GMM-based classification from noisy features

Alexey Ozerov[1]*, Mathieu Lagrange*[2] *and Emmanuel Vincent*[1]

[1]INRIA, Centre de Rennes - Bretagne Atlantique
[2] STMS Lab IRCAM - CNRS - UPMC

alexey.ozerov@inria.fr, mathieu.lagrange@ircam.fr, emmanuel.vincent@inria.fr

Abstract

We consider Gaussian mixture model (GMM)-based classification from noisy features, where the uncertainty over each feature is represented by a Gaussian distribution. For that purpose, we first propose a new GMM training and decoding criterion called *log-likelihood integration* which, as opposed to the conventional *likelihood integration* criterion, does not rely on any assumption regarding the distribution of the data. Secondly, we introduce two new Expectation Maximization (EM) algorithms for the two criteria, that allow to learn GMMs directly from noisy features. We then evaluate and compare the behaviors of two proposed algorithms with a categorization task on artificial data and speech data with additive artificial noise, assuming the uncertainty parameters are known.

Experiments demonstrate the superiority of the likelihood integration criterion with the newly proposed EM learning in all tested configurations, thus giving rise to a new family of learning approaches that are insensitive to the heterogeneity of the noise characteristics between testing and training data.
Index Terms: Uncertainty-based classification, Gaussian mixture model, expectation maximization algorithm

1. Introduction

Classification and detection systems generally have to face a wide variety of data distortion phenomena. This results in noisy observations or features that the system has to handle as robustly as possible. In this paper, we focus on classification from noisy data, whatever the type of noise (e.g., additive or convolutive). While our approach is quite general, we mostly consider classification of audio data in the experimental part and in the examples throughout the paper.

In order to reduce the sensitivity of the classifier to noise, many approaches can be taken at different levels. At the signal level, one can respectively apply noise suppression [1] or source separation [2] techniques, for stationary or nonstationary additive noise. At the feature level, one can define features that are robust to certain noises (e.g., additive or convolutive noise) [3] or to the interferences and artifacts produced by a noise suppression or a source separation algorithm applied at the signal level. Finally, at the classifier level, one can account for possible distortion of the features, given some information about this distortion [4, 5]. In this paper, we focus on the latter approach considering a Gaussian mixture model (GMM)-based generative classification task [6].

When facing non-stationary distortions, the features are sometimes completely masked. In this case, the features are effectively missing and should be disregarded. Assuming that the GMMs have been trained from clean data, some works have derived decoding systems that are able to cope with the so-called *missing data* problem [4, 7, 8, 9].

Even though the availability of clean data for model training is a reasonable assumption in some scenarios, many others have to deal with noisy data at the learning stage. Indexing television or radio shows is an example, where we need to recognize which speaker is speaking at a given time. In this case, recordings of the speakers of interest in a controlled environment are usually not available. Moreover, one could also want to adapt available models (potentially learned on clean data) to the noisy data received during the decoding or to learn them directly from noisy data. In that case, one needs to define learning algorithms that are able to cope with the missing feature problem such as the one proposed in [10].

In practice, the features are usually neither completely clean nor completely masked. In order to benefit from this range of uncertainty, one may represent the *uncertainty* over the features by a *probability distribution*. For each feature we here assume a Gaussian distribution with its mean and covariance matrix representing, respectively, the expected value of the feature and its uncertainty. Again, assuming that the GMMs have been trained on clean data, one only needs to take the uncertainty into account at the decoding stage as proposed in [5, 11]. However, the approach in [5, 11] suffers from the following issues:

1. it is assumed that the clean data underlying the noisy observations have been generated by the GMMs, and
2. uncertainty is taken into account only at the decoding stage, assuming that the GMMs were trained from some clean data that are not always available, as mentioned above.

We address the first issue by defining a new criterion for GMM learning and decoding that, as opposed to the conventional criterion from [5, 11], does not rely on any assumption regarding the distribution of the data. To address the second issue, we here derive two new Expectation Maximization (EM) algorithms [12] allowing learning GMMs from noisy data with Gaussian uncertainty for the conventional and the newly proposed criteria, respectively. The first EM algorithm generalizes both the algorithm proposed in [10], that is restricted to binary uncertainty, and the algorithm proposed in [13]. The algorithm in [13] was not investigated in the context of classification and it is restricted to the case of uncertainty with diagonal covariances and zero-mean GMMs with diagonal covariances.

The remaining of the paper is organized as follows. After a brief review of uncertainty decoding in Section 2, we introduce in Section 3 the two criteria to be optimized for training and classification over noisy data as well as the corresponding algorithms derived from the EM framework. Their behaviors

This work was supported in part by the Quaero Programme, funded by OSEO.

are then evaluated and compared in Section 4 with a categorization task on artificial data and speech data with additive artificial noise.

2. GMM decoding from noisy data

Classification is the problem of assigning a sequence of observation vectors $\mathbf{x} = \{\mathbf{x}_n\}_{n=1}^N$ ($\mathbf{x}_n \in \mathbb{R}^M$) to a class C. Each class C is modeled by a GMM $\theta = \{\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i, \omega_i\}_{i=1}^I$ [1], where $i = 1, \ldots, I$ are state indices, and $\boldsymbol{\mu}_i$, $\boldsymbol{\Sigma}_i$ and ω_i ($\sum_i \omega_i = 1$) are respectively the mean, the covariance matrix and the weight of the i-th state. In other words, each vector $\mathbf{x}_n$ is modeled as follows:

$$(\mathbf{x}_n | q_n = i) \sim \mathcal{N}(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i), \qquad \mathbb{P}(q_n = i) = \omega_i, \quad (1)$$

where q_n denotes the state at time n. Under this model the likelihood of the observation sequence $\mathbf{x}$ is given by

$$p(\mathbf{x}|\theta) = \prod_{n=1}^N p(\mathbf{x}_n|\theta), \quad (2)$$

with

$$p(\mathbf{x}_n|\theta) = \sum_{i=1}^I \omega_i N(\mathbf{x}_n|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i), \quad (3)$$

where

$$N(\mathbf{x}_n|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i) \triangleq \frac{1}{\sqrt{(2\pi)^M |\boldsymbol{\Sigma}_i|}} \left[-\frac{(\mathbf{x}_n - \boldsymbol{\mu}_i)^T \boldsymbol{\Sigma}_i^{-1} (\mathbf{x}_n - \boldsymbol{\mu}_i)}{2} \right]. \quad (4)$$

Within this framework the common generative classification approach [6] consists in the following two steps:

1. *Training:* For each class C the corresponding parameters θ are estimated from some sequence of training vectors by maximizing the likelihood (1).
2. *Classification:* An observation sequence $\mathbf{x}$ is assigned to the class C for which the likelihood (1) is maximum.

Note, however, that one does not need the GMMs to represent perfectly the data distribution: the most important for classification is to obtain a good separator between the classes. This was confirmed by considering discriminative training of GMMs for classification [14].

2.1. Binary uncertainty

In the case where some components $\mathbf{x}_n$ are missing and the remaining are available ($\mathbf{x}_n = [\mathbf{x}_n^a, \mathbf{x}_n^m]$), the probabilistic likelihood $p(\mathbf{x}_n|\theta)$ (3) cannot be evaluated in the usual manner. A first approach, called marginalization, is to approximate the likelihood by $p(\mathbf{x}_n|\theta) \simeq p(\mathbf{x}_n^a|\theta)$ [10]. A second approach, called data imputation, is to interpolate missing values so that the likelihood can be computed in the usual manner ($p(\mathbf{x}_n|\theta) \simeq p([\mathbf{x}_n^a, \tilde{\mathbf{x}}_n^m]|\theta)$). Numerous imputation methods can be considered from the simplest one, i.e., interpolating a given component with the mean value of this component over the training set, to more refined ones, like drawing interpolated values from the GMM conditionally to the evaluated class [4].

The first approach is theoretically more seducing and empirically demonstrated to be more powerful in many cases as it roots the classification decision over observed data only. However, it should be noticed that it requires an adaptation of the classification mechanism which may not be feasible in some situations. In the latter case, only imputation is feasible.

[1] For the sake of brevity we omit here the class label C in the set of the model parameters θ.

2.2. Probabilistic uncertainty

By contrast with the binary uncertainty framework, we assume that $\mathbf{x}_n$ is unknown and distributed as

$$\mathbf{x}_n \sim \mathcal{N}(\mathbf{y}_n, \bar{\boldsymbol{\Sigma}}_n), \quad (5)$$

where the parameters $\mathbf{y}_n$ and $\bar{\boldsymbol{\Sigma}}_n$ are known.

For example, $\mathbf{y}_n$ can be a feature computed from a distorted signal and $\bar{\boldsymbol{\Sigma}}_n$ an estimate of the corresponding zero-mean distortion [2] covariance. Alternatively, $\mathbf{y}_n$ can be a feature computed from a signal provided by some source separation algorithm or an estimate of this feature obtained using the underlying source separation model [15]. In this case $\bar{\boldsymbol{\Sigma}}_n$ is a covariance matrix of the corresponding estimation error. In this paper we have the latter application in mind rather than the former.

Since $\mathbf{x}$ is unknown, one cannot directly compute the likelihood (2), and this quantity should be redefined so as to take the uncertainty specified by (5) into account. We consider two cases presented in the following subsections.

2.2.1. Likelihood integration

In a generative framework, the uncertainty equation (5) can be also rewritten as

$$\mathbf{y}_n \sim \mathcal{N}(\mathbf{x}_n, \bar{\boldsymbol{\Sigma}}_n), \quad (6)$$

or, in other words, it is assumed that

$$\mathbf{y}_n = \mathbf{x}_n + \mathbf{e}_n, \quad (7)$$

where $\mathbf{e}_n$ is a zero-mean Gaussian noise with covariance matrix $\bar{\boldsymbol{\Sigma}}_n$ that is independent from $\mathbf{x}_n$ distributed as specified by (3). Indeed, it is reasonable to assume that the noise and the features are independent. With these assumptions the joint likelihood of $\mathbf{x}$ and $\mathbf{y}$ is well-defined and it can be written as follows:

$$p(\mathbf{x}, \mathbf{y}|\theta, \bar{\boldsymbol{\Sigma}}) = p(\mathbf{x}|\theta) p(\mathbf{y}|\mathbf{x}, \bar{\boldsymbol{\Sigma}}), \quad (8)$$

where $\mathbf{y} = \{\mathbf{y}_n\}_{n=1}^N$ and $\bar{\boldsymbol{\Sigma}} = \{\bar{\boldsymbol{\Sigma}}_n\}_{n=1}^N$. The corresponding Bayesian network is shown in Figure 1.

The *likelihood integration* approach consists in marginalizing the joint likelihood (8) over the missing features $\mathbf{x}$, i.e., in considering the likelihood of $\mathbf{y}$. The likelihood $p(\mathbf{x}|\theta)$ from (2) is replaced by the following objective function [5, 11]:

$$f_{\mathrm{LI}}(\mathbf{y}, \bar{\boldsymbol{\Sigma}}|\theta) = \int_{\mathbb{R}^{M \times N}} p(\mathbf{y}|\mathbf{x}, \bar{\boldsymbol{\Sigma}}) p(\mathbf{x}|\theta) d\mathbf{x} \quad (9)$$

$$= \prod_{n=1}^N \sum_{i=1}^I \omega_i N(\mathbf{y}_n|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i + \bar{\boldsymbol{\Sigma}}_n). \quad (10)$$

This approach is compatible with the marginalization approach for binary uncertainty [10] (see Sec. 3.3).

2.2.2. Log-likelihood integration

As a *generative approach*, likelihood integration assumes that the distribution of the hidden data $\mathbf{x}_n$ is accurately modeled by the GMM of the corresponding class, which may not always be the case in practice. We propose a new approach called *log-likelihood integration* that is totally *data-driven*. It does not rely on any assumption regarding the distribution of $\mathbf{x}$, but makes as if all values encoded by (5) were actually observed. Note also

[2] Note that considering zero-mean distortion does not mean reducing the generality of the approach. Indeed, in case of distortion with non-zero mean $\hat{\boldsymbol{\mu}}_n$ one should simply consider $\mathbf{y}_n - \hat{\boldsymbol{\mu}}_n$ instead of $\mathbf{y}_n$.

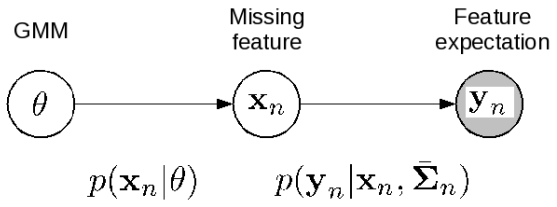

Figure 1: Bayesian network representing GMM-modeled missing features under the uncertainty model (6).

that, in contrast to the likelihood integration, this approach does not have any probabilistic formulation that can be represented by a Bayesian network.

The log-likelihood $\log p(\mathbf{x}|\theta)$ from (2) is replaced by its expectation over the observations (5), which results in the following objective function:

$$f_{\text{LLI}}(\mathbf{y}, \bar{\mathbf{\Sigma}}|\theta) = \mathbb{E}_{\mathbf{x}}\left[\log p(\mathbf{x}|\theta)|\mathbf{y}, \bar{\mathbf{\Sigma}}\right] = \int_{\mathbb{R}^{M\times N}} p(\mathbf{x}|\mathbf{y}, \bar{\mathbf{\Sigma}}) \log p(\mathbf{x}|\theta) d\mathbf{x} = \sum_{n=1}^{N} \int_{\mathbb{R}^M} p(\mathbf{x}_n|\mathbf{y}_n, \bar{\mathbf{\Sigma}}_n) \log \sum_{i=1}^{I} \omega_i N(\mathbf{x}_n|\boldsymbol{\mu}_i, \mathbf{\Sigma}_i) d\mathbf{x}_n. \quad (11)$$

Unfortunately, a closed form computation of the integral in (11) is only possible in the case of a single-state ($I = 1$) GMM. An approximate expression may be obtained by assuming that all data $\mathbf{x}_n$ drawn from (5) in a given time frame n correspond to the same (unknown) state i, resulting in

$$f_{\text{LLI}}(\mathbf{y}, \bar{\mathbf{\Sigma}}|\theta) \approx \sum_{n=1}^{N} \log \int_{\mathbb{R}^M} p(\mathbf{x}_n|\mathbf{y}_n, \bar{\mathbf{\Sigma}}_n) \sum_{i=1}^{I} \omega_i N(\mathbf{x}_n|\boldsymbol{\mu}_i, \mathbf{\Sigma}_i) d\mathbf{x}_n = \sum_{n=1}^{N} \log \sum_{i=1}^{I} \omega_i N(\mathbf{y}_n|\boldsymbol{\mu}_i, \mathbf{\Sigma}_i) e^{-\frac{1}{2}\text{tr}\left(\mathbf{\Sigma}_i^{-1}\bar{\mathbf{\Sigma}}_n\right)}. \quad (12)$$

Note also that, using Jensen's inequality [12], this approximation can also be shown to be an upper bound of (11).

3. GMM learning from noisy data

As discussed in the introduction, there are many application scenarios where it is desirable to train the models over noisy data.

3.1. Likelihood integration

To optimize criterion (10) we propose an EM algorithm, considering the true features $\mathbf{x}$ and the GMM state indices $\mathbf{q} = \{q_n\}_{n=1}^N$ as latent variables. The resulting EM updates are summarized in Algorithm 1.

3.2. Log-likelihood integration

To optimize criterion (12) we propose an EM algorithm, considering only the GMM state indices $\mathbf{q} = \{q_n\}_{n=1}^N$ as latent variables. The resulting EM updates are summarized in Algorithm 2.

Algorithm 1 One iteration of the EM algorithm for the likelihood integration-based GMM learning from noisy data.

E step. Conditional expectations of natural statistics:

$$\gamma_{i,n} \propto \omega_i N(\mathbf{y}_n|\boldsymbol{\mu}_i, \mathbf{\Sigma}_i + \bar{\mathbf{\Sigma}}_n), \quad \text{and} \quad \sum\nolimits_i \gamma_{i,n} = 1, \quad (13)$$

$$\hat{\mathbf{x}}_{i,n} = \mathbf{W}_{i,n}\left(\mathbf{y}_n - \boldsymbol{\mu}_i\right) + \boldsymbol{\mu}_i, \quad (14)$$

$$\widehat{\mathbf{R}}_{\mathbf{xx},i,n} = \hat{\mathbf{x}}_{i,n}\hat{\mathbf{x}}_{i,n}^T + \left(\mathbf{I} - \mathbf{W}_{i,n}\right)\mathbf{\Sigma}_{\mathbf{x},i}, \quad (15)$$

where

$$\mathbf{W}_{i,n} = \mathbf{\Sigma}_i\left[\mathbf{\Sigma}_i + \bar{\mathbf{\Sigma}}_n\right]^{-1}. \quad (16)$$

M step. Update GMM parameters:

$$\omega_i = \frac{1}{N}\sum_{n=1}^{N}\gamma_{i,n}, \quad (17)$$

$$\boldsymbol{\mu}_i = \frac{1}{\sum_{n=1}^{N}\gamma_{i,n}}\sum_{n=1}^{N}\gamma_{i,n}\hat{\mathbf{x}}_{i,n}, \quad (18)$$

$$\mathbf{\Sigma}_i = \frac{1}{\sum_{n=1}^{N}\gamma_{i,n}}\sum_{n=1}^{N}\gamma_{i,n}\widehat{\mathbf{R}}_{\mathbf{xx},i,n} - \boldsymbol{\mu}_i\boldsymbol{\mu}_i^T. \quad (19)$$

3.3. Discussion

These two algorithms were derived using different sets of latent variables: $\{\mathbf{x}, \mathbf{q}\}$ for likelihood integration and $\mathbf{q}$ for log-likelihood integration. This is due to the fact that the complete data criterion corresponding to criterion (12) can be optimized in closed form, considering only $\mathbf{q}$ as latent variables, while it is not possible in general for criterion (10). Note also that both EM algorithms reduce to the classical EM algorithm for GMM learning [12] when the covariance matrices $\bar{\mathbf{\Sigma}}_n$ are all zero. Moreover, Algorithm 1 reduces asymptotically to the binary uncertainty-based EM proposed in [10] when the covariance matrices $\bar{\mathbf{\Sigma}}_n$ are diagonal with the entries corresponding to observed data being zero and the entries corresponding to the missing data tending to infinity.

Algorithm 2 One iteration of the EM algorithm for the log-likelihood integration-based GMM learning from noisy data.

E step. Conditional expectations of natural statistics:

$$\gamma_{i,n} \propto \omega_i N(\mathbf{y}_n|\boldsymbol{\mu}_i, \mathbf{\Sigma}_i) e^{-\frac{1}{2}\text{tr}\left(\mathbf{\Sigma}_i^{-1}\bar{\mathbf{\Sigma}}_n\right)}, \quad \text{and} \quad \sum\nolimits_i \gamma_{i,n} = 1, \quad (20)$$

M step. Update GMM parameters:

$$\omega_i = \frac{1}{N}\sum_{n=1}^{N}\gamma_{i,n}, \quad (21)$$

$$\boldsymbol{\mu}_i = \frac{1}{\sum_{n=1}^{N}\gamma_{i,n}}\sum_{n=1}^{N}\gamma_{i,n}\mathbf{y}_n, \quad (22)$$

$$\mathbf{\Sigma}_i = \frac{\sum_{n=1}^{N}\gamma_{i,n}(\mathbf{y}_n - \boldsymbol{\mu}_i)(\mathbf{y}_n - \boldsymbol{\mu}_i)^T + \bar{\mathbf{\Sigma}}_n}{\sum_{n=1}^{N}\gamma_{i,n}}. \quad (23)$$

4. Experiments

We now consider two evaluation frameworks based on a classification task in order to compare the benefits of the two uncertainty handling methods. The first experiment considers artificial data sampled from GMMs whereas the second one considers features extracted from speech utterances of the TIMIT database [16].

In these two cases, the uncertainty over each feature, i.e., the covariance matrix $\bar{\mathbf{\Sigma}}_n$, is assumed correspond to the actual amount of noise in the data. Even though this cannot be realistically assumed in practice, this study focuses on validating the methodological aspects of the proposed approaches. For that purpose, controlling the characteristics of the uncertainty is valuable as it will be shown in the remaining of this section. For the two experiments reported here, a wide range of possible setups have been considered. Finally, to ease the control of the uncertainty characteristics, all the covariance matrices $\bar{\mathbf{\Sigma}}_n$ are considered as diagonal, i.e.,

$$\bar{\mathbf{\Sigma}}_n = \mathrm{diag}\left\{[\bar{\sigma}^2_{m,n}]_m\right\}. \tag{24}$$

However, our framework is not limited to this specific case.

The uncertainty in these experiments is controlled by two parameters. The level of noise called *Feature to Noise Ratio (FNR)* is expressed in dB as

$$\mathrm{FNR} = 10\log_{10}\frac{\sum_n \|\mathbf{x}_n\|^2}{\sum_n \|\mathbf{x}_n - \mathbf{y}_n\|^2}, \tag{25}$$

where the summations are over all features in all classes. As previously stated, interfering sources are highly non-stationary. As a consequence, a high level of variability of noise can be assumed from a feature component to another and from a time frame to the next. In order to reflect this phenomenon, we introduce a variability parameter called *Noise Variation Level (NVL)* and computed as the standard deviation of variances from (24) expressed in dB:

$$\mathrm{NVL} = \mathrm{stdev}\left(\left\{10\log_{10}\bar{\sigma}^2_{m,n}\right\}_{m,n}\right). \tag{26}$$

As the noise over the data can be different for training and classification, we considered four parameters: FNR_{train} and NVL_{train} control the noise over the training data and FNR_{test} and NVL_{test} control the noise over the data to be classified. The following values are considered:

$$\begin{aligned} \mathrm{FNR}_{train},\, \mathrm{FNR}_{test} &= \{-20, -10, 0, 10, 20\}, \\ \mathrm{NVL}_{train} &= \{0, 4, 8\}, \\ \mathrm{NVL}_{test} &= \{0, 2, 4, 6, 8\}. \end{aligned}$$

This gives a total number of 375 setups. For each of these setups, training and classification are performed and the average number of correct classification is recorded.

For each setup specified by its FNR and NVL, artificial noise is generated in the following manner. First, the log-variances $\{\log\bar{\sigma}^2_{m,n}\}_{m,n}$ are drawn from the zero-mean unit variance Gaussian random distribution and scaled so as to satisfy (25). Second, the expected values $\mathbf{y}$ of the missing features are drawn from the Gaussian distribution (6), given $\mathbf{x}$ and $\bar{\mathbf{\Sigma}}_n = \{\bar{\mathbf{\Sigma}}_n\}_n$ defined by (24), and both $\mathbf{y} - \mathbf{x}$ and $\{\bar{\sigma}_{m,n}\}_{m,n}$ are scaled by the same factor in order to satisfy (26). In order to remove additional variability the seed of the random number generator was re-initialized to the same value for each setup.

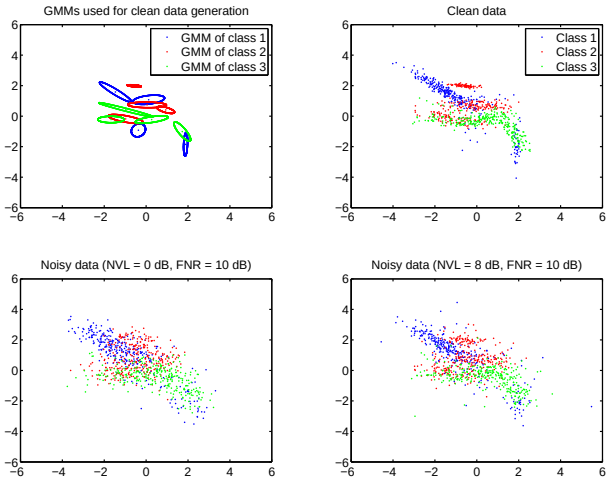

Figure 2: Artificial data examples: GMMs used for clean data generation (top, left), clean data $\mathbf{x}$ (top, right), noisy data $\mathbf{y}$ with zero NVL (bottom, left), and noisy data $\mathbf{y}$ with high NVL (bottom, right).

4.1. Artificial data

In the first experiment, two-dimensional artificial features were generated from three 4-states GMMs representing the corresponding three classes (Fig. 2, top, left). For each setup and each class a training sequence of length 300 and 100 test sequences of length 100 were randomly drawn from distributions of the corresponding GMMs. Artificial noise was generated and added as explained above.

Figure 2 shows an example of clean data $\mathbf{x}$ and two examples of noisy data $\mathbf{y}$ with the same FNR = 10 dB and with different NVLs (0 and 8 dB). One can note that, as compared to low NVL, the distribution of the data with high NVL is closer to the clean data, but at the same time there are more outliers. Thus, for a given FNR, it should be easier to classify and to learn from data with large NVL, since the influence of the outliers can be diminished by taking into account the uncertainty.

Three approaches including likelihood integration, log-likelihood integration and the baseline conventional GMM-based classification [6] that does not take the uncertainty into account were evaluated for all 375 setups described above. To investigate the impact of the FNR and the NVL on the classification performance, selected results are shown in Figure 2.

The left part of Figure 2 shows the impact of the FNR at the training and testing stage. The likelihood integration approach is the most powerful. It gets the best correct classification rate and has the lowest dependency to the *heterogeneity* of the noise level between the testing and training sets. Compared to the baseline approach that does not consider the uncertainty, the log-likelihood integration approach gets slightly better results in terms of accuracy but remains sensitive to the heterogeneity of the noise level between training and testing data. For example, to classify a test set with an FNR at 0 dB, a classifier trained with an FNR at 0 dB is almost three times better than a classifier trained with a cleaner dataset (FNR at 20 dB).

The right part of Figure 2 shows the impact of the NVL both at the training and testing stage. The likelihood approach behaves as expected (c.f., the above informal comparison of two noisy datasets represented at the bottom of Figure 2), i.e., the

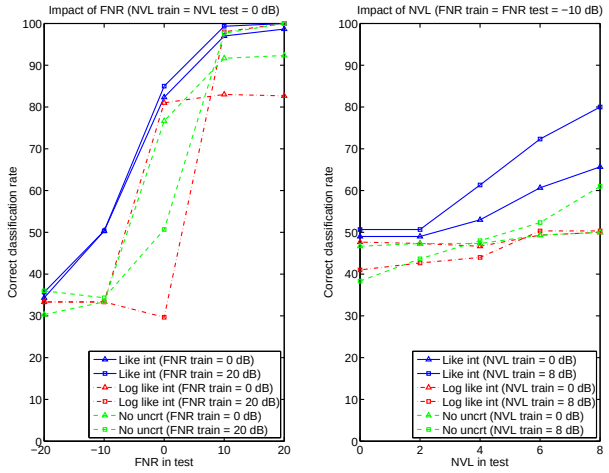

Figure 3: Artificial data results: impact of the FNR in train and test (left) and impact of the NVL in train and test (right). Approaches: "Like int" = likelihhod integration, "Log like int" = log-likelihhod integration, and "No uncrt" = conventional GMM-based classification [6] that does not take into account the uncertainty.

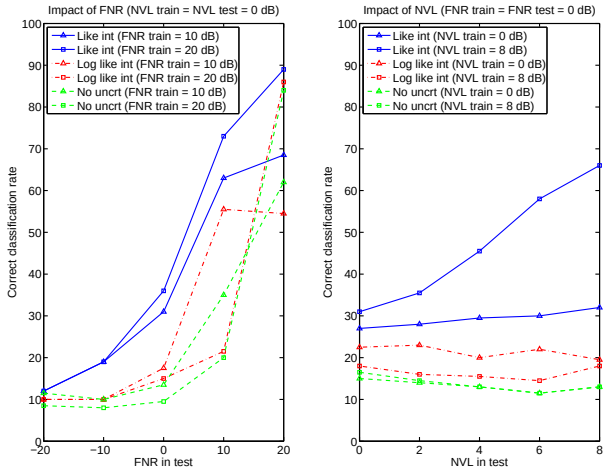

Figure 4: Speech data results: impact of the FNR in train and test (left) and impact of the NVL in train and test (right). Approaches: "Like int" = likelihhod integration, "Log like int" = log-likelihhod integration, and "No uncrt" = conventional GMM-based classification [6] that does not take into account the uncertainty.

larger the NVL at train and test, the better the accuracy. The two other approaches do not benefit so much from a large NVL and face the same dependency to the above discussed heterogeneity. Indeed, for both the log-likelihood and baseline approaches, a classifier trained over a dataset at 0 dB NVL is better at classifying a test-set at 0 dB NVL than a classifier trained over a dataset at 8 dB NVL.

4.2. Speech data

The second experiment is mostly based on the one considered in [6]. The task is to recognize the speaker who pronounced the test sentence among the speakers for which GMMs have been trained. A subset of 10 male speakers of the TIMIT database [16] is considered. For each speaker and each setup, a full-covariance 16-states GMM is trained using the two *sa* sentences and three *si* sentences of this speaker. The remaining five *sx* sentences were cut into 4 pieces of equal length that were used for testing. This gives a total of 200 test sequences of approximately half a second.

The Mel frequency cepstral coefficients (MFCCs) [17] are usually used as features for the speaker recognition task [6]. One of the advantages of the MFCCs is that they can reasonably be assumed to be decorrelated, and thus can be modeled by GMMs with diagonal covariance matrices, which decreases computational complexity. However, the error introduced by a source separation algorithm cannot be assumed to be decorrelated over MFCC vector dimensions, as we assume within this methodological evaluation framework (see (24)). This error would be rather decorrelated on the logarithm of Mel-frequency filter-bank outputs (LMFFB). In fact, at least for probabilistic model-based source separation approaches (see e.g., [2]), the estimation error is usually decorrelated between different frequency bands [15], thus it is nearly decorrelated between the dimensions of the LMFFBs (there is still a small correlation due to the overlap of Mel-frequency filters). Thus, we here consider the LMFFBs with diagonal uncertainty covariance matrices and model them by GMMs with full covariance matrices. Since the MFCCs are computed from the LMFFBs by the discrete cosine transform (DCT), which is an orthogonal transform, this setting is very similar to the usual setting with the MFCCs, where the features are modeled by GMMs with diagonal covariance matrices and the uncertainty covariance matrices should be non-diagonal. We use LMFFB vectors of dimension 20.

The same experiments as in the case of artificial data were carried out and selected results are shown on Figure 2.

The left part of Figure 4 shows the impact of the FNR at the training and testing stage. The conclusions made over the synthetic dataset still hold: the likelihood integration approach also gets the best correct classification rate and has the lowest dependency to the heterogeneity of the noise level between the testing and training sets.

The right part of Figure 4 shows the impact of the NVL both at the training and testing stage. The likelihood approach behaves as expected, i.e., the higher the NVL at train and test, the better the accuracy. On contrary, the accuracy of the other two approaches appears to be rather independent of the NVL and it is just slightly superior to that of a random classifier.

5. Conclusions

We studied in this paper the learning of GMMs models over noisy data using two criteria. The first criterion called likelihood integration, proposed in [5], assumes that the data have been generated by a GMM. We proposed a second criterion called log-likelihood integration that does not make such assumption, which theoretically seemed beneficial while facing real data. For both criteria, we proposed EM algorithms in order to estimate the parameters of the GMMs over noisy data.

Experimental evaluation conducted over synthetic and speech data with knowledge of the uncertainty demonstrated the superiority of likelihood integration over log-likelihood integration and the standard approach that does not consider uncertainty information, thus validating the approach taken in [5]

for both the training and testing stages. The experiments also demonstrated that considering the uncertainty allows us to: (i) handle the heterogeneity of noise between the training and testing sets, (ii) exploit the variability of noise for improved performance.

Future work will consider realistic uncertainty estimates from source separation algorithms over a larger evaluation dataset in order to evaluate the practical benefit of this new classification scheme. Another interesting research direction would be to consider the log-likelihood integration within a GMM-based classification framework with discriminative training [14].

6. References

[1] Y. Ephraim, "Statistical-model-based speech enhancement systems," *Proceedings of the IEEE*, vol. 80, no. 10, pp. 1526–1555, 1992.

[2] A. Ozerov, E. Vincent, and F. Bimbot, "A General Modular Framework for Audio Source Separation," in *9th International Conference on Latent Variable Analysis and Signal Separation (LVA/ICA'10)*, 2010, pp. 27–30.

[3] C. Nadeu, P. Pachès-Leal, and B.-H. Juang, "Filtering time sequences of spectral parameters for speech recognition," *Speech Communication*, vol. 22, pp. 315–332, 1997.

[4] M. Cooke, "Robust automatic speech recognition with missing and unreliable acoustic data," *Speech Communication*, vol. 34, no. 3, pp. 267–285, Jun. 2001.

[5] L. Deng, J. Droppo, and A. Acero, "Dynamic compensation of HMM variances using the feature enhancement uncertainty computed from a parametric model of speech distortion," *IEEE Transactions on Speech and Audio Processing*, vol. 13, no. 3, pp. 412–421, May 2005.

[6] D. Reynolds, "Large population speaker identification using clean and telephone speech," *IEEE Signal Processing Letters*, vol. 2, no. 3, pp. 46–48, Mar. 1995.

[7] J. Barker, M. Cooke, and D. Ellis, "Decoding speech in the presence of other sources," *Speech Communication*, vol. 45, no. 1, pp. 5–25, Jan. 2005.

[8] S. Srinivasan and D. Wang, "Transforming Binary Uncertainties for Robust Speech Recognition," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 15, no. 7, pp. 2130–2140, Sep. 2007.

[9] Y. Shao, S. Srinivasan, Z. Jin, and D. Wang, "A computational auditory scene analysis system for speech segregation and robust speech recognition," *Computer Speech & Language*, vol. 24, no. 1, pp. 77–93, Jan. 2010.

[10] Z. Ghahramani and M. Jordan, "Supervised learning from incomplete data via an EM approach," in *Advance on Neural Information Processing Systems*, 1994, pp. 120–127.

[11] D. Kolossa, R. Fernandez Astudillo, E. Hoffmann, and R. Orglmeister, "Independent Component Analysis and Time-Frequency Masking for Speech Recognition in Multitalker Conditions," *EURASIP Journal on Audio, Speech, and Music Processing*, vol. 2010, pp. 1–14, 2010.

[12] A. P. Dempster, N. M. Laird, and D. B. Rubin., "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 39, pp. 1–38, 1977.

[13] S. Arberet, A. Ozerov, R. Gribonval, and F. Bimbot, "Blind spectral-GMM estimation for underdetermined instantaneous audio source separation," in *Proc. Int. Conf. on Independent Component Analysis and Blind Source Separation (ICA'09)*, 2009, pp. 751–758.

[14] C. M. del Alamo, F. J. Caminero Gil, C. dela Torre Munilla, and L. Hernandez Gomez, "Discriminative training of gmm for speaker identification," in *Proc. Conf. IEEE Int Acoustics, Speech, and Signal Processing ICASSP-96*, vol. 1, 1996, pp. 89–92.

[15] K. Adiloglu and E. Vincent, "An uncertainty estimation approach for the extraction of individual source features in multisource recordings," in *EUSIPCO, 19th European Signal Processing Conference*, 2011, submitted.

[16] J. S. Garofolo, L. F. Lamel, W. M. Fisher, J. G. Fiscus, D. S. Pallett, and N. L. Dahlgren, "DARPA TIMIT acoustic phonetic continuous speech corpus CDROM," 1993.

[17] R. Vergin, D. O'Shaughnessy, and A. Farhat, "Generalized mel frequency cepstral coefficients for large-vocabulary speaker-independent continuous-speech recognition," *IEEE Transactions on Speech and Audio Processing*, vol. 7, no. 5, pp. 525–532, 1999.