ACCURACY IMPROVING METHODS FOR PARAMETRIC TRAJECTORY MODELING AND ITS USE IN A* SEARCH

Yi-Yan Zhang
National Laboratory of Pattern Recognition, Institute of Automation
Chinese Academy of Sciences, Beijing
yyzhang@nlpr.ia.ac.cn

Wen-Ju Liu
National Laboratory of Pattern Recognition, Institute of Automation
Chinese Academy of Sciences, Beijing
lwj@nlpr.ia.ac.cn

Bo Xu
National Laboratory of Pattern Recognition, Institute of Automation
Chinese Academy of Sciences, Beijing
xubo@nlpr.ia.ac.cn

ABSTRACT
In this paper, we first address the measurements to improve classification accuracy for parametric trajectory modeling (PTM), exploring the effect of context-dependent information, prosody knowledge (pitch, duration) and derivative features (to depict speech dynamics further besides the advantage of PTM on this aspect). Experiment shows 61.585% error reduction with these techniques. We then use it in the A* search for continuous Mandarin digit recognition. Here two implementations are introduced. First PTM score is linearly combined with HMM acoustic score as the cost function of partial path. This method did not influence result much. Then PTM score is used as confidence measure in A* search. After PTM validating, if the HMM result has a high confidence, a high weight is given for the HMM score in the cost function and Vice Versa. This time we get 7.81% string error reduction for uniphone model and 12.15% for triphone model, and the corresponding del-error, in-error and sub-error degrade too.

1. INTRODUCTION
Continuous digit recognition is one typical small vocabulary continuous speech recognition task in ASR domain. It is very useful in many information services. For example, voice telephone dialing, identification of identity card number and so on. Especially we may suppose the following scenario: You would like to give someone a ring but your hands and eyes are too busy to input the phone number, e.g. while driving a car. Instead of stopping your current activity, you just utter the telephone number one by one and you will talk with the desired person in the next minute. How convenient and surprising are!

However, the accurate recognition of Mandarin digit recognition has a long way to go for practical application. This is mainly because:

- All digits are mono-syllables, and almost half of them are sonorant.
- “1” “2” and “5” are null INITIAL and result in the high insert and delete errors.
- Some digit pairs are very confusable, such as “2” and “8”, “6” and “9”, “6” and “Yao1” and so on, for they are similar in the universal used MFCC.

To obtain high performance recognition system, many methods are implemented to improve the accuracy [1][2][3]. They made a lot of modification from standard training to recognizing measurements, built precise modeling and utilized more useful information, which had reached a new improvement. We conclude that all these have done in the HMM framework. A more uniform alternative method—Segment Models (SM), however is rarely used.

SM use features on a segment instead of frame level, which can fully explore the features’ dynamics and temporal sequence characteristic. So it is superior on modeling ability compared with HMM. The disadvantage is its huge search space and subsequent tremendous computation. To solve this problem, we may use the phone boundary that HMM provides. Under this condition, SM can be used to guide the output score or as a confidence measure of the result. In this paper, we use SM rescoring as a heuristic score in A* search.

The remainder of the paper is organized as follows: In section 2, we briefly address a segment model—parametric trajectory modeling. Next, in section 3, we describe methods to improve classification accuracy for PTM and analysis their contribution. Finally, PTM rescoring application in A* search is given in section 4 and conclusion is given in section 5.
2. PARAMETRIC TRAJECTORY MODELING

Gish and Ng proposed a mathematical framework for the Parametric Trajectory Modeling [4], where the dynamics of feature were characterized by the time-varying means in a segment and the means is parameterized through constant, linear, quadratic or even higher order polynomial function. In this model, the observation vector sequences, \( O = \{O_{t,i}\}, t = 1,2,\cdots,T \) for frame and \( i = 1,\ldots,D \) for dimension, are characterized by:

\[
O_t = Z_tB_i + E_t
\]  

Where the first term is the trajectory component for feature \( i \) analogous to the means; the second term is the residual noise assumed to be independent.

3. ACCURACY IMPROVING METHOD

We will first improve PTM classification accuracy. This task is to serve for continuous Mandarin digit recognition, its data corpus is same as the one described in section 4.1. We use pre-segmental information to align the continuous digit string and then the derived isolated digits are trained for the corresponding model. This is the uniphone model. On the basis of this, we develop it to the context-dependent triphone model. The left and right questions are shown in table 2. We tie some triphones for which the amount of training data is less than the preset threshold. At last 279 models are gained.

<table>
<thead>
<tr>
<th>Feature Set</th>
<th>Accuracy Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>13MFCC+uniphone</td>
<td>96.72%</td>
</tr>
<tr>
<td>13MFCC+triphone</td>
<td>97.51%</td>
</tr>
<tr>
<td>13MFCC+PSI + triphone</td>
<td>97.99%</td>
</tr>
<tr>
<td>13MFCC+PSI + triphone +GMDDM</td>
<td>98.07%</td>
</tr>
<tr>
<td>13MFCC+PSI + Δτ+Δτ+triphone +GMDDM</td>
<td>98.74%</td>
</tr>
</tbody>
</table>

Table1: Digits’ classification accuracy

Note: The result for triphone is correct when the base model recognition is correct. The left and right environments are ignored however.

We give some concise explanations and discussions for table 1: PSI means pitch soft integration, where pitch is modeled as the 14th dimension feature just as MFCC and normalized energy do. What pitch differs from other features is that its polynomial trajectory is tone, which has lexical meaning in Mandarin. From this point of view, pitch is deserved further study in PTM.

GMDDM [6] means Gaussian Mixture Density Duration Model. Here we use multiple Gaussian mixtures to depict the duration distribution and the mixture number is determined automatically by ML criteria. Suppose that we have to estimate the duration distribution of the base model \( i \) according to \( N \) duration data \( d_1, d_2, \cdots, d_N \). By using k-Means clustering algorithm, we could figure out \( k \) mixture duration pdf:

\[
b_k(d) = \sum_{j=1}^{k} C_j N(u_j, \sigma_j^2)
\]

and the mixture number \( k \) is determined by:

\[
k^* = \arg \max_{1 \leq k \leq N} \sum_{j=1}^{N} \log(P(d_j | b_k(d)))
\]

The duration model is simple but effective.

We see from table 1 that context-dependent modeling and PSI contribute much to the recognition system. Though PTM can characterize feature’s dynamics effectively, the assumption for simplifying the model structure results in a lot of limitations. So derivative feature improves the recognition accuracy too. However the duration modeling is inferior to them.

4. A* SEARCH USING PTM INFORMATION FOR CONTINUOUS MANDARIN DIGIT RECOGNITION

4.1 Data Corpus Description

A continuous digit database collected in 1995 was used for training and recognition. Our data corpus includes 55 males and each person has 80 utterances. The length of each utterance varies from 1 to 7. In our experiments, we take 40 males’ data for training and the remaining 15 males’ data for testing. All the following experiments take the same training set and testing set.

In order to make our data corpus contains as many phenomena as possible, we deliberately designed our data corpus. The digit string in our data corpus has the following properties:

- All digits have the same probability to be uttered
- All digits connection are considered and balanced
- Every digit’s position (begin/middle/end) in the string is balanced

The data are recorded in the lab environment, and the SNR is less than 30db. The sample rate is 16k Hz and the quantification is 16 bit.

4.2 The Techniques Implemented in HMM [5]

In the system, we use 12-MFCC, normalized energy and their 1-order and 2-order derivatives as the baseline features. We use HMM as acoustic model. Each model is represented by a 6-state HMM with 16 mixtures of Gaussian densities per state, with exception of the background/silence that include only a single state.

For each digit, we build a word model independently. We build 12 Mono-word models in total; these models are “0” “1” “2” “3” “4” “5” “6” “7” “8” “9” “Yao1” “Silence”.

Besides the baseline uniphone model, we build context-dependent tri-word model to depict the acoustic models in detail. Here the left context is classified by FINAL part of the preceding digit, while right context is grouped by INITIAL part of the next digit. As table 2 shows, 10 groups for left context and 9 groups for right one are obtained.

<table>
<thead>
<tr>
<th>Left context</th>
<th>Right context</th>
</tr>
</thead>
<tbody>
<tr>
<td>{Silence},{0},{1},{7},{2},</td>
<td>{Silence},{0},{1,2,yao1},{3,4},{5},{6,9},{8}</td>
</tr>
<tr>
<td>{3},{4},{5},{6,9},{8}</td>
<td></td>
</tr>
</tbody>
</table>

Table2: Question set for triphone model
We then use data-driven decision-tree to generate the triphone models.

The time-synchronous Viterbi-Beam search algorithm we implemented here is as the following:

*Initialization*: For all the word state of tri-models, which can start a utterance

*Induction*:

For time $t = 1$ to $T$

For all Active States do

*Intra-word transitions*

For all active word-final states do

*Inter-word transitions*

*Pruning*: find the cost for the best theory and decide the beam threshold

*Termination*:

Word graph is derived based on Viterbi-Beam search.

### 4.3 A* SEARCH

When tracing the viterbi-beam search, a word graph is derived. A* search is then performed to decode it.

The backward A* search is a best-first search that proceeds backwards. What information is chosen as heuristic score is worth investigation [6]. In this paper, PTM score is first linearly combined with HMM score as the cost function of partial path.

As fig 1 shown, when a partial path P1 spans from time $t_1$ to the path P2 at time $t_2$, the word “$w_i$” is added, and the cost function is defined as:

\[
\begin{align*}
\beta(P2) &= \beta(P1) + \text{HMM } \_\text{Score}(w_i) + \\
\alpha \cdot \text{PTM } \_\text{Score}(O_{i_1}, \cdots, O_{i_r} | w_i)
\end{align*}
\]

(4)

Where $O_j$ is the $j$th feature vector, and $\delta(t, s_{ij})$ is the likelihood score of model $i$ at state $j$ and time $t$. We get prediction cost of some path from the first viterbi-beam search, as illustrated in formula 4.

<table>
<thead>
<tr>
<th>Search method</th>
<th>String - rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>One pass</td>
<td>87.2%</td>
</tr>
<tr>
<td>A*+PTM(score)</td>
<td></td>
</tr>
<tr>
<td>$\alpha = 10$</td>
<td>87.2%</td>
</tr>
<tr>
<td>$\alpha = 100$</td>
<td>87.2%</td>
</tr>
<tr>
<td>$\alpha = 500$</td>
<td>87.1%</td>
</tr>
<tr>
<td>$\alpha = 1000$</td>
<td>87.1%</td>
</tr>
</tbody>
</table>

Table 3: the result for PTM score linearly combined in A* search

This linear combination method did not contribute to or worsen the result much. The reason is that the PTM scores are not clearly distinguished though it has a high recognition performance.

Then we use PTM recognition result as the confidence measures in A* search. For the decoding speech boundary HMM provided, we use PTM to rescore it. Then we arrange the models in the order of their scores by decrease. According to its order in the sequence, we can decide the reliability of HMM result, as shown in table 4.

<table>
<thead>
<tr>
<th>The order of HMM result in the PTM score sequence</th>
<th>The reliability of HMM result (w)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.0</td>
</tr>
<tr>
<td>2</td>
<td>0.9</td>
</tr>
<tr>
<td>3</td>
<td>0.8</td>
</tr>
<tr>
<td>4</td>
<td>0.7</td>
</tr>
<tr>
<td>5</td>
<td>0.6</td>
</tr>
<tr>
<td>6</td>
<td>0.5</td>
</tr>
<tr>
<td>7</td>
<td>0.4</td>
</tr>
<tr>
<td>8</td>
<td>0.3</td>
</tr>
<tr>
<td>9</td>
<td>0.2</td>
</tr>
<tr>
<td>10</td>
<td>0.1</td>
</tr>
<tr>
<td>others</td>
<td>0.05</td>
</tr>
</tbody>
</table>

Table 4: the reliability of HMM result

We re-define the cost function as:

\[
\begin{align*}
\beta(P2) &= \beta(P1) + \lambda \cdot w(\text{PTM } \_\text{Score}) \\
\text{HMM } \_\text{Score}(w_i) \\
\alpha(P2) &= \delta(t_2, s_{ij})
\end{align*}
\]

(5)
And the total result is seen in table 5.

<table>
<thead>
<tr>
<th>models</th>
<th>Search methods</th>
<th>String-rate</th>
<th>Word-error</th>
<th>Del-error</th>
<th>In-error</th>
<th>Sub-error</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>One pass</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>uniphone</td>
<td></td>
<td>87.2%</td>
<td>10.8941%</td>
<td>1.2847%</td>
<td>2.6721%</td>
<td>6.9373%</td>
</tr>
<tr>
<td></td>
<td>$\lambda = 2$</td>
<td>87.9%</td>
<td>9.4553%</td>
<td>2.1069%</td>
<td>1.5930%</td>
<td>5.7554%</td>
</tr>
<tr>
<td></td>
<td>$\lambda = 3$</td>
<td>88.1%</td>
<td>9.3011%</td>
<td>2.1069%</td>
<td>1.5416%</td>
<td>5.6526%</td>
</tr>
<tr>
<td></td>
<td>$\lambda = 4$</td>
<td><strong>88.2%</strong></td>
<td><strong>9.2497%</strong></td>
<td><strong>2.1069%</strong></td>
<td><strong>1.5416%</strong></td>
<td><strong>5.6021%</strong></td>
</tr>
<tr>
<td></td>
<td>$\lambda = 5$</td>
<td>88.2%</td>
<td>9.2497%</td>
<td>2.1069%</td>
<td>1.5416%</td>
<td>5.6012%</td>
</tr>
<tr>
<td></td>
<td>A*+PTM_CM</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>$\lambda = 1$</td>
<td>89.3%</td>
<td>8.0576%</td>
<td>0.5276%</td>
<td>2.7338%</td>
<td>4.7962%</td>
</tr>
<tr>
<td></td>
<td>$\lambda = 2$</td>
<td>89.8%</td>
<td>7.4820%</td>
<td>0.5276%</td>
<td>2.1583%</td>
<td>4.7962%</td>
</tr>
<tr>
<td></td>
<td>$\lambda = 3$</td>
<td><strong>90.6%</strong></td>
<td><strong>7.3861%</strong></td>
<td><strong>1.5827%</strong></td>
<td><strong>1.0072%</strong></td>
<td><strong>4.7962%</strong></td>
</tr>
<tr>
<td></td>
<td>$\lambda = 4$</td>
<td>90.2%</td>
<td>7.8657%</td>
<td>1.6787%</td>
<td>1.0552%</td>
<td>5.1319%</td>
</tr>
<tr>
<td></td>
<td>$\lambda = 5$</td>
<td>90.4%</td>
<td>7.7218%</td>
<td>1.6787%</td>
<td>1.0072%</td>
<td>5.0360%</td>
</tr>
</tbody>
</table>

Table 5: the result for PTM score as a confidence measure in A* search

When the uniphone model is substituted by the context-dependent triphone model, the string recognition correct rate for the baseline goes up from 87.2% to 89.3%, and the error reduction is 16.4%. For the A* search using PTM confidence measure, the performance is improved for both models too, and the corresponding error reductions are 7.81% and 12.15% respectively. We may also see that, for uniphone model with the increment of $\lambda$, to a certain range (here is 5), the delete error, insert error and substitute error decrease consistently. However for triphone model, these items are not so regular.

5. CONCLUSION

In this paper, we present methods to improve PTM classification accuracy, and mainly introduce its application in A* search for continuous Mandarin digit recognition. From the experiment results, we point out:

- PTM score as a direct linear part in the cost function did not influence the result because for a speech segment the scores of different models did not differentiate much.
- PTM score as a confidence measure improves the string recognition rate. Here the choice of $\lambda$ and $w$ need more investigation further.
- PTM score may be used as a confidence measure of the final result. Compared with this use in A* search, it can not affect the result, but just reject ones with low confidence. However, it will result in less computation and deserves us to investigate in the future.

6. REFERENCES