TecnoLógicas

Artículo de Investigación/Research Article

Detección de Eventos Sonoros en Señales de Música Usando Procesos Gaussianos

Sound Event Detection for Music Signals Using Gaussian Processes

Pablo A. Alvarado-Durán[1]
Mauricio A. Álvarez López[2]
Álvaro A. Orozco-Gutiérrez[3]

Recibido: 07 de mayo de 2013
Aceptado: 15 de agosto de 2013

1 Ingeniero Electrónico, Programa de Ingeniería Eléctrica, Universidad Tecnológica de Pereira, Pereira-Colombia
paalalvarado@utp.edu.co
2 Doctor en Ciencias de la Computación, Programa de Ingeniería Eléctrica, Universidad Tecnológica de Pereira, Pereira-Colombia
malvarez@utp.edu.co
3 Doctor en Bioingeniería, Programa de Ingeniería Eléctrica, Universidad Tecnológica de Pereira, Pereira-Colombia
aaog@utp.edu.co

Resumen

En este artículo se propone una metodología para detectar eventos sonoros en señales de música usando procesos Gaussianos. En el algoritmo presentado, las señales de audio de entrada son transformadas a un espacio tiempo-frecuencia utilizando la Transformada de Tiempo Corto de Fourier para obtener el espectrograma, cuya dimensión es posteriormente reducida pasando de la frecuencia en escala lineal en Hertz a la escala logarítmica en Mel por medio de un banco de filtros triangulares. Finalmente, se clasifica entre "evento" y "no evento" cada uno de los espectros de tiempo corto contenidos en el espectrograma en escala Mel por medio de un clasificador binario basado en procesos Gaussianos. Como parte del proceso de evaluación, se compara el desempeño de la metodología propuesta con el desempeño de algunas técnicas ampliamente utilizadas para detectar eventos en este tipo de señales. Para tal fin, se implementa en MATLAB® cada una de estas técnicas y se ponen a prueba utilizando dos bases de datos compuestas por segmentos de audio de diferente complejidad; definida por el tipo y cantidad de instrumentos tocados al mismo tiempo. Los resultados indican que la metodología propuesta supera el desempeño de las técnicas hasta ahora planteadas, presentando un mejoramiento en la medida F de 1,66 % para la base de datos uno y de 0,45 % para la base de datos dos.

Palabras clave

Clasificación con procesos Gaussianos, aprendizaje de máquina supervisado, espectrograma, detección de eventos, señales de música.

Abstract

In this paper we present a new methodology for detecting sound events in music signals using Gaussian Processes. Our method firstly takes a time-frequency representation, i.e. the spectrogram, of the input audio signal. Secondly the spectrogram dimension is reduced translating the linear Hertz frequency scale into the logarithmic Mel frequency scale using a triangular filter bank. Finally every short-time spectrum, i.e. every Mel spectrogram column, is classified as "Event" or "Not Event" by a Gaussian Processes Classifier. We compare our method with other event detection techniques widely used. To do so, we use MATLAB® to program each technique and test them using two datasets of music with different levels of complexity. Results show that the new methodology outperforms the standard approaches, getting an improvement by about 1.66 % on the dataset one and 0.45 % on the dataset two in terms of F-measure.

Keywords

Gaussian processes classification, supervised machine learning, spectrogram, event detection, music signals.

1. INTRODUCCIÓN

Una señal de música puede considerarse como una señal de audio compuesta por una sucesión de eventos, tales como una nota tocada, el canto de una voz, un instrumento de percusión siendo golpeado, o cualquier otro sonido. Localizar el instante de tiempo en el cual empieza cada evento se conoce como *Detección de Inicio* (Zhou *et al.*, 2008). La detección de eventos sonoros es útil para extraer características de la música tales como tempo, beat y ritmo, a partir de señales de audio. Prueba de ello es que muchos procesamientos de alto nivel como *Transcripción Automática de la Música* (Klapuri y Davy, 2006)*, Seguimiento del Beat* (Degara *et al.*, 2012)*, Seguimiento del Tempo* (Cemgil *et al.*, 2001), y *Acompañamiento Interactivo* (Robertson y Plumbley, 2007) tienen como parte fundamental una etapa de detección de eventos. Las técnicas utilizadas con este fin hacen parte del campo de investigación multidisciplinario *Music Information Retrieval* (MIR). Dicho campo abarca ciencias de la computación y extracción de información, musicología y teoría de la música, ingeniería de audio y procesamiento digital de señales, ciencias cognitivas, entre otros (Futrelle y Downie, 2002).

El enfoque general para localizar los instantes donde empiezan los eventos se compone de dos etapas denominadas reducción y selección de picos. La etapa de reducción consiste en extraer una representación intermedia que refleje de forma simplificada la estructura local de la señal original, esto se conoce como *curva de novedad* o función de detección (Degara *et al.*, 2011). Esta representación facilita la ubicación de eventos, los cuales se manifiestan como picos. Estas curvas se extraen para evitar analizar directamente la señal de música, lo cual se considera inapropiado debido a su complejidad (Bello *et al.*, 2005). En la segunda etapa, se extrae la ubicación de los picos presentes en la curva de novedad. En esta etapa primero se pos-procesa la curva de novedad, suprimiendo la media local y aplicando un filtro pasa bajas con el fin de reducir el ruido. Posteriormente, debido a que no todos los picos presentes en la función de detección corresponden a eventos, se decide cuáles picos serán descartados y cuáles no, con base en un umbral adaptativo. Finalmente, se extrae la ubicación temporal de

cada uno de los picos que no fueron descartados, los cuales se consideran como eventos detectados (Eyben *et al.*, 2010).

Las técnicas de extracción de curvas de novedad o funciones de detección, pueden clasificarse en técnicas basadas en características temporales y espectrales. Los métodos basados en características temporales hacen uso de cambios abruptos en la amplitud o energía de la señal para detectar eventos. Por otro lado, los métodos basados en características espectrales utilizan una representación tiempo-frecuencia de la señal, basada en la transformada de tiempo corto de Fourier, conocida como espectrograma, la cual permite visualizar la forma como evoluciona en el tiempo tanto la magnitud como la fase de cada banda de frecuencia (Bello *et al.*, 2005). Algunas técnicas propuestas recientemente utilizan redes neuronales para clasificar entre "evento" y "no evento" cada espectro de tiempo corto contenido en el espectrograma, para así generar la curva de novedad (Böck *et al.*, 2012; Eyben *et al.*, 2010). Una de las dificultades en emplear las redes neuronales para esta tarea consiste en la correcta selección del modelo, que incluye determinar el número de neuronas, el número de capas ocultas de la red, y la rata de aprendizaje en el algoritmo de entrenamiento de la red neuronal, entre otros. Este paso de selección del modelo es considerado como una de las mayores desventajas para el uso de la red neuronal en la práctica.

En presencia de un número suficiente de neuronas, es posible mostrar que una red neuronal converge a lo que se conoce como un *Proceso Gaussiano*. La ventaja práctica del proceso Gaussiano es que el problema de selección del modelo es menor puesto que el número de parámetros a sintonizar se reduce considerablemente. Inclusive, en un enfoque Bayesiano completo del proceso Gaussiano, todos los parámetros se marginalizan, por lo cual el problema de selección del modelo desaparece.

La finalidad de este artículo es proponer una metodología para la detección de eventos sonoros en señales de música, en la que la extracción de la curva de novedad se realice por medio de un clasificador binario basado en procesos Gaussianos.

2. MÉTODOS Y MATERIALES

En esta sección se definen conceptos básicos de teoría de señales, en la subsección *Sonido, forma de onda y eventos sonoros* se define el concepto de forma de onda del sonido o señal de audio, y se define qué es un evento sonoro además de sus propiedades. En la subsección *Curva de novedad* se introduce el concepto de curva de novedad y se presentan algunos ejemplos. Posteriormente se abordan algunas técnicas ampliamente usadas para detectar eventos en señales de música, se describen las bases de datos utilizadas y las medidas empleadas para evaluar el desempeño de los algoritmos.

2.1 Sonido, Forma de Onda y Eventos Sonoros

Desde un punto de vista físico, un sonido o una señal de audio es generada por algún objeto vibrante, como las cuerdas vocales de un cantante o las cuerdas vibrantes de un violín junto con su caja de resonancia. Estas vibraciones causan desplazamientos y oscilaciones de las partículas en el aire, lo que a su vez causa regiones de compresión y expansión. Dicha presión alternante viaja como una onda desde la fuente hasta el oyente o micrófono a través del aire, la cual puede ser percibida como un sonido por el oído humano, o ser convertida en una señal eléctrica por un micrófono. El cambio en la presión del aire en cierto punto del espacio puede ser representado como una gráfica *presión-tiempo*, también llamada la *forma de onda* del sonido, la cual muestra la desviación de la presión del aire con respecto a su presión promedio usualmente medida en Pascales (Müller, 2007).

Cuando se toca una sola nota en un instrumento musical el sonido resultante está lejos de ser un simple tono puro con una frecuencia bien definida. Intuitivamente, un tono musical se considera como la superposición de varios tonos puros, llamados *armónicos*, cuya frecuencia difiere en múltiplos enteros de determinada frecuencia fundamental. Adicionalmente, un tono musical contiene ruido y componentes transitorios que típicamente aparecen durante el *Ataque* de la mayoría de instrumentos, por ejemplo cuando se golpea una cuerda de guitarra (Fig. 1).

La Fig. 2 ilustra el *Inicio, Ataque* y *Transitorio* correspondiente al evento sonoro mostrado en la Fig. 1a. El *Ataque* de un evento es el intervalo de tiempo durante el cual aumenta la envolvente de la amplitud. El concepto *Transitorio* (Bello *et al.*, 2005) se refiere a cortos intervalos de tiempo durante los cuales la señal evoluciona rápidamente de manera no trivial o relativamente impredecible. En instrumentos acústicos, el transitorio a menudo corresponde al tiempo donde la excitación se aplica y después se amortigua; dejando solo la lenta decadencia de las frecuencias de resonancia del cuerpo. Finalmente, el *Inicio* (*Onset)* de un evento, es un solo instante elegido para marcar dónde comienza el transitorio (Bello *et al.*, 2005).

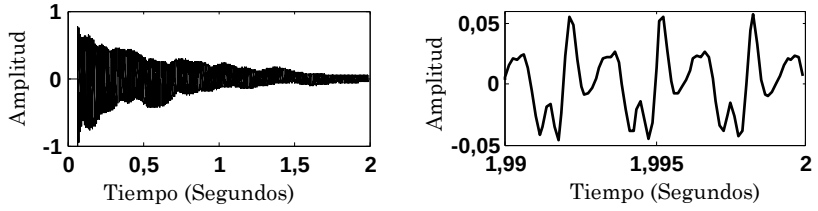

Fig. 1. a) Forma de onda producida al tocar una cuerda de guitarra, b) segmento ampliado de la forma de onda de la Fig. 1a. Fuente: Autores

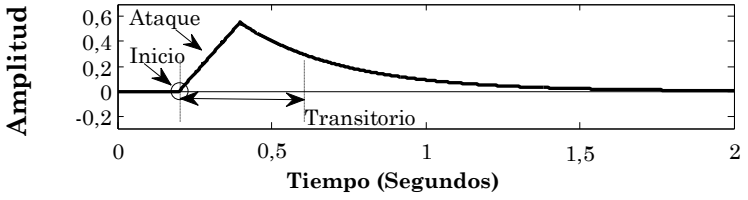

Fig. 2. Definición de Ataque, Inicio y Transitorio. Fuente: Bello *et al.*, 2005

2.2 Curva de Novedad

En el caso de señales de música es inapropiado buscar eventos en la señal original en el dominio del tiempo, debido a la complejidad de las señales. Por lo tanto, se recurre a una representación

intermedia que refleje de forma simplificada la estructura local de la señal original. Dicha representación intermedia se conoce como *curva de novedad, función de detección,* o *función de novedad* (Bello *et al.*, 2005). El enfoque general de las técnicas de extracción de curvas de novedad es capturar cambios repentinos en alguna de las propiedades de la señal, los cuales son provocados generalmente por el inicio de un nuevo evento. La curva que se obtiene está compuesta por picos que indican la ubicación de posibles eventos (Müller *et al.*, 2011).

La Fig. 3a muestra la forma de onda obtenida al tocar una batería (instrumento de percusión) en la que fácilmente pueden identificarse los instantes de tiempo en los que han ocurrido eventos debido a los cambios abruptos en la amplitud. Las líneas continuas verticales en la parte inferior indican las posiciones donde ocurren los eventos. La Fig. 3b muestra la curva de novedad correspondiente.

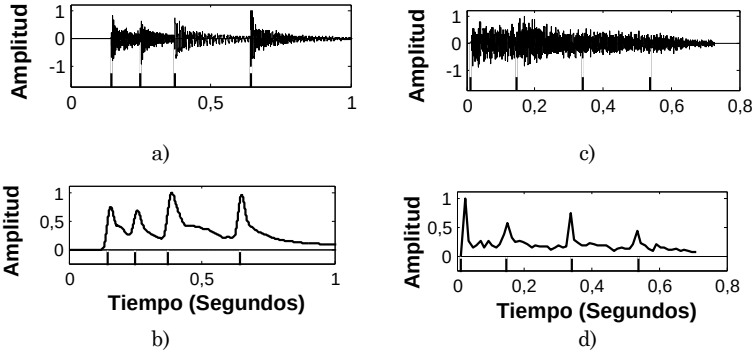

Fig. 3. a) Forma de onda de una batería (instrumento de percusión) y líneas verticales que indican las posiciones de los eventos, b) Curva de novedad correspondiente a la forma de onda de la Fig. 3a, c) Forma de onda de un piano al tocarse cuatro notas de manera sucesiva y líneas verticales que indican en que instantes fueron tocadas, d) Curva de novedad correspondiente a la forma de onda de la Fig. 3c.
Fuente: Autores

En otros casos no es posible detectar cuando han ocurrido eventos analizando la señal original en el dominio del tiempo. Como ejemplo la Fig. 3c muestra la forma de onda producida por un piano al tocarse cuatro notas de manera sucesiva; en este caso

no es evidente en qué momentos han sido tocadas las cuatro notas. Las líneas continuas verticales en la parte inferior indican en que instantes fueron tocadas. La Fig. 3d muestra la curva de novedad correspondiente.

2.3 Técnicas de Detección de Eventos Basadas en Características Temporales de la Señal

Usualmente, la ocurrencia de un evento produce un aumento en la amplitud y energía de la señal. Esto ocurre en señales sencillas producidas usualmente por instrumentos de percusión donde los sonidos ocurren sobre un fondo silencioso.

La función de detección $LE(n)$ (1) consiste en promediar por segmentos la energía de la señal original. Este promedio es afectado por la ventana $w(m)$ la cual asigna un peso a cada una de las muestras correspondientes a determinado segmento. La función de detección $LE(n)$ se calcula como:

$$LE(n) = \frac{1}{N} \sum_{m=-\frac{N}{2}}^{\frac{N}{2}-1} [x(n+m)]^2 w(m), \tag{1}$$

Donde $w(m)$ es una ventana de tamaño N, centrada en $m = 0$, y $x(n)$ es la señal original. Algunos algoritmos estándar de detección de eventos trabajan con la Primera Diferencia de la Energía (2), de manera que incrementos repentinos en la energía se transforman en picos angostos.

$$DEL(n) = LE(n) - LE(n-1) \tag{2}$$

Adicionalmente, se puede aplicar una rectificación de media onda (3) para tener en cuenta solo aquellos cambios en los que hay un incremento en la energía, es decir enfatizar el inicio (*Onset*) de un evento en lugar de su final (*Offset*) (Bello *et al.*, 2005).

$$H(n) = \frac{DEL(n) + |DEL(n)|}{2} \tag{3}$$

El tipo y tamaño de la ventana $w(m)$ afecta considerablemente esta clase de funciones de detección. La Fig. 4b y la Fig. 4c ilustran diferentes curvas de novedad de una misma señal compuesta por un solo evento Fig. 4a.

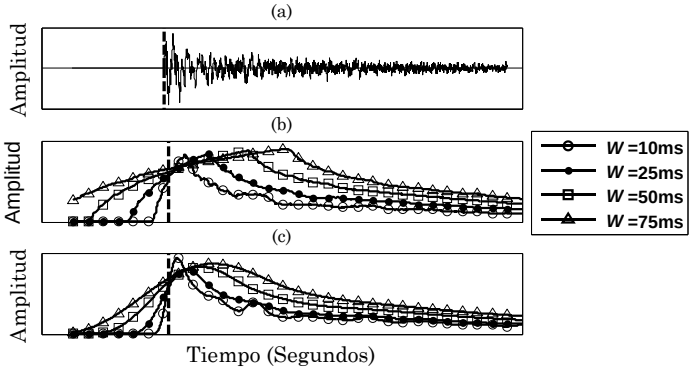

Fig. 4. a) Señal compuesta por un solo evento, b) Curvas de novedad extraídas de la señal de la Fig. 4a, utilizando una ventana rectangular de diferente tamaño W, c) Curvas de novedad extraídas de la señal de la Fig. 4a utilizando una ventana tipo Hann de diferente tamaño W. Fuente: Autores

En la Fig. 4b se muestran curvas de novedad extraídas utilizando una ventana tipo rectangular, es decir $w(m) = 1$, utilizando tamaños de ventana de $10\,ms$, $25\,ms$, $50\,ms$ y $75\,ms$. Estas curvas tienen un alto nivel de ruido, y la ubicación del valor máximo (pico) de cada curva se aleja rápidamente de la ubicación real del evento (línea negra discontinua vertical) a medida que aumenta el tamaño de la ventana.

En la Fig. 4c se muestran curvas de novedad extraídas utilizando una ventana $w(m)$ tipo Hann (4) definida como,

$$w(n) = \frac{1}{2}\left[1 - \cos\left(2\pi\frac{n}{N}\right)\right], \qquad 0 \le n \le N \tag{4}$$

donde el tamaño de la ventana $w(n)$ es $L = N + 1$.(Oppenheim y Schafer, 1989). La Fig. 5 muestra un ejemplo de este tipo de ventana.

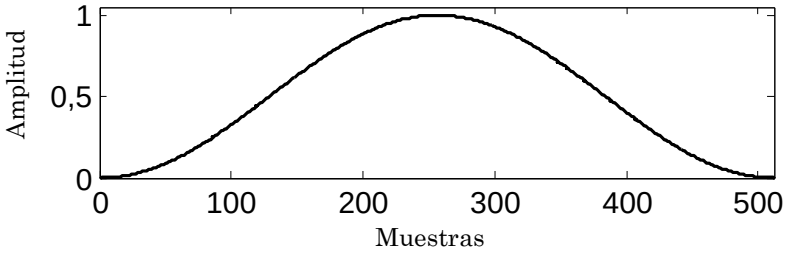

Fig. 5. Ejemplo ventana Hann. Fuente: Autores

La Fig. 4c muestra los resultados de aplicar una ventana Hann para tamaños de $10\,ms$, $25m\,s$, $50m\,s$ y $75\,ms$. Este tipo de curvas presenta menos ruido y los máximos (picos) de estas curvas están más cerca de la ubicación real del evento.

2.4 Técnicas de Detección de Eventos Basadas en Características Espectrales de la Señal

Estas técnicas utilizan una representación tiempo-frecuencia de la señal basada en la *Transformada de Tiempo Corto de Fourier* (5) (Benesty *et al.*, 2008). En este caso no se transforma la señal completa: la señal se divide en segmentos que pueden traslaparse o no, y se transforman estos segmentos, los cuales son afectados por una ventana en este caso tipo Hann. La representación que se obtiene se conoce como espectrograma:

$$X(n,k) = \sum_{m=-\frac{N}{2}}^{\frac{N}{2}-1} x(hn+m)w(m)e^{\frac{-2j\pi mk}{N}}, \tag{5}$$

donde N es el tamaño de la ventana $w(m)$, $x(n)$ es la señal a transformar y $X(n,k)$ corresponde al k-ésimo coeficiente de Fourier del n-ésimo segmento.

En el dominio espectral, incrementos de energía ligados a eventos tienden a aparecer como un fenómeno de banda ancha. Así mismo, la energía de la señal se concentra usualmente en bajas frecuencias. Los cambios debidos a eventos son más evidentes en

altas frecuencias. Para enfatizar esto, el espectro puede ser ponderado preferencialmente hacia altas frecuencias para así obtener una medida ponderada de la energía. La ecuación (6) define la función de detección *Contenido en Alta Frecuencia*:

$$\tilde{E}(n) = \frac{1}{N} \sum_{k=-\frac{N}{2}}^{\frac{N}{2}-1} W_k |X_k(n)|^2, \tag{6}$$

donde la ponderación dependiente de la frecuencia W_k es igual a $|k|$. $X_k(n)$ es el k-ésimo coeficiente de Fourier perteneciente al n-ésimo segmento.

Un enfoque más general basado en cambios en el espectro consiste en definir la función de detección como la distancia entre espectros de ventanas sucesivas, tratándolos como puntos en un espacio N-dimensional. Dependiendo de la métrica utilizada para medir estas distancias se pueden construir diferentes funciones de detección basados en la *diferencia espectral*. La siguiente función de detección, conocida como *Diferencia Espectral Norma-$\boldsymbol{L_1}$* (7) mide el cambio en la magnitud de cada coeficiente, y suma todas estas diferencias (Dixon, 2006):

$$SD_{L1}(n) = \sum_{k=-\frac{N}{2}}^{\frac{N}{2}-1} H(|X(n,k)| - |X(n-1,k)|), \qquad H(x) = \frac{\boldsymbol{x} + |\boldsymbol{x}|}{\boldsymbol{2}}, \tag{7}$$

Con esta rectificación solo se tienen en cuenta aquellas frecuencias donde hay un aumento en la energía, por lo tanto enfatiza el inicio del evento (*Onset*) más que el final.

2.5 Técnicas de Detección de Eventos Basadas en Redes Neuronales y Características Espectrales de la Señal

En este tipo de técnicas, primero se transforma las señales de audio de entrada al dominio de la frecuencia por medio de una o varias transformadas de tiempo corto de Fourier paralelas con

diferentes tamaños de ventanas. Posteriormente, se reduce la dimensión del espectro por medio de una conversión a la escala de frecuencia Mel utilizando un banco de filtros triangulares. Finalmente, el espectro reducido, así como su correspondiente diferencia de primer orden, son utilizados como entrada a la red neuronal, lo cual produce como salida una curva de novedad. La estructura más utilizada corresponde a una red neuronal recurrente. Este tipo de redes poseen conexiones cíclicas hacia atrás. Esto genera un tipo de memoria del contexto pasado, permitiendo así a los valores de entrada permanecer en las capas ocultas, e influenciar la salida futura de la red (Eyben *et al.*, 2010). La Fig. 6 muestra la estructura general de este tipo de técnicas.

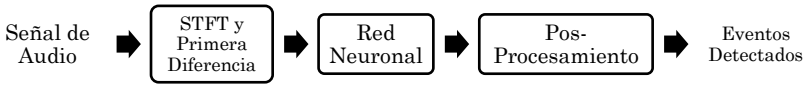

Fig. 6. Estructura algoritmos basados en redes neuronales.
Fuente: Eyben *et al.*, 2010

2.6 Selección de Picos

Después de haber sido extraída la curva de novedad, la siguiente etapa consiste en decidir cuáles picos (máximos locales) son clasificados como eventos y cuáles no. Generalmente estos picos están sujetos a algún nivel de variabilidad en forma y tamaño, también pueden estar afectados por ruido. Este proceso se divide en tres pasos: pos-procesamiento, umbralización, y proceso final de decisión (Bello *et al.*, 2005; Eyben *et al.*, 2010).

En la etapa de *Pos-procesamiento* se sustrae la media local de la curva de novedad (8). La media local es extraída utilizando una ventana de aproximadamente 100 ms.

$$\overline{df}(n) = df(n) - mean(\, df(n-M), \ldots, df(n+M)\,) \tag{8}$$

donde $\overline{df}(n)$ es la curva de novedad después de haber sido extraída la media local, $df(n)$ la curva de novedad inicial y $2M$ el ancho de la ventana utilizada para calcular la media. Posterior-

mente se normaliza la curva de novedad restando la media global y dividiendo por la desviación máxima absoluta. Por último, se aplica un suavizador (9) para reducir posible ruido presente en la curva de novedad. Dicho suavizado se obtiene utilizando una ventana $w(m)$ tipo Hann de 50 ms. La curva de novedad suavizada $df_f(n)$ se define como:

$$df_f(n) = \frac{1}{N} \sum_{m=-\frac{N}{2}}^{\frac{N}{2}-1} \overline{df}(n+m)w(m), \tag{9}$$

En la etapa de *Umbralización se* define un umbral (10) que separe efectivamente picos correspondientes a eventos y picos que no están relacionados con eventos, esto se debe a que casi siempre la curva de novedad tendrá un numero de picos que no están relacionados con eventos, incluso después de pasar por la etapa de pos procesamiento. Para tal fin se define un umbral adaptativo $\tilde{\delta}(n)$ utilizando una ventana de 25 *ms*:

$$\tilde{\delta}(n) = \delta + median(|df(n-M)|, \dots, |df(n+M)|), \tag{10}$$

donde $df(n)$ es la curva de novedad, $2M$ es el tamaño de la ventana y δ es una constante positiva. La curva de novedad final (11) contiene solo los valores superiores a este umbral adaptativo.

$$df(n) = \begin{cases} df(n), para\ df(n) > \tilde{\delta}(n) \\ 0, \quad en\ otro\ caso \end{cases} \tag{11}$$

Por último, el *Proceso final de decisión* después del pos procesar y umbralizar la curva de novedad consiste en identificar máximos locales. Por lo tanto se realiza una búsqueda de picos utilizando (12), donde los valores de $O(n)$ diferentes a cero indican la ubicación de los eventos detectados.

$$O(n) = \begin{cases} df(n) \quad para\ \ df(n-1) \leq df(n) \geq df(n+1) \\ 0 \quad en\ otro\ caso \end{cases}, \tag{12}$$

2.7 Bases de Datos

En esta investigación se utilizaron dos bases de datos. Una de ellas ha sido ampliamente usada en diferentes investigaciones (Collins, 2005; Dixon, 2006; Eyben *et al.*, 2010; Degara *et al.*, 2011). Esta base de datos fue facilitada por Juan Pablo Bello (Bello *et al.*, 2005), y está compuesta por 23 registros de audio comercial y no comercial, abarcando varios estilos musicales e instrumentos. Todas las señales tienen las siguientes características: un solo canal (mono), frecuencia de muestreo 44,1 kHz, 16 bits, y duración de los registros entre 2 y 60 segundos. Los registros están divididos en cuatro grupos de acuerdo a las características de los eventos: *Pitched Non-Percussive* contiene eventos con tono producidos por instrumentos de cuerda frotada (ejemplo: violín), *Pitched Percussive* corresponde a eventos con tono producidos por un instrumento de percusión (ejemplo: piano), *Non-Pitched Percussive* contiene eventos sin tono producidos por instrumentos de percusión (ejemplo: bombo, caja, platillos), por ultimo *Complex Mix* contiene mezclas de varios instrumentos (ejemplo: música Pop, Rock), ver Tabla 1.

Tabla 1. Descripción base de datos uno. Fuente: Autores

	Pitched Non-Percussive	Pitched Percussive	Non-Pitched Percussive	Complex Mix	Total:
Número de eventos	93	489	212	271	1065
Registros por categoría	1	9	6	7	23

Así mismo, se implementó una segunda base de datos (Tabla 2) con 23 registros de audio con las siguientes características: un solo canal (mono), frecuencia de muestreo 44,1 kHz, 16 bits, y duración de los registros entre 2 y 60 segundos. Los registros están agrupados en tres categorías de acuerdo a las características de los eventos: *Instrumentos de Percusión (Sintético)* contiene señales producidas por instrumentos de percusión (bombo, platillos). Este tipo de señales fueron producidas de forma sintética utilizando CUBASE® y EZDRUMMER®. La categoría *Instrumentos de Percusión (Reales)* contiene eventos sin tono producidos por instru-

mentos de percusión. Este tipo de señales corresponde a segmentos de grabaciones comerciales. Los segmentos se extrajeron utilizando ADOBE AUDITION®. Finalmente, la categoría *Instrumentos con Tono (Real)* contiene eventos con tono producidos por distintos instrumentos (bajo, chelo, violín, guitarra acústica). Estas señales corresponden a segmentos de grabaciones comerciales y no comerciales, además de archivos seleccionados de la base de datos *Open Path Music* (http://wiki.laptop.org/go/Sound_samples). Las anotaciones, es decir, la ubicación de los eventos presentes en cada registro, se realizaron de forma manual utilizando SONIC VISUALISER®.

Tabla 2. Descripción base de datos dos. Fuente: Autores

	Instrumentos de percusión (sintético)	Instrumentos de percusión (real)	Instrumentos con tono (real)	Total
Número de eventos	263	190	316	769
Registros por categoría	6	6	11	23

2.8 Medidas de Desempeño Utilizadas

Según el procedimiento de evaluación establecido por MIREX (*The Music Information Retrieval Evaluation eXchange*) (www.music-ir.org; 04/03/2013) para algoritmos de detección de eventos, la medida F (13) es el criterio principal para medir el desempeño,

$$F = \frac{2PR}{P+R}, \tag{13}$$

donde P (*Precision)* (14) y R (*Recall)* (15) están definidos por el número de eventos correctamente detectados N_{tp}, el número de falsas alarmas N_{fp}, y el número de eventos perdidos N_{fn}.

$$P = \frac{N_{tp}}{N_{tp}+N_{fp}}, \tag{14}$$

$$R = \frac{N_{tp}}{N_{tp} + N_{fn}}. \tag{15}$$

Las detecciones son contadas como correctas cuando están dentro de una ventana $t_{tol} = \pm 50\, ms$ alrededor del evento anotado. Si hay más de un evento en esta ventana de tolerancia, solo uno es contado como verdadero positivo, los otros son contados como falsos positivos.

2.9 Curva P/R

En orden de obtener una descripción más detallada del desempeño de un algoritmo, se varía en pequeños pasos el umbral δ utilizado en la etapa de selección de picos. De esta manera, se obtienen los valores P y R para diferentes valores de umbral, y se crean curvas P/R ubicando los valores P en la abscisa y los valores R en la ordenada. En los diagramas P/R el mejor detector de eventos, en términos de *F-measure*, es el algoritmo cuya curva P/R este más cerca de la esquina superior derecha del diagrama (Holzapfel y Stylianou, 2010).

3. METODOLOGÍA

La metodología propuesta está basada en la estructura general de los algoritmos basados en redes neuronales, con la diferencia de que en la etapa de clasificación se reemplaza la red neuronal por un clasificador basado en procesos Gaussianos. El método propuesto para detectar eventos sonoros está compuesto por tres etapas principales, como se muestra en la Fig. 7, *Pre-Procesamiento*, *Clasificación basada en procesos Gaussianos*, y *Selección de Picos*. El algoritmo tiene como entrada una señal de audio que es transformada al dominio de la frecuencia utilizando la transformada de tiempo corto de Fourier (STFT), la magnitud del espectro obtenido además de su diferencia de primer orden son utilizadas como entrada un clasificador basado en procesos Gaussianos, cuya salida es una curva de novedad. Finalmente, se utiliza una etapa de selección de picos (ver subsección *Selección de*

Picos) a la salida del clasificador, para localizar los instantes en que inician los eventos.

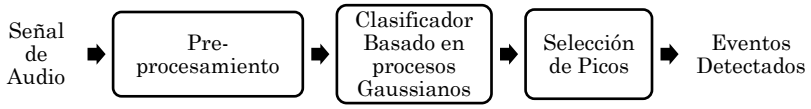

Fig. 7. Diagrama metodología propuesta. Fuente: Autores

En la etapa de *Pre-procesamiento* las señales de audio de entrada son primero normalizadas sustrayendo la media y dividiendo por la desviación máxima absoluta. Posteriormente son divididas en segmentos de 1024 muestras (23,2 ms) con un traslape del 50 %. Estos segmentos son transformados al dominio de la frecuencia utilizando la transformada de tiempo corto de Fourier y una ventana Hann, obteniendo así el espectrograma $X(k,n)$, donde n es el índice del segmento transformado, y k el índice de la banda de frecuencia. Del resultado anterior solo se utiliza la magnitud del espectrograma, es decir $|X(k,n)|$. La dimensión de la magnitud del espectrograma se reduce aplicando una conversión a la escala de frecuencia Mel utilizando un banco de 20 filtros triangulares equidistantes en esta escala, obteniendo así una representación compacta o espectrograma-Mel $M(m,n)$, donde $m = [1,2,\dots,M]$ siendo M el número de filtros utilizados, de decir $M = 20$. Posteriormente se pasa a una representación logarítmica aplicando (16).

$$M_{log}(m,n) = \log(M(m,n) + 1) \tag{16}$$

Por último se calcula la diferencia de primer orden (17) de $M_{log}(m,n)$, aplicando una rectificación de media onda $H(x) = \frac{x+|x|}{2}$:

$$D^{+}(m,n) = H(\, M_{log}(m,n) - M_{log}(m,n-1)\,) \tag{17}$$

3.1 Clasificación Binaria Basada en Procesos Gaussianos

Un proceso Gaussiano es una colección de variables aleatorias, donde cualquier número finito de estas variables tiene una distribución conjunta normal. Un proceso Gaussiano real $f(\mathrm{x})$ está

totalmente especificado por su función de media $m(\mathrm{x})$ (18) y su función de covarianza $k(\mathrm{x}, \acute{\mathrm{x}})$,

$$\begin{aligned} m(\mathrm{x}) &= \mathbb{E}[f(\mathrm{x})] \\ k(\mathrm{x}, \acute{\mathrm{x}}) &= \mathbb{E}[(f(\mathrm{x}) - m(\mathrm{x}))(f(\acute{\mathrm{x}}) - m(\acute{\mathrm{x}}))], \end{aligned} \tag{18}$$

Comúnmente los procesos Gaussianos están definidos sobre el tiempo, es decir donde el dominio de entrada de las variables aleatorias es el tiempo. En este caso el dominio de entrada $\mathcal{X}$ es el conjunto de posibles entradas, en general $\mathbb{R}^D$ (Rasmussen y Williams, 2006), donde $D = 40$, puesto que el vector de entrada al clasificador está compuesto por 20 coeficientes del espectrograma en escala Mel, además de otros 20 valores definidos por su correspondiente diferencia de primer orden.

Los procesos Gaussianos se usan como prior para describir la incertidumbre en funciones. Es lo que se conoce comúnmente como una función de probabilidad sobre funciones. Cuando además del proceso Gaussiano se define una función de verosimilitud, es posible emplear el teorema de Bayes para describir la función de probabilidad posterior sobre las funciones que se analizan. En un problema de clasificación biclase, la función de verosimilitud se suele asumir como una función logística o una función probit, en cuyo caso el cálculo del posterior o de la función predictiva se debe realizar a través de un procedimiento de inferencia Bayesiana aproximado. Entre las opciones de aproximación se encuentran la aproximación de Laplace y el algoritmo de Propagación de la Esperanza (detalles de ambas metodologías se pueden encontrar en Rasmussen y Williams, 2006, capítulo 3). La aproximación que se emplea en este artículo es la aproximación de Laplace.

Para clasificación binaria se utiliza una etiqueta discreta $c \in \{1,0\}$ por convención. Sea $p(c|x)$ la probabilidad posterior de la clase discreta c dado la entrada x, y sea y un espacio continuo latente definido por un proceso Gaussiano, dicho espacio será después mapeado a una probabilidad de clase usando (19).

$$p(c|x) = \int p(c|y)p(y|x)dy, \tag{19}$$

donde $p(y|x)$ está definida por una distribución normal de la siguiente manera:

$$p(y|x) = N(\, y|m(x), k(x,x')\,) \tag{20}$$

y $p(c|y)$ representa la probabilidad de la clase c dada la activación latente y, definida como,

$$p(c|y) = \sigma\big((2c-1)y\big), \tag{21}$$

donde $\sigma(x)$ es la función de transferencia logística definida como,

$$\sigma(x) = \frac{1}{1+e^{-x}}. \tag{22}$$

La distribución de probabilidad $p(c|y)$ (21) es válida ya que

$$p(c=1|y) = \sigma(y), \tag{23}$$

mientras que

$$p(c=0|y) = \sigma(-y) = 1-\sigma(y) \tag{24}$$

Adicionalmente, cuando se tienen algunos datos de entrada de entrenamiento $X = \{x^1, \dots, x^N\}$ con sus respectivas etiquetas $C = \{c^1, \dots, c^N\}$, la probabilidad de la clase discreta c^* para una nueva entrada x^* está definida como,

$$p(c^*|x^*, C, X) = \int p(c^*|y^*)p(y^*|X, C)dy^*, \tag{25}$$

donde

$$p(y^*|X, C) \propto \int p(y^*, Y, C|X, x^*)dY = \int p(C|Y)p(y^*, Y|X, x^*)dY, \tag{26}$$

estableciendo $Y = \{y^1, \dots, y^N\}$. El problema se puede reformular de la siguiente manera:

$$p(y^*, Y, C|X, x^*) \propto p(y^*, Y|X, x^*, C) \propto p(y^*|Y, x^*, X)p(Y|C, X) \quad (27)$$

Una dificultad que se presenta, es que el término de mapeo de clases no lineal hace que calcular la distribución posterior sea difícil (21), ya que las integrales sobre $y^1, \ldots, y^N$ no pueden ser realizadas de forma analítica. Por lo tanto se aplica alguna técnica de aproximación, en este caso se utilizó el método de Laplace (Rasmussen y Williams, 2006), el cual aproxima una distribución no-Gaussiana por medio de una distribución Gaussiana $q(Y|C, X)$.

$$p(Y|C, X) \approx q(Y|C, X) = N\left(Y\middle|\hat{Y}, A^{-1}\right) \quad (28)$$

donde $\hat{Y} = \arg\max_Y p(Y|C, X)$ y $A = -\nabla\nabla \log p(Y|C, X)|_{Y=\hat{Y}}$. Finalmente, de (26) y (28) se pueden hacer predicciones aproximadas de acuerdo a (29) (Barber, 2012).

$$p(y^*, Y|X, x^*, C) \approx p(y^*|Y, x^*, X)q(Y|C, X) \quad (29)$$

4. RESULTADOS Y DISCUSIÓN

En esta sección se presentan los resultados obtenidos sobre cada una de las bases de datos. Las abreviaciones utilizadas para referirse a los algoritmos implementados corresponden a: *GP* = Procesos Gaussianos, *NN* = Red Neuronal, *SDL1* = Diferencia Espectral Norma-L1, *DEL* = Primera Diferencia de la Energía Local, *HFC* = Contenido en Alta Frecuencia. Todos los algoritmos se realizaron en MATLAB®.

La red neuronal recurrente implementada está compuesta por tres capas ocultas con 20 unidades en cada una, y una sola neurona en la capa de salida. Las capas ocultas, así como la capa de salida, utilizan la función de activación *tangente hiperbólica*. La red fue entrenada utilizando el Toolbox de redes neuronales de MATLAB®. Uno de los problemas que surge cuando se entrena la red neuronal es el *sobre entrenamiento*, esto ocurre cuando la red memoriza los datos con los que fue entrenada pero no aprende a generalizar para nuevos datos. Para prevenir este problema se utilizó *Interrupción temprana*. En esta técnica, el entrenamiento

es interrumpido cuando el error sobre un subconjunto de los datos de entrenamiento incrementa durante un determinado número de iteraciones. La estructura de la red y la forma como fue entrenada van de acuerdo a lo propuesto por Böck *et al.* (2012), Eyben *et al.*, (2010).

Para la implementación del clasificador basado en procesos Gaussianos se utilizó el Toolbox GPML (Gaussian Processes for Machine Learning) para MATLAB versión 3.2 (http://www.gaussianprocess.org/gpml/code/matlab/doc/index.html; 04/03/2013). Se utilizó la aproximación de Laplace como método de inferencia tanto para estimar los hiperparámetros que definen la función de media y la función de covarianza del proceso Gaussiano como para realizar predicción. La función de verosimilitud utilizada fue la función Probit. El clasificador fue entrenado con el 70 % de la base de datos uno, el 30 % restante de utilizó para test.

4.1 Resultados sobre la Base de Datos Uno

En la Tabla 3 se presenta el desempeño más alto, expresado en términos de la medida F, alcanzado por cada uno de los algoritmos implementados, evaluados sobre el conjunto de validación (30 % de la base de datos uno). También se incluye los correspondientes valores de *Precision* (P) y *Recall* (R) con los cuales se obtuvo este resultado. Puede observarse que la metodología propuesta presenta un desempeño 1,66 % superior en la medida F a los demás algoritmos, así mismo presenta el valor más alto en *Precision* (93,27 %). Esto indica que el detector de eventos basado en procesos Gaussianos produce una menor cantidad de falsos positivos. Por otro lado, el valor alcanzado de *Recall* (88,18 %) es igual al obtenido con la red neuronal recurrente. Esto muestra que ambos algoritmos producen la misma cantidad de detecciones correctas.

Como comparación gráfica, la Fig. 8 muestra la *Curva P/R* de cada uno de los algoritmos. Puede observarse que en general la metodología propuesta (GP) presenta el mejor desempeño, alcanzando los valores más altos en Precision y Recall, seguida por la red neuronal recurrente (NN). Adicionalmente, se observa una gran diferencia entre los algoritmos basados en aprendizaje supervisado (NN y GP) y los algoritmos basados en características de la

señal (HFC, SDL1 y DLE), presentando estos últimos el desempeño más bajo.

Tabla 3. Resultados sobre la base de datos uno. Fuente: Autores

Función de detección	F (%)	P (%)	R (%)
GP	90,65	93,27	88,18
NN	88,99	89,81	88,18
SDL1	77,04	84,32	70,91
DLE	74,36	75,59	73,18
HFC	71,13	72,30	70,00

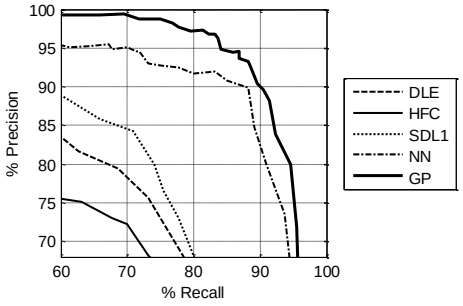

Fig. 8. Curvas *P/R* para base de datos uno. Fuente: Autores

En la Tabla 4 se muestra el desempeño de cada uno de los algoritmos para cada uno de los tipos de eventos en los que está dividida la base de datos uno. En este caso el subconjunto *Pitched Non-Percussive* no es analizado debido a que esta categoría solo tiene asociado un registro, el cual estaba incluido dentro del conjunto de datos de entrenamiento.

Se observa que el desempeño de los algoritmos *Contenido en Alta Frecuencia* (HFC) y *Diferencia de la Energía Local* (DLE) es altamente dependiente de la naturaleza de los eventos analizados. HFC presenta un buen desempeño en eventos producidos por instrumentos de percusión sin tono (*Non-Pitched Percussive)* y para señales compuestas por mezclas de varios instrumentos dentro de los cuales hay instrumentos de percusión (*Complex Mix*), puesto que en este tipo de señales los cambios de energía produci-

dos por eventos son muy evidentes en altas frecuencias, lo cual va de acuerdo con (Bello *et al.*, 2005). Por otro lado DLE muestra un buen desempeño para señales de un solo instrumento (*Pitched Percussive* y *Non-Pitched Percussive)*, sin embargo DLE presenta una mayor cantidad de falsos positivos cuando las señales están compuestas por una mezcla de varios instrumentos, esto se ve reflejado en el bajo valor de *Precision* (63,54 %). SDL1 presenta un rendimiento similar para todos los tipos de eventos, produciendo baja cantidad de falsos positivos, dado alto valor en *Precision* para cada categoría. Sin embargo la cantidad de detecciones correctas disminuye para señales compuestas por una mezcla de varios instrumentos, esto se ve reflejado en el bajo valor de *Recall* (77,33 %).

Tabla 4. Resultados sobre cada una de las categorías de la base de datos uno. Fuente: Autores

Función de Detección	Pitched Percussive			Non-Pitched Percussive			Complex Mix		
	F (%)	P (%)	R (%)	F (%)	P (%)	R (%)	F (%)	P (%)	R (%)
GP	97,37	100	94,87	96,55	97,22	95,89	87,25	87,84	86,67
NN	93,51	94,74	92,31	97,93	98,61	97,26	88,74	88,16	89,33
SDL1	88,89	96,97	82,05	91,78	91,78	91,78	85,93	96,91	77,33
DLE	86,49	91,43	82,05	85,27	98,21	75,34	71,35	63,54	81,33
HFC	66,67	62,22	71,79	85,94	100	75,34	83,21	91,94	76,00

Los algoritmos basados en aprendizaje supervisado (NN y GP) presentan un desempeño superior en cada una de las categorías de la base de datos uno, dado el alto valor en la medida F. Así mismo, son los algoritmos que producen mayor cantidad de detecciones correctas, puesto que presentan valores altos en *Recall* para cada una de las categorías. La metodología propuesta (GP) presenta el mejor desempeño en la categoría *Pitched Percussive*, alcanzando 97,37 % en la medida F, 100 % en *Precision,* y 94,87 % en *Recall*, lo que indica, para un determinado umbral δ, la inexistencia de falsos positivos con una detección del 94,87 % de los eventos presentes en las señales analizadas. Por otro lado la metodología propuesta es levemente superada por la red neuronal recurrente en las categorías *Non-Pitched Percussive* y *Complex Mix*.

La Fig. 9 muestra un ejemplo de la curva de novedad producida por cada uno de los algoritmos implementados, para la misma señal producida por un piano Fig. 9a, además se muestra el correspondiente espectrograma reducido Fig. 9b. Se observa que la curva de novedad producida por la metodología propuesta Fig. 9c tiene picos prominentes en los instantes donde ocurren eventos, por otro lado presenta una alta cantidad de ruido (picos de menor amplitud que no corresponden a eventos) los cuales ocasionan falsos positivos. La curva de novedad producida por la red neuronal recurrente Fig. 9d presenta una menor cantidad de ruido, sin embargo la amplitud de los picos que corresponden a eventos también disminuye. Las curvas de novedad producidas por SDL1 Fig. 9e, HFC Fig. 9f y DEL Fig. 9g presentan una alta cantidad de ruido, esto dificulta visualizar cuales picos corresponden realmente a eventos.

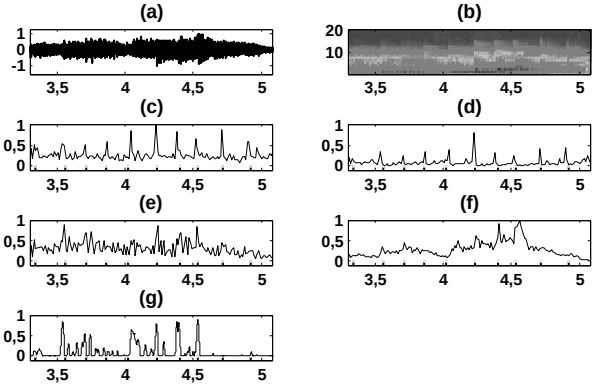

Fig. 9. a) Señal de entrada, b) Espectrograma reducido. Curvas de novedad obtenidas con: c) Procesos Gaussianos GP, d) red neuronal NN, e) Diferencia espectral SDL1, f) Contenido en alta frecuencia HFC, g) Diferencia de la energía local DEL. Fuente: Autores

4.2 Resultados sobre la Base de Datos Dos

De igual forma, los algoritmos basados en aprendizaje supervisado, es decir el detector de eventos basado en redes neuronales (NN) y el detector de eventos basado en procesos Gaussianos (GP),

entrenados con el 70 % de la base de datos uno, y el resto de algoritmos, fueron puestos a prueba sobre toda la base de datos dos. En la Tabla 5 se presenta el desempeño más alto, expresado en términos de la medida F, alcanzado por cada uno de los algoritmos implementados, evaluados sobre toda la base de datos dos. También se incluye los correspondientes valores de *Precision* (P) y *Recall* (R) con los cuales se obtuvo este resultado. Puede observarse que la metodología propuesta presenta un desempeño 0,45 % superior en la medida F a los demás algoritmos, así mismo presenta el valor más alto en *Precision* (85,45 %). Esto indica que el detector de eventos basado en procesos Gaussianos produce una menor cantidad de falsos positivos, similar a los resultados obtenidos sobre la base de datos uno.

Tabla 5. Resultados sobre toda la base de datos 2. Fuente: Autores

Función de detección	F (%)	P (%)	R (%)
GP	88,09	85,45	90,90
SDL1	87,64	82,77	93,11
NN	85,03	85,14	84,92
DLE	77,20	82,64	72,43
HFC	71,91	67,66	76,72

Como comparación gráfica, la Fig. 10 muestra la *Curva P/R* de cada uno de los algoritmos. Puede observarse que, en general, la metodología propuesta (GP) presenta el mejor desempeño, alcanzando los valores más altos en *Precision* y *Recall*, seguida por la red neuronal recurrente (NN). Sin embargo, el algoritmo basado en la diferencia espectral (SDL1) presenta una mejora en la curva P/R para valores en *Recall* superiores al 90 %, superando a la red neuronal. También se puede observar que para valores de *Recall* superiores a 90 % las curvas producidas por GP y SDL1 son muy similares. Esto indica que a partir de determinado umbral $\boldsymbol{\delta}$ estos algoritmos producen la misma cantidad de falsos positivos y detecciones correctas.

En la Tabla 6 se presenta el desempeño más alto, expresado en términos de la medida F, de cada uno de los algoritmos para cada una de las categorías en las que está dividida la base de datos dos.

También se incluye los correspondientes valores de *Precision* (P) y *Recall* (R) con los cuales se obtuvieron estos resultados. Puede observarse que la metodología propuesta supera a los demás algoritmos en las categorías *Instrumentos de Percusión (sintético)* e *Instrumentos de Percusión (real)* alcanzando valores en la medida F de 98,66 % y 98,10 % respectivamente. En la categoría *Instrumentos con Tono (real)* el algoritmo basado en diferencia espectral (SDL1) presenta el rendimiento más alto, alcanzando 78,45 % en la medida F. En general todos los algoritmos presentan un buen desempeño en las primeras dos categorías. Así mismo el desempeño de todos los algoritmos es inferior en la categoría *Instrumentos con Tono (real).*

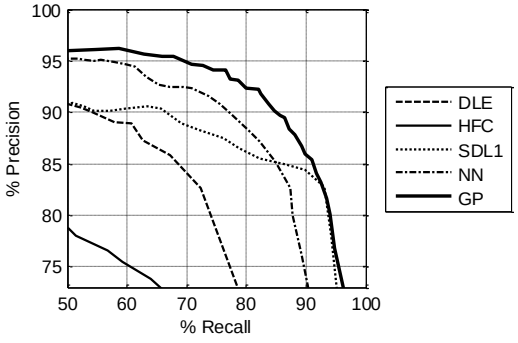

Fig. 10. Curvas P/R para base de datos dos. Fuente: Autores

Tabla 6. Resultados sobre cada una de las categorías de la base de datos dos. Fuente: Autores

Función de Detección	Instrumentos de Percusión (sintético)			Instrumentos de Percusión (real)			Instrumentos con Tono (real)		
	F (%)	P (%)	R (%)	F (%)	P (%)	R (%)	F (%)	P (%)	R (%)
GP	98,66	99,23	98,10	99,21	99,47	98,95	74,10	70,69	77,85
NN	94,30	94,30	94,30	94,21	94,21	94,21	73,33	71,47	75,32
SDL1	98,09	98,47	97,72	97,38	96,88	97,89	78,45	73,95	83,54
DLE	79,08	100	65,40	87,00	87,70	86,32	71,10	68,12	74,37
HFC	93,07	97,11	89,35	87,09	83,90	90,53	49,39	43,06	57,91

La Fig. 11 muestra un ejemplo de la curva de novedad producida por cada uno de los algoritmos implementados, para la misma señal Fig. 11a, además se muestra el correspondiente espectrograma reducido Fig. 11b. Se observa que la curva de novedad producida por la metodología propuesta Fig. 11c tiene picos prominentes en los instantes donde ocurren eventos, por otro lado presenta una alta cantidad de ruido (picos de menor amplitud que no corresponden a eventos) los cuales ocasionan falsos positivos. La curva de novedad producida por la red neuronal recurrente Fig. 11d presenta una menor cantidad de ruido, sin embargo la amplitud de los picos que corresponden a eventos no es muy homogénea. Las curvas de novedad producidas por SDL1 Fig. 11e, HFC Fig. 11f y DEL Fig. 11g presentan un comportamiento similar, debido a la variabilidad de la amplitud de los picos correspondientes a eventos.

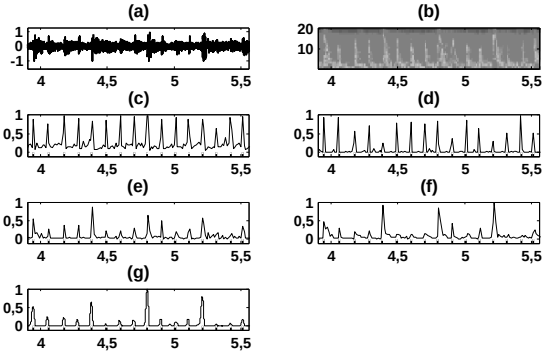

Fig. 11. a) Señal de entrada, b) Espectrograma reducido. Curvas de novedad obtenidas con: c) Procesos Gaussianos GP, d) red neuronal NN, e) Diferencia espectral SDL1, f) Contenido en alta frecuencia HFC, g) Diferencia de la energía local DEL. Fuente: Autores

5. CONCLUSIONES

En este artículo se compararon diferentes técnicas de detección de eventos sonoros puestas a prueba sobre diferentes tipos de señales de música. Los resultados muestran que la metodología propuesta la cual utiliza clasificación basada en procesos Gaussia-

nos presenta en general un mejor desempeño comparado con las técnicas comúnmente utilizadas, alcanzando un valor en la medida *F* de 90,65 % para la base de datos uno y 88,09 % para la base de datos dos.

Según los resultados obtenidos con la medida *Recall* (R) en la base de datos dos se puede afirmar que la metodología propuesta genera una cantidad mayor de detecciones correctas que la metodología basada en redes neuronales. Esto se debe a que las curvas de novedad obtenidas con clasificación basada en procesos gaussianos presentan picos de mayor amplitud, a pesar de que estas curvas presentan mayor ruido que las curvas de novedad obtenidas con redes neuronales (ver Fig. 9 y Fig. 11). Por otro lado, según los resultados obtenidos sobre la base de datos uno, la metodología propuesta se desempeña mejor en señales producidas por un solo instrumento, dado los valores de medida F conseguidos en las categorías *Pitched Percussive* (97,37 %) y *Non-Pitched Percussive* (96,55 %). Por el contrario, cuando la señal analizada es una mezcla de varios instrumentos el desempeño disminuye, esto se ve reflejado en el valor de la medida F obtenido para la categoría *Complex Mix* (87,25 %). Debido a esto, como trabajo futuro se pretende establecer si es posible aumentar el desempeño de la metodología propuesta adicionando una etapa de separación de fuentes previa a la etapa de clasificación, ya que esto permitiría, dada una mezcla de instrumentos, obtener de forma separada la señal producida por cada uno de estos.

6. AGRADECIMIENTOS

Los más sinceros agradecimientos al editor y a los evaluadores por sus comentarios sobre este trabajo, ya que sus sugerencias contribuyeron en gran medida a mejorar la calidad de este artículo.

7. REFERENCIAS

Barber, D. (2012). *Bayesian reasoning and machine learning*. 1ª Edición. Cambridge University Press, New York: USA.

Bello, J.P., Daudet, L., Abdallah, S., Duxbury, C., Davies, M., Sandler, M.B. (2005). A tutorial on onset detection for music signals. *IEEE Transactions on Speech and Audio Processing*, *13*(5), 1035-1047.

Benesty, J., Sondhi, M.M., Huang, Y. (2008). *Springer handbook of speech processing*. 1st Edition. Springer, Berlin: Germany.

Böck, S., Arzt, A., Krebs, F., Schedl, M. (2012). *Online real-time onset detection with recurrent neural networks*. 15th International Conference on Digital Audio Effects, 1-4, York: UK.

Cemgil, A.T., Kappen, B., Desain, P., Honing, H. (2000). On tempo tracking tempogram representation and kalman filtering. *Journal of New Music Research*, *28*(4), 259-273.

Collins, N. (2005). *A comparison of sound onset detection algorithms with emphasis on psychoacoustically motivated detection functions*. Audio Engineering Society Convention, 1-12, Barcelona: Spain.

Degara, N., Argones, E., Pena, A., Torres, S., Davies, M., Plumbley, M. (2012). Reliability-informed beat tracking of musical signals. *IEEE Transactions on Audio Speech and Language Processing, 20*(1), 290-301.

Degara, N., Davies, M., Pena, A., Pumpley, M. (2011). Onset event decoding exploiting the rhythmic structure of polyphonic Music. *IEEE Journal of Selected Topics in Signal Processing, 5*(6), 1228-1239.

Dixon, S. (2006). *Onset detection revisited, 9th Conference on Digital Audio Effects*. Montreal: Canada.

Eyben, F., Böck, S., Schuller, B., Graves, A. (2010). *Universal onset detection with bidirectional long short-term memory neural networks*. 11th International Society for Music Information Retrieval Conference, 589-594, Utrecht: Holland.

Futrelle, J., Downie, S. (2002). *Interdisciplinary communities and research issues in music information retrieval*. 3th International Society for Music Information Retrieval Conference, 1-3, Paris: France.

Holzapfel, A., Stylianou, Y. (2010). Three dimensions of pitched instrument onset detection. *IEEE Transactions on Audio Speech and Language Processing, 18*(6), 1523-1526.

Klapuri, A., Davy, M. (2006). *Signal processing methods for music transcription*. 1st Edition, 101-127. Springer, New York: USA.

Müller, M. (2007). *Information retrieval for music and motion*. 1st Edition. 14-24, Springer, Berlin: Germany.

Müller, M., Ellis, D., Klapuri, A., Richard, G. (2011). Signal processing for music analysis. *IEEE Journal of Selected Topics in Signal Processing, 5*(6), 1088-1108.

Oppenheim, A.V., Schafer, R.W. (1989). *Discrete-time signal processing*. 1st Edition, 447-448. Prentice-Hall, New York: USA.

Rasmussen, C.E., Williams C. (2006). *Gaussian processes for machine learning*. 1st Edition, 7-75. Massachusetts Institute of Technology, London: England.

Robertson, A., Plumbley, M. (2007). *B-keeper: A beat tracker for live performance*. 7th Conference on New Interfaces for Musical Expression, 234-237, New York: USA.

Zhou, R., Mattavelli, M., Zoia, G. (2008). Music onset detection based on resonator time frequency image. *IEEE Transactions on Audio Speech and Language Processing, 16*(8), 1685-1695.