features

$A = \{a_1, a_2, a_3, a_4, a_5, a_6\}$, harmony $H^1 = \{0, 1, 1, 0, 0, 0\}$translates into feature subset $B_{H^1} = \{a_2, a_3\}$. The binary encoding of feature subsets is a straightforward mapping. It allows the procedures of HS: initialisation and iteration, as illustrated in Fig3.2, to be executed in the same fashion as that of standard numerical optimisation tasks.

	p^1	p^2	p^3	p^4	p^5	p^6	Represented subset B
H^1	0	1	1	0	0	0	$\{a_2, a_3\}$
H^2	1	0	0	0	$0 \rightarrow 1$	1	$\{a_1, a_5, a_6\}$

3.3.1 Iteration Steps

The initialisation step involves filling the harmony memory with randomly generated feature subsets, i.e. random-valued string of bits. In order to improvise a new harmony, each musician randomly selects a value from their respective note domain. Together, such selected values form a new bit set. This set is then translated back into a feature subset and evaluated. If the evaluation score is higher than any of the feature subsets in the harmony memory, it replaces the worst candidate feature subset; otherwise, the new bit set is discarded. The process repeats until the maximum number of iterations g_{max} has been reached [53].

In this approach, the harmony memory considering rate δ has little practical impact because the amount of available notes (0 and 1) for each musician are very limited. The most significant use of it is in terms of flipping the bit value, which includes a previously unselected feature, or vice versa. Hence, in this initial development, the parameter δ is simply implemented as the bit flipping rate, and its effect is demonstrated by the second harmony H^2 in Table 3.1. $0 \rightarrow 1$ signifies a forced value change due to δ activation, which causes the affected musician p^5 to change its decision to the opposite value [53].

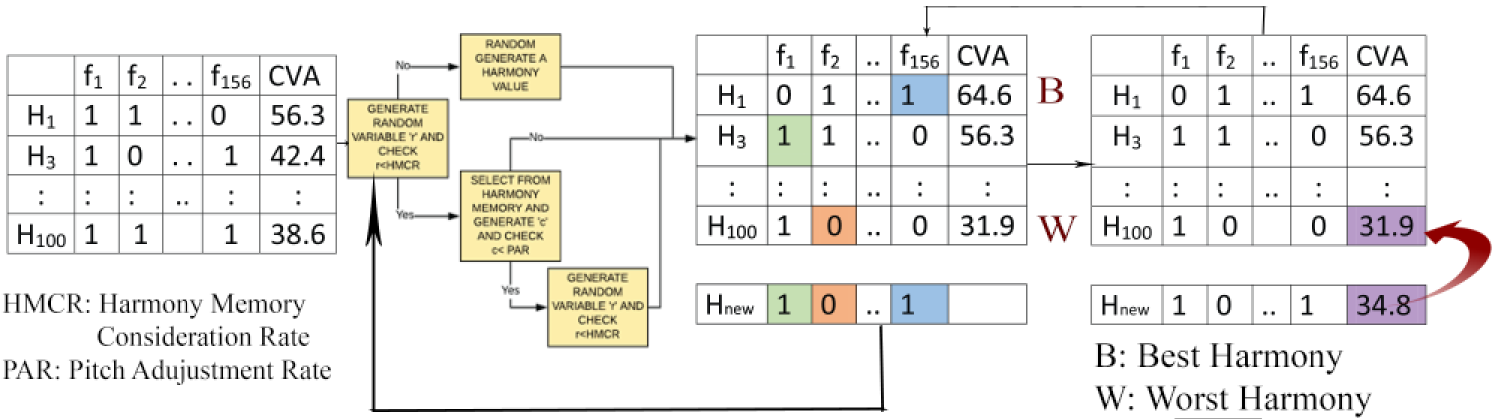


Fig 3.3: Process of feature selection using harmony search in binary representation

Chapter 4: Results and Discussion

This chapter is dedicated to a discussion of simulations performed and the results thereof. We have used MATLAB for the simulations.

4.1 Evaluation Measures

A confusion matrix is a table that is often used to describe the performance of a classification model on a set of test data for which the true values are known. All the measures except AUC can be calculated by using left most four parameters. So, let's talk about those three parameters first.

	Predicted class		
Actual Class		Class = Yes	Class = No
	Class = Yes	True Positive	False Negative
	Class = No	False Positive	True Negative

Table II: Class Classification Parameters

True positive and true negatives are the observations that are correctly predicted and therefore shown in green. We want to minimize false positives and false negatives so they are shown in red colour. These terms are a bit confusing. So let's take each term one by one and understand it fully.

True Positives (TP)

These are the correctly predicted positive values which means that the value of actual class is yes and the value of predicted class is also yes. E.g. if actual class value indicates that this passenger survived and predicted class tells you the same thing.

True Negatives (TN)

These are the correctly predicted negative values which means that the value of actual class is no and value of predicted class is also no. E.g. if actual class says this passenger did not survive and predicted class tells you the same thing.

False positives and false negatives, these values occur when your actual class contradicts with the predicted class.

False Positives (FP)

When actual class is no and predicted class is yes. E.g. if actual class says this passenger did not survive but predicted class tells you that this passenger will survive.

False Negatives (FN)

When actual class is yes but predicted class in no. E.g. if actual class value indicates that this passenger survived and predicted class tells you that passenger will die.

Accuracy

Accuracy is the most intuitive performance measure and it is simply a ratio of correctly predicted observation to the total observations. One may think that, if we have high accuracy then our model is best. Yes, accuracy is a great measure but only when you have symmetric datasets where values of false positive and false negatives are almost same. Therefore, you have to look at other parameters to evaluate the performance of your model. For our model, we have got 0.803 which means our model is approx. 80% accurate.

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN}$$

Precision

Precision is the ratio of correctly predicted positive observations to the total predicted positive observations. The question that this metric answer is of all passengers that labeled as survived, how many actually survived? High precision relates to the low false positive rate. We have got 0.788 precision which is pretty good.

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

Recall (Sensitivity)

Recall is the ratio of correctly predicted positive observations to the all observations in actual class - yes. The question recall answers is: Of all the passengers that truly survived, how many did we label? We have got recall of 0.631 which is good for this model as it’s above 0.5.

$$Recall = \frac{TP}{TP + FN}$$

4.2 Databases

(a) Emo-DB

A popular example is the Berlin database of emotional speech (Emo-DB) in which recordings were taken in an anechoic chamber of the Technical University, using 5 Male and 5 Female actors at a sampled frequency of 16 MHz. The number of samples recorded for 6 emotions and neutral for an average duration of 2.78 seconds is as shown below

Emotion	Anger	Boredom	Disgust	Fear	Happy	Sad	Neutral	Total
Number of samples	127	81	46	69	71	62	79	535

Table III: Distribution of Emotional speeches in EMODB Database

(b) IITKGP-SEHSC

This is an Indian Institute of Technology Kharagpur Simulated Emotion Hindi Speech Corpus (IITKGP-SEHSC) recorded using professional artists from Gyanavani FM radio station, Varanasi, India. The speech corpus is collected by simulating eight different emotions using neutral (emotion free) text prompts. The emotions present in the database are anger, disgust, fear, happy, neutral, sad, and sarcastic and surprise. We have considered all the emotions excluding sarcastic one for our project A Distribution of emotion in corpus is give below:

Emotion	Anger	Surprise	Disgust	Fear	Happy	Sad	Neutral	Total
Number of samples	1396	1441	1440	1437	1442	1442	1426	10024

Table IV: Distribution of Emotional speeches in IITKGP-SEHSC Database

4.3 Experimental Settings

Feature Extraction

The feature set used in this project is Mel Frequency Cepstral Coefficient (MFCC), which is recognized as an effective feature in [1]. We extracted 156 MFCC coefficients and compute their mean, variance, kurtosis and standard deviation as MFCC features. These MFCC features are extracted for both Emo-DB an IITKGP-SEHSC.

MFCC features	13
MFCC+ΔMFCC+ΔΔMFCC	13 x 3 = 39
Mean, Standard deviation, Variance, Kurtosis	39 x 4 = 156

Table V: Feature Extraction

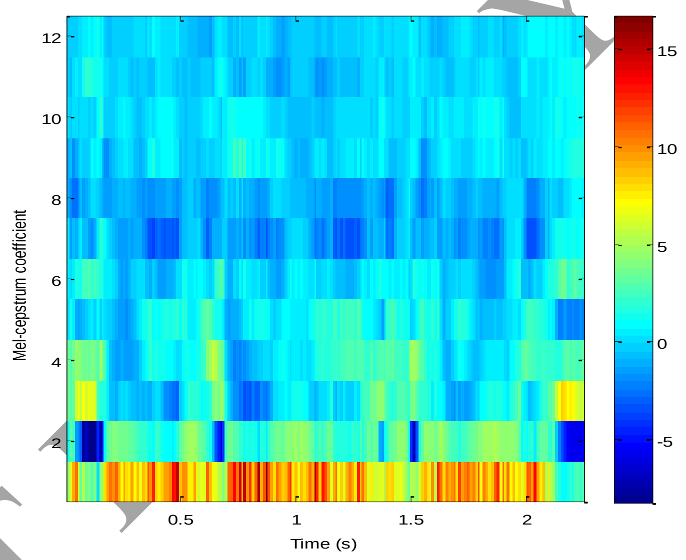


Fig. 4.1: Spectrogram of extracted MFCCs

Feature Selection

We have set HM size as 50, i.e. there will be 50 rows in harmony memory. In our project, we have used binary representation for harmony search based feature section in which 1 denotes that particular feature is selected, 0 denotes no selection. Then, HMCR is 0.8, PAR is 0.4, and stopping criteria is 50 improvisations. 10 fold cross validation is used for evaluation purpose. Then looping HS algorithm to select subset from MFCC feature set both on Emo-DB and IITKGP-SEHSC. Next, we compared the accuracy between unselected and selected feature sets. We have set box constraint for linear SVM as 0.08 as specified in.

4.4 Experimental Results

Confusion Matrices

- Classifier Used: SVM (One vs. All)
- Kernel Used: Linear
- Total features: 156

	Anger	Bore	Disgust	Fear	Happy	Neutral	Sad
Anger	**55.81**	0	4.65	16.28	16.28	6.98	0
Bore	0	**77.42**	9.68	0	3.23	9.68	0
Disgust	0	58.33	**12.5**	0	0	12.5	0
Fear	7.41	3.7	0	**40.74**	22.22	25.93	0
Happy	56	0	4	8	**24**	8	0
Neutral	0	30.43	0	0	4.35	**65.22**	0
Sad	0	73.91	21.74	0	0	4.35	**0**

Table VI: Confusion Matrix for EMODB Database

This takes high computation time. Also as seen, samples of 'Sad' emotion are misclassified mainly as 'Bore' and 'Disgust'. Even 'Happy' samples are misclassified as 'Anger'. This is because excitation levels are very close of these emotions in which confusion occurs.

	Happy	Angry	Neutral	Sad	Surprise	Fear	Disgust
Happy	**62.58**	6.80	5.89	5.66	9.52	1.81	7.70
Angry	6.85	**72.58**	3.29	0	8.88	2.03	6.34
Neutral	5.16	0.93	**78.40**	8.21	0.70	3.05	3.52
Sad	4.30	0.22	11.33	**64.85**	2.04	16.32	0.90
Surprise	6.12	12.47	2.49	2.94	**67.12**	5.89	2.94
Fear	0.68	2.51	4.57	14.41	5.94	**67.73**	4.12
Disgust	9.54	7.04	5.68	2.04	5.45	2.04	**68.18**

Table VII: Confusion Matrix for IITKGP-SEHSC Database

The computation was too high. Mainly confusion was observed in 1) 'Angry', 'Happy' and 'Surprise' 2) 'Fear', 'Sad' and 'Neutral'. The levels of excitation being same, misclassification occurs.

- Selected features: 79

	Anger	**Boredom**	**Disgust**	**Fear**	**Happy**	**Neutral**	**Sad**
Anger	**76.74**	0	2.32	20.9	0	0	0
Boredom	3.22	**70.96**	19.35	0	0	6.45	0
Disgust	4.1	25	**66.67**	0	0	4.1	0
Fear	29.63	0	7	**55.6**	3.7	3.7	0
Happy	80	0	8	8	**0**	4	0
Neutral	0	34.78	13.04	0	4.3	**45.8**	0
Sad	0	30.43	39.13	0	0	26.08	**4.34**

Table VIII: Confusion Matrix for EMODB Database

The feature set was reduced to 50.64% than original with a 2.36% increment in 10-fold cross-validation accuracy and a 6.35% improvement in test accuracy of EMODB database. 'Happy' was misclassified as 'Anger' while 'Sad' was misclassified as 'Boredom', 'Disgust' and 'neutral', due to confusion in classification.

- Selected Features: 87

	Happy	**Angry**	**Neutral**	**Sad**	**Surprise**	**Fear**	**Disgust**
Happy	**49.65**	9.97	10.65	6.12	12.92	1.81	8.84
Angry	7.86	**63.95**	4.31	0.76	13.19	3.04	6.85
Neutral	8.92	1.40	**66.90**	11.73	1.17	4.92	4.92
Sad	4.98	0.22	15.41	**55.78**	3.85	17.68	2.04
Surprise	7.48	16.78	4.08	3.17	**55.78**	7.25	5.44
Fear	1.14	2.28	4.34	15.78	6.86	**64.53**	5.03
Disgust	12.5	8.40	7.5	1.81	9.54	3.40	**56.81**

Table IX: Confusion Matrix for IITKGP-SEHSC Database after feature selection

The feature set was reduced to 55.77% than original with a 3.5% decrement in cross-validation accuracy and a reduction of 5.1% in test accuracy of IITKGP_SEHSC database. Misclassification occurred in emotions showing same excitation levels.

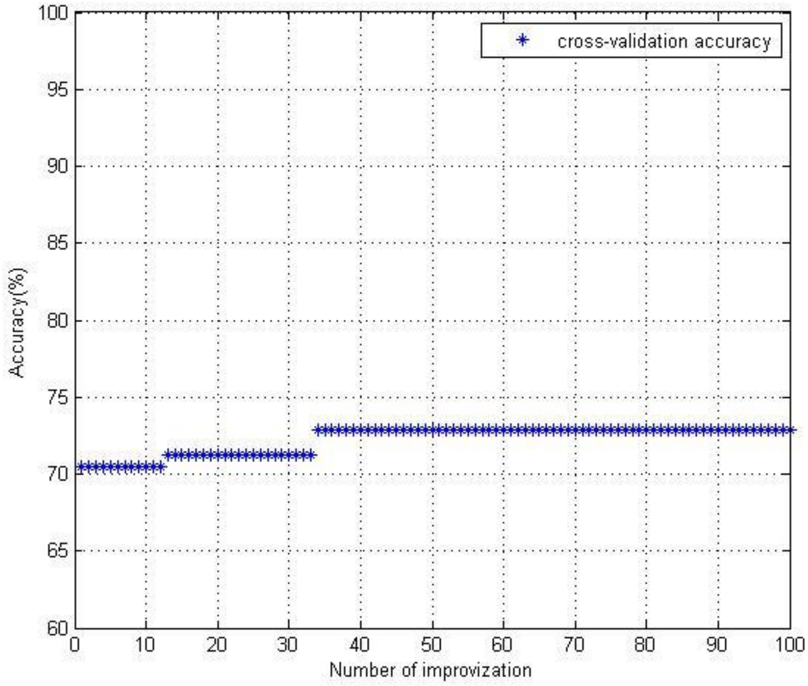


Fig 4.2: Convergence of Accuracy while processing EMODB database

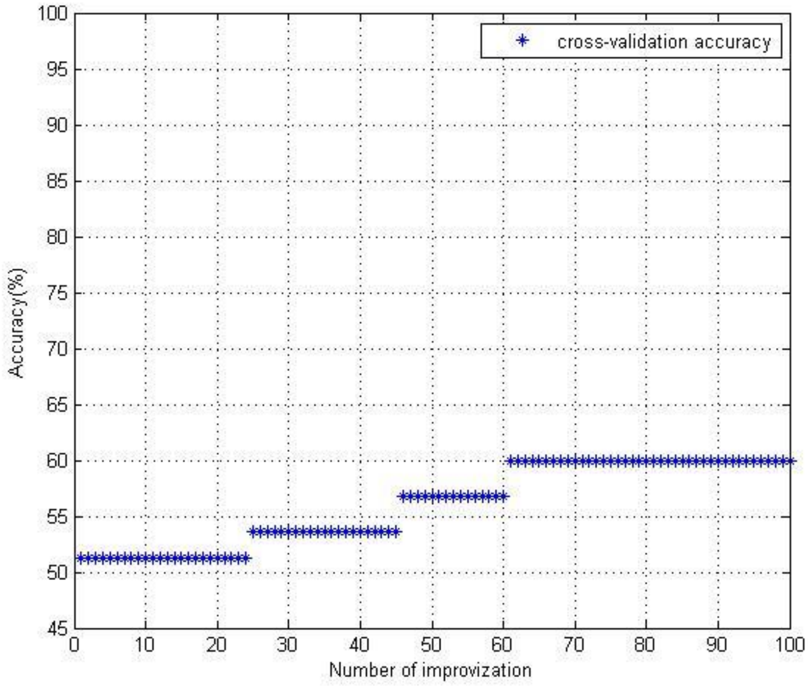


Fig 4.3: Convergence of Accuracy while processing IITKGP-SEHSC database

From both images it is clear that accuracy is not always maximum. Over iterations, by method of Harmony Search, different combination of features result in generation of a combination that gives our best accuracy out of set, which is maintained unless another best one is obtained. Also each iteration need not produce best combination, hence it is observed that accuracy increases but after a particular point, may be as only 200 combinations ie. 100 random combination of memory + 100 improved combination . However the time consumed in this selection process cosumes lot of time dpeneding on number of iterations and processing capacity of device used. This consumes time only while selection process and creating model while test result is obtained almost in 3 to 4 seconds.

4.5 Comparision of Selected and Unselected features:

1) EMODB database

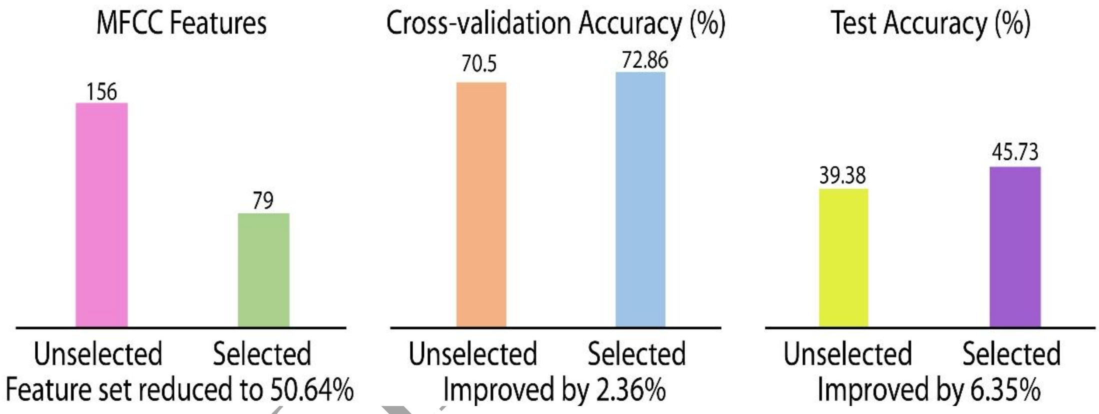


Fig 4.4: comparison of unselected and selected feature size on EMODB

2) IITKGP-SEHSC database

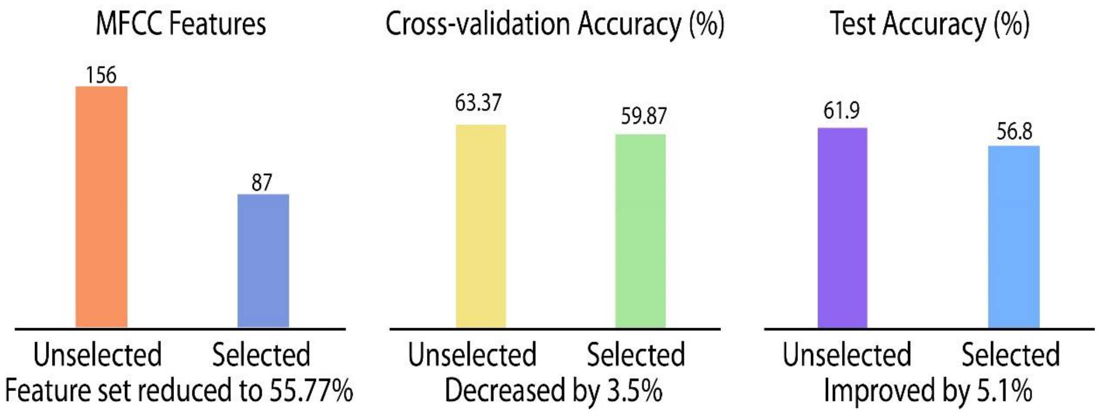


Fig 4.5: comparison of unselected and selected feature size on IITKGP-SEHSC

4.6 Application User Interface

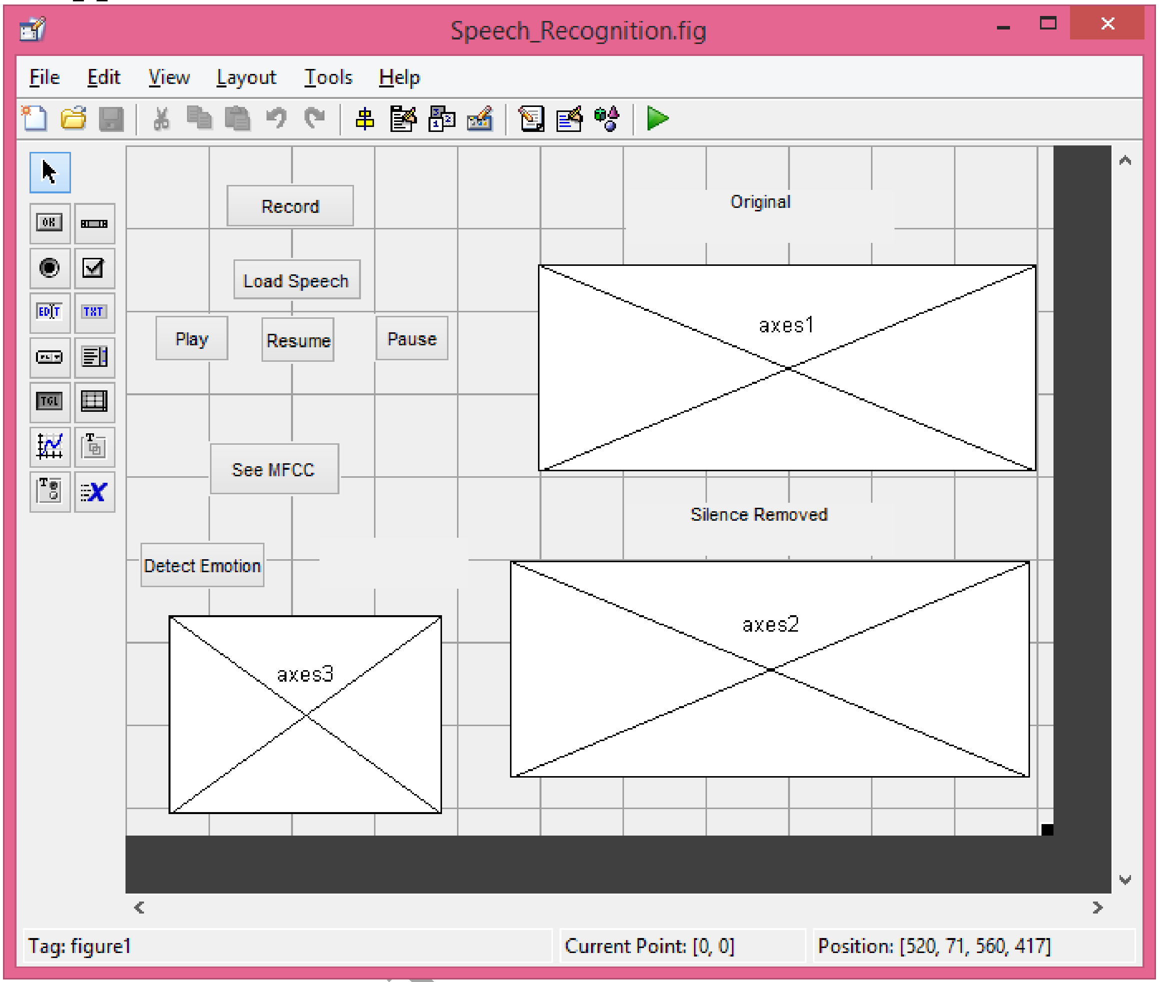


Fig. 4.6: GUI guide

GUI layout is shown below. It consists of the following:

1. Record: It is a push button which records the audio from mic for real time emotion recognition
2. Load Speech: It is a push button which is used to select a audio from computer
3. See MFCC: For seeing MFCC spectrogram of that particular audio
4. Play: It is a push button which can be used to play a particular audio
5. Pause: It is a push button which can be used to pause a audio which is currently being played.
6. Resume: It is a push button which is used to resume a audio which is paused.
7. Detect Emotion: In case, It will detect emotion of that particular audio, this push button is pressed.

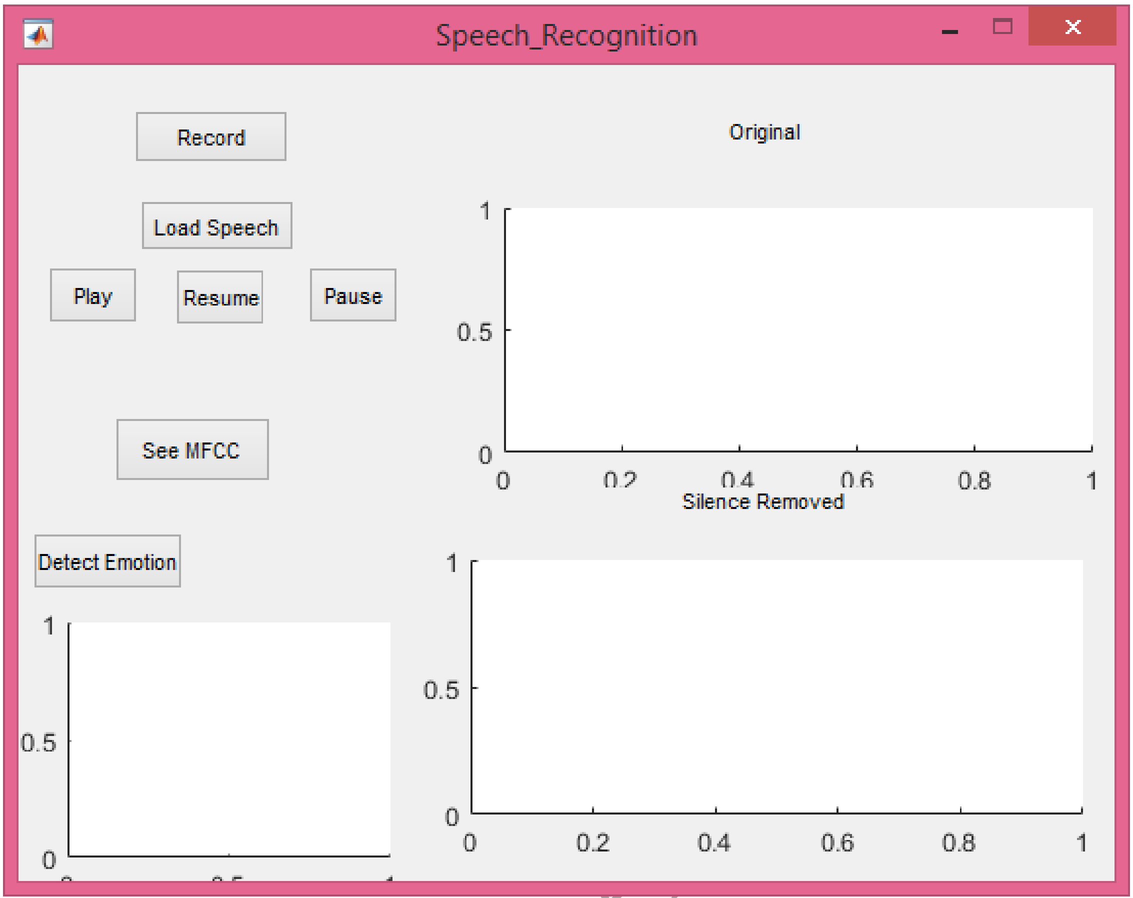


Fig. 4.7: Speech Emotion Detector (User interface)

On pressing the 'Select Speech' button, window will come to select audio from computer

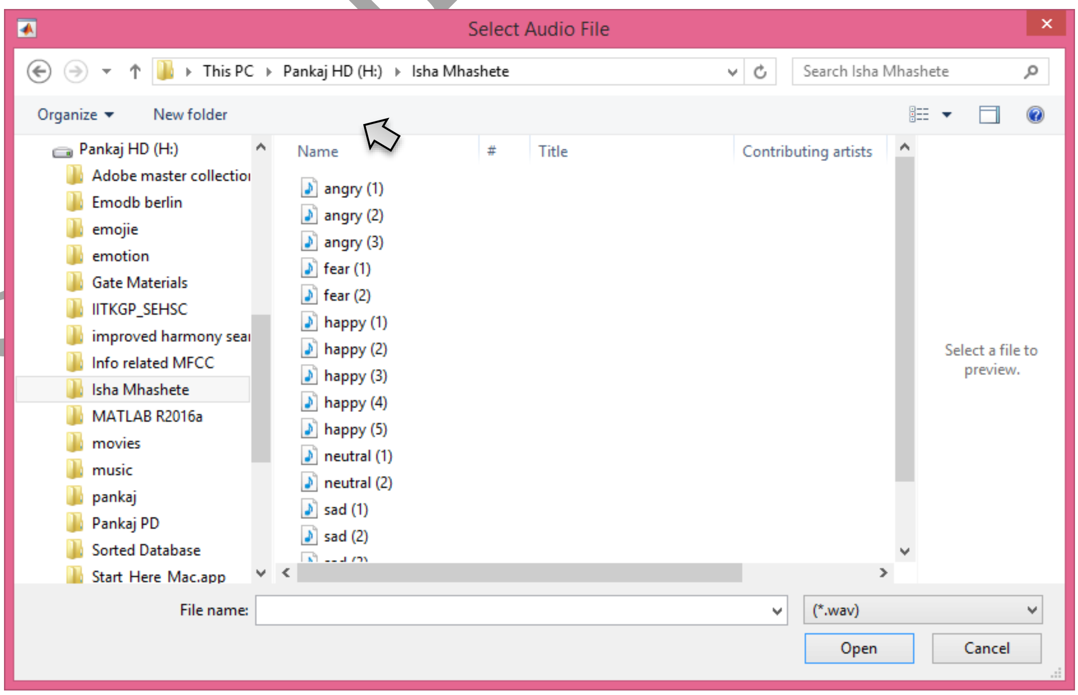


Fig. 4.8: Selection of emotional speech for processing

By selecting audio from the pc, original audio and its silence removed version I is displayed on axes. We can play that particular audio by pushing the ‘Play’ button as well.

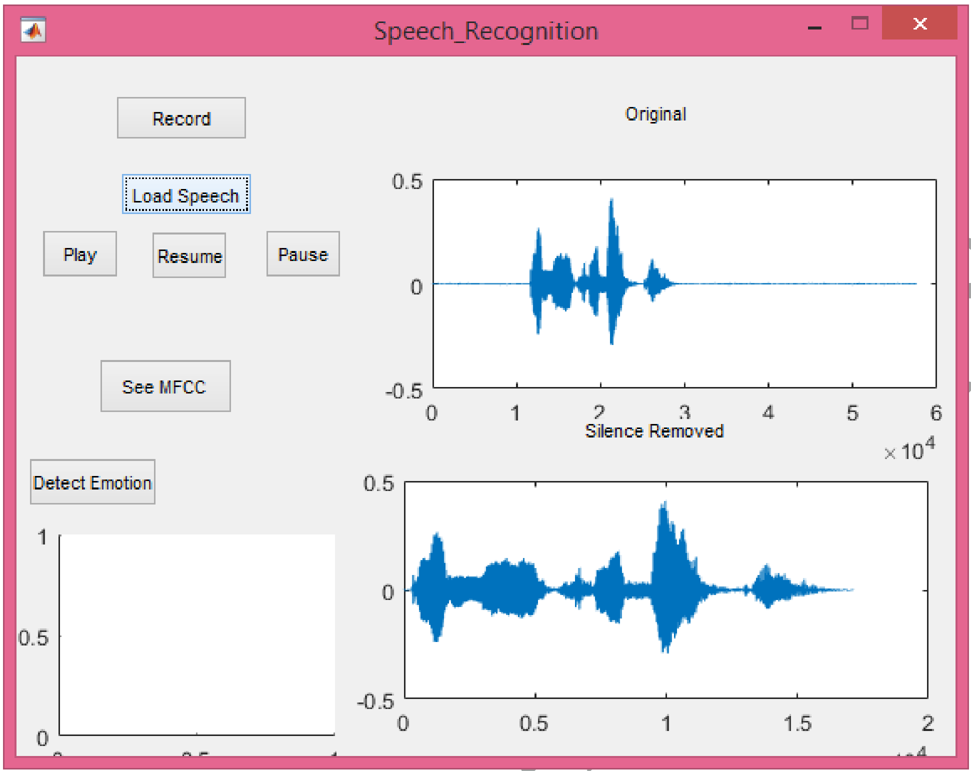


Fig. 4.9: Playing selected speech

To see MFCC Spectrogram for the selected audio, push the ‘See MFCC’ button.

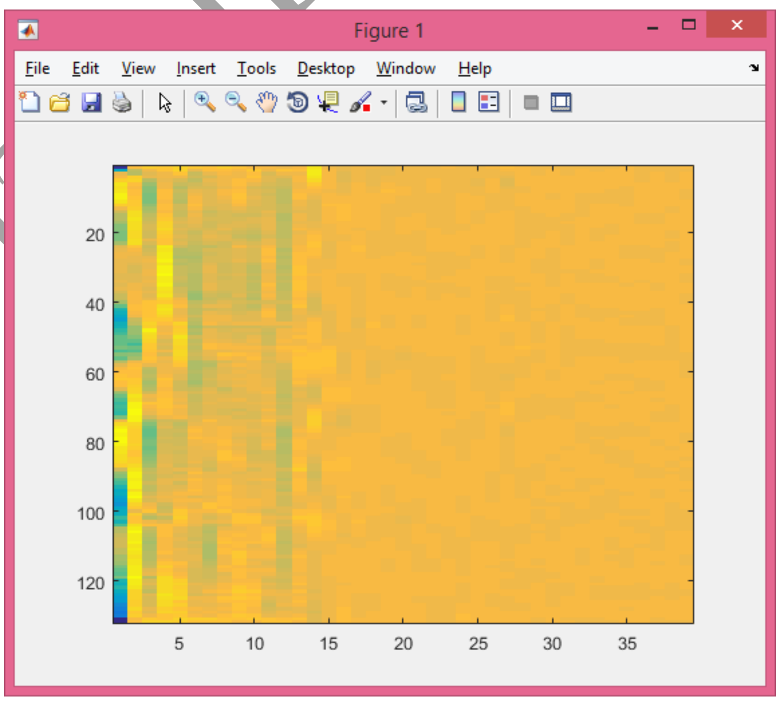


Fig. 4.10: Displaying Spectrogram of MFCC

Finally the GUI displays emotion of the selected track when you push ‘Detect Emotion’ button.

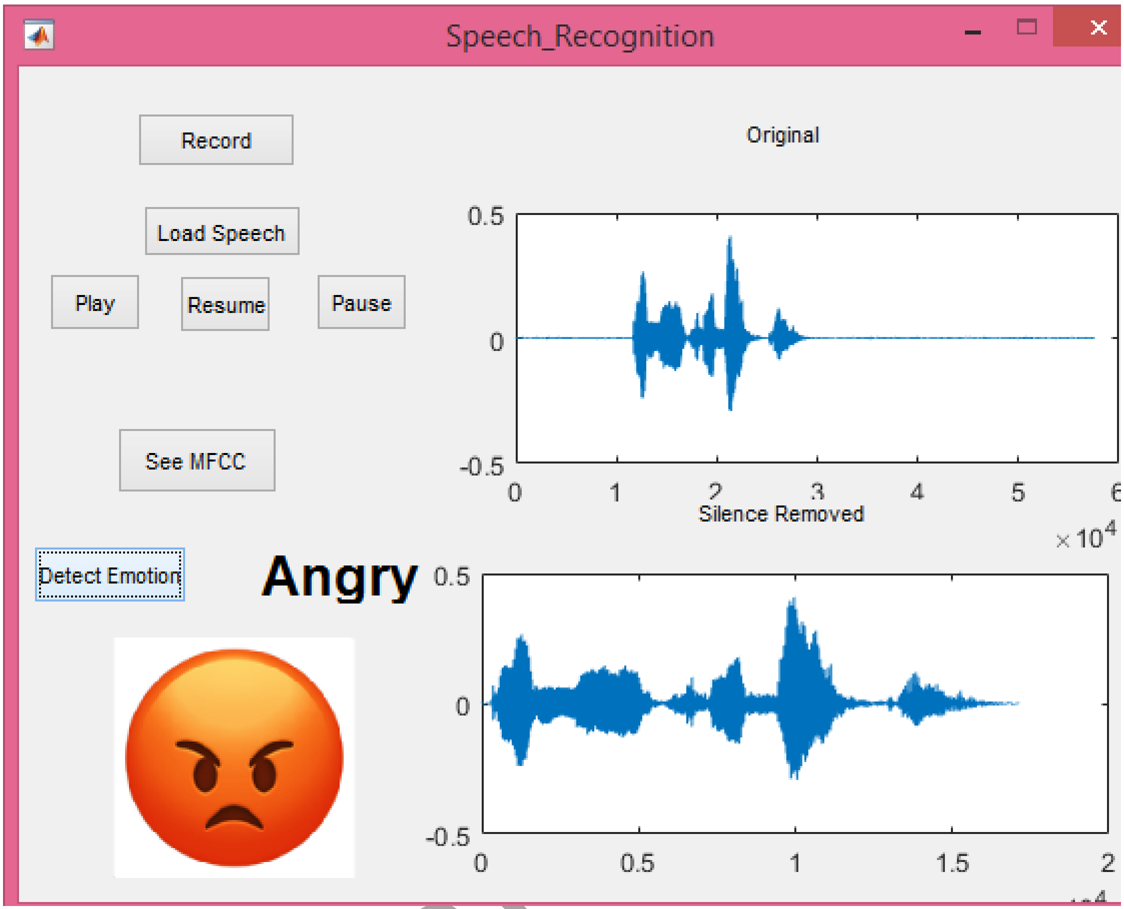


Fig. 4.11: Detection of emotional state

Chapter 5: Conclusion and Future Work

5.1 Conclusion

This project focused on selecting subset from dataset. We extracted MFCC feature set on corpus EMODB and IITKGP-SEHSC. Then, selecting features with HS algorithm and verified the effectiveness of HS with 10-fold cross validation on linear SVM.

The feature set was reduced to 50.64% than original with a 2.36% increment in 10-fold cross-validation accuracy and a 6.35% improvement in test accuracy of EMODB database.

The feature set was reduced to 55.77% than original with a 3.5% decrement in cross-validation accuracy and a reduction of 5.1% in test accuracy of IITKGP_SEHSC database.

5.2 Future Work

As we can see from the results, Happiness was classified as anger and sadness as neutral boredom and disgust. Also neutral to some extent was classified as boredom. So there is need of better speech features is realized which contains emotional information.

So in future we will we work other features along with MFCC features. Also reoccurring HS means selected features may can be fed again to HS for more reduced feature set. Now the extraction of other features like ‘Fourier Parameters’ has to be done. Also the computation time is observed to be large due to calculation and consideration of all features increases error probability, to minimize the number of features considered, Harmony Search Method needs to be implemented.

Although simple in concept, the use of binary-valued note domain limits the efficiency and explorative potential of HS. To better address these problems, an integer-valued HSFS algorithm has to be deployed on speech emotion recognition, providing more freedom for the choice of playable notes, and allowing the stochastic mechanisms of HS to be exploited more thoroughly.

APPENDIX A

MATLAB

Matlab is a cross-platform numerical computational package and a high-level, numerically oriented programming language. It can be used for machine learning, signal processing, image processing, computer vision, communications, computational finance, control design, robotics, and much more. The MATLAB platform is optimized for solving engineering and scientific problems. The matrix-based MATLAB language is the world's most natural way to express computational mathematics. Built-in graphics make it easy to visualize and gain insights from data. A vast library of pre-built toolboxes lets you get started right away with algorithms essential to your domain. The desktop environment invites experimentation, exploration, and discovery. These MATLAB tools and capabilities are all rigorously tested and designed to work together.

Voicebox Toolbox

VOICEBOX is a speech processing toolbox consists of MATLAB routines. The routine VOICEBOX.M contains various installation-dependent parameters which may need to be altered before using the toolbox. In particular it contains a number of default directory paths indicating where temporary files should be created, where speech data normally resides, etc. One can override these defaults by editing VOICEBOX.M directly or, more conveniently, by setting an environment variable VOICEBOX to the path of an initializing m-file.

OpenSMILE

The openSMILE feature extraction tool enables you to extract large audio feature spaces in real-time. It combines features from Music Information Retrieval and Speech Processing. SMILE is an acronym for Speech & Music Interpretation by Large-space Extraction. It is written in C++ and is available as both a standalone command-line executable as well as a dynamic library. The main features of openSMILE are its capability of on-line incremental processing and its modularity. Feature extractor components can be freely interconnected to create new and custom features, all via a simple configuration file. New components can be added to openSMILE

Application User Interface

The graphical user interface is a type of user interface that allows users to interact with electronic devices through graphical icons and visual indicators such as secondary notation, instead of text-based user interfaces, typed command labels or text navigation.

To create a MATLAB GUI interactively we used GUIDE function. GUIDE (GUI development environment) provides tools to design user interfaces for custom apps. Using the GUIDE Layout Editor, you can graphically design your UI. GUIDE then automatically generates the MATLAB code for constructing the UI, which you can modify to program the behavior of your app. We can add dialog boxes, user interface controls (such as push buttons and sliders) and containers (such as panels and button groups).

We have developed a Graphical User Interface using MATLAB for demonstrating the working of our project. GUIs provide point-and-click control of software applications, eliminating the need to learn a language or type commands in order to run applications. MATLAB apps are self-contained MATLAB programs with GUI front ends that automate a task or calculation. The GUI typically contains controls such as menus, toolbars, buttons, and sliders.

Basic components of GUI include:

- Icons: Small pictures that represent commands, files, or windows. By moving the pointer to the icon and pressing a mouse button, you can execute a command or convert the icon into a window.
- Push Button: It has a textual label and is designed to invoke an action when pushed.
- List Box: It is a component that defines a scrollable list of text items.
- Text Field: It is a component that implements a single line of text.
- Panels: It is a container for grouping together *UI* components.
- Figure Window: It is a container for graphics or user interface components

EMODB Database

A popular example is the Berlin database of emotional speech (Emo-DB) in which recordings were taken in an anechoic chamber of the Technical University, using 5 Male and 5 Female actors at a sampled frequency of 16 MHz. The number of samples recorded for 6 emotions and neutral for an average duration of 2.78 seconds is as shown below

Emotion	Anger	Boredom	Disgust	Fear	Happy	Sad	Neutral	Total
Number of samples	127	81	46	69	71	62	79	535

Table X: Number of samples per emotion in Emo-DB

The actors on being directed to speak in the direction of the microphone kept at a distance of 30cm away, were asked to remember the real situation from their past when it came to emoting the required emotion. The sentences that were used for recording the database cam be seen in table below

code	Text (German)	English Translation
a01	Der Lappen liegt auf dem Eisschrank.	The tablecloth is lying on the fridge.
a02	Das will sie am Mittwoch abgeben.	She will hand it in on Wednesday.
a04	Heute abend könnte ich es ihm sagen.	Tonight I could tell him.
a05	Das schwarze Stück Papier befindet sich da oben neben dem Holzstück.	The black sheet of paper is located up there besides the piece of timber.
a07	In sieben Stunden wird es soweit sein.	In seven hours it will be.
b01	Was sind denn das für Tüten, die da unter dem Tisch stehen?	What about the bags standing there under the table?
b02	Sie haben es gerade hochgetragen und jetzt gehen sie wieder runter.	They just carried it upstairs and now they are going down again.
b03	An den Wochenenden bin ich jetzt immer nach Hause gefahren und habe Agnes besucht.	Currently at the weekends I always went home and saw Agnes.
b09	Ich will das eben wegbringen und dann mit Karl was trinken gehen.	I will just discard this and then go for a drink with Karl.
b10	Die wird auf dem Platz sein, wo wir sie immer hinlegen.	It will be in the place where we always store it.

Table XI: Sentences used for each emotion in Emo-DB

APPENDIX B

Timeline Chart of the Project

TIMELINE CHART FOR SEMESTER VII																	
MONTH	JULY				AUGUST					SEPTEMBER				OCTOBER			
WEEK NO.	W1	W2	W3	W4	W1	W2	W3	W4	W5	W1	W2	W3	W4	W1	W2	W3	W4
WORK TASKS																	
1.PROBLEM DEFINITION																	
Domain Selection		■															
Selection of topic and paper			■														
Identify the goal of the project				■													
2.PREPARATION																	
Main Paper Study				■	■												
Study of related IEEE papers and books					■	■	■										
Study and understanding of MFCC							■	■									
Study and understanding of SVM								■	■								
Study of Harmony Search Algorithm									■	■	■	■	■	■			
3.PLANNING																	
Installing MatLab R2014								■									
Installing OpenSmile											■	■	■				
4.EXECUTION OF THE PROJECT																	
Extracting MFCC feature								■	■	■							
Implementing SVM										■	■	■	■				
Extracting features using OpenSmile												■	■	■			

TIMELINE CHART FOR SEMESTER VIII																	
MONTH	JANUARY					FEBRUARY				MARCH				APRIL			
Week No.	W 1	W 2	W 3	W 4	W 5	W 1	W 2	W 3	W 4	W 1	W 2	W 3	W 4	W1	W 2	W 3	W 4
WORK TASKS																	
1. EXECUTION OF THE PROJECT																	
Implementing HARMONY SEARCH																	
Processing Emodb Database																	
Acquisition of IITKGP-SEHSC Database																	
Processing IITKGP-SEHSC Database																	
2. GUI																	
Designing final GUI																	
3. BLACKBOOK AND DOCCUMENTATIO N																	
Blackbook																	
Presentation																	
Poster																	

References

[1] El Ayadi, Moataz, Mohamed S. Kamel, and Fakhri Karray. "Survey on speech emotion recognition: Features, classification schemes, and databases." Pattern Recognition 44.3 (2011): 572-587.

[2] Tao, Yongsen, et al. "Harmony search for feature selection in speech emotion recognition." Affective Computing and Intelligent Interaction (ACII), 2015 International Conference on. IEEE, 2015.

[3] B. Schuller, G. Rigoll, M. Lang, Speech emotion recognition combining acoustic features and linguistic information in a hybrid support vector machine-belief network architecture, in: Proceedings of the ICASSP 2004, vol. 1, 2004, pp. 577–580.

[4] D.J. France, R.G. Shiavi, S. Silverman, M. Silverman, M. Wilkes, Acoustical properties of speech as indicators of depression and suicidal risk, IEEE Trans. Biomedical Eng. 47 (7) (2000) 829–837.

[5] J. Hansen, D. Cairns, Icarus: source generator based real-time recognition of speech in noisy stressful and Lombard effect environments, Speech Commun. 16 (4) (1995) 391–422.

[6] J. Ma, H. Jin, L. Yang, J. Tsai, in: Ubiquitous Intelligence and Computing: Third International Conference, UIC 2006, Wuhan, China, September 3–6, 2006, Proceedings (Lecture Notes in Computer Science), Springer-Verlag, New York, Inc., Secaucus, NJ, USA, 2006

[7] R. Bellman, Dynamic Programming. Princeton, NJ, USA: Princeton University Press,

[8] ——, “Are more features better? a response to attributes reduction using fuzzy rough sets,” IEEE Trans. Fuzzy Syst., vol. 17, no. 6, pp. 1456–1458, 2009.

[9] J. Lee and M. Verleysen, Nonlinear Dimensionality Reduction, ser. Information Science and Statistics. Springer, 2007.

[10] R. Banse, K. Scherer, Acoustic profiles in vocal emotion expression, J. Pers. Soc. Psychol. 70 (3) (1996) 614–636.

[11] T. Nwe, S. Foo, L. De Silva, Speech emotion recognition using hidden Markov models, Speech Commun. 41 (2003) 603–623.

[12] L. Kaiser, Communication of affects by single vowels, Synthese 14 (4) (1962) 300–319.

[13] Drisya, P. S., and Rajeev Rajan. "Significance of teo slope feature in speech emotion recognition." *Networks & Advances in Computational Technologies (NetACT), 2017 International Conference on*. IEEE, 2017.

[14] Wang, Kunxia, et al. "Speech emotion recognition using fourier parameters." *IEEE Transactions on affective computing*6.1 (2015): 69-75.

[15] Chris Ding and Xiaofeng He, "K-Means Clustering via Principal Component Analysis", In proceedings of the 21st International Conference on Machine Learning, Banff, Canada, 2004

[16] Sorzano, Carlos Oscar Sánchez, Javier Vargas, and A. Pascual Montano. "A survey of dimensionality reduction techniques." *arXiv preprint arXiv:1403.2877* (2014).

[17] "Feature selection for classification," *Intelligent Data Analysis*, vol. 1, pp. 131–156, 1997.

[18] *Computational Methods of Feature Selection (Chapman & Hall/CRC Data Mining and Knowledge Discovery Series)*. Chapman & Hall/CRC, 2007.

[19] N. Mac Parthaláin, R. Jensen, Q. Shen, and R. Zwiggelaar, "Fuzzy-rough approaches for mammographic risk analysis," *Intelligent Data Analysis*, vol. 14, no. 2, pp. 225–244, Apr. 2010.

[20] N. Mac Parthaláin, "Guiding rough and fuzzy-rough feature selection using alternative evaluation functions and search strategies," Ph.D. dissertation, University of Wales Aberystwyth, 2006.

[21] H. ming Lee, C. ming Chen, J. ming Chen, and Y. lu Jou, "An efficient fuzzy classifier with feature selection based on fuzzy entropy," *IEEE Trans. Syst., Man, Cybern. B*, vol. 31, pp. 426–432, 2001.

[22] C. Shang and D. Barnes, "Fuzzy-rough feature selection aided support vector machines for mars image classification," *Computer Vision and Image Understanding*, vol. 117, no. 3, pp. 202–213, 2013.

[23] Q. Shen and R. Jensen, "Selecting informative features with fuzzy-rough sets and its application for complex systems monitoring," *Pattern Recognition*, vol. 37, no. 7, pp. 1351–1363, 2004.

[24] E. P. Xing, M. I. Jordan, and R. M. Karp, "Feature selection for high-dimensional genomic microarray data," in *Proceedings of the 18th International Conference on Machine Learning*. Morgan Kaufmann, 2001, pp. 601–608.

[25] I. Kononenko, E. Simec, and M. Robnik-Sikonja, "Overcoming the myopia of inductive learning algorithms with RELIEFF," *Applied Intelligence*, vol. 7, pp. 39–55, 1997.

[26] M. Dash and H. Liu, "Consistency-based search in feature selection," *Artificial Intelligence*, vol. 151, no. 1-2, pp. 155–176, Dec. 2003.

[27] M. Hall, "Correlation-based feature subset selection for machine learning," Ph.D. dissertation, University of Waikato, Hamilton, New Zealand, 1998.

[28] C.-N. Hsu, H.-J. Huang, and D. Schuschel, "The annigma-wrapper approach to fast feature selection for neural nets," *IEEE Trans. Syst*, vol. 32, no. 2, pp. 207–212, 2002.

[29] R. Jensen and Q. Shen, "Fuzzy-rough sets assisted attribute selection," *IEEE Trans. Fuzzy Syst.*, vol. 15, no. 1, pp. 73–89, 2007.

[30] *Computational Intelligence and Feature Selection: Rough and Fuzzy Approaches*. Wiley-IEEE Press, 2008.

[31] "New approaches to fuzzy-rough feature selection," *IEEE Trans. Fuzzy Syst.*, vol. 17, no. 4, pp. 824–838, Aug. 2009.

[32] R. Kohavi and G. H. John, "Wrappers for feature subset selection," *Artificial Intelligence*, vol. 97, no. 1, pp. 273–324, 1997.

[33] M. Lippi, M. Jaeger, P. Frasconi, and A. Passerini, "Relational information gain," *Machine Learning*, vol. 83, pp. 219–239, 2011.

[34] N. Mac Parthaláin, Q. Shen, and R. Jensen, "A distance measure approach to exploring the rough set boundary region for attribute reduction," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 3, pp. 305–317, Mar. 2010.

[35] S. Senthamarai Kannan and N. Ramaraj, "A novel hybrid feature selection via symmetrical uncertainty ranking based local memetic search algorithm," *Knowledge-Based Systems*, vol. 23, no. 6, pp. 580–585, Aug. 2010.

[36] Z. Zheng, "Feature selection for text categorization on imbalanced data," *ACM SIGKDD Explorations Newsletter*, vol. 6, p. 2004, 2004.

[37] Z. Zhu, Y.-S. Ong, and M. Dash, "Wrapper-filter feature selection algorithm using a memetic framework," *IEEE Trans. Syst., Man, Cybern. B*, vol. 37, no. 1, pp. 70–76, 2007.

[38] E. P. Xing, M. I. Jordan, and R. M. Karp, "Feature selection for high-dimensional genomic microarray data," in Proceedings of the 18th International Conference on Machine Learning, pp. 601–608, Morgan Kaufmann, 2001.

[39] Lee, K.S., Geem, Z.W.: A new structural optimization method based on the harmony search algorithm. Computers and Structures 82, 781–798 (2004)

[40] Chen, G.T., Pelizzari, C.A., Spelbring, D.R., Awan, A.: Evaluation of treatment plans using dose volume histograms. Front Radiat. Ther. Oncol. 21, 44–55 (1987)

[41] Chen, Y., Boyer, A.L., Xing, L.: A dose-volume histogram based optimization algorithm for ultrasound guided prostate implants. Medical Physics 27, 2286–2292 (2000)

[42] Kemmerer, T., Lahanas, M., Baltas, D., Zamboglou, N.: Dose-volume histograms computation comparisons using conventional methods and optimized fast Fourier transforms algorithms for brachytherapy. Medical Physics 27, 2343–2356

[43] Karouzakis, K., Lahanas, M., Milickovic, N., Giannouli, S., Baltas, D., Zamboglou, N.: Brachytherapy dose–volume histogram computations using optimized stratified sampling methods. Medical Physics 29, 424–432 (2002)

[44] Schreibmann, E., Lahanas, M., Xing, L., Baltas, E.: Multiobjective evolutionary optimization of the number of beams, their orientations and weights for intensity-modulated radiation therapy. Phys. Med. Biol. 49, 747–770 (2004)

[45] Lessard, E., Pouliot, J.: Inverse planning anatomy-based dose optimization for HDRbrachytherapy of the prostate using fast simulated annealing algorithm and dedicated objective function. Medical Physics 28, 773–779 (2001)

[46] Jozsef, G., Streeter, O.E., Astrahan, M.A.: The use of linear programming in optimization of HDR implant dose distributions. Medical Physics 30, 751–760 (2003)

[47] Geem, Z.W., Kim, J.H., Loganathan, G.V.: A new heuristic optimization algorithm: harmony search. Simulation 76, 60–68 (2001)

[48] Kim, J.H., Geem, Z.W., Kim, E.S.: Parameter estimation of the non-linear Muskingum model using harmony search. Journal of the AWRA, 37, 1131–1138 (2001)

[49] Geem, Zong Woo, ed. *Music-inspired harmony search algorithm: theory and applications*. Vol. 191. Springer, 2009.

[50] K. S. Lee, Z. W. Geem, S.-h. Lee, and K.-w. Bae, "The harmony search heuristic algorithm for discrete structural optimization," Engineering Optimization, vol. 37, no. 7, pp. 663–684, 2005

[51] Z. W. Geem, Ed., Recent Advances In Harmony Search Algorithm, ser. Studies in Computational Intelligence. Springer, 2010, vol. 270

[52] ——, "Two new approaches to feature selection with harmony search," in IEEE International Conference on Fuzzy Systems, Jul. 2010, pp. 1–7.

[53] Diao, Ren, and Qiang Shen. "Feature selection with harmony search." *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 42, no. 6 (2012): 1509-1523.

Acknowledgement

We are thankful to a number of individuals who have contributed towards our final year project and without their help; it would have not been possible t complete it. Firstly, we take this opportunity to express our profound gratitude and deep regards to our project guide, Mr. Kevin D'souza for his exemplary guidance, monitoring and timely encouragement throughout this project. We are thankful to Professor K. S. Rao from IIT Kharagpur for providing us the permission to use IITKGP- SEHSC Hindi Database. And special thanks to all our friends that helped in making of the database.

We are grateful to all project coordinators for their valuable inputs to our project. We are also grateful to the college authorities and the entire faculty of EXTC Department for their support in providing us with the facilities required throughout this semester.

We are also highly grateful to Dr. Gautam A. Shah, Head of Department (EXTC), Principal, Dr. (Mrs.) Sincy George, and Director Bro. Jose Thuruthiyil for providing the facilities, a conducive environment and encouragement.

Signatures of all the students in the group

(Pankaj Chauhan)

(Gaurang Date)

(Neha Nagarkoti)