FACTA UNIVERSITATIS (NIŠ)
SER.: ELEC. ENERG. vol. 19, no. 2, August 2006, 287-298

Enhancing Robustness of Speech Recognizers by Bimodal Features

Inge Gavat, Gabriel Costache, and Claudia Iancu

Abstract: In this paper a robust speech recognizer is presented based on features obtained from the speech signal and also from the image of the speaker. The features were combined by simple concatenation, resulting composed feature vectors to train the models corresponding to each class. For recognition, the classification process relies on a very effective algorithm, namely the multiclass SVM. Under additive noise conditions the bimodal system based on combined features acts better than the unimodal system, based only on the speech features, the added information obtained from the image playing an important role in robustness improvement.

Keywords: Robust speech, bimodal system, support vector machines, neural networks.

1 Introduction

The main problem of many classification systems is that there are not robust, their performances are not constant especially when the conditions (environment, user, application) are changed. There are two causes for that: first the source of the signals that should be classified can be corrupted with noisy unwanted components and second, the classifiers cannot deal properly with new variants of the same pattern. Concerning the first problem, in image classification systems (especially face recognition or detection) for example, different illuminations and positions of the object to be recognized can be seen as introducing unwanted components. In the

Manuscript received Mart 10, 2006. An earlier version of this paper was presented at the 7th Seminar on Neural Network Applications in Electrical Engineering, NEUREL-2004, September 23-25, 2004, Belgrade, Serbia and Montenegro.

The authors are with University Politehnica of Bucharest, Romania (emails: igavat @alpha.imag.pub.ro, [gabi, i_claudia]@lpsv.pub.ro).

audio classification systems such unwanted components are represented by the inherent noise that is captured along with the signal to be classified. The usual solution for this kind of problems is a preprocessing of the signal before classification in order to eliminate the unwanted components, with the draw back to affect also the original signal. Another possible and more viable solution could be the use of features obtained from more sources, connected with the object to be classified, acting in a multimodal way. Concerning the second problem, Artificial Neural Networks [1] and many statistical methods offer solutions by allowing to form models of one pattern using more variants of the pattern. Furthermore this models can be re-trained using new particular occurrences of the pattern so that the system is able to learn from examples.

In this paper is proposed a robust speech recognition system, based on a bimodal structure using features obtained from two sources: the speech signal and the speaker image and applying for classification the Support Vector Machines [8] algorithm that combines the advantages of ANNs and statistical approaches by having good generalization and learning properties. SVMs were successfully used in a multimedia classifier [2].

A bimodal system is a particular case of multimodal system, namely that system that uses features obtained not only from the signals that should be classified but also from other signals related with them.

The bimodal systems act in two major steps like each unimodal classification system. In the first step feature extraction is performed, where are determined only the important characteristics of the signal, in the second, the recognition is realized, where based on a classification algorithm is made a decision. There are two main strategies to build multimodal system [3].

The first method is to apply decision fusion and means taking a decision for each source of information and combine those two to make the final decision. The most common way to implement the decision fusion algorithm is using neural networks or Markov models where the entries of the network are the output of each classifier from each source.

The other method to construct a multimodal system is using the feature fusion. This means that after feature extraction from each source a combined feature set is realized as basis for multimodal models and then applying any classification method we make the decision. The main disadvantage of this second method is that we have to synchronize the signals from the different sources.

For each source we can use different parameterization methods depending on the signals. Depending on the application, the signals can be images, audio signals and others.

2 Architecture

The recognition system we have experimented is given in Figure 1 and is based on fusion of parameters obtained from speech and from image. In order to combine the feature vectors, the two signals have to be synchronized, this being the main weakness of this type of bimodal system architecture. Because the database we used had synchronization between speech and image, we can apply without problems the architecture based on parameters fusion.

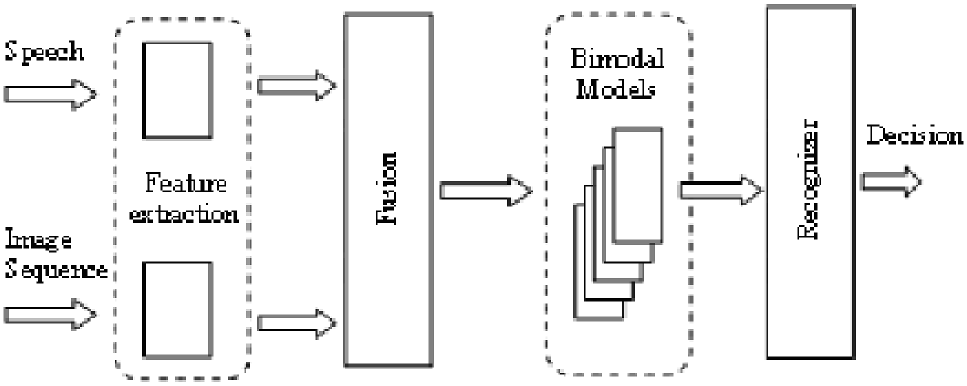


Fig. 1. Bimodal speech recognition system

The first step in the system is feature extraction where we extract only the important characteristics of the signals. For speech parameterization we used perceptual approaches of two well known methods: linear prediction and cepstral analysis. From image we extract geometric features of speaker's mouth. For that, first a face tracking algorithm based on Gaussian Mixture Models and then a deformable template was used to model the face. The deformation was calculated so the template would contain as many pixels from the face as possible. The decision for each pixel to be or not in the face class was taken using the Bayes statistical criteria.

Features were combined by simple concatenation of feature vectors for each analyzed window (or frame). After fusion we construct bimodal models for the patterns we want to classify.

For classification we choused to use a statistical approach called Support Vector Machines. SVM is a binary decision method with a good generalization property and is based on finding an optimal hyperplane as a decision boundary between classes. Also SVM is a kernel method meaning that the hyperplane is found in a feature space using a non-linear transformation which transform the input space in a feature space which has a much bigger dimension and we don't have to calculate the transformation for each data sample, we have to calculate only some kernel products in order to find the hyperplane.

In order to extend the binary algorithm to multiclass decision we combined several binaries SVMs using Directly Acyclic Graph SVM (DAGSVM) algorithm.

The first stage in the classification process is to train the support vector network (find the hyperplane) using some of the data samples (bimodal models) from the database and next we test the trained network using the other models from database or the same models used in the training process.

3 Signal Processing

First step in all recognition or classification tasks is signal analysis, where the signal is processed in order to obtain the important characteristics, further called features or parameters. By using only the important characteristics of the signal, the amount of data used for comparisons is greatly reduced and thus, less computation and less time is needed for comparisons.

3.1 Audio Signal Processing

Our audio parameter extraction is based on perceptual linear predictive coding and perceptual cepstral coding, methods that will be further summarized. There are few blocks commune in both linear prediction and cepstral coding. The first common stage is frame blocking, used because audio signals is fundamentally a non stationary signal, so we cut short fragments during which the speech signal can be approximated as a quasi-stationary random process. Then we passed each frame through a Hamming window. We can compute at this time the energy of each frame and we can use the energy set of coefficients in the recognition process for more accuracy. Next in order to obtain the perceptual version of the LPC and cepstral coefficients we manipulate the spectrum of the speech signal. The spectral manipulation for mel-cepstral coding is represented by a set of filters. The outputs of the filters are calculated using eq. 1, where i is the number of the filter.

$$Y(i) = \sum_{k=0}^{\frac{N}{2}} \log|S(k,m)|H_i(k\frac{2\pi}{N}) \tag{1}$$

First and second order variations of the mel-cepstral coefficients are used for speech recorded in noisy environments or under influence of stress or emotional factors [1]. The spectral manipulation for perceptual linear prediction is represented in Fig. 2 The PLP audio analysis method is more adapted to human hearing, in comparison to the classic Linear Prediction Coding (LPC).

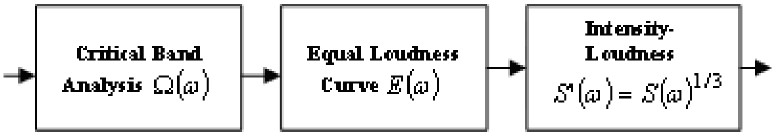

Critical Band Analysis $\Omega(\omega)$
Equal Loudness Curve $E(\omega)$
Intensity-Loudness $S'(\omega) = S(\omega)^{1/3}$

Fig. 2. Block representation for perceptual linear prediction.

The power spectrum is computed as follows

$$P(w\omega) = (\Re S(\omega))^2 + (\Im S(\omega))^2 \tag{2}$$

The first step is a conversion from frequency to bark, which represents a better representation of the human hearing resolution in frequency. The bark frequency corresponding to an audio frequency is:

$$\Omega(\omega) = 6\ln\left(\frac{\omega}{1200\pi} + \left(\left(\frac{\omega}{1200\pi}\right)^2 + 1\right)^{0.5}\right) \tag{3}$$

The resulting warped spectrum is convoluted with the power spectrum of the critical band-masking curve, which act like a bank of filters centered on Ω_i. The spectrum is pre-emphasized by an equal loudness curve, which is an approximation to the non-equal sensitivity of human hearing at different frequencies, at about 40dB level. A filter having the following transfer function gives the curve:

$$E(\omega) = \frac{(\omega^2 + 56.8 \times 10^6)\omega^4}{(\omega^2 + 6.3 \times 10^6)^2 \times (\omega^2 + 0.38 \times 10^9)} \tag{4}$$

The last operation prior to the all-pole modelling is the cubic-root amplitude compression (Intensity - Loudness Conversion), which simulates the non-linear relation between the intensity of sound and its perceived loudness. Together with the psychophysical equal-loudness preemphasis, this operation also reduces the spectral amplitude variation of the critical-band spectrum so that the following all-pole modelling can be done by a relatively low model order [4].

Autoregressive modelling is the final stage of the PLP analysis, and consists of approximating the spectrum by an all-pole model, using the autocorrelation method. An Inverse Discrete Fourier Transformation is applied to the spectrum samples, resulting the dual autocorrelation function. For a M-th order all-pole model, only the first M+1 autocorrelation values are needed. The Levinson - Durbin recursive algorithm is used to solve the Yule - Walker equations.

3.2 Image Processing

For our proposed bimodal speech recognition system that will be presented in the application part we had to extract geometric features form the speaker mouth. In order to do that, a face tracking algorithm was used [5].

The algorithm is based on a statistical modeling for the face colors and also is based on using a deformable template to model the oral cavity.

The first stage of the algorithm is to calculate for each pixel in the frame the probability to be in one of the two defined classes: the ’face class’ and the ’non face class’ (background). Than, using the deformable template, we will group all the pixels that are probably in the face class. The deformation of the template is calculated to contain as many pixels of the face class and as less of the non face class. The optimal deformation is searched using the algorithmic search method. For each pixel in the current frame w_1 is the hypothesis that the pixel is in the face class and w_2 that the pixel is in the background class. In order to calculate $P(w_1 \backslash x)$ which is the probability that pixel x to be in the face class, we need first to estimate the color distribution in the face zone $P(x \backslash w_1)$ and in the background zone $P(x \backslash w_2)$. For that a GMM (Gaussian Mixture Model) was used with two components one for each class.

$$P(x \backslash w_i) = \alpha_{i1} \cdot N(\mu_{i1}, C_{i1}) + \alpha_{i2} \cdot N(\mu_{i2}, C_{i2}) \tag{5}$$

where $N(\mu, C)$ is a Gaussian distribution with mean μ and variance C and α_{ij} is the weight of the distribution .

Than we can calculate, using the Bayes rule, the probabilities.

$$\begin{aligned} P(\omega_1 \backslash x) &= P(x \backslash \omega_1) \bullet P(\omega) \\ P(\omega_2 \backslash x) &= P(x \backslash \omega_2) \bullet P(\omega) \end{aligned} \tag{6}$$

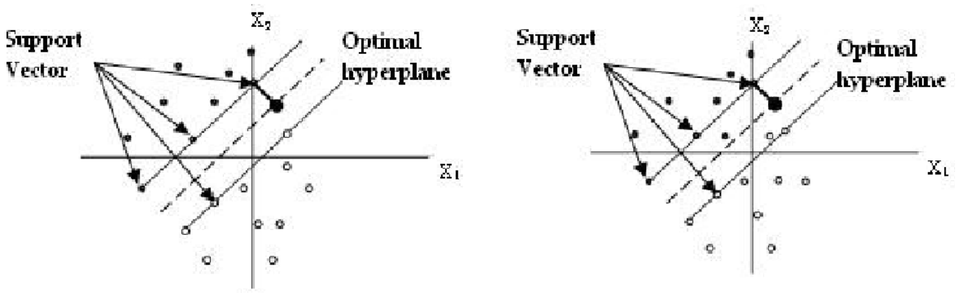
To track the face movement we deformed an elliptic model in order to minimize the energy function (eq. 8) of the region R which contained the face.

$$f(R) = \sum_{r \in R} \log \frac{P(x_r \backslash \omega_2)}{P(x_r \backslash \omega_1)} \tag{7}$$

where r is a pixel in the R region and x_r is the value of the pixel. So the face tracking problem became a problem of minimizing this function by deforming the template.

4 Support Vector Machines

The foundations of Support Vector Machines (SVM) have been developed by Vapnik [6]. The formulation is based on Structural Risk Minimization (SRM) principle, which minimizes an upper bound on the generalization error, as opposed to Empirical Risk Minimization (ERM) which minimizes the error on the training data. Support Vector Machines (SVM) is a statistical algorithm with a great potential to generalize, that can successfully be used in pattern recognition and information retrieval tasks. The main idea in training a SVM system is finding a hyperplane as a decision boundary between two classes. Fundamentally SVM is a binary decision method, but there are several techniques that allow the use in classification tasks with more than 2 classes. There are two possible cases, the case of separable patterns, and the case of non separable patterns, as shown in Fig. 3.



(a) (b)

Fig. 3. Suport vectors for (a) separable patterns (b) nonseparable patterns.

The equation that is verified for each data sample in the case of separable patterns is:

$$d_i(\boldsymbol{w}^T\boldsymbol{x}_i + b) \geq 1 \quad \text{for} \quad i = 1, 2, \ldots, N \tag{8}$$

where d_i is the label for sample data x_i and it can be $+1$ or -1 and w_i and b are the weights and the bias which describe the hyperplane. The support vectors are the data samples for which eq. 9 is verified with the equal sign. After the training process only the support vector will be kept from all data.

In the case of non separable patterns, eq. 9 becomes

$$d_i(\boldsymbol{w}^T\boldsymbol{x}_i + b) \geq 1 - \xi_i \quad \text{for} \quad i = 1, 2, \ldots, N \tag{9}$$

where ξ_i represents the number of data samples left inside the decision area, giving the number of training errors. The problem of finding the optimal hyperplane becomes a problem of minimizing the cost function described by eq. 11

$$\Phi(\boldsymbol{w}, \xi) = \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} + C\sum_{i=1}^{N}\xi_i \tag{10}$$

where minimizing the first term means maximizing the distance between the two classes and minimizing the second term means reducing the number of training errors. Under those circumstances, parameter C becomes a balance between a smaller training error and a bigger distance between classes. The minimization of eq 11 is done using Lagrange multipliers method. Another important part of SVM is the use of the inner product kernel functions. Cover's theorem says that giving a input space where the patterns are non separable, there is a transformation that will lead to another space where with high probability the patterns are separable with two conditions: one, the transformation is non linear and two, the dimension of the output space is high enough. We can use this theorem in solving the Lagrange multipliers systems. We will not calculate the transformation for each data sample in the output space, we will only have to calculate products called inner product kernels, like in eq. 12:

$$\begin{aligned} K(\boldsymbol{x},\boldsymbol{x}_i) &= \varphi^T(\boldsymbol{x}_i)\varphi(\boldsymbol{x}) \\ &= \sum_{j=0}^{m_i} \varphi_j(\boldsymbol{x})\varphi_j(\boldsymbol{x}_i) \quad \text{for} \quad i = 1,2,\ldots,N \end{aligned} \tag{11}$$

We can use any type of kernels: polynomial, radial basis function, two layers perceptron and so on. Fig. 4 gives an example of how a polynomial kernel $(\boldsymbol{x}^T\boldsymbol{x}_i + 1)^p$ works.

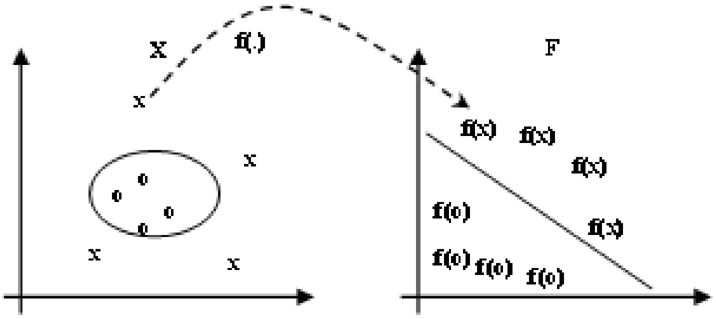


Fig. 4. Polynomial kernel.

Using this kernel arhitecture, SVM can be seen as a NN based system with 3 layers: first input layer with the dimension equal with the number of features of the pattern, than an hidden layer in the future (kernel) space and finally the output layer which will give the binary decision.

5 Multiclass SVM

Like we said in the beginning, SVM is a binary decision method but it can be extended to multiclass task using different algorithms. The most common algorithms use combinations of binary SVMs: 'one against one' method, 'one against all' method and DAGSVM (Directly Acyclic Graph SVM). The oldest method, ’one against all’, consists in building several binary SVMs (equal with the number of classes). In the training phase we will train each SVM with one of the classes against the rest of the classes and in the testing phase we test the test data with all SVMs and the decision is taken based on the distance between data test and the hyperplanes from all SVMs.

’One against one’ method consists in building more binary SVMs where we train each class with another class until we trained each class with all the other classes. In the testing phase we test the current data with all SVMs and if for the classes (i, j) binary SVM the decision is that is in the class i for example the index of i is increased with one and the index of j is decreased with one. In the case of DAGSVM algorithm we start at the top of the graph and if the decision is that the sample is in the i class, then we go to the left path if not we go to the right path and continuing until the end of the tree where we will have the final decision.

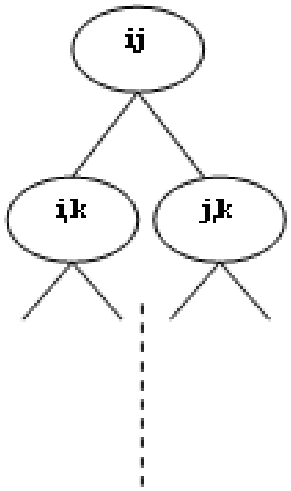


Fig. 5. DAGSVM tree.

6 Experimental results

We used for feature extraction from the speech signal two methods: the perceptual linear prediction (PLP) [7] analysis and the mel-cepstral analysis [8]. For each window, we extract 5 PLP [4] coefficients or 13 mel-cepstral coefficients. For the image sequence we use a face-tracking algorithm and we extract geometric features of the speaker face. For each frame we extract 3 geometric features (the mouth width and the height of the upper lip and downer lip) [5]. For synchronization between image and speech, the video sequence was recorded at 30 fps and we made the length of the analysis window for the speech to be 33ms. So for each frame we will have 3 features from image and 5 PLP or 13 MFC coefficients for speech. For fusion we used simple concatenation between the two feature vectors. Then we formed a 'supervector' putting together the features calculated for each window and we construct bimodal models using those 'supervectors'. For classification we used the DAGSVM algorithm.

We tested our system using database created by the Advanced Multimedia Laboratory from the Carnegie Mellon University. The database contains 10 words (digits from one to ten) spoken by 10 peoples each with 10 pronunciations.

We performed two types of test: first with enrolled speakers which mean that speaker where involved both in training SVMs and testing SVMs. We used five pronunciations for training and five for testing. For the second type of test with unenrolled speakers we used the leave- one- out method. For each word we trained the SVM net with 9 speakers and tested with the 10^{th} repeating the procedure for each speaker. The results are presented in the Table 1 and Table 2

Table 1. Recognition rates for unenrolled speakers.

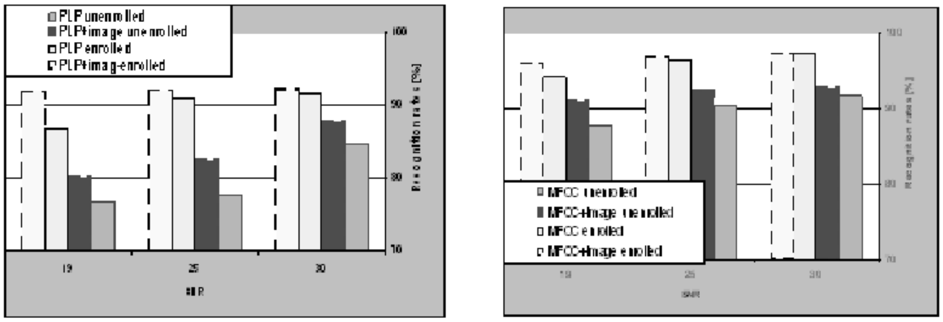
Coefficients(No.)	PLP(5)	MFCC(13)	PLP+image(8)	MFCC+image(16)
SNR=30dB	84.75%	91.71%	87.73%	92.84%
SNR=25dB	77.71%	90.49%	82.42%	92.49%
SNR=29dB	76.8%	87.89%	80.08%	91.13%

Table 2. Recognition rates for enrolled speakers.

Coefficients(No.)	PLP(5)	MFCC(13)	PLP+image(8)	MFCC+image(16)
SNR=30dB	91.71%	97.74%	92.13%	97.42%
SNR=25dB	90.85%	96.53%	91.98%	96.85%
SNR=29dB	86.71%	94.13%	91.85%	96.14%

In Fig. 6 are represented the variations of recognition rates when artificial noise is added over the speech signal.

The performance obtained using bimodal recognition compared with classic unimodal recognition based only on the speech signal is sensible higher, especially under difficult conditions, namely when the speech signal is corrupted with noise. It can be observed that when using coefficients both from image and speech the variations of recognition rates are considerably smaller than when using only speech parameters.



(a)

(b)

Fig. 6. Recognition rates for PLP and MFCC coefficients.

7 Conclusions

In this paper a new approach for building robust speech recognizer systems was presented. The robustness was accomplished by using additional features obtained from the speaker image along with the features obtained from the speech signal. We extract features from the speech signal using the PLP and the mel-cepstral technique and from the image of the speaker we extract geometric features. For classification we used the SVM algorithm which we extended to multiclass decision using the DAGSVM algorithm. The experimental results confirmed the stability of the recognition rates when we added artificial noise over the speech signal. Another observation from the experimental results is that when using the MFC coefficients (best 97.42%) the rate of recognition is higher than when using PLP coefficients (best 91.71%). The difference between recognition rates for the enrolled speakers (best 97.42%) and for unenrolled speakers (best 92.84%) is not so high which indicate that SVM has a good generalization property.

References

[1] S. Haykin, *Neural Networks: A Comprehensive Foundation*. Prentice-Hall, 1999.

[2] G. Costache and I. Gavat, “Multimedia classifier,” in *Proc. ESA-EUSC 2004 Conference*, Madrid, Spain, Mar. 2004.

[3] C. C. Chibelushi, F. Deravi, and J. Mason, “A review of speech based bimodal recognition,” *IEEE Trans. on Multimedia*, vol. 4, pp. 23–38, Mar. 2002.

[4] J. Koehler, N. Morgan, H. Hermansky, H. Hirsch, and T. G, “Integrating RASTA-PLP into speech recognition,” in *Proc. ICASSP9́4*, Adelaide, Australia, Apr. 1994, pp. 421–424.

[5] F. J. Huang and T. Chen, “Real-time lip-synch face animation driven by human voice,” in *Proc. IEEE Workshop on Multimedia Signal Processing*, Los Angeles, California, Dec. 1998.

[6] V. Vapnik, “An overview of statistical learning theory,” *IEEE Trans. on Neural Networks*, vol. 10, pp. 988–1000, Jan. 1999.

[7] H. Hermansky, “Perceptual linear predictive (PLP) analysis of speech,” *J. Acoust. Soc. America*, vol. 87, pp. 1738–1752, Apr. 1990.

[8] I. Gavat, C. Dumitru, G. Costache, and D. Militaru, “Continuous speech recognition based on statistical methods,” in *Proc. of Sped2003 Conference*, Bucharest, Romania, Apr. 2003, pp. 115–127.