# A tool for analyzing audio and video advertising in real time

S. Barbato, A. Scarone* and G. Rubino†
Contact address: G. Rubino,
INRIA / IRISA,
Campus de Beaulieu
35042 Rennes Cedex, France
E-mail: rubino@irisa.fr

June 7, 2005

**Abstract**

Recognizing an audio sequence among a set of $n$ previously stored ones is a standard task in signal processing if (i) $n$ is small and (ii) we have as much time as desired to perform the task. Even in these cases, we must set the parameters of the process (number of coefficients in the decompositions used, sampling rates...) "high" enough, say, if the error probabilities must be kept very small.

This paper points out the fact that these standard techniques can also handle the case of very large values of $n$ and very small processing times following two lines: first, combining several different standard techniques and second, controlling the techniques and their combination using a powerful optimization process.

## 1 Introduction

The problem considered here is the recognition of an audio signal received through a radio or TV channel, that is, the identification of the signal among a large data base of previously stored ones. It can be seen as belonging to the pattern recognition area, more specifically, pattern recognition with errors (as in [7]). The difficulties here are twofold. First, the goal is to reach a very high accuracy in the recognition process, that is, very low probabilities of false positive and false negative answers. Second, the process must cost less in time than the time needed to play the flow (that is, the radio channel), and, if possible,

*S. Barbato and A. Scarone are with *Mediciones y Mercado*, San Salvador 2222, Montevideo CP 11200, Uruguay

†G. Rubino is with INRIA/IRISA, Campus de Beaulieu, 35042 Rennes Cedex, France

much less. A third difficulty is also quantitative in nature: among the problems to be solved we had the fact that the data base of previously stored sequences was assumed to be very large (hundreds of thousands of items). This is also a strong constraint to handle.

In principle, this task can be done using standard techniques, for instance, by decomposing the signals using Fourier transforms and comparing the coefficients of the received sequence to those of the components of the data base. Other possible approach is to evaluate correlation coefficients again between the received sequence and the signals in the base. More sophisticate possibilities exist through some modelling of the audio sequences (for instance using Hidden Markov Models). All these techniques are well established, and some of them are used for instance in cellular phones (the problem is, however, slightly different, because the followed procedure is a supervised learning one; moreover, the emitted signal is always sent by the same phone user, under similar circumstances). The main problem in the application discussed here is the computational cost when the size of the data base is large, which means here hundreds of thousands of elements. To the best of our knowledge, there is no standard technique allowing to reach a high accuracy level with very low processing times, specifically using less than the time needed to play the whole channel flow (that is, in "less than real time"), and working with large data bases.

The goal of this paper is to claim that, finally, basic signal processing techniques can allow performing the described task with high performances, (i) using several techniques at the same time and (ii) through a fine tuning of their parameters and a combination of their outputs, both processes optimized using a powerful tool called a Random Neural Network. It has then been possible to build a system capable of automatically analyzing the advertising segments of radio or TV streams only through their associated audio channels, and to implement the resulting tool into a commercial and operating system. (For details about the tool, the reader can go to `www.avauditor.com`[1]). The paper also describes some important problems that can be the subject of future research efforts.

## 2 The problem and the performances reached by our solution

Consider an advertising analysis system operating in the following way: TV and/or radio (AM and/or FM) channels are analyzed and the different ads are manually copied into a database the first time they appear (in particular, when the system starts operating). Then, the system continuously monitors a selected

[1](The text in those pages is in Spanish; it will be soon in English.) The commercial system implementing these detection tools offers many other services through a general architecture allowing analyzing an arbitrary number of channels using an arbitrary number of interconnected computers like load balancing, fault tolerance, etc. The system is today fully operational; it is currently analyzing 17 media in real time and has more than 250000 ads in its database.

(by the user) set of channels (TV, audio AM, audio FM) and automatically detects when and where each of the ads stored in the database appears, or when and where a new ad appears.

## 2.1 Global Efficiency Factor

Call Global Efficiency Factor (GEF) the ratio between 1 hour and the time the hardware needs to analyze 1 hour of signal. The task previously described has been always considered as a hard one if GEF factors much higher than 1 are targeted (this, obviously, with a high accuracy).

We have been exploring different existing approaches and by a fine combination of standard signal processing techniques and the help of powerful optimization methods used to find good values for the parameters controlling the way those techniques are used and merged, we reach today values of GEF close to 30; that is, 1 hour of signal is analyzed in about 2 min[2]. It must be underlined that this performance is achieved using standard hardware (composed of good but common workstations and audio processing cards).

## 2.2 Automatic Recognition Index

Another important factor is the Automatic Recognition Index (ARI), defined as the ratio between the correct detections over the number of ads analyzed. The system currently working reaches ARI = 97%. This means that in 97 cases over 100, on the average, the system correctly detects an ad in the stream, the remaining 3% consisting of false positives and false negatives. This high ARI partly explains the value GEF = 30. Observe that this combination of high GEF and high ARI resumes the difficulty of the problem and, at the same time, is the key of the technical success of the methodology. Another important parameter is the precision of the system in detecting the beginning and the end of each ad in the steams, where the same signal analysis tools appropriately combined allow reaching an optimal precision in more than 99% of the cases.

In the following sections we describe our general procedure and the key mathematical tool that has been used to allow the announced performances. It must be pointed out that even for TV commercials, there is no need for specific techniques specially designed for video analysis (such as those in [8]), audio methods are powerful enough to reach the performance level needed by the application. In the Conclusions of the paper we also discuss some possible directions for further research.

[2]This time is partly due to the human component of the work impossible to avoid, since needed to check the system's results and to add new commercials to the database.

# 3 The global approach

Assume you first make a choice between the possible available techniques you can use to perform the recognition task. For instance, you can consider a Fourier decomposition and select only the first $k$ coefficients. You must also decide which sliding window $w$ use. The same is done for other methods. The result is a set of techniques, each characterized by a small set or parameters.

The second step is to consider simple algorithms to combine the answers of the different standard techniques considered before into a single one. For instance, one of the things we tried was to use two versions of the same basic technique with different settings leading to different performances and different costs (an example is correlations between sequences with two different window sizes $w_1$ and $w_2$, $w_1 < w_2$; the first one was used a preliminary filter to eliminate bad candidates to matching, the second one allowing finer discriminations). In some experiments we tested giving weights to different techniques and composing the global answer using a weighted sum which was compared to a threshold.

A last and important set of parameters $\mathcal{P}$ was added to these two lists. They consisted of numbers "characterizing" some way the sequences (for instance, the number of changes of signs in the derivative of the signal over some time interval, or its energy), even if the characterization was obviously extremely rough. The key property here was to use numbers (we also call them *descriptors*) computable very quickly as the signal arrived.

The idea was then to use an important number of sequences for which a human operator checked whether they were represented in the data base or not. Then, optimization methods were used in order to identify the best combination of parameters and techniques, taking into account the sequence itself (through the values of the descriptors in $\mathcal{P}$), from the performance point of view. This led to a very efficient procedure where the system dynamically adapts the parameters controlling the basic tools and their combination to the received sequence. The obtained performance is characterized by the numbers given before.

In the optimization process, we discretized all the variables and we used first a standard GRASP (Greedy Randomized Adaptive Search Procedure, [6], [12]) procedure. Then, we applied a powerful technique to refine the optimums. For this purpose, we used a Random Neural Network, briefly described in the next section. It is in part responsible of the good performance levels reached by the tool[3].

# 4 Random Neural Networks

A Random Neural Network can be seen as a very specific queuing network. The model was invented by E. Gelenbe in a series of papers [1, 2, 5] by merging concepts from neural networks and queuing theory.

[3]The details about which basic techniques were finally selected with which specific parameters and values, how they were combined and how the specific neural network was designed are confidential, because of the commercial implementation of the procedure.

The set of neurons in the network is $\mathcal{N} = \{1, \cdots, N\}$. The network receives two types of signals from outside, called *positive* and *negative*. Neuron $i$ receives positive signals from outside with rate $\lambda_i^+ \geq 0$ and negative ones from outside with rate $\lambda_i^- \geq 0$. Both signal arrival processes are Poisson. At least one neuron receives positive signals from outside; that is, $\sum_{i\in\mathcal{N}} \lambda_i^+ > 0$.

At each time $t$, a neuron in the network has a *potential*, which is a non-negative integer. The neuron is said to be *active* if its potential is strictly positive. When neuron $i$ is active, it sends signals to another neuron or to outside with rate $\mu_i > 0$. The signal goes to neuron $j$ as a positive one with (routing) probability $p_{i,j}^+$ and as a negative one with (routing) probability $p_{i,j}^-$. The signal is sent to outside with probability $d_i = 1 - \sum_{j\in\mathcal{N}}(p_{i,j}^+ + p_{i,j}^-)$.

The previous description means that when neuron $i$ is active, it sends positive signals to neuron $j$ with rate (also called *weight*) $w_{i,j}^+ = \mu_i p_{i,j}^+$ and negative ones to neuron $j$ with rate $w_{i,j}^- = \mu_i p_{i,j}^-$; it sends signals to outside with rate $\delta_i = \mu_i d_i$. Observe that $\delta_i + \sum_{j\in\mathcal{N}}(w_{i,j}^+ + w_{i,j}^-) = \mu_i$.

When a neuron sends a signal, its potential decreases by one unit. When it receives a positive signal from outside or from another neuron, its potential increases by one, and in case of a negative signal, its potential decreases by one. Signals travel in the network instantaneously.

Denote by $X_i(t)$ the potential of neuron $i$ at time $t$. In the stable case, the *activity rate* of neuron $i$ is $\varrho_i = \lim_{t\to\infty} \Pr(X_i(t) > 0) > 0$; also, the mean throughput of positive signals that arrive at neuron $i$ is $T_i^+ = \lambda_i^+ + \sum_{j\in\mathcal{N}} \varrho_j w_{j,i}^+$, and the mean arrival throughput of negative signals at $i$ is $T_i^- = \lambda_i^- + \sum_{j\in\mathcal{N}} \varrho_j w_{j,i}^-$.

Assume process $X_i()$ is stationary. The mean flow conservation theorem applied to signals gives $T_i^+ = \varrho_i(\mu_i + T_i^-)$ and thus,

$$\varrho_i = \frac{T_i^+}{\mu_i + T_i^-}.$$

Last, assume the standard independence conditions concerning all arrival, service and switching (choosing next neuron and signal class, or output, for a signal leaving a neuron) processes. Then (see [1, 2]), the vector of potentials $(X_1(t), \cdots, X_N(t))$ is a Markov chain with state space $\mathbb{N}^N$. Assuming that it is irreducible (this depends on the routing probabilities of the model and on the arrival rates), and considering the relations

$$T_i^+ = \lambda_i^+ + \sum_{j\in\mathcal{N}} \varrho_j w_{j,i}^+,$$

$$T_i^- = \lambda_i^- + \sum_{j\in\mathcal{N}} \varrho_j w_{j,i}^-$$

$$\text{and } \varrho_i = \frac{T_i^+}{\mu_i + T_i^-}$$

as a (non-linear) system of equations in the set of unknowns $(T_i^+,\ T_i^-,\ \varrho_i)_{i=1,\cdots,N}$ we have:

(i) The network is stable iff the system of equations has a solution where for $i = 1, \cdots, N$ we have $\varrho_i < 1$; in this case, the solution is unique.

(ii) In the stable case, the network is of the product-form type, and we have

$$\lim_{t\to\infty} \Pr(X_1(t) = n_1, \cdots) = \prod_{i=1}^{N} (1 - \varrho_i)\varrho_i^{n_i}.$$

We have described so far a dynamical system using a terminology similar to that of Artificial Neural Networks (ANNs). This tool can effectively be used as an ANN, either for learning or for optimizing, and its performances are reported to be excellent. We specifically compared it to other tools (see some remarks on this in [10]) and its behaviour was by far the best observed one. In the application described here we used it as an optimization tool, as with any analogous ANN. For examples of this, see for instance [3], or [4]. To complete the general description, the rates of the flows of signals between neurons play exactly the same role as the classic weights in ANNs. The states of the neurons are the numbers $\varrho_i$'s, with the convention that in the unstable case, $\varrho_i = 1$. This can happen for some of the neurons in the network (that is, some neurons can be saturated while other neurons in the network have stationary potential processes associated with). We do not develop this issue further here; see the references given in the paper.

In the case we use the model to learn some, say, real function $f()$, the learning variables are again the weights $w_{ij}^+$ and $w_{ij}^-$, and the inputs of $f()$ are mapped into the rates of the arrival processes. Since we assumed that $f()$ had values in $\mathbb{R}$, the network will have a single output neuron, $o$, and seen as a black-box, the output of the network is the activity rate $\varrho_o$ of neuron $o$. Learning means finding values of the weights such that the output of the RNN is close to the value given by $f()$, for the same input vector.

In both applications (optimization or learning), the basic step is the computation of the activity rates of the neurons in the network. Due to the general result described before, this consists of solving the previously described nonlinear system of equations, looking for the vector $(\varrho_1, \cdots, \varrho_N)$. We can write it as the system with $N$ equations and $N$ unknowns

$$\varrho_i = \frac{\lambda_i^+ + \sum_{j\in\mathcal{N}} \varrho_j w_{j,i}^+}{\mu_i + \lambda_i^- + \sum_{j\in\mathcal{N}} \varrho_j w_{j,i}^-}, \quad 1 \le i \le N.$$

This fixed-point equation can then be easily solved by the immediate and usual iterative process. To deal with the unstable case, the approximation used is simply the following: starting from some $\varrho_i^{(0)} < 1$, $i = 1, \cdots, N$,

$$\varrho_i^{(k+1)} = \min\left\{1\ ,\ \frac{\lambda_i^+ + \sum_{j\in\mathcal{N}} \varrho_j^{(k)} w_{j,i}^+}{\mu_i + \lambda_i^- + \sum_{j\in\mathcal{N}} \varrho_j^{(k)} w_{j,i}^-}\right\}.$$

For examples of uses of this tool, see [9] and the references therein. See also [10] where we used it specifically for learning, or [11] in this same conference.

# 5 Conclusions

This paper reports how an automatically designed combination of elementary and standard signal processing techniques allowed building a global system reaching high performances in solving a difficult audio analysis task. The task is the identification of a sequence having a few dozens of seconds length (a commercial) inside a very large data base of previously stored sequences, in much less time than needed to play the *supporting* sequence (the flow sent through the channel), with very high accuracy (very low error rates) and using standard hardware. This was surprising since the problem as stated is assumed to be hard in the area.

The present performance reached by our approach is high enough for a successful commercial application, but further improvements can still be possible, in particular finding new descriptors providing a better characterization of sequences (or sequences types). Also, the success of the technique allowed using only audio analysis, without looking at the video part of the commercials. One of the on-going research directions today is to combine now the analysis of the two parts of the signals, to improve further the global performances of the tool.

# References


[1] E. Gelenbe. *Random Neural Networks with negative and positive signals and product form solution*. Neural Computation, 1(4):502–511, 1989.

[2] E. Gelenbe. *Stability of the Random Neural Network model*. In Proc. of Neural Computation Workshop, pages 56–68, Berlin, Germany, February 1990. ISBN 3540522557.

[3] E. Gelenbe and F. Batty. *Minimum Cost Graph Covering with the Random Neural Network*. in Computer Science and Operations Research, pp. 139–147, Pergamon Press, 1992.

[4] E. Gelenbe, V. Koubi and F. Pekergin. *Dynamical Random Neural Network approach to the Travelling Salesman Problem*. In IEEE Symposium on Systems, Man and Cybernetics, 2:630–635, 1993.

[5] E. Gelenbe. *Learning in the recurrent Random Neural Network*. Neural Computation, 5(1):154–511, 1993.

[6] T. A. Feo and M. G. C. Resende. *Greedy randomized adaptive search procedure*. Journal of Global Optimization, 6:109–133, 1995.

[7] R. Baeza-Yates and C. Perleberg. *Fast and practical approximate pattern matching*. Information Processing Letters, 59:21-27, 1996.

[8] J. M. Sánchez and X. Binefa. *Automatic digital TV commercial recognition*. Proceedings of the VIII National Symposium of Pattern Recognition, Bilbao, 1: 313-320, 1999.

[9] H. Bakircioglu and T. Kocak. *Survey of Random Neural Network applications*. European Journal of Operational Research, vol. 126, no. 2, pp. 319–330, 2000.

[10] S. Mohamed, G. Rubino, and M. Varela. *Performance evaluation of real-time speech through a packet network: a Random Neural Networks-based approach*. Performance Evaluation, vol. 57, no. 2, pp. 141–162, 2004.

[11] G. Rubino and M. Varela. *Wireless VoIP at Home. Are We There Yet?*. In Mesaqin'05, Praga, June 2005.

[12] H. Cancela, G. Rubino and F. Robledo. *A GRASP algorithm with RNN based local search for designing a WAN access network*. To appear in Electronic Notes in Discrete Mathematics, 2005.