Survey on Phoneme Recognition using Support Vector Machine

[1] Fathima Nazarath P. A
[1]Department of Electronics and Communication Engineering
[1]Adi Shankara Institute of Engineering and Technology, Kalady, India

Abstract

Automatic Speech Recognition (ASR) is a process in which speech signal is converted into a sequence of words, other linguistic units by making use of an algorithm which is implemented as a computer program. The speech recognition system would support many valuable applications that require human interaction with machine. The major objective with which ASR works is the development of the techniques and a system that enables the computers to recognize speech as input. Most precisely speech recognition means phoneme recognition. Good phonetic decoding leads to good word decoding, and the ability to recognize the English phones accurately will undoubtedly provide the basis for an accurate word recognizer. In this work, a detailed survey on a classification technique called support vector machine (SVM) is carried out. Linear SVM are mainly used where linear separation between two classes is possible and in nonlinear SVM's kernel functions are used for conversion purpose. Among two types of SVM classifier soft margin SVM is used to improve the performance when error occurs and least square SVM is used for large scale classification.

Keyword- Phoneme Recognition, Mel Frequency Cepstral Coefficient (MFCC), Support Vector Machine (SVM)

I. INTRODUCTION

Automatic Speech Recognition (ASR) also known as computer speech recognition is a process in which speech signal is converted into a sequence of words, other linguistic units by making use of an algorithm which is implemented as a computer program. The major objective with which ASR works is the development of the techniques and a system that enables the computers to recognize speech as input. Speaking/communicating directly with the machine to achieve desired objectives make usage of modern devices easier and convenient. The goal of automatic speech recognition is for a machine to be able to hear, understand, and act upon spoken information, that is to analyze, extract characterize and recognize information about the speaker identity. The process of speech recognition can be divided into the following consecutive steps.[5]

- Pre-processing
- Feature extraction
- Decoding
- Post-processing

Machine transcription of fluent speech so-called 'phonetic typewriter' remains a challenging research goal. However, automatic speech recognition (ASR) has reached the point where, given some restrictions on speakers, vocabulary and environmental noise, reliable recognition of isolated words or short phrases is possible. Speech is composed of certain distinct sounds. Phoneme is a contrastive unit in the sound system a particular language that helps us distinguish between meanings of words from a set of similar sounds corresponding to it pronounced in one or more ways. For each language, there is a specific set of phonemes. There are several notations how to transcribe these phonemes. The most notable is the International Phonetic Alphabet (IPA) which was developed by the International Phonetic Association beginning in 1888 with the goal of describing the sounds of all human languages.[3]

A. Phoneme Recognition

The microphone will pick up the speech signal which is an analog signal and convert it into electrical signal where the signal is discretized with a sampling frequency. The total frequency range of human speech is typically limited to the 5Hz to 3.7 kHz range, and has shown in research to be sufficient for speech recognition applications. It is possible to remove frequencies below 100 Hz with a high-pass filter. As speech recognition systems will classify any sound to any phoneme with some probability, background noise can cause insertions of phonemes or words into the recognition result, if the noise resembles the parameters of a phoneme model better than those of a silence model. Such insertions can be reduced by removing areas from the speech signal between the start of the recording and the point of time when the user starts to speak, and after the end of the utterance.

Phoneme recognition consists of two parts, training and recognition. In training mode, many examples of each class or phoneme is passed to a feature extraction stage in which the features from them are extracted. Feature extraction technique is used to convert these features into feature vectors. Based on these feature vector the classifier classify the trained data and each one of

the resulting class are used to build a model for that particular class and these models are subsequently stored. In recognition mode, feature extraction is carried out for the testing speech and feature vectors are obtained which lead to a particular pattern. This pattern of unknown class is compared with each model and classified according to the model to which it is 'closest'. When performing recognition of phoneme, each different phoneme is regarded as a class. Figure below shows schematic outline of phoneme recognizer. Each step is described below in detail.

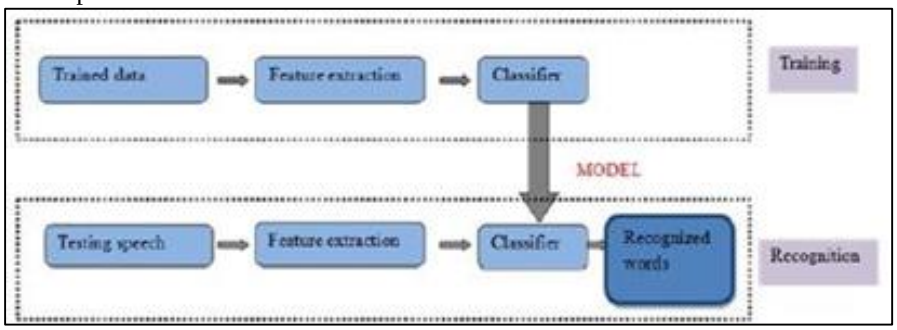


Fig. 1: Stages of phoneme recognition

B. Feature Extraction

Feature extraction is the main part of the speech recognition system. It is considered as the heart of the system. It is a process where speech signal is converted into sequence of feature vectors coefficients which contains only necessary information required for speech recognition. Feature extraction compresses the magnitude of the input signal (vector) without causing any harm to the power of speech signal. To calculate features, acoustic observations are extracted over time frames of uniform length. Within these frames, the speech signal is assumed to be stationary. The length of these frames is typically around 25 ms, for the acoustic samples in this window one multidimensional feature vector is calculated. The time frames are overlapping and shifted by typically 10 ms. on the time window, a fast Fourier transformation is performed, moving into the spectral domain.

Most commonly used technique is MFCC, the mel frequency cepstral coefficients, which is a spectral analysis technique. The MFCC tries to mimic the human ear, where frequencies are non-linearly resolved across the audio spectrum. Hence, the purpose of the Mel filters is to distort the frequency such that it obey the spatial relationship of the hair cell distribution of the human ear. Hence, the mel frequency scale corresponds to a linear scale below 1 kHz, and algorithmic scale above the 1 kHz. MFCC methodology is based on the short-term analysis, where feature vector is computed from each frame separately. The coefficients are extracted by taking speech sample as an input and then hamming window is applied to reduce the discontinuities of a signal. Then the Mel filter bank is generated by applying FFT. Power spectrum is computed by performing Fourier analysis. According to Mel frequency warping, as width of the triangular filters differs therefore log total energy in a critical band around the center frequency is combined. The numbers of coefficients are then obtained after warping. In last, the Inverse Discrete Fourier Transformer is applied for the calculation of cepstral coefficients. It transforms the log of the domain coefficients to the frequency domain where N is the length of the FFT.

C. Dimensionality Reduction

To achieve a high degree of accuracy in phoneme recognition, it is important to choose a feature space in which individual phonemes are well separated and easily distinguished from one another. A large number of features extracted from speech signals have been often used in phoneme recognition in the past, with Mel frequency cepstral coefficients (MFCC). When we use so many feature parameters for training and testing a classifier, the well-known curse of dimensionality often emerges.

After feature extraction, it is desirable to transform the features into a relatively low dimensional feature space while preserving information relevant to the phoneme recognition task. This process, namely dimensionality reduction, aims to produce features which are concise low dimensional representations that retain the most discriminating information for the intended application. Dimensionality reduction also decreases the computational cost associated with subsequent processing. These methods can be categorized as linear or nonlinear.

Linear Methods

– Principal Component Analysis (PCA)
– Non Linear Methods

Locally Linear Embedding (LLA)

Isomap

1) Principal components analysis (PCA): constructs a low-dimensional representation of the data that describes the variance in the data as possible. PCA is by computing the eigen values and eigenvectors of the covariance matrix of the dataset and eigenvectors with largest eigen values represent the dimensions that have the strongest correlation in the dataset.
2) Locally linear embedding (LLE): is an unsupervised learning algorithm that computes low dimensional embedding's of high dimensional data. The principle of LLE is that nearby points in the high dimensional space remain nearby and similarly colocated with respect to one another in the low dimensional space. In other words, the embedding is optimized to preserve local neighbourhoods.

3) Isomap: A nonlinear generalization of multidimensional scaling that seeks a mapping from a high dimensional dataset to a low dimensional dataset that preserves geodesic distances between pairs of data points. In particular, Isomap preserve the global geometric properties of the manifold while LLE attempts to preserve the local geometric properties of the manifold.

D. Classification

The speech recognition problems are kind of pattern recognition problems. The speech recognition problems are non-linear pattern classification problems. The SVMs are based on the Structural Risk Minimization (SRM) is an inductive principle for model selection used for learning from finite training data sets to minimize the risk functional with respect to both terms, the empirical risk, and the confidence interval. The SRM describes a general model of capacity control and provides a trade-off between hypothesis space complexity and the quality of fitting the training data. The SVMs are also called maximum margin classifier as they satisfy the property that, they simultaneously minimize the empirical classification error and maximize the geometric margin. While designing the SVMs input data points dimensions should be equal, the chosen kernel function incorporates all the prior knowledge about the data. In SVM, original input space is mapped into a higher dimensional feature space in which an optimal separating hyper plane is constructed on the basis of SRM to maximize the margin between two classes which maximizes the generalization ability of the system. The original optimal hyperplane algorithm proposed by Vapnik based on linear classifier. The kernel functions can be applied to find maximum-margin hyper plane for non-linear classifiers. The key features of SVMs are the use of kernels, the absence of local minima, the sparseness of the solution and the capacity control obtained by optimizing the margin. In this case, the nonlinear classifiers are constructed by replacing every dot product to kernel function.

II. SUPPORT VECTOR MACHINE

The underlying concept behind an SVM is structural risk minimization. A learning machine is chosen that minimizes the upper bound on the risk (or test error), which is a good measure of the generalizability of the machine. This is estimated as the ratio of misclassified vectors over the total number of training vectors when using a leave-one out method [6]. It can be shown that this is equal to the ratio of expected number of support vectors to the total number of training vectors.

SVM is a binary classifier. i.e it divides the feature vectors into two class by separating them with a margin. Now the problem is that among large separation between two classes defining feature vectors how we will select the best margin? (Figure 3). For obtaining the solution consider two classes with two predictors H1 and H2. Suppose that the two classes are linearly separable. Then a natural approach is to find the straight line that gives the biggest separation between the classes.

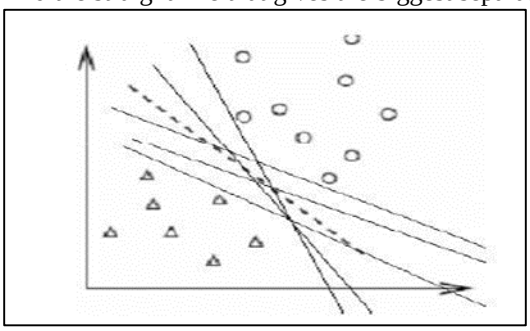


Fig. 3: Choosing Best Separator

A. Linear SVM

Let x , i = 1, 2, . . . ,N, be the feature vectors of the training set,X. Our goal is to design a hyperplane.

$g(x)= w^T x + b = 0$

Choose a hyperplane which give more space on either side.So that it provide generalization property. Every hyperplane is characterized by its direction (w) and its exact position in space (b).Goal is to search for the direction that gives the maximum possible margin. Where margin is the sum of perpendicular distance to the closest sample on either side of the plane.

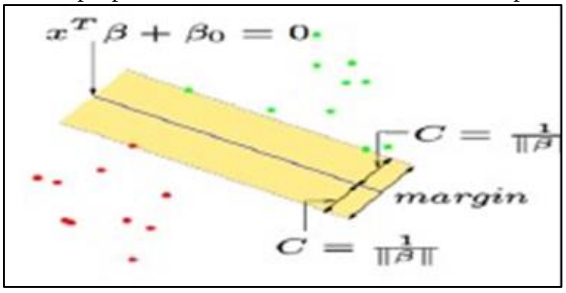


Fig. 4: Classification of two linear separable classes

Distance of a point from a hyperplane is given by

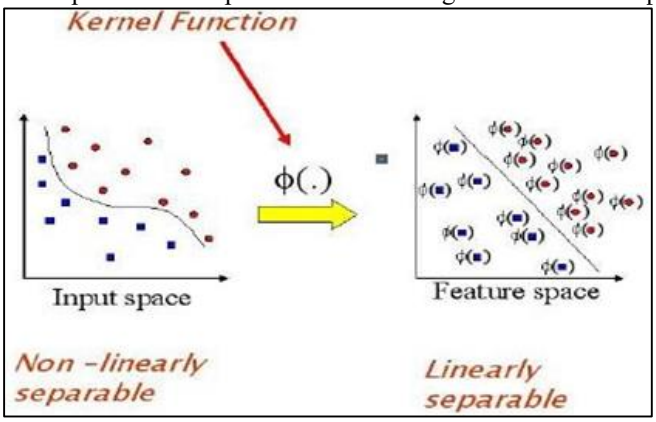
$$Z= \frac{\|g(x)\|}{\|w\|}$$

Scale w, b so that the value of g(x), at the nearest points in H1 and H2 is equal to 1 for H1 and -1 for H2. Thus equation becomes
x_i.w+b ≥ +1 for H1
x_i..w+b ≤ -1 for H2
These equations can be combined into:
Y(x_i.w+b) – 1 ≥ 0 for all i
The hyperplane's equidistance from H1 and H2 means the SVM's margin. In order to orientate the hyperplane to be as far from the Support Vectors as possible, we need to maximize this margin. Simple vector geometry shows that the margin is equal to $\frac{1}{\|w\|}$ and maximizing it subject to the above constraint is equivalent to finding min. $\|w\|$. Minimizing $\|w\|$ is equivalent to minimizing $\frac{1}{2}\|w\|^2$. To solve the above problem use Lagrangier multiplier resulting following inferences.

- Any data point satisfying α_i. y_i = 0 which is a Support Vector x_s will have the form: y_s (x_s.w + b) = 1
- Substituting value of w in above eqn: y ($\sum x_m y_m \alpha_m . x_s$ + b) = 1 □ multiplying through by y_s and then using $y_s{}^2$ = 1, b becomes: b = y_s - $\sum x_m y_m \alpha_m . x$

B. Non Linear SVM

If the data are not linearly separable it is possible to map the data's into a higher-dimensional space.



SVM transform n-dimensional feature vector x in to an N dimensional feature vectors φ:^{N}-- R^N. This transformation is realized via Kernel functions. The linear classifier relies on dot product between vectors K $(x_i; x_j) = x_i{}^T . x_j$.During transformation, the dot product becomes: K (x_i; x_j) = φ(x_i)^T.$\varphi(x_j)$.A kernel function is some function that corresponds to an inner product in some expanded feature space . There are different types of Kernel functions used in SVM.

C. Types of SVM

Mainly SVM is classi_ed as follows

- Hard margin SVM classifier: linear SVM classifiers.
- Soft margin SVM classifier: used when the data's are not linearly separable and having errors. If the samples are not completely separable linearly, positive weakness variables are used to solve the problem. To produce solution for every possible situation within the training data, an upper-limit value (C) is added to the system. C parameter is considered as Lagrange multipliers within the range, 0≤ α_i ≤ C. The Lagrangian function is solved using the KKT conditions and the duality of the primal problem is obtained. The obtained dual problem is solved by quadratic programming and it's α, w and b are calculated.
- Least square SVM classifier: used for large scale classification problems. An insensitive loss function is employed with the function estimation of support vector method. Equality constraints are used instead of inequalities in problem formulation.

D. Multiclass SVM

Phoneme recognition is a multiclass problem, since we have to classify n phonemes into n classes. For the construction of n-class classifier using binary classifier, we use

- One-versus-rest approach- Train 'k' binary classifier that compares each class against the rest. Consider a set of M two-class problems. For each classes g_i(x) is designed, where i = 1,2,….M so that g_i(x) > g_j(x),for i ≠ j ,if x € w_i.An optimal hyperplane is designed, i.e g_i(x) = 0 is the separating class w_i. from all the others. Thus, each classifier is designed to give for g_i(x) > 0 for x € w_i and $g(x)$ < 0. Otherwise classification is achieved by using following rule: Assign x in w ,if i= arg m$ax_k g_k$ (x).

– One-versus-one approach: Train $\frac{k(k-1)}{2}$ binary classifier each of them comparing two classes. It evaluates all possible pairwise classifiers. Applying each classifier to a test example would give one vote to the winning class and a test example is labeled to the class with the most votes.

III. PHONEME RECOGNITION USING SVM

SVM classifier was used as phoneme classifier to train and test low dimensional feature data. For that the feature vectors obtained through feature extraction technique i.e MFCC which are high dimensional vectors are transformed into low dimensional feature vector through dimensionality reduction. Dimensionality reduction was conducted on both the primitive training data and testing data to produce the corresponding low dimensional features data space. The consideration of a small feature vector is highly desirable, since this helps in reducing the complexity of the classifier stage and improves also the speech performance.

SVM classifier demonstrated the highest accuracy when compared to other classifiers. The observed superiority of the SVM algorithm was expected, since it does not require discretized attributes. Except this, SVM perform well in higher dimensional spaces since they do not suffer from the curse of dimensionality. Moreover, SVMs have the advantage over other approaches, that their training always reaches a global minimum. As it is wellknown some phonemes, which share a similar manner of articulation and therefore possess kindred acoustic features, areoften misclassified. This results to pairs as well as groups of phonemes that could be confused during the recognition process. The correlation among the phonemes can be seen from the confusion matrix. Using LS-SVM overcome this problem during classification because it offers high generalization property. To increase the phoneme recognition performance and facilitate the speech and language recognition components, Least Squares version (LS-SVM) is since it can be used in large scale data to obtain N-best list of candidate phonemes with highest accuracy.[12]

Since the task of speech recognition is a multiclass problem, multiclass SVM methods are required. When constructing and evaluating a multiclass method, it is easy to forget the importance of the underlying binary SVMs. This must be not be done given that the generalization ability of the multi-class classifier is fully dependent on the generalization abilities of the binary SVMs. An analysis of the individual binary SVMs should be carried out before the extension to the multi-class case is done.[11]

– One-against-all: A classifier for each phoneme is constructed, where each classifier constructs a hyperplane between the corresponding phoneme and the other phonemes.
– One-against-one: A classifier is designed for each possible pair of phonemes, thus separating each phoneme from the others. The majority voting scheme is then used in order to classify a phoneme.

The choice of a suitable multiclass SVM method heavily depends on the specific characteristics of the problem at hand. Both the size of the database and the complexity of the speech recognition task addressed in this work advise using one-against-one multiclass classifier for phoneme recognition since we are dealing with large training datasets.[13] Although the one-versus-one method must train more binary classifiers than the other approaches, each classifier is trained with a smaller fraction of the database. Each binary classifier in the one-versus-one approach deals with a more simple, balanced and easily separable problem. Finally, the reduction of the whole multiclass problem into smaller binary classification tasks allows for the use of larger training datasets, which provide more varied acoustical information for the speech recognition task.[4] Thus we can conclude that SVM will be a good choice for classification in phoneme recognition since it is the simplest and most effective classifier with less complexity and great accuracy.

IV. CONCLUSION

In this study it was able to know more about support vector machine as a linear binary classifier and nonlinear binary classifier. For improving the accuracy of a phoneme recognizer, dimensionality reduction was conducted on both the primitive training data and testing data to produce the corresponding low dimensional features data space for that studied about different manifold learning algorithms including Isomap, LLE and a linear reduction technique called PCA. It was able to identify the need of different kernel functions in non-linear SVM. Two types of SVM (softmargin SVM and Least square SVM) were identified along with different multiclass approaches (one-against-one and one-against-all). From the previous knowledge it was able to identify that phoneme recognizer can use SVM for classification. Using least square SVM and one-against-one multiclass approach a best phoneme classifier can be developed. Thus we can conclude that SVM will be a good choice for classification in phoneme recognition since it is the simplest and most effective classifier with less complexity and great accuracy.

REFERENCES

[1] Eray, Osman, Sezai Tokat, and Serdar Iplikci. "An application of speech recognition with support vector machines." 2018 6th International Symposium on Digital Forensic and Security (ISDFS). IEEE, 2018.
[2] Aida-zade, Kamil, Anar Xocayev, and Samir Rustamov. "Speech recognition using support vector machines." 2016 IEEE 10th International Conference on Application of Information and Communication Technologies (AICT). IEEE, 2016.
[3] Chen, JinBiao, and Shiqing Zhang. "Manifold learning-based phoneme recognition." 2009 International Conference on Image Analysis and Signal Processing. IEEE, 2009.

[4] TombaloĞlu, Burak, and Hamit Erdem. "A SVM based speech to text converter for Turkish language." 2017 25th Signal Processing and Communications Applications Conference (SIU). IEEE, 2017.
[5] Sonkamble, Balwant A., and D. D. Doye. "An overview of speech recognition system based on the support vector machines." 2008 International Conference on Computer and Communication Engineering. IEEE, 2008.
[6] Fletcher, Tristan. "Support vector machines explained." Tutorial paper (2009).
[7] Suykens, Johan AK, and Joos Vandewalle. "Least squares support vector machine classifiers." Neural processing letters9.3 (1999): 293-300.
[8] Theodoridis, Sergios, and Konstantinos Koutroumbas. "Pattern recognition and neural networks." Advanced Course on Artificial Intelligence. Springer, Berlin, Heidelberg, 1999.
[9] Solera-Urena, Rubén, et al. "Real-time robust automatic speech recognition using compact support vector machines." IEEE Transactions on Audio, Speech, and Language Processing 20.4 (2012): 1347-1361.
[10] Cui, Jianguo, et al. "The application of support vector machine in pattern recognition." 2007 IEEE International Conference on Control and Automation. IEEE, 2007.
[11] Salomon, Jesper. "Support vector machines for phoneme classification." Master of Science, School of Artificial Intelligence, Division of Informatics, University of Edinburgh (2001).
[12] Cutajar, Michelle, et al. "Discrete wavelet transforms with multiclass SVM for phoneme recognition." Eurocon 2013. IEEE, 2013.
[13] Mporas, Iosif, et al. "Recognition of greek phonemes using support vector machines." Hellenic Conference on Artificial Intelligence. Springer, Berlin, Heidelberg, 2006.