14th European Signal Processing Conference (EUSIPCO 2006), Florence, Italy, September 4-8, 2006, copyright by EURASIP

CLASSIFICATION OF SPEECH UNDER STRESS USING FEATURES SELECTED BY GENETIC ALGORITHMS

Salvatore Casale, Alessandra Russo and Salvatore Serrano

Dipartimento di Ingegneria Informatica e delle Telecomunicazioni
University of Catania, Viale A. Doria, 6 - 95125 Catania, Italy
email: (scasale, arusso, sserrano)@diit.unict.it

ABSTRACT

Determination of an emotional state through speech increases the amount of information associated with a speaker. It is therefore important to be able to detect and identify a speaker's emotional state or state of stress. The paper proposes an approach based on genetic algorithms to determine a set of features that will allow robust classification of emotional states. Starting from a vector of 462 *features, a subset of features is obtained providing a good discrimination between neutral, angry, loud and Lombard states for the SUSAS simulated domain and between neutral and stressed states for the SUSAS actual domain.*

1. INTRODUCTION

Technological progress has allowed for an increasing degree of man/machine interaction. This interaction can be improved and accelerated by means of spoken communication. In speech-based communications, emotions play an important role, sometimes playing an even bigger role than the logical information also included in the speech. One important research challenges in the last few years has thus been automatic recognition of the emotional state of a speaker through speech. It is a well-known fact that the state of stress in which a speech signal is produced alters the features of the signal. Being able to understand and identify the stress a speaker is under is therefore an important objective [1].
Some researchers have combined various techniques to enhance performance in recognition of emotional states through speech, often using different parameters at the same time [2]. In [3] we proposed a genetic algorithm feature selection approach to distinguishing between positive and negative emotional states. The aim of this paper is a broader classification taking various speech styles into account. To this end, nonlinear parameters will also be considered. It is not, in fact, correct to view the flow of air through the oral cavity as being linear and to assume that sound wave propagation is planar. It is more appropriate to see the production of speech signals as depending on interaction between different types of movement by the airflow. Studies conducted by Teager [4] suggest the presence of vortices in the proximity of the vocal chords which interact with the primary flow and are the main source of excitement during closure of the chords.
As emerges from the tests described in [5], it is important to consider parameters that are capable of detecting the presence of stress as independently as possible from the information contained in a phoneme. The Critical Band Based Teager Energy Operator Autocorrelation Envelope Area (TEO-CB-Auto-Env) has proved to be the most efficient parameter for this purpose. It is therefore useful to consider this nonlinear parameter if the aim is not only a simple distinction between neutral and stressed but a broader classification taking different speech styles into account. Starting from a vector of 462 features a genetic algorithm features selection procedure is implemented to distinguish between neutral, angry, loud and Lombard states for the SUSAS simulated domain and between neutral and stressed states for the SUSAS actual domain. This paper thus addresses the effect of insertion of TEO-CB-Auto-Env in the feature selection procedure, and comments on the results obtained in subsequent test phases.

2. DATABASE

The extraction of speech parameters in the presence of different emotional states was performed using the SUSAS Speech Corpus (Speech Under Simulated and Actual Stress). Two SUSAS domains were used: the *Simulated Domain* and the *Actual Domain*. Four different styles of speech from the first domain were considered: *angry*, *loud*, *Lombard* and *neutral*. The styles considered from the second domain were *neutral* and *stressed*. Since the TEO is more applicable for voiced sounds than for unvoiced sounds, only high-energy voiced sections (i.e., vowels, diphthongs, liquids, glides, nasals) were extracted from the word utterances [5]. For each of these speech styles, and for each of the domains, a subset of words was chosen and then used in the parameter selection, HMM training and test phases. The words were those used in [5], i.e. "freeze", "help", "mark", "nav", "oh", "zero". For each style of speech in the *Simulated Domain*, the nine speakers were asked to utter each word twice. We thus had 108 words per style: 6 (words) x 2 (utterances) x 9 (speakers) = 108.
In the *Actual Domain* we had 90 words.

3. SELECTION OF THE SUBSET OF FEATURES BY A GENETIC ALGORITHM

The audio files containing the words were processed using a pre-emphasis filter to highlight the high-frequency components and then split into 30ms frames at a rate of 10ms. Various parameters were extracted from each frame:

- 4 LPC Spectrum based Formants (F_{1-4})
- 16 Mel-Cepstral based parameters ($MFCC_{1-16}$)
- 16 Real Cepstrum based parameters ($RCEPS_{1-16}$)
- the Energy Level ($\log E$)
- autocorrelation based estimation of the Pitch (F_0)
- 17 Autocorrelation Coefficients (AC_{1-17})
- 16 Linear Prediction Coefficients (LPC_{1-16})
- 16 Reflection Coefficients ($PARCOR_{1-16}$)
- 16 Log Area Ratio Coefficients (LAR_{1-16})
- 16 Line Spectral Frequencies Coefficients (LSF_{1-16})
- 17 LPC Cepstral based parameter ($LPCC_{1-17}$)

- the Zero Crossing Rate (*ZCR*)
- the variance of the Linear Prediction Error ($\sigma^2_{E_{LPC}}$)
- 16 Critical Band Based Teager Energy Operator Autocorrelation Envelope Area ($TEO-CB-Auto-Env_{1-16}$)

The first- and second-order time differences are also computed as

$$\begin{aligned}\Delta x(n) &= x(n+1)-x(n-1)\\ \Delta\Delta x(n) &= \Delta x(n+1)-\Delta x(n-1)\end{aligned} \tag{1}$$

The selection system had $n = 462$ values to work on for each frame. To obtain the best subset of m variables out of a total of n for classification between positive and negative emotional states, a certain separation criterion has to be defined. The criterion we used is the scatter matrix [6]. A *within-class scatter-matrix* shows the scatter of samples around their respective expected class vectors:

$$\begin{aligned}S_w &= \textstyle\sum_{i=1}^{L} P_i E\left\{(X-M_i)(X-M_i)^T|\omega_i\right\} =\\ &= \textstyle\sum_{i=1}^{L} P_i \Sigma_i\end{aligned} \tag{2}$$

where: P_i is the a priori probability for class i, X is the parameter vector, M_i is the mean vector for class i, Σ_i is the covariance matrix for class i, ω_i represents class i, and L is the number of classes. The *between-classes scatter matrix* represents the scatter of the expected vectors around the mixture mean as

$$S_b = \sum_{i=1}^{L} P_i (M_i - M_0)(M_i - M_0)^T \tag{3}$$

where $M_0 = E\{x\} = \sum_{i=1}^{L} P_i M_i$ represents the expected vector of the mixture distribution. The separation index used J_1 was calculated from the scatter matrixes [3] on the basis of the following relation

$$J_1 = tr(S_w^{-1} S_b) \tag{4}$$

The aim was to determine an optimal subset of features for classification between different emotional states. It is too complex to do this via analysis of all the possible combinations (with $n = 462$ and $m = 48$ there are $5.13 \cdot 10^{65}$ possible combinations). In [3] we used two suboptimal techniques, *forward selection* and a technique based on *genetic algorithms (GAs)* and we showed that the performance obtained with the selection technique based on *a genetic algorithm* was consistently better than that of the *forward selection* technique. The GA is a stochastic global search method that mimics the metaphor of natural biological evolution. GAs operate on a population of potential solutions applying the principle of survival of the fittest to produce (hopefully) better and better approximations to a solution. At each generation, a new set of approximations is created by the process of selecting individuals according to their level of fitness in the problem domain and breeding them together using operators borrowed from natural genetics. This process leads to the evolution of populations of individuals that are better suited to their environment than the individuals that they were created from, just as in natural adaptation. We therefore used a technique based on *genetic algorithms (GAs)*, obtaining a subset containing $m = 48$ features [7]. The fitness function used to run the *GAs* was equal to the inverse of the separation index J_1^{-1}. Having set the number of individuals making up the initial population, $NIND = 100 \cdot m$, the first chromosome is randomly generated, comprising a matrix of size $NIND \cdot n$, in which each element is either 0 or 1 and such that the number of 1*s* in each row is equal to m.
The algorithm generating the first chromosome is shown in Fig. 1. The function *randperm(n)* returns a random permutation of the first n integers. A selective reproduction operator

```
ROUTINE CREATE CHROMOSOME
INPUT:
nothing
OUTPUT:
chromosome C

C_{i,j}=0 :  i=1..NIND, j=1..n
I = randperm(n)
j1 = 1
j2 = m
i = 1
while i <= NIND
   C_{i,I[j1..j2]}=1
   j1 = j1 + m
   j2 = j2 + m
   if j2 > n
      if j1 < n
         i = i + 1
         C_{i,I[j1..n]}=1
         C_{i,I[1:j2-n]}=1
      end
      I = randperm(n)
      j1 = 1
      j2 = m
   end
   i = i + 1
end
```

Figura 1: Algorithm used to create a new chromosome.

(Selch) selects a new chromosome from the old one on the basis of the fitness functions for each row; the new chromosome is of the same size and has a number of 1*s* per row equal to m; the crossover and mutation operators are applied to this new chromosome. The crossover operation is applied with a probability of $P_x = 0.7$ when the pairs are chosen for breeding. Fig. 2 shows the algorithm used for recombination. Let O and E be the arrays containing the indexes of the features selected for the parents; having generated a random floating number between 0 and 1 (rand(1.0) function) recombination is only performed when this number is lower than the pre-established P_x. An integer position x (randint function), is selected uniformly at random between 1 and the string length, m, and the genetic information exchanged between the individuals about this point; then two new offspring strings O^* and E^* are produced. When the parents have indexes in common, the offspring may have fewer than m features selected. For this reason the check routine illustrated in Fig. 3 is used, which ensures offspring with the pre-established number of features, m. This is achieved by exploiting the indexes not shared by the parents and the offspring produced (in the algorithm in Fig. 3 the "\" operator yields all indexes in the

```
ROUTINE CROSSOVER
INPUT:
fitness ordered old chromosome C
OUTPUT:
new chromosome C*

C*_{i,j}=0 :  i=1..NIND, j=1..n
while i <= NIND
   if rand(1.0) < P_x
      k=1, h=1
      for j=1...n
         if C_{i,j}==1
            O_h = j
            h=h+1
         end
         if C_{i+1,j}==1
            E_k = j
            k=k+1
         end
      end
      x = randint(m)
      O*=[O_1 ...O_x E_{x+1} ...E_m]
      E*=[E_1 ...E_x O_{x+1} ...O_m]
      O* = check(O*, i, O, E)
      E* = check(E*, i+1, O, E)
      for h=1...m
         C*_{i,O*_h}=1
         C*_{i+1,E*_h}=1
      end
   end
   i = i + 2
end
```

Figura 2: Algorithm used for crossover.

array that appear in the first operand but not in the second). The mutation algorithm is applied in such a way that it can be verified with a probability of $P_m = 0.7$ for each member of the population. When one or more members invert their value, passing from 0 to 1 or 1 to 0, the number of elements with a value of 1 must be equal to m. Once again a check algorithm is used.

For each generation cycle the positions of the 1s in the row with the lowest fitness value indicate the m best parameters for each generation. The generational cycle is repeated 300 times and at each generation the system stores the set of m parameters with the best performance in terms of the separation index. At the end of the generational cycle the set chosen is the one with the best separation index.

Fig. 4 is an example of the trend followed by the separation index (the inverse of the fitness function) as the number of generation cycles progresses.

Table 1 indicates the 48 features selected using the GA technique to classify between the 4 states of the *Simulated Domain* and between the 2 states of the *Actual Domain*.

4. EVALUATIONS

To evaluate the performance of the subset of features selected using the GA technique, we performed three different evaluations:

```
ROUTINE CHECK
INPUT:
offspring index array X
chromosome position p
first parent index array P_1
second parent index array P_2
OUTPUT:
checked offspring index array X

if Σ_{j=1}^{n} C*_{p,j} < m
   X*=sort(X)
   A = [P_1 P_2]
   D = A \ X*
   I = randperm(length(D))
   k=1
   for h = 1..m
      if X*_h == X*_{h+1}
         X*_h = D(I(k))
         k=k+1
         if k > length(D) k = 1
      end
   end
end
X = X*
```

Figura 3: Algorithm used to maintain a constant number of features selected after crossover.

- Text-Dependent Pairwise Stress Classification
- Text-Independent Pairwise Stress Classification
- Text-Independent Multistyle Stress Classification

The classifier used in the test was a baseline five-state HMM-based classifier with continuous distributions, each with two Gaussian mixtures. The HMMs were trained and tested using the HTK-3 tool.

The performance of the selected subset of features are compared with traditional pitch and mel-frequency cepstrum coefficients (MFCC) and with the TEO-CB-Auto-Env non linear parameter.

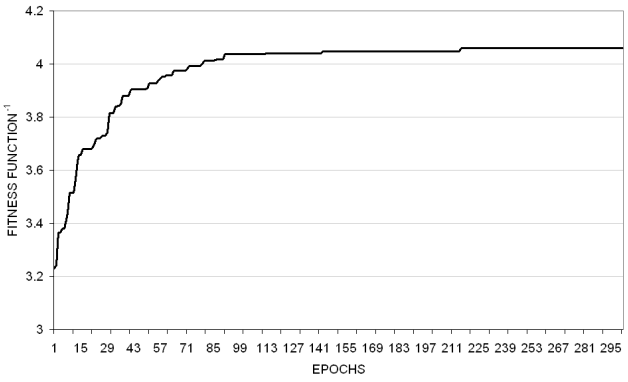

Figura 4: Example of fitness function trend.

Tabella 1: *Genetic Algorithm* Features

Parameter	Features Selected Δ^0	Δ^1	Δ^2	#
AC_{1-17}	1	15	-	2
F_0	1	-	1	2
F_{1-4}	1,2,3,4	2,4	2,4	8
LAR_{1-16}	3	-	4	2
$\log E$	1	1	-	2
LPC_{1-16}	-	-	2	1
$LPCC_{1-17}$	3,4,5,7,15	-	13	6
LSF_{1-16}	1,5,12	-	1	4
$MFCC_{1-16}$	9	-	-	1
$PARCOR_{1-16}$	1,2,3,4,6,9,11	-	1,11	9
$RCEPS_{1-16}$	1,2,8	-	-	3
TEO_{1-16}	1,3,7,11	-	8,12,16	7
$\sigma^2_{E_{LPC}}$	1	-	-	1
ZCR	-	-	-	0

4.1 Text-Dependent Pairwise Stress Classification

The first step involved text-dependent pairwise classification, in which the HMMs were trained and tested with the same words. An HMM was trained with the voiced part of each of the words from each style of speech chosen for the training phase. There are thus 24 HMMs (6 words x 4 styles of speech) for the *Simulated Domain* and 12 (6 words x 2 styles of speech) for the *Actual Domain*. The HMMs were trained with a series of "replicas" of the same word were uttered by various speakers. Due to the low number of tones available for pairwise classification, the "round-robin" method used in [5] was applied (e.g. in the *Simulated Domain* for each of the 18 "replicas" of a word the relative HMM is trained with 17 of the replicas and tested with the remaining word). The results of this classification are shown in Fig. 5, from which it can be observed that the system using features obtained by GA classification gives on average a 5% improvement. Only

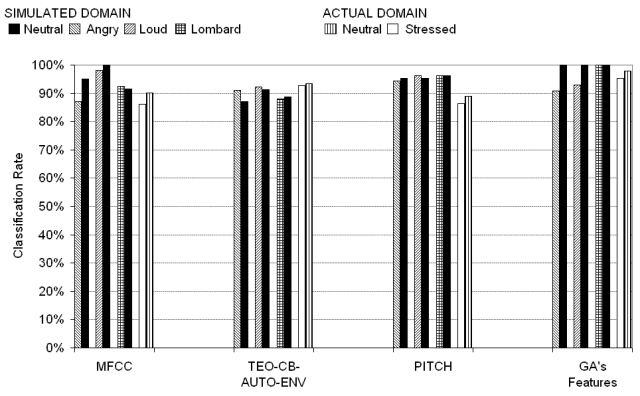

Figura 5: Text-Dependent pairwise stress classification results.

in the case of classification between *loud* and *neutral* was better performance achieved using MFCC parameters.

4.2 Text-Independent Pairwise Stress Classification

The second test involved text-independent pairwise classification to see whether the performance of these parameters dependent, and to what extent, on the information contained in a text or phoneme. A single HMM was trained for each style of speech in the two domains: for the *Simulated Domain* four HMMs were trained with 108 words belonging to the four styles, whereas 270 different words were used in the test phase. For the *Actual Domain* the two HMMs for the *neutral* and *stressed* styles were trained with 94 words each and the tests were performed using 140 different words. Fig. 6 shows the results of this classification. Text-independent

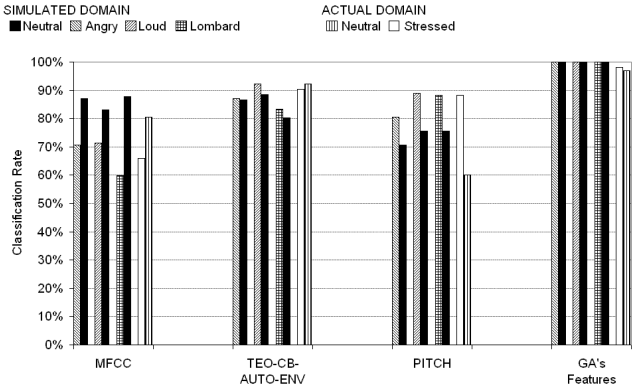

Figura 6: Text-Independent pairwise stress classification results.

classification using the GA features performed very well as regards the pairs belonging to the *Simulated Domain*, and also in the *Actual Domain* performance was clearly better than that achieved using the other parameters.

4.3 Text-Independent Multistyle Stress Classification

The aim of the last phase was multistyle text-independent stress classification. The aim was to verify the accuracy of the parameters in distinguishing between neutral and stress-affected speech, and then to evaluate their efficiency in classifying various types of stress. The Actual SUSAS domain was not considered in this phase as the stress present in the tones in this domain is strong and less easy to detect in most real cases. Each of the 270 words outside the vocabulary used in the Text-Independent Pairwise test phase was classified using the four HMMs for the four speech styles in the *Simulated Domain*. The output was therefore not simply words classified as neutral or stressed but as belonging to one of the four styles of stress considered.

The results obtained are given in Table 2: the first two columns give the rate of correct recognition of words belonging to the neutral and stressed classes, while the following three give the percentage of classification of the various stress styles. Comparison is made exclusively with the TEO-CB-AUTO-ENV parameter as it allows for better multistyle classification (Table 3). Analysis of the tables shows that when GA features are used, performance is considerably better in classification of the *neutral* style. Considerable improvement is also achieved in classification of the *angry* and *Lombard* states. A slight deterioration is observed in classification

Tabella 2: Text-Independent multistyle classification using GA selected features.

Test **S**peech **S**tyle	Correct Detection Rate (%)		Distribution of Stress Detection Rate (%)		
	Neutral	**S**tressed	**A**ngry	**L**oud	**L**ombard
Neutral	100	0	0	0	0
Angry	9	91	75.44	3.04	12.52
Loud	10	90	0	25.38	64.62
Lombard	0	100	3.78	5.27	90.95

Tabella 3: Text-Independent multistyle classification using TEO-CB-AUTO-ENV.

Test **S**peech **S**tyle	Correct Detection Rate (%)		Distribution of Stress Detection Rate (%)		
	Neutral	**S**tressed	**A**ngry	**L**oud	**L**ombard
Neutral	73.55	26.45	4.32	2.1	20.03
Angry	7.4	92.6	67.21	15.99	16.80
Loud	0.74	99.26	36.3	35.49	28.21
Lombard	15.55	84.45	10.55	9.65	79.80

of the *loud* state, which is often misclassified as a *Lombard* state.

5. CONCLUSIONS

The paper has proposed an approach based on GA selection procedure to determine a set of features that allow to distinguish between different styles of stress. In the feature selection procedure the TEO-CB-Auto-Env parameter is also inserted, because non-linear parameter is essential to obtain a broader classification taking different speech styles into account. It has been demonstrated that the recognition system using the parameters selected by GAs performed better than the traditional pitch and MFCCs and the non linear parameter TEO-CB-Auto-Env in the three different evaluations: Text-Dependent Pairwise Stress Classification, Text-Independent Pairwise Stress Classification and Text-Independent Multistyle Stress Classification.

Rather than recognising emotional states from the way a single word is uttered, the authors think that better results could be obtained by analysing whole sentences uttered under a given type of stress. Sentences could be divided into sections of finite duration and the technique could then be applied to each section.

Acknowledgment

The authors would like to thank TIM (Telecom Italia Mobile) for supporting this work.

REFERENCES

[1] R. Cowie, E. Douglas-Cowie, N. Tsapatsoulis, G. Votsis, S. Kollias, W. Fellenz, and J.G. Taylor, "Emotion recognition in human-computer interaction," *IEEE Signal Processing Magazine*, vol. 18, no. 1, pp. 32–80, Jan 2001.

[2] S. E. Bou-Ghazale and J. H. L. Hansen, "A comparative study of traditional and newly proposed features for recognition of speech under stress," *IEEE Transaction on Speech and Audio Processing*, vol. 8, no. 4, pp. 429–442, Jul 2000.

[3] F. Beritelli, S. Casale, A. Russo, and S. Serrano, "A genetic algorithm feature selection approach to robust classification between positive and negative emotional state in speakers," in *39th Annual Asilomar Conference on Signals, Systems, and Computers*, Oct-Nov 2005.

[4] H. Teager, "Some observations on oral air flow during phonation," *Acoustics, Speech, and Signal Processing IEEE Transactions on*, vol. 28, no. 5, pp. 599–601, Oct 1980.

[5] G. Zhou, J. H. L. Hansen, and J. F. Kaiser, "Nonlinear feature based classification of speech under stress," *IEEE Transaction on Speech and Audio Processing*, vol. 9, no. 3, pp. 201–216, Mar 2001.

[6] K. Fukunaga, *Introduction to Statistical Pattern Recognition*, chapter 10, pp. 446–448, Academic Press, 1990.

[7] D.E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley, 1989.