Emotion state detection via speech in spoken Hindi

G. No. 15: Prakhar Kulshreshtha (13485) and Soumya Gayen (13708)

Mentor: Prof. Piyush Rai, IIT Kanpur

April 20, 2016

Abstract

In this project, simulated Hindi emotional speech database has been borrowed from a subset of IITKGP-SEHSC dataset(2 out of 10 speakers). Emotional classification is attempted on the corpus using spectral features. The spectral features used are Mel Frequency Cepstral Coefficients(MFCCs) and Subband Spectral Coefficents(SSCs) The feature vector in use has 273 features, obtained from 7 individual features of 13 banks of MFCCs and 26 SSCs computed over the dataset. This dataset is trained on multiple classifiers, wherein with each classifier, related learning and error penalty rate parameters have been varied to find the best set of classifiers. The lists of accuracies, precisions, and f1-scores are compared. Our methods show that Support Vector Machines with Radial Basis Function kernel provides the best accuracy rates, with accuracy for male dataset being 89.08% and for female dataset being 83.16%. The results are on par with the results obtained by training on full IITKGP-SEHSC dataset.

Keywords-**Spectral Features, MFCCs, SSCs, SVM, RBF, Deep ANN, MLP, Adaboost**

1 INTRODUCTION:

Understanding human-speech has been an integral and fascinating part of AI as well as Digital Speech Processing for a long time. Emotion recognition is also an integral component of understanding speech. Same phrases can convey different emotions when spoken differently. In our project we explore different classifiers to categorize the spoken utterance discretely into 8 states: anger, fear, disgust, happiness, surprise, neutral, sadness and sarcastic, to obtain a system capable of recognizing emotions in speech utterances, with reasonable accuracy. Also, since most people in India are familiar with the spoken Hindi, we chose a Hindi Emotional speech corpus for our testing. However, the system we have built should be able to train nicely, and give reasonable performance on Emotion-Corpus of of speech in any language.

2 DATASET AND PLATFORM:

The dataset we are using is a subset of IIT-KGP SEHSC: Simulated Emotion Hindi Speech Corpus. The dataset contains 15 spoken sentences, each sentence being emotionally neutral in meaning, being spoken in 8 different emotions, in 10 sessions of increasing intensity, hence $15 \times 8 \times 10 = 1200$ utterances. We have these utterances for one male speaker and one female speaker.
The platform we are using is Python, and the following opensource libraries were borrowed:

1. Scikit Learn
2. Pybrain
3. Python Speech Features, MFCC and SSC, James Leon

3 CLASSIFICATION EVALUATION METHOD:

We want our system to be independent of the verbal content of the utterance itself, as the semantic content of the neutral speech used has no distinguishing features. For this, we are using an 'unrandomized' K-Fold Cross Validation. As per our dataset, for one speaker, we have 15 sentence, each spoken in 8 emotions in 10 sessions, so our dataset is divided into 15 folds, each fold containing $8 \times 10 = 80$ utterances corresponding to one sentence. Then the classifier is simply tested for each fold (after training it on the remaining 14 folds), and then the average of accuracies of all the 15 cases is evaluated and maximized.

4 MFCC AND SSC AS FEATURES:

4.1 POWER CEPSTRUM:

In DSP (Digital Signal Processing), a cepstrum is defined as the Inverse Fourier Transform (IFT) of the log of the Power-Spectrum of a signal. 'Power Cepstrum', defined by:

$$PowerCepstrum = \left\| F^{-1}\left\{ log(\left\| F\left\{ f(t)\right\} \right\|^2)\right\} \right\|^2$$

4.2 MEL FREQUENCY CEPSTRUM:

"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" - Wikipedia.

4.3 MEL FREQUENCY CEPSTRUM COEFFICIENTS (MFCC):

MFCCs are the coefficients that can characterise an MFC. Since we don't want our system to be dependent on the 'words' which are spoken, we use these cepstral features. For a given discrete time finite length signal window, MFCC is calculated as (source: http://practicalcryptography.com
, MFCC tutorial):
1. Frame the signal into short frames.
2. For each frame calculate the periodogram estimate of the power spectrum.
3. Apply the mel filterbank to the power spectra, sum the energy in each filter.
4. Take the logarithm of all filterbank energies.
5. Take the DCT of the log filterbank energies.
In our case we split our audio files into smaller 25ms long 'windows', and then we calculate the MFCC using a python library. We are using 512 windows for calculating the FFT, 26 frequency subbands and then finally obtain 13 coefficients for each frame.

4.4 SPECTRAL SUBBAND CENTROID (SSC):

For each of the Mel Frequency sub-band, SSC coefficient is calculated as:

$$SSC(i) = \frac{\sum_{k=1}^{n} f_i(k)x_i(k)}{\sum_{k=1}^{n} x_i(k)}, i = 1 to 26$$

Where, $f_i(k)$ is the kth frequency belonging to the ith bank, and $f_i(k)$ is its corresponding power amplitude, which is acting as a weight here. SSC features are used alongside MFCC because MFCC features aren't robust to white noise or to the variation of overall intensity of the spoken sound. So we tried out this feature in hope of increasing the overall accuracy, which it did, and hence we have concatenated this feature as well.

4.5 Removing the effect of 'Different audio lengths':

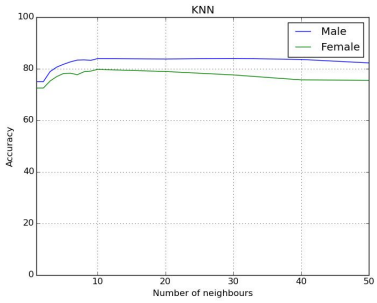
We extracted the MFCC and SSC, but then we were facing two problems:
1. Average length of the files was around 7 seconds, window length being 25ms, so we had around $7 \times (13+26) \times 1000/25 = 10920$ features per utterance!
2. Our audio files are of different lengths so extracted features will be of different lengths as well (since no. of features d = no. of frames * 39)
Initially we concatenated all features of all frames into a single vector, and patched extra zeroes to make lengths equal. It was obviously giving very bad results. We found that people tackle this problem by representing each of the coefficient of MFCC and each subband centroid by its mean and variance over all the frames, thus removing the different-audio-length effect, and reducing the feature length to 39*2 = 78. We added many other things as well like maxima, minima, etc and after various combinations our final representation of each mel coefficient and SSC included mean, variance, maximum, minimum, variance of derivative over all the frames and mean of first half frames, and mean of second half frames as well. Hence, finally we have 7*(13+26) = 273 features per utterance.

5 Classification:

For preparing our data for classification we subtracted the datapoints from mean and then scaled it to the unit variance (basic normalisation). For building our Classification-System, we tried the following multi-class classifiers:

5.1 K Nearest Neighbors (KNN):

This is the simplest and fastest Classifier that we used. The training was done via One-vs-All method on training data, and then we ran the classifier on testing data. Following is the variation of the accuracy wrt no. of neighbors:



5.2 Support Vector Machine (SVM):

In case of SVM, we tried various kernels available in the library like:

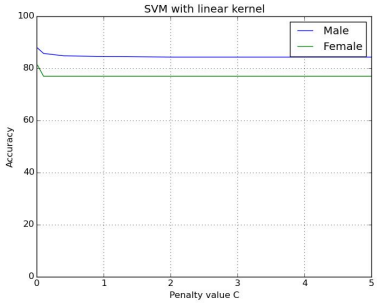
1. Linear Kernel

2. Polynomial Kernel

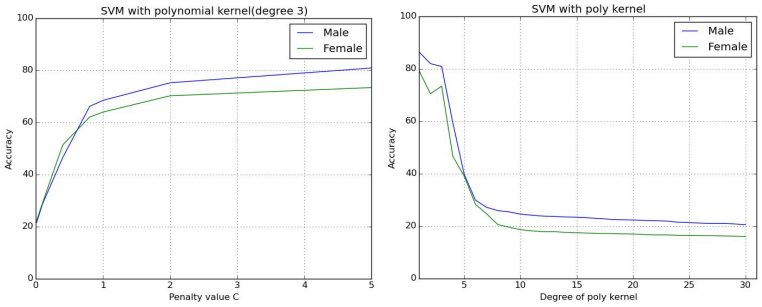
3. RBF kernel

The parameters of these kernels were optimised by observing the graphs of Accuracy vs Parameter for individual parameters, and then tweaking them a bit by hand.

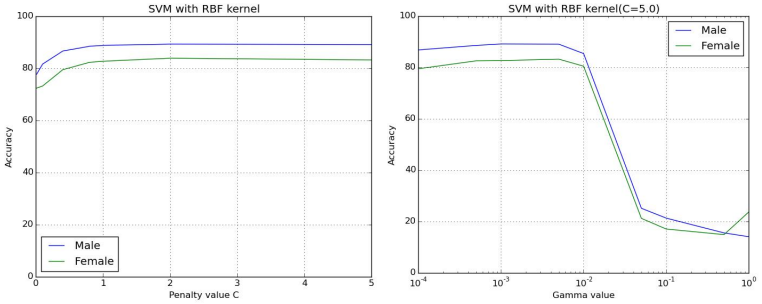
Variation of C for Linear Kernel:



Variation of C at degree=3, and of degree at C=5 are shown below:

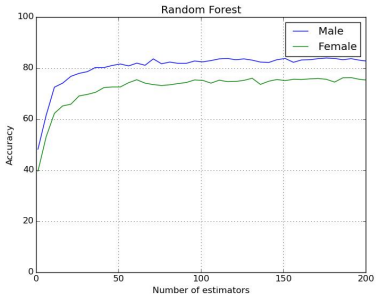


Variation of C at γ=auto and then variation of γ at C=5 are shown below:



5.3 Random Forests (RF):

Random forests are generated by bagging of features in datasets and training a decision tree on each bag. We vary the number of tree models to generate an accuracy distribution showing the variation of accuracy wrt no. of estimators (decision trees):



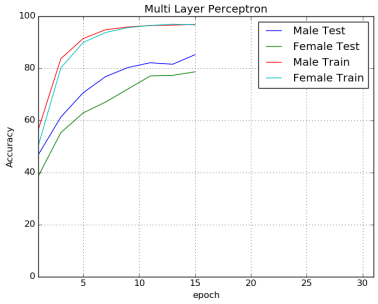
The accuracy scores quickly plateaus out at around 70 estimators.

5.4 Neural Networks:

The following two types of Neural Network Systems were used:

5.4.1 Simple Multilayer Perceptron:

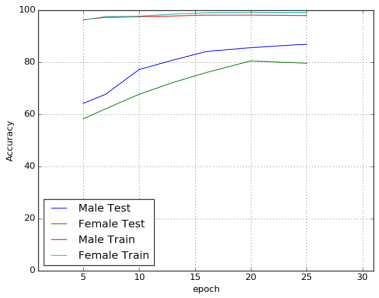
Made of three layers.Input linear layer, a hidden layer of 100 sigmoid neurons, and a softmaxLayer as output. We tried different epochs:



Finally we settled on using 50 epoch cycles and default values of momentum and weight decay.

5.4.2 Deep Artificial Neural Network:

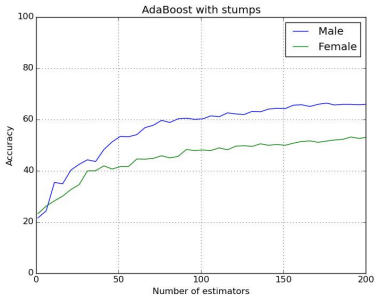
We tried to implement a deep neural network as well, using an additional layer of 100 Tanh neurons between input and sigmoid layer. The results were best in this combination only.



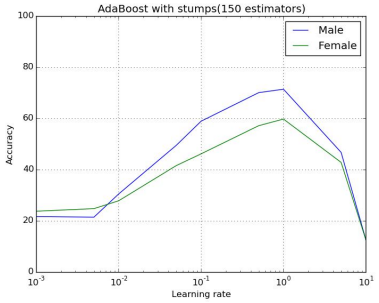
We observed that reducing the neurons or epoch cycles from 50 decreased the accuracy. Usually deeper ANN networks should outperform the simple single-hidden-layed MLP, but in our case it was always behind the best performance of MLP, even though we tried many different combinations of layers and neurons on this network. This must be because of overfitting on the training data.

5.5 AdaBoost with Stumps:

We attempt Adaptive Boosting on the dataset using weak learning decision tree stumps. To get an idea of the ideal number of estimators required, we plot the number of estimators vs accuracy graph, learning rate value being 0.02.



It is observed that the accuracy graph effectively plateaus out at 150 estimators. Now we vary the learning rate for this classifier. Variation of learning rate at 150 estimators, are given below:



As inferred, Adaboost of 150 weak learning decision stumps with a learning rate of 1.0, gives better accuracy than other adaboost classifiers.

6 Conclusion:

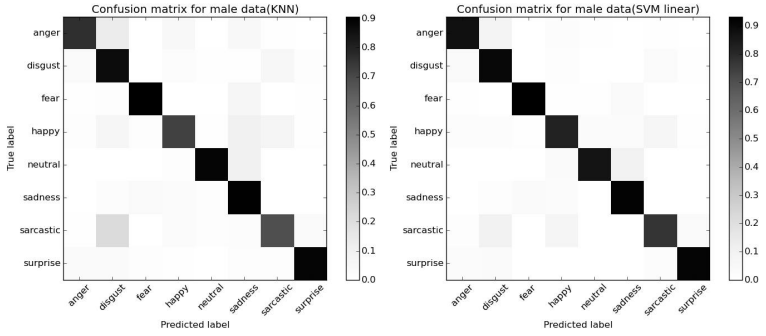
6.1 Summary of Results:

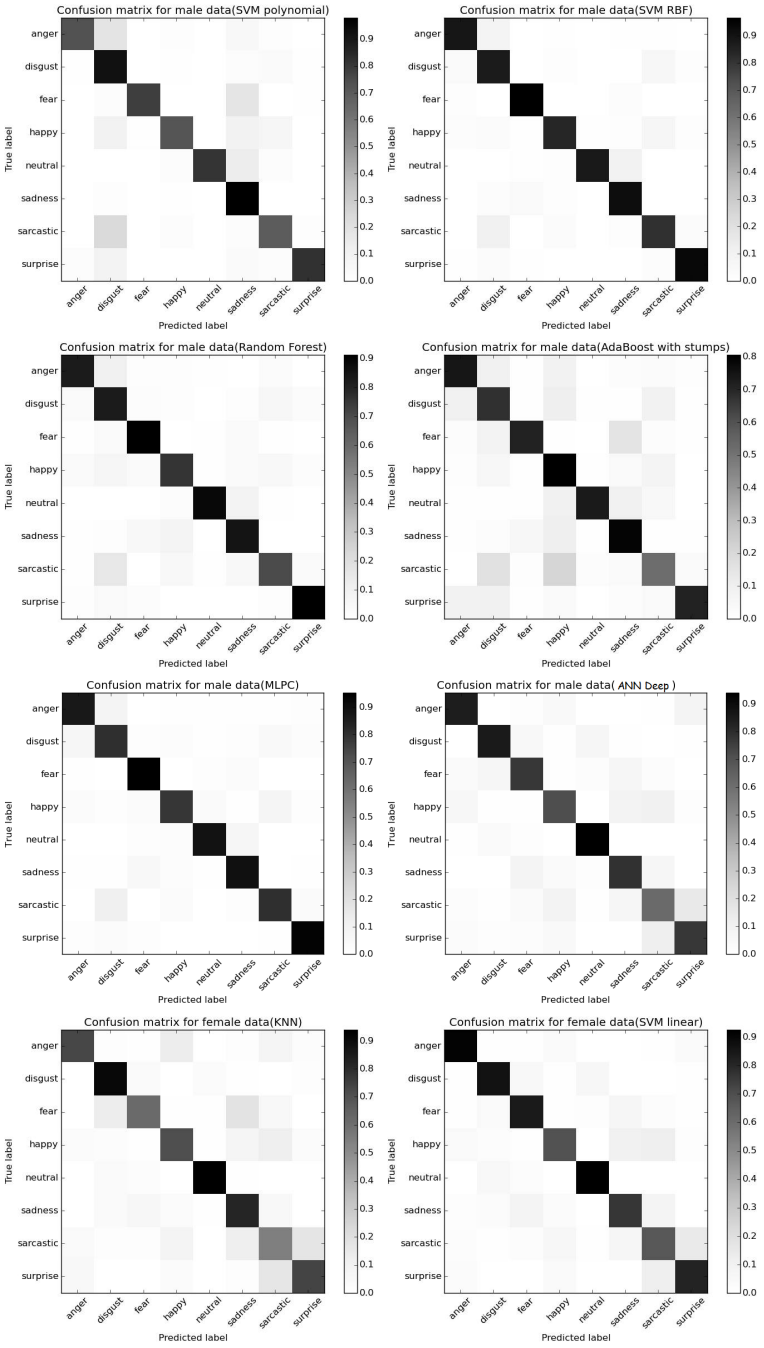
Following table represents the average Precision, average Recall and average F1-Score over all the classes for the best performance that we obtained, for each of the classifiers. Please note that average accuracy will be equal to average recall here, since class sizes are exactly same.

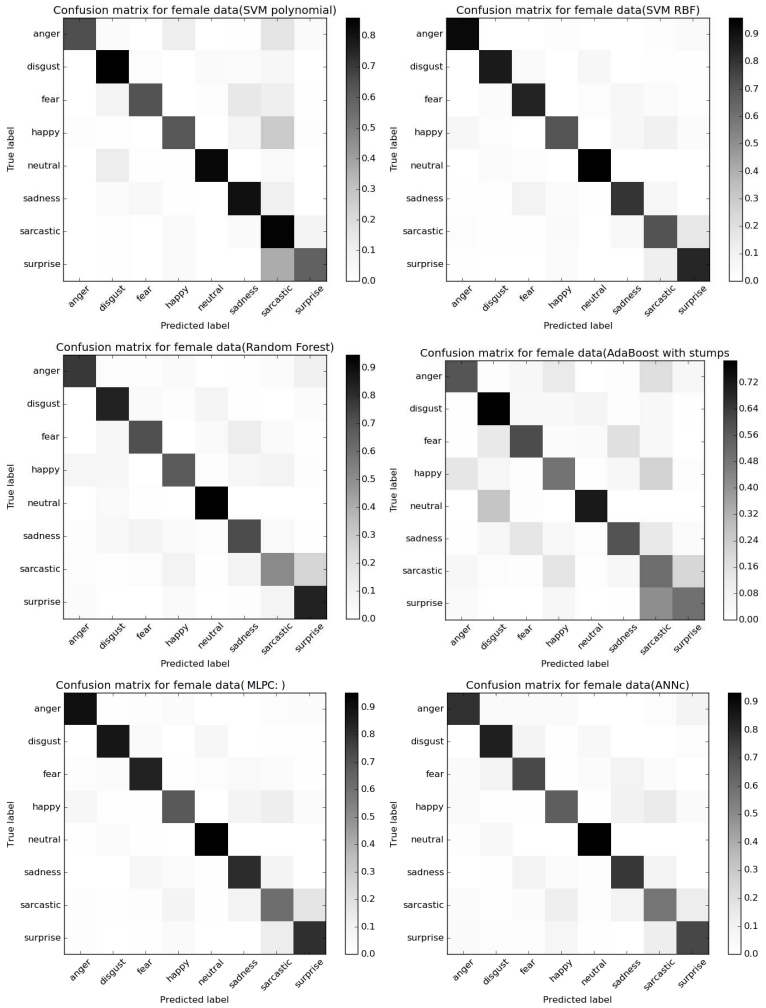
CLASSIFIER	avg recall		avg precision		F1-SCORE	
	M	F	M	F	M	F
KNN	82.75	75.33	85.9	79.05	82.4	74.45
SVM-Linear	87.75	81.17	89.56	83.74	87.8	80.46
SVM-Poly	80.83	73.33	87.28	82.1	81	73.06
SVM-RBF	**89.08**	**83.16**	**90.92**	**86.23**	**89.06**	**82.41**
Random Forests	83.42	75.25	86.98	79.41	83.47	74.14
Adaboost with Stumpts	71.33	59.67	77.64	65.51	71.34	59.18
MLP (in-100-out, 50 epochs	87	81.17	89.37	84.26	86.99	80.31
ANN	87.33	78.83	89.35	82.8	87.3	77.83

6.2 Confusion Matrices:

Following are the confusion matrices for the best performance of different classifiers of emotions on male and female speaker. They are intentionally simplified on a grayscale for the sake of simplicity. For detailed Confusion-Matrices, and Precision-Recall data of individual emotions, for each classifier, click here.




Confusion matrix for male data(SVM polynomial)
Confusion matrix for male data(SVM RBF)
Confusion matrix for male data(Random Forest)
Confusion matrix for male data(AdaBoost with stumps)
Confusion matrix for male data(MLPC)
Confusion matrix for male data(ANN Deep)
Confusion matrix for female data(KNN)
Confusion matrix for female data(SVM linear)
True label
Predicted label
anger
disgust
fear
happy
neutral
sadness
sarcastic
surprise



6.3 INFERENCE:

In our exploration we have found and reported that:

1. It is indeed possible for an AI system to recognize emotion from a spoken utterance, even if the system doesn't understand the meaning of the utterance at all.

2. Even basic classifiers are performing far better than the random guess, which in our case, is 100/8 = 12.5%.

3. Out of all the classifiers we tested the best performance was given by SVM with RBF Kernel. The details of this are given here:

<female>	anger	disgust	fear	happy	neutral	sadness	sarcastic	surprise
anger	0.93	0.00	0.00	0.01	0.00	0.00	0.01	0.05
disgust	0.00	0.87	0.05	0.00	0.07	0.00	0.01	0.01
fear	0.00	0.03	0.84	0.00	0.02	0.07	0.03	0.01
happy	0.07	0.02	0.00	0.71	0.00	0.07	0.11	0.03
neutral	0.00	0.03	0.01	0.00	0.96	0.00	0.00	0.00
sadness	0.01	0.01	0.09	0.03	0.01	0.80	0.06	0.00
sarcastic	0.02	0.01	0.01	0.04	0.00	0.05	0.71	0.16
surprise	0.01	0.00	0.00	0.03	0.00	0.00	0.13	0.83

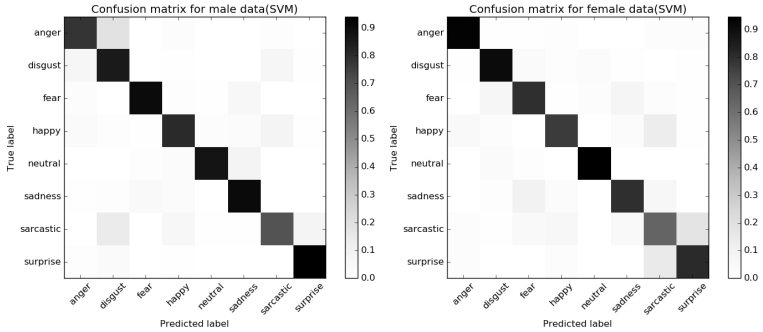
<male>	anger	disgust	fear	happy	neutral	sadness	sarcastic	surprise
anger	0.89	0.09	0.00	0.01	0.00	0.01	0.01	0.00
disgust	0.04	0.87	0.00	0.01	0.00	0.00	0.06	0.01
fear	0.01	0.00	0.97	0.00	0.00	0.02	0.00	0.00
happy	0.02	0.03	0.01	0.84	0.01	0.01	0.07	0.01
neutral	0.00	0.00	0.01	0.01	0.88	0.10	0.00	0.00
sadness	0.00	0.02	0.05	0.01	0.00	0.92	0.00	0.00
sarcastic	0.00	0.11	0.00	0.03	0.00	0.01	0.81	0.03
surprise	0.01	0.03	0.01	0.00	0.00	0.00	0.02	0.94

Emotion	Male			Female		
	recall	precision	f1-score	recall	precision	f1-score
anger	0,89	0,93	0,9	0,93	0,92	0,92
disgust	0,87	0,81	0,83	0,87	0,93	0,88
fear	0,97	0,95	0,95	0,84	0,88	0,84
happy	0,84	0,93	0,87	0,71	0,9	0,76
neutral	0,88	0,99	0,93	0,96	0,93	0,94
sadness	0,92	0,87	0,89	0,8	0,84	0,79
sarcastic	0,81	0,85	0,82	0,71	0,71	0,68
surprise	0,94	0,95	0,94	0,83	0,79	0,79

When we test our system by using cross-validation on sessions instead of sentences (i.e. train on 9 sessions and test on 1), we get the following results:

1. male accuracy (speaker 3): 84.42%
2. female accuracy (speaker 4): 82.58%

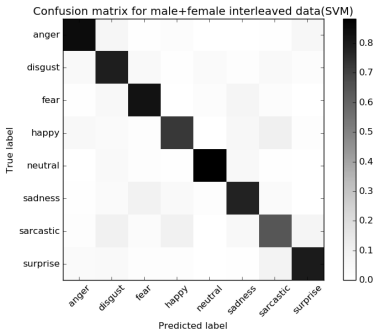
The IIT-KGP researchers, who created the same dataset, are getting an accuracy of 87.22% on speaker no. 7 (source). We couldn't test on the same speaker since we were granted data of only speakers 3 and 4, but even then the results seem comparable.



It should also be noted that our original method of cross-validation on sentences instead of sessions gives better accuracies:

1. male accuracy (speaker 3): 89.08%
2. female accuracy (speaker 4): 83.16%

Lastly we also interleaved male and female data to make one big dataset, and on applying the similar 15-fold cross-validation on SVM, we got average accuracy = 85.50%, without doing any additional tuning (details here.). Hence, if we are provided with enough speaker samples, our currently speaker dependent Emotion-Recognition system should be able to work globally, independent of the speaker, at least in Hindi language.



7 REFERENCES:

1. Advances in Multimedia Information Processing âĂŤ PCM 2002: Third IEEE ... - Google Books
2. IITKGP-SEHSC : Hindi speech corpus for emotion analysis Shashidhar G. Koolagudi, Ramu Reddy, Jainath Yadav, K. Sreenivasa Rao, IIT-KGP
3. <Python Speech features, explained and implemented, by James Lyons>
4. SPECTRAL SUBBAND CENTROID FEATURES FOR SPEECH RECOGNITION Kuldip K. Paliwal School of Microelectronic Engineering Griffith University Brisbane, QLD 4111, Australia
5. Emotion Recognition in Speech Using MFCC and Wavelet Features, K.V.Krishna Kishore, P. Krishna Satish, Computer Science and Engineering Vignan University
6. MFCC tutorial
7. Emotion and Gender Recognition of Speech Signals Using SVM S.Sravan Kumar, T.RangaBabu
8. Features and classifiers for emotion recognition from speech: a survey from 2000 to 2011 Christos-Nikolaos Anagnostopoulos Âů Theodoros Iliou Âů Ioannis Giannoukos