# ON THE MODELING OF TIME INFORMATION FOR AUTOMATIC GENRE RECOGNITION SYSTEMS IN AUDIO SIGNALS

**Nicolas Scaringella**
Signal Processing Institute (ITS-LTS3)
École Polytechnique Fédérale de Lausanne
EPFL, Lausanne, CH-1015 Switzerland
nicolas.scaringella@epfl.ch

**Giorgio Zoia**
Signal Processing Institute (ITS-LTS3)
École Polytechnique Fédérale de Lausanne
EPFL, Lausanne, CH-1015 Switzerland
giorgio.zoia@epfl.ch

## ABSTRACT

The creation of huge databases coming from both restoration of existing analogue archives and new content is demanding fast and more and more reliable tools for content analysis and description, to be used for searches, content queries and interactive access. In that context, musical genres are crucial descriptors since they have been widely used for years to organize music catalogues, libraries and shops. Despite their use musical genres remain poorly defined concepts which make of the automatic classification problem a non-trivial task. Most automatic genre classification models rely on the same pattern recognition architecture: extracting features from chunks of audio signal and classifying features independently. In this paper, we focus instead on the low-level temporal relationships between chunks when classifying audio signals in terms of genre; in other words, we investigate means to model short-term time structures from context information in music segments to consolidate classification consistency by reducing ambiguities. A detailed comparative analysis of five different time modelling schemes is provided and classification results are reported for a database of 1400 songs evenly distributed over 7 genres.

**Keywords:** musical genres, content analysis and indexing, machine learning, features extraction.

## 1 INTRODUCTION

Musical genres are the main top-level descriptors used by music dealers and librarians to organize their music collections. Though they may represent a simplification of one artist's musical discourse, they are of a great interest as summaries of some shared characteristics in music pieces.

With Electronic Music Distribution (EMD), music catalogues tend to become huge; in that context, associating a genre to a musical piece is crucial to help users in finding what they are looking for. In fact, the amount of digital music data urges for new means of automatic annotation since manual labeling would be too time-consuming.

At the same time, even if terms such as *jazz*, *rock* or *pop* are widely used, they remain poorly defined concepts so that the problem of automatic genre classification becomes a non-trivial task.

In this paper, we assume that genre taxonomy is given and we focus only on the ways to uniquely and automatically associate a song to a genre. More specifically, to improve results so far reported in literature special attention will be paid on different approaches to model the inner temporal structure of music; these approaches will be described and compared, and their impact on classification results will be evaluated. In this sense, the paper analyzes five different machine-learning algorithms that "encode" more or less explicitly relationships between successive audio chunks. These relationships represent an attempt to identify some temporal structural patterns inside music excerpts; even if these patterns are hidden and not appearing at the surface as a set of clear rules, nevertheless they emerge from music during machine training processes and allow obtaining robust results on rather general and non constrained data base.

The paper is organized as follows: section 2 will briefly review the state of the art in genre classification. Section 3 will describe the extraction of features characterizing the audio signal. Section 4 will present the classification schemes evaluated in this work while section 5 will be devoted to the discussion of results obtained on a database of 1400 songs evenly distributed over 7 genres. The last section will reach some conclusions.

## 2 RELATED WORK

Though unsupervised clustering of music collections based on similarity measures is gaining more and more interest in the music information retrieval community (see [1] and [2]), most works related to classification of music titles into genres are based on supervised techniques. These methods suppose that a taxonomy of genres is given and they try to map a database of songs into it by machine learning algorithms.

Soltau et al. [3] have compared a Hidden Markov Model to one new classification architecture, the ETM-NN (Explicit Time Modelling with Neural Networks) in a classification experiment involving 360 songs distributed over 4 genres.

Tzanetakis and Cook [4] and Li et al. [5] have worked on a database of 1000 songs over 10 genres and

have compared the use of different audio features (timbral features, rhythmic features, pitch features, wavelet features) and different classifier (Support Vector Machines, Gaussian Mixtures, Linear Discriminant Analysis, K-nearest neighbours) on time-independent chunks.

Burred and Lerch [6] have proposed a hierarchical classification scheme evaluated on a database of 850 songs over 17 classes (including some speech classes and background noise),

West and Cox [7] have used a Maximal Classification Binary Tree along with Linear Discriminant Analysis to classify a set of 900 songs into 6 genres.

Dixon et al. [8] extracted the main rhythmical pattern of a song and classify it according to features derived from the pattern, the tempo plus other timbral features. They evaluate their system on a database of 698 files in 8 subgenres of Standard and Latin ballroom dance music.

# 3 FEATURE EXTRACTION

The first step of analysis systems consists of extracting some *features* from the audio data to manipulate more meaningful information and to reduce the further processing of the classification task.

In this paper we focus on classification strategies taking into account as far as possible the time structure of a musical piece. Consequently, we select a fixed set of low-level features to characterize analyzed audio signals and to allow for direct comparison between classification schemes. However, it is clear that some particular features are more discriminative when trying to isolate one particular genre from another. As a matter of fact, feature selection techniques should be systematically used when trying to build a robust classifier.

### 3.1 Segmentation into analysis frames

The audio content used in our experiments is sampled at 44100 Hz and converted to a mono signal. The resulting signal is then analyzed through sliding windows of 20 ms overlapped by 50%. Each analysis frame is multiplied by a hamming-window and zero-padded to a power of two.

### 3.2 Timbre features

Mel-frequency Cepstral Coefficients (MFCC) are then computed from the analysis frames. Each analysis frame is parameterized with 6 MFCCs. MFCCs have proved to be successful in speech recognition applications and have since then been widely used in music genre classification ([3],[4],[6],[7]). They are a good choice in our case, as we did not try yet to select the features characterizing optimally a given genre.

### 3.3 Rhythmic features

In [9], Klapuri et al. introduce a beat, measure bars and tempo tracker for audio signals. It induces tempo from a so-called periodicity function, which summarizes the strength of different periodicities in the *region of pulse sensation*. Simple statistical descriptors including mean, standard deviation, skewness, kurtosis and maximum of the periodicity function are computed from this periodicity function to describe its shape, which characterizes the strength of the different periodicities and their relations.

### 3.4 Texture window

Both our timbre and rhythmic features may be computed at the analysis frame rate (i.e. every 10 ms). Yet the information contained at this time scale is not sufficient and too many variations occur. A solution is to average features at the frame rate over *texture windows* so that greater portions of the signal are considered.

In the following experiments, we compare three types of window – one vector per 30 seconds, one vector per 1 second and a more *musical* modelling: one vector centred on each beat averaging frames over the local beat period (with beats and beat rate extracted through the system proposed in [9]).

Mean, standard deviation, skewness and mean of the absolute value of each timbre feature are computed over the size of the considered window. Periodicity function is averaged over the considered window and its mean, standard deviation, skewness and maximum are evaluated. A vector of 28 features thus characterizes each texture window.

# 4 CLASSIFICATION SCHEMES

Once audio signals are parameterized in terms of feature vectors, the genre classification problem is reduced to a typical pattern classification task. Our goal being to associate an audio signal to a class of an a-priori defined set, we use a supervised approach.

In most reviewed papers in literature, genre classification is obtained by statically analysing an excerpt feature vector or subdividing the excerpt into chunks, which are considered as time-uncorrelated. However, in similar problems like e.g. speech recognition and natural language analysis, temporal relationships from one phoneme or word to the following provide important hooks to improve recognition and obtain more robust results.

Moving from a similar approach, special attention is paid in this paper to ways to represent temporal progression and contextual information in the classification process. Five classification schemes are evaluated with that purpose in mind: a Support Vector Machine (SVM), Support Vector Machines with delayed inputs, a recurrent neural network (the Elman network), an Explicit Time Modelling Neural Network (ETM-NN), and a Hidden Markov Model (HMM).

### 4.1 Support Vector Machines

The underlying idea of SVM classification [10] is to project data in a high dimensional space in which it is easier to separate into classes. A simple linear discrimi-

nant function which maximizes the margin between classes is then found in that high-dimension space.

Though SVMs have proved to be very efficient for classification tasks, they are not able to handle temporal sequences. As a matter of fact, they can only classify statically one vector into a given class.

A first solution to consider temporal evolution of musical signals is to consider texture windows shifted along time. The features associated to each texture window are classified independently and a majority vote is used to decide the class of the complete excerpt (as done in [4], [5], [6] and others in literature).

Three SVM classifiers with radial basis kernels were built. Multi-class classification is achieved by using error-correcting codes [11]. One classifier is trained with 30 seconds windows (it is referred to as **SVM-30s**). Another one is trained with 1 second windows (**SVM-1s**) and the last one with windows centred on beats with length the beat rate (**SVM-beats**). The feature vector associated with a window is classified independently and the final decision for a song is taken as the most represented class in the song. **SVM-30s** actually stands for the case where a single feature vector is used for the complete song as we have been working with excerpts of 30 seconds.

### 4.2 Support Vector Machines with delayed inputs

As it was already said, SVMs (as well as neural networks and a number of other machine learning algorithms) can only process static patterns.

A solution to handle temporal sequences is to build a spatial representation out of it and to use it as input of the classifier. By using a tapped delay line, one can present to the classifier a sequence of feature vectors.

Notice however that this scheme suffers from a number of weaknesses:

1. the delay must be large enough to contain a significant sequence: this implies that the number of parameters of the classifier will be larger and thus a very large number of examples is needed for the training.
2. the classifier is not invariant to time-shifting i.e. a very large number of examples is needed for every output class and every position in the delay line.
3. the classifier is sensitive to time-variation i.e. it requires the delays to precisely match the input time intervals (this may be corrected by having feature vectors synchronized to the beats of the musical signal).

Experiments were conducted using a delay line of 3 feature vectors so that each pattern presented to the SVM is the concatenation of 3 feature vectors corresponding to 3 adjacent texture windows. We denote by **SVM-delay-1s** the SVM trained with feature vectors corresponding to 1 second texture windows and by **SVM-delay-beats** the SVM with texture windows corresponding to beats with length equal to the beat rate.

### 4.3 Recurrent neural networks

The most widely used artificial neural network is the multi-layer perceptron. Neural networks suffer from the same weakness as SVMs, i.e. they can only process static patterns (SVMs have proved however to be more suited to classification tasks than neural networks).

Neural networks may be used in the same manner as the previously presented SVMs (with texture windows and with delayed inputs). An alternative is to use a recurrent or partially recurrent network (i.e. all or some of the layers of the network have their output connected to their input). More specifically, we evaluate the performance of the Elman network [12], which is typically a two-layer network with feedback from the first layer output to the first layer input. The feedback connections allow taking the close past into account when classifying a new feature vector.

We use fully connected networks with 100 neurons. Two cases are considered: **ELM-1s** refers to the Elman network fed with vectors corresponding to 1 second windows while **ELM-beats** refers to the Elman network with vectors corresponding to windows centred on beats with length equal to the beat rate. The final decision for the complete song is obtained by integrating over time each output and selecting the maximal integrated output as the correct class.

### 4.4 Explicit Time Modelling with Neural Networks

Soltau et al [3] have introduced in the context of recognition of music genres an original method for explicit time modelling of temporal structure of music. This new architecture is referred to as ETM-NN (Explicit Time Modelling with Neural Networks).

In this architecture, a multi-layer perceptron is trained to recognize given input feature vectors that are 0.4 seconds long. The main idea is to use the hidden layer of the perceptron and not its output, which is supposed to give the genre. As a matter of fact, it is known that the first half of a feed-forward network performs a specific non-linear transformation of the input data into a space in which the discrimination should be simpler.

The activation of these hidden neurons corresponds to the use of a compact representation of the input feature vector. Each hidden neuron can be understood as an abstract musical event – not necessarily related to an actual musical meaning. An abstract event $e_i$ occurs if the hidden unit $i$ has the highest activation of all hidden units.

The sequence of abstract events over time is then analysed to build one single feature. More specifically, the number of events $e_i$ (unigram), the number of pairs $e_i\ e_j$ (bigram) and the number of triplets $e_i\ e_j\ e_k$ (trigram) are evaluated as well as the event durations, the mean, the maximum and the variance of event activations. All of these features, normalized over the length of the sequence are combined into a single vector which is given to another neural network; this latter implements the final decision about the genre of the musical piece.

We consider two cases: **ETMNN-1s** with evaluation of abstract events for each 1 second windows and

**ETMNN-beats** with evaluation of abstract events for each beat. All the considered neural networks have a single hidden layer with a number of neurons equal to the number of genres considered.

### 4.5 Hidden Markov models

Hidden Markov models have been extensively used in speech recognition [13] because of their capacity to handle time series data. HMMs may be understood as a doubly embedded stochastic process: one process is not observable (hidden) and can only be observed through another stochastic process (observable) that produces the time set of observations.

A HMM is defined by its number of states, the transition probability between its states, the initial state distribution and the observation symbol probability distribution.

One HMM was trained per musical genre using mixtures of 3 Gaussians to model the state probability densities. Each HMM has 4 states and an ergodic transition model. Other topologies have been briefly explored but some additional work would be needed to find the best topology depending on the genre.

Here again were consider two cases: **HMM-1s** for the sequence of 1 second windows and **HMM-beats** for the sequence of windows centred on beats.

## 5 EXPERIMENTAL RESULTS

Experimentations were run to compare the performances of the five main classification schemes and the type of bag of frames.

### 5.1 Dataset

The dataset used contain 1400 songs over 7 genres (i.e. 200 songs per genre). For each song, 30 seconds after the initial 30 seconds were used. The used genres are: Blues, Classical, Electronica, Jazz, R&B/Soul, Rap and Rock. Songs were assigned a genre label according to the AllMusic[1] guide.

### 5.2 Experiment setup

The reported results are obtained by 10-cross validation. Namely, for each classification scheme, experiments are run 50 times: each time is characterized by a different random split of the database (90 % of training data and 10 % of testing data). Reported results give the average of the 50 runs.

### 5.3 Results

Table 1 shows the accuracy of the different classification schemes. Recognition rates are given in percentage for the 7 genres and for the complete set (the best result among all classifiers for specific genres are in bold).

As a comparison, it should be remembered that the accuracy of random guess on this dataset is 14%. Moreover the human performance for genre classification has been studied in [14]: college students were able to classify songs correctly in 70% of the cases after listening to 3 seconds of the songs (on a database where chance would give 10%). This result should not be understood as an upper boundary to automatic classification accuracy but rather as an expected accuracy.

### 5.4 Analysis of the different classifiers

The SVM with delayed inputs and texture windows synchronised on beats gives the best overall results (69.98% of correct classification). Yet some methods show significantly better results on some specific genres. As a matter of fact, likewise some features are more suitable than others when classifying into a given set of subgenres, some classification schemes may be more suited to a particular genre. Let us discuss more in depth the pros and cons of each classification scheme.

#### *5.4.1 Results of SVMs*

Support vector machines are known to give excellent results in many classification tasks including musical genre classification (SVMs are successfully used in [5]).

It is interesting to notice how in our experiments the size of the texture window influences the results. In the case of Electronica, Jazz and R&B/Soul, better results are obtained when considering a window of all frames rather than smaller chunks of data. It may be understood in the case of Electronica as smaller chunks may be locally misclassified because of the use of samples from other genres (mostly jazz, soul and funk).

The use of beat-synchronised texture windows improves results only for Rock when used with SVMs. This may be explained of course by the fact that beats are not extracted perfectly. Another reason is that the faster the detected tempo, the smaller is the texture window. This may be harmful as Tzanetakis and Cook report in [4]: when the window is too small, the classification accuracy drops.

#### *5.4.2 Results of SVMs with delayed inputs*

SVMs with delayed inputs give the best classification results. The inherent weaknesses of such architecture (see section 4.2) are probably overcome by the large amount of data used in our experiments (200 songs per genre).

The fact that SVMs with delayed inputs beat simple SVMs in the genre classification task is an interesting result: it confirms the importance of time structure in musical genre understanding. Indeed SVMs with delayed inputs encode a simple (non-explicit) representation of time structure by considering adjacent texture windows (with a delay of 3 windows, unigrams, bigrams and trigrams are encoded).

However the training time of such a model is considerably larger than the training of a SVM with single texture windows.

[1] http://www.allmusic.com

| | Blues | Classical | Electronica | Jazz | R&B/Soul | Rap | Rock | All genres |
|---|---|---|---|---|---|---|---|---|
| **SVM-30s** | 72.12 | 89.89 | 49.43 | 67.42 | **54.02** | 70.32 | 62.18 | 66.48 |
| **SVM-1s** | 78.07 | 91.76 | 38.90 | 61.51 | 43.31 | 79.03 | 66.95 | 65.65 |
| **SVM-beats** | 76.48 | 90.77 | 29.92 | 60.25 | 39.19 | 77.20 | 69.36 | 63.31 |
| **SVM-delay-1s** | 84.95 | 92.33 | **52.35** | 65.49 | 51.01 | 75.43 | 64.27 | 69.40 |
| **SVM-delay-beats** | **86.74** | 89.50 | 45.58 | **71.50** | 52.75 | 76.29 | 67.52 | **69.98** |
| **ELM-1s** | 66.90 | 89.56 | 30.60 | 56.99 | 41.25 | 70.92 | 61.30 | 59.65 |
| **ELM-beats** | 66.77 | 87.92 | 33.18 | 60.13 | 39.33 | 72.25 | 64.34 | 60.56 |
| **ETMNN-1s** | 69.18 | 90.12 | 30.42 | 57.67 | 41.76 | 71.23 | 62.20 | 60.37 |
| **ETMNN-beats** | 68.95 | 89.60 | 32.15 | 61.18 | 39.78 | 73.02 | 64.79 | 61.35 |
| **HMM-1s** | 66.41 | 91.53 | 27.60 | 58.99 | 36.93 | **84.72** | 72.56 | 62.68 |
| **HMM-beats** | 61.54 | **92.98** | 30.37 | 64.08 | 33.56 | 82.91 | **74.65** | 62.87 |

Table 1. Recognition rate for the different genres and classifiers

*5.4.3 Results of Elman networks*

Elman networks give the worst overall results. Yet their modelling of time structure, though it may be too simple, is comparable to the modelling of networks with delayed inputs.

Indeed, each texture window is classified according to its feature vector plus a feedback of the previous state of the hidden units of the network. In the recurrence, the hidden units are decreased by a multiplicative constant, which determines the memory depth of the network. Thus the network models some local structure by taking into account adjacent windows with a decaying integration factor; this is similar to the case of networks with delayed inputs, considering that this time there is no integration factor and the memory depth is fixed by the size of the delay.

In other words, neural networks with delayed inputs may be compared to Finite Impulse Response filters while recurrent networks may be compared to Infinite Impulse Response filters: the two architectures are able to model the same problems as long as their parameters are properly estimated. Recurrent networks may indeed be as efficient as networks or SVMs with delayed inputs. Yet a general weakness of systems with feedback loops is their tendency to become instable and this also applies to recurrent networks.

As a matter of fact, because of possible instability problems, Elman networks are particularly tricky to train properly. In our experiments, Elman networks were sometimes overspecialized for certain classes while being weak for other classes (though on average it is not noticeable in the presented results).

*5.4.4 Results of ETMNNs*

The results obtained with ETMNNs are a little disappointing compared to those reported by Soltau et al. [3]: he reports that his architecture significantly outperforms HMMs while this does not occur in our case.

In fact, our implementation of the ETMNN differs slightly from the one initially proposed by Soltau: we use the same feature vectors as in our previous experiments (vectors of dimension 28) while he uses vectors composed of the concatenation of the first 5 cepstral coefficients of 10 adjacent frames of 50 ms (vectors of dimension 50). In any case, our results in terms of ETMNN and HMM performances are sometimes difficult to compare to those reported by Soltau as he used a database of 360 songs over 4 genres (rock, pop, techno and classical) while we used a database of 1400 songs over 7 genres.

In any case, the ETMNN architecture is not so different from the HMM architecture. As a matter of fact, the first network is selecting an *abstract event* in the terminology of the ETMNN: such event correspond to an HMM *state*.(notice by the way, that neural networks can be used to model the probability densities of the HMM states). Moreover the use of events' unigrams and bigrams correspond to the connections between states in a standard HMM. To model trigrams, one has to consider second order relations between states of an HMM which is usually not the case with HMMs (in most HMMs settings, the so-called *first-order Markovian assumption* is supposed, i.e. the probability of being in a state depends solely on the previous state).

*5.4.5 Results of HMMs*

HMMs are more suited to model time sequences than any other experimented model. Yet they are outperformed by SVMs for the overall performance. As a matter of fact, a number of hypotheses, which make it possible to optimize these models, limit their generality and are at the root of some of their weaknesses.

HMMs have indeed a low discriminative power because they are usually trained with a maximum likelihood criterion rather than with an optimal maximum *a-posteriori* criterion. In other words, during the training, the likelihood that the model of a genre did produce an

observation is maximised but the likelihood of the other models is not minimized. A number of alternative solutions for the training of HMMs have been proposed to ensure them a better discrimination power [15].

Moreover, in our experiments we have modelled the emission probability of the different states by mixtures of three Gaussians. This implies a strong assumption on the distribution of the emission probability. This assumption may be relaxed by modelling emission probabilities with neural networks (see [16]).

## 6 CONCLUSION

We have compared 5 different methods taking low-level, short-term time relationships into account to classify audio excerpts into musical genres. SVMs with delayed inputs proved to give the best results with a simple modelling of time structures. However, we do not claim overall that SVMs perform better than other classifiers such as multi-layer perceptrons or linear discriminant analysis: instead, the main outcome is that a simple model (using context and synchronisation on musical rhythm) somehow improves musical genre classification results in many cases.

Reported results may now be greatly improved by considering hierarchical classification techniques to model the underlying genre taxonomy. Feature selection at each node of the hierarchy would allow optimization of the classification task to a specific set of genres. Since some classification algorithms seem more suitable to some particular genres, one could also consider using different classifiers for each node of the hierarchy or using multiple classifiers and combining their results like in architectures based on a mixture of experts.

## REFERENCES


[1] A. Rauber, E. Pampalk, D. Merkl, "Using psycho-acoustic models and self-organizing maps to create a hierarchical structuring of music by sound similarity", in Proc. 3rd Int. Conf. on Music Information Retrieval (ISMIR), Paris, France, 2002.

[2] X. Shao, C. Xu. M. Kankanhalli, "Unsupervised classification of musical genre using hidden Markov model", in IEEE Int. Conf. of Multimedia Explore (ICME), Taibei, Taiwan, China, 2004.

[3] H. Soltau, T. Schultz, M. Westphal, A. Waibel, "Recognition of music types", in Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP), Seattle, USA, 1998.

[4] G. Tzanetakis, P. Cook, "Musical genre classification of audio signals", in IEEE Transactions on Speech and Audio Processing, Vol. 10, No. 5, July 2002.

[5] T. Li, M. Ogihara, Q. Li, "A comparative study on content-based music genre classification", in Proc. Of the 26th Annual Int. ACM SIGIR Conf. on Research and Development in Information Retrieval, Toronto, Canada, 2003.

[6] J.J. Burred, A. Lerch, "A hierarchical approach to automatic musical genre classification", in Proc. Of the 6th Int. Conf. on Digital Audio Effects (DAFx), London, UK, 2003.

[7] K. West, S. Cox, "Features and classifiers for the automatic classification of musical audio signals", in Proc. of the 5th Int. Conf. on Music Information Retrieval (ISMIR), Barcelona, Spain, 2004.

[8] S. Dixon, F. Gouyon, G. Widmer, "Towards characterization of music via rhythmic patterns", in Proc. Of the 5th Int. Conf. on Music Information Retrieval (ISMIR), Barcelona, Spain, 2004.

[9] A. Klapuri, A. Eronen, J. Astola, "Analysis of the meter of acoustic musical signals", in IEEE Trans. Speech and Audio Proc., 2004.

[10] C. Burges, "A tutorial on support vector machines for pattern recognition", in Data Mining and Knowledge Discovery, 2, 121-167, 1998.

[11] T.K. Huang, R.C. Weng, C.J. Lin, "A generalized Bradley-Terry model: from group competition to individual skill", in 18th Int. Conf. on Neural Information Processing Systems, 2004.

[12] J.L. Elman, "Finding structure in time", in Cognitive Science, vol. 14, pp. 179-211, 1990.

[13] L. Rabiner, "A tutorial on Hidden Markov Models and selected applications in speech recognition", in Proc. of the IEEE, 77:2 (257-286), 1989.

[14] D. Perrot, R. Gjerdigen, "Scanning the dial: an exploration of factors in identification of musical style", in Proc. Society for Music Perception and Cognition, Evanston, IL, USA, 1999.

[15] S. Katagiri, C.H. Lee, B.H. Juang, "New discriminative training algorithms based on the generalized probabilistic descent method", in IEEE Proc. Workshop on Neural Networks for Signal Processing, pp. 299-308, 1991.

[16] N. Morgan, H. Bourlard, "Continuous speech recognition: an introduction to the hybrid HMM/connectionist approach", in IEEE Signal Processing Magazine, vol. 12, n°3, pp. 25-42, 1995.