EE 4541: EMOTION RECOGNITION FROM SPEECH SIGNALS

NAME: SAUPTIK DHAR (dharx007@umn.edu)

CONTENTS PAGE

1. INTRODUCTION………………………………………………………………… 1
2. DATASET USED……………………………………………………………….. 2
3. METHOD USED……………………………………………………………….. 3
4. EXPERIMENTAL SETUP……………………………………………………. 11
5. RESULTS……………………………………………………………………….. 11
6. DISCUSSION of RESULTS/ ISSUES……………………………………… 14
7. CONCLUSION………………………………………………………………… 14

APPENDIX………………………………………………………………………….. 15
REFERENCE……………………………………………………………………….. 24

1. INTRODUCTION

EMOTION RECOGNITION from SPEECH signals has been a prime domain for research for the past four decades. One of the primary steps for building a good system for emotion recognition is to extract the discriminative features of the speech signals. To this end researchers have used the statistics of the different attributes of speech for a 'good' representation of the signal. Basically, these attributes have been broadly categorized as contextual and non-context based attributes.

NON CONTEXT BASE

In this case the speech data were collected in which some actors were asked to speak some prescribed utterance feigning the desired emotion. As the data is totally uncorrelated to the environment; it precludes any paralinguistic utterance for discrimination. ([2], [3]).The popular features used in classification for this category are:-

Prosodic: - pitch-related feature, energy-related features, and speech rate.
Spectral features: - MFCC and cepstral features.

CONTEXT BASED

In this case the paralinguistic displays play an important role in emotion recognition. Researchers moved to the next step where the database was based on the naturalistic audio recordings. The very first preliminary experiment in this direction was that of Devillers and Vidrascu [4], where they found lexical cues resulted in a better performance than using paralinguistic cues to detect relief, anger, fear, and sadness in human-human medical call conversations. Some more work investigated the effect of Linguistic features which shows better performance with the added features [4], [5]. Although, the above studies indicated recognition improvement by using information on language, discourse, and context, the automatic extraction of these related features is a difficult problem. This is because existing automatic speech recognition (ASR) systems cannot reliably recognize the verbal content of emotional speech (e.g., [6]).

In this project I mainly concentrate upon building up a basic system using just the prosodic and the spectral features. My work is inline with the works reported in [1], [2].Rather than deriving some new features based upon some intuitions, I concentrate mainly upon the basic system as described in [2], [7] and try to understand the important features that are important from the emotion recognition perspective. This problem has been addressed in a number of previous works. A very basic setup for this has been explained in [2],[7]. In this case they derive a number of statistics to derive the feature vector representation. More recently, this problem has also been addressed in [1]. However, unlike my basic model they tried to segment the data based on the voiced and unvoiced regions. For my present project I shall stick to the very basic model. Further I shall propose a method based on sparse coding to derive the 'important' features necessary for emotion recognition from speeches. Further, I compare the proposed method with the different feature selection methods already proposed in the literature.

2. DATASET USED

In this section I provide a description of the data used for this project. The data used for this project is the Emo-DB (http://www.expressive-speech.net/emodb).

Basically in this dataset there are 10 different speakers (Male=5, Female=5). Each one of the speakers is asked to speak 10 different texts (in German). These are:-

Text (in GERMAN)	Translation (in ENGLISH)
Der Lappen liegt auf dem Eisschrank.	The tablecloth is lying on the frigde.
Das will sie am Mittwoch abgeben.	She will hand it in on Wednesday.
Heute abend könnte ich es ihm sagen.	Tonight I could tell him.
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.
In sieben Stunden wird es soweit sein.	In seven hours it will be.
Was sind denn das für Tüten, die da unter dem Tisch stehen?	What about the bags standing there under the table?
Sie haben es gerade hochgetragen und jetzt gehen sie wieder runter.	They just carried it upstairs and now they are going down again.
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.
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.
Die wird auf dem Platz sein, wo wir sie immer hinlegen.	It will be in the place where we always store it.

Moreover, each one of the speakers speaks these with different emotions. The different emotions shown during speaking these words are: - anger, boredom, disgust, fear, happiness, sadness and neutral. Basically, for each emotion the number of samples is:-

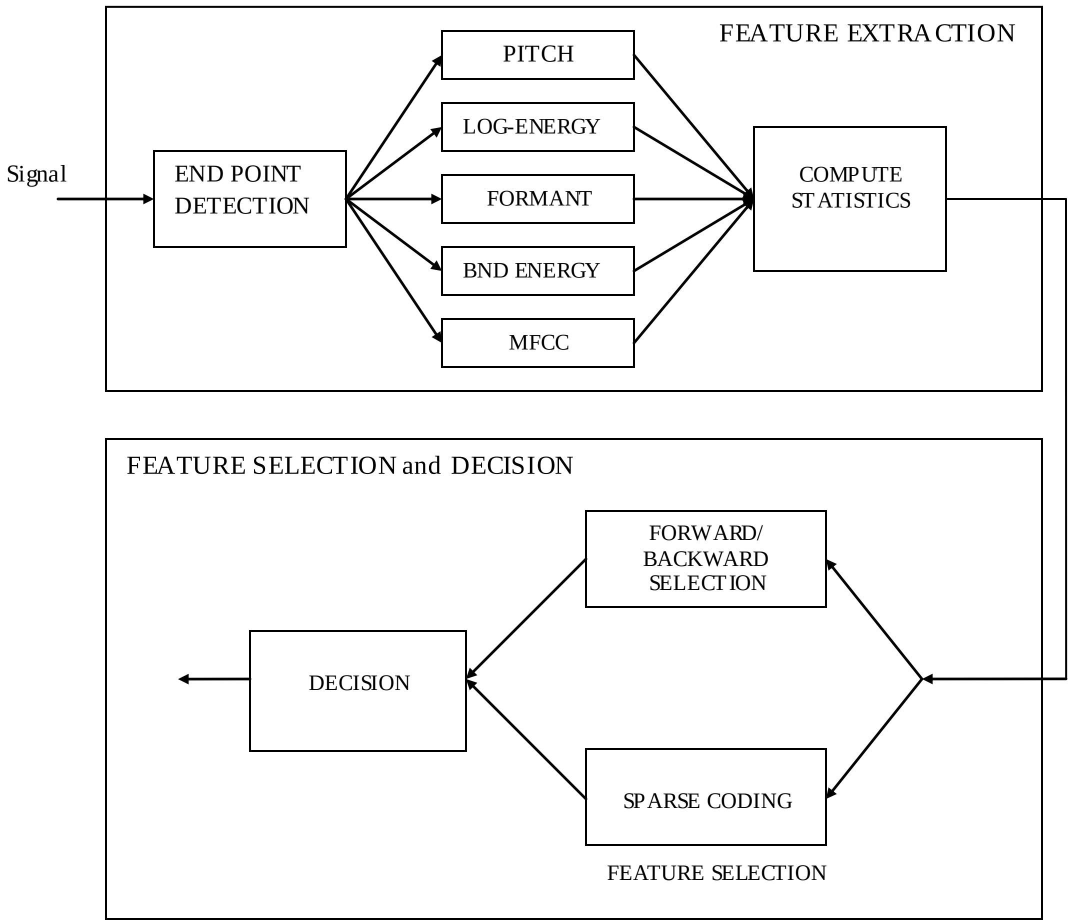
Emotions	Anger	Boredom	Disgust	Fear	Happiness	Sadness	Neutral
No. of samples	127	81	46	69	71	62	79

Moreover, as stated in [1] apart from classifying between the different emotions the data is also grouped into making a distinction between the following groups of states:

Activation: (anger, disgust, fear, happiness) - (boredom, sadness) - (neutral).
Evaluation: (anger, boredom, disgust, anxiety, sadness) - (happiness) - (neutral).

3. METHOD USED

The Basic model for Emotion Recognition typically used is:-



This block diagram is mostly based upon [2],[7]. A brief description of the different blocks and the algorithm used for the different blocks is provided below. Moreover a brief description of the codes usage for the entire project is provided in Appendix 3.

BRIEF DESCRIPTION (OF BLOCKS)

END-POINT DETECTION*:* - An important problem in speech processing is to detect the presence of speech in a background of noise. For this purpose I have implemented the algorithm described in [8]. The basic concept for this algorithm is that speech signals generally have high energy and low zero-crossing rates as compared to non-speech signals. An illustration of the algorithm is provided below.

In Fig 1 I show the original signal and the signal after end point detection. We can clearly see that the initial and the final noise of the signal have been removed. Fig 2 shows the Energy and the zero-crossing rates.

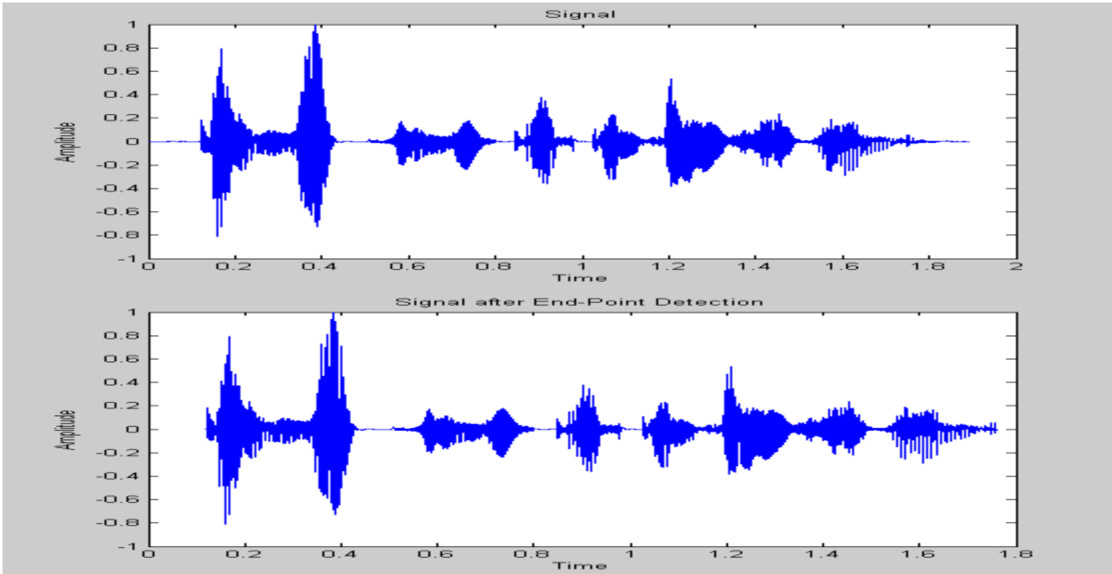


Fig 1: Speech signal before and after end point detection

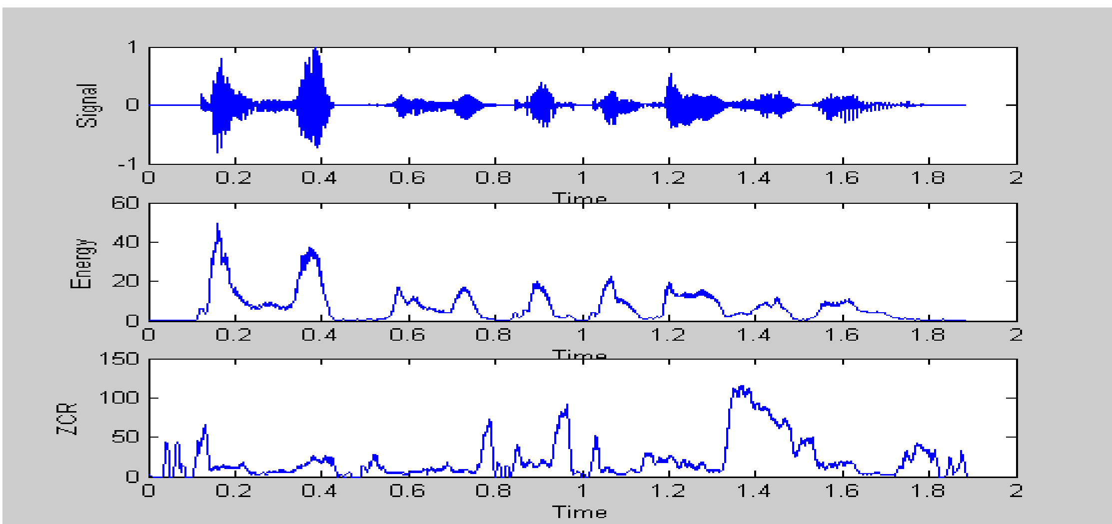


Fig 2: Illustration of the End-Point Detection Algorithm

After the end point detection we make frames of size 40ms with frame-shifts of 10ms and extract the features as described next for each of the frames. For each frames I use a Hamming window of the same size as the frame-size.

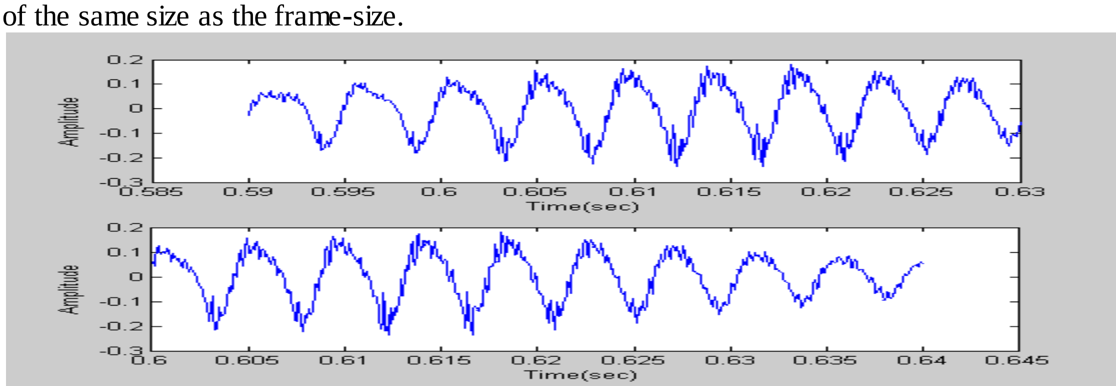


Fig3 Plot showing the 60th and 61st frames of the signal after End-Point Detection (for Happy).

FEATURE EXTRACTION

I select several basic speech attributes for the feature extraction. These are:-

- Pitch (F0)
- Log-Energy
- Formant Frequencies (F1, F2, F3).
- Mel-Band Energy.(5 Mel Band Energy)
- Mel-Frequency Cepstral coefficients (C0-C12).

PITCH

A pitch is the Fundamental Frequency (F0) of the quasi-periodic speech signals. There are a number of algorithms to detect the Pitch of the quasi-periodic speech signals. For my project I detect the pitch based on the cepstrum method as described in [9].Basically the concept behind this is that if we consider that the log amplitude spectrum contains many regularly spaced harmonics, then the Fourier analysis of its spectrum will show a peak corresponding to the spacing between the harmonics: i.e. the fundamental frequency.

A graphical display of the output for the 60th Frame is shown in Fig 3. (This starts at 0.59 sec to around 0.63 sec).

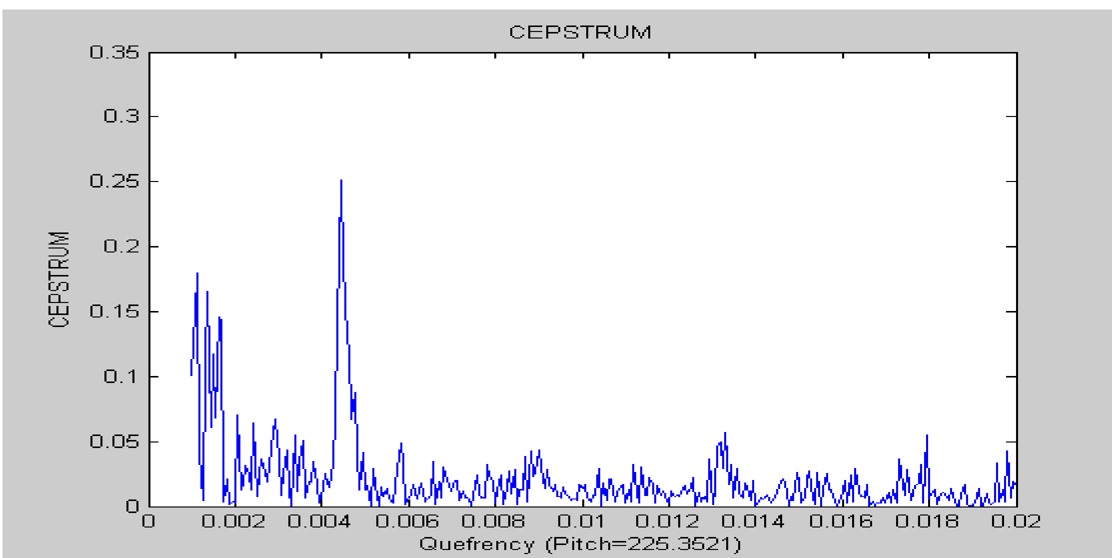


Fig 4: Pitch detection based on cepstrum.

LOG ENERGY

This is the Log of the energy of each frame. I just find the frame energy in the db scale. Note that in the signal the voiced region near the Frame 1 is small as against the Frame 60.Which is what we also see from the Log-Energy Plot.

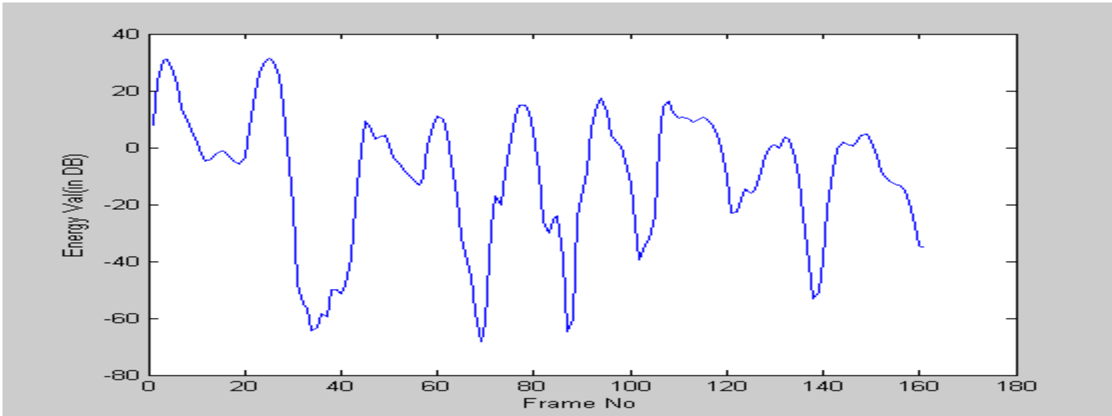


Fig 4: Log-Energy Values.

FORMANT ESTIMATION

In speech signals the Formant frequencies typically depict the resonance of the vocal tract. In this project I find the Formant frequencies (F1,F2,F3) by finding the poles of the AR model of the Vocal Tract.(I use the Levinson-Durbin recursive algorithm available in MATLAB)[7].The output for Frame 60 is shown below:-

The 3 Formant frequencies are F1= 261.79, F2=2254.2 and F3=2539.7

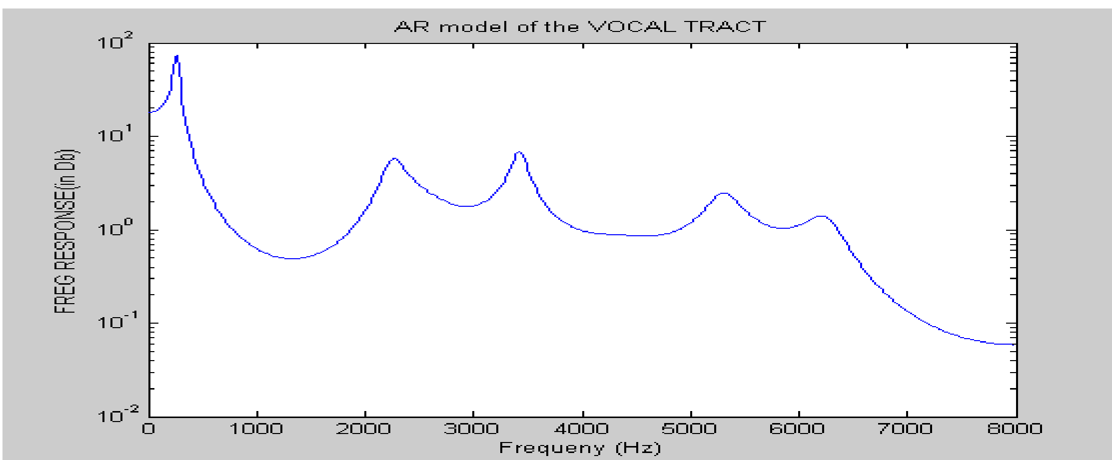


Fig 4: |H (w)| of the AR (all pole) model. (in Db)

MEL-BAND ENERGY

I use first 5-Mel Band Energy for my purpose [2]. Basically I first design the nonlinear Mel Filter banks and derive the signal energy by dividing the frequency band from (130-6800 Hz) [12] using 40-Filter banks. Then I compute the energy within the first 5 filter banks.

MEL-FREQUENCY CEPSTRAL COEFFICIENTS (C0-C12)

I use the log-energy of the 40 Filter banks and then compute the Discrete Cosine Transform to get the MFCC coefficients. I use the first 13 MFCC coefficients. A detailed description for deriving the MFCC s can be found in [11]. For the purpose of display I show the 40-Filter Banks in Fig 5.

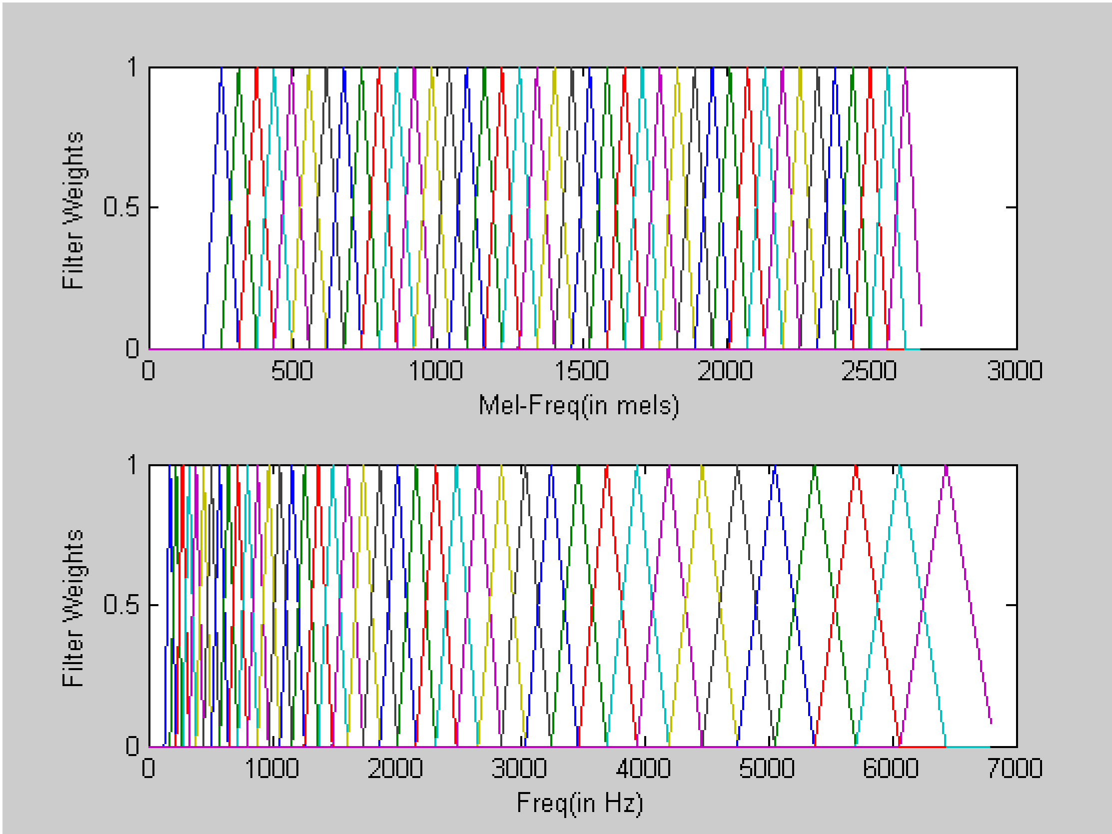


Fig 5: The Filter bank weights in Mel scale and the frequency scale.

Note: I could have weighted the Filter weights in accordance to the filter bank widths as suggested in [7]. However, I intended to follow the algorithm in the original paper [10].

COMPUTE STATISTICS

In this block I compute the statistics of the attributes obtained for all the frames. Basically, the statistics that I compute for each signal attribute are :- mean, deviation, 1st quartile, 2nd quartile,3rd quartile, semi-interquartile deviation, 90 percentile, max, min, skew, kurtosis. I use these 11 statistical values computed for all the signal attributes as a representative for the signal. As such, any signal sample will be represented by a feature vector of dimension 253. Note that,

12 x [Pitch(1)+log-energy(1)+Formant Frequencies(3)+Mel-Band Energy (5)+ MFCC(13)] =253

This completes the block for feature extraction.

FEATURE SELECTION

In this block we do feature selection. Basically we explore 3 feature selection algorithms.

- Forward Selection.
- Backward Selection.
- Sparse Coding.

FORWARD SELECTION (METHOD 1)

This is one of the most basic algorithms that are used in feature selection for speech recognition. Here I add the features sequentially. At each step I add the feature that minimizes the loss function .Finally I use a stopping criterion where the relative improvement is less than 10^{-10}. An illustration of this algorithm is given below:-

ALGORITHM

[STEP 1**]:** Approximate the loss function as a linear function.

$$y = sign(< \mathbf{w}, \mathbf{x} >)$$

[STEP 2]: At each step add a feature that provides the least error.

[STEP 3]: Stop when the relative improvement is <= 10^{-10}

(This is already available in MATLAB using the routine **sequentialfs**)

BACKWARD SELECTION (METHOD 2)

This is similar to the FORWARD SELECTION algorithm except that in this case we initially take all the features and remove the feature that reduces the loss function the minimum. The stopping criterion is set similar to that as before. An illustration of this algorithm is given below:-

ALGORITHM

[STEP 1**]:** Approximate the loss function as a linear function.

$$y = sign(< \mathbf{w}, \mathbf{x} >)$$

[STEP 2]: At each step remove a feature that provides with the least degradation in accuracy.

[STEP 3]: Stop when the relative degradation is <= 10^{-10}

(This is already available in MATLAB using the routine **sequentialfs**)

SPARSE CODING (METHOD 3)

In this project I mainly use this method to enhance the performance of the existing methods. The intuition behind this is that, for this case I need only the features that are totally necessary for discriminating between the different emotions. The concept of Sparse coding mainly stress upon the fact that we shall use only the features that contributes strongly to the discrimination of the emotions.

Mathematically, in the basic form sparse coding can be given as:-

$$w* = \arg min_w \sum_i l\left(y_i, f(\boldsymbol{x}_i, \boldsymbol{w})\right) \quad s.t \;\; \|\boldsymbol{w}\|_0^0 \le L \qquad \text{(Equation 3.1)}$$

Here,

l=Some appropriate loss function.

w = parameter for the parametric function.

L= Some Threshold.

$\|\boldsymbol{w}\|_0^0$= Pseudo-norm. (Indicates the sum of non-zero elements in the parameter)

Mathematically, it is not possible to implement the Pseudo-norm. So there are a number of ways to relax the system. One such popular method is to use the L-1 norm. This converts the problem to a convex optimization problem. There is also an alternative to use the L-2 norm and then to threshold with an optimal threshold. This would still induce sparsity; but there are a number of drawbacks to this approach. We explain them by reproducing the experiment provided in [13].

In the experiment they take a matrix A ($R^{100\times30)\ \text{and}}$ vector b (R^{100})(chosen at random, but the results are typical), and compute the ℓ1-norm and ℓ2-norm approximate solutions of Ax ≈ b

From the plots of the penalty functions we note that:-

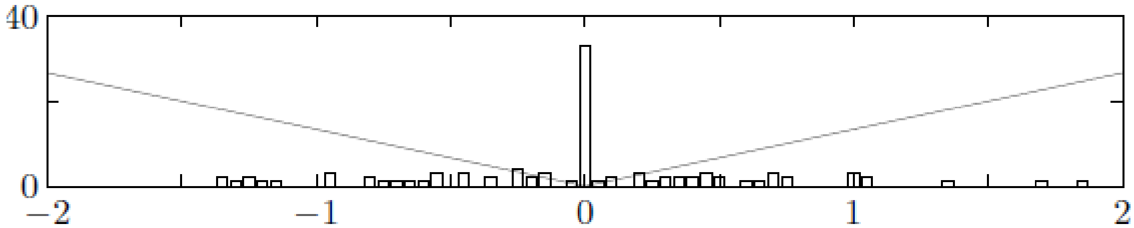


Fig 6 Histogram of residual amplitudes for L1 penalty function

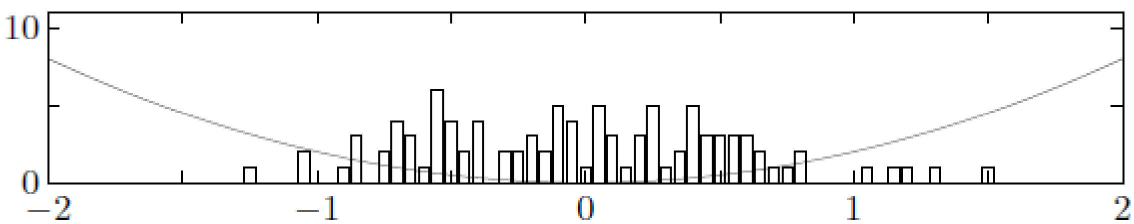


Fig 7 Histogram of residual amplitudes for L-2 penalty function

L1 penalization: For L1 penalization the samples (residual value) are mostly clustered near the origin with very sparse samples scattered away from origin.
L2 penalization: For L2 penalization the samples are mostly clustered on the range of -1 to 1.But the samples are mostly dense in this region. (it would be a bit hard to threshold.)

Quiet conspicuous from the experiment is that the L1 norm is a better choice for inducing sparsity and is much easier to apply a threshold in that case as well. Thus we relax the equation 3.1 to a more easily achievable equation provided below:-

$$w *= \arg min_w \sum_i l\big(y_i, f(\boldsymbol{x}_i, \boldsymbol{w})\big) \; + \; \lambda \|\boldsymbol{w}\|_1$$

Here,
λ= the regularization parameter which strikes a balance between the degree of sparsity and the accuracy of the model.

Thus we use an L1 regularization factor to obtain a sparse parameter. Using this we can obtain a sparse prototype vector representation by using the algorithm below:-

ALGORITHM

STEP 1: Use a Fixed vector representation of speech signals.(Let the representation be given by X).We assume a model as

$$y \cong \boldsymbol{w}^T * \boldsymbol{x} \quad \text{(A least square solution)}$$

STEP 2: Perform supervised feature selection on the representations. (with sparse coding).
Let,

$$f(x,w) = w^T * x$$

Thus, we have

$$w *= \arg min_w \|y - f(x,w)\|_2^2 + \lambda \|\boldsymbol{w}\|_1$$

STEP 3: Compute the relevant prototype representation.

$$\boldsymbol{a} = \boldsymbol{x}(i) \qquad \boldsymbol{s.t}\ i \epsilon R$$

where,

$$R = \{i | w * > threshold\}$$

$\boldsymbol{a}$ ≡ Retain only the features present in R and discard all the other features.

Note:
In the above algorithm the λ is selected by using 5-Fold Cross validation. Moreover I have deviated from my previous proposal of using Sparse Coding with contradiction. Currently the dataset is linearly separable. As such; any further improvement on this dataset is likely to be trivial. However in this project for the purpose of illustration I have shown the results with this current dataset. Moreover, recently sparse coding for continuous speech has been applied in [14].However rather than complicating the model using dictionaries; I use the raw data directly. It also has a better interpretability. For the Sparse Coding I used a general convex optimization solver YALMIP [15]. Internally I used the SDPT3 toolbox [16].

4. EXPERIMENTAL SETUP

[**SETUP 1**]: For this case I use all the 7 Emotions and perform one vs. all classification for all the emotions. As a baseline method I select the Linear Support Vector Machine (SVM) with the range of C as [0.00001 0.001 0.01, 0.1, 1, 10, 100, 1000] without any feature selection. (I use the libsvm toolbox for this)[17]. The baseline method is compared with Method 1, Method 2 and Method 3 on the basis of (5, 5) Double Re-sampling Prediction Error. [18]

Note: - For method 3 we report the double re-sampling classification error for different threshold values. The different threshold values are selected in the range of [1e-10,1e-9,1e-8,1e-7,1e-6,1e-5,1e-4,1e-3,1e-2,1e-1,0.5] times the max(w).

[**SETUP 2**]: For this case I use the data **Activation** and **Evaluation**. The baseline method is then compared with the Methods 1, Method 2 and Method 3 as described in SETUP1.

5. RESULTS

[**SETUP 1**]: The results for this experiment are provided in Table 1. In this table we provide the Prediction error for the different methods after doing double re-sampling. I also report the total number of features selected for the different experiments in the parenthesis. The different features selected for the different cases (using sparse coding) are provided in the Appendix 1.

EMOTION VS ALL	**METHOD**			
	SVM	**FORWARD SELECTION**	**BACKWARD SELECTION**	**SPARSE CODING (BEST RESULT)**
ANGER	0.020561(253)	0.054206(6)	0.020561(250)	0.020561(251)
BOREDOM	0(253)	0.014953(10)	0(252)	0(69)
HAPPY	0.0318 (253)	0.078505(6)	0.029907 (247)	**0.0280(182)**
SAD	0.005607(253)	0.005607(5)	0.005607(252)	**0.0037(77)**
FEAR	0.011215(253)	0.024299(12)	0.011215(252)	**0.0056(195)**
DISGUST	0.009346 (253)	0.085981(3)	0.009346 (252)	**0.0037(94)**
NEUTRAL	0.0075(253)	0.028037(13)	0.009346(249)	0.0075(179)

Table 1 Double Re-sampling Prediction error for different methods

The plots of the Prediction Error are given for sparse coding for the different Methods. The red line indicates the performance (double re-sampling classification error) of the standard SVM without sparse coding and the blue line indicates the performance of the standard SVM with sparse coding. Although the plots do not show any distinct pattern for the selection of threshold it seems more or less the curve seems to dip near to the 10^{-2} time the maximum weight range.

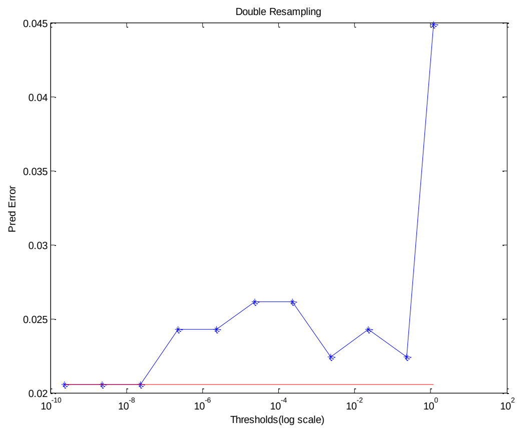


Fig 8 Prediction Error for different thresholds for Anger vs All.

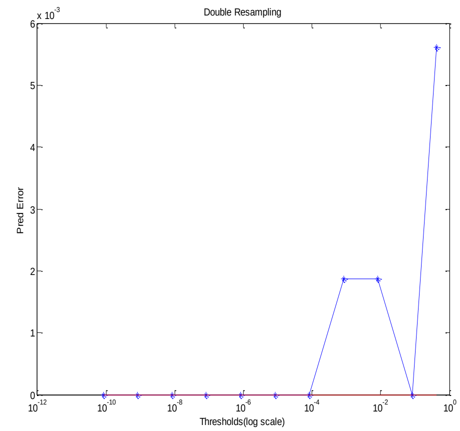


Fig 9 Prediction Error for different thresholds for Boredom vs All.

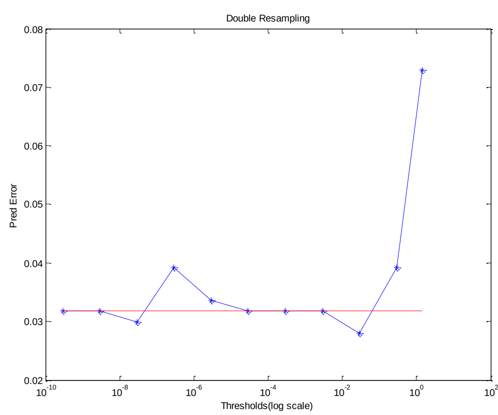


Fig 10 Prediction Error for different thresholds for Happy vs All.

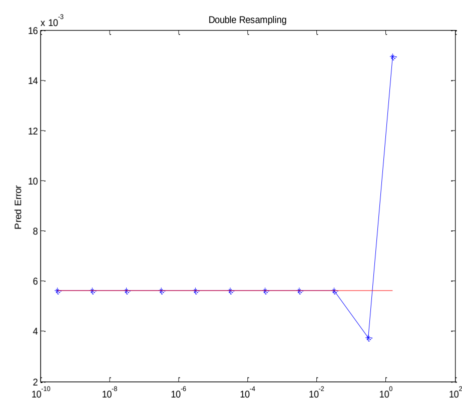


Fig 11 Prediction Error for different thresholds for Sad vs All

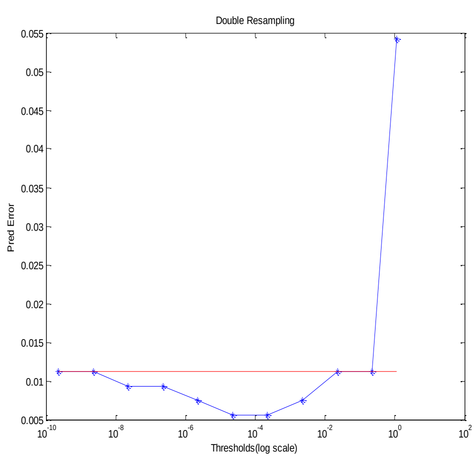


Fig 12 Prediction Error for different thresholds for Fear vs All.

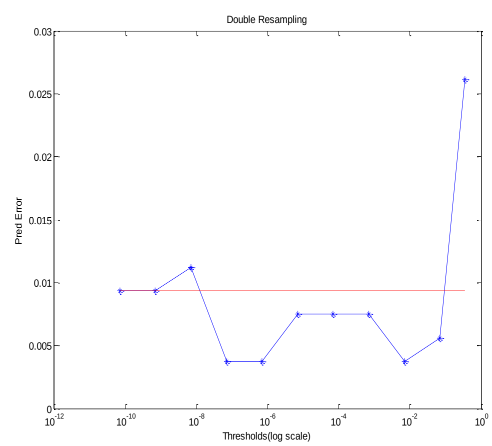


Fig 13 Prediction Error for different thresholds for Disgust vs All.

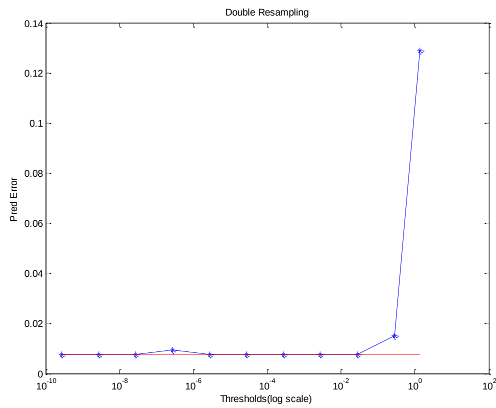


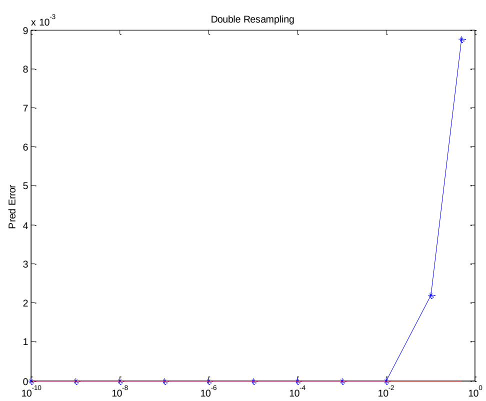
Fig 14 Prediction Error for different thresholds for Neutral vs All.

[**SETUP 2**]: The results for this experiment are provided in Table 2.

DATASET USED	METHOD			
	SVM	**FORWARD SELECTION**	**BACKWARD SELECTION**	**SPARSE CODING (BEST RESULT)**
Activation	0(253)	0.030698(5)	0(250)	0(87)
Evaluation	0.0329(253)	0.096536(6)	0.030721(247)	0.030721(97)

Table 2 Double Re-sampling Prediction error for Setup 2.

The Figures for the different datasets are given below:-



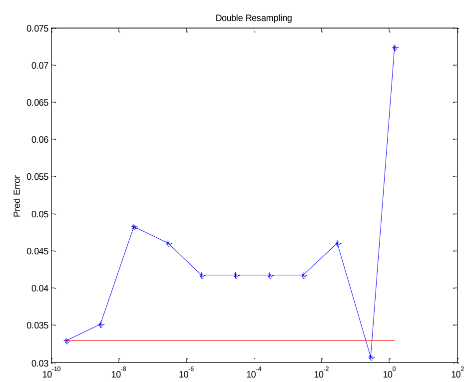


Fig 15 Prediction Error for different thresholds for Activation. Fig 16 Prediction Error for different thresholds for Evaluation.

6. DISCUSSION of RESULTS

[**SETUP 1**]

From Table 1 we can clearly see that the Method 3(Sparse coding) outperforms all the other methods. Further Fig 8-Fig14 shows the performace of the method 3 for different treshold values. The results obtained from this setup suggests that it may be useful to perform feature selection with sparse coding before applying a classifier. Further from Appendix 1 we see that the features that are mostly selected are the features related to the mfcc or the mel-band energy. This is in total agreement with the results obtained in literature.

[**SETUP 2**]

From Table 2 we can similarly see that the sparse coding is helpful.Thus it may be worth to perform this methodology before performing classification.

Note:- There are certain difference in my method of attribute selection in comparison to [1] specifically in terms of :-

- Frame size of the window. (I have chosen 40 ms whereas [1] it is 20ms)
- Segmentation based on voiced and unvoiced region. (I have not removed the unvoiced regions).

DISCUSSION of ISSUES

[1] I do not have the understanding of how to select the best threshold. Currently the theshold used is selected by exhaustively searching over a grid of [1e-10,1e-9,1e-8,1e-7,1e-6,1e-5,1e-4,1e-3,1e-2,1e-1,0.5]* max(w). [Although for this dataset it seems like it may be worthwhile to limit the search within the range of 10^{-2}].

[2] The results reported in this document suggest that the improvement is trivial. A strong conclusion cannot be made unless we test this method on some other dataset which is not highly separable.

7. CONCLUSIONS

Based on the results obtained in this report it seems like sparse coding could yield better performance. However, it would be better to make conclusions after testing the method on some different database.

APPENDIX

APPENDIX 1 (SELECTED FEATURES for SETUP I)

Anger vs All

F0_mean	F2_min	melE3_Q24	c2_mean	c5_min	c9_Q57	c13_mean
F0_dev	F2_skew	melE3_SID	c2_dev	c5_skew	c9_SID	c13_dev
F0_Q1	F2_kurt	melE3_90p	c2_Q34	c5_kurt	c9_90p	c13_Q67
F0_Q2	F3_mean	melE3_max	c2_Q35	c6_mean	c9_max	c13_Q68
F0_Q3	F3_dev	melE3_min	c2_Q36	c6_dev	c9_min	c13_Q69
F0_SID	F3_Q13	melE3_skew	c2_SID	c6_Q46	c9_skew	c13_SID
F0_90p	F3_Q14	melE3_kurt	c2_90p	c6_Q47	c9_kurt	c13_90p
F0_min	F3_Q15	melE4_mean	c2_max	c6_Q48	c10_mean	c13_max
F0_skew	F3_SID	melE4_dev	c2_min	c6_SID	c10_dev	c13_min
F0_kurt	F3_90p	melE4_Q25	c2_skew	c6_90p	c10_Q58	c13_skew
log_mean	F3_max	melE4_Q26	c2_kurt	c6_max	c10_Q59	c13_kurt
log_dev	F3_min	melE4_Q27	c3_mean	c6_min	c10_Q60	
log_Q4	F3_skew	melE4_SID	c3_dev	c6_skew	c10_SID	
log_Q5	F3_kurt	melE4_90p	c3_Q37	c6_kurt	c10_90p	
log_Q6	melE1_mean	melE4_max	c3_Q38	c7_mean	c10_max	
log_SID	melE1_dev	melE4_min	c3_Q39	c7_dev	c10_min	
log_90p	melE1_Q16	melE4_skew	c3_SID	c7_Q49	c10_skew	
log_max	melE1_Q17	melE4_kurt	c3_90p	c7_Q50	c10_kurt	
log_min	melE1_Q18	melE5_mean	c3_max	c7_Q51	c11_mean	
log_skew	melE1_SID	melE5_dev	c3_min	c7_SID	c11_dev	
log_kurt	melE1_90p	melE5_Q28	c3_skew	c7_90p	c11_Q61	
F1_mean	melE1_max	melE5_Q29	c3_kurt	c7_max	c11_Q62	
F1_dev	melE1_min	melE5_Q30	c4_mean	c7_min	c11_Q63	
F1_Q7	melE1_skew	melE5_SID	c4_dev	c7_skew	c11_SID	
F1_Q8	melE1_kurt	melE5_90p	c4_Q40	c7_kurt	c11_90p	
F1_Q9	melE2_mean	melE5_max	c4_Q41	c8_mean	c11_max	
F1_SID	melE2_dev	melE5_min	c4_Q42	c8_dev	c11_min	
F1_90p	melE2_Q19	melE5_skew	c4_SID	c8_Q52	c11_skew	
F1_max	melE2_Q20	melE5_kurt	c4_90p	c8_Q53	c11_kurt	
F1_min	melE2_Q21	c1_mean	c4_max	c8_Q54	c12_mean	
F1_skew	melE2_SID	c1_dev	c4_min	c8_SID	c12_dev	
F1_kurt	melE2_90p	c1_Q31	c4_skew	c8_90p	c12_Q64	
F2_mean	melE2_max	c1_Q32	c4_kurt	c8_max	c12_Q65	
F2_dev	melE2_min	c1_Q33	c5_dev	c8_min	c12_Q66	
F2_Q10	melE2_skew	c1_SID	c5_Q43	c8_skew	c12_SID	
F2_Q11	melE2_kurt	c1_90p	c5_Q44	c8_kurt	c12_90p	
F2_Q12	melE3_mean	c1_max	c5_Q45	c9_mean	c12_max	
F2_SID	melE3_dev	c1_min	c5_SID	c9_dev	c12_min	
F2_90p	melE3_Q22	c1_skew	c5_90p	c9_Q55	c12_skew	
F2_max	melE3_Q23	c1_kurt	c5_max	c9_Q56	c12_kurt	

Boredom vs All

F0_Q1	c5_skew
log_90p	c6_max
log_min	c6_skew
log_kurt	c6_kurt
F1_Q9	c7_dev
F1_kurt	c7_skew
F2_dev	c7_kurt
F2_max	c8_SID
F2_skew	c8_90p
F3_dev	c8_min
F3_Q13	c9_dev
F3_kurt	c9_SID
melE1_dev	c9_90p
melE1_90p	c9_max
melE1_min	c9_min
melE3_Q24	c9_skew
melE3_90p	c10_max
melE3_min	c10_skew
melE4_SID	c10_kurt
melE5_90p	c11_Q63
melE5_max	c11_kurt
melE5_min	c12_Q64
c1_dev	c12_Q66
c1_SID	c12_skew
c1_90p	c13_mean
c1_min	c13_SID
c2_SID	c13_90p
c2_kurt	c13_max
c3_Q39	c13_min
c3_90p	
c3_max	
c3_min	
c3_skew	
c3_kurt	
c4_dev	
c4_max	
c4_skew	
c4_kurt	
c5_Q43	
c5_Q45	

Happy vs All

F0_mean	melE1_dev	c1_Q32	c6_90p	c11_Q62
F0_Q1	melE1_Q16	c1_Q33	c6_max	c11_SID
F0_Q2	melE1_Q17	c1_SID	c6_min	c11_90p
F0_Q3	melE1_SID	c1_90p	c6_skew	c11_max
F0_90p	melE1_90p	c1_min	c6_kurt	c11_min
F0_min	melE1_min	c1_skew	c7_dev	c11_skew
F0_skew	melE1_skew	c1_kurt	c7_Q49	c11_kurt
F0_kurt	melE1_kurt	c2_dev	c7_Q50	c12_dev
log_mean	melE2_dev	c2_Q34	c7_Q51	c12_Q65
log_dev	melE2_Q19	c2_Q35	c7_90p	c12_SID
log_SID	melE2_Q20	c2_SID	c7_max	c12_max
log_90p	melE2_SID	c2_max	c7_min	c12_skew
log_max	melE2_90p	c2_skew	c7_skew	c12_kurt
log_min	melE2_max	c2_kurt	c7_kurt	c13_mean
log_skew	melE2_min	c3_mean	c8_dev	c13_Q68
log_kurt	melE2_skew	c3_dev	c8_Q53	c13_Q69
F1_mean	melE2_kurt	c3_Q37	c8_Q54	c13_SID
F1_dev	melE3_Q22	c3_Q38	c8_SID	c13_90p
F1_Q7	melE3_Q24	c3_90p	c8_90p	c13_max
F1_Q8	melE3_90p	c3_max	c8_max	c13_min
F1_90p	melE3_max	c3_min	c8_min	c13_skew
F1_max	melE3_min	c3_skew	c8_skew	c13_kurt
F1_min	melE3_skew	c3_kurt	c9_mean	
F1_kurt	melE3_kurt	c4_dev	c9_dev	
F2_mean	melE4_mean	c4_Q40	c9_Q56	
F2_Q12	melE4_dev	c4_Q41	c9_SID	
F2_SID	melE4_Q25	c4_90p	c9_90p	
F2_90p	melE4_SID	c4_max	c9_max	
F2_max	melE4_90p	c4_min	c9_min	
F2_min	melE4_max	c4_skew	c9_skew	
F2_skew	melE4_min	c5_Q45	c9_kurt	
F2_kurt	melE4_skew	c5_90p	c10_mean	
F3_dev	melE5_dev	c5_max	c10_dev	
F3_Q15	melE5_Q28	c5_min	c10_Q59	
F3_SID	melE5_Q29	c5_skew	c10_max	
F3_90p	melE5_Q30	c5_kurt	c10_skew	
F3_min	melE5_max	c6_dev	c10_kurt	
F3_skew	melE5_skew	c6_Q46	c11_mean	
F3_kurt	melE5_kurt	c6_Q47	c11_dev	
melE1_mean	c1_dev	c6_Q48	c11_Q61	

SAD vs ALL

F0_Q1	c2_mean
F0_Q2	c2_Q35
F0_skew	c2_Q36
F0_kurt	c3_mean
log_mean	c3_dev
log_Q5	c3_Q38
log_skew	c3_Q39
log_kurt	c3_SID
F1_mean	c3_skew
F1_Q7	c4_mean
F1_Q8	c4_Q40
F1_Q9	c4_Q42
F1_skew	c4_skew
F2_dev	c5_mean
F2_Q10	c5_Q43
melE1_dev	c6_mean
melE1_Q17	c6_Q46
melE1_Q18	c6_Q47
melE1_skew	c6_Q48
melE2_mean	c8_mean
melE2_dev	c8_dev
melE2_Q20	c8_Q52
melE2_Q21	c8_Q53
melE2_max	c8_max
melE3_mean	c9_mean
melE3_Q22	c9_Q55
melE3_Q23	c9_Q57
melE3_Q24	c10_mean
melE3_max	c10_dev
melE3_skew	c10_Q59
melE3_kurt	c10_Q60
melE4_mean	c11_mean
melE4_Q27	c11_Q61
melE4_SID	c11_Q63
melE5_mean	c11_SID
melE5_dev	c12_dev
melE5_Q29	c12_SID
melE5_Q30	
c1_Q31	
c1_skew	

DISGUST vs ALL

F0_dev	c2_90p	c11_90p
log_SID	c2_min	c11_max
log_max	c2_kurt	c11_min
log_skew	c3_dev	c11_skew
F1_Q7	c3_90p	c11_kurt
F1_max	c3_max	c12_mean
F1_kurt	c3_min	c12_Q66
F3_dev	c3_skew	c12_90p
F3_SID	c4_90p	c12_max
F3_max	c4_max	c12_min
F3_skew	c4_min	c12_skew
F3_kurt	c4_skew	c12_kurt
melE1_dev	c5_mean	c13_dev
melE1_Q17	c5_dev	c13_skew
melE1_90p	c5_Q45	
melE1_min	c5_kurt	
melE2_Q19	c6_mean	
melE2_Q20	c6_SID	
melE2_SID	c6_90p	
melE3_dev	c6_min	
melE3_SID	c6_kurt	
melE3_90p	c7_mean	
melE3_min	c7_Q51	
melE4_mean	c7_90p	
melE4_Q27	c7_max	
melE4_SID	c7_min	
melE4_max	c8_mean	
melE4_kurt	c8_dev	
melE5_dev	c8_Q53	
melE5_Q28	c8_SID	
melE5_max	c8_skew	
melE5_skew	c8_kurt	
c1_dev	c9_Q56	
c1_Q32	c9_max	
c1_Q33	c9_skew	
c1_90p	c9_kurt	
c1_max	c10_Q59	
c1_min	c10_max	
c1_skew	c10_kurt	
c1_kurt	c11_Q63	

FEAR vs ALL

F0_mean	F3_Q15	melE5_Q29	c5_max	c9_skew
F0_dev	F3_90p	melE5_Q30	c5_min	c9_kurt
F0_Q1	F3_max	melE5_90p	c5_skew	c10_mean
F0_Q2	F3_min	melE5_max	c5_kurt	c10_Q58
F0_SID	F3_skew	melE5_min	c6_mean	c10_Q59
F0_90p	F3_kurt	melE5_skew	c6_dev	c10_90p
F0_min	melE1_dev	c1_dev	c6_Q46	c10_max
F0_skew	melE1_Q16	c1_Q31	c6_Q47	c10_min
F0_kurt	melE1_Q17	c1_Q33	c6_Q48	c10_skew
log_mean	melE1_90p	c1_90p	c6_90p	c10_kurt
log_dev	melE1_max	c1_max	c6_max	c11_dev
log_Q5	melE1_min	c1_min	c6_min	c11_Q62
log_Q6	melE1_skew	c1_skew	c6_skew	c11_Q63
log_SID	melE1_kurt	c1_kurt	c6_kurt	c11_SID
log_90p	melE2_dev	c2_Q35	c7_dev	c11_90p
log_max	melE2_Q19	c2_SID	c7_Q50	c11_max
log_min	melE2_Q21	c2_90p	c7_Q51	c11_min
log_skew	melE2_90p	c2_max	c7_SID	c11_skew
log_kurt	melE2_max	c2_min	c7_90p	c11_kurt
F1_dev	melE2_min	c2_skew	c7_max	c12_mean
F1_Q7	melE2_skew	c3_Q37	c7_min	c12_dev
F1_90p	melE2_kurt	c3_Q38	c7_skew	c12_Q65
F1_max	melE3_mean	c3_90p	c7_kurt	c12_Q66
F1_min	melE3_dev	c3_max	c8_mean	c12_SID
F1_skew	melE3_Q22	c3_min	c8_dev	c12_90p
F1_kurt	melE3_Q23	c3_skew	c8_Q53	c12_max
F2_mean	melE3_max	c3_kurt	c8_Q54	c12_min
F2_dev	melE3_skew	c4_mean	c8_SID	c12_skew
F2_Q11	melE3_kurt	c4_dev	c8_90p	c12_kurt
F2_Q12	melE4_dev	c4_Q40	c8_max	c13_mean
F2_SID	melE4_Q26	c4_Q41	c8_min	c13_dev
F2_90p	melE4_Q27	c4_90p	c8_skew	c13_SID
F2_max	melE4_SID	c4_max	c8_kurt	c13_max
F2_min	melE4_90p	c4_min	c9_mean	c13_skew
F2_skew	melE4_max	c4_skew	c9_dev	c13_kurt
F2_kurt	melE4_min	c4_kurt	c9_Q56	
F3_mean	melE4_skew	c5_dev	c9_SID	
F3_dev	melE4_kurt	c5_Q45	c9_90p	
F3_Q13	melE5_mean	c5_SID	c9_max	
F3_Q14	melE5_Q28	c5_90p	c9_min	

NEUTRAL vs ALL

F0_mean	F3_skew	melE5_90p	c5_skew	c11_Q63
F0_dev	F3_kurt	melE5_max	c5_kurt	c11_SID
F0_Q1	melE1_mean	melE5_min	c6_dev	c11_90p
F0_Q2	melE1_dev	melE5_skew	c6_Q47	c11_min
F0_SID	melE1_Q16	melE5_kurt	c6_SID	c11_skew
F0_90p	melE1_Q17	c1_dev	c6_90p	c11_kurt
F0_min	melE1_90p	c1_Q32	c6_max	c12_Q65
F0_kurt	melE1_max	c1_90p	c6_min	c12_Q66
log_dev	melE1_min	c1_max	c6_skew	c12_90p
log_Q5	melE1_skew	c1_min	c6_kurt	c12_max
log_Q6	melE1_kurt	c1_skew	c7_dev	c12_min
log_SID	melE2_dev	c1_kurt	c7_Q50	c12_skew
log_90p	melE2_Q21	c2_Q34	c7_90p	c12_kurt
log_max	melE2_SID	c2_Q35	c7_max	c13_mean
log_min	melE2_90p	c2_Q36	c7_min	c13_SID
log_skew	melE2_min	c2_90p	c7_skew	c13_90p
log_kurt	melE2_skew	c2_max	c7_kurt	c13_min
F1_dev	melE2_kurt	c2_min	c8_dev	c13_skew
F1_Q8	melE3_mean	c3_dev	c8_Q53	c13_kurt
F1_Q9	melE3_dev	c3_Q38	c8_SID	
F1_SID	melE3_Q22	c3_Q39	c8_90p	
F1_max	melE3_Q23	c3_SID	c8_max	
F1_skew	melE3_Q24	c3_90p	c8_min	
F1_kurt	melE3_90p	c3_max	c8_skew	
F2_mean	melE3_max	c3_min	c8_kurt	
F2_dev	melE3_min	c3_skew	c9_dev	
F2_Q10	melE3_skew	c3_kurt	c9_Q55	
F2_Q12	melE4_mean	c4_dev	c9_Q56	
F2_90p	melE4_Q26	c4_Q40	c9_Q57	
F2_max	melE4_Q27	c4_Q41	c9_90p	
F2_min	melE4_SID	c4_Q42	c9_max	
F2_kurt	melE4_90p	c4_90p	c9_min	
F3_mean	melE4_max	c4_min	c9_skew	
F3_dev	melE4_min	c4_kurt	c10_mean	
F3_Q13	melE4_skew	c5_dev	c10_SID	
F3_Q14	melE4_kurt	c5_Q43	c10_max	
F3_Q15	melE5_dev	c5_Q45	c10_skew	
F3_90p	melE5_Q29	c5_90p	c10_kurt	
F3_max	melE5_Q30	c5_max	c11_mean	
F3_min	melE5_SID	c5_min	c11_Q62	

APPENDIX 2 (SELECTED FEATURES for SETUP II)

EVALUATION data

F0_Q2	melE5_skew	c9_Q56
F0_kurt	melE5_kurt	c9_max
log_SID	c1_dev	c9_skew
log_max	c1_Q32	c9_kurt
log_kurt	c1_Q33	c10_dev
F1_mean	c1_SID	c11_mean
F1_dev	c2_dev	c11_dev
F1_Q7	c2_Q35	c11_Q62
F1_max	c2_SID	c11_SID
F1_kurt	c2_skew	c11_max
F2_Q12	c2_kurt	c11_skew
F2_SID	c3_mean	c12_dev
F2_max	c3_dev	c12_Q64
F2_skew	c3_Q38	c13_mean
F3_dev	c3_SID	c13_Q68
F3_kurt	c3_90p	c13_SID
melE1_mean	c3_min	c13_max
melE1_dev	c3_kurt	
melE1_Q16	c4_mean	
melE1_Q17	c4_Q40	
melE1_SID	c4_Q41	
melE1_kurt	c4_max	
melE2_Q19	c4_skew	
melE2_Q20	c5_90p	
melE2_SID	c5_max	
melE2_90p	c5_min	
melE2_min	c5_kurt	
melE2_skew	c6_mean	
melE2_kurt	c6_Q47	
melE3_Q22	c6_Q48	
melE3_SID	c6_skew	
melE3_skew	c6_kurt	
melE3_kurt	c7_dev	
melE4_SID	c7_Q49	
melE4_max	c7_Q50	
melE5_dev	c7_max	
melE5_Q28	c7_kurt	
melE5_Q29	c8_dev	
melE5_Q30	c9_mean	
melE5_max	c9_dev	

Activation data

F0_Q2	c1_dev	c11_SID
F0_90p	c1_Q31	c11_max
F0_min	c1_Q33	c11_kurt
F0_skew	c1_90p	c12_dev
F0_kurt	c1_max	c12_Q65
log_Q5	c1_min	c12_kurt
log_90p	c2_Q36	c13_mean
log_min	c2_90p	c13_SID
log_kurt	c2_max	c13_max
F1_SID	c2_min	
F1_90p	c3_dev	
F1_max	c3_Q39	
F1_min	c3_90p	
F1_kurt	c3_max	
F2_dev	c3_min	
F2_90p	c3_kurt	
F2_max	c4_dev	
F2_min	c4_kurt	
F2_skew	c5_Q45	
F3_mean	c5_skew	
F3_dev	c5_kurt	
F3_Q14	c6_Q48	
F3_kurt	c6_max	
melE1_dev	c6_skew	
melE1_90p	c7_dev	
melE1_min	c7_Q49	
melE2_dev	c7_skew	
melE2_max	c8_SID	
melE2_skew	c8_max	
melE3_Q24	c8_skew	
melE3_90p	c8_kurt	
melE3_max	c9_dev	
melE3_min	c9_SID	
melE4_Q27	c9_max	
melE4_90p	c9_skew	
melE4_max	c10_mean	
melE4_min	c10_dev	
melE4_kurt	c10_max	
melE5_Q28	c10_skew	

APPENDIX 3 (How to use the MATLAB CODES)

The Codes are organized as shown in the Block diagram in Page 3.

FEATURE EXTRACTION

I have provided an example code to obtain the features based on the interfaced developed. In this I have provided the way we shall need to obtain the features for the data for the emotions ‘disgust’ and ‘fear’.

```
function [dataprocessed]=featureextraction()
%--------------------------------------------------------------------------
% AUTHOR:- Sauptik
% Description:-
% [dataprocessed]=featureextraction()
% This is the code used to call all the feature extraction codes.
% INPUT:- NONE(However be sure to provide a correct data path internally.)
% OUTPUT:-dataprocessed:- This is the processed dataset.
%         dataprocessed.X= the X values of the data.
%         dataprocessed.y= the y values of the data.
%--------------------------------------------------------------------------

[data]=loadData('C:\Documents and Settings\Owner\My Documents\SAUPTIK\RESEARCH\DSP
PROJECT\DATA\wav');

data1=data.disgust;
data2=data.fear;

% This is for the data for class 1
for i=1:length(data1)

    %ENDPOINT DETECTION
    [segment] = endpointEZdetect1(data1(i).y,data1(i).FS,0);

    % MAKE FRAMES
    [dataseg]=segData(segment,0.04,0.01,data1(i).FS,0,0);   % FRAME SIZE=40ms,FRAME
SHIFT=10ms.

    % OBTAIN THE FEATURES FOR
    for j=1:length(dataseg.frame)
        frame=dataseg.frame(j).y;           %COMPUTE BASED ON FRAMES
        if(all(frame==0))continue; end

        % PITCH
        [F0(j),cepstrum, spectrum] = pitchCepstrum(frame, data1(i).FS,0);

        % LOG-ENERGY
      [lE(j)]=logEnergy(frame,data1(i).FS,1);

        % FORMANTS F1,F2,F3
        [F1(j),F2(j),F3(j)] = formantsLPC(frame,data1(i).FS,0);
```

```
%MEL-BAND ENERGY
        [E]=melBandEnergy(frame,data1(i).FS,40,130,6800,0);
        mE(:,j)=E(1:5);

        % MFCC
        [cc]=mfcc(E);
        c(:,j)=cc;
    end
    [features_1(i,:)]=calcStat(F0,lE,F1,F2,F3,mE,c);
    fprintf('Please Wait .... step %d out of %d\n',i,length(data1));
end

%This is for the data for class 2
for i=1:length(data2)

    %ENDPOINT DETECTION
    [segment] = endpointEZdetect1(data2(i).y,data2(i).FS,0);

    % MAKE FRAMES
    [dataseg]=segData(segment,0.04,0.01,data2(i).FS,0,0);   % FRAME
SIZE=40ms,FRAME SHIFT=10ms.

    % OBTAIN THE FEATURES FOR
    for j=1:length(dataseg.frame)
        frame=dataseg.frame(j).y;           %COMPUTE BASED ON FRAMES

        if(all(frame==0))continue; end

        % PITCH
        [F0(j),cepstrum, spectrum] = pitchCepstrum(frame, data2(i).FS,0);

        % LOG-ENERGY
        [lE(j)]=logEnergy(frame,data2(i).FS,1);

        % FORMANTS F1,F2,F3
        [F1(j),F2(j),F3(j)] = formantsLPC(frame,data2(i).FS,0);

        %MEL-BAND ENERGY
        [E]=melBandEnergy(frame,data2(i).FS,40,130,6800,0);
        mE(:,j)=E(1:5);

        % MFCC
        [cc]=mfcc(E);
        c(:,j)=cc;
    end
    [features_2(i,:)]=calcStat(F0,lE,F1,F2,F3,mE,c);
    fprintf('Please Wait .... step %d out of %d\n',i,length(data2));
end
dataprocessed.X=[features_1;features_2];
dataprocessed.y=[ones(size(features_1,1),1);-ones(size(features_2,1),1)];
dataprocessed.X=[features_1];
dataprocessed.y=[ones(size(features_1,1),1)];
```

FEATURE SELECTION

For the sake of convenience the preprocessed data is provided. In order to reproduce the results perform the following steps:-

FORWARD SELECTION.

1. Load the intended data.

```
clear;
load angervsall
```

2. Perform Feature selection.

```
[data1,index,fs,history]=FeatureSel(data,'forward',[],1e-10);
```

3. Perform Double Re-sampling SVM

```
[Rpred_rate,Rpred]=buildSVMModel_Lin(data1,5,5);
```

BACKWARD SELECTION.

1. Load the intended data.

```
clear;
load angervsall
```

2. Perform Feature selection.

```
[data1,index,fs,history]=FeatureSel(data,'backward',[],1e-10);
```

3. Perform Double Re-sampling SVM

```
[Rpred_rate,Rpred]=buildSVMModel_Lin(data1,5,5);
```

SPARSE CODING

1. Load the intended data.

```
clear;
load angervsall;
```

2. Perform Sparse coding.

```
[w,lambda]= selectlambda(data,5);
```

3. Perform Double Re-sampling SVM and thresholding.

```
[h,RPred,RpredSVM,thrset,bestindex,Rmin]=ExperimentwithSparse(data,w,5,5);
```

REFERENCE

[1] Casale,S. Russo,A. Scebba,G. Serrano S."*Speech Emotion Classification Using Machine Learning Algorithms*",2008 IEEE International Conference on Semantic Computing.

[2] Kwon O., Chan K., Hao J., Lee T. "*Emotion Recognition by Speech Signals*", Proc. of Eurospeech. 2003, Genewa, p. 125-128, September 2003.

[3] A. Austermann, N. Esau, L. Kleinjohann, and B. Kleinjohann, "Prosody-Based Emotion Recognition for MEXI," Proc. IEEE/RSJ Int'l Conf. Intelligent Robots and Systems (IROS '05), pp. 1138-1144,2005.

[4] L. Devillers and I. Vasilescu, "Real-Life Emotions Detection with Lexical and Paralinguistic Cues on Human-Human Call Center Dialogs," Proc. Ninth Int'l Conf. Spoken Language Processing (ICSLP),2006.

[5] M. Graciarena, E. Shriberg, A. Stolcke, F. Enos, J. Hirschberg, and S. Kajarekar, "Combining Prosodic, Lexical and Cepstral Systems for Deceptive Speech Detection," Proc. Int'l Conf. Acoustics, Speech and Signal Processing (ICASSP '06), vol. I,pp. 1033-1036, 2006.

[6] T. Athanaselis, S. Bakamidis, I. Dologlou, R. Cowie, E. Douglas-Cowie, and C. Cox, "ASR for Emotional Speech: Clarifying the Issues and Enhancing Performance," Neural Networks, vol. 18, pp. 437-444, 2005.

[7] L.R. Rabiner and B.H. Juang, **Fundamentals of Speech Recognition**, Prentice Hall',1993

[8] L.R.Rabiner and M.R.Sambur,'An Algorithm for Determining the Endpoints of Isolated Utterances',The Bell System Technical Journal, Vol. 54,No. 2, Feb. 1975, pp. 297-315.

[9] Project: Pitch Detection (http://note.sonots.com/SciSoftware/Pitch.html)

[10] S.Davis and P.Mermelstein,'Comparison of Parametric Representations for Monosyllabic Word Recognition in Continuously Spoken Sentences',IEEE TRANSACTIONS ON ACOUSTICS, SPEECH, AND SIGNAL PROCESSING, VOL. ASSP-28, NO. 4, AUGUST 1980

[11] X.Huang, A.Acero, H.W Hon, **Spoken Language Processing**,A guide to theory,Algorithm and System development,Prentice Hall 2001

[12] CMU SPHINX PROJECT SPECIFICATIONS
http://cmusphinx.sourceforge.net/sphinx4/javadoc/edu/cmu/sphinx/frontend/frequencywarp/MelFrequencyFilterBank.html

[13] Stephen Boyd and Lieven Vandenberghe,**Convex Optimization**,Cambridge University Press,March 2004.

[14] WJ Smit, '*Continuous speech recognition with sparse coding* ',Computer Speech and Language archive,Volume 23 , Issue 2 ,April 2009.

[15] YALMIP (http://control.ee.ethz.ch/~joloef/wiki/pmwiki.php?n=Main.Download)
[16] SDPT3(http://www.math.nus.edu.sg/~mattohkc/sdpt3.html)
[17] libsvm (http://www.csie.ntu.edu.tw/~cjlin/libsvm/)

[18] V.Cherkassky, F,Mulier, **Learning From Data**: *Concepts, Theory and Methods*, Wiley 2007.