THE VITERBI ALGORITHM FOR ASR WITH CONTINUOUS-DENSITY HMMS
Rob Oxspring, Mark Greenwood
Dept. of Computer Science, University of Sheffield
Regents Court, 211 Portobello St, Sheffield, S1 4DP, UK
u7rjo2@dcs.shef.ac.uk; u7mag@dcs.shef.ac.uk
Electronic copy and resources available from http://www.dcs.shef.ac.uk/~u7rjo2/com326

ABSTRACT
This paper describes the implementation of two speech recognisers, for isolated and connected digit recognition. Both recognisers are implemented using a token passing version of the Viterbi algorithm. The isolated recogniser is shown to be 100% accurate across the limited test set. The connected recogniser, performs poorly with a maximum recognition accuracy of 49% percent after manual tweaking of the HMM probabilities. Alternative strategies are discussed but not implemented within this paper.

1. INTRODUCTION
The problem of speech recognition can be divided into two different classes of problem. Isolated word recognition is a much simpler task than connected word recognition, but both are important problems that need to be successfully solved. This paper deals with implementing an isolated and connected digit recogniser using the Viterbi algorithm.

The experiments in this paper are implemented in Matlab and all rely on code provided as part of [1]. This code includes the hidden Markov models and the probabilities of moving state and of the models generating the current time frame.

For more information on the Viterbi algorithm, and the possible implementations see [1] and [2].

2. ASR IMPLEMENTATIONS
2.1. Isolated Digits
The implementation of the isolated digit recognition system, is a very simple version of the Viterbi algorithm, implemented using token passing.

The algorithm works by iterating over each frame in the utterance, calculating the probabilities of all the tokens staying in the same state and for moving into the next state of each HMM. The final step of the iteration is to choose the most probable of the two options for each state-model combination and use these tokens as the starting point for the next iteration. The algorithm completes by taking the most probable token that is in the last state of a model in the last time frame. This is the token which represents the most likely path through the model for the utterance and hence the digit that has been recognised.

2.2. Connected Digits
The implementation of the connected recogniser is very similar to the isolated recogniser. The two difference being that we are storing the path that each token has taken through the models, and that the algorithm is no longer simply throwing away the last column at every time frame.

At each time frame the last column of the token array is appended to the beginning of the token array, effectively shifting all the other tokens on by one state, this is the difference between being able to recognise only a single digit and multiple digits.

The path through the models that is stored with each token is built up by storing the digit that was recognised when the last column is moved to the front of the array. This limits us to only being able to recognise a string of X digits, where X is the number of time frames minus the number of states in the models. This is a practical limit, but in reality this will not be a problem as a single digit is always going to last for more than a single time frame.

3. RESULTS
3.1. Isolated Recognition
The simple implementation of Viterbi for recognising isolated digits performs exceptionally well, even on very noisy data. Table 3-1 shows the results obtained from evaluating the isolated recogniser over the supplied test set of 38 different utterances.

<table>
<thead>
<tr>
<th>Data</th>
<th>Accuracy</th>
<th>Hits</th>
<th>Subs</th>
<th>Ins</th>
<th>Dels</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clean</td>
<td>100.00%</td>
<td>38</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Noisy</td>
<td>97.37%</td>
<td>37</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Very noisy</td>
<td>81.58%</td>
<td>31</td>
<td>7</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 3-1 Isolated accuracy
As you can see from the results table above, on clean speech the recogniser performs flawlessly over the limited test set available to us. Even on the very noisy speech the accuracy is still an impressive 82%.
3.2. Connected Recognition

The results of the connected recogniser were very poor. As you can see from Table 3-2 the main problem with this recogniser is a significant number of insertions.

<table>
<thead>
<tr>
<th>Data</th>
<th>Accuracy</th>
<th>Hits</th>
<th>Subs</th>
<th>Ins</th>
<th>Dels</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clean</td>
<td>17.95%</td>
<td>36</td>
<td>3</td>
<td>29</td>
<td>0</td>
</tr>
<tr>
<td>Noisy</td>
<td>-48.72%</td>
<td>30</td>
<td>9</td>
<td>49</td>
<td>0</td>
</tr>
<tr>
<td>Very noisy</td>
<td>-23.08%</td>
<td>22</td>
<td>15</td>
<td>31</td>
<td>2</td>
</tr>
</tbody>
</table>

Table 3-2 Connected accuracy

This table also shows no deletions, except when applied to very noisy speech that produces only 2. The actual insertions in clean speech can be seen in Table 3-3, which shows insertions being mainly the duplication of existing correct digits.

The figure below shows the confusion matrix for connected recognition achieving only 18% accuracy. On first inspection the image does not agree with the percentage but this is because the matrix does not show any insertions or deletions. The implication of this is that the recogniser is performing quite accurately, but is inserting and deleting too often.

This leads to the idea that the algorithm is moving forward too often, when it should have been electing to stay in the same state. Since the basic algorithm had been shown to be correct for isolated digits, the authors assumed that the algorithm was not at fault.

It was decided to try and account for the insertions by introducing a penalty for moving forward, effectively increasing the number of time frames taken to traverse an entire model. Multiple approaches were postulated, with the best performance coming from squaring the utterance probability. The reason this is so effective is unclear, but the evidence is presented in Table 3-4.

<table>
<thead>
<tr>
<th>Data</th>
<th>Accuracy</th>
<th>Hits</th>
<th>Subs</th>
<th>Ins</th>
<th>Dels</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clean</td>
<td>48.72%</td>
<td>22</td>
<td>11</td>
<td>3</td>
<td>6</td>
</tr>
<tr>
<td>Noisy</td>
<td>5.19%</td>
<td>9</td>
<td>17</td>
<td>7</td>
<td>13</td>
</tr>
<tr>
<td>Very noisy</td>
<td>17.95%</td>
<td>7</td>
<td>16</td>
<td>0</td>
<td>16</td>
</tr>
</tbody>
</table>

Table 3-4 Accuracy obtained using move next penalty

As you can see, this penalty has significantly improved the results of the connected digit recogniser, although it is still much worse than the isolated recogniser. The matrix below shows the results on the same clean speech as Figure 3-2, but thanks to the reduction of insertions the image seems to be in better agreement with the accuracy figure.
4. FUTURE WORK

More work is required to increase the performance of the connected recogniser, possibly using a different approach. One problem with the existing system may be that only one tokens matrix is being used for the entire string, this is ruling out possible paths and maybe the correct one. A solution here could be to take a working copy of the tokens matrix each time a wrap around is performed, and so not overwrite any data. This would obviously lead to a large memory overhead that would have to be reduced in some way to be efficient.

Another avenue for improvement builds on the fact that the last recognised digit is always correct. This lead to the idea that recognition may be better if working backwards, possibly with an initial forward pass to identify the last digit. This approach would probably not be useful in real time situations.

The final thing to mention is the benefit of a larger test set. This would probably reduce the isolated recogniser performance but would provide more constant results over the connected recogniser.

5. CONCLUSION

From the results, which have been provided in this paper, it is clear that the Viterbi algorithm is a useful algorithm for speech recognition. Recognising single digits is relatively simple, and very good performance is obtainable. Recognising strings of digits is a much harder task to perform accurately. Manual tweaking of the connected algorithm is not an acceptable long term solution and it is clear that more time needs to be spent working on the connected recogniser.

6. REFERENCES

