Effect of performance parameters of SVM and k-NN on speech recognition for articulatory Handicapped people

S. S. Bhabad, Associate Professor, Matroshri College of engineering and research Center, Nashik, India, ssb.eltx@gmail.com

Dr. Prof. G. K. Kharate, Principal, Matroshri College of engineering and research Center, Nashik, India, gkkharate@gmail.com

Abstract - Speech Recognition is the biggest challenge in case of disordered speech, because of unavailability and diversity of database. In this paper, we use MFCC as feature extraction method as they provides speech features similar to the way how human hears and perceives sounds. For decision of predicted word k-NN and SVM classifiers are used. The fundamental target of this paper is to discover the execution of k-NN and SVM classifier. Classifier performance evaluated on various parameters. The database consists of different samples includes zero to ten digits spoken by different speakers who suffer from different types of speech disorders. Experimental results show that k-NN gives highest prediction accuracy than SVM.

Keywords: k-Nearest neighbor, MFCC, SVM

I. INTRODUCTION

Following quite a while of innovative work, the accuracy of speech recognition for articulatory handicapped people stays one of the vital research challenges. The outline of speech recognition system requires needful considerations to the accompanying issues: Definition of different kinds of speech classes, speech representation, feature extraction methods, database and execution assessment. [1]

MFCC is perceptually motivated (near log f) frequency resolution, which uses a nonlinear frequency unit to simulate the human auditory system. Mel frequency scale is the most generally utilized component of the speech, with a simple calculation, good ability of the distinction, anti-noise and other advantages [2, 3].

Among all machine learning SVM is one of the more powerful technique that has been used for data classification. SVM is supervised learning technique in which previously training is done on some inputs, to classify data. SVM uses hyper plane i.e. hypothesis which used to separate the boundary points of the classes. These boundary points are used as support vectors. For data classification SVMs uses either linear or nonlinear hyperplanes [4].

K-nearest neighbor (K-NN) is task driven learning algorithm when used for classification or regression. The k nearest neighbor (k-NN) technique does not have training stage. First it calculates the distance between the test sample and all training samples to obtain its nearest neighbors and then classification is done [5]. K-NN classifier is one of the most broadly utilized classification algorithms for speech recognition because of their straightforwardness, discriminative nature and lack of training. K-nearest neighbor (k-NN) is non-parametric classifier; as there are no assumptions about the functional form of the problem being solved [2].

This paper is structured as; Section II illustrates the feature extraction using MFCC techniques. Section III highlights the SVM and k-NN classifier as speech prediction model. Section VI describes the preparation of database .In section V experimental results are carried out.

II. MEL FREQUENCY CEPSTRAL COEFFICIENTS AS FEATUR EXTRACTION TECHNIQUE

This paper uses the MFCC feature extraction technique based on the standard defined by the ETSI Aurora (ETSI 2000) group [6]. The method is illustrated in Fig 1.

MFCC coefficients are successful in audio classification as they have perceptually motivated decibel magnitude scale and their performance is reasonably well under robust conditions [6]. We have used frame size of length 25ms frame duration with overlap 10ms.

Steps to Calculate Mel-Frequency Cepstrum Coefficients

1) Frame the signal into short frames.
2) Apply these short frames as input for pre-emphasis. It is used to boost high frequency components that were suppressed during speech production. Pre-emphasis uses 1st order FIR high pass filter, whose transfer function is,

$$H(z) = 1 - a * Z^{-1} \text{……..…………. (1)}$$

Where, $0.95 \leq a \leq 1$

The value ’a’ depends on the nature of the medium or channel that will be used communicate the speech signal.

3) Windowing helps to reduce the effect of spectral leakage. Frequently hamming window is used.

DOI : 10.18231/2454-9150.2018.0511

Hamming window has almost zero values towards the both ends which confirm the continuity of the signal in successive frames.

Hamming window is represented as below,

$$W_{hm} = \left\{0.54 - 0.46\cos\left(\frac{2\pi(\mathrm{n}-1)}{\mathrm{M}-1}\right)\right\} .. 0 \le n \le M-1$$

……... (2)

Where, M is number of samples per frame

4) To get short term power spectrum Fast Fourier Transform (FFT) must be applied on windowed speech segments.

5) The Mel filter bank consists of triangular band pass filters. They are placed linearly up to 1000 Hz and above that placed non linearly. The short term power spectrum is filtered by this Mel Filter Bank. The linear frequency to Mel frequency related as follows,

$$m_f = 2595 log_{10}\left(1+\frac{f}{700}\right), m_f = 1,2 \dots\dots, L \dots\dots\dots\dots (3)$$

Where, L in number of filters used.

By taking logs of power at each of Mel frequencies we can adjust dynamic range in spectrum.

6) DCT is used to convert frequency domain coefficient into time domain. Due to presence of log term the coefficient are known as cepstral coefficients. DCT compressed the dimensions of the power spectrum. The output of DCT is known as MFCC. We have used 16 MFCC vectors.

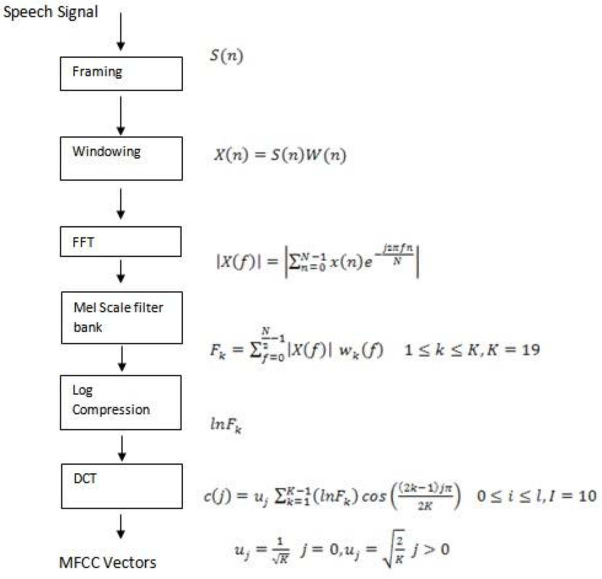
Speech Signal
Framing
$S(n)$
Windowing
$X(n) = S(n)W(n)$
FFT
$|X(f)| = \left|\sum_{n=0}^{N-1} x(n)e^{-\frac{j2\pi fn}{N}}\right|$
Mel Scale filter bank
$F_k = \sum_{f=0}^{\frac{N}{2}-1} |X(f)|\ w_k(f) \quad 1 \le k \le K, K = 19$
Log Compression
lnF_k
DCT
$c(j) = u_j \sum_{k=1}^{K-1}(lnF_k)\cos\left(\frac{(2k-1)j\pi}{2K}\right) \quad 0 \le i \le l, I = 10$
$u_j = \frac{1}{\sqrt{K}}\ j = 0, u_j = \sqrt{\frac{2}{K}}\ j > 0$
MFCC Vectors

Fig. 1 ETSI Aurora standard for computing MFCC vector

III. SPEECH PREDICTION MODEL USING SVM AND K-NEAREST NEIGHBOR (K-NN)

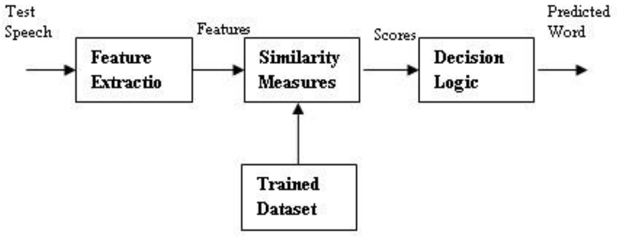

Fig. 2 Speech Prediction Model

Fig.2 shows speech prediction model used in this paper. SVM uses supervised learning technique to classify data. SVMs can be used for linear as well as non-linear classification. For linear SVMs feature vectors denoted by$S_i \in R^d, i = 1..M$, where M is a number of training samples, d is number of features of speech signal.

To predict correct speech, the speech unit classifies in the two classes $O_i = -1$ (Out of training dataset) or $O_i = +1$ (from training dataset) by hyperplane.

This hyperplane is defined as follows,

$$w\,.\,S + b = 0 \dots\dots\dots\dots\dots\dots\dots\dots (4)$$

Where w is normal to the plane and, b is bias.

To find the boundaries of both classes, SVMs construct the hyper plane using following equations

$$w\,.\,S_i + b \ge +1\ for\ O_i = +1 \dots. (5)$$

$$w\,.\,S_i + b \le -1\ for\ O_i = -1 \dots.. (6)$$

Maximizing the margin with constraints in (5) (6), finds the best hyperplane.

In case of Non-linear classification the SVMs are directly applied to a higher dimensional feature space S_i instead of input space R^d.

$$\Phi : R^d \rightarrow S_i \dots\dots\dots\dots\dots\dots (7)$$

This transformation is implemented using different types of kernel functions.

Linear Kernel: $K(S_i, S_j) = S_i^T S_j$

Radial Bases Kernel: $K(S_i, S_j) = e^{-\left(\frac{\|S_i - S_j\|^2}{2\sigma^2}\right)}$

Polynomial Kernel: $K(S_i, S_j) = (S_i\,.\,S_j + a)^b$

Sigmoidal Kernel: $K(S_i, S_j) = \tanh(aS_i . S_j - b)$

Where, a and b are Kernel's parameters [4, 5].

Different training observations are stored and utilized for finding the distance between unknown samples.

The training stage of k-NN is quick and it stores all the training information will be utilized as a part of the testing stage. The prediction of unknown sample is depends on the complete training data set [7, 8].

To find out nearest neighbor in k-NN, the distance between the unknown sample and all training samples must be calculated.

The k-NN classification problem is defined as follows:

- Consider train data which is set of data feature vectors and class labels,

$$\mathrm{Train\_}Data = \{[\underline{a}(1), \underline{b}(1)], [\underline{a}(2), \underline{b}(2)], \dots\dots\dots\dots\dots [\underline{a}(n), \underline{b}(n)]\},$$

Where a(j) denoted as j^{th} data feature vector

- a(j) Represented as j^{th} row of a $n \times c$ matrix, where c denoted as MFCC coefficients.
- b(j) denoted as class label of the j^{th} feature vector
- We may have different class values such as b=1, b=2…
- Consider $\underline{z}$ be a unknown feature vector.
- For this unknown feature vector $\underline{z}$ we have to find class label.
- Search $Train\_Data$ for the closest feature vector to $\underline{z}$
- let this "closest feature vector" be $\underline{a}(j)$
- Classify $\underline{z}$ with the same label as, i.e. $\underline{z}$ is assigned label
- To search nearest neighbor to $\underline{z}$ from $Train\_Data$

DOI : 10.18231/2454-9150.2018.0511

- Arrange the feature vectors as per the different distance measures in ascending order. Assign k vectors which gives closest distance to $\underline{z}$
- Prediction

Arrangement of k feature vector provides set of class label. Select the most common class label from the set ("vote")

So, the class of $\underline{z}$ is predicted accordingly.

The value of k can be found out experimentally. k-NN approach is the most suitable classifier for infinite amount of training data [8].

- Minkowski distance

$$d_{st} = \sqrt[p]{\sum_{j=1}^{n} |X_{sj} - y_{tj}|^{p}} \ldots\ldots\ldots\ldots\ldots..(8)$$

X is trained data matrix represented by $[x_1, x_2, \ldots.x_n]$.
y is test data matrix represented by $[y_1, y_2 \ldots.y_m]$

The distances between the vector x_s and y_t can be calculated as follows:

The special case of Minkowski distance,

1. For city block distance the value of p = 1,
2. For Euclidean distance the value of p = 2,
3. For Chebychev distance the value of p = ∞,

- Cosine Distance

$$d_{st} = 1 - \frac{x_s y'_t}{\sqrt{(x_s x'_s)(y_t y'_t)}} \ldots\ldots\ldots\ldots\ldots\ldots.(9)$$

- Standardized Euclidean Distance

$$d_{st}^2 = (x_s - y_t)U^{-1}(x_s - y_t) \ldots\ldots\ldots\ldots\ldots\ldots..(10)$$

Where, U is denoted as $l \times l$ diagonal matrix, $(P\ \ (j))^{2}$ has j^{th}diagonal element and for each dimension P is vector of scaling factors.

IV. DATABASE CREATION

Database consists of set of 0-10 digits spoken from different articulatory handicapped people. Each digit was recorded for number of times by each speaker in noise proof room. The RODE NT1 MIC microphone and NUENDO 4 software was used for recording the data. Nuendo uses a system of input and output buses to transfer audio between the program and the audio hardware. The speech files are converted into .wav file using sampling frequency 44100Hz. Total 1100 different speech samples are recorded, out of that 80% used for training stage and 20% used for testing stage. MATLAB 2017 b software is used to carry out these experiments.

V. RESULTS AND DISCUSSION

We have tried the algorithm for speech recognition of eleven words (i.e. zero to ten). The experimentation is finished with various sorts of 990 training samples and 220 test samples. These 1100 samples are spoken by different articulatory handicapped peoples. The accuracy acquired in various cases as takes after:

A. Support Vector Machine(SVM)

For different kernel function the accuracy of predicted word is calculated (for different number of features selected) for all digits as shown in Table 1.

The features are selected in sequential forward sequence (SFS) way. The experimental results show that the medium Gaussian SVM using 16 number of features gives best accuracy as compare to other.

Table 1 A Effect of kernels in SVM on average accuracy for All Digits

Sr. No	No. of Features	Accuracy in %						
		8	10	12	14	16	18	20
1	Linear SVM	64.09	64.09	72.27	74.09	75	74.45	75
2	Quadratic SVM	72.27	78.64	87.27	85.46	85.91	89.09	90.91
3	Cubic SVM	79.09	30.46	85.45	85.91	85.45	88.64	90
4	Fine Gaussian SVM	75.08	84.09	83.18	81.36	79.09	73.18	65
5	Medium Gaussian SVM	76.82	80	85.45	89.09	90.91	90.91	88.18
6	Coarse SVM	57.27	59.09	59.55	61.36	65	67.27	66.8

Table 2 shows the effect of validation on average accuracy for all digits for different types of kernels used in SVM. The 16 features are used. The experimental results show that the accuracy of classifier is not affected by validation process except Quadratic SVM. Cubic and medium Gaussian SVM gives better accuracy.

Table 2 A Effect of validation in SVM on average accuracy for All Digits (No. of features=16)

Sr. No	Type of SVM	Accuracy in %		
		No validation	5 fold cross validation	10 fold cross validation
1	Linear SVM	75	75	75
2	Quadratic SVM	85.91	88.64	88.64
3	Cubic SVM	90	90	90
4	Fine Gaussian SVM	65	65	65
5	Medium Gaussian SVM	90.91	90.91	90.91
6	Coarse SVM	66.82	66.82	66.82

B.k Nearest Neighbor Algorithm

As we can see in Table 3 the value of k increases, the average accuracy of predicted word decreases.

Euclidean distance measure gives better average accuracy for all values of k than other distance measures. However Hamming and Jaccard distance measures performs the worst for disorder speech database.

Table 3 A Effect of k-value on average accuracy for All Digits (No. of features=16)

Sr. No.	Distance Measures	% of Average Accuracy		
		k=1	k=3	k=3
1	Euclidean	92.73	92.27	91.82
2	Jaccard	12.73	10.45	9.55
3	Cityblock	92.18	91.06	91.64
4	Seuclidean	92.18	92.27	91.82
5	Hamming	12.73	10.45	9.55
6	Chebychev	90	87.72	85.45
7	Cosine	92	92.09	92.13
8	Mahalanobis	91.82	92.27	90.45
9	Minkowski	90.18	92.27	91.82
10	correlation	91.82	90.73	90.45

DOI : 10.18231/2454-9150.2018.0511

Table 4 A Effect of different types of k-NN on average accuracy for All Digits

Sr. No	No. of Features	Accuracy in %						
		8	10	12	14	16	18	20
1	Fine KNN	81.82	87.72	90.91	88.18	92.73	92.73	91.82
2	Medium KNN	76.36	80	83.64	85	84.09	87.27	90.45
3	Coarse KNN	50	50.45	49.09	52.27	49.09	52.27	54.09
4	Cosine KNN	72.27	77.27	81.36	81.82	83.64	87.72	86.81
5	Cubic KNN	74.55	77.73	83.18	85	84.09	85.91	86.36
6	Weighted KNN	80.91	85.91	88.64	86.82	90	91.82	90.9

Table 5 shows the effect of validation on average accuracy for all digits for different types of k-NN. The 16 features are used. The experimental results show that the accuracy of classifier is not affected by validation process. Fine and weighted k-NN gives better accuracy.

Table 5 A Effect of validation in k-NN on average accuracy for All Digits (No. of features=16 and Euclidean distance measure)

Sr. No	Type of KNN	Accuracy in %		
		No validation	5 fold cross validation	10 fold cross validation
1	Fine KNN	92.73	92.73	92.73
2	Medium KNN	90.45	90.45	90.45
3	Coarse KNN	54.09	54.09	54.09
4	Cosine KNN	86.82	86.82	86.82
5	Cubic KNN	86.36	86.36	86.36
6	Weighted KNN	90.91	90.91	90.91

Table 6 Overall performance of classifier on average accuracy for All Digits

Sr. No.	classifier	No. of features used	Accuracy in %
1	Fine KNN	16	92.73
2	Medium Gaussian SVM	16	90.91

Table 6 shows that overall performance of k-NN classifier is better than SVM.

Performance comparison of kernel method Vs. Statistical method

In case of kernel method (SVM) the factors of one class are learn on the sample of all classes. However, in statistical method (k-NN) the factors of one class are assessed from the samples of its own class only. We can compare the performance of two classifier with reference to following points[9].

1. Learning complexity
The performance of k-NN classifier is depends on the distance measure. The learning time is linear with the number of samples. But SVMs are learned by quadratic procedure so the learning time is proportional to the square number of samples.

2. Recognition Accuracy
For large data kernel method works well as compare to statistical method. If the nature of dataset is belongs to same category then statistical method gives better accuracy.

3. Memory requirement and operational complexity
k-NN classifier requires less memory than SVM because k-NN needs less parameter for performance.

4. Learning flexibility
The Learning flexibility of k-NN classifier is better than SVM as in case of k-NN new class can be easily added to an existing classifier because the learning time of k-NN is linear and for SVM it is proportional to the square of the number of classes and new samples or new class needs relearning.

VI. CONCLUSION

The experimental results shows that for k-NN classification the performance of classifier is depend upon selection of the distance measures and k value. The value of k can be selected experimentally. Euclidean distance measures give highest accuracy of 92.73% for Fine k-NN. It shows that SVM with medium Gaussian SVM gives 90.91% of accuracy. The experimentation shows that k-NN is more suitable classifier than SVM as it requires less training storage and computational complexity.

This work in future can be extended to correct the recognized word by separating the phonemes of word using different techniques.

REFERENCES

1. S. S. Bhabad , G. K. Kharate, "An Overview of Technical Progress in Speech Recognition", International Journal of Advanced Research in Computer Science and Software Engineering, , March 2013, Volume 3, Issue 3, ISSN: 2277 128X

2. LiYu Hu , Min Wei Huang, Shih Wen Ke, ChihFong Tsai "The distance function effect on k nearest neighbor classification for medical datasets", Hu et al. SpringerPlus (2016), 5:1304 DOI 10.1186/s40064-016-2941-7

3. Han Y, Wang G, Yang Y, "Speech emotion recognition based on MFCC", Journal of Chong Qing University of Posts and Telecommunications (Natural Science Edition), ,2008, 20(5).

4. Kamil Aida-zade , Anar Xocayev, Samir Rustamov , "Speech Recognition using Support Vector Machines", 2016 IEEE, 2472-8586

5. Kamil Aida-zade, Anar Xocayev, Samir Rustamov "Speech Recognition using Support Vector Machines", Baku, Azerbaijan volume I, 2012, pp. 213-217.

6. Aamir Khan, Muhammad Farhan, Asar Ali, "Speech Recognition: Increasing Efficiency of Support Vector Machines", International Journal of Computer Applications, December 2011, Volume 35– No.7 (0975 – 8887).

7. Ben Milner, Xu Shao , "Clean speech reconstruction from MFCC vectors and fundamental frequency using an integrated front-end", Speech Communication Volume- 48 (2006), 697–715.

8. Muhammad Rizwan , David V. Anderson , "Using k-Nearest Neighbor and Speaker Ranking for Phoneme Prediction", , 2014 IEEE, 978-1-4799-7415-3/14

9. Mohanty S., H. N. sbebartta, Performance Comparison of SVM and K-NN for Oriya Character Recognition, International Journal of Advanced Computer Science and Applications, Special Issue on Image Processing and Analysis, 2011, pp-112-116

DOI : 10.18231/2454-9150.2018.0511