2014 International Joint Conference on Neural Networks (IJCNN)
July 6-11, 2014, Beijing, China

Power Normalized Cepstral Coefficients based supervectors and i-vectors for small vocabulary speech recognition

Emanuele Principi, Stefano Squartini, and Francesco Piazza
Department of Information Engineering, Università Politecnica delle Marche, Ancona, Italy
e-mail: {e.principi,s.squartini,f.piazza}@univpm.it

***Abstract*—Template-matching and discriminative techniques, like support vector machines (SVMs), have been widely used for automatic speech recognition. Both methods require that varying length sequences are mapped to vectors of fixed lengths: in template-matching, the problem is solved by means of dynamic time warping (DTW), while in SVM with dynamic kernels. The supervector and i-vector paradigms seem to represent a valid solution to such a problem when SVM are employed for classification. In this work, Gaussian mean supervectors (GMS), Gaussian posterior probability supervectors (GPPS) and i-vectors are evaluated as features both for template-matching and for SVM-based speech recognition in a comparative fashion. All these features are based on Power Normalized Cepstral Coefficients (PNCCs) directly extracted from speech utterances. The different methods are assessed in small vocabulary speech recognition tasks using two distinct corpora, and they have been compared to DTW, dynamic time alignment kernel (DTAK), outerproduct of trajectory matrix, and PocketSphinx as further recognition techniques to be evaluated. Experimental results showed the appropriateness of the supervector and i-vector based solutions with respect to the other state-of-the art techniques here addressed.**

I. Introduction

In the majority of automatic speech recognizers, acoustic models are represented by hidden Markov models (HMM) [1]. Approaches based on template-matching [2] are also widely studied because of their low storage requirements and their effectiveness when the amount of training data is limited. Generally, in template-matching sequences of different lengths are aligned using dynamic time warping (DTW) [2] and classification is based on the distance with a set of reference patterns. The problems with the original DTW formulation are its high computational burden and the low performance in speaker independent tasks. In the literature, particular attention has been devoted to develop efficient versions of DTW for devices with limited computational resources [3]–[5].

Support Vector Machines (SVMs) have also been extensively employed for recognizing speech [6]. Originally, SVM has been developed to solve binary classification problems of sequences of fixed length, but it can be easily extended to multiclass problems, e.g., using the "one vs one" or the "one vs all" strategies [7]. Its direct employment in speech recognition tasks is not possible, since input utterances are composed of a varying number of feature vectors. The approaches followed in the literature to solve the problem are either based on hybrid SVM/HMM architectures [8] or on dynamic kernels [9]. The latter techniques comprise methods that explicitly map a variable length sequence to fixed length vector (e.g., the Fisher kernel [10], or the outerproduct of trajectory matrix (OTM) [11]) and alignment kernels, e.g., the dynamic time alignment kernel (DTAK) [12] or the dynamic time warping kernel (DTWK) [13].

In this paper, supervectors and i-vectors are evaluated to model variable length utterances in a small vocabulary speech recognition task. The resulting fixed-length vectors are classified with SVM or with a distance metric. Thus, in the latter case, the method represents an alternative to DTW for template-matching speech recognition, while in the former to dynamic kernels for SVM. Mapping is performed by training a Gaussian mixture model (GMM) that represents the acoustic space, and then extracting three set of features: Gaussian mean supervectors (GMS), Gaussian posterior probability supervectors (GPPS), and i-vectors. In particular, i-vectors have been originally proposed for speaker recognition [14], but they have been successfully employed for speech emotion classification [15], accent recognition [16], and acoustic event categorization [17]. Power Normalized Cepstral Coefficients (PNCCs) [18] are employed as low-level features, and the evaluation includes DTW, DTAK, OTM and PocketSphinx [19] in order to compare the proposed approach to recognizers based on template-matching, SVM and HMM. Two corpora have been employed in the experiments: TIDIGITS [20] and ITAAL [21]. The first has been employed to evaluate the performance on a single-digit recognition task using a well-known corpus by the scientific community. ITAAL allows to assess the performance in a more realistic scenario, since it contains utterances spoken in Italian acquired with close-talking and distance-talking microphones. The algorithms have been evaluated in a speaker-independent task, and in a low-resourced speaker-dependent task. The results demonstrate that GMS with SVM outperforms other approaches in the TIDIGITS task and that in the ITAAL tasks i-vectors, either coupled with SVM or with distance-based classification, are able to achieve superior performance respect to DTW, DTAK and PocketSphinx.

The outline of the paper is the following: Section II introduces the speech recognition problem. Section III describes the proposed approach to speech utterance classification. Section IV briefly describe DTAK and OTM as alternatives to the proposed approaches. Section V shows the experiments conducted to evaluate the performance of the approaches. Finally, Section VI concludes the paper and presents future developments.

978-1-4799-1484-5/14/$31.00 ©2014 IEEE

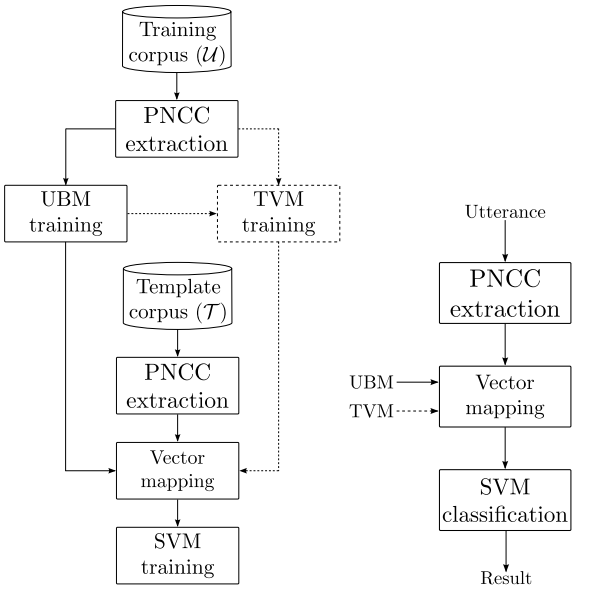
II. Problem formulation

Consider a set of template utterances $\mathcal{T} = \{(\mathbf{X}_1, C_1), (\mathbf{X}_2, C_2), \ldots, (\mathbf{X}_K, C_K)\}$ where $\mathbf{X}_k = \{\mathbf{x}_{k,1}, \mathbf{x}_{k,2}, \ldots, \mathbf{x}_{k,L_k}\}$, $\mathbf{x}_{k,l}$ is the low-level feature vector of utterance k at the time frame index l and C_k is the corresponding label. Given a test utterance $\mathbf{Y} = \{\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_{L_y}\}$, the problem is finding the corresponding label $C_y \in \{C_1, C_2, \ldots, C_K\}$ based on a certain classification criterion. In this work, two classifiers are employed: support vector machines, and distance based classification. Both methods require that sequences of different lengths are mapped to vectors of the same lengths. This paper proposes three methods for utterance length normalization for small vocabulary speech recognition: Gaussian mean supervectors (GMS), Gaussian posterior probability supervectors (GPPS), and i-vectors.

III. The proposed approach

The general scheme of the approach is shown in Fig. 1. PNCCs are employed as low-level features in all the processing steps. A Universal Background Model (UBM) represents the entire acoustic space and it is modelled by means of a Gaussian Mixture Model (GMM). The UBM is trained using a corpus $\mathcal{U} = \{\mathbf{U}_1, \mathbf{U}_2, \ldots, \mathbf{U}_P\}$ where $\mathbf{U}_p$ is an utterance represented by its PNCC feature vectors. When i-vectors are considered, training of the total variability matrix ("TVM" in Fig. 1) is also required. In the figure, i-vectors related processing blocks are depicted with dashed lines. In the "Vector Mapping" block, each template utterance $\mathbf{X}_k \in \mathcal{T}$ is mapped to a GMS, a GPPS or an i-vector. The set of template vectors can be then directly employed for classification using a distance metric, or used for training an SVM. For simplicity, Fig. 1 shows the SVM based approach.

In the classification phase, an input utterance is mapped to the fixed length vector as in the training phase, and then it is classified accordingly.



(a) Training. (b) Classification.

Fig. 1: Scheme of the proposed approach. i-vector related processing blocks are shown with dashed lines. TVM denotes the "total variability matrix".

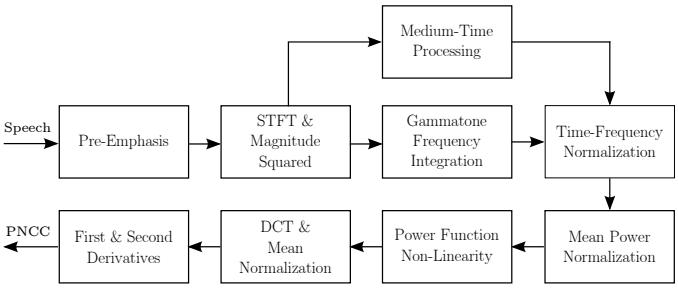


Fig. 2: The PNCC feature extraction pipeline.

A. Power Normalized Cepstral Coefficients

Several techniques have been proposed in the literature for the extraction of low-level features from speech signals. A popular choice for speech recognition tasks is represented by Mel-Frequency Cepstral Coefficients (MFCCs) [22]. However, their recognition performance in noisy and reverberated scenarios is poor, and additional techniques are needed to improve the accuracy. For example, speech enhancement techniques [23], such as spectral subtraction [24] or Ephraim & Malah log-spectral amplitude estimator [25] operate before the feature extraction pipeline. Other approaches, such as Vector Taylor Series speech enhancement [26] or single [27] and multi-channel MFCC-MMSE [28] modify the extraction algorithm, or directly normalize the features statistics [29], [30]. An alternative approach consists in using a different set of features that are intrinsically more robust than MFCCs. Recently, PNCCs [18] have demonstrated their effectiveness at the cost of a modest increment of computational burden. PNCCs have been employed as low-level features in this paper, and they will be now briefly described.

Fig. 2 illustrates the main steps needed for the extraction of PNCCs: the main innovations with respect to MFCCs reside in the replacement of the logarithmic non-linearity with a power function law and the introduction of the "Medium-Time Processing".

The first stages of the extraction pipeline are the same of the MFCC extraction one. The first difference in PNCCs calculation is the replacement of the mel-spaced filterbank with a gammatone one [31]. The motivation behind this choice is that the latter slightly improves the recognition accuracy. The subsequent steps mark the real difference between PNCCs and MFCCs. The "Medium-Time Processing" stage exploits a longer-duration temporal analysis (e.g., 5 frames) to estimate the noise floor level and to subtract it from the instantaneous power of the input signal. Instead of directly using the filtered signal, the output of the "Medium-Time Processing" stage is a transfer function that modulates the original signal in the "Time-Frequency Normalization" step. In the "Mean Power normalization" stage, the signal power is normalized dividing the input by a running average of the overall power. In MFCCs, the logarithm non-linearity is applied to the output of the mel filter-bank. Here, instead, a power function non-linearity with exponent 1/15 is applied. The motivation arises from studies on the non-linear curve that relates the sound pressure level in

dB to the auditory-nerve firing rate. Experiments demonstrated that replacing the logarithmic non-linearity with the power-function one improves the recognition accuracy [18]. The final stages of the PNCC pipeline are the computation of the DCT and the mean normalization as in the MFCC pipeline.

B. Fixed-length vector mapping

1) Gaussian Mean Supervectors: Consider an input utterance $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2 \ldots, \mathbf{x}_L\}$ where L is the number of frames in the utterance and each $\mathbf{x}_l$ is a vector of low-level descriptors (e.g., Mel-Frequency Cepstral Coefficients) of size $D \times 1$. The GMM representing an UBM is given by

$$p(\mathbf{x}_l|\lambda) = \sum_{j=1}^{J} w_j p(\mathbf{x}_l|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j), \tag{1}$$

where $\lambda = \{w_j, \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j | j = 1, 2, \ldots, J\}$, w_j are the mixture weights, and $p(\cdot|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)$ is a multivariate Gaussian distribution with mean vector $\boldsymbol{\mu}_j$ of size $D \times 1$ and diagonal covariance matrix $\boldsymbol{\Sigma}_j$ of size $D \times D$.

The GMS $\mathbf{M}$ of an utterance $\mathbf{X}$ is obtained by adapting the means of the UBM model with maximum a posteriori (MAP) adaptation and then concatenating the mean vectors:

$$\mathbf{M} = [\boldsymbol{\mu}_1^T, \boldsymbol{\mu}_2^T, \cdots, \boldsymbol{\mu}_J^T]^T, \tag{2}$$

where T denotes the transpose operator. Regardless the length of the input utterance, $\mathbf{M}$ is a $DJ \times 1$ vector.

2) Gaussian Posterior Probability Supervectors: Consider an UBM with $J = 1024$ gaussians and an acoustic feature vector with $D = 39$. The resulting GMS is composed of 39936 elements, and this can result in a significant computational burden of the classification stage. GPPS [32] is an effective way of reducing the supervector size, since the final vector has exactly J elements. The GPPS of an input utterance $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2 \ldots, \mathbf{x}_L\}$ is given by:

$$\mathbf{b} = [b_1, b_2, \cdots, b_J]^T, \tag{3}$$

where

$$b_j = \frac{1}{L} \sum_{l=1}^{L} \frac{w_j p(\mathbf{x}_l|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}{\sum_{j'=1}^{J} w_{j'} p(\mathbf{x}_l|\boldsymbol{\mu}_{j'}, \boldsymbol{\Sigma}_{j'})}, \quad j = 1, 2, \ldots, J. \tag{4}$$

Basically, the GPPS vector $\mathbf{b}$ captures the dissimilarity between the input utterance $\mathbf{X}$ and the generic utterance modelled by the UBM.

3) i-vectors: The i-vector technique was originally developed for speaker recognition tasks [14]. In particular, it was noticed that the channel factors estimated in Joint Factor Analysis (JFA) [33] contain information about speaker voices.

In i-vector modelling, the supervector $\mathbf{M}$ of an utterance is decomposed as follows:

$$\mathbf{M} = \mathbf{m} + \mathbf{T}\mathbf{v}, \tag{5}$$

where $\mathbf{m}$ is the UBM supervector, $\mathbf{T}$ is the total variability matrix and $\mathbf{v}$ is the i-vector, a random variable with zero-mean and unit-variance normal distribution.

The following Baum-Welch statistics are needed to extract the i-vector of an input utterance $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2 \ldots, \mathbf{x}_L\}$:

$$\begin{aligned} \mathbf{N}_j &= \sum_{l=1}^{L} p(j|\mathbf{x}_l, \lambda), \qquad \mathbf{F}_j = \sum_{l=1}^{L} p(j|\mathbf{x}_l, \lambda)\mathbf{x}_l, \\ \tilde{\mathbf{F}}_j &= \sum_{l=1}^{L} p(j|\mathbf{x}_l, \lambda)(\mathbf{x}_l - \boldsymbol{\mu}_j), \qquad j = 1, 2, \ldots, J. \end{aligned} \tag{6}$$

Denoting with $\mathbf{N}$ a $JD \times JD$ diagonal matrix with diagonal blocks $\mathbf{N}_j\mathbf{I}$ and with $\tilde{\mathbf{F}}$ a $JD \times 1$ vector obtained concatenating the $\tilde{\mathbf{F}}_j$, the i-vector $\mathbf{v}$ can be calculated as:

$$\mathbf{v} = (\mathbf{I} + \mathbf{T}^T\boldsymbol{\Sigma}^{-1}\mathbf{N}\mathbf{T}^T)^{-1}\mathbf{T}^T\boldsymbol{\Sigma}^{-1}\tilde{\mathbf{F}}. \tag{7}$$

The total variability matrix $\mathbf{T}$ is calculated as the eigenvoice matrix in JFA [34]. The matrix $\boldsymbol{\Sigma}$ is $JD \times JD$ a diagonal covariance matrix calculated during the total variability matrix training process.

The ALIZE toolkit [35] has been employed to extract i-vectors and to train the total variability matrix.

C. Classification

1) Support Vector Machines: SVMs are binary classifiers that discriminate whether an input vector $\mathbf{x}$ belongs to class $+1$ or to class -1 based on the following discriminant function:

$$f(\mathbf{x}) = \sum_{i=1}^{N} \alpha_i t_i K(\mathbf{x}, \mathbf{x}_i) + d, \tag{8}$$

where $t_i \in \{+1, -1\}$, $\alpha_i > 0$ and $\sum_{i=1}^{N} \alpha_i t_i = 0$. The terms $\mathbf{x}_i$ are the "support vectors" and d is a bias term that together with the α_i are determined during the training process of the SVM. The kernel function $K(\cdot, \cdot)$ can assume different forms [36]. In this work, the radial basis function (RBF) kernel $K(\mathbf{x}, \mathbf{x}_i) = \exp(-\gamma\|\mathbf{x} - \mathbf{x}_i\|^2)$ has been employed. The input vector $\mathbf{x}$ is classified as $+1$ if $f(\mathbf{x}) \geq 0$ and -1 if $f(\mathbf{x}) < 0$.

In this work, the multiclass problem has been addressed using the "one versus all" strategy. LIBSVM [37] has been employed both in the training and testing phases of the SVM.

2) Distance-based classification: SVM is a powerful classification technique, but it can require a significant amount of training data in order to achieve satisfactory performance. An alternative solution is deciding whether an input utterance belongs to a class based on the distance between the test utterance and the templates. In this work, a cosine distance scoring has been employed. The cosine distance D_C between two vectors $\mathbf{x}$ and $\mathbf{y}$ is defined as:

$$D_C(\mathbf{x}, \mathbf{y}) = \frac{\mathbf{x} \cdot \mathbf{y}}{\|x\|\|y\|}, \tag{9}$$

where $\cdot$ indicates the dot-product between two vectors.

As stated in Section II, in the template training set $\mathcal{T}$ the same class can be associated to one or more utterances. To obtain a single template vector for one class, all the vectors extracted from utterances belonging to the same class are averaged. Classification is then performed selecting the reference pattern whose distance with the input utterance vector is the smallest.

IV. Alternative approaches

In this section, two alternative methods to the approaches previously described will be briefly reminded. DTAK belongs to the family of dynamic kernels for SVM, and it operates modifying the kernel function of the classifier to deal with variable length input sequences. OTM transforms variable length inputs in fixed-lengths ones, thus it operates similarly to the approaches described in the previous section.

A. Dynamic-Time Alignment Kernel (DTAK)

DTAK has been originally proposed in [12], and basically it consists in modifying the expression of an SVM kernel introducing the DTW distance between two input sequences in the kernel feature space.

More in details, denoting with $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2 \ldots, \mathbf{x}_{L_x}\}$ and $\mathbf{Y} = \{\mathbf{y}_1, \mathbf{y}_2 \ldots, \mathbf{y}_{L_y}\}$ two input sequences, the DTAK kernel $K(\cdot,\cdot)$ coupled with and RBF kernel $K_{RBF}(\cdot,\cdot)$ is given by

$$K(\mathbf{X},\mathbf{Y}) = \max_{\psi_X,\psi_Y} \frac{1}{M_\psi} \sum_{k=1}^{L} m(k) K_{RBF}(\mathbf{x}_{\psi_X(k)}, \mathbf{y}_{\psi_Y(k)}), \tag{10}$$

where $\psi_X(k)$ and $\psi_Y(k)$ are two warping functions subject to

$$1 \leq \psi_X(k) \leq \psi_Y(k+1) \leq L_x, \tag{11}$$

$$1 \leq \psi_Y(k) \leq \psi_Y(k+1) \leq L_y, \tag{12}$$

the term $m(k)$ is a non-negative path weighting coefficient, and M_ψ is a normalization factor usually set to $L_x + L_y$.

As in DTW, the optimization problem of equation (10) is solved by dynamic programming. In particular, the following recursion formula is defined

$$D(i,j) = \max \left\{ \begin{array}{l} D(i,j) + K_{RBF}(\mathbf{x}_i, \mathbf{y}_j), \\ D(i-1,j-1) + 2K_{RBF}(\mathbf{x}_i, \mathbf{y}_j), \\ D(i,j-1) + K_{RBF}(\mathbf{x}_i, \mathbf{y}_j) \end{array} \right\}. \tag{13}$$

Notice that differently from DTW, in (13) the "min" operator is replaced with the "max" one, and the euclidean distance with the RBF kernel. This means that differently from DTW that finds the optimal path that minimizes the accumulated distance, here the algorithm finds the optimal path that maximizes the accumulated similarity [12]. It is worth pointing out that DTAK is a positive semi-definite kernel only under certain conditions [38].

B. Outerproduct of trajectory matrix (OTM)

This method for equalizing the length of speech utterances has been originally proposed in [11]. Given an input utterance composed of L feature vectors of dimension D, the trajectory matrix is an $L \times D$ matrix defined as:

$$\mathbf{U} = [\mathbf{u}_1^T, \mathbf{u}_2^T, \ldots, \mathbf{u}_L^T]^T. \tag{14}$$

The outerproduct trajectory matrix is then defined as:

$$\mathbf{Z} = \mathbf{U}^T\mathbf{U}. \tag{15}$$

Regardless the number of frames L in the input utterance, $\mathbf{Z}$ is a $D \times D$ matrix. Note, also, that $\mathbf{Z}$ is symmetric, thus it contains $D(D+1)/2$ unique elements. The final feature vector $\mathbf{z}$ is a $D(D+1)/2 \times 1$ vector obtained vectorizing the outerproduct trajectory matrix and choosing the unique elements.

Differently from DTAK that directly modifies the SVM kernel, OTM maps each variable length utterance to a fixed-length vector before the classification stage, and thus it can be employed both for training an SVM and for distance-based classification. In the experiments, the first approach will be denoted as "OTM - SVM" and the second "OTM - Distance".

V. Experiments

The proposed approaches have been evaluated on two corpora: TIDIGITS [20] and ITAAL [21]. In TIDIGITS, the task consists in recognizing isolated digits. ITAAL is a recently presented corpus of distress calls and home automation commands in Italian, and it has been employed to assess the performance in a more realistic scenario. In both experiments, the speech signals have been downsampled to 16 kHz and silence portions have been removed using the voice activity detector of the Audio Segmentation Toolkit[1].

Regarding the parameters of feature extraction pipelines, the PNCC pipeline has been configured as follows:

- pre-emphasis coefficient (μ): 0.97;
- frame-size/frame-shift: 25 ms/10 ms;
- number of filters in the gammatone filterbank: 40.

The remaining parameters have been set as in [18].

The evaluation metric employed for evaluating the system performance is the "sentence recognition accuracy", i.e., the ratio between the number of correctly recognized sentences and the total number of sentences.

A. Experiments on the TIDIGITS corpus

The adult set of the TIDIGITS corpus is divided in two subsets, one for training and one testing. The training set contains 112 speakers and it has been further divided in two subsets: one set, $\mathcal{T}$, with single-digits utterances having a total duration of 39.68 s, and the other, $\mathcal{U}$, with sequences of digits utterances having a total duration of 210.80 s. The validation set contains the single-digit utterances of 20 speakers (half males, half females) of the original TIDIGITS test set. The test set of the experiment contains the single-digit utterances of the remaining 83 speakers. The validation set has been employed to obtain all parameters of the algorithms.

The UBM is composed of 8 gaussians and it has been trained on the set $\mathcal{U}$. GMS, GPPS and i-vectors have then been extracted from the set $\mathcal{T}$ with i-vectors having 70 elements. The values of the penalty parameter and the RBF coefficient in all SVM-based approaches (DTAK included) have been selected on the validation set using a grid search as suggested in [37]. The acoustic model in PocketSphinx has been trained on the entire TIDIGITS adult training set (i.e., the set $\mathcal{U} \cup \mathcal{T}$). Each phone is modelled with a 3 states HMM without skip and 8 gaussians per state. The number of tied states has been set to 500. DTAK has been trained on the set $\mathcal{T}$.

TABLE I shows the recognition results. The lowest performing algorithms are DTW and GPPS-based approaches. DTW performance can be explained considering that the

[1] http://gforge.inria.fr/projects/audioseg/

TABLE I: Sentence recognition accuracy on the TIDIGITS single-digit utterances.

	Accuracy (%)
DTW	62.56
DTAK	99.73
PocketSphinx	98.90
GMS - SVM	**99.81**
GMS - Distance	93.58
GPPS - SVM	64.07
GPPS - Distance	45.24
OTM - SVM	99.42
OTM - Distance	95.33
i-vectors - SVM	99.50
i-vectors - Distance	96.60

algorithm suffers in speaker-independent tasks [2]. Regarding GPPS, given the same UBM, these features are not capable of capturing a sufficient amount of information for speech recognition tasks. Experiments have demonstrated that increasing the number of gaussians in the UBM indeed results in better performance for GPPS. "GMS - SVM" and "i-vectors - SVM" provide similar performance and comparable to DTAK and PocketSphinx. "GMS - Distance" and "i-vectors - Distance" accuracies are respectively 6.23% and 2.90% below their SVM counterparts. Note that "GMS - Distance" and "i-vectors - Distance" give superior performance respect to DTW, demonstrating that they are more able to deal with speaker variability.

B. Experiments on the ITAAL corpus

ITAAL[2] is an Italian corpus of home automation commands and distress calls spoken by 20 native Italian speakers (10 males, 10 females) [21]. Each utterance has been acquired with a close-talking microphone and with an array composed of four microphones. The acquisition room had a reverberation time of 0.72 s, the average signal-to-noise ratio of the close-talking microphone signals is 51.46 dB, and the one of the distant microphone signals is 34.08 dB. Each person spoke the corpus sentences standing in front of the microphone array at a distance of 3 m. The corpus is composed of 15 home automation commands, 5 distress calls, each repeated three times and uttered both in normal and shouted conditions. The vocabulary is composed of 24 words. The recognition performance has been evaluated on the close-talking microphone signal and on the central microphone signal of the array. Results are reported separately for commands and distress calls.

Due to the limited amount of data in ITAAL, the UBM has been trained on the "speaker independent" set of the APASCI corpus [39]. This set has a total duration of 174 minutes and it is composed of 2170 phonetically rich sentences uttered by 100 speakers.

1) Speaker independent task: In this task, algorithms have been tested using leave-one out cross-validation with a three-way data split, i.e., using iteratively one speaker for validation, one for testing and the remaining for creating reference patterns. Since speakers in the test set are not included in the training set, the algorithms operate in a "speaker independent" manner as in the previous TIDIGITS experiment.

The number of gaussians in the UBM has been set to 16, and the i-vectors dimension to 250. The PocketSphinx acoustic model has been trained on the same set used to train the UBM. Each phone is modelled with a 3 states HMM without skip and 4 gaussians per state. The number of tied states has been set to 200. In addition, since the proposed approaches operate in matched acoustic conditions, the PocketSphinx acoustic model has been adapted with maximum likelihood linear regression (MLLR) on the same utterances employed for creating templates. Experiments have been performed in matched conditions, i.e., with reference and test sentences uttered with the same vocal effort and acquired with the same microphone. SVM hyper-parameters have been tuned using a grid search on the validation set.

[2]Audio samples are available at http://www.a3lab.dii.univpm.it/projects/itaal

TABLE II: Recognition accuracy (%) in the ITAAL speaker independent task. "H": headset microphone signals. "D": distant microphone signals.

	Commands		**Distress calls**		**Average**
	H	D	H	D	
DTW	67.38	40.49	65.87	48.31	55.51
DTAK	84.28	88.46	91.48	93.96	89.55
PocketSphinx	95.94	37.41	97.67	78.84	77.46
OTM - SVM	85.70	62.99	96.81	85.42	82.73
OTM - Distance	41.76	32.70	62.92	52.22	47.40
GMS - SVM	95.83	79.92	98.61	93.47	91.96
GMS - Distance	81.67	65.32	89.87	83.55	80.10
GPPS - SVM	61.01	54.40	83.09	77.96	69.12
GPPS - Distance	50.79	27.99	70.09	50.33	49.80
i-vectors - SVM	98.05	85.66	99.21	99.10	**95.51**
i-vectors - Distance	93.40	80.14	97.69	95.94	91.79

TABLE III: Recognition accuracy (%) in the ITAAL speaker dependent task. "H": headset microphone signals. "D": distant microphone signals.

	Commands		**Distress calls**		**Average**
	H	D	H	D	
DTW	94.33	89.13	98.39	93.85	93.93
DTAK	33.54	31.72	28.57	27.82	30.41
PocketSphinx	98.50	30.90	99.73	55.88	71.25
OTM - SVM	95.71	85.36	99.58	98.75	94.85
OTM - Distance	92.32	90.18	100.00	98.75	95.31
GMS - SVM	97.14	91.43	100.00	99.58	97.04
GMS - Distance	95.13	94.40	100.00	100.00	97.38
GPPS - SVM	74.11	69.58	91.73	77.60	78.25
GPPS - Distance	63.64	60.73	85.51	67.01	69.22
i-vectors - SVM	97.46	94.19	100.00	100.00	97.91
i-vectors - Distance	97.59	95.20	100.00	100.00	**98.20**

The recognition results are shown in TABLE II. Regarding the proposed approaches, GPPS is the lowest performing as in TIDIGITS. As expected, DTW performs poorly since it operates in a speaker independent task, while DTAK outperforms "OTM - SVM" by 6.82%. On average, the "i-vectors - SVM" solution is the best performing, and in general all the SVM-based approaches outperform the distance-based ones. Notice, however, that differently from the TIDIGITS experiment, DTAK and OTM performance are constantly below "GMS - SVM" and "i-vectors - SVM". The reasons behind this is probably due to the limited amount of data for training the SVM in DTAK, which is compensated by the a-priori knowledge provided by the UBM in GMS and i-vectors approaches. Compared to PocketSphinx, the accuracy of the "i-vectors - SVM" approach is similar when the headset microphone is employed, while is below in the distant microphone case.

2) Speaker dependent task: Each sentence in the ITAAL corpus is repeated three times by each speaker. In this experiment, reference patterns are created from two of the three repetitions and testing is performed on the third. Reference and test repetitions are varied iteratively and the results are averaged. This means that reference and test utterances belong to the same speaker. The objective of this experiment is to assess the performance of the algorithms in a speaker dependent scenario and with a limited amount of training data. The algorithms parameters are the same as in the previous ITAAL experiment.

TABLE III shows the obtained results. Due to the limited amount of data for training the SVM, DTAK performance is considerably below the other approaches. Notice, however, that the same does not hold for "OTM - SVM": this can be explained with the good discriminative capabilities of the OTM features, which are confirmed by the results of "OTM - Distance" approach. A similar behaviour can be observed for "GMS - SVM" and "i-vectors - SVM", whose accuracies are above 90%. This means that they are more robust when the amount of training data is low. Regarding distance-based approaches, DTW accuracy is now significantly better, since it operates in a speaker dependent task, while on average "i-vectors - Distance" performs better then the other algorithms. With the only exception of GPPS, distance based approaches perform better than SVM-based ones. In addition, they perform considerably better than DTW, thus confirming the i-vectors and GMS represent a more effective way to map variable length utterances to fixed-length vectors in template-matching techniques.

VI. Conclusion

In this paper, Gaussian posterior probability, Gaussian mean supervectors and i-vectors have been evaluated in a small vocabulary speech recognition task. PNCCs represent the low-level features that are firstly extracted from the speech signal, and GMM-based background model is trained to map the sequence of PNCCs to a fixed length supervector. In the case of i-vectors, the total variability matrix is trained using the same corpus employed for training the background model. Classification of input utterances has been performed using support vector machines and with cosine-distance. The experiments to assess the performance of the algorithms have been conducted on the TIDIGITS and on the ITAAL corpora. In the first, the performance has been assessed in the single-digit recognition scenario. In the second, the algorithms have been assessed on a smart-home scenario, and with signals acquired both with a close-talking and a distant-talking microphone. Both speaker dependent and speaker independent tasks have been tested. The algorithms have been compared to three popular speech recognition approaches: DTW, DTAK, and PocketSphinx. The results demonstrated the effectiveness of i-vectors and GMS both when classification is performed with SVM and with cosine-distance.

Future works, will consider supervectors and i-vectors extracted from UBM modelled as hidden Markov models instead of Gaussian mixture models [40]. In addition, the algorithm capability to reject out-of-domain sentences will be evaluated. Finally, the robustness of the approaches in mismatched acoustic and vocal effort conditions will be also addressed.

References

[1] G. Saon and J.-T. Chien, "Large-Vocabulary Continuous Speech Recognition Systems: A Look at Some Recent Advances," *IEEE Signal Process. Mag.*, vol. 29, no. 6, pp. 18–33, 2012.

[2] L. Rabiner and B.-H. Juang, *Fundamentals of Speech Recognition*. Prentice Hall PTR, Apr. 1993.

[3] X. Zhang, J. Sun, Z. Luo, and M. Li, "Confidence Index Dynamic Time Warping for Language-Independent Embedded Speech Recognition," in *Proc. ICASSP*, Vancouver, Canada, May 26-31 2013, pp. 8066–8070.

[4] C. Kim and K. D. Seo, "Robust DTW-based Recognition Algorithm for Hand-Held Consumer Devices," *IEEE Trans. Consum. Electron.*, vol. 51, no. 2, pp. 699–709, 2005.

[5] X. Anguera, "Information Retrieval-based Dynamic Time Warping," in *Proc. of Interspeech*, Lyon, France, Aug. 25-29 2013, pp. 1–5.

[6] R. Solera-Ureña, D. Martín-Iglesias, A. Gallardo-Antolín, C. Peláez-Moreno, and F. D. de Mara, "Robust ASR using Support Vector Machines," *Speech Communication*, vol. 49, no. 4, pp. 253–267, 2007.

[7] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.

[8] A. Ganapathiraju, J. Hamaker, and J. Picone, "Hybrid SVM/HMM Architectures for Speech Recognition," in *Proc. of ICSLP*, Beijing, China, Oct. 16-20 2000, pp. 504–507.

[9] V. Wan and S. Renals, "Evaluation of Kernel Methods for Speaker Verification and Identification," in *Proc. of ICASSP*, Orlando, FL, USA, May 2002, pp. 669–672.

[10] T. Jakkola, M. Diekhans, and D. Haussler, "A Discriminative Framework for Detecting Remote Protein Omologies," *J. Comput. Biol.*, vol. 7, no. 1-2, pp. 95–114, 2000.

[11] R. Anitha, D. S. Satish, and C. C. Sekhar, "Outerproduct of Trajectory Matrix for Acoustic Modeling using Support Vector Machines," in *Proc. of IEEE Workshop on Machine Learning for Signal Processing*, Sao Luis, Brazil, 29 Sep.–1 Oct. 2004, pp. 355–363.

[12] H. Shimodaira, K.-I. Noma, M. Nakai, and S. Sagayama, "Dynamic Time Alignment Kernel in Support Vector Machine," in *Proc. of Advances in Neural Information Processing Systems*, vol. 14, 2002, pp. 921–928.

[13] B. Haasdonk and C. Bahlmann, "Learning with distance substitution kernels," in *Proc. 26th Annu. Pattern Recogn. Symp. (DAGM '04)*, Tübingen, Germany, 2004, pp. 220–227.

[14] N. Dehak, P. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-End Factor Analysis for Speaker Verification," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 19, no. 4, pp. 788–798, 2011.

[15] R. Xia and Y. Liu, "Using I-Vector Space Model for Emotion Recognition," in *Proc. of Interspeech*, Portland, OR, USA, Sep. 9-13 2012, pp. 2230–2233.

[16] M.-H. Bahari, R. Saeidi, H. Van hamme, and D. Van Leeuwen, "Accent Recognition Using I-Vector, Gaussian Mean Supervector and Gaussian Posterior Probability Supervector for Spontaneous Telephone Speech," in *Proc. of ICASSP*, Vancouver, Canada, May 26-31 2013, pp. 7344–7348.

[17] Z. Huang, Y.-C. Cheng, K. Li, V. Hautamäki, and C.-H. Lee, "A Blind Segmentation Approach to Acoustic Event Detection Based on I-Vector," in *Proc. of Interspeech*, Lyon, France, Aug. 25-29 2013, pp. 2282–2286.

[18] C. Kim and R. M. Stern, "Power-normalized coefficients (PNCC) for robust speech recognition," in *Proc. of Int. Conf. on Acoustics, Speech, and Signal Processing*, Kyoto, Japan, Mar. 2012, pp. 4101–4104.

[19] D. Huggins-Daines, M. Kumar, A. Chan, A. Black, M. Ravishankar, and A. Rudnicky, "PocketSphinx: A free, real-time continuous speech recognition system for hand-held devices," in *Proc. of ICASSP*, vol. 1, Toulouse, France, May 15-19 2006, pp. 185–188.

[20] R. Leonard, "A Database for Speaker-Independent Digit Recognition," in *Proc. of ICASSP*, vol. 9, Mar. 1984, pp. 328–331.

[21] E. Principi, S. Squartini, F. Piazza, D. Fuselli, and M. Bonifazi, "A Distributed System for Recognizing Home Automation Commands and Distress Calls in the Italian Language," in *Proc. of Interspeech*, Lyon, France, Aug. 25-29 2013, pp. 2049–2053.

[22] S. B. Davis and P. Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," vol. 28, no. 4, 1980, pp. 357–366.

[23] A. Hussain, M. Chetouani, S. Squartini, A. Bastari, and F. Piazza, "Nonlinear speech enhancement: An overview," in *Progress in Nonlinear Speech Processing*, ser. Lecture Notes in Computer Science, Y. Stylianou, M. Faundez-Zanuy, and A. Esposito, Eds. Springer Berlin Heidelberg, 2007, vol. 4391, pp. 217–248.

[24] S. Boll, "Suppression of acoustic noise in speech using spectral subtraction," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 27, no. 2, pp. 113–120, Apr. 1979.

[25] Y. Ephraim and D. Malah, "Speech enhancement using a minimum mean-square error log-spectral amplitude estimator," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-33, p. 443445, Apr. 1985.

[26] V. Stouten, "Robust Automatic Speech Recognition in Time-varying Environments," Ph.D. dissertation, K. U. Leuven, Leuven, the Netherlands, 2006.

[27] D. Yu, L. Deng, J. Droppo, J. Wu, Y. Gong, and A. Acero, "Robust Speech Recognition Using a Cepstral Minimum-Mean-Square-Error-Motivated Noise Suppressor," *IEEE Trans. Audio, Speech, and Lang. Process.*, vol. 16, no. 5, pp. 1061–1070, Jul. 2008.

[28] E. Principi, R. Rotili, S. Cifani, L. Marinelli, S. Squartini, and F. Piazza, "Robust speech recognition using feature-domain multi-channel bayesian estimators," in *Proc. of IEEE International Symposium on Circuits and Systems (ISCAS)*, Paris, France, May 30 - Jun. 2 2010, pp. 2670–2673.

[29] C.-W. Hsu and L. shan Lee, "Higher order cepstral moment normalization for improved robust speech recognition," *IEEE Trans. on Audio, Speech, and Lang. Process.*, vol. 17, no. 2, pp. 205–220, Feb. 2009.

[30] S. Squartini, M. Fagiani, E. Principi, and F. Piazza, "Multichannel cepstral domain feature warping for robust speech recognition," in *Neural Nets WIRN10: Proceedings of the 20th Italian Workshop on Neural Nets*, ser. Frontiers in Artificial Intelligence and Applications, B. Apolloni, S. Bassis, A. Esposito, and F. C. Morabito, Eds. Amsterdam, The Netherlands: IOS Press, 2011, vol. 226, pp. 284–292.

[31] R. D. Patterson, K. Robinson, J. Holdsworth, D. McKeown, C. Zhang, and M. Allerhand, "Complex sounds and auditory images," in *Auditory physiology and perception*, Y. Cazals, L. Demany, and K. Horner, Eds., vol. 83. Oxford, UK: Oxford: Pergamon Press, 1992, pp. 429–446.

[32] X. Zhang, X. Xiao, H. Wang, H. Suo, Q. Zhao, and Y. Yan, "Speaker Recognition using a Kind of Novel Phonotactic Information," in *Proc. of ISCSLP*, Kunming, China, Dec. 16-19 2008, pp. 1–4.

[33] P. Kenny, G. Boulianne, P. Ouellet, and P. Dumouchel, "Joint Factor Analysis Versus Eigenchannels in Speaker Recognition," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 15, no. 4, pp. 1435–1460, 2007.

[34] P. Kenny, G. Boulianne, and P. Dumouchel, "Eigenvoice Modeling with Sparse Training Data," *IEEE Trans. Speech Audio Process.*, vol. 13, no. 3, pp. 345–354, 2005.

[35] A. Larcher, J.-F. Bonastre, B. Fauve, K. A. Lee, C. Lévy, H. Li, J. S. Mason, and J.-Y. Parfait, "ALIZE 3.0 - Open Source Toolkit for State-of-the-Art Speaker Recognition," in *Proc. of Interspeech*, Lyon, France, Aug. 25-29 2013, pp. 2768–2772.

[36] C. Bishop, *Pattern Recognition and Machine Learning*. New York: Springer Science+Business Media, LLC, 2006.

[37] C.-C. Chang and C.-J. Lin, "LIBSVM: a library for support vector machines," *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 3, p. 27, 2011.

[38] M. Cuturi, J. P. Vert, O. Birkens, and T. Matsui, "A kernel for time series based on global alignements," in *Proc. ICASSP*, Honolulu, HI, USA, Apr. 2007, pp. 413–416.

[39] B. Angelini, F. Brugnara, D. Falavigna, D. Giuliani, R. Gretter, and M. Omologo, "Automatic segmentation and labeling of english and italian speech databases," in *Proc. of Eurospeech*, Berlin, Germany, Sep. 22-25 1993, pp. 653–656.

[40] C. Dong, Y. Dong, J. Li, and H. Wang, "Support Vector Machines Based Text Dependent Speaker Verification Using HMM Supervectors," in *Proc. of Odyssey*, Stellenbosch, South Africa, Jan. 21-24 2008.