Unsupervised Stream-Weights Computation in Classification and Recognition Tasks

Eduardo Sánchez-Soto, Alexandros Potamianos, *Member IEEE* and Khalid Daoudi, *Member IEEE*

***Abstract*— In this paper, we provide theoretical results on the problem of optimal stream weight selection for the two stream classification problem. It is shown that in the presence of estimation or modeling errors using stream weights can decrease the total classification error. Specifically, we show that stream weights should be selected to be proportional to the feature stream reliability and informativeness. Next, we turn our attention to the problem of unsupervised stream weights computation in real tasks. Based on the theoretical results we propose to use models and "anti-models" (class-specific background models) to estimate stream weights. A non-linear function of the ratio of the inter- to intra-class distance is proposed for stream weight estimation. The resulting unsupervised stream weight estimation algorithm is evaluated on both artificial data and on the problem of audio-visual speech classification. Finally the proposed algorithm is extended to the problem of audio-visual speech recognition. It is shown that the proposed algorithms achieve results comparable to the supervised minimum-error training approach for classification tasks under most testing conditions.**

***Index Terms*— Multi-stream weights estimation, Robust speech recognition, Decision fusion.**

I. Introduction

THE problem of fusion or combination of various information sources is central to the machine learning community, especially, for signal processing and pattern recognition applications where a variety of features are available to the classifier. For example, for automatic speech recognition (ASR) and audio-visual speech recognition (AV-ASR) applications, the optimal combination of features extracted from the audio or visual data at different time scales is an open research problem. Features or information sources are often combined in a statistical pattern recognition framework using the notion of "feature streams". A fundamental assumption behind streams is that the information sources/features are independent of each other and thus the probability distribution functions (pdfs) of the two streams can be multiplied to obtain the joint observation pdf. Although this approach is theoretically sound because it minimizes the Bayes error, in the real world the independence assumption rarely holds. In addition, estimation and modeling errors further complicate the problem. It turns out that in the presence of estimation or modeling errors using the joint feature distribution in the Bayes classifier is suboptimal even for independent feature streams, i.e., although the Bayes error is minimized the total classification error (including errors due to poor estimation/modeling) might not be minimal. A practical solution to this problem is to use "stream weights" (exponents weighting the contribution of each stream pdf) in order to reduce the total classification error. Although these weights can be optimally computed in a supervised training setting using a minimum error criterion, the computation of the optimal weights in an unsupervised setting is still an open research problem.

The performance of speech recognition systems has improved significantly in the past decade. However, speech recognition in adverse or "mismatched" conditions is a hard problem, e.g., speech recognition in noise using acoustic models trained in "clean" conditions. As the signal to noise ratio (SNR) decreases additional sources of information, e.g., visual information or noise-robust features, can be used to avoid performance loss. Such features are often combined with traditional mel-frequency cepstrum coefficients (MFCCs) audio features using an information fusion method. Examples of information fusion methods that have been employed for speech processing applications can be found in the literature starting from early-work on the combination of MFCCs with their first and second discrete time derivatives. Another example of a system that combines features with different reliabilities is the work on multi-band ASR [1], where features extracted from certain frequency bands might be more (or less) affected by noise. In [2], features such as rate-of-speech (ROS) and fundamental frequency are used as auxiliary information for ASR. There is also much work in the area of audio-visual speech recognition were audio and visual features are fused, see for example [3]. Visual features have provided consistent ASR performance improvement especially in noisy or mismatched recording conditions.

The selected fusion strategy is characterized by the stage at which the information obtained from the different "modalities" is merged. The simplest approach is to fuse at the feature level. This technique, called early integration (EI), concatenates the features into a single feature vector before classification is performed [4], [5]. Feature selection or feature reduction algorithms, e.g., linear discriminant analysis (LDA), are often applied to reduce the dimension of and the dependencies within the feature vector for EI. Another approach is to perform integration at the decision level. In this approach, called late integration (LI), the classifier scores are combined assuming independence among the information sources. An important detail here is at which "level" the scores are com-

Eduardo Sánchez-Soto was with IRIT-CNRS, Toulouse 31062, France, He is now with France Telecom R&D, 4 rue de Clos Courtel 35512 Cesson Sévigné, France; email: eduardo.sanchezsoto@orange-ftgroup.com.

Alexandros Potamianos is with the Dept. of Electronics & Computer Engineering, Technical Univ. of Crete, Chania 73100, Greece; email: potam@telecom.tuc.gr; tel:+30-28210-37221; fax:+30-28210-37542.

Khalid Daoudi is with IRIT-CNRS, Toulouse 31062, France; email: daoudi@irit.fr; tel: +33-561-557432; Fax: +33-561-556886.

Eduardo Sánchez-Soto was funded by the EU FP6-IST network of excellence "MUSCLE". Alexandros Potamianos and Khalid Daoudi were partially funded by the EU FP6-IST projects "HIWIRE" and "MUSCLE".

bined, i.e., at the frame, word [6], [7] or utterance [5] level. A third approach, called middle integration (MI), allows the recognition system to define specific word or sub-word models and permits synchronous continuous speech recognition [8].

As discussed above, late and middle integration schemes may be suboptimal, both because the stream independence assumption rarely holds and because of the existence of estimation/modeling errors. In such cases, feature streams with higher *reliability* in the estimation process or *informativeness* (for the classification task) should be weighted more in the decision process in order to maximize performance. Therefore a mechanism for weighting the stream contribution in the final decision is needed. Algorithms for computing exponential stream weights can be either supervised, i.e., assume that the speech transcription is known, or unsupervised. Unsupervised algorithms are especially relevant for mismatched training and testing conditions, or when the stream weights vary with time. In [9], the author proposes to use the static and dynamic features of the speech signal as two different streams, which are weighted based on a maximum likelihood training algorithm under two different constraints. Given these constraints the author re-estimates the parameters by maximizing the partial Baum's auxiliary expression as a function of the weights. In [10], [11], [12], [13], a Generalized Probabilistic Descent (GPD) algorithm is used to estimate the stream weights using a minimum error classification criterion. Unsupervised algorithms often compute the stream weights based on reliability estimates of the environmental conditions, e.g., the SNR. In [14], the authors propose to weight dynamically the modalities as a function of the SNR and the phonetic content of the utterance. In [15], the authors present a decision fusion approach for AV-ASR, where the estimates of audio stream reliability and informativeness are based on the degree of voicing present in the utterance. In [7], it is assumed that the reliability and informativeness of each stream has a direct relation with the difference of the probability score among the first N candidates produced by the recognizer. A similar approach is presented in [16]. In [17], the stream weights are estimated by minimizing the misclassification error on a held-out data set. Three stream confidence measures are investigated, namely the stream entropy, the N-best likelihood ratio average, and an N-best stream likelihood dispersion measure. Finally, in [18], stream weights are computed based on likelihood value normalization; weights are selected to maximize the difference between the N-best likelihood scores.

In this paper, we investigate the problem of unsupervised stream weight estimation with an application to audio-visual speech recognition. First the stream weight estimation problem is posed as a probability of classification error minimization problem based on our prior work in [19]. These theoretical results are experimentally verified. Then stream weights estimation algorithms based on the concept of "anti-models" are proposed extending the work in [20]. Extensive evaluation experiments are included that demonstrate the potential of these unsupervised stream weight estimation algorithms for classification and recognition problems. The contributions of this paper are: (i) extensions to the theoretical framework of [19] and experimental verification of these results, and (ii) theoretically motivated unsupervised stream weight estimation algorithms that extend the work of [20], along with detailed experimentation for the problem of AV-ASR.

The organization of this paper is as follows. In Section II, the theoretical underpinnings of the weighted multi-stream classification are presented. These results are experimentally evaluated in Section II-B. In Section III, an algorithm for estimating stream weights in an unsupervised manner is proposed and evaluated for the problem of audio-visual speech recognition. In Section IV, the proposed algorithm is extended to the problem of AV-ASR recognition and evaluated. The paper concludes with Section V.

II. Optimal Stream-Weight Computation

In [19], the two-class w_1, w_2, statistical classification problem with feature pdfs $p(x|w_1)$, $p(x|w_2)$ and class priors $p(w_1)$, $p(w_2)$ is presented. Based on the assumption that the estimation/modeling error for the feature pdfs is a random variable z_i then the deviation of the decision boundary from the optimal Bayes boundary is also a random variable z that is assumed zero-mean Gaussian with variance σ^2.

The classification decision is then a function of the random variable z and the total classification error is computed as

$$P(error) = \int_{\mathbb{R}^d} \int_{f(x)}^{+\infty} \mathcal{N}(z;0,\sigma^2)dz\ p(x|w_2)p(w_2)dx + \int_{\mathbb{R}^d} \int_{-\infty}^{f(x)} \mathcal{N}(z;0,\sigma^2)dz\ p(x|w_1)p(w_1)dx, \quad (1)$$

where $f(x) = p(x|w_2)p(w_2) - p(x|w_1)p(w_1)$.

For multi-stream classification we assume that the feature vector x is broken up into two independent streams x_1, x_2 of dimension d_1 and d_2 respectively, and that the feature "probabilities" are given by

$$p(x|w_i) = \prod_{j=1}^{2} p(x_j|w_i)^{s_j}, \quad (2)$$

where s_1, s_2 are the exponential stream weights, and $\sum_j s_j = 1$. Note that the total error Eq. (1) also holds for a two-stream classifier provided that $f(x)$ is substituted by

$$f(x) = \prod_{j=1}^{2} [p(x_j|w_2)p(w_2)]^{s_j} - \prod_{j=1}^{2} [p(x_j|w_1)p(w_1)]^{s_j}. \quad (3)$$

According to [19], since the total error functional in Eq. (1) cannot be minimized directly, an approximation is to compute weights that minimize the variance of the decision boundary deviation σ^2. By minimizing σ^2 the estimation/modeling error is minimized; however, there are no guarantees that the total error is also minimized. The deviation is shown to be a function of the stream weights as follows:

$$\sigma^2 \sim p(x_1|w_1)^{2s_1} p(x_2|w_1)^{2s_2}\ [s_1^2\sigma_{x_1}^2 + s_2^2\sigma_{x_2}^2], \quad (4)$$

where $\sigma_{x_1}^2$, $\sigma_{x_2}^2$ is the variance of the decision boundary deviation for each stream. The stream deviation variances can

be expressed as a function of the estimation error variance as follows:

$$\sigma_{x_j}^2 = \sum_{i=1}^{2} \sigma_{ij}^2, \tag{5}$$

where σ_{ij}^2 is a variance of the estimation/modeling error for the posterior distribution of the i^{th} class and j^{th} stream. As noted in [19], from the equations above it follows that "*stream weights may reduce estimation error only when either the pdf estimation errors of the single-stream (stand-alone) classifiers are different, i.e., one feature stream is more reliable than the rest, and/or the Bayes error of the single-stream classifiers are different, i.e., one stream contains more information pertinent to the classification problem than the rest.*"

It is also shown that if two streams have the same *informativeness* (equal Bayes classification errors) the stream weights are inversely proportional to the sum of the variances of the pdf estimation error for each of the classes of that given stream $\sigma_{x_j}^2$, i.e., proportional to a measure of the stream *reliability* as it is represented by the next equation:

$$\frac{s_1}{s_2} = \frac{\sum_{i=1}^{2} \sigma_{i2}^2}{\sum_{i=1}^{2} \sigma_{i1}^2} = \frac{\sigma_{x_2}^2}{\sigma_{x_1}^2}. \tag{6}$$

Similarly if two streams are equally reliable (equal estimation error variances) the stream weights should be approximately inversely proportional to the classification error of the single stream classifiers, i.e., proportional to a measure of the stream *informativeness*. Specifically,

$$\frac{s_1}{s_2} \approx \frac{p(x_2|w_1)}{p(x_1|w_1)} \quad \text{for} \quad 0.67 \leq \frac{p(x_1|w_1)}{p(x_2|w_1)} \leq 1.5. \tag{7}$$

where $p(x_1|w_1)$ and $p(x_2|w_1)$ are the observation probabilities for the first and second streams close to the decision boundary.

Combining these two results we conclude that: (i) stream weights should be proportional to the feature stream reliability and informativeness, (ii) the inverse of the variance of the posterior probability estimate is a good measure of reliability, and (iii) the inverse classification error is a good measure of informativeness.

A. First Order Correction

The analysis of [19] is approximate in the sense that the selected weights do not minimize the total classification error but rather the decision boundary deviation variance σ^2. The use of non-equal stream weights however moves the decision boundary away from the Bayes decision boundary, i.e., the value that minimizes Bayes error. As a result, by selecting stream weights using the formulas proposed above the estimation/modeling error will be minimized, but at the same the Bayes error will increase disproportionately. Thus, a first order correction is needed so that the total error (sum of Bayes and estimation/modeling error) is minimized; this correction will bring the stream weights closer to the $s_1 = s_2 = 0.5$ value.

The increase in Bayes error is approximately proportional to the change in the decision boundary namely:

$$\{\prod_{j=1}^{2}[p(x_j|w_2)p(w_2)]^{s_j} - \prod_{j=1}^{2}[p(x_j|w_1)p(w_1)]^{s_j}\} - \{\prod_{j=1}^{2}[p(x_j|w_2)p(w_2)]^{0.5} - \prod_{j=1}^{2}[p(x_j|w_1)p(w_1)]^{0.5}\}. \tag{8}$$

One could attempt the joint minimization of σ^2 and the quantity above, although the relative weighting of the two criteria is non-trivial to compute. Instead we observe that for Gaussian distributions the quantity above is approximately proportional to the deviation from equal weights, i.e., $s_1 - 0.5$ (or equivalently $0.5 - s_2$), if we use the logarithmic discriminative function. Thus, for small deviations from equal weighting a first order correction factor could be applied that is a function of the deviation, as follows:

$$\hat{s}_j = s_j - a(s_j - 0.5) = (1-a)s_j + 0.5\ a\ , \tag{9}$$

where s_j is the original stream weight estimate and $\hat{s}_j$ is the corrected one. The positive factor a can be empirically estimated from experiments. Note, however, that for large deviations from equal weighting higher order corrections might be needed.

B. Numerical Simulations

In this section, we experimentally verify the accuracy of the estimation process outlined above. For this purpose, we design a two-class two-stream classification experiment, where the features follow a Gaussian distribution. It is also assumed that there is a known estimation error[1] for the mean or the variance of these Gaussian distributions. Since both the actual and estimated parameters of the distribution are known, the Bayes and total errors can be computed directly, either for equally weighted feature streams or, in general, for arbitrary stream weights $\{s_1, s_2\}$. Our goal is to compare the optimal stream weights, i.e., the stream weights that minimize the total classification error, with the stream weights computed using Eqs. (6), (7).

The two classes w_1, w_2 classification problem with two streams x_1, x_2 can be visualized in Fig. 1. For a given stream x_j, each class w_i, is modeled with a 1-D Gaussian distribution. The Bayes error in stream x_2 is shown as the shaded region (overlapping area under the two Gaussian distributions). The joint distributions, assuming independence between x_1 and x_2 and equal weights $s_1 = s_2$, are also shown for classes w_1 and w_2.

For example, consider the two-stream two-class classification problem with the true distributions shown in Table I. The *a-priori* class probabilities are assumed equal. Note that the means and variances of the distributions have been selected so that the single stream Bayes error is equal for x_1 and x_2. This

[1] Alternatively, the estimation error could be negligible, and the difference in the actual and estimated parameters could be due to a mismatch between the training and testing conditions, i.e., a modeling error. In general, the errors will be due to both poor estimation and inaccurate modeling.

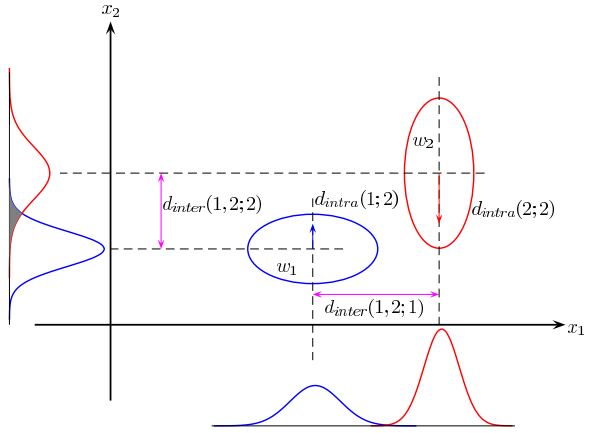

Fig. 1. *Representation, in two dimensions, of the the two classes* $\{w_1, w_2\}$ *classification problem. Each axis represents one stream* $\{x_1, x_2\}$.

is equivalent to saying that $P_1(error) = P_2(error)$, where the Bayes error for each stream is defined as

$$P_j(error) = \int_{\Omega_1} p(x_j|w_2)p(w_2)dx + \int_{\Omega_2} p(x_j|w_1)p(w_1)dx \tag{10}$$

where Ω_1 and Ω_2 are the decision regions for w_1 and w_2 respectively.

TABLE I

Parameters for the two-stream $\{x_1, x_2\}$ *two-class* $\{w_1, w_2\}$ *Gaussian distributions.*

		Class			
		w_1		w_2	
stream	x_1	$\mu_{11} = 4.0$	$\sigma^2_{11} = 2.0$	$\mu_{21} = 6.0$	$\sigma^2_{21} = 1.5$
	x_2	$\mu_{12} = 1.5$	$\sigma^2_{12} = 1.5$	$\mu_{22} = 3.5$	$\sigma^2_{22} = 2.0$

Next, we demonstrate that in the presence of estimation (or modeling) errors, one can significantly reduce the total classification error by using stream weights. In the example shown here, it is assumed that errors exist only in the estimation of the means of the first stream x_1. The total error is computed using Eq. (10) and shown in Fig. 2 as a function of the stream weight s_1 ($s_1 = 0.5$ corresponds to equal stream weights) Three total error curves are shown for estimation error 0, 1 and 3 respectively. As expected, equal weights achieve minimum error if there is no estimation error. However, in the presence of estimation errors, the (more) "corrupted" feature stream should be weighted less in the final decision in order to maximize performance. As shown, the higher the estimation error the smaller the corresponding stream weight s_1 should be.

In the next experiment, we attempt to verify the results of Eq. (6). We assume that for each pdf $\mathcal{N}(\mu_{ij}, \sigma^2_{ij})$ shown at Table I, the pdf estimation error is a random variable z_{ij} that follows a Gaussian zero-mean distribution $\mathcal{N}(z_{ij}; 0, \sigma^2_{z_{ij}})$ according to the assumptions that lead to Eq. (6). The optimal weights are computed empirically by generating $100,000$ samples for each of the pdfs, computing the observation probability for each of the samples and adding random estimation errors to the computed probability that follow a zero-mean Gaussian distribution with variance $\sigma^2_{z_{ij}}$. The optimal weights

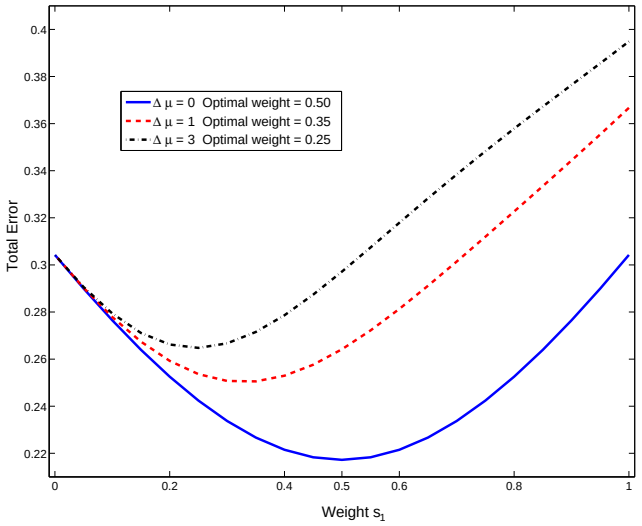

Fig. 2. *Total Error as a function of the "corrupted" stream weight* s_1. *Error curves are provided for absolute estimation errors 0, 1, and 3.*

are then computed in order to maximize classification performance and compared with the estimated weights from Eq. (6). The results are shown at Table II for various probability error variances in the first and second streams. The last column of Table II shows the estimated weights following the application of the first order correction shown in Eq. (9). The value of $a = 0.085$ was chosen to minimize the mean square error between the estimated and optimal weights. Overall, there is good agreement between the optimal and estimated weights, especially after the first order correction is applied.

TABLE II

Optimal and estimated weights for s_1 *as a function of (actual) probability error variance.*

Probability Error Variance				Optimal Weight	Estimated Weight	First Order Correction
$\sigma^2_{z_{11}}$	$\sigma^2_{z_{12}}$	$\sigma^2_{z_{21}}$	$\sigma^2_{z_{22}}$			
0.01	0	0	0.01	0.50	0.50	0.50
0.01	0	0	0.02	0.66	0.67	0.66
0.01	0	0	0.03	0.73	0.75	0.73
0.01	0	0	0.04	0.77	0.80	0.77

In practice, however, estimation errors typically appear in the mean and, especially, the variances of the pdfs. Next, we experiment with various amounts of estimation errors in the variance of the Gaussian distributions of Table I. We use a *stream weight estimate that is inversely proportional to the single stream classifier total error*. Results are presented in Table III. In the first row of of this table, the variances for all four pdfs are increased by the same amount ($+0.5$), resulting in approximately the same pdf error variance and classification errors in both streams. Indeed for this case, both the optimal and estimated weights are equal to 0.5. Then the variance σ^2_{22} is increased further to create an imbalance in the estimation errors between the two streams. As expected, the weight for the less "corrupt" stream s_1 should then be

higher to maximize performance. Note that there is good agreement between the optimal and estimated values for small estimation errors and stream weight ratios close to 1. For large estimation errors, the approximation in Eq. (7) does not hold and the difference between the actual and estimated weights is significant. In general, for cases where the informativeness and/or reliability of the two streams is very different further research is necessary to determine formulas for estimating the optimal stream weights.

TABLE III
Optimal and estimated weights for s_1 as a function of the pdf variance estimation error.

Variance Estimation Error				Optimal Weight	Estimated Weight
σ^2_{11}	σ^2_{12}	σ^2_{21}	σ^2_{22}		
+0.5	+0.5	+0.5	+0.5	0.50	0.50
+0.5	+0.5	+0.5	+1	0.53	0.53
+0.5	+0.5	+0.5	+1.5	0.56	0.55
+0.5	+0.5	+0.5	+2	0.60	0.58
+1	+1	+1	+2	0.60	0.57
+1	+1	+1	+3	0.70	0.62

III. Unsupervised Stream Weight Estimation for Classification Tasks

In real-world applications, we have no access to the true distribution but only to the estimated one. Hence, in this case the estimation/modeling error can not be computed analytically. For example, in audio-visual speech recognition it is common that the recording conditions are both time-varying and different from the conditions under which the models were trained. In this case, the stream weights for the audio and video streams have to be adapted to their optimal values without knowledge of the transcription or "class labels". Our goal here is to devise a robust unsupervised estimation algorithm of the optimal stream weights using small amounts of unlabeled data and based on the theoretical results summarized in Section II. In this practical context, the theoretical results are not directly applicable because of two reasons: theoretical results are available only the two-class classification problem, and for each observation x the knowledge of class membership is required.

A. The two-class problem

It is well known that for the two-class classification problem, when $p(x|w_i)$ follow Gaussian distributions $\mathcal{N}(\mu_i, \sigma^2)$, the Bayes error is a function of $D = |(\mu_1 - \mu_2)|/\sigma$. In general, the quantity D can be estimated in an unsupervised way, by performing k-means classification and then using the inter- and intra-class distances to estimate the quantities in the nominator and denominator respectively [20]. Indeed the inter-class distance is the average distance between the means of each class and the intra-class distance an estimate of the average class variance.

To gain better insight into the use of the inter- and intra-class ratio, we display in Fig. 1 a two-stream two-class classification problem: axis x_1 and x_2 correspond to the features in the two streams; the (Gaussian) distributions for classes w_1 and w_2 are shown for each stream and jointly. The relationship between the Bayes error (shaded area) and the inter- and intra-class distances is approximately inversely and directly proportional respectively.

Overall, the stream weights are computed using the inter-class distance $d_{inter}(1,2;j)$ between classes 1 and 2, normalized by the intra-class distance $d_{intra}(i;j)$ for the class i in each stream. Specifically:

$$\frac{s_1}{s_2} = c\, f\left(\frac{d_{inter}(1,2;j)/\sum_i d_{intra}(i;j)|_{j=1}}{d_{inter}(1,2;j)/\sum_i d_{intra}(i;j)|_{j=2}}\right), \qquad (11)$$

where $f(.)$ is a nonlinear function that relates D with the Bayes error ($erf()$ function) and c is a constant accounting for the difference in estimation error in the two streams[2].

1) Evaluation on synthetic data: In this experiment, we used the parameters shown in Table I for the 1-D Gaussian distributions of the two classes $\{w_1, w_2\}$. A number $N = 250$ of samples was generated using those parameters and the total classification error was computed for different weights. The samples were used to estimate the distributions for the two classes by a clustering process. The k-means algorithm with $k = 2$ was employed to cluster the samples. The estimated clusters were used to compute the distances as shown in Eq. (11). Results are shown in Fig. 3 for two examples. The solid (black) lines represent the estimated distributions, while the thin (blue and red) lines represent the real distributions. A solid (green) line connects the k-mean estimated centroids.

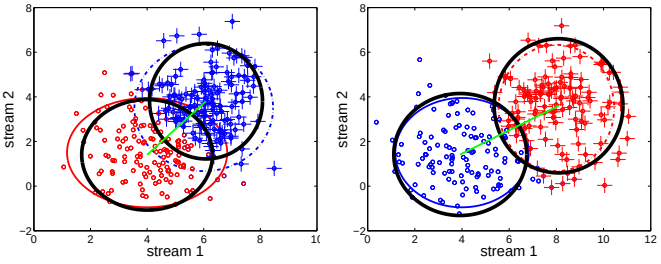

Fig. 3. *Clustering and distance computation representation for the two class problem.*

In the figure on the left, the optimal stream weight that minimized the total error was equal to 0.5, while the estimated weight using the proposed approach was $s_1 = 0.47$ (using $c = 1$ and $f()$ the identity function). In the figure on the right, one of the class means was moved from $\mu_{21} = 6$ to $\mu_{21} = 8$ to introduce additional modeling error. The optimal weight was in this case $s_1 = 0.6$ and the estimated one was 0.65. Overall, for artificial data and for the two-class two-stream problem, the proposed approach gives satisfactory results.

B. The multi-class problem using anti-models

Another issue that that has to be addressed is the generalization of the stream weight estimation process to multiple

[2] The quantity D and the stream error are related through the $erf()$ function only for the ideal case of Gaussian distributions. In general, this relationship might be nonlinear and can be approximated by a polynomial or sigmoid function estimated on held-out data (see also next section).

classes. Currently, theoretical results are available only for the two-class classification problem, while in general the multi-class classification problem is of interest.

To resolve the multi-class problem, we introduce the concept of *anti-models*[3]. Specifically, during training and for each class we separate the training data into two groups: one containing the training examples of the class of interest and the other containing the rest of the training examples. Models and "anti-models" are thus built from the two training sets; anti-models can be thought of as class-specific "background/garbage" models. By creating models and anti-models the multi-class classification problem is reposed as multiple two-class classification problems.

C. Application to audio-visual speech classification

To investigate the potential of our approach, a set of experiments using real data was performed. An audio-visual speech classification task was evaluated with the two feature streams containing audio and visual information respectively.

For the purposes of this experiment the CUAVE audio-visual speech database was employed [22]. The subset of the CUAVE database used in these experiments consists of videos of 36 persons each uttering 50 connected digits utterances. The training set is made up of 30 speakers (1500 utterances) and the test set contains 6 speakers (300 utterances). The audio signal was corrupted by additive babble noise at various SNR levels; the video signal was clean in all experiments. The audio features used were the "standard" Mel-Frequency Cepstrum Coefficients (MFCC) computed for frames with duration 20 ms, extracted every 10 ms. The acoustic vectors with dimension $d_A = 39$, consist of 12-dimensional Mel-frequency cepstral coefficients (MFCCs), energy, and their first and second order derivatives. The visual features were extracted from the mouth region of each video frame by gray-scaling, down-sampling and finally performing a 2-D Discrete Cosine Transform (DCT). The first 13 most "energetic" DCT coefficients within the odd columns were kept [23] resulting in a video feature vector of dimension $d_V = 39$ including the first and second order derivatives. Hidden Markov Models (HMMs) were used for both acoustic and video model training. Context-independent whole-digit models with 8 states per digit and a single Gaussian density distribution per state were used. Each one of the Gaussians is treated separately during the weight computation process. The HTK HMM toolkit was used for training each stream, audio and video, and also for testing (using HTK's built-in multi-stream capabilities).

An important part of the training process is the generation of "anti-models" [21]. The class and anti-class models are both built during the training phase using only "clean" data. The class model for each stream is created following the traditional training process. The anti-class models are trained using all the data that does not belong to the corresponding class. For example, the model for the digit *one* is created using all training data labeled as *one*, while the anti digit model *one* is trained using all the data not labeled as *one*. At the end of this process twenty models are obtained for each stream, ten models for the digits (0-9) and ten anti-digits all with the same number of parameters.

During the test phase these class and anti-class models are used to initialize the k-means classification algorithm. Specifically, the means of the Gaussian distribution in the class and anti-class model are used as the initial k-mean centroids[4]. Given that *a-priori* it is not known to which class each utterance belongs, the features in each utterance are split into two classes ($k = 2$) in ten different ways one for each digit and anti-digit model[5]. The stream weights are estimated using Eq. (11). The inter- d_{inter} and intra-class d_{intra} distance is computed for each of the ten splits and the resulting inter- to intra-class ratio is averaged over the ten splits. Note that the stream weights are estimated for *each utterance*.

Specifically the steps of the stream weight estimation algorithm are as follows[6]. During training:

1) HMM models and anti-models are trained for each digit.
2) For each model and anti-model, the average MFCC vector is computed across **all** states (to be used for k-means initialization in testing).

During testing:

1) For each analysis frame of a test digit the MFCC feature vectors are computed.
2) The k-means ($k = 2$) algorithm is run on the extracted MFCC feature vectors using ten different initializations, one for each model and anti-model pair.
3) The inter- and intra-class distances are computed for the two classes resulting from the k-means algorithm. The process is repeated for each of the ten different k-means initializations.
   - The inter-class distance is computed as the Euclidean distance between the class centroids.
   - The intra-class distance is computed as the average Euclidean pair-wise distance between all class members.
4) The ratio between the inter-class and sum of intra-class distances is computed for each stream and averaged over all ten k-means initializations.
5) The stream weights are computed using Eq. (11).
6) The computed weights are averaged over all digits in an utterance to come up with the per-utterance stream weight estimate.

In Fig. 4, the digit classification results are shown for various stream weight estimation algorithms. The thick solid curve (green) represents the results obtained searching by hand for the optimal values of the weights. The solid curve (black) uses equal weights in both streams (0.5). These two curves serve as reference and are used to evaluate our approach. The

[3] Anti-digit models have been employed in utterance verification [21].

[4] It is important to remark that these anti-class models are only used to initialize the clustering process and that the models are trained using data recorded in "clean"conditions (different than the test conditions).

[5] Note that phone and (especially) word-models consist of multiple states with different acoustic characteristics. If each state is considered a separate class, one could end-up building HMM state and anti-state models. Although, such an approach has its merits, in this work, we treat all states of an HMM model as a single class and build word and anti-word models. The approach is simpler and the (state-level) segmentation problem is avoided.

[6] For simplicity, the processing steps for the audio stream are outlined next. Note that the same steps have to be followed also for the visual stream.

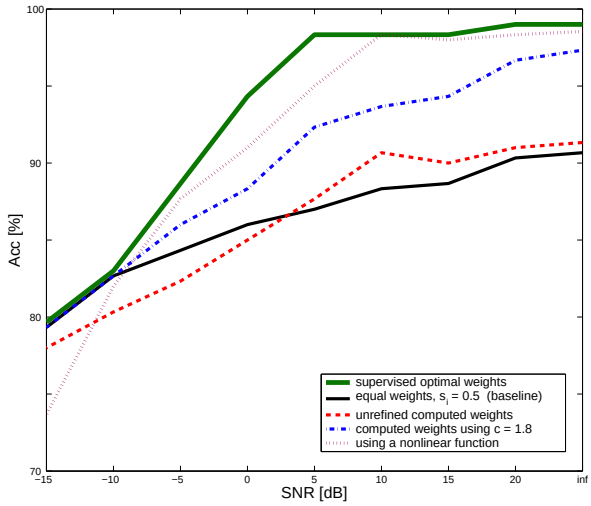

Fig. 4. *Digit accuracy (optimal supervised, baseline and obtained with the proposed method) as a function of SNR for the audio-visual digit classification task.*

first (and crudest) stream weight estimate is shown with the dashed curve (red) and corresponds to Eq. (11) with $c = 1$ and $f()$ being the identity function. To take into account the estimation error a constant c can be estimated on held-out data and used to improve the results; this is represented with the dashed-dotted curve (blue). As seen in Eq. (11), the optimal weights are a non-linear function $f()$ of the distances. The non-linear transformation used here is $f(x) = x^b$, where b is a parameter estimated on held-out data. For this set of experiments, $b = 0.5$ and $f(x) = \sqrt{x}$. The dotted curve (magenta) shows the results obtained using this nonlinear transformation of the weights. This last curve provides a good match between the D value and the Bayes error and results in performance comparable to the hand-picked optimal stream weight values, with the exception of the -15dB SNR data point.

IV. Unsupervised Stream Weight Computation for Recognition Tasks

In this section, we investigate the problem of unsupervised stream weight estimation for audio-visual speech recognition. Although recognition is in principle a more difficult problem than classification, our extension makes it actually easier to implement stream weight computation and without the need of anti-models[7].

We do so based on the observation that larger separation between class distributions in a given stream implies better discriminative power. Concretely, the inter-class distance is computed among all the classes by summing up all pair-wise inter-class distances; the total inter-class distance measures the average separation between classes[8]. The intra-class distances are computed as before to obtain an estimate of the class variance. The inter- and intra-class distances are combined to yield an estimate of the misclassification error for each stream.

As delineated in the previous section the initial centroids are obtained directly from the (HMM) models learned in training. This time the k-means algorithm is performed over all the classes only one time (k this time is the number of HMM models). The total inter-class distance $d_{inter}(j)$ is computed by summing the inter-class distance over all the possible combinations of two classes, as follows:

$$d_{inter}(j) = \sum_{i=1}^{k} \sum_{l=i+1}^{k} d_{inter}(i, l; j), \qquad (12)$$

where $d_{inter}(i, l; j)$ is the inter-class distance between class i and l of stream j, and k is the total number of classes.

Finally, the stream weights are computed using the total inter-class distance $d_{inter}(j)$ normalized by the sum of the intra-class distances $d_{intra}(i; j)$ in the corresponding stream j, as follows:

$$\frac{s_1}{s_2} = c\, f\left(\frac{d_{inter}(j) / \sum_{i=1}^{k} d_{intra}(i; j)|_{j=1}}{d_{inter}(j) / \sum_{i=1}^{k} d_{intra}(i; j)|_{j=2}}\right), \qquad (13)$$

where c is a constant and $f()$ is a nonlinear function.

The stream weight estimation algorithm, also shown in Fig. 5, is summarized next. During training:

1) For each HMM model, the average MFCC vector is computed across **all** model states (to be used for k-means initialization in testing). The process is repeated also for the visual stream.

During testing:

1) For each utterance, the MFCC feature vectors are computed.
2) The k-means algorithm is run on the extracted MFCC feature vectors (k equals number of HMM models, i.e., $k = 10$ for digit recognition).
3) The inter- and intra-class distances are computed for the k classes resulting from the k-means algorithm (see Eq. (12)).
4) Steps 1-3 are repeated for the visual stream and then the stream weights are computed using Eq. (13).

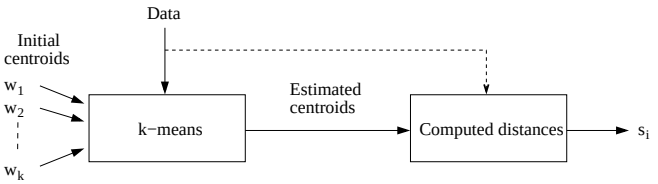

Fig. 5. *Stream weight estimation process.*

[7]In order to use anti-models previously, we assumed that the digit segmentation boundaries are known. One could compute approximately the digit segmentation boundaries in a first recognition pass and employ the algorithm proposed in Section III also for recognition problems. Such a two-pass approach is beyond the scope of this paper; herein we investigate unsupervised stream weight estimation for single pass recognition.

[8] Note that, as before, each HMM model is considered a single class (alternatively each HMM state can be considered a separate class).

A. Application to audio-visual speech recognition

A set of experiments using real data was performed to evaluate the proposed stream weight estimation algorithm. As before, a two stream audio-visual recognition task was investigated using the CUAVE audio-visual speech database [22]. The audio signal was corrupted by additive babble noise at various SNR levels; the video signal was clean in all the experiments. The results are given as a function of the audio SNR ({infinity 20 15 10 5 0 -5 -10 -15} all given in dB). The front-end, HMM topology and train/test data split were identical to those used in the classification experiment. It is important to remark that clean data were used in this experiment for both HMM training and k-mean centroid initialization. Only two iterations of the k-means algorithm were performed to compute the new centroids.

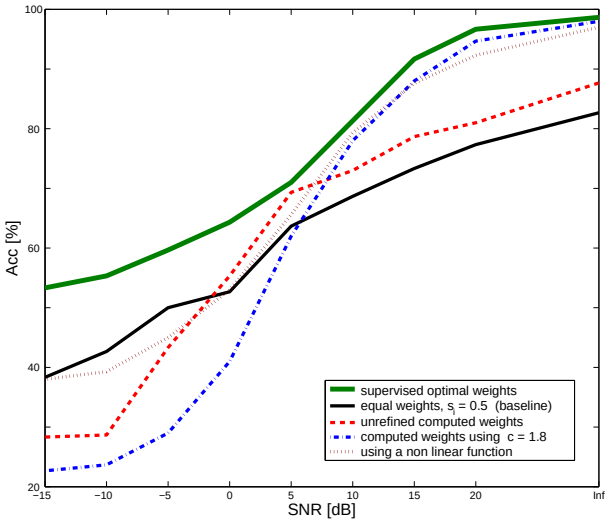

Fig. 6. *Digit accuracy (optimal supervised, baseline and obtained with the proposed method) as a function of SNR for the audio-visual connected digit recognition task.*

The digit accuracy as a function of audio stream SNR for various stream weight estimation methods is presented in Fig. 6. The thick solid curve (green) represents the results obtained searching (by hand) in a supervised manner for the optimal weight values. The solid curve (black) uses equal weights in both streams ($s_1 = s_2 = 0.5$). These two curves serve as reference and are used to evaluate our approach[9]. The first (and crudest) stream weight estimate is shown with the dashed curve (red) and corresponds to Eq. (13) with $c = 1$ and $f()$ being the identity function. Even this crude estimate improves the equal weighting scheme over most SNR values. To take into account the estimation error a constant $c = 1.8$ was estimated on held-out data. The corresponding word accuracy for $c = 1.8$ and $f()$ being the identity function is shown with the dashed-dotted curve (blue). Good performance is achieved over 10dB SNR. However, below 10dB the performance is poor, which shows that (as suggested by the theory and observed in classification) a non-linear mapping function $f()$ is needed for low SNRs. This is confirmed by the word accuracy results shown with a dotted (magenta) curve, where the non linear function $f(x) = \sqrt{x}$ is used. One can see that performance improves further, especially for low SNRs. Finally, we tested the first order weight correction shown in Eq. (9) with $a = 0.085$ (not shown in the plot). A small improvement in performance was obtained for low SNR values, but the improvement was not significant. Overall, the results are satisfactory but not as good as the ones obtained for classification tasks.

One probable explanation for the worse performance of the stream weight estimation algorithm for recognition is the increased number of classes used by the k-means algorithm, i.e., by using models and anti-models $k = 2$ for classification tasks, while for digit recognition $k = 10$ (or in general k is equal to the number of HMM models). Next, we show that indeed the stream weight estimation algorithm is sensitive to the initial choice of k-means centroids. The initial centroids are obtained directly from the mean values of the probability density functions (pdfs) learned during training. These pdfs are estimated on clean data, thus, for low SNR values there is significant mismatch between the computed centroids and the actual data used by the k-mean algorithm. To investigate the importance of the k-means initialization we perform an experiment where the "correct" centroids are used to compute the inter- and intra-distances (and the stream weights). Specifically, we assume that the boundaries of each digit and the digit transcription are known for the test data; the centroid for each class is then computed using the average MFCC vector of each digit occurrence. The results are shown in Fig. 7. The digit accuracy obtained using the "correct" centroids for stream weight estimation (labeled "modified centroids") is compared with the performance of the proposed algorithm. Note that for both cases the centroids are computed on the same test data, the only difference is the initialization of the k-means algorithm.

For all SNR values, the weights estimated using the "correct" centroids outperform, in terms of digit accuracy, the weights estimated using the proposed unsupervised stream weight estimation algorithm. We observe that the difference between the two methods is small for high SNR values, while the difference is significant for low SNRs. Based on these results, it is clear that the initialization of the k-means algorithm plays an important role, i.e., poor initialization results in worse performance. In addition, the performance gap increases for low SNR where the mismatch between the "correct" and initialized (from clean data) centroids is higher. It is important to remark that the centroids are just used to compute the stream weights and that the model pdfs are not modified in this experiment.

V. Conclusions

In this paper, we have presented theoretical and experimental results for the problem of optimal stream weight

[9]When making comparisons between Figs. 4 and 6 bear in mind that results shown are from classification and recognition experiments, respectively. The significantly lower performance for mid and low SNRs in Fig. 6 is due to the many insertions that occur in the visual stream recognizer.

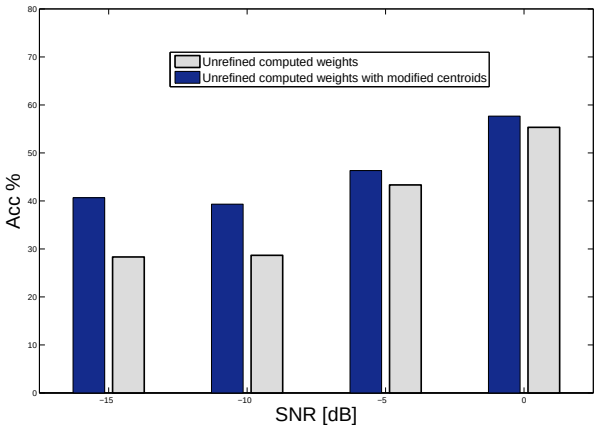

Fig. 7. *Speech recognition accuracy as a function of SNR using k-means initialization from clean data training (unrefined weights) and "correct" k-mean centroids (unrefined weights with modified centroids).*

computation for multi-stream classification and recognition. It was shown that stream weights should be proportional to the feature stream reliability and informativeness to optimize classification performance. Metrics of reliability and informativeness were derived theoretically and tested experimentally. Then a fully unsupervised method for computing stream weights was proposed making use of an "anti-model" technique. The proposed method employs only the information contained in the trained models and requires a single utterance to compute the stream weights. The proposed method achieved comparable performance with supervised minimum error estimation of the weights. Finally, the problem of stream weight estimation for recognition was addressed with good stream weight estimation results; although the performance is worse than for the classification problem where anti-models were employed.

The results are encouraging but more research work is needed on both the theoretical and algorithmic front. Ongoing work includes the extension of the theoretical results to multi-class classification and recognition problem, improved criteria for performing clustering, the use of acoustic model adaptation to improve k-mean centroids initialization, as well as, the extension of the algorithm for the computation of time-varying stream weights.

ACKNOWLEDGMENTS

The authors wish to express their sincere appreciation to Isidoros Rodomagoulakis for independently reproducing the results in this paper using different visual front-end and recognizer setups.

REFERENCES

[1] H. Bourlard and S. Dupont, "A new ASR approach based on independent processing and recombination of partial frequency bands," in *Proc. ICSLP*, Philadelphia, PA, October 1996.

[2] T. Stephenson, M. Mathew, and H. Bourlard, "Modeling Auxiliary Information in Bayesian Network Based ASR," in *Proc. EUROSPEECH*, Aalborg, Denmark, September 2001.

[3] G. Potamianos, C. Neti, G. Gravier, A. Garg, and A. Senior, "Recent Advances in the Automatic Recognition of Audiovisual Speech," *Proceedings of the IEEE*, vol. 91, no. 9, pp. 1306–1326, September 2003.

[4] C. Chibelushi, J. Mason, and F. Deravi, "Integration of acoustic and visual speech for speaker recognition," in *Proc. EUROSPEECH*, Berlin, Germany, September 1993.

[5] S. Dupont and J. Luettin, "Audio-visual speech modeling for continuous speech recognition," *IEEE Transactions on Multimedia*, vol. 2, no. 3, pp. 141–151, September 2000.

[6] M. Heckmann, F. Berthommier, and K. Kroschel, "Noise adaptive stream weighting in audio-visual speech recognition," *EUROASIP Journal on Applied Signal Processing*, vol. 1, no. 11, pp. 1260–1273, November 2002.

[7] A. Adjoudani and C. Benoit, "On the integration of Auditory and Visual Parameters in an HMM-based ASR," *Springer Verlang, Series F: Computer and Systems Sciences*, vol. 150, pp. 465–472, 1996.

[8] J. Luettin, G. Potamianos, and C. Neti, "Asynchronous stream modeling for large vocabulary audio-visual speech recognition," in *Proc. ICASSP*, Salt Lake City, UT, May 2001.

[9] J. Hernando, "Maximum Likelihood weighting of dynamic speech features for CDHMM speech recognition," in *Proc. ICASSP*, Munich, Germany, April 1997.

[10] G. Potamianos and H. P. Graf, "Discriminative Training of HMM Stream Exponents for Audio-Visual Speech Recognition," in *Proc. ICASSP*, Seattle, WA, May 1998.

[11] C. Miyajima, K. Tokuda, and T. Kitamura, "Audio visual speech recognition using MCE-based HMMs and model dependent stream weights," in *Proc. ICSLP*, Beijing, China, October 2000.

[12] G. Potamianos, J. Luettin, and C. Neti, "Hierarchical discriminant features for audio-visual LVCSR," in *Proc. ICASSP*, Salt Lake City, UT, May 2001.

[13] S. Nakamura, K. Kumatani, and S. Tamura, "Robust bi-modal speech recognition based on state synchronous modeling stream weight optimization," in *Proc. ICASSP*, Orlando, FL, May 2002.

[14] A. Rogozan, P. Deléglise, and M. Alissali, "Adaptive determination of audio and visual weights for automatic speech recognition," in *Proc. Workshop on Audio-Visual Speech Processing*, Rhodes, Greece, September 1997.

[15] H. Glotin, D. Vergyri, C. Neti, G. Potamianos, and J. Luettin, "Weighting Schema for Audio-Visual Fusion in Speech Recognition," in *Proc. ICASSP*, Salt Lake City, UT, May 2001.

[16] S. Nakamura, H. Ito, and K. Shikano, "Stream weight optimization of speech and lip image sequence for audio-visul speech recognition," in *Proc. ICSLP*, Beijing, China, October 2000.

[17] G. Potamianos and C. Neti, "Stream Confidence Estimation for Audio-Visual Speech Recognition," in *Proc. ICSLP*, Beijing, China, October 2000.

[18] S. Tamura, K. Iwano, and S. Furui, "A Stream-Weight Optimization Method for Multi-Stream HMMs Based on Likelihood Value Normalization," in *Proc. ICASSP*, Philadelphia, PA, March 2005.

[19] A. Potamianos, E. Sánchez-Soto, and K. Daoudi, "Stream Weight Computation for Multi-Stream Classifiers," in *Proc. ICASSP*, Toulouse, France, May 2006.

[20] E. Sánchez-Soto, A. Potamianos, and K. Daoudi, "Unsupervised stream weight computation using anti-models," in *Proc. ICASSP*, Honolulu, Hawaii, April 2007.

[21] M. Rahim, C. Lee, and B. Juang, "Discriminative Utterance Verification for Connected Digits Recognition," *IEEE Transactions on Speech and Audio Processing*, vol. 5, no. 3, pp. 266–277, May 1997.

[22] E. Patterson, S. Gurbuz, Z. Tufekci, and J. N. Gowdy, "CUAVE: A new audio-visual database for multimodal human-computer interface research," in *Proc. ICASSP*, Orlando, FL, 2002.

[23] G. Potamianos and P. Escanlon, "Exploiting Low Face Symmetry in Appearance-Based Automatic Speechreading," in *Proc. Workshop on Audio-Visual Speech Processing*, British Columbia, Canada, July 2005.

Eduardo Sánchez-Soto received the Engineering degree in Telecommunications from the National Autonomous University of Mexico (UNAM) in 1999, the Master of Sciences (DEA) degree in Signal Processing from the University of Rennes I in 2001, and the Ph.D. degree in Images and Signal Processing from the National Superior School of Telecommunications (ENST) in Paris in 2005. From 2005 to 2007, he was a Postdoctoral Fellow at the Dept. of Electronics & Computer Engineering of the Technical University of Crete and then at the Research Institute in Informatics of Toulouse (IRIT) where he worked on multi-modal data fusion and language recognition. Currently he is with the Research and Developement department at France Telecom working in Audio Quality measurement. His research interests lie mainly in speech processing and statistical modeling.

Alexandros Potamianos (M'92) received the Diploma in Electrical and Computer Engineering from the National Technical University of Athens, Greece in 1990. He received the M.S and Ph.D. degrees in Engineering Sciences from Harvard University, Cambridge, MA, USA in 1991 and 1995, respectively.

From 1991 to June 1993 he was a research assistant at the Harvard Robotics Lab, Harvard University. From 1993 to 1995 he was a research assistant at the Digital Signal Processing Lab at Georgia Tech. From 1995 to 1999 he was a Senior Technical Staff Member at the Speech and Image Processing Lab, AT&T Shannon Labs, Florham Park, NJ. From 1999 to 2002 he was a Technical Staff Member and Technical Supervisor at the Multimedia Communications Lab at Bell Labs, Lucent Technologies, Murray Hill, NJ. From 1999 to 2001 he was an adjunct Assistant Professor at the Department of Electrical Engineering of Columbia University, New York, NY. In the spring of 2003, he joined the Department of Electronics and Computer Engineering at the Technical University of Crete, Chania, Greece as an associate professor.

His current research interests include speech processing, analysis, synthesis and recognition, dialog and multi-modal systems, nonlinear signal processing, natural language understanding, artificial intelligence and multimodal child-computer interaction.

Prof. Potamianos has authored or co-authored over eighty papers in professional journals and conferences. He is the co-author of the paper "Creating conversational interfaces for children" that received a 2005 IEEE Signal Processing Society Best Paper Award. He is the co-editor of the book "Multimodal Processing and Interaction: Audio, Video, Text". He holds four patents. He is a member of the IEEE Signal Processing Society since 1992 and he is currenlty serving his second term at the IEEE Speech Technical Committee.

Khalid Daoudi received both the D.E.A and the Ph.D. degrees in applied mathematics from University Paris 9 Dauphine in 1993 and 1996, respectively. His Ph.D. dissertation was prepared at the Fractals Group of INRIA Rocquencourt, France. During 1997, he held a post-doctoral position at the Department of Mathematics, Ecole Polytechnique de Montral, Canada. From December 1997 to July 1999, he held a post-doctoral position at the Stochastic Systems Group (SSG) of the Laboratory for Information and Decision Systems (LIDS), Massachusetts Institute of Technology (MIT), Cambridge, USA. Since October 1999, he has a permanent position at INRIA Lorraine within the Speech Group. Since October 2003, he is on leave at CNRS with the SAMOVA team of IRIT. His research interests include statistical modeling and estimation, machine learning, Bayesian networks, speech and speaker recognition.