Acoustic Event Classification using Multi-resolution HMM

Paul M. Baggenstoss
Fraunhofer FKIE, Fraunhoferstrasse 20
53343 Wachtberg, Germany
Email: p.m.baggenstoss@ieee.org

***Abstract*—Real-world acoustic events span a wide range of time and frequency resolutions, from short clicks to longer tonals. This is a challenge for the hidden Markov model (HMM), which uses a fixed segmentation and feature extraction, forcing a compromise between time and frequency resolution. The multi-resolution HMM (MR-HMM) is an extension of the HMM that assumes not only an underlying (hidden) random state sequence, but also an underlying random segmentation, with segments spanning a wide range of sizes and processed using a variety of feature extraction methods. It is shown that the MR-HMM alone, as an acoustic event classifier, has performance comparable to state of the art discriminative classifiers on three open data sets. However, as a generative classifier, the MR-HMM models the underlying data generation process and can generate synthetic data, allowing weaknesses of individual class models to be discovered and corrected. To demonstrate this point, the MR-HMM is combined with auxiliary features that capture temporal information, resulting in significantly improved performance.**

I. Introduction

A. Motivation and Background

The classification of acoustic events is part of an emerging field of machine hearing [1]. Approaches based on the hidden Markov model (HMM) [2] have origins in speech recognition [3]. As the field matures, more methods emerge [4], [5], [6], [7], [8], [9]. Many newer methods are inspired by the machine learning field (deep neural networks) [10], [11], [12], [13]. These discriminative methods avoid the daunting task of designing generative models for each signal class. But, if performance is equal, a generative model is always prefererable because it models the processes underlying the data, and can generate data, allowing the quality of the model to be judged. The HMM is a versatile generative model, but is limited by the reliance on a fixed (uniform) segmentation of the data and a fixed feature extraction. With maximum entropy PDF projection (MEPP) [14], [15], the freedom exists to combine multiple feature extractors and window (segment) sizes together in a single generative model. The multi-resolution HMM (MR-HMM) was introduced to do this [16], [17], [18], [19]. In this paper, the MR-HMM is demonstrated as an acoustic event classifier.

B. MR-HMM description

The MR-HMM [17] is a type of graphical model [20] with random segment size, modeling the raw data (instead of features). It uses features indirectly through PDF projection [21]. The best way to explain the MR-HMM is to review the HMM, then convert it to a MR-HMM in 4 conceptual steps.

1) The HMM: The HMM is a statistical model for a sequence of features extracted from the time-series using equally-spaced overlapping processing windows. Let $\mathbf{Z}$ be a sequence of T feature vectors $\mathbf{Z} = [\mathbf{z}_1, \mathbf{z}_2 \dots \mathbf{z}_T]$. The HMM assumes an underlying discrete Markov state label $s_i \in [1, L]$ associated with feature vector $\mathbf{z}_i$, that is chosen in time order, according to a discrete probability distribution depending only on the last chosen state. This Markov process is described by an $L \times 1$ prior distribution (for the first feature vector) and an $L \times L$ state transition matrix (STM). The HMM further assumes that the feature vector $\mathbf{z}_i$ is a sample drawn from the probability density function (PDF) given by $p(\mathbf{z}|s_i)$, depending only on s_i. Let $\mathbf{s}$ be one possible state sequence, $\mathbf{s} = [s_1, s_2 \dots s_T]$. A key HMM assumption is mutual conditional independence of the feature vectors, so that $\log p(\mathbf{Z}|\mathbf{s}) = \sum_{i=1}^{T} \log p(\mathbf{z}_i|s_i)$. To get the joint marginal PDF $P(\mathbf{Z})$, it is necessary to average over *all* state sequences $p(\mathbf{Z}) = \sum_{\mathbf{s}} p(\mathbf{Z}|\mathbf{s})\ P(\mathbf{s})$, where $P(\mathbf{s})$ is just a product over the Markov state transition probabilities. Fortunately, $p(\mathbf{Z})$ can be computed efficiently using the well-known *forward procedure* [3]. Let Λ be the set of all HMM parameters. The HMM efficiently solves the three fundamental problems (a) **Segmentation** : $\hat{\mathbf{s}} = \arg\max_{\mathbf{s}} p(\mathbf{Z}|\mathbf{s}; \Lambda)$, and (b) **Classification** : $\hat{m} = \arg\max_m p(\mathbf{Z}; \Lambda_m)$. (c) **Estimation** : $\hat{\Lambda} = \arg\max_{\Lambda} \sum_l \log p(\mathbf{Z}_l; \Lambda)$,

2) Step 1: Time Series HMM (Proxy HMM): The first conceptual step is to convert the HMM from a feature sequence model into a time-series model by replacing the feature vector $\mathbf{z}_i$ by a short time-series segment of length K samples $\mathbf{x}_i = [x_{1+(K-1)i}, \dots x_{Ki}]$. Assume no segment overlap and no shading function, so that a sequence of T short segments of length K is equivalent to the full time-series of length N_t $\mathbf{X} = [x_1, x_2 \dots x_{N_t}]$, where $N_t = K \cdot T$. The HMM with K-sample segmentation of the raw data is conceptual only, and will be used later as a proxy for computing the necessary probabilities.

3) Step 2: Random segment sizes: The second step is to relax the assumption of constant segment size. Assume that the segment size is chosen from a set $\mathcal{K}$ of n elements, for example $\mathcal{K} = \{24, 36, 48, 72, 96, 144, 192, 288, 384, 768\}$, where $n = 10$, and all segment sizes are divisible by the base segment size K, $K = 12$ in the above example. The MR-HMM generates a sequence of S segments as follows. The segment counter i is initialized to 1, then

1) If $i = 1$, select the discrete segment state s_i according to the initial state probabilities $\pi(s)$, where $\sum_{s=1}^{L} \pi(s) = 1$, otherwise use the STM $A(s_{i-1}, s_i)$.

ISBN 978-90-827970-1-5 © EURASIP 2018

2) Select the segment size index j_i according to the discrete probability $\rho_{s_i,j}$, where $\sum_{j=1}^{n} \rho_{s_i,j} = 1$. Segment i will have length $k(j_i) = \mathcal{K}(j_i)$, which is the j_i-th element of set $\mathcal{K}$.
3) If $i < S$, increment i and go to step 1.

This process results in a random segmentation denoted by $\mathbf{q} = \{s_1, s_2, \ldots s_S, \quad j_1, j_2, \ldots j_S\}$. The total length of the time-series will be $N_t = \sum_{i=1}^{S} \mathcal{K}(j_i)$. Note that $\mathbf{q}$ specifies not only the segment size assignment, but also the segment class label assignments. Based on this generation process, the probability of generating a particular segmentation $\mathbf{q}$ is:

$$P(\mathbf{q}) = \pi(s_1)\rho_{s_1,j_1} \prod_{i=2}^{S_q} A(s_{i-1}, s_i)\rho_{s_i,j_i}. \tag{1}$$

Note that (1) is the exact probability of generating a random segmentation of S segments which has random length. It is only an approximation, albeit a good one, for a time-series of fixed length[1]. Using mutual conditional independence, $\log p(\mathbf{X}|\mathbf{q}) = \sum_{i=1}^{S} \log p(\mathbf{x}_i|s_i, j_i)$. Expanding $p(\mathbf{X})$,

$$p(\mathbf{X}) = \sum_{\mathbf{q}} p(\mathbf{X}|\mathbf{q})\ P(\mathbf{q}), \tag{2}$$

with the summation over *all* possible segmentations $\mathbf{q}$. Equation (2) is the key expression of the MR-HMM, giving the probability distribution of $\mathbf{X}$ as the expected value of $p(\mathbf{X}|\mathbf{q})$, with expectation taken over the distribution of segmentations $\mathbf{q}$, given the set $\mathcal{K}$ of allowable segmentation sizes. It summarizes the information from all the possible segmentations in a single quantity! It still remains to describe how (2) is efficiently computed.

4) Step 3: Mapping to Proxy HMM: In order to avoid the brute-force implementation of (2), it is possible to map the MR-HMM to a proxy HMM, which can compute (2) using the well-known forward procedure. The proxy HMM is a standard HMM operating on hypothetical base segments of length K. Therefore, for every MR-HMM segment of length k base segments, there are k proxy HMM segments, and the proxy HMM segment log-likelihood functions are just $1/k$ times the segment log-likelihood of the MR-HMM segment. For every MR-HMM segmentation $\mathbf{q}$ there must exist a corresponding path through the simpler proxy HMM trellis. Therefore, there must be enough proxy HMM states to encode any segmentation: the state s_i, the segment size j_i, and the position t within the segment (called *wait state*), a total of $\sum_{s=1}^{L} \sum_{j=1}^{n} k(j)$ states. The proxy STM is very large, but sparse and highly structured, forcing each wait state to be visited in turn, and only allowing transitioning to a different state label s and segment size index j once the last wait state is reached. The proxy HMM STM is constructed in the obvious way from $\pi(s)$, $A(s, s')$, $\rho_{s,j}$ so that the probability of a given path through the state diagram is (1). Using the proxy HMM and the methods developed for the HMM [3], the fundamental problems are solved: (a) **Segmentation** : $\hat{\mathbf{q}} = \arg\max_{\mathbf{q}} p(\mathbf{X}|\mathbf{q}; \Lambda)$, and (b) **Classification** : $\hat{m} = \arg\max_m p(\mathbf{X}; \Lambda_m)$. It also solves the third: (c) **Estimation** : $\hat{\Lambda} = \arg\max_{\Lambda} \sum_l \log p(\mathbf{X}_l; \Lambda)$, as explained in below.

[1]Near the end of the time-series, the probability of choosing a segment length longer than the remining number of samples must be zero.

5) Step 4: Mapping to features: Using the proxy HMM, we have already solved the problem of summarizing the likelihood functions from different-sized segments. But, how can we compute the likelihood functions from large time-series segments, and how can we use different feature extractors? To solve this problem, we turn to PDF projection [21]. Much has been written on the topic [21], [14], [15], so only the main points are reviewed here. Let $\mathbf{x} \in \mathcal{R}^N$ be a segment of the input time-series (raw data), of dimension N, where $N = k(j) \cdot K$. Let $\mathbf{z} = T(\mathbf{x})$ be an arbitrary feature, where $\mathbf{z} \in \mathcal{R}^D$, and $D << N$. Assume for the moment that the feature PDF, denoted by $g(\mathbf{z})$, is known. Then, the log-projected PDF is given by

$$\log G(\mathbf{x}; H_0) = \log p(\mathbf{x}|H_0) - \log p(\mathbf{z}|H_0) + \log g(\mathbf{z}), \tag{3}$$

where H_0 is a fixed statistical reference hypothesis. It may be shown [21], [14], [15] that $G(\mathbf{x}; H_0)$ is a PDF, so $\int_{\mathbf{x}} G(\mathbf{x}; H_0) = 1$. Samples draw from $G(\mathbf{x}; H_0)$, and passed through the feature transformation $T(\mathbf{x})$ will have exactly distribution $g(\mathbf{x})$, so $G(\mathbf{x}; H_0)$ is *consistent* with $g(\mathbf{x})$. All PDFs consistent with $g(\mathbf{x})$ may be written in the form (3), and depend only on the chosen reference hypothesis H_0. Subject to some mild requirements [14], H_0 may be chosen for maximum entropy (MaxEnt) of $G(\mathbf{x}; H_0)$. Maximizing the entropy is a well established principle in PDF estimation [22], [15] that allows the PDF estimate to express not only the available knowledge about $\mathbf{x}$, but also the ignorance. Let H_0^* be this maximum entropy (MaxEnt) reference hypothesis. This defines a unique MaxEnt projected PDF, denoted by $G^*(\mathbf{x}) = G(\mathbf{x}; H_0^*)$. For the data treated in this paper, i.e., Gaussian-like data, raw acoustic recordings, with no strict upper or lower bonds in amplitude, the Gaussian reference hypothesis $p(\mathbf{x}|H_0^*) = (2\pi)^{-N/2} \exp\left\{-\frac{\sum_{i=1}^{N} x_i^2}{2}\right\}$ produces maximum entropy.

For feature extraction that involves several stages of feature reductions, the combined feature chain is most easily analyzed using the chain-rule, which allows stage-by-stage application of MaxEnt PDF projection [21], [19].

To apply PDF projection to the time-series segment probability $p(\mathbf{x}|s, j)$ one needs feature extractor, denoted by $\mathbf{z} = T_{s,j}(\mathbf{x})$, and an estimate of the feature distribution $\hat{p}(\mathbf{z}|s, j)$. Then

$$\log \hat{p}(\mathbf{x}|s, j) = \log p(\mathbf{x}|H_0^*) - \log p(\mathbf{z}|H_0^*) + \log \hat{p}(\mathbf{z}|s, j). \tag{4}$$

Estimation of $\hat{p}(\mathbf{z}|s, j)$ proceeds analogous to the standard HMM [3]. For example, the estimate of the mean of the distribution $\hat{p}(\mathbf{z}|s, j)$ is just the weighted average of $\mathbf{z}$ - weighted by the *a posteriori* probabilities that the segment belongs to state s, segment size index j. Re-estimation of covariances and the transition probabilities $\pi(s)$, $A(s, s')$ is also analogous [17].

Segment time-series data is generated from PDF (4), using the 2-step process called uniform manifold sampling (UMS) [15] (a) generate a sample of the feature PDF $\hat{p}(\mathbf{x}|s, j)$, denoted by $\mathbf{z}^*$, (b) determine the level set $\mathcal{M}_z = \{\mathbf{x} : T_{s,j}(\mathbf{x}) = \mathbf{z}^*\}$ and draw a sample $\mathbf{x}$ from $\mathcal{M}_z$ uniformly - so that no element of $\mathcal{M}_z$ is more likely to be chosen than another. More on this topic can be found in the references (Section III.B of [15]).

ISBN 978-90-827970-1-5 © EURASIP 2018

Configuration 1 ($K = 12$)		Configuration 2 ($K = 32$)	
Segment Size	Band functions	Segment Size	Band functions
36	3	32	3
48	4	64	4
72	6	96	6
96	8	128	8
144	12	192	12
192	16	256	16
288	24	384	24
384	32	768	48
576	48		
768	64		

TABLE I. BASE SEGMENT SIZE K, FFT (SEGMENT) SIZES AND NUMBER OF MEL-SPACED BAND FUNCTIONS FOR TWO DIFFERENT MR-HMM CONFIGURATIONS. $L = 6$ STATES WERE USED FOR BOTH CONFIGURATIONS.

MR-HMM Only (Configuration 1, $K = 12$, $L = 6$)					
Data Set	Errors	Tests	Error (%)	% Correct	Best Published
Office Sounds	11	2448	0.45%	99.55%	99.6% [24]
Freiburg106	10	732	1.37%	98.63%	98.9% [5] (*)
NAR	10	252	3.97%	96.03%	96.0% [26] (**)
MR-HMM Only (Configuration 2, $K = 32$, $L = 6$)					
Data Set	Errors	Tests	Error (%)	% Correct	Best Published
Office Sounds	12	2448	0.49%	99.51%	99.6% [24]
Freiburg106	10	732	1.37%	98.63%	98.9% [5] (*)
NAR	8	252	3.17%	96.83%	96.0% [26] (**)

TABLE II. CLASSIFICATION PERFORMANCE USING MR-HMM ONLY. (*) F-SCORE. (**) TEN-FOLD CROSS-VALIDATION.

II. MR-HMM CONFIGURATION AND TRAINING

A. MR-HMM Features

As explained, the MR-HMM can apply different feature extraction methods for each combination of segment size and state. For simplicity, however, we used just one feature type across all segment sizes. We used MEL-cepstral features, widely used in speech processing, which are described in detail in ([14], page 2821). A rectangular time-domain window function was used (no shading), which is justified by the automatic segmentation inherent to MR-HMM. The parameters of the various cepstral features are provided in Table I for two different segmentation configurations $\mathcal{K}$, providing either $n = 10$ or $n = 8$ different segment sizes. All segment sizes are divisible by the respective base segment size. The number of proxy HMM states for the two configurations in Table I, are 217 and 60 proxy states per state, respectively (We used $L = 6$ states). The second configuration is more efficient from computational point of view. Note that the ratio between the segment size and number of band functions is constant. In this way, frequency and time resolution are traded off in proportion and the aggregate number of features for a given time-series is independent of the segmentation $\mathbf{q}$.

B. MR-HMM initialization and training

To initialize the MR-HMM, a general-purpose time-series segmentation algorithm was employed to divide each time-series into segments of various size with similar spectral content [23]. This segmentation is only used for the purpose of initialization. To automatically determine a set of L states, the spectral content of the segments of the initial segmentation is clustered into L clusters. For a spectral representation that was independent of segment size, we used cepstral coefficients (the DCT of the log of the magnitude-squared DFT), keeping always 25 coefficients regardless of segment size. If K was less than 48, we zero-padded to 25 coefficients. Then, the collection of 25-dimensional segment cepstral vectors from all segments of various sizes were clustered using K-means to L clusters. Each segment was then classified as one of the L states to initialize the MR-HMM segment feature PDFs, modeled as Gaussian, for each combination of segment size and state. All the discrete probabilities including the initial state probabilities $\pi(s)$, the segment size probabilities $\rho_{s,j}$, and the state transition probabilities $A(s, s')$ were intitialized to flat (uniform) distributions. We trained the MR-HMM with state "ganging" (See [17], Section III.C) for 40 iterations, or until $\sum_i \log p(\mathbf{X}_i; \Lambda)$ stopped increasing, whichever came first. We used specialized software written in C to take advantage of the sparse and highly structured state transition matrix of the proxy HMM to achieve about two orders of magnitude processing time reduction.

III. EXPERIMENTAL RESULTS

A. Data description

1) Office Sounds database: The Office Sounds database [24], [25] contains twenty-four signal classes containing 102 samples of each class, a total of 2448 example sounds created by dropping common objects or operating office tools such as scissors or staplers. All time-series are 16128 samples long (1/2 second in duration at 32000 Hz). The dataset was divided using 2-fold cross-validation into even and odd-numbered sets of 51 samples per class. Both "folds" were tested producing 2448 classifier decisions.

2) NAR database: The NAR dataset [26], [2] contains 12 event categories, with 21 events of each category. The data is recorded in a home (kitchen) environment at 48000 Hz using inexpensive microphones. We used 4-fold data holdout by selecting every fourth event for testing, and the remaining 3/4 of the data for training. We conducted all four "folds" and averaged the results. There were 252 classifier decisions.

3) Freiburg 106 database: The Freiburg-106 dataset [5], [9] contains 1,479 audio-based human activities of 22 categories recorded in a home (kitchen) environment at 44100 Hz using an inexpensive microphone. We used only the first channel in our experiments[2]. We shortened event 80 of Category 22 to 88000 samples so that it did not disproportionately influence training. We used 2-fold cross-validation as in [5], [9]. There were 732 classifier decisions.

B. MR-HMM Results

We measured classification performance using both MR-HMM configurations in Table I. When classifying, we used a Bayesian class prior $P(H_m)$ proportional to the total number of testing samples. Classification performance for all three data sets is shown on Table II using the specific data holdout method described for each data set. The results are comparable to the best published results, which are also provided in Table II. Below it is shown how the MR-HMM can be easily complemented in order to achieve significantly better results.

[2]The existence of the second channel is not explained in the references.

ISBN 978-90-827970-1-5 © EURASIP 2018

C. Auxiliary Features

Once the MR-HMM is trained, time-series data can be generated using the process described in Section I-B3 to generate a segmentation, then using the process of Section I-B5 to generate the segment data. By synthesizing events and comparing them with real data, it can be determined what is missing in the statistical model. In Figure 1, synthetic events are shown for three of the classes from "Freiburg106" data set. The synthetic event for the first class is quite

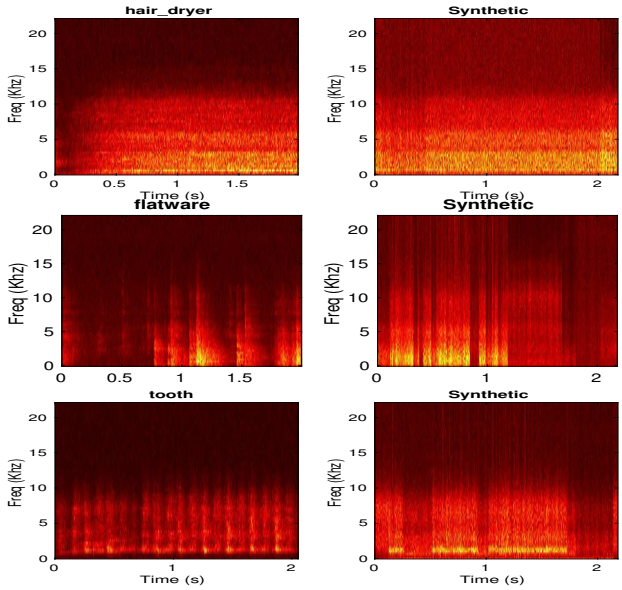

Fig. 1. Spectrograms of real data (left) and synthetic data (right) for classes "hair_dryer" (top) and "flatware_sorting" (middle), and "brushing teeth" (bottom).

good, showing excellent spectral and temporal fidelity. For the second and third classes, the spectral representation is good, but the temporal distribution is poor. The Markov model cannot adequately represent exponentially decaying sounds ("flatware_sorting") or periodic sounds ("brushing teeth"). This inability to adequately represent the temporal dimension is independent of data set, it is a weakness of the Markov model. To address the defficiencies in the statistical model as demonstrated by the synthesis above, we employ a set of auxiliary feature designed to capture any missing characteristic, such as temporal distribution.

1) Energy Cepstrum: The energy cepstrum is the MEL frequency cepstral coefficients (MFCC) applied to the log of the short-time averaged energy, which was measured at a rate of 240 Hz. After taking the log, the energy time-series was de-meaned, then passed into a MEL filter bank of 24 bands, log, DCT, and the lowest 12 DCT coefficients were retained.

2) Piecewise Linear Fit to Log-Energy: Many event categories contain bangs caused by striking hard objects producing sounds with fast onset and exponentially decaying sound level. In the log-energy domain, this looks like a piece-wise linear function. To capture this effect in a set of features, we fit a piecewise-linear function to the log-energy time-series. In a straight-forward application of dynamic programming, we found the best set of line-segments that matched the log-energy in terms of least-squares error. A best solution was found for each number of line segments from 1 to 24. As more line-segments are added, the total squared error decreases and the log-likelihood (assuming Gaussian model) increases. But, if a minimum description length (MDL) penalty [27] is subtracted from the log-likelihood, it reaches a peak and then falls as more segments are added. Figure 2 shows an example of fitting to the log-energy time-series of a sample of "microwave_door" from "Freiburg106". The solution at 4 segments had the highest MDL-corrected log-likelihood and was selected. We gathered three features from the piecewise-linear fit: average segment length, average segment slope, and log-likelihood improvement (w/respect to 1 segment).

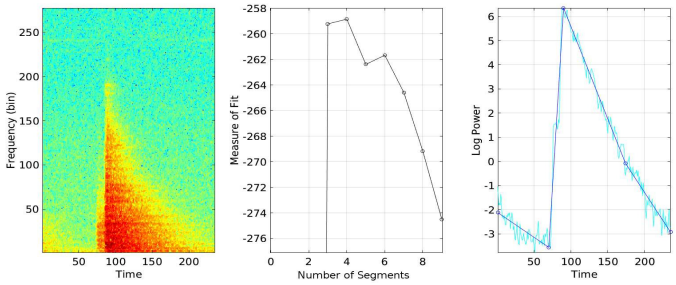

Fig. 2. Piecewise linear fit to log-energy for a sample of "microwave_door" from "Freiburg106". Left: spectrogram. Center: MDL-corrected log-likelihood as a function of number of segments. Right: Fit at 4 segments.

3) High-order moment: Some data classes have an impulsive character. To capture the impulsiveness of the data, we measured the 4-th moment of the raw data (with mean removed and normalized for variance 1).

D. Hybrid MR-HMM/Auxiliary Feature Classifier

The 16 auxiliary features (12 energy MFCC, 3 piecewise-linear-fit, and 1 high-order-moment) were modeled by a Gaussian distribution assuming independence among the features. A combined classifer was created by additively combining the MR-HMM and auxiliary feature classifier using $L_m = \log p(\mathbf{X}; \Lambda_m)/n + \alpha \log p_m(\mathbf{z})$, where $p(\mathbf{X}; \Lambda_m)$ is the MR-HMM likelihood function (2), and $p_m(\mathbf{z})$ is the auxiliary feature likelihood function, α is the mixing factor, and n is the total number of samples in the test time-series. Figure 3 shows the total number of classifier errors as a function of the mixing factor for all three data sets. A mixing factor between 0.1 and 0.4 seems to be suitable for all three data sets. The minumum errors are tabulated in Table III and exceeded best published results. If we insist on having a constant mixing factor, then using a value of 0.2 is a good compromise among all data sets, with performance worsening by no more than a single additional error.

IV. Conclusions

We have demonstrated the usefulness of the MR-HMM as an acoustic event classifier. For a three public data sets, the MR-HMM by itself performed comparable to best published results. However, MR-HMM is based on a Markov model, so does not offer an accurate statistical model of the temporal dynamics, which was vividly demonstrated by synthesizing

ISBN 978-90-827970-1-5 © EURASIP 2018

MR-HMM + Aux (Configuration 1, $K = 12$, $L = 6$)					
Data Set	Errors	Tests	Error (%)	% Correct	Best Published
Office Sounds	3	2448	0.125%	99.875%	99.6% [24]
Freiburg106	6	732	0.82%	99.18%	98.9% [5] (*)
NAR	0	252	0.0%	100.0%	96.0% [26] (**)
MR-HMM (Configuration 2, $K = 32$, $L = 6$) + Auxiliary Features					
Office Sounds	5	2448	0.20%	99.80%	99.6% [24]
Freiburg106	5	732	0.68%	99.32%	98.9% [5] (*)
NAR	0	252	0.00%	100%	96.0% [26] (**)

TABLE III. CLASSIFICATION PERFORMANCE USING MR-HMM AND AUXILIARY FEATURES. (*) F-SCORE. (**) TEN-FOLD CROSS-VALIDATION.

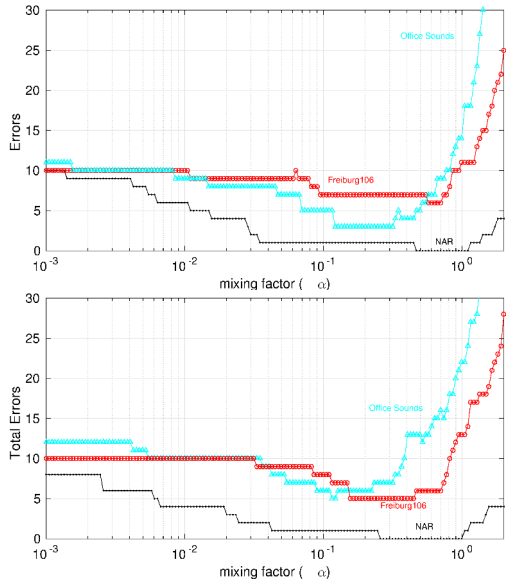

Fig. 3. Results for combining with auxiliary feature classifier in total number of errors. Top: MR-HMM configuration 1, Bottom: MR-HMM configuration 2. Shown on the X-axis is the mixing factor, which approaches zero to the left (MR-HMM only).

synthetic events. When this temporal information was re-introduced through an auxiliary classifier based on a set of simple temporal features, a significant improvement in classification performance was obtained, achieving over 99% correct classification in all cases. Obtaining top performance with a generative model is noteworthy at a time when discriminative methods are at the forefront. The ability to assess and repair the weakness in the MR-HMM classifier, was possible because it is a generative classifier. such as deep networks. As future work, it is planned to investigate way that the MR-HMM can complenent discriminative networks.

REFERENCES

[1] R. F. Lyon, "Machine hearing: An emerging field [exploratory dsp]," *IEEE Signal Processing Magazine*, vol. 27, pp. 131–139, Sept 2010.

[2] J. Maxime, X. Alameda-Pineda, L. Girin, and R. Horaud, "Sound representation and classification benchmark for domestic robots," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6285–6292, May 2014.

[3] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, pp. 257–286, February 1989.

[4] J. Beltrán, E. Chávez, and J. Favela, "Scalable identification of mixed environmental sounds, recorded from heterogeneous sources," *Pattern Recognition Letters*, vol. 68, pp. 153–160, 2015.

[5] H. Phan, L. Hertel, M. Maass, R. Mazur, and A. Mertins, "Audio phrases for audio event recognition," *Proceedings of EUSIPCO 2015, Nice, France*, Sep 2015.

[6] J. Dennis, Q. Yu, H. Tang, H. D. Tran, and H. Li, "Temporal coding of local spectrogram features for robust sound recognition," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 803–807, May 2013.

[7] P. Foggia, A. Saggese, N. Strisciuglio, and M. Vento, "Cascade classifiers trained on gammatonegrams for reliably detecting audio events," in *2014 11th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, 2014.

[8] J. Ye, T. Kobayashi, and M. Murakawa, "Urban sound event classification based on local and global features aggregation," *Applied Acoustics*, vol. 117, p. 246256, 2017.

[9] J. A. Stork, L. Spinello, J. Silva, and K. O. Arras, "Audio-based human activity recognition using non-markovian ensemble voting," in *2012 IEEE RO-MAN: The 21st IEEE International Symposium on Robot and Human Interactive Communication*, pp. 509–514, Sept 2012.

[10] I. McLoughlin, H. Zhang, Z. Xie, Y. Song, W. Xiao, and H. Phan, "Continuous robust sound event classification using time-frequency features and deep learning," in *PLoS ONE*, vol. 12, Sept 2017.

[11] I. McLoughlin, H. Zhang, Z. Xie, Y. Song, and W. Xiao, "Robust sound event classification using deep neural networks," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 23, pp. 540–552, March 2015.

[12] E. Cakir, "Multilabel sound event classification with neural networks," *Tampere University of Technology, Master of Science Thesis*, Sep 2014.

[13] K. J. Piczak, "Environmental sound classification with convolutional neural networks," in *2015 IEEE 25th International Workshop on Machine Learning for Signal Processing (MLSP)*, pp. 1–6, Sept 2015.

[14] P. M. Baggenstoss, "Maximum entropy PDF design using feature density constraints: Applications in signal processing," *IEEE Trans. Signal Processing*, vol. 63, June 2015.

[15] P. M. Baggenstoss, "Uniform manifold sampling (UMS): Sampling the maximum entropy pdf," *IEEE Transactions on Signal Processing*, vol. 65, pp. 2455–2470, May 2017.

[16] P. M. Baggenstoss, "A multi-resolution hidden markov model using class-specific features," *Proceedings of EUSIPCO 2008, Lausanne, Switzerland*, Aug 2008.

[17] P. M. Baggenstoss, "A multi-resolution hidden markov model using class-specific features," *IEEE Transactions on Signal Processing*, vol. 58, pp. 5165–5177, Oct 2010.

[18] B. F. Harrison and P. M. Baggenstoss, "A multi-resolution hidden markov model for optimal detection, tracking, separation and classification of marine mammal vocalizations," in *Proc. Oceans 2008, Quebec City*, September 2008.

[19] P. M. Baggenstoss, "Optimal detection and classification of diverse short-duration signals," in *Proceedings of the International Conference on Cloud Engineering*, (Boston, MA), pp. 534–539, 2014.

[20] J. A. Bilmes, "Graphical models and automatic speech recognition," *Mathematical Foundations of Speech and Language Processing*, 2003.

[21] P. M. Baggenstoss, "The PDF projection theorem and the class-specific method," *IEEE Trans Signal Processing*, pp. 672–685, March 2003.

[22] E. T. Jaynes, "On the rationale of maximum-entropy methods," *Proceedings of IEEE*, vol. 70, no. 9, pp. 939–952, 1982.

[23] P. Baggenstoss, "Time-series segmentation," *United States Patent 6907367*, June 2005.

[24] P. M. Baggenstoss, "Class-specific model mixtures for the classification of acoustic time-series," *IEEE Trans. AES*, Aug. 2016.

[25] P. M. Baggenstoss, "Derivative-augmented features as a dynamic model for time-series," in *2015 23rd European Signal Processing Conference (EUSIPCO)*, pp. 958–962, Aug 2015.

[26] M. Janvier, X. Alameda-Pineda, L. Girin, and R. Horaud, "Sound-Event Recognition with a Companion Humanoid," in *Humanoids 2012 - IEEE International Conference on Humanoid Robotics*, (Osaka, Japan), pp. 104–111, IEEE, Nov. 2012.

[27] J. Rissanen, "Modeling by the shortest data description," *Automatica*, vol. 14, pp. 465–471, 1978.

ISBN 978-90-827970-1-5 © EURASIP 2018