ABSTRACT
Speech Recognition for Urdu language is an interesting and less developed task. This is primarily due to the fact that linguistic resources such as rich corpus are not available for Urdu. Yet, few attempts have been made for developing Urdu speech recognition frameworks using the traditional approaches such as Hidden Markov Models and Neural Networks. In this work, we investigate the use of three classification methods for Urdu speech recognition task. We extract the Mel Frequency Cepstral Coefficients, the delta and delta-delta features from the speech data and train the classifiers to perform Urdu speech recognition. We present the performance achieved by training a Support Vector Machine (SVM) classifier, a random forest (RF) classifier and a linear discriminant analysis classifier (LDA) for comparison with SVM. Consequently, the experimental results show that SVM gives better performance than RF and LDA classifiers on this particular task.

General Terms:
Automatic Speech Recognition

Keywords:
Linear Discriminant Analysis, Mel-Frequency Cepstral Coefficients, Random Forest, Support Vector Machines, Urdu

1. INTRODUCTION
Speech processing is an interesting area of research. One of the useful application of speech processing is automatic speech recognition (ASR) which is an effective tool for human-machine interaction. An ASR system enables a person to talk to a machine. Such applications are used in voice operated activities, automated credit card activation, security and surveillance facilities. More recent integration of ASR systems have enabled users talking to OK Google and Apple Siri using smart phones, tablets and phablets. Similarly, isolated words recognition or spoken digits recognition are widely popular in data entry automation, PIN code operated devices/applications/services, banking automation and voice dialing applications. The field of ASR has observed tremendous developments during the past few decades (for developed languages, in particular). These include ASR systems for English, French, Mandarin and Japanese [1,2]. For languages which are not rich in terms of available computational linguistics resources, the research is still less mature. This holds true for Urdu language as well, having insufficient progress for ASR development even though it is one of the largest languages of the world with approximately 60-70 million speakers around the world [3]. It is the national language of Pakistan, widely spoken and understood in India and has several hundred-thousand speakers in Bangladesh, Nepal, United Kingdom, South Africa, and United States [4]. It also shares significant vocabulary with Arabic and Persian, for example, dunya (meaning world) and kitab (meaning book).

In this paper, we present our work on speech recognition of Urdu digits using Support Vector Machines (SVMs), Random Forests (RF) and Linear Discriminant Analysis (LDA) classifiers. To the best of our knowledge, these approaches (excluding LDA) have not been reported for Urdu ASR before. We expect that this work will serve as a baseline for future research work on Urdu ASR as it is based on a more diverse Urdu speech corpus, unlike previous attempts which were based on mono-speaker database only (we outline this in Section 2). The remaining of this paper is organized as follows: In section 2, we summarize different existing approaches and results as available for Urdu ASR. In section 3, we provide an explanation of the features namely Mel-Frequency Cepstral Coefficients and the classification algorithms namely SVMs, RF and LDA. We then explain our experiment and present the results in Section 4. Finally, we conclude our paper in section 5 and outline future directions as well.

2. RELATED WORK
Ghai et al [6] have briefly presented several developments on Urdu ASR in their work on ASR of Indo-Aryan languages such as Hindi, Punjabi, Bengali and Gujarati. Akram et al [7] have presented an ASR system for continuous Urdu speech. However it lacks identification of a standard corpus used in the experimentation. It also misses information on the size of training and test sets and the mechanism for selecting the two sets. The reported accuracy is limited to 54%. In similar work on Urdu digits recognition, Ahad
et al [8] have reported the use of a multilayer perceptrons (MLP) for Urdu digits ASR from 0 to 9, however the work is limited to single speaker corpus and thus limited to speaker-specific applications only as they have not utilized a multiple speaker corpus. Hasnain & Awan [9] have also reported Urdu ASR for digits using a neural network model with two hidden layers for classification, however, their work is based on the use of fourier descriptors and correlations coefficients and not on MFCCs. Much higher accuracy rate has been reported, though, it is not clear whether the reported values are for training set or only have been obtained for an explicit test set. More sophisticated work on speech recognition use Hidden Markov Models (HMM) [10]. The HMM have also been used by [11] for Urdu, presenting a speaker-independent speech recognition system. The framework utilized by [11] is an open source Sphinx-4. They represent each word as a single phoneme which means that the system performance may suffer degradation for longer words. More recent work for continuous speech Urdu ASR is by Huda et al [12] using Sphinx open source toolkit. We are not making direct comparisons to Huda’s work [12] as the data scale as well as application scale differ from our work. Besides, the speech data used in their experimentation is limited to one particular accent only. Ali et al [13] have reported results on the same dataset of Urdu, as used in the work reported in this paper, and have analyzed the use of Discrete Wavelet Transform (DWT) features and compared the performance with MFCCs. They conclude their discussion in favor of using MFCCs. But the classifier is essentially the same to the one used in their previous work i.e. Linear Discriminant Analysis as in [14–16]. Their work does not involve making investigation of performance of more robust classifiers. We, on the other hand, suggest the use of SVM and compare it with a random forest classifier too [17]. In this work, we limit our experimentation to Urdu digits recognition.

3. METHODOLOGY

3.1 Mel Frequency Cepstral Coefficients

The Mel Frequency Cepstral Coefficients (MFCC) are popular and widely used within the speech processing community. MFCCs are based on the mel scale (a logarithmic scale) and attempt to mimic the human ear response. The Mel scale is directly related to the mechanism behind the human ear response to frequencies in an audio signal (this response is non-linear). The Mel-scale possess a linear spacing for frequencies below 1000 Hz and a non-linear spacing for frequencies above 1000 Hz [18]. The relation between the Mel scale frequencies and the Hertz frequencies can be represented by Eq. [1]

$$f_{mel} = 2595 \times \log (1 + \frac{f}{700Hz})$$

Several methods for MFCC extraction have been proposed by [18–20]. For MFCC calculation, we use the Malcom’s implementation [11]. We calculate 12 coefficients for each audio file and combine them with the delta and delta-delta coefficients. Delta and delta-delta coefficients are the time derivatives of the original coefficients and are helpful in retaining the dynamic information in the speech data. A basic step wise approach to calculate the MFCC and the delta, delta-delta MFCCs has been summarized in Algorithm 1.

### Algorithm 1 Algorithm for MFCC calculation

1. for $i = 0$ to No. of Frames do
2. Calculate Power Spectrum
3. end for
4. for $i = 0$ to No. of Filter Coefficients do
5. Mel Filter Bank Calculation
6. Apply the filter bank to the spectrum
7. $sumE \leftarrow \sum$ the energy in each filter
8. $logE \leftarrow log(sumE)$
9. end for
10. Discrete Cosine Transformation for the $logE$
11. Retain N coefficients
12. if $D \neq 0$ then
13. repeat
14. $Coeff(j) = Coeff(j) – Coeff(j-1)$ \{calculate delta coefficients\}
15. $j \leftarrow j – 1$
16. until $j = 0$
17. else
18. $Coeff \leftarrow Coeff$
19. end if
20. if $DD \neq 0$ then
21. repeat
22. $Coeff(j) = Coeff(j) – Coeff(j-1)$ \{calculate delta-delta coefficients\}
23. $j \leftarrow j – 1$
24. until $j = 0$
25. else
26. $Coeff \leftarrow Coeff$
27. end if

3.2 Support Vector Machines

Support Vector Machine (SVM) is a popular kernel based discriminative classification algorithm. SVMs were introduced first by Boser et al [21] and got their popularity for handwritten character recognition [22]. SVMs are regarded to be the large-margin boundary classifier as they attempt to separate the data with hyper planes by maximizing the distance from the data of both classes (for binary classification case). SVM is characterized by the underlying kernel function, say linear, polynomial and Gaussian. SVMs are usually easy to train and does not suffer the local optima problem. However, the selection of kernel function is critical. SVMs have been used for various machine learning task, for example, hand-written character recognition [23], object recognition [24], speaker recognition and language recognition [25]. SVM is a binary classification algorithm and is comprised of sums of kernel function $k(x_i, x_j)$:

$$f(x) = \sum_{i=1}^{N} \alpha_i t_i K(x_i, x_j) + d$$

where $t_i$ represents the ideal outputs, $\sum_{i=1}^{N} \alpha_i t_i = 0$ and $\alpha_i > 0$. The ideal outputs are either +1 or −1 depending upon the class the data sample belongs to. The value of $f(x)$ compared against a threshold decides the output class of a particular test sample. For multi-class problems, the one-vs-all approach is adapted usually to achieve classification. For SVM, we use the libSVM library [29], which can handle the problem of multi-class data. We train the SVM with

---


[28]: A nice tutorial on SVM is available from Burges [28].
The objective function for LDA transformation can be defined by

\[ K(x_i, x_j) = \exp(\gamma(||x_i - x_j||)^2) \]  

(3)

We select the optimal hyper-parameters \( \gamma \) and regularization constant \( C \) for the SVM after multiple iterations on the training and test data.

3.3 Random Forests

Decision Trees have been popular in computer vision applications where decision trees have been used as stand-alone entities. Random Decision Forest or Random Forest (RF) is the ensemble version of decision trees where the trees are randomly trained [17]. The use of RF was first reported by [30] for handwritten digit recognition. In an RF classifier, the trees are trained with randomly selected features and the average of the class posteriors from the trees is calculated for final prediction. RF classifiers are reported to be very successful in a variety of machine learning tasks [31]. In our work, we use an RF classifier on the task of speech recognition and our results for the RF classifier are based on 300 trees.

3.4 Linear Discriminant Analysis

The Linear Discriminant Analysis (LDA) [32] is both a classification and a dimensionality reduction technique. The LDA transforms the data to find a matrix \( \Theta \) and seeks to maximize the ratio between the inter-class variance and the intra-class variance, thus achieving greater separability. For classification, the Euclidean distance is calculated over each test example. Generally, for \( n \) classes, \( n \) Euclidean distances are calculated for each test example. The smallest distance then determines the class for the test example. The objective function for LDA transformation can be defined by \( J(\Theta) \):

\[ J(\Theta) = \frac{|\Theta^T \Psi(\Theta)|}{|\Theta^T W(\Theta)|} \]  

(4)

where \( W \) is the average within-class variance, \( \Psi \) represents the total variance matrix, \(|.| \) is the determinant value. We use LDA classification besides SVM and RF classifiers and compare the results on the digit recognition task.

4. EXPERIMENT

4.1 Dataset

This experiment is based on the dataset of Urdu digits recorded by ten speakers. The speakers include native/non-native, male/female speakers of different age. We normalize our data with zero-mean and variance of 1. We train our classifier with randomly selected training data. We choose a 7:3 ratio for training and test set distribution.

4.2 Confusion Matrix

We summarize the results of correct recognition of the test instances in a confusion matrix. For \( P \) number of words, we get a \( P \times P \) matrix. The general representation has been shown by \( CM \)

\[
CM = \begin{bmatrix}
m_{11} & m_{12} & m_{13} & \ldots & m_{1P} \\
m_{21} & m_{22} & m_{23} & \ldots & m_{2P} \\
m_{31} & m_{32} & m_{33} & \ldots & m_{3P} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
m_{P1} & m_{P2} & m_{P3} & \ldots & m_{PP}
\end{bmatrix}
\]  

For this confusion matrix, \( CM \), the diagonal entries show the correct word recognition i.e. \( m_{ij} \) for \( i = j \). Similarly, the number of mis-judgments of a test word against a particular word are shown by the non-diagonal entries, i.e. \( m_{ij} \) for \( i \neq j \). We report an overall test accuracy of 73% for the SVM classifier. We also report the results for a random forest classifier and LDA classifier, both resulting in an accuracy of 63%. The confusion matrix plots for SVM based classification, Random Forest based classification and LDA classification have been shown in Figure 1, Figure 2 and Figure 3 respectively. We show the recognition rate for each digit as achieved by SVM classifier, RF classifier and LDA classifier in Table 1. For this particular task, the accuracy achieved for LDA classifier is not different than the accuracy for RF classifier as both give an overall accuracy of 63%. However, by looking at the confusion matrix graph for the LDA classifier, it turns out that LDA classifier is the only classifier generating 0% accuracy for the word number 7 (See the empty 7th column in Figure 1).

5. CONCLUSION AND FUTURE WORK

In this paper, we have presented the use of SVM, RF and LDA classifiers for speech recognition of Urdu dataset. We trained the SVM classifier with MFCC features and observed a recognition rate of 73% on the test set. We also trained a random forest classifier and an LDA classifier with the same data and reported an overall accuracy of 63% for both. We observed better performance for the SVM classifier. The experiments are based on speech data of ten speakers including both native/non-native, male/female speakers of different age. We expect these results to serve as a baseline for future experimentation. We believe that these results will prompt further research on Urdu ASR. We suggest that the use of more robust classifier can help with achieving better performance. Similarly, exploring the use of SVM-HMM hybrid model is also an interesting dimension for future experimentation. Besides, an interesting and useful task would be to extend the work to continuous speech recognition of Urdu. We will also be exploring the use of deep learning models for Urdu ASR as they have been highly successful on large vocabulary speech recognition tasks.
### Table 1. Recognition Rate for Digits

<table>
<thead>
<tr>
<th>S. No</th>
<th>Word Number</th>
<th>Recognition Rate for SVM</th>
<th>Recognition Rate for RF</th>
<th>Recognition Rate for LDA</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>001</td>
<td>100%</td>
<td>66.67%</td>
<td>100%</td>
</tr>
<tr>
<td>2</td>
<td>002</td>
<td>66.67%</td>
<td>33.33%</td>
<td>33.33%</td>
</tr>
<tr>
<td>3</td>
<td>003</td>
<td>66.67%</td>
<td>100%</td>
<td>66.67%</td>
</tr>
<tr>
<td>4</td>
<td>004</td>
<td>100%</td>
<td>66.67%</td>
<td>66.67%</td>
</tr>
<tr>
<td>5</td>
<td>005</td>
<td>66.67%</td>
<td>33.33%</td>
<td>66.67%</td>
</tr>
<tr>
<td>6</td>
<td>006</td>
<td>33.33%</td>
<td>100%</td>
<td>66.67%</td>
</tr>
<tr>
<td>7</td>
<td>007</td>
<td>66.67%</td>
<td>66.67%</td>
<td>66.67%</td>
</tr>
<tr>
<td>8</td>
<td>008</td>
<td>100%</td>
<td>66.67%</td>
<td>0%</td>
</tr>
<tr>
<td>9</td>
<td>009</td>
<td>66.67%</td>
<td>33.33%</td>
<td>33.33%</td>
</tr>
<tr>
<td>10</td>
<td>010</td>
<td>66.67%</td>
<td>33.33%</td>
<td>100%</td>
</tr>
</tbody>
</table>

---

6. REFERENCES


