Using MATLAB for Analysis of TRAP System

Martin KARAFIÁT [1,2], *František GRÉZL* [1]

[1] Dept. of Computer Graphics and Multimedia, Brno Univ. of Technology, Božetěchova 2, 612 00 Brno, Czech Republic
[2] Dept. of Computer Science, University of Sheffield, Portobello Street 211, S1 4DP Sheffield, United Kingdom

karafiat@fit.vutbr.cz, grezl@fit.vutbr.cz

Abstract. *This article describes a Matlab function for reading and processing file outputs from a structure of classifiers. These classifiers - neural nets - are used in speech recognition based on temporal trajectories (TRAP) of energy in frequency bands. This nonstandard approach is introduced and the program is presented. The utility of resulting figures is enhanced by the possibility of reading and displaying results from all critical bands at once. Resulting analyses are more focused on the reliability of classifiers than on the basic accuracy measure. The first uses colors and their depth to display both cues, reliability and accuracy, in one informative picture. Others are focused on more precise measures, where it is possible to precisely define classifier mistakes.*

Keywords

TRAP, speech recognition, features extraction, estimated probability analysis

1. Introduction

We have been working on speech recognition using a system based on temporal trajectories (TRAP) from critical bands [5]. Several probability estimators are used, trained to classify input vectors into one of the target classes.

An overview of TRAP use in different domains of speech processing until 2002 is summarized in Cernocky's habilitation thesis [5]. At the beginning of TRAP research, the TRAP-based features didn't reach the same recognition performance as standard MFCC or PLP coefficients. On the other hand, the combination of TRAPs with these features always leads to improvement.

TRAP features were used with success in Voice Activity Detection (VAD) task by A. Adami and P. Jain [3]. Here, TRAP features themselves brought a notable improvement. TRAP system was also incorporated into a Qualcomm-ICSI-OGI (QIO) feature extraction investigated for AURORA 2 competition [1]. Here the TRAP system was reduced to fit in AURORA restriction about total system latency and computation complexity.

With further investigation of TRAP system, the features became better also in small vocabulary speech recognition tasks. An improvement over basic system is reported in [2]. However the performance of standard features is not mentioned here, we know that TRAP based features overcomes them. Additional processing of critical band spectrogram was the important step to reach those results. Also phoneme recognition is a field where TRAP based features outperformed standard ones (see Schwarz's and Matejka's work [4]).

Current work focuses on TRAP-based large vocabulary continuous speech recognition in challenging meeting environments. The results obtained so far are preliminary, but quite promising.

We focus here on analysis of the probability estimators' outputs. Correctness of these ones outputs was our primary interest. The estimator produces a correct estimation if the output with the highest value has the same label as the transcription. Of secondary interest is the confidence of the estimator. The estimator is confident about its decision only if one output is high and the others are low.

We created a Matlab function for analysing these net outputs to facilitate better understanding of the TRAP system (see Section 3).

2. TRAP System

After speech segmentation into 25 ms frames and computation of the power spectrum, spectrum energies are integrated into M filter bands (15 Bark scaled trapezoidal filters) and logarithm is taken. The following processing is carried out in each frequency band:

1. Actual frame with +/- 50 frames context is taken, giving a 101 points long TRAP vector.
2. Mean and variance normalization of TRAP vectors.
3. Hamming windowing.
4. Linear transformation.

The resulting vector is put into the *band probability estimator* - a three layer neural net. This net is trained to classify the input vector into one of the N classes. The estimator has an input layer the size of the input vector, one hidden layer and a third output layer, the size of which is equal to the number of classes N. We used 29 phonemes as target classes.

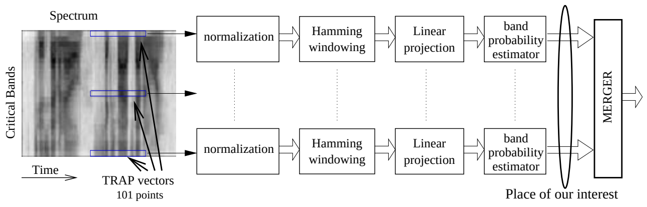

Fig. 1. TRAP system.

We obtain an N point long vector at the output of each band probability estimator. All output vectors are concatenated into a vector $M \times N$ points long. This vector goes through a negative logarithmic nonlinearity and then forms the input for the *merger probability estimator*. Merger probability estimator is also a three layer neural net trained to classify the input vector into the classes. The first layer has $M \times N$ points and the third layer again has N points - the same target classes as the band probability estimators. Its function is to merge the particular band estimations into one final posterior probability vector. The scheme of the whole TRAP system is shown in Fig. 1.

3. Description of Matlab program

As mentioned above, our function works with neural net outputs. Program *quicknet* [7], which is used for training and evaluation of these networks, saves results into the file in *rap* format (see section 3.1). These files represent outputs from band probability estimators and are used as inputs for our function where each band (file) can be processed separately or outputs from all critical bands can be read and analysed at the same time.

To begin with the program loads the desired outputs from *labelfile* into memory and than opens one or all input *rap* files according to the input specification. It next reads the first number (amount of output classes) and then starts to fill frame-by-frame the matrix of neural net outputs. The analysis is performed when the end of the sentence is reached. Finally, the system waits for any keyboard press before it continues processing the next sentence.

The program performs three kinds of analysis:

1. Accuracy and reliability in each critical band.
2. Comparison between desired outputs and outputs generated by networks.
3. Displaying critical band outputs for each class (phoneme) separately.

3.1 Input Data and RAP Format Structure

Data in RAP format are saved in uncompressed binary form in the order depicted in Fig. 2.

The file starts with the number of neural net output classes (neurons in output layer). Results from all output neurons for each input frame are concatenated into the small cells. The cell header is the frame number in the input sentence (indicating position in sentence) in integer form. The body is a block of neural net outputs in floating point form. A stream of these cells generates an output sentence. The number -1 (in integer form), is used as an *End of Sentence Identifier (ESI)*, and is added after the last cell of each sentence for easy detection of the sentence end.

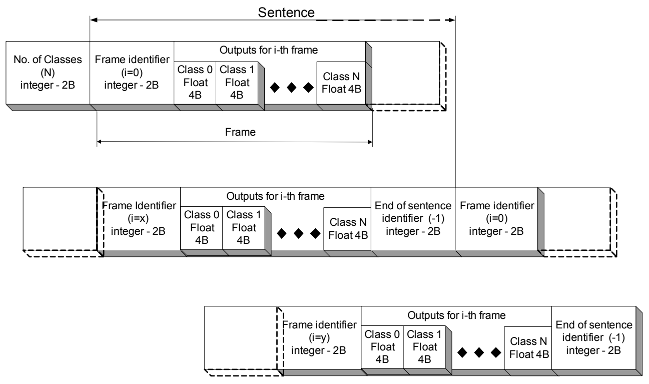
Sentence
No. of Classes (N) integer - 2B
Frame identifier (i=0) integer - 2B
Outputs for i-th frame
Class 0 Float 4B
Class 1 Float 4B
Class N Float 4B
Frame
Frame Identifier (i=x) integer - 2B
Outputs for i-th frame
Class 0 Float 4B
Class 1 Float 4B
Class N Float 4B
End of sentence identifier (-1) integer - 2B
Frame identifier (i=0) integer - 2B
Frame identifier (i=y) integer - 2B
Outputs for i-th frame
Class 0 Float 4B
Class 1 Float 4B
Class N Float 4B
End of sentence identifier (-1) integer - 2B

Fig. 2. Structure of RAP file.

Correct reading of integer and floating point parts is aided by the fact that each non-negative integer is followed by an exact count of floating point numbers, whereas a negative integer (*ESI*), is followed by a single integer.

In general our function opens one or all input *rap* files, then detects the amount of output classes, and finally fills a matrix of neural net outputs frame-by-frame. Outputs are numbers between 0 and 1 and their sum is 1 – they represent a posterior probability vector

$$0 < X(f,c,b) < 1, \qquad \sum_c X(f,c,b) = 1, \tag{1}$$

where f is the frame number, c is the output class number, and b is number of critical band.

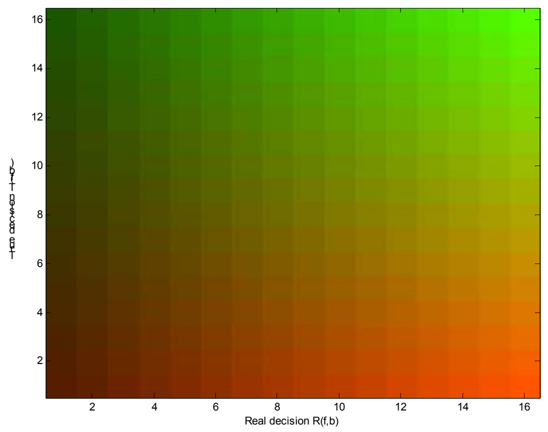

Fig. 3. Color scheme for functions $R(f,b)$ and $T(f,b)$.

3.2 Displaying Reliability and Accuracy of Decisions in Critical Bands

The purpose of this analysis is to display a visualization of the reliability of network outputs. The input is the value of the strongest output (real decision) $R(f,b)$ and the value of the desired output $T(f,b)$ (which corresponds to the label), where f is the frame number and b is number of critical band. We decided to map $R(f,b)$ and $T(f,b)$ into green-red color scheme according to figure 3, in order to generate a compact display of the information.

The value of $R(f,b)$ is always either larger (bad decision) or the same as the $T(f,b)$ value. Therefore output colors always lie either on the main diagonal or under it. Dark color expresses an unreliable network decision. Bright color expresses a reliable network decision. Green expresses a correct decision. Red expresses an erroneous decision.

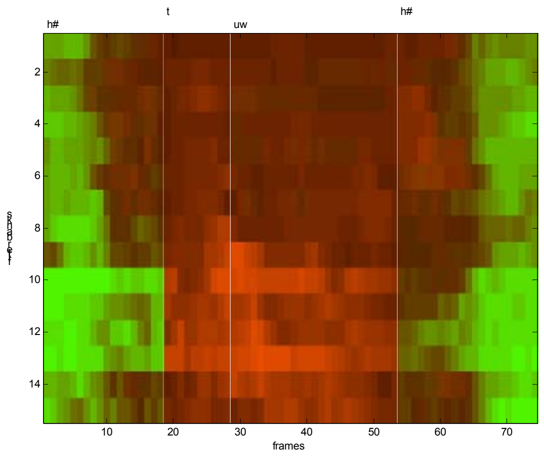

Fig. 4. Matrix of reliability decision for an utterance of digit “two”.

Adjusting of data and Matlab color scheme is important for mapping operation:

1. **Color scheme** – A *colormap* (Matlab color look-up table) is indexed in the range 0 to 255, so that blue is set to zero and the balance between red and green depends on the index of the corresponding color in figure 3. The indexing starts at point (0,0) and continues from left to right until point (16,16). This is programmed as:

```
V1 = (5:10/15:15)/15;
for ii = 0:15
  for y = 0:15
    RGB = [V1(ii+1)*(V1(16-y)),...
           V1(ii+1)* V1(y+1), 0];
    colormap1(ii+y*16+1,:) = RGB;
  end
end
```

2. **Data** – The input data matrix is transformed according to the equation:

$$M(f,b)=16\,T(f,b)+R(f,b)\,,$$
$$T(f,b)=15\;X\left[f,L(f),b\right], \qquad (2)$$
$$R(f,b)=15\max\left[X(f,c,b)\right],$$

where $L(f)$ is the index of the desired output according to the known label file. This is programmed as:

```
for i = 1:NoOfBands
  [max_in_crb(i,:),I] = max(X(:,:,i)');
    T(i,:) = round(15*max_in_crb(i,:));
    for ii=1:NoOfFrames+1
      R(i,ii) = round(15*XXnorm(ii,...
                      L(ii),i));
      M(i,ii) = R(I,ii)*16 + T(i,ii);
    end
end
figure(1);
colormap( colormap1);
image( M);
```

Resulting display for an utterance of digit “two” is shown in fig. 4. The procedure is discussed in par. 1 of sect. 3.5.

3.3 Comparison Between Real and Desired Network Outputs

We use matrices that were already created according to eqn. (2) for this analysis. The range of this data is from 0 to 15 and complex color mapping is not important. Consequently, we create a new *colormap* using only a black and white color scheme. The highest value (15) is represented by a light grey color and zero is represented by black.

This is programmed as:

```
V2 = (3:12/15:15)/15;
colormap2 = [V2’, V2’, V2’];
```

Resulting display for an utterance of digit “two” is shown in fig. 5. The procedure is discussed in the paragraph 2 of the section 3.5.

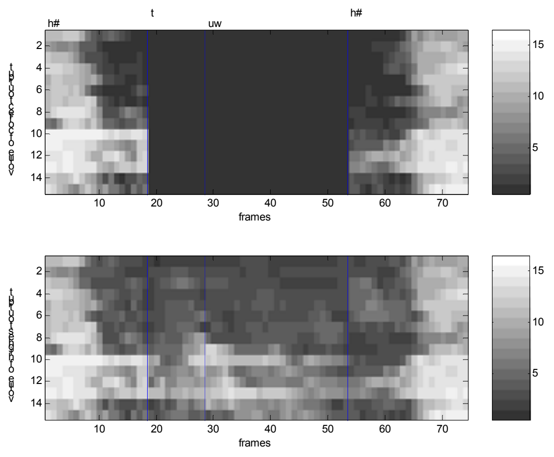

Fig. 5. Comparison of values in desired output (Up) and the strongest output (Down) for an utterance of digit “two”.

3.4 Critical Band Outputs for Each Class Separately

Displaying of detailed information about values in critical bands for each output class gives more information about the TRAP system. We can easily recognize classes where the network makes mistakes, the drawback being a large number of graphs.

We use the same *colormap* as in section 3.3 for displaying graphs and each output is multiplied by 15 in order to scale them into this *colormap* range.

Outputs for an utterance of digit "two", for 5 classes ("h#", "ax", "er", "uw", "ow"), are displayed in fig. 6.

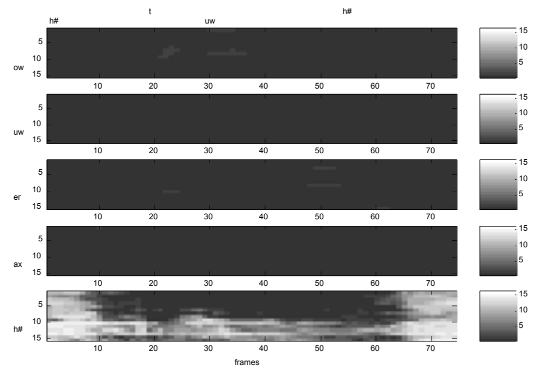

Fig. 6. The strongest outputs in classifier's classes "h#", "ax", "er", "uw" and "ow" for utterance of digit "two".

3.5 Summary of Analyses

1. **Reliability and accuracy** – The output from this analysis is a single picture that allows us to quickly estimate the quality of the classifiers. The analysis shown in fig. 4., tells us the system is confident in recognition of silence (symbol "h#") only, particularly in the 13th critical band, which generates the most reliable results. In low frequency bands the reliability of decisions falls for the whole utterance. In high frequency bands the classifiers generate confident but erroneous decisions in classifying the phonemes "t" and "uw". However, we are unable to identify the label(s) that were chosen instead of the desired label.
2. **Comparison between desired outputs and outputs generated by networks –** The purpose of this analysis (see figure 5) is very similar to the previous one, but it enables us to see that in the "t" and "uw" segments the output level is almost zero in all bands. This means that the networks are unable to recognize these phonemes from individual bands but this does not exclude the possibility that all unreliable band classifiers may converge to the right estimation at the output of the merger (main classifier).
3. **Critical band outputs for each class separately –** This analysis displays all information from the classifiers grouped by output class. We can see from figure 6 that the outputs for phoneme "h#" (silence) are high in segments where phonemes "t" and "uw" are the desired classes. This shows that many band classifiers are generating phoneme "h#" instead of the correct label, which is an unanswered issue from both previous analyses.

4. Conclusions

Even if we perform the core of our experimental work with software different from Matlab (*quicknet* and a combination of C-programs and shell scripts), Matlab is an excellent tool for visualization of classifier results. The generated figures help us to assess the performance of neural nets for different training and testing data and different experimental setups. The created program can be used to evaluate the results of any classifier producing posterior probabilities on data, where reference labels are available.

Acknowledgements

This work has been partially supported by EC project Multi-modal meeting manager (M4), No. IST-2001-34485 and by Czech Grant Agency (project no.102/02/0124). Martin Karafiát was supported by project FRVŠ 2003/842.

References

[1] ADAMI, A., BURGET, L., DUPONT, S., GARUDADRI, H., GREZL, F., HERMANSKY, H., JAIN, P., KARAJEKAR, S., MORGAN, N., SIVADAS, S., Qualcomm-ICSI-OGI features for ASR. In *Proceedings of International Conference on Spoken Language Processing, ICSLP 2002*, Denver, Colorado, USA, 2002.

[2] GREZL, F., HERMANSKY, H., Local averaging and differentiating of spectral plane for trap-based ASR. In *Proceedings of Eurospeech 2003*, Geneva, Switzerland, 2003.

[3] KINGSBURY, B., JAIN, P., ADAMI, A., A hybrid HMM/TRAPS model for robust voice activity detection. In *Proceedings of International Conference on Spoken Language Processing, ICSLP 2002*, Denver, Colorado, USA, 2002.

[4] ČERNOCKÝ, J., MATĚJKA, P., SCHWARZ, P., Recognition of phoneme strings using trap technique. In *Proceedings of Eurospeech 2003*, Geneva, Switzerland, 2003.

[5] ČERNOCKÝ, J., *Temporal processing for feature extraction in speech recognition*. Habilitation thesis. Brno: Brno Univ. of Technology, 2002. http://www.fit.vutbr.cz/~cernocky/publi/2002/habil.pdf

[6] HERMANSKY, H., SHARMA, S., TRAPS - classifiers of temporal patterns. In *Proceedings of 5th International Conference on Spoken Language Processing, ICSLP 98*, Sydney, Australia. Paper 615.

[7] JOHNSON, D., ftp://ftp.icsi.berkeley.edu/pub/real/davidj/quicknet-v0 96.tar.gz.

[8] The MathWorks, Inc., *Using MATLAB*, Natick, MA, 1997.

About Authors...

Martin KARAFIÁT was born in 1976 in Pelhřimov in Czech republic. He received Ing. degree in Electronics from Brno University of Technology (VUT) in 2001. Currently, he is a PhD student in the Speech Group of the Faculty of Information Technologies (FIT) VUT. Since 2003 he has been with the Speech and Hearing Group of the University of Sheffield, UK, as a research fellow. His major interests are in signal processing and speech recognition, especially continuous speech recognition.

František GRÉZL was born in 1977 in Šternberk in Czech republic. He received Ing. degree in Electronics from VUT in 2000. Currently, he is a PhD student in the Speech Group of FIT VUT. From 2001 to 2003, he was with the ASP Group at Oregon Graduate Institute, Portland, USA as a research assistant. His major interests are in signal processing, neural networks and speech recognition.