Ingeniería de Sonido

Identificación automática de instrumentos musicales acústicos a partir de señales monofónicas digitales.

Autor: **Andrés Marafioti**
andimarafioti@gmail.com

Tutor: **Dr Juan Ignacio Mieza**
mieza@cnea.gov.ar

15 de marzo de 2016

Resumen

En esta tesis se desarrolló un sistema de identificación automática de instrumentos musicales acústicos mediante técnicas de aprendizaje autónomo. Para ello, se evaluaron bases de datos de instrumentos musicales recomendadas por institutos reconocidos en el campo de la recuperación de información de la música (MIR). A partir de estas, se construyó una base de datos específica para esta investigación. Luego, se caracterizaron las muestras con un grupo de parámetros cuantitativos. Cada parámetro fue seleccionado para describir una instancia específica de la señal sonora. Finalmente, se diseñó el sistema de identificación a partir de los descriptores y se evaluó el rendimiento del sistema con sonidos desconocidos.

Se estudiaron los algoritmos de K-means y Máquinas de Vectores Soporte como opciones para la tarea identificación y se midió su desempeño usando los criterios estándar de precisión, exhaustividad y Medida-F. Se utilizaron bases de datos públicas y gratuitas de muestras sonoras de instrumentos acústicos. Los métodos y algoritmos fueron implementados en Python usando paquetes de software libre y gratuito como Scikit-Learn y Essentia entre otros. El resultado es un algoritmo, que una vez entrenado, puede identificar automáticamente hasta 21 instrumentos acústicos diferentes y permite una evaluación cuantitativa de la información específica que aporta cada descriptor acústico a la identificación.

Palabras clave: *Identificación, MIR, Instrumentos Acústicos, Aprendizaje Autónomo, SVM, Python.*

Abstract

The aim of this study was to develop a software capable of automatically identifying acoustical musical instruments through autonomous learning techniques. In order to achieve this, databases recommended by recognized institutes in the field of Music Information Retrieval (MIR), were evaluated. From these, a database for this specific investigation was created. Later on, the sound samples were characterized using a set of carefully chosen quantitative parameters. Each parameter was selected to describe a specific feature of the sound signal. Finally, the identification system was designed using the parameters as input and the performance of the system was evaluated using unknown sounds.

The K-means and Support Vector Machines algorithms were both evaluated as options for identification techniques. The performance was measured using standard tools as accuracy, recall and F-Score. The public acoustical sound databases used are free and easily obtainable. The system was developed in Python using several open software packages such as Scikit-Learn and Essentia. The final system, after being trained, can automatically identify up to 21 different acoustic instruments and it can allow a quantitative evaluation of the specific information that each acoustic descriptor contributes towards algorithm identification.

Keywords: *Identification, MIR, Acoustical Instruments, Machine Learning, SVM, Python.*

Agradecimientos

Me considero muy afortunado por ser parte de la cátedra de Procesamiento Digital de Señales en la Universidad Nacional de Tres de Febrero. Allí se presentaron diversos campos de investigación y tuve la oportunidad de trabajar en un ambiente amistoso y cálido. En este sentido, quiero comenzar por darle un agradecimiento destacado al profesor titular Dr. J. Ignacio Mieza. Sin su dedicación a la docencia e investigación no se hubiera formado este grupo de profesionales. Por otro lado, tuve la suerte de que aceptara ser mi director de tesis. Su presencia, orientación y criterio han sido fundamentales. Su apoyo se mostró desde el primer momento, aún antes de que el tema de estudio específico fuera definido. Sin su guía difícilmente hubiera logrado realizar un trabajo tan gratificante.

Quisiera también brindar un reconocimiento especial para dos amigos muy próximos. A Tomás Ciccola, por acompañarme a lo largo de los años de formación informática. Sin su sostén, difícilmente hubiera desarrollado las herramientas de programación necesarias para esta tesis. Y al Lic. en Dirección Orquestal Lorenzo Guggenheim por haber fortalecido mi interés por la música, primero como compañero y luego como maestro. En especial, le reconozco su ayuda para definir y precisar una clasificación de instrumentos musicales útil a los fines de esta tesis.

Le agradezco también a mi familia por su apoyo incondicional a lo largo de los años. Gracias a ellos pude estudiar una carrera universitaria. Me enseñaron a valorar el beneficio que tuve por nacer en una casa privilegiada desde el punto de vista social e intelectual, eso me permitió acceder al título de ingeniero. Por supuesto, existe el esfuerzo personal para obtener un título universitario, pero la mayor carga la tuvieron mis padres Roberto y Cristina inculcándome el valor del trabajo a través del ejemplo sostenido durante años. Quiero también brindar mi gratitud a mis hermanas Laura, Julieta y Mariana por enseñarme a tomar decisiones. En particular, porque me apoyaron cuando decidí estudiar una carrera que nadie de mi familia conocía ni entendía de qué se trataba. Y por último quiero agradecerle a mi novia Milagros por ser mi compañera en las extensas jornadas en las que trabajé en esta tesis. Ella me motivó a dedicar el esfuerzo que esta investigación necesitó.

Índice general

Índice de figuras

Índice de tablas

Capítulo 1

Introducción y objetivos

En esta tesis se aborda el estudio de la identificación automática de instrumentos musicales acústicos. Para ello, primero se caracterizan sonoramente a estos instrumentos y luego se los agrupa por similitud usando criterios matemáticos de proximidad. Se decidió poner el enfoque en la identificación de sonidos aislados, es decir, instrumentos produciendo una única nota, porque es un campo donde existen todavía muchos problemas abiertos y en el que además existe un amplio corpus bibliográfico específico. Aunque debe mencionarse que existen estudios que utilizan señales más complejas como frases monotímbricas, dúos o incluso polifonías mayores [1].

Para comenzar, se explicarán algunos conceptos básicos que son necesarios para la comprensión global del trabajo. En particular, se describirán las familias de instrumentos musicales, el reconocimiento de patrones, y algunos métodos de aprendizaje autónomo (*Machine Learning*). Luego, se presentará el análisis realizado en las señales mediante los descriptores acústicos, los algoritmos de clasificación y las bases de datos utilizadas.

A continuación, se expondrá el núcleo de la investigación que es el algoritmo desarrollado para la identificación automática de instrumentos y el programa (*software*) con las dependencias y detalles de implementación. Se hará el análisis y la discusión de los resultados y se mostrará cómo afectan al resultado las variables involucradas en el proceso.

Finalmente se presentarán las conclusiones y las futuras líneas de trabajo que surjen de esta investigación.

1.1. Clasificación de instrumentos musicales

Por 'clasificación' se entiende al proceso de asignarle una clase a un objeto observado. Este objeto se define típicamente con un conjunto de magnitudes cuantitativas que representa ciertos atributos. En el caso del análisis de sistemas musicales, los atributos se computan a partir de un archivo de audio mediante técnicas de procesamiento digital de la señal. A lo largo de este trabajo, siguiendo la literatura consultada, a los atributos se los denominará 'descriptores' para indicar que son parámetros específicos de un sonido. Un ejemplo de un descriptor sonoro es la frecuencia fundamental. A las clases se las define como el tipo de instrumento musical; por ejemplo, una clase puede ser entonces 'Trompeta' o 'Contrabajo'.

Las clases se pueden clasificar mediante estructuras generales que posean las siguientes características [2]: Consistencia en los principios de clasificación, exclusividad mutua entre categorías y cobertura completa del universo que se intenta describir. Las clases que se utilizan en esta tesis se seleccionaron siguiendo estos lineamientos.

1.1.1. Familias de Instrumentos

En el presente trabajo se relaciona el concepto de clase con las familias de instrumentos musicales. Se puede pensar a estas últimas como una clase de mayor abstracción que la de los instrumentos específicos. En la tabla 1.1 puede verse un modelo de clasificación en familias de instrumentos (taxonomía) basado parcialmente en Hornbostel y Sach [3] que a su vez está relacionado con un sistema desarrollado a fines del siglo XIX por Victor Mahillon [4], restaurador de la colección del Conservatorio Real de Bruselas. El sistema Mahillon fue uno de los primeros en clasificar de acuerdo con el material o parte del instrumento que producía el sonido, pero estaba limitado en su mayor parte a los instrumentos occidentales usados en música clásica. El sistema Hornbostel-Sachs es una extensión del de Mahillon, en el que es posible clasificar cualquier instrumento musical de cualquier cultura. Más tarde, Sachs agregó una quinta categoría, los electrófonos, como el Theremín, que producen sonido por medios electrónicos. En la tabla 1.1 los instrumentos electrófonos fueron tomados de la revisión de la clasificación de Hornbostel-Sachs realizada por el MIMO Consortium [5].

Tabla 1.1: Taxonomía simplificada de los instrumentos musicales.

Familia	Sub-Familia	Tipo	Instrumento
Cordófono		Frotado	Violín, Viola, Violoncello, Contrabajo
		Pulsado	Ukelele, Guitara, Arpa
		Percutido	Piano
Aerófono	Madera	Bisel	Flauta, Quena
		Lengüeta Simple	Clarinete, Saxofón
		Lengüeta Doble	Oboe, Fagot
	Bronce	Con Válvulas	Trompeta, Tuba
		Sin Válvulas	Trombón
Idiófono		Percutido	Xilófono, Marimba
		Pulsado	Kalimba, Guimbarda
Membráfono		Percutido	Timbales, Tambores
Electrófono	Electroacústico	Idiófono	Piano Eléctrico
		Cordófono	Guitarra Eléctrica
	Digital	Sintetizador	Yamaha DX7, Serie CZ de Casio

Vale la pena también señalar la clasificación utilizada por Fletcher y Rossing [6], que simplemente dividen en instrumentos de cuerdas, de viento y percusivos. Esta clasificación si bien resulta menos técnica y/o precisa, al ser más sencilla resulta también más intuitiva.

1.1.2. Hombres y Máquinas

La tarea de clasificación de instrumentos ha sido históricamente realizada por humanos hasta la irrupción de las computadoras y los algoritmos de cálculo automático. En esta sección se presentarán ambos enfoques, mostrando las similitudes y diferencias.

Con repecto al desempeño de los humanos existen pocos estudios que hayan evaluado la capacidad de clasificar sonidos provenientes de diferentes instrumentos musicales. Martin [7] y Srinivasan et al. [8] han encontrado que los humanos, aún con entrenamiento musical, rara vez identifican bien qué instrumento están percibiendo en más del 90 % de los casos; y en estudios más complejos (con más clases) el desempeño cae al 40 %. En relación con lo anterior, verificaron que la clasificación es mejor cuando las personas tienen algún grado de educación musical y/o se los expone con anterioridad a la clasificación a muestras sonoras de distintos instrumentos. Además, notaron que la confusión en la identificación entre ciertos instrumentos

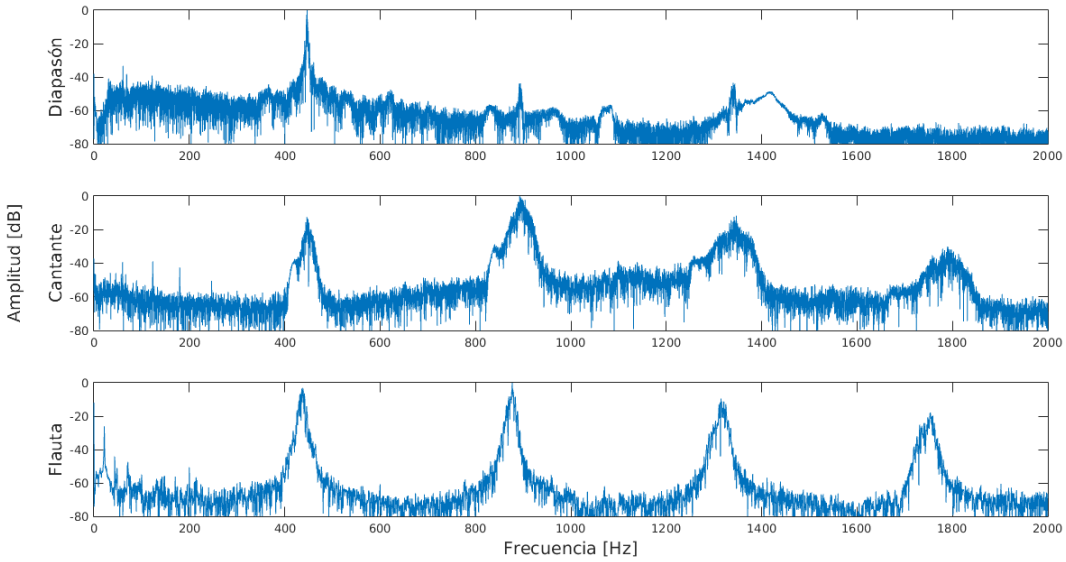


Figura 1.1: Comparación del espectro para tres instrumentos musicales distintos tocando la nota La-4 (fundamental en 440Hz). Puede señalarse que el diapasón, que normalmente se identifica con un sonido 'puro', tiene el espectro más sencillo de los tres. A su vez, la voz cantada tiene un espectro con picos anchos y poco definidos.

'similares' es sumamente común. Un ejemplo de esto se da entre la trompeta y el trombón. Concluyeron también que resulta más fácil identificar a la familia del instrumento que a este en si mismo. Por último, la información contextual (escuchar frases musicales en vez de notas aisladas) ayuda mucho a mejorar el proceso de identificación [9, 10].

Griffiths [11] busca relacionar aspectos fisiológicos de la audición con la capacidad de discriminación de sonidos. Sostiene que la identificación espacial se basa en información espectral y temporal que se extrae en el primer procesamiento entre la cóclea y la corteza auditiva primaria. Por otra parte, afirma que la clasificación de la fuente sonora en clases se hace en centros auditivos más elevados [11]. Para el caso particular de los instrumentos musicales afinados, la energía relativa de los armónicos parciales determina en cierto nivel las sensaciones de timbre y puede estar ligada a la identificación (ver figura 1.1). Parecería además que para sonidos sostenidos, la parte estable provee mucha más información que el ataque [12].

Con respecto a los métodos de clasificación algorítmica de instrumentos no puede hacerse una aproximación general ya que son múltiples los enfoques y métodos utilizados. Sin embargo, los resultados que se obtienen utilizando máquinas superan ampliamente a los hechos por humanos. Por ejemplo, el sistema de clasificación automática realizado por Szczuko et al. [13] logra clasificar 16 instrumentos musicales con acierto del 86 % para instrumentos individuales y del 89 % al clasificar la familia del instrumento musical. Por otro lado, el sistema desarrollado por Kitahara et al., 2003 [14] distingue 19 instrumentos distintos con un acierto del 80 %.

Como se señaló en la sección anterior, para los humanos la identificación de la familia de un instrumento musical resulta más sencilla que la del instrumento en si mismo. A partir de esta idea es que distintos investigadores realizan sus algoritmos de clasificación mediante un modelo llamado 'jerárquico'. Este primero define la familia del instrumento para así acotar las posibilidades a la hora de clasificarlo puntualmente.

La clasificación automática de sonidos de instrumentos musicales posee relevancia en varias áreas del conocimiento [1]:

- Desde la acústica, para entender qué hace que un sonido particular sea 'identificable' respecto a otros instrumentos;

- Desde el estudio de la percepción, para entender qué hace que dos instrumentos suenen 'similares';

- En la organización de bases de datos sonoras, para proveer automáticamente de clases de sonidos y así poder realizar búsquedas particulares. En este ámbito también entran las librerías de sintetizadores.

- Desde la musicología, para localizar la instrumentación de una pieza musical y los solos en la misma.

1.2. Aprendizaje Autónomo (*Machine Learning*)

Un algoritmo de *Machine Learning*[1] es un programa que puede 'aprender' a partir de un conjunto de información. En las palabras de Mitchell [15]: 'Se dice que un programa de computadora aprende de la experiencia E con respecto a una clase de tareas T con una medida de *performance* P, cuando su performance en las tareas T, medida por P, mejora a partir de la experiencia E'. En el caso de esta tesis, la tarea es clasificar muestras sonoras según el instrumento musical al que pertenecen. La experiencia viene de repetir esta acción para varias muestras sonoras distintas conocidas; y la performance se puede medir gracias a varios métodos que se presentan más adelante.

1.2.1. Las Tareas, T

Bengio et al. [16] afirman que el aprendizaje autónomo es interesante por las tareas que permite realizar. Desde el punto de vista ingenieril, este paradigma de programación nos permite afrontar tareas que serían demasiado difíciles de resolver con programas fijos escritos y diseñados por humanos. Cabe destacar que 'aprender' en si mismo no es la tarea. Aprender es el medio de obtener la habilidad necesaria para realizar la tarea.

Existen muchas tareas que se pueden resolver con algoritmos de *Machine Learning*. Como por ejemplo:

- *Clasificación*: En este tipo de tarea, el programa debe especificar a cuál de entre k categorías pertenece algún tipo de objeto de entrada. Para resolver esta tarea, se le pide al algoritmo de aprendizaje que produzca una función del tipo de $f : \mathbb{R}^n \rightarrow \{1, ..., k\}$ que pueda ser aplicada en el objeto de entrada. Entonces, la salida $F(x)$ puede interpretarse como una estimación de la categoría a la que x pertenece. Existen otras variantes de la tarea de clasificación, por ejemplo, donde f produce una distribución de probabilidades sobre las clases. Un ejemplo de esta tarea es el reconocimiento de objetos en imágenes. Estas imágenes suelen describirse como un conjunto de píxeles y la salida es un código numérico identificando un objeto en la imagen. Esta tecnología es la que permite que las computadoras reconozcan caras y puede ser utilizada para etiquetar automáticamente personas en una colección de fotos [17].

- *Regresión*: En este tipo de tarea, el programa debe predecir un valor numérico a partir de un objeto de entrada. Para resolver esta tarea, se le pide al algoritmo de aprendizaje que produzca una función del tipo de $f : \mathbb{R}^n \rightarrow \mathbb{R}$. Este tipo de tarea es similar a la clasificación, pero el formato de su salida es diferente. Un ejemplo de una tarea de regresión es la predicción de cuánto dinero debe dar una compañía de seguros para compensar un infortunio o predecir el precio futuro de la canasta básica de alimentos.

[1] En el resto de la tesis se usará indistintamente el término aprendizaje autónomo y *Machine Learning*

- *Detección de anomalías*: En este tipo de tarea, el programa analiza una serie de eventos u objetos y señala los inusuales o atípicos. Un ejemplo de detección de anomalías es la detección de fraudes en las compras de una tarjeta de crédito. Al modelar los hábitos de compra de un cliente, la compañía de tarjetas de crédito puede detectar usos extraños de sus tarjetas. Si su tarjeta de crédito es robada, es esperable que se utilice para compras muy diferentes de las habituales. En ese momento la compañía puede frenar la transacción y congelar la cuenta hasta que haya un tipo de confirmación extra por parte del dueño.

- *Síntesis y muestreo*: En este tipo de tarea, se le pide al algoritmo de *Machine Learning* que genere nuevos ejemplos similares a los de una base de datos. Esto es útil en aplicaciones multimedia donde puede ser costoso que un artista genere grandes cantidades de contenido manualmente. En la industria de los videojuegos por ejemplo, se generan texturas para objetos grandes o paisajes utilizando este tipo de algoritmo en lugar de requerir que el artista cree cada píxel [18]. En algunos casos, el resultado que se busca es más específico. Por ejemplo, en una tarea de síntesis vocal, se le da al programa un oración escrita y se le pide que genere un archivo de audio conteniendo una versión hablada de la oración. En este tipo de tarea, no existe una única versión correcta del audio que puede entregar el algoritmo, sino que se busca que exista variación para cada vez que se deba generar la frase. De esta forma, el resultado es más natural y realista.

- *Denoising*: En este tipo de tarea, el algoritmo es alimentado con un ejemplo ruidoso $\tilde{x} \in \mathbb{R}^n$ obtenida a partir de una muestra pura $x \in \mathbb{R}^n$ que ha sido corrompida. El algoritmo de aprendizaje debe predecir el ejemplo puro x a partir de su version ruidosa $\tilde{x}$, o más generalmente predecir la distribución de probabilidad $P(x|\tilde{x})$.

Cabe destacar que la lista de tareas presentada aquí no es exhaustiva. De hecho es un resumen de la lista hecha por Bengio et al. [16]. La lista original tampoco pretende proveer una taxonomía rígida de las tareas sino dar una serie de ejemplos.

1.2.2. Medida del desempeño, P

Para evaluar las habilidades de un algoritmo de *Machine Learning* se debe diseñar una medida cuantitativa de su desempeño. Generalmente la forma de medir el desempeño es específica a la tarea que está siendo abordada por el sistema. Por ejemplo, para la tarea de clasificación se suele medir la *exactitud* del modelo, que simplemente indica la proporción de ejemplos para los cuales el modelo entrega el resultado correcto. Se puede obtener información equivalente midiendo la proporción de ejemplos para los cuales el modelo entrega un resultado incorrecto [16].

Generalmente, el interés está en ver cuán bien se comporta el algoritmo frente a la información que no conoce. Esto se debe a que esta es la medida indicativa de cómo se comportará en el mundo real. Por consiguiente, el desempeño se evalúa sobre un grupo de muestras llamadas 'conjunto de pruebas' que es distinto al 'conjunto de entrenamiento'. Las bases de datos y los distintos tipos de conjuntos de datos se presentan con mayor profundidad en el capítulo 3.

La elección del método de medición del desempeño puede parecer trivial y objetiva, pero no lo es. El método seleccionado para evaluar el comportamiento del sistema es la herramienta principal para evaluar el método de aprendizaje adoptado. Un caso donde esta elección errónea puede ser nefasta se da en la tarea de detección de anomalías. En este tipo de problemas la base de datos suele estar compuesta esencialmente (95 %) de muestras normales. Un sistema que simplemente clasifique como muestra normal a cualquier muestra sin procesarla podría tener un nivel de desempeño muy elevado. En el capítulo 4 de esta tesis se profundiza sobre este tema y se presenta el método utilizado en el presente trabajo.

1.2.3. La experiencia, E

Los algoritmos de aprendizaje autónomo pueden dividirse según la experiencia o entrenamiento en dos categorías: 'supervisados' y 'no supervisados'.

Los algoritmos de aprendizaje no supervisado se entrenan con una base de datos que contiene descriptores de un objeto x e intentan obtener propiedades útiles sobre la estructura de esta. En estos algoritmos, el sistema observa al grupo de ejemplos desconocidos e intenta implícita o explícitamente obtener la función de distribución de probabilidad $p(x)$, u otra característica interesante sobre su distribución. En este grupo caen los algoritmos de 'agrupamiento' cuyo objetivo es dividir los vectores de x en grupos a través de un análisis sobre su proximidad matemática.

Los algoritmos de aprendizaje supervisado, en cambio, se entrenan con una base de datos que contiene descriptores sobre un objeto x y donde además cada ejemplo está asociado a una 'clase' o 'etiqueta'. Por consiguiente, el algoritmo intenta deducir la función que debe implementar en los vectores de entrada para obtener la clase esperada a la salida. En este grupo están los algoritmos de 'clasificación'.

Capítulo 2

Marco Teórico

2.1. Descriptores y su selección

La clasificación no sería posible sin descriptores que capturaran las diferencias o similitudes entre dos objetos. En el caso de la clasificación de instrumentos, los descriptores son acústicos y se extraen de las señales de audio. De acuerdo a la bibliografía estudiada, la correcta selección de los descriptores es más importante que la elección del algoritmo de agrupamiento, y la clasificación en si misma es más sencilla cuando los descriptores son suficientemente informativos [19].

El término descriptor denota una cantidad o cualidad que caracteriza a un objeto. También se conoce como característica o atributo. Como el objetivo general de la clasificación es distinguir entre ejemplos que pertenecen a diferentes clases, uno de los objetivos particulares en la selección de descriptores es minimizar las variaciones de los valores del atributo dentro de una clase, mientras que se maximiza la variabilidad entre las clases. Además, se pretende que los descriptores no sean redundantes. En esta tesis este objetivo se logra gracias al estudio realizado de los atributos acústicos y la información que proveen. Sin embargo, si no se poseyera este conocimiento, se podría automatizar la selección de los descriptores con métodos estadísticos y de proyección matemática.

2.1.1. Selección de los descriptores

Como ya se mencionó, la selección de descriptores puede estar automatizada o realizarse mediante un estudio del problema. En esta investigación se estudiaron ambos métodos. En un principio se desarrolló un prototipo donde la selección de descriptores fue automática y para el trabajo final se realizó un estudio específico de las señales para la elección de los descriptores. El tipo de selección realizada posee mucho mayor poder teórico ya que requiere una comprensión del problema previo a la clasificación de las muestras. Los descriptores además fueron seleccionados cuidando que proporcionaran distintos tipos de información, dado que Herrera sugiere que este enfoque tiene efectos positivos [1]. Por otro lado, siguiendo la recomendación informal de Jain, Duin, y Mao [20] se seleccionaron al menos diez veces menos descriptores que la cantidad de muestras en la base de datos.

Para evaluar la independencia de los descriptores elegidos se realizó un estudio de correlación de Pearson [21], también conocido como correlación lineal [22]. Este coeficiente determina el grado de proporcionalidad o linealidad existente entre dos variables. El coeficiente de correlación de Pearson puede consultarse en la ecuación 2.1 donde x, y son los descriptores evaluados

y $\bar{x}$ indica la media de x.

$$r_{xy} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}} \tag{2.1}$$

Los descriptores seleccionados para el software pueden consultarse en la tabla 2.1. Por otro lado, en la tabla 2.2 puede observarse el análisis de la correlación final que se obtuvo para estos descriptores. Dado que para ningún par de descriptores la correlación en la mayoría de las muestras es elevada, no se descartó ningún descriptor en esta etapa.

Tabla 2.1: Se listan las 4 clases de descriptores utilizadas en el software junto con los descriptores específicos utilizados en cada caso.

Tipo de Descriptores	Descriptores
Energéticos	La Energía.
Temporales	El tiempo logarítmico de ataque, la desviación estándar, la varianza, skewness y kurtosis.
Espectrales	Los Coeficientes Cepstrales en las Frecuencias de Mel (MFCC), sus desviaciones estándar, sus varianzas, sus skewness y sus kurtosis.
Armónicos	La inarmonicidad con su primera y segunda derivada.

Tabla 2.2: Se listan los 10 pares de descriptores que mostraron correlación lineal en la mayor cantidad de ejemplos para una base de datos de 1550 muestras. En este análisis, la correlación solo es tenida en cuenta si su valor es mayor a 0,85.

Descriptores	Repeticiones	Promedio
MFCC[0]-MFCC[1]	724	0.96
MFCC[0]-[2]	429	0.89
MFCC[1]-[2]	239	0.88
MFCC[10]-[11]	214	0.89
MFCC[9]-[10]	183	0.88
MFCC[0]-[4]	181	0.88
MFCC[3]-[4]	179	0.87
MFCC[8]-[9]	169	0.88
MFCC[1]-[4]	150	0.88
MFCC[7]-[8]	147	0.88

2.1.2. Espacios tímbricos

La selección de los descriptores se abordó también desde el estudio de los espacios tímbricos. Un espacio tímbrico es una representación abstracta utilizada para clasificar el grado de similitud percibida entre dos sonidos. Al ser un método ligado a la percepción humana no es necesariamente óptimo, pero puede dar indicios sobre qué propiedades acústicas utilizar en primera instancia para identificar instrumentos en forma automática .

Para determinar los espacios tímbricos, primero se le pide a una audiencia que realice juicios sobre el grado de similitud entre un dueto o tríada de sonidos. Esto es expresado usando una escala que va de 'muy similar' a 'muy diferente'. Los resultados del test se procesan usando una técnica de reducción de dimensionalidad conocida como 'escalamiento multidimensional' (MDS) para poder ser visualizados. Así, se genera un espacio continuo delimitado por dos o tres parámetros que tienen un rol importante en la sensación de timbre. Se pretende que las muestras de un mismo instrumento se agrupen en este espacio al mismo tiempo que la distancia entre grupos sea máxima. Luego, se calcula la correlación entre los parámetros y descriptores acústicos objetivos calculados sobre las mismas muestras sonoras que utilizó la audiencia para la prueba. De esta manera, pueden obtenerse conclusiones significativas.

El pionero en la elaboración de los espacios tímbricos fue Grey en el año 1977 [23], usando sonidos sintetizados para emular doce instrumentos orquestales de las familias de los cordófonos y los aerófonos. Luego, utilizó una audiencia de 20 personas para que juzguen la similitud entre pares de muestras y aplicó MDS para derivar los resultados en un espacio tímbrico de tres dimensiones. La descripción cualitativa de los ejes que encontró al calcular la correlación con descriptores acústicos fue:

1. La distribución espectral de la energía,
2. El sincronismo en el comienzo de los transitorios,
3. Las variaciones temporales en la forma de onda del espectro del sonido.

Más adelante tanto Wessel [24] en 1982, como Krumhansl [25] en 1989, y Lakatos [26] en el año 2000 sostienen la predominancia de una dimensión de 'brillo' (profundamente relacionada con el primer momento de la distribución espectral de energía). Este parámetro es el predominante para organizar los sonidos en espacios tímbricos. Otro atributo importante está relacionado con el tiempo de ataque. Y finalmente una tercer dimensión, con menos apoyo en las investigaciones citadas, relacionada con el 'flujo espectral' (cuán rápido cambia el espectro de potencia de la señal) . Estos y otros descriptores fueron incluidos para la descripción tímbrica en el estándar MPEG-7 [27].

2.1.3. Pre-procesamiento de la Señal

Antes de extraer los descriptores, todas las señales deben ser pre-procesadas para minimizar la información inútil obtenida. Este tipo de información es la que está ligada no al objeto de estudio (el instrumento) sino a la forma en que se lo observa (cómo fue grabado). Por consiguiente, se realizaron dos operaciones importantes sobre la base de datos:

- **Silencio**: Se evaluó dónde comienza la información útil en cada objeto sonoro y dónde termina. De esta forma, los descriptores se calculan únicamente sobre el fragmento donde el instrumento está sonando.
- **Niveles**: Se divide a cada punto de la señal por su valor máximo (normalización). De esta forma, todas las señales tienen un nuevo valor máximo igual a 1. Así se intenta disminuir la influencia de la estructura de ganancia utilizada en la grabación de las señales.

2.1.4. Descriptores Temporales y Energéticos

Los descriptores temporales y energéticos se evalúan de manera conjunta. Los parámetros temporales brindan información sobre cómo está distribuida la energía en el tiempo, mientras que los energéticos evalúan el contenido de energía completo de la señal.

Temporales

Para los descriptores temporales se utilizan cinco parámetros; la desviación estándar, varianza, skewness y kurtosis de la energía en el tiempo y el 'tiempo de ataque logarítmico'.

El 'tiempo de ataque logarítmico' es el logaritmo en base 10 del tiempo de ataque de la envolvente de la señal. El tiempo de ataque se define como la duración temporal desde que el sonido es audible hasta que llega a su máxima intensidad. Para este trabajo, el principio del ataque se consideró cuando la envolvente de la señal llega al 20 % de su valor máximo y el final del ataque cuando la envolvente de la señal llega al 90 % de su valor máximo. De esta forma, con el valor mínimo se puede asegurar que el instrumento se ha desprendido del ruido de fondo y con el valor máximo se evitan imprecisiones en instrumentos como la trompeta donde el pico de la señal ocurre luego de que el ataque haya ocurrido.

Energéticos

El parámetro energético seleccionado fue simplemente la energía de las señales. Esta se obtiene siguiendo la fórmula clásica:

$$Energía(señal) = \sum_{t=0}^{N} señal[n]^2 \tag{2.2}$$

Dada la naturaleza de las señales, se sabe que son señales de energía y que este valor debe ser finito y mayor a cero.

2.1.5. Descriptores Espectrales

Para analizar la información espectral de las señales se recurrió a los Coeficientes Cepstrales de las Frecuencias de Mel (MFCC). Estos son derivados del análisis cepstral y se utilizaron en un principio para el reconocimiento automático del habla [28]. El análisis cepstral (MFC) es una representación del espectro de potencia en períodos cortos de tiempo. Está basado en la transformada coseno del espectro logarítmico de potencia luego de realizar un escalamiento 'Mel' no lineal a la frecuencia de la señal. Cada coeficiente se utilizó para el análisis como un descriptor de las señales. Se seleccionaron estos coeficientes en lugar de la transformada de Fourier (FT) o la transformada coseno discreta (DCT) ya que los MFCC sitúan las bandas de frecuencia logarítmicamente (según la escala Mel). Esto modela la respuesta auditiva humana más apropiadamente que las bandas espaciadas linealmente de FT o DCT. Además, se ha encontrado empíricamente que estos coeficientes forman un conjunto mucho más confiable para el reconocimiento del habla y la identificación de hablante que la Codificación Predictiva Lineal (LPC) u otro conjunto de parámetros equivalentes [29].

El diagrama en bloques presentado en la figura 2.1 expone un algoritmo usual para la extracción de los Coeficientes Cepstrales de la Frecuencias de Mel [30] [31].

1. Se toma la transformada de Fourier de una porción de la señal mediante una ventana.
2. Se mapea la energía del espectro obtenido a la escala Mel usando ventanas triangulares superpuestas.
3. Se calculan los logaritmos de la potencia para cada banda de frecuencias de Mel.
4. Se realiza la transformada discreta del coseno para cada banda de energía logarítmica, como si fuera una señal.

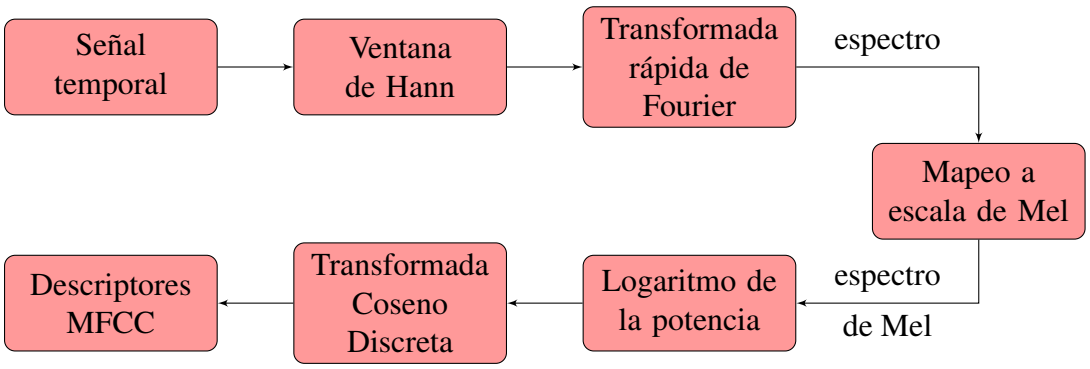


Figura 2.1: Diagrama en bloques del cálculo de las MFCC para un frame.

5. Los MFCCs son las amplitudes del espectro resultante.

Además de estos coeficientes se realizó un análisis estadístico de la distribución de la energía en las bandas de frecuencia. Para esto se calculó en cada ventana la desviación estándar, la varianza, la kurtosis y la skewness entre las 12 bandas seleccionadas para los coeficientes. Por otro lado, se calculó la primera y la segunda derivada temporal para cada banda de los coeficientes. De esta forma, si bien el algoritmo promedia los coeficientes mfcc en el tiempo, se estima que las derivadas aporten la información de la distribución espectral respecto al tiempo.

2.1.6. Descriptores Armónicos

Como descriptor armónico se eligió la inarmonicidad junto con su primer y segunda derivada. La inarmonicidad representa la divergencia de los componentes espectrales de la señal evaluada de los de una señal armónica pura [32]. El diagrama en bloques del algoritmo utilizado para calcular este descriptor en un segmento (*frame*) temporal puede consultarse en la figura 2.2. El algoritmo se encarga de, una vez dividida la señal en segmentos, estimar la frecuencia fundamental de la señal y detectar todos los picos en el espectro de la señal (ver picos en la figura 1.1). Una vez detectada la frecuencia fundamental, se calculan los armónicos 'puros' como múltiplos enteros de esta. Gracias a la detección previa de los picos en el espectro, puede calcularse la diferencia entre estos picos y los múltiplos enteros de la frecuencia fundamental. A esta diferencia es que se la conoce como la inarmonicidad de la señal. Luego de realizar esta acción sobre cada frame de la señal, se realiza un promedio de los valores obtenidos para generar el valor utilizado en el algoritmo de clasificación.

2.1.7. Post-procesamiento de los descriptores

Estandarización de los descriptores

Antes de poder ingresar el vector de descriptores en el algoritmo de clasificación, este debe ser normalizado. Como cada descriptor puede tener un rango de valores particular, si estos no se estandarizan, el algoritmo de clasificación tendrá mayores dificultades para separar los objetos. Puede interpretarse que algunas dimensiones parecen 'aplastadas' (ver la figura 2.3 para un ejemplo gráfico).

En su libro 'Data Mining - Practical Learning Tools and Techniques' Witten, Frank, y Hall [33] afirman que esto suele realizarse de tres maneras: (1) se puede normalizar a un rango fijo dividiendo todos los valores por el máximo, (2) substrayendo el mínimo valor y dividiendo por el rango que poseen los valores o, (3) substrayendo el promedio de los valores y dividiendo por la desviación estándar. Los dos últimos métodos fueron utilizados en esta tesis y se encontró que para el algoritmo particular utilizado el tercer método provee mucha mejor performance.

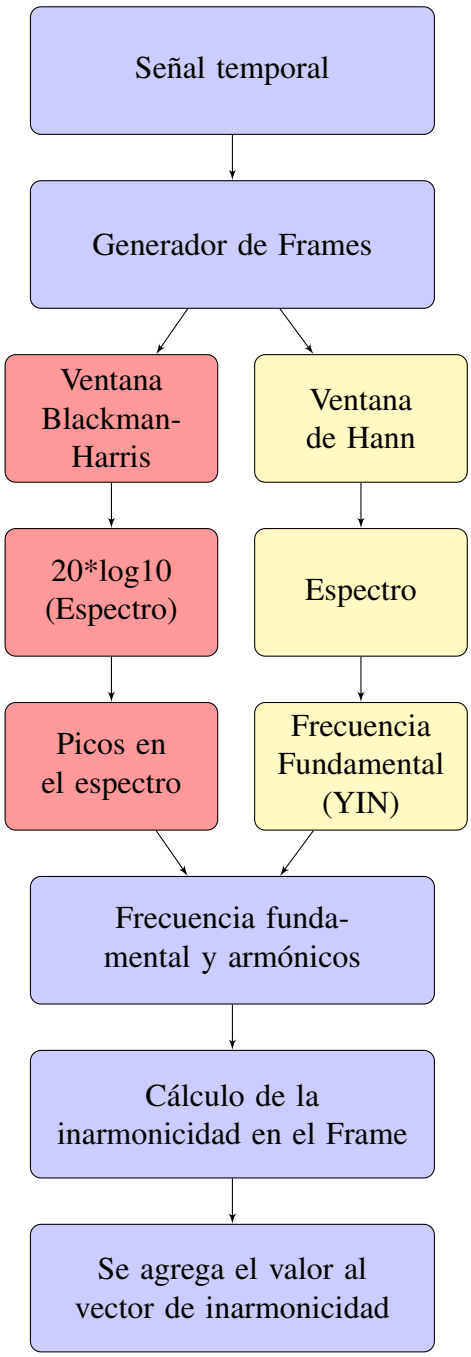


Figura 2.2: Diagrama en bloques del cálculo de la inarmonicidad para un frame.

Por consiguiente la normalización utilizada consistió en realizar la siguiente operación:

$$Descriptor\ Normalizado = \frac{Descriptor - \overline{Descriptor}}{\sigma^2(Descriptor)} \tag{2.3}$$

Proyección de los descriptores

Siguiendo los consejos de los trabajos realizados por Peeters en el IRCAM [32, 34, 35] se decidió realizar una proyección de los descriptores para disminuir las dimensiones del vector que los agrupa. De esta forma se logran dos objetivos. Por un lado, se puede analizar cómo se interrelacionan los descriptores y evaluar que cumplan con las características planteadas al principio de este capítulo (minimizar las variaciones dentro de una clase y maximizar la

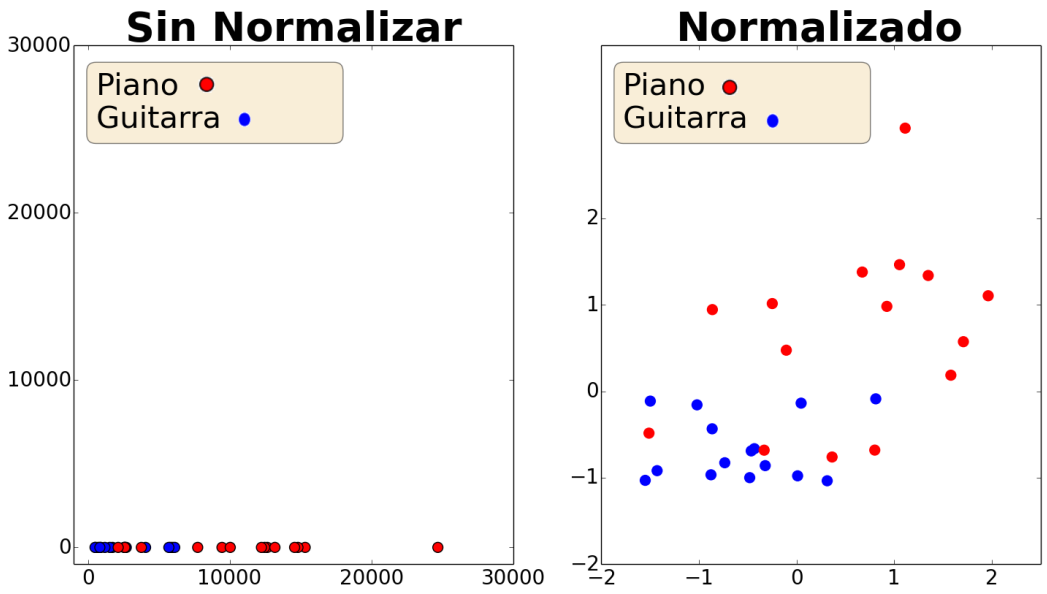


Figura 2.3: Se puede observar gráficamente cómo la diferencia de rango para los descriptores aporta problemas a la hora de la clasificación.

variabilidad entre las clases). Por otro, la cantidad de dimensiones del vector está directamente ligada a la cantidad de operaciones que deberá realizar el algoritmo de clasificación sin importar de qué tipo sea. Esto se debe que a las operaciones matriciales tienden a poseer una complejidad de $O(N^3)$, donde N es la cantidad de dimensiones de la matriz. Esto significa que al duplicar el tamaño de la matriz, el tiempo para realizar una operación aumenta por un factor de $2^3 = 8$. En otras palabras, al duplicar el tamaño del problema, el tiempo de cálculo será aproximadamente diez veces mayor [36].

Janert [36] sostiene que el Análisis de Componentes Principales (PCA) se puede utilizar para los dos objetivos propuestos. De hecho los define como técnicas exploratorias y técnicas preparatorias. Para el algoritmo utilizado se siguió la implementación del modelo probabilístico de M. Tipping y C. Bishop presentado en su artículo 'Probabilistic Principal Component Analysis' [37]. A su vez, este algoritmo fue provisto por la librería Scikit-learn que será presentada en el capítulo 3 de esta tesis.

2.2. Algoritmos de clasificación

En la literatura consultada no existe consistencia sobre las herramientas utilizadas en la clasificación de instrumentos musicales acústicos. En la tabla 2.3 se listan varios trabajos sugeridos por Herrera [1] dado que utilizan una cantidad suficiente de muestras diferentes de cada instrumento y además tratan con distintas condiciones de grabación para estas. Puede observarse que todos los trabajos tienen una combinación única de descriptores acústicos, algoritmos de clasificación e incluso medición del desempeño (aunque estos métodos no estén listados en la tabla). A su vez, si bien en las investigaciones de esta área suelen utilizarse sistemas de descriptores acústicos basados en la transformada rápida de Fourier [1], algunos trabajos se basan en wavelets [38] [39], correlogramas [7], transformada de Hough [40], predicción lineal [41] [42], y transformada de Q constante [43] [10] [44]. También cabe destacar que algunas investigaciones realizan una clasificación jerárquica. Para esto, agrupan distintos instrumentos en una clase más amplia que el instrumento en si mismo como por ejemplo su familia (para el violín, las

cuerdas). Luego, generan un sistema con un algoritmo que identifique estas familia y luego para cada familia, otro algoritmo detecta el instrumento en particular.

Tabla 2.3: Compilación de investigaciones orientadas al reconocimiento automático de instrumentos musicales acústicos utilizando señales monofónicas. La columan NC muestra el Número de Clases para cada sistema. En la última columna, el desempeño de sistemas jerárquicos está indicado entre paréntesis como '(J:)'.

Autor, año [ref]	Cantidad de muestras	NC	Algoritmo de Clasificacion	Desempeño (%)
Eronen, 2001 [41]	5286	29	k-Vecinos Cercanos (k-NN)	35 (J:30)
Livshin et al., 2003 [35]	4381	16	Análisis de discriminantes lineales (LDA) y k-NN	47-69
Peeters, 2003 [32]	4163	23	LDA & Modelo de Mezclas Gaussianas (GMM)	54 (J: 64)
Eronen, 2003 [45]	5895	7	Modelo Oculto de Márkov (HMM)	68
Kitahara et al., 2003 [14]	6247	19	Bayes (k-NN tras PCA y LDA)	80
Kostek et al., 2004 [46]	no informa	12	Perceptrón multicapa (MLP)	71
Szczuko et al., 2004 [13]	2517	16	MLP	86 (J: 89)
Park et al., 2005 [47]	829	12	MLP	71
Chétry et al., 2005 [48]	4415	11	K-means	95

Al diseñar un clasificador para sonidos de instrumentos musicales acústicos, surge el problema de seleccionar un único clasificador para todas las clases o utilizar varios clasificadores 'enfocados'. Para el primer caso, se habla de clasificadores planos, en el segundo hay un espectro de enfoques disponibles que van desde los clasificadores jerárquicos hasta los conjuntos de clasificadores [1].

Para esta tesis, se utilizó un clasificador plano dado que el objetivo es clasificar el instrumento en particular y no se encuentra una ventaja en averiguar la familia del mismo. Sin embargo, quedan sentadas las bases necesarias para implementar un clasificador jerárquico. Queda como trabajo a futuro evaluar qué metodología otorga resultados más útiles.

El procedimiento para la clasificación supervisada se puede resumir de la siguiente forma:

1. Se seleccionan ciertos descriptores acústicos para describir a las señales de audio.

2. Se calculan los valores de estos descriptores para una base de datos de entrenamiento que contiene ya clases asignadas para cada muestra.

3. Un algoritmo de entrenamiento usa los descriptores seleccionados para aprender a distinguir entre las clases de instrumentos. A su vez, se lo optimiza con la base de datos.

4. Se evalúa la capacidad de generalización del procedimiento de aprendizaje

2.2.1. K-means

Para el primer prototipo del sistema se implementó el algoritmo de agrupamiento conocido como "k-means". Este método se utiliza para encontrar grupos en un conjunto de información no clasificada utilizando como métrica la distancia euclídea [49].

Para utilizar este algoritmo, se deben describir las muestras que se desea separar en grupos y colocarlas en un hiper-plano generado a partir de sus vectores de descriptores. k-means funciona colocando centroides en este hiper-plano y desplazándolos en el mismo para minimizar la varianza dentro de cada grupo. El algoritmo básicamente itera en dos pasos:

1. Para cada centroide, se define un subconjunto de puntos de entrenamiento que estén más cerca de este centroide que de cualquier otro.

2. El promedio de cada descriptor de este grupo es calculado y este a su vez se transforma en el nuevo centroide.

Ambos pasos se repiten hasta que el algoritmo converja. Típicamente los centroides iniciales son elegidos al azar entre las muestras de la base de datos.

La programación se realizó utilizando la librería SciPy [50]. Al ser esta libre, se garantiza la reproducibilidad de la investigación.

2.2.2. Máquinas de vectores de soporte (*Support Vector Machines, SVM*)

En esta tesis se obtuvo una base de datos con etiquetas correspondientes a las clases reales de las muestras, por lo que el clasificador puede ser supervisado. Dado que los algoritmos de k-means son semi-supervisados o no supervisados, se decidió que no era la herramienta que aprovechara al máximo la información recopilada. A su vez, la clasificación del prototipo resultaba lenta, por lo que se buscó un algoritmo que realizara una clasificación más veloz. En particular, Herrera [1] sostiene que tanto las Redes Neuronales (NN) como las Máquinas de Vectores de Soporte (SVM) cumplen con esta condición una vez que fueron entrenados.

Las SVM son modelos de aprendizaje supervisados capaces de analizar información y reconocer patrones. Se utilizan tanto para la clasificación como para la regresión. Se seleccionó utilizar un clasificador de Máquinas de Vectores de Soporte ya que requiere menos parámetros, menos tiempo para ser entrenado y usado y ofrece un desempeño similar o mayor al de las Redes Neuronales con un sobre-entrenamiento mínimo [1].

Algunas de las características principales de las SVM se listan a continuación:

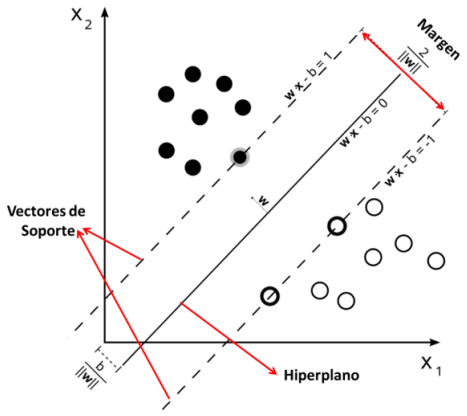
- Robustas cuando existen una gran cantidad de variables y pocas muestras de entrenamiento.
- Pueden usarse para aprender desde problemas sencillos hasta de otros de gran complejidad.
- Usa un grupo de puntos de entrenamiento en la función de decisión (conocidos como los vectores de soporte), por lo que tiene un uso de memoria eficiente.
- Por su construcción tienden a evitar el sobreentrenamiento del modelo.
- Son versátiles. Mediante distintas funciones de *'kernel'* se puede representar a las variables en distintas dimensiones y configuraciones.
- Las SVM no proporcionan estimaciones de probabilidad de manera directa. Estos pueden ser calculados pero los algoritmos son costosos en materia de cómputos.

Para la implementación se utilizó la librería de Python llamada 'Scikit-Learn' [51] que se presenta en la sección 3.4.2.

Clasificación Lineal

Las máquinas de vectores de soporte en su versión original se utilizan para hacer clasificaciones binarias [52]. El método busca la separación óptima entre las clases a clasificar mediante un hiperplano, maximizando la distancia (margen) entre los puntos de las clases que están más próximos. Finalmente, lo que se resuelve es un problema de optimización que tiene una solución eficiente mediante programación cuadrática. Aquellos puntos que forman la frontera se denominan vectores de soporte; y el hiperplano que separa las clases se encuentra en el medio del margen entre dichas fronteras. Para ilustrar estos conceptos se muestran en la Figura 2.4 los elementos de las SVM para un caso de dos dimensiones.

Figura 2.4: Elementos de las SVM para el caso de una clasificación en dos dimensiones [53].



Más formalmente, la ecuación de un hiperplano se indica en la Ec. 2.4., en donde w es un vector de pesos y x el vector de los descriptores. En particular, dicha ecuación, representa a una recta en el caso de 2 dimensiones y a un plano en el caso de 3 dimensiones.

$$w * x - b = 0 \tag{2.4}$$

Luego, dado un conjunto de datos de entrenamiento x_i, y_i, en donde los x_i corresponden a vectores (en este trabajo a los descriptores de las señales acústicas), y los y_i pueden tomar como valor a 1 o -1 para identificar alguna de las dos clases posibles. Se puede demostrar que si los datos son separables linealmente, la distancia entre los vectores de soporte (hiperplanos) que definen el margen es $\frac{2}{\|w\|}$. El método trata de maximizar dicho margen o distancia, o lo que es equivalente a minimizar el término $\|w\|$. Dicha minimización puede realizarse utilizando los métodos estándar de programación cuadrática (la función a optimizar es cuadrática con restricciones lineales).

Clasificación No Lineal (Kernel Trick)

Cuando los datos que se utilizan no pueden separarse linealmente, como se muestra en la Figura 2.5, se puede utilizar una función Φ (*kernel*) que modifica los datos x (en general aumenta la dimensión) y así permite que se separen mediante un hiperplano.

Este algoritmo no lineal es conceptualmente similar al caso lineal, pero se reemplazan todos los productos internos entre vectores por las funciones no lineales de los *kernels*. Esto permite aplicar el criterio descrito para encontrar el máximo margen en el nuevo espacio transformado. Algunos de los *kernels* comunes que se utilizan son [54, 55]: lineal, polinomial, Gaussiano y tangente hiperbólico. El *kernel* elegido determina la forma que toma la frontera de decisión como puede observarse en la figura 2.6.

Figura 2.5: Aplicación de un kernel que transforma los datos no separables linealmente en 2D (a la izquierda) en datos en 3D separables mediante un plano (a la derecha). $w * \Phi(x) - b = 0$.

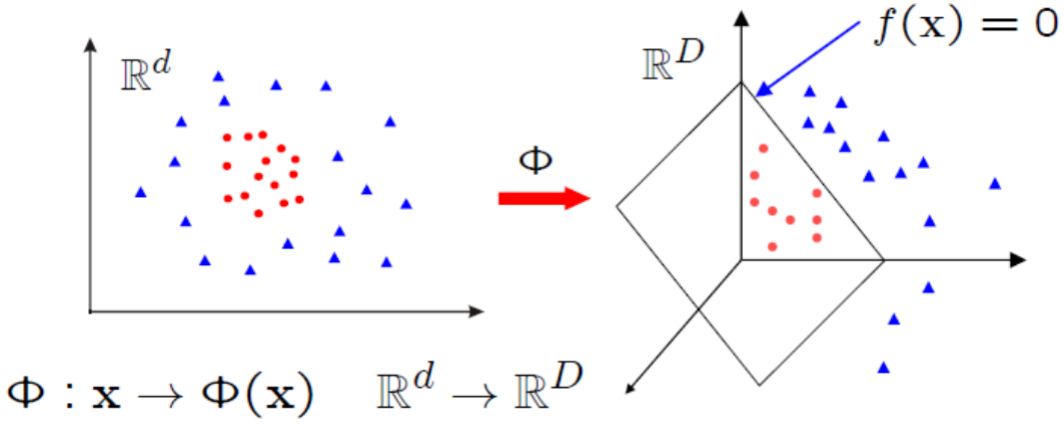
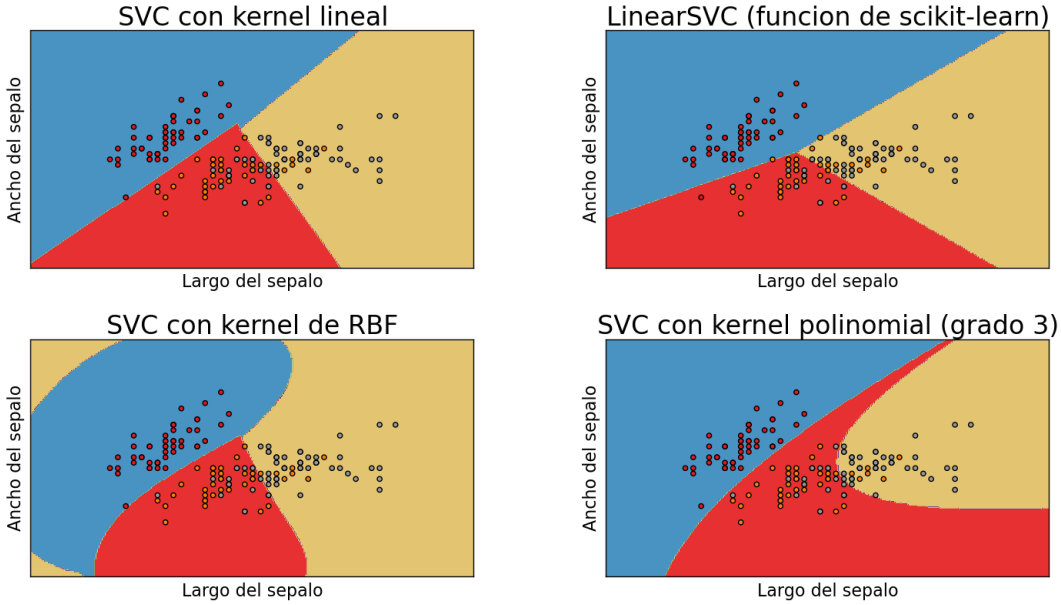


Figura 2.6: Evaluación de la frontera de decisión según la función de kernel, imagen tomada de la documentación de Scikit-learn [56].



En particular, el *kernel* Gaussiano, también denominado RBF (Radial Basis Function), se muestra en la ecuación 2.5 porque es el que se utiliza en este trabajo.

$$k(x_i, x_j) = e^{-\gamma \|x_i - x_j\|^2} \tag{2.5}$$

Este evalúa la similitud entre una muestra x_i y un 'punto de evaluación' x_j. Para esto, mide el cuadrado de la distancia euclidiana entre ambas muestras, lo multiplica por el parámetro de regularización γ y calcula la función exponencial con el negativo del valor encontrado. Por consiguiente, si ambas muestras poseen valores similares, el kernel tendrá valores cercanos a 1 y si las muestras poseen valores distintos, el kernel será cercano a 0. En el algoritmo desarrollado para esta tesis, la proximidad matemática se calcula sobre los valores que toman los descriptores. Los puntos de evaluación son un grupo de muestras de la base de entrenamiento.

Clasificación con Márgen Suave (Soft Margin)

Una modificación del algoritmo original [57], conocido como margen suave (Soft Margin) permite la inclusión de una cantidad de datos clasificados erróneamente. El método busca el hiperplano que mejor separa a los datos a través de un parámetro de penalización. El control de dicho margen se hace a través de otro parámetro de regularización indicado habitualmente como C.

Clasificación Multiclase

Como se dijo anteriormente, las SVM pueden abordar problemas de clasificación binaria. En el caso de existir más de una clase (como en este trabajo, cada instrumento es una clase), debe hacerse una extensión del algoritmo original de las SVM para que pueda utilizarse en una clasificación multiclase.

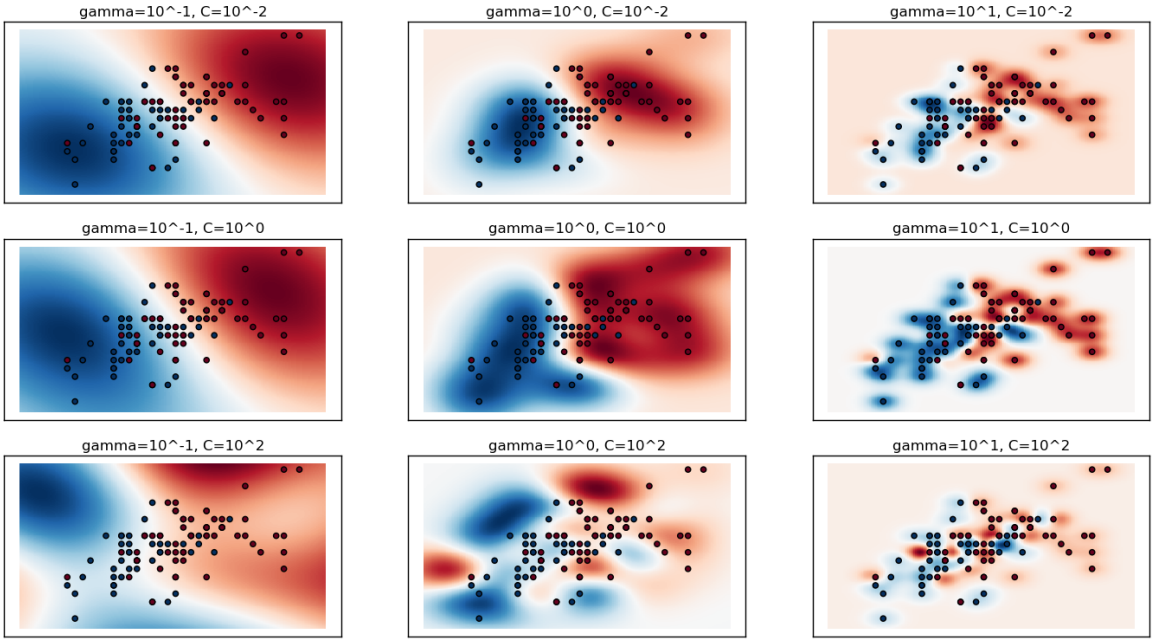
Una de las formas de resolver este problema es repetir la clasificación binaria para todas las clases entre sí y luego asignar puntajes según los resultados para separar las clases. Existen dos métodos comunes que implementan este mecanismo, y se conocen como una contra todas (one versus all) y una contra una (one versus one), refiriéndose a las clases. Se describirá brevemente la última de ellas por ser la utilizada en este trabajo.

En el formato una contra una, el algoritmo clasifica cada clase contra cada una de las demás y compara cuál de las clasificaciones ocurre más veces asignándole un puntaje. La que obtiene más puntaje es la que se define como clase.

Parámetros de Regularización γ y C.

La regularización es una operación que introduce información adicional para poder resolver problemas mal definidos (*ill-possed*) y/o prevenir el sobreentrenamiento en sistemas de aprendizaje autónomo. Para la implementación de este trabajo hay dos parámetros de regularización γ y C. El parámetro γ, puede pensarse [51] como la inversa del radio de influencia de las muestras seleccionadas por el modelo de soporte vectorial y C como un intercambio entre la posibilidad de clasificar mal un ejemplo de entrenamiento y la sencillez de la superficie de decisión. Si C es bajo (margen angosto), esta superficie es suave, mientras que si C es elevado (margen grande), el algoritmo intenta clasificar todos los puntos correctamente. Además, un C elevado permite al modelo seleccionar más muestras como vectores de soporte. En la Figura 2.7 pueden verse la influencia de estos parámetros para la clasificación de un ejemplo.

Figura 2.7: Evaluación de la frontera de decisión para distintos C y γ, imagen tomada de la documentación de Scikit-learn [56].



Capítulo 3

Implementación

3.1. Bases de datos

Para la construcción y evaluación de algoritmos de clasificación automática, es recomendable el uso de bases de datos que contengan un muestreo amplio del universo de casos a clasificar [58]. Habitualmente se utilizan tres bases de datos; la primera, generalmente la más grande, se denomina 'base de entrenamiento' y se utiliza para calcular los criterios matemáticos de proximidad y agrupar las muestras en clases. A la segunda se la llama 'base de validación cruzada' y cumple la función de evaluar los parámetros de clasificación extraídos de la base de datos de entrenamiento. Finalmente, la tercera es la 'base de prueba' con la que se evalúa el desempeño global del algoritmo. En la etapa de validación pueden modificarse distintos parámetros del algoritmo para mejorar su desempeño.

Existen distintas formas de configurar estas tres bases de datos. Para esta tesis se utilizó el método conocido como validación cruzada simple (*hold-out cross validation*). La bibliografía consultada indica que este divide la base de datos en 3 partes, teniendo en cuenta que el porcentaje de las muestras totales en la base de entrenamiento esté alrededor del 75 % y que la base de prueba sea la más pequeña de las 3. A su vez, la división de esta base de datos debe ser realizada nuevamente para cada uso del sistema.

En la implementación de esta tesis se corroboró empíricamente que esta separación de manera aleatoria no siempre funciona correctamente. Esto se debe al tamaño de la base de datos y la cantidad de clases. La separación aleatoria puede dejar alguna base de datos sin muestras de una clase, generando errores para esa clase. Por consiguiente, la división aleatoria debió realizarse para cada clase en particular en partes iguales, sin tomar ningún otro recaudo especial.

En esta etapa del trabajo se utiliza una base de datos que posee muestras de notas monofónicas. En esta, cada archivo corresponde a una nota de un instrumento. Por otro lado, con la finalidad de más adelante ampliar el horizonte de aplicaciones del sistema, se recopilaron bases de datos que incluyen piezas musicales con predominancia melódica y presencia de un único instrumento.

3.2. Confección de la Base de Datos

Para la confección de la base de datos se consultó en primer lugar a la Sociedad Internacional para la Recuperación de Información de la Música (ISMIR) [59]. Esta posee una serie de bases de datos que han sido recopiladas con la finalidad de impulsar investigaciones académicas y son reconocidas por su alta calidad. A su vez, se encontraron otras bases de datos en investigaciones

sobre MIR[1]. Además, se sumó una base de datos de pianos utilizada recientemente para la estimación automática de frecuencias fundamentales de pianos acústicos [60]. Finalmente, el sistema cuenta con las muestras necesarias para caracterizar 21 instrumentos distintos.

Las bases de datos recopiladas son:

- MIS: Es la base de datos de Muestras de Instrumentos Musicales de la Universidad de Iowa. Está dirigida por Lawrence Fritts, director del centro de Estudios Musicales Electrónicos [61].
- UMA: Es una base de datos de sonidos de piano creada en la Universidad de Málaga para estimular el desarrollo de algoritmos de transcripción de música monofónica y/o polifónica [62].
- IRMAS: Es una base de datos para el reconocimiento de instrumentos en señales de audio musicales. Fue creada por el grupo de tecnología musical de la Universitat Pompeu Fabra [63].
- SMD Western Music: Es una base de datos de piezas de música clásica occidental interpretadas por estudiantes o miembros del staff de la Hochschule für Musik Saar [64, 65].

3.2.1. Extensión de la base de datos

En la evaluación de algoritmos automáticos que utilizan bases de datos, es una práctica común generar nuevas muestras a partir de las existentes mediante transformaciones matemáticas [58]. En este caso se amplió la base de datos generando muestras con distintos efectos de reverberación. Con esta finalidad se obtuvo una base de datos de respuestas al impulso de distintos espacios generada por la universidad de Aachen, Alemania llamada 'AIR' [66]. Se realizó una convolución entre estas respuestas al impulso y las muestras originales para duplicar la cantidad de señales utilizadas por el sistema.

3.3. Caracterización de la base de datos

Las bases de datos utilizadas para la realización de esta tesis son 'MIS' y 'UMA'. De esta forma, la clasificación se realiza entre 21 instrumentos musicales acústicos distintos que pueden consultarse en la tabla 3.1. Como puede observarse en esta tabla, la base no posee la misma cantidad de muestras para cada clase por lo que puede decirse que está 'desbalanceada'.

Si bien la ISMIR [59] sugiere para la clasificación automática de instrumentos musicales utilizar las bases de datos 'IRMAS' y 'SMD Western Music', estas exceden el alcance de esta tesis dado que ambas están formadas por fragmentos de piezas musicales con predominancia de algún instrumento. Se prefirió por consiguiente utilizar bases de datos de sonidos monofónicos e individuales (en lugar de frases musicales).

3.3.1. MIS

Desde 1997, la Universidad de Iowa posee una base de datos de Muestras de Instrumentos Musicales [61]. El director del departamento de Estudios Musicales Electrónicos, Lawrence

[1]En particular, se encontró una recopilación extensa de bases de datos en el siguiente sitio web visitado por ultima vez el 31 de Octubre del 2015: `//grh.mur.at/sites/default/files/mir_datasets_0.html`

Tabla 3.1: Se listan los instrumentos musicales utilizados por el algoritmo principal del software. Estos son en los que se trabajó principalmente y, por consiguiente, en los que se desempeña mejor el algoritmo.

Familia	Sub-Familia	Tipo	Instrumento	Muestras
Cordófono		Frotado	Violín	364
			Viola	400
			Violoncello	390
			Contrabajo	415
		Pulsado	Guitarra	49
		Percutido	Piano	1550
Aerófono	Madera	Bisel	Flauta	152
			Flauta Alto	72
			Flauta Baja	76
		Lengüeta Simple	Clarinete en Eb	77
			Clarinete en Bb	92
			Clarinete Bajo	91
			Saxofón Soprano	128
			Saxofón Alto	128
		Lengüeta Doble	Oboe	70
			Fagot	77
	Bronce	Con Válvulas	Trompeta	142
			Corno	75
			Trombón Tenor	64
			Trombón Bajo	51
			Tuba	73

Fritts, se encarga de supervisarla. Esta puede descargarse desde el sitio web provisto por la universidad y utilizarse sin restricciones. Ha sido utilizada en más de 270 artículos de investigación y libros.

Hasta el 2011, las grabaciones se hacían con un micrófono Neumann KM 84 de tipo condensador con un patrón polar cardiode. Además, los archivos se grababan con una resolución de 16 bits y 44.1 kHz de frecuencia de muestreo. El objetivo fue grabar nota por nota de las escalas cromáticas de cada instrumento con tres niveles de intensidad en la ejecución. Al querer representar la estructura dinámica del instrumento, se preservó el nivel de los preamplificadores en cada sesión de grabación. Además, algunos instrumentos fueron interpretados con diferentes técnicas como puede ser pizzicato, vibrato y sin vibrato.

A partir del 2011, las grabaciones se hicieron con tres micrófonos Earthworks QTC-40 configurados en una formación de Decca Tree. Esta formación se dispuso a 1.5 metros del instrumento y con 30 centímetros de separación entre las cápsulas. Con los micrófonos de la izquierda y la derecha se generaron archivos estéreo con 24 bits de resolución y 96 kHz de frecuencia de muestreo. El micrófono central en cambio fue remuestreado a 44.1 kHz y 16 bits para generar archivos consistentes con las grabaciones anteriores al 2011.

Con la excepción del piano, todos los instrumentos fueron grabados en una cámara anecóica en el Centro Wendell Johnson para el Habla y la Escucha. Esta posee 765 metros cúbicos (9.15 x 9.15 x 9.15 metros) y está aislada del resto del edificio. El diseño de esta aislación es del tipo *box in a box*. Se asegura así que el comportamiento anecóico de la sala se extienda en las frecuencias graves hasta los 60 Hz.

3.3.2. UMA

Esta base de datos de piano fue generada por la Universidad de Málaga (UMA) en el 2007 [62]. La finalidad fue proporcionar una herramienta para impulsar la transcripción de muestras polifónicas de piano a un formato legible. Con el paso del tiempo, la base de datos creada excedió las expectativas gracias al interés de la comunidad internacional en este tipo de recurso. Fue así que en el 2011 la editorial Springer decidió distribuir el trabajo. Los autores además proporcionan un análisis ingenieril del concepto de armonía, para ayudar a los investigadores en el campo de MIR.

Las grabaciones se hicieron con una frecuencia de muestreo de 44.1 kHz y se cuantizaron a 16 bits. El piano utilizado para las grabaciones fue un Kawai CA91. Por otra parte, las ejecuciones de las notas se realizaron con tres intensidades *Forte, Mezzo y Piano* y tres estilos *Normal, Staccato y Pedal*.

3.3.3. IRMAS

Esta base de datos fue creada por el Grupo de Tecnologia Musical (MTG) de la Universitat Pompeu Fabra de Barcelona. Incluye extractos de señales de audio musicales con anotaciones indicando el instrumento predominante en las mismas. Fue utilizada para la evaluación por Bosch, Janer, Fuhrmann y Herrera [67] en el 2012.

IRMAS fue concebida para ser usada en el entrenamiento y prueba de métodos para el reconocimiento automático de instrumentos predominantes en señales de audio musicales. Para la presente tesis, se evaluó sólo la sección de entrenamiento de esta base de datos. Esta, al ser más general, coincide con la problemática que ha sido propuesta. Los instrumentos que contiene son: Violoncello, clarinete, flauta, guitarra acústica, guitarra eléctrica, órgano, piano, saxofón, trompeta, violín y voz cantada. Esta base de datos se derivó de la compilada por Ferdinand Fuhrmann [68].

Información para el entrenamiento:

Se compone de 6705 archivos de audio con 16 bit de resolución en formato estéreo muestreado a 44.1kHz. Hay extractos de 3 segundos de más de 2000 grabaciones distintas. Además, contiene archivos de textos con notas que indican el instrumento predominante de cada extracto.

Esta base de datos incluye música de la década actual y de varias décadas del siglo pasado, por lo que la calidad del audio no es homogénea. También cubre una gran variación en los tipos de instrumentos musicales, los intérpretes, las articulaciones y el tipo de producción. Adicionalmente, el Grupo de Tecnologia Musical intentó maximizar el número de géneros presentes en la base de datos para evitar extraer información relacionada con un género en particular [63].

Información para el prueba:

Se compone de 2874 archivos de audio con 16 bit de resolución en formato estéreo muestreado a 44.1kHz. Además, contiene archivos de texto para cada extracto sonoro. Más de un instrumento puede estar anotado en cada extracto.

3.3.4. SMD Western Music

Saarland Music Data (SMD) es la base de datos generada por la Escuela para la música de Saarland Alemania [64] con el objetivo de impulsar investigaciones en el campo de la Recuperación de Información de la Música. Esta escuela alemana además trabajó con el Instituto Max-Planck de Informática (MPII) [69] para completar este desarrollo. En sus propias palabras, el objetivo fue desarrollar una plataforma donde los científicos de la computación y los músi-

cos pudieran explorar y discutir la aplicación de métodos computacionales para el análisis y la educación de la música [65].

Esta base de datos está compuesta de 200 piezas de música clásica. Algunas fueron interpretadas con un único instrumento, pero la mayoría han sido grabadas con varios instrumentos en simultáneo. La base de datos posee además anotaciones indicando qué instrumentos existen en cada archivo junto con información útil para el estudio del mismo.

3.4. Python

Para la implementación del algoritmo de identificación, se trabajó principalmente con Python. Este es un lenguaje de programación interpretado [70]. Su filosofía de diseño enfatiza el código legible y su sintaxis permite a los programadores expresar conceptos usando una menor cantidad de lineas de código que en otros lenguajes como C++ o Java [71, 72]. Es un lenguaje multiparadigma ya que soporta programación orientada a objetos, imperativa y funcional. Tiene asignaciones dinámicas de tipo para las variables, gestión automática de memoria y una extensa biblioteca estándar [73]. Es administrado por la *Python Software Foundation*. Posee una licencia de código abierto, denominada *'Python Software Foundation License, 1'* que es compatible con la Licencia pública general de GNU a partir de la versión 2.1.1. Los intérpretes de Python están disponibles en la mayoría de los sistemas operativos, por lo que el código puede ser utilizado en una gran variedad de sistemas. Además, los programas escritos en Python pueden ser empaquetados en archivos ejecutables, permitiendo su utilización sin necesidad de instalar un interprete.

3.4.1. Numpy y Scipy

Para la obtención de los descriptores estadísticos se recurrió a dos paquetes abiertos de Python llamados NumPy [74] y SciPy [50]. Estos han sido desarrollados por una comunidad de científicos y su validación está en la gran cantidad de investigaciones que los utilizan.

3.4.2. Scikit-learn

Para los algoritmos de clasificación, se utilizó Scikit-learn. Esta es una libreria *open-source* de *machine learning* para Python [51]. Posee varios algoritmos de clasificación, regresión y agrupamiento entre los cuales incluye Máquinas de Soporte Vectorial, Selvas Aleatorias ('Random Forests'), k-means, Redes Neuronales y agrupamiento espacial basado en densidad de aplicaciones con ruido (DBSCAN). Está diseñada para trabajar en conjunto con las librerías de NumPy [74] y SciPy [50].

3.5. Essentia

Para la obtención de los descriptores acústicos se recurrió a la librería abierta de C++ desarrollada por la Universidad Pompeu Fabra de Barcelona llamada Essentia [75]. La misma fue generada para impulsar trabajos del campo de la recuperación de información de la música.

Essentia presenta más de 90 descriptores distintos ya probados por referentes del campo. Además, puede ser utilizada con lenguajes de alto nivel como Python lo que facilita la modularización de la implementación.

Capítulo 4

Descripción del sistema y de sus ajustes técnicos

El sistema se divide en tres partes. En la primera se procesa la base de datos y se obtienen los descriptores. Con estos descriptores se conforma un vector para cada muestra. En la segunda parte se toman estos vectores y se realiza el entrenamiento del algoritmo de clasificación.

En este capítulo, se presentan estas dos partes. La última consiste del clasificador ya entrenado que recibe señales desconocidas y será introducida en el próximo capítulo.

4.1. Procesamiento de las señales

Para la extracción de los descriptores el primer paso consiste en identificar la porción del archivo de audio que es útil. Esto se realiza mediante el algoritmo de detección de silencio presentado en la sección 2.1.3. A su vez, la señal se divide punto a punto por su máximo valor para garantizar que todas las señales tengan valor máximo unitario. Los descriptores son extraídos de esta nueva señal normalizada y sin silencio (ver Figura 4.1).

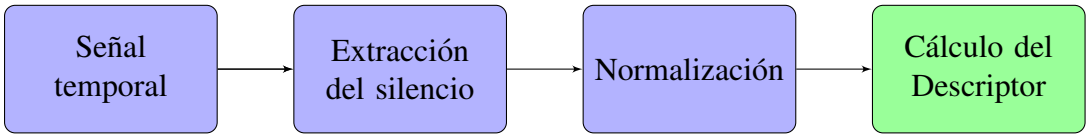


Figura 4.1: Diagrama en bloques de la preparación de una señal temporal para la extracción de los descriptores.

Una vez obtenidos los descriptores, el tratamiento posterior varía para cada uno. Para los MFCC, además de su valor se toma en cuenta su desviación estándar, su varianza, su kurtosis y su skewness en el dominio de la frecuencia, y las primeras dos derivadas con respecto al tiempo. Por otro lado, a cada uno de estos vectores de valores se les realiza un promediado para obtener un único valor (en el caso de los MFCC y sus derivadas, se obtiene un valor por banda de frecuencia). Se debe obtener un único valor por descriptor dado que el algoritmo de clasificación debe entrenarse con la misma cantidad de descriptores para cada muestra. Puede observarse en las tablas 4.1 y 4.2, un ejemplo de los valores obtenidos al caracterizar una nota tocada por un Cello. Antes de utilizar estos descriptores con el algoritmo de clasificación, se los normaliza siguiendo la ecuación 2.3.

Tabla 4.1: Ejemplo de los valores de descriptores obtenidos para un Cello tocando en pizzicato una nota A3. Los números presentados aquí no están normalizados.

Descriptor	Valor	Descriptor	Valor
Inarmonicidad	0.24	Desviación Estándar Frecuencial de MFCC	262.5
$\frac{d(Inarmonicidad)}{dt}$	5e-3	Varianza Frecuencial de las bandas de MFCC	69e3
$\frac{d^2(Inarmonicidad)}{dt^2}$	-1e-4	Skewness Frecuencial de las bandas de MFCC	-2.6
Tiempo de ataque Logarítmico	-1.99	Kurtosis Frecuencial de las bandas de MFCC	6.35
Desviación Estándar Temporal	0.07	MFCC	ver Tabla 4.2
Varianza Temporal	5.7e-3	$\frac{d(MFCC)}{dt}$	ver Tabla 4.2
Skewness Temporal	1.19	$\frac{d^2(MFCC)}{dt^2}$	ver Tabla 4.2
Kurtosis Temporal	33.8	Energía	928

Tabla 4.2: Ejemplo de los valores obtenidos para MFCC y sus primeras dos derivadas temporales en una nota A3 de un Cello tocado en pizzicato.

MFCC												
Banda	0	1	2	3	4	5	6	7	8	9	10	11
MFCC	-997	146	61.8	19.2	1.79	-7.16	-12.6	-20	-27.0	-29.2	-29.9	-29.3
$\frac{d(MFCC)}{dt}$	0.53	0.31	-0.17	-0.06	0.02	-0.06	-0.01	0	-0.05	-0.02	-0.01	-0.01
$\frac{d^2(MFCC)}{dt^2}$	-1.39	-0.94	0.24	0.07	-0.10	0.12	0.05	0	0.12	0.06	0.01	0.03

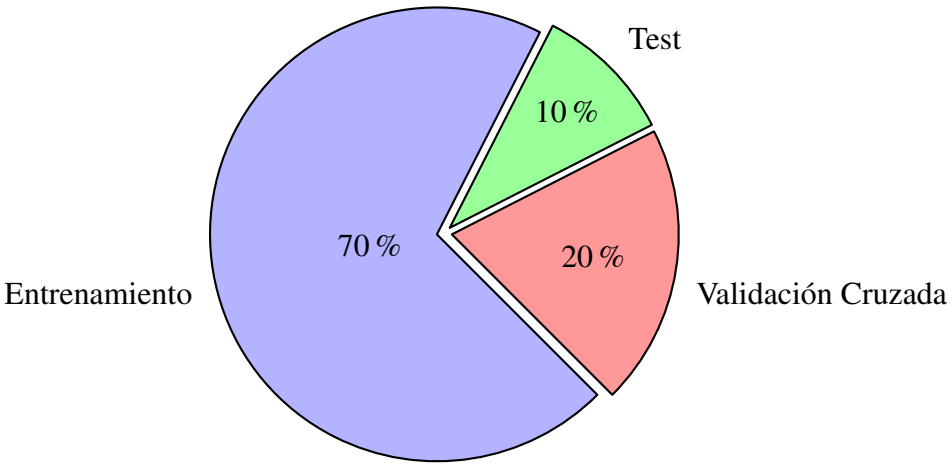
4.2. División de la base de datos

En la sección 3.1 se presentó la necesidad de generar tres bases de datos para la correcta implementación del sistema. Estas son la base de 'Entrenamiento' con un 70 % de las muestras, la de 'Validación Cruzada' con un 20 % y la de prueba o 'Test' con un 10 % (ver Figura 4.2). Para esta separación se generó un algoritmo que descompone a la base de datos original en tres bases nuevas. Originalmente, este programa simplemente tomaba aleatoriamente de la base original la cantidad de muestras necesarias para cada una de las nuevas bases. Este enfoque presentó problemas dado que cada clase no tenía una representación adecuada en todas las bases. Por consiguiente, el sistema final realiza una separación aleatoria por clase en cada una de las bases. De esta forma, se asegura que todas las bases posean el mismo porcentaje de muestras de cada instrumento.

4.3. Medición del desempeño

Para analizar el desempeño del algoritmo de clasificación, se debe establecer un criterio cuantitativo de medición. Cabe aclarar, que dado que la base de datos utilizada está desbalancea-

Figura 4.2: División de la base de datos.



da (ver sección 3.3) utilizar simplemente la precisión (el porcentaje de acierto) sería una mala elección. A modo de ejemplo, si la base de datos estuviera compuesta en un 95 % por muestras de piano y el algoritmo simplemente clasificara como piano a todas las muestras, la precisión sería alta aunque el sistema no esté funcionando adecuadamente. Siguiendo las sugerencias hechas por Andrew Ng [58] para problemas de este tipo, se utiliza la medida-F ($F_1\ Score$) para evaluar el desempeño. Como se presenta a continuación, la medida-F se calcula individualmente para cada clase. Para obtener un valor único, se realiza el promedio del valor obtenido para cada clase. A este enfoque se lo conoce como macro-promediado y trata a todas las clases por igual, por lo que no se ve afectado por la composición de la base de datos [76].

4.3.1. Medida-F, precisión y exhaustividad

Las medidas de desempeño utilizan las siguientes definiciones para relacionar la clasificación predicha respecto de la muestra real (Ver tabla 4.3):

- Positivo Correcto: La muestra era de la clase buscada y fue clasificada correctamente.
- Negativo Incorrecto: La muestra era de la clase buscada y fue clasificada como miembro de otra clase.
- Positivo Incorrecto: La muestra no era de la clase buscada y fue clasificada como miembro de la clase buscada.
- Negativo Correcto: La muestra no era de la clase buscada y fue clasificada como miembro de otra clase.

Tabla 4.3: Clasificación realizada y clasificación esperada.

		Clase Predicha	
		Sí	No
Clase real	Sí	Positivo Correcto	Negativo Incorrecto
	No	Positivo Incorrecto	Negativo Correcto

Finalmente, se utilizan las ecuaciones 4.1, 4.2 y 4.3 para calcular el valor del desempeño del clasificador en cada clase. El valor del desempeño total es un promedio de los desempeños obtenidos.

$$Precisión = \frac{Positivos\ correctos}{Positivos\ correctos + Positivos\ incorrectos} \tag{4.1}$$

$$Exhaustividad = \frac{Positivos\ correctos}{Positivos\ correctos + Negativo\ incorrecto} \tag{4.2}$$

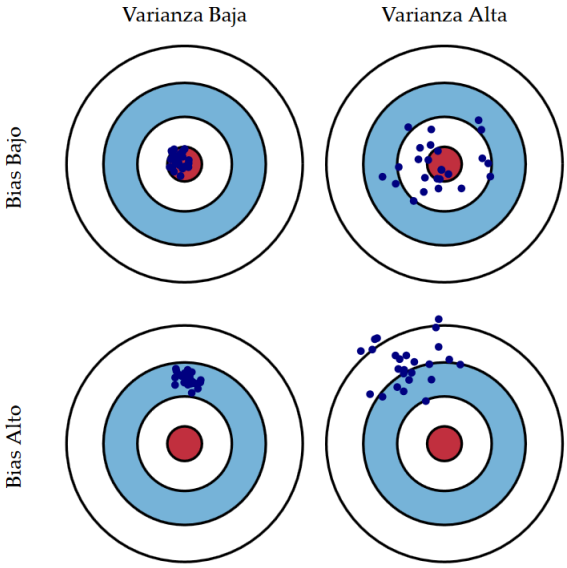
$$F_1 = 2 * \frac{Precisión * Exhaustividad}{Precisión + Exhaustividad} \tag{4.3}$$

4.3.2. Sobre-entrenamiento, Bias

En el área de la recuperación de información es común utilizar la medida-F para medir el desempeño [33]. Sin embargo, el desempeño debe medirse en dos bases de datos distintas. Una es la base de datos de entrenamiento y la otra es la de evaluación. Esta última se selecciona entre la de validación cruzada y la testeo de acuerdo a la etapa en la que se encuentre el desarrollo siguiendo la lógica presentada en la sección 3.1.

En el ajuste del modelo de aprendizaje se debe atender a dos parámetros: la Varianza y el Sesgo (Bias). En la figura 4.3 tomada del trabajo de Scott [77] puede verse una descripción gráfica de estos dos parámetros. La varianza indica el nivel de consistencia que posee el clasificador; un nivel bajo indica que la consistencia será alta y la clasificación será pareja. Por consiguiente, sucesivas muestras similares deberían obtener clasificaciones parecidas. El Bias indica la diferencia entre la clasificación promedio obtenida y la clasificación promedio esperada. Un nivel bajo indica que en promedio la clasificación es correcta.

Figura 4.3: Comparación de clasificadores según su varianza y su bias. Cada punto azul representa un sistema entrenado con información correspondiente al mismo fenómeno. La predicción correcta en cada caso es el área roja.



Es habitual encontrar que hay una relación entre el Bias y la Varianza. Si un modelo es demasiado ‘simple’ y tiene pocos parámetros, entonces puede que tenga un Bias elevado y una Varianza baja; pero si es demasiado ‘complejo’ y tiene muchos parámetros, entonces probablemente tenga una Varianza alta y un Bias bajo [58].

Capítulo 5

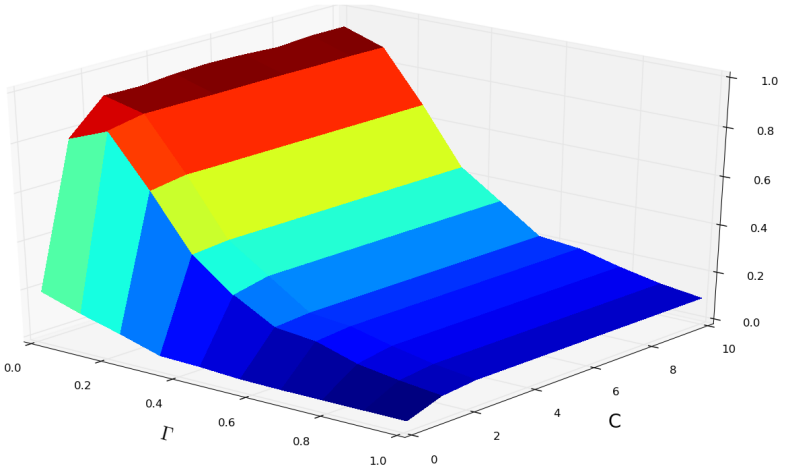
Resultados y discusión

5.1. Selección del parámetro de regularización 'C' y 'Γ'

Al entrenar un algoritmo de Máquinas de Soporte Vectorial (SVM) con un kernel gaussiano se deben considerar los parámetros de regularización: C y Γ [51]. Estos parámetros fueron presentados en la sección 2.2.2. C es común a todos los kernels de las SVM e indica la complejidad de la frontera de decisión. Γ es un parámetro específico del kernel gaussiano que define cuánta influencia tiene cada ejemplo del entrenamiento. Un valor pequeño de C hace la frontera más sencilla (recta en dos dimensiones), mientras que un valor de C elevado intenta clasificar todos los ejemplos correctamente. En cuanto a Γ, cuanto mayor sea el valor más cerca deben estar los demás ejemplos para ser afectados.

Para encontrar los valores adecuados para los parámetros se realizó una búsqueda de grilla. De esta forma, el sistema fue entrenado con una gran variedad de pares de valores, evaluando su desempeño en la base de Validación Cruzada. Para la primera aproximación se combinó un vector con 10 valores de C (entre 0.1 y 10) con otro de 10 valores de Γ (entre 0.01 y 1). Los resultados de este evaluación pueden verse en la Figura 5.1. Como puede observarse, al aumentar el Γ más allá 0.2, el desempeño del algoritmo decae abruptamente. También se destaca que para los valores de C menores a 1 el sistema tampoco obtiene resultados aceptables.

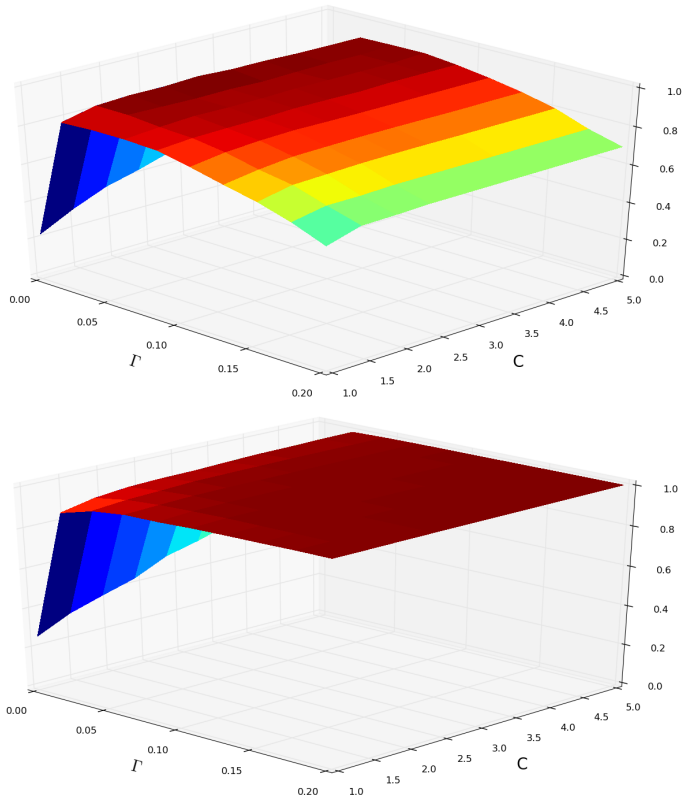
Figura 5.1: Primera evaluación de los parámetros de regularización. Se evaluó C entre 0.1 y 10, y Γ entre 1e-2 y 1. El sistema fue entrenado con la base de entrenamiento para cada par de parámetros. Luego, se utilizó la medida-F para evaluar el error sobre la base de Validación Cruzada.



A partir de estos resultados preliminares, se decidió entrenar el algoritmo con diez valores de C entre 1 y 5 y diez valores de Γ entre 1e-3 y 0.2 (ver Figura 5.2). En esta etapa es importante

también considerar los valores de la medida-F para la clasificación de la base de Entrenamiento. El mejor clasificador no será el que menos error posea en la base de Validación Cruzada, sino el que además tenga un error similar en la base de entrenamiento. Esto se debe a que un error mucho más bajo en la base de Entrenamiento indicaría que el clasificador tiene un Bias bajo pero una Varianza elevada y esta siendo sobre-entrenado.

Figura 5.2: Segunda evaluación de los parámetros de regularización. Se evaluó C entre 1 y 5, y Γ entre 1e-3 y 2e-1. El sistema fue entrenado con la base de entrenamiento para cada par de parámetros. Luego, se utilizó la medida-F para evaluar la clasificación de la base de Validación Cruzada (arriba) y la base de entrenamientos (abajo).



Teniendo en consideración todos los factores y observando los resultados, se comprueba que los valores más elevados de la medida-F en la base de validación cruzada se dan para $\Gamma = 2, 3e - 2$. Sin embargo, el parámetro C no se puede definir con claridad. Por consiguiente, se evaluó C cómo única variable. Los resultados pueden consultarse en la Figura 5.3. De esta forma se comprueba que el resultado óptimo se da con $C = 2, 77$.

5.2. Cantidad de muestras en la base de datos utilizadas por el algoritmo de clasificación

Una vez seleccionados los parámetros de regularización del algoritmo de aprendizaje, se procede a evaluar la incidencia de distintos factores en el desempeño global del sistema. En un primer lugar, se evaluó la cantidad de muestras de la base de datos. Para esto, se generó un algoritmo que reduce en un porcentaje la cantidad de muestras de cada clase de la base de datos. De esta forma, se obtuvo una serie de nuevas bases de datos con la misma relación de muestras entre clases que la original, pero con una cantidad total de señales menor. Con estas, se entrenó al sistema para clasificar la base de entrenamiento y la de validación cruzada. Utilizando el procedimiento descripto se generó el gráfico expuesto en la Figura 5.4.

Figura 5.3: En última instancia, una vez elegido Γ, se evaluó el desempeño del sistema en ambas bases de datos. Puede observarse claramente un pico en $C = 2,77$. Además, en este punto la relación entre el desempeño en ambas bases es cercano al máximo.

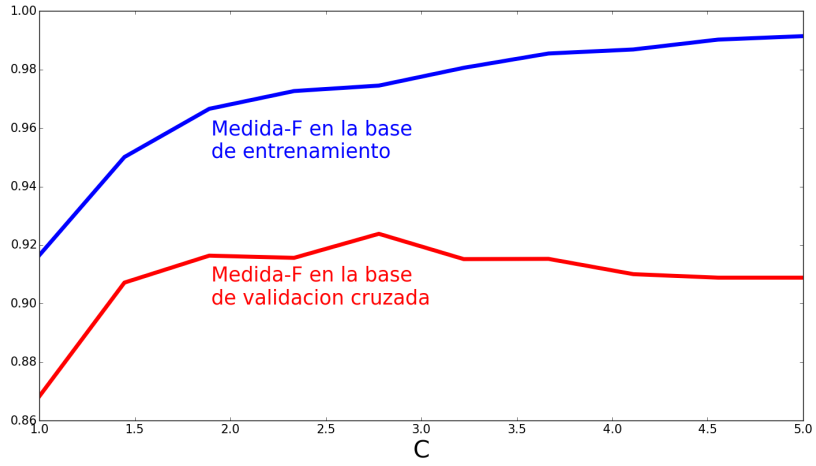
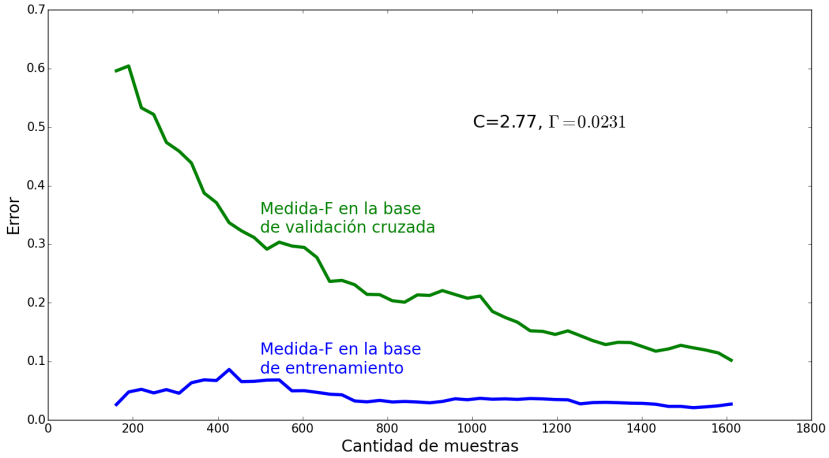


Figura 5.4: Evaluación del error en la clasificación del sistema según la cantidad de muestras.

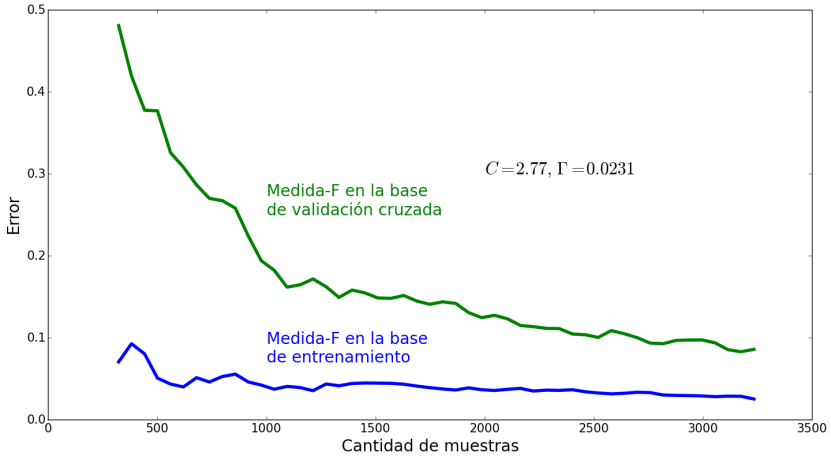


A partir de estos resultados, se concluyó que el sistema puede beneficiarse de un aumento en la base de datos. Esto se debe a que la tasa de error disminuye progresivamente pero no llega a estancarse. Por consiguiente, se realizaron modificaciones a la base de datos original para aumentar la cantidad total de muestras siguiendo el procedimiento descripto en 3.2.1. Luego, se realizó nuevamente el mismo análisis generando el gráfico expuesto en la Figura 5.5.

De este gráfico, no se puede concluir que aumentar la base de datos mediante transformaciones en la misma haya beneficiado al sistema. El sistema no solo no llega a estancarse en un valor de desempeño, sino que tampoco se aprecia una disminución significativa del error total.

Las Figuras 5.4 y 5.5 surgen de una única evaluación del algoritmo por punto. De estos no se puede concluir si agregar las muestras con reverberación a la base de datos generó una mejora en el desempeño del sistema. Por consiguiente, se evaluó el promedio de 100 evaluaciones para el total de muestras. Así, se observó que sin reverberaciones, el desempeño F es de 0.9058 y con reverberaciones es de 0.9194. Por lo tanto, se concluye que agregar las reverberaciones disminuye el error del sistema.

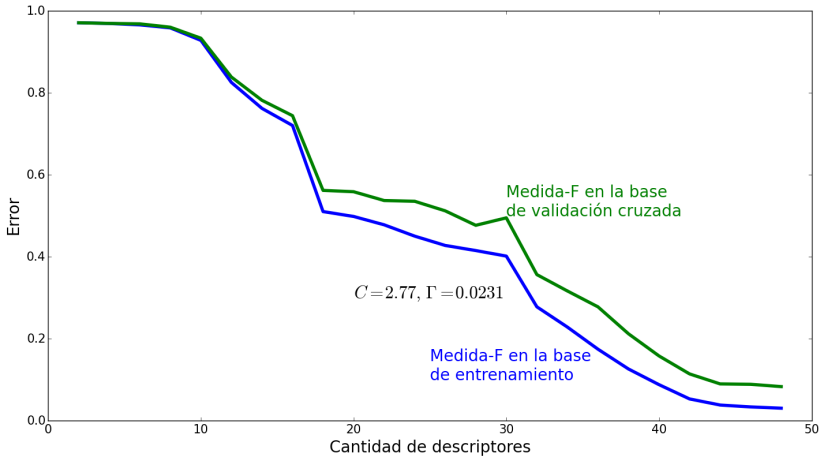
Figura 5.5: Evaluación del error en la clasificación del sistema según la cantidad de muestras, usando transformaciones en la base de datos.



5.3. Cantidad de Descriptores utilizados por el algoritmo de clasificación

Para continuar con el análisis del sistema, se evaluó la cantidad de descriptores utilizados. Para esto, se entrenó y evaluó el sistema variando el vector de descriptores. Se comenzó utilizando solo 2 descriptores y se fueron adicionando de a 2, generando el gráfico expuesto en la Figura 5.6. Cabe aclarar además que cada punto del gráfico representa el promedio de diez evaluaciones con esa cantidad de descriptores. Al evaluar el promedio luego de entrenar varias veces el sistema, se disminuye el error en el análisis del sistema.

Figura 5.6: Evaluación del error en la clasificación del sistema según la cantidad de descriptores.



Para analizar los resultados de la Figura 5.6, se puede consultar en la tabla 5.1 la posición en la cual se agrega al sistema cada descriptor. Puede verse que entre los 18 y 30 descriptores el desempeño del sistema deja de mejorar como lo venía haciendo y se estanca. Esta zona se corresponde con la inclusión de la segunda derivada temporal de los coeficientes de MFCC al vector de descriptores. Se ve de todas formas que para estos 12 descriptores el error de la medida-F baja 0.11. Por otro lado, al final del gráfico se observa una nueva zona donde el desempeño se estanca a partir de los 42 descriptores. Esto se corresponde con la inclusión

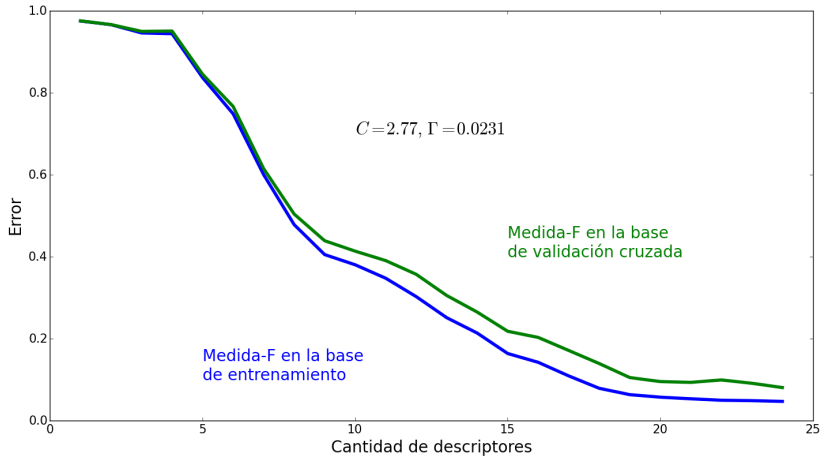
de la primer derivada temporal de los coeficientes de MFCC. Estos descriptores ocupan las posiciones 38 en adelante por lo que se ve que para las frecuencias altas la mejora introducida por 7 descriptores es de 0.02.

Tabla 5.1: Referencia de la posición en la que se agregan los descriptores al sistema. Esta tabla complementa la Figura 5.6.

Descriptor	Posición	Descriptor	Posición
Inarmonicidad	2	Desviación Estándar Frecuencial de MFCC	32
$\frac{d(Inarmonicidad)}{dt}$	3	Varianza Frecuencial de las bandas de MFCC	19
$\frac{d^2(Inarmonicidad)}{dt^2}$	37	Skewness Frecuencial de las bandas de MFCC	1
Tiempo de ataque Logarítmico	33	Kurtosis Frecuencial de las bandas de MFCC	17
Desviación Estándar Temporal	35	MFCC	5-16
Varianza Temporal	34	$\frac{d^2(MFCC)}{dt^2}$	38-49
Skewness Temporal	4	$\frac{d(MFCC)}{dt}$	20-31
Kurtosis Temporal	36	Energía	18

Dado el análisis anterior, se evaluó el error en la clasificación del sistema sin los descriptores donde el sistema parecía estancarse (las derivadas de los MFCC). Los resultados se exponen en la Figura 5.7. Puede observarse que la nueva curva parece tener una caída más regular, donde el desempeño en un principio mejora en un mayor grado para cada descriptor y luego progresivamente va llegando a un nivel de error mínimo. Cabe destacar que este nivel de error está en el mismo orden del que existía utilizando los descriptores que fueron sustraídos para este análisis. Por consiguiente, dado que se removieron casi la mitad de los descriptores, el sistema resulta mucho más compacto y eficiente.

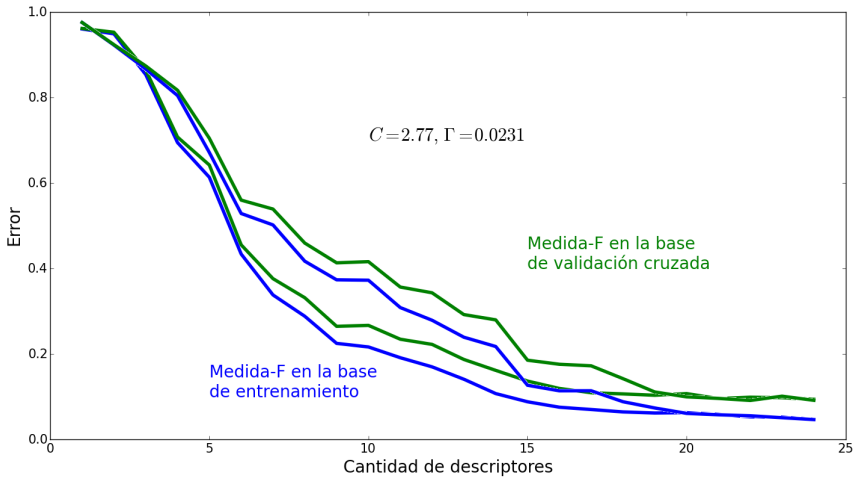
Figura 5.7: Evaluación del error en la clasificación del sistema sin las derivadas de los MFCC.



Se evaluó también la causa de la pendiente inicial del error en las Figuras 5.6 y 5.7. En estas, el error disminuye lentamente hasta el cuarto descriptor. Para evaluar si este efecto es una característica propia del algoritmo de identificación se repitió el proceso con el que se generaron estas gráficas, pero ordenando los descriptores de manera aleatoria. El resultado puede consultarse en la Figura 5.8, donde se superponen dos curvas generadas. Puede corroborarse que el

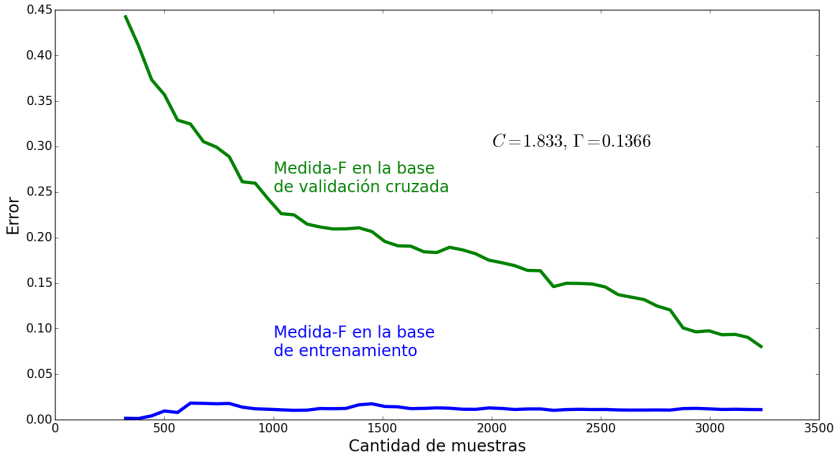
error inicial no está presente de la misma forma. Por consiguiente, no puede afirmarse que el algoritmo de identificación sea el responsable de este comportamiento.

Figura 5.8: Evaluación del error en la clasificación del sistema ordenando los descriptores de manera aleatoria.



Como se eliminaron varios descriptores, los parámetros C y Γ fueron calculados nuevamente. Este trabajo se realizó siguiendo el análisis presentado en la sección 5.1 para obtener los parámetros óptimos para el sistema. Finalmente los valores obtenidos fueron C=1.833 y Γ=0.1366. Con estos valores se repitió la prueba realizada en la sección 5.2 para ver cómo se desempeña el sistema según la cantidad de muestras. El resultado se presenta en la figura 5.9. Puede afirmarse a partir de esta que aumentar la cantidad de muestras de la base de datos mejoraría el desempeño del sistema.

Figura 5.9: Evaluación del error en la clasificación del sistema según la cantidad de muestras usando los parámetros corregidos del sistema.

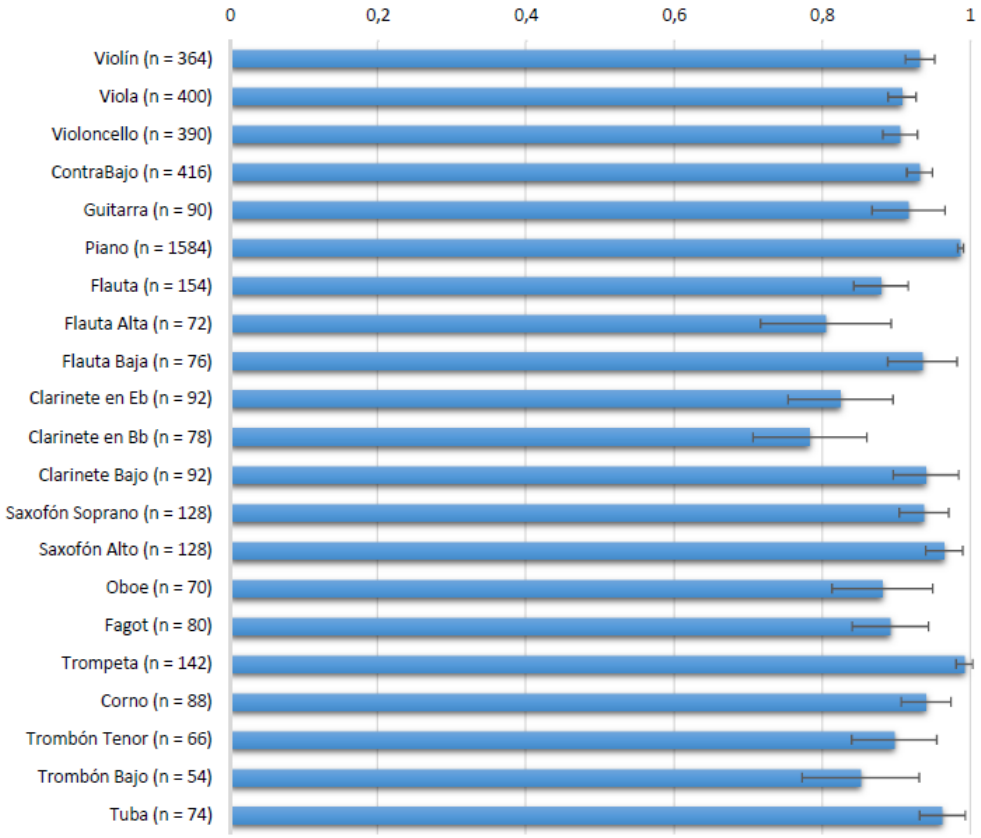


5.4. Clasificación del sistema para cada instrumento

Tras realizar los ajustes necesarios para el buen funcionamiento general del sistema con la base de datos de validación cruzada, se pasó a analizar el desempeño para la clasificación de

cada instrumento en la base de datos de Prueba. Para esto, primero se configuró el sistema con los parámetros obtenidos en el presente capítulo y se lo entrenó en la base de entrenamiento. Luego se clasificó la base de datos de Prueba y se utilizó la medida-F para evaluar el desempeño. Este procedimiento se repitió 100 veces y a partir de esto se obtuvo un resultado promedio y una desviación estándar en la clasificación de cada instrumento. Los valores pueden consultarse en la Figura 5.10. Allí además puede apreciarse la cantidad de muestras con las que dispuso el sistema para el análisis de cada instrumento. A partir de esto, puede señalarse que la desviación estándar parece ser inversamente proporcional a la cantidad de muestras de la clase. Sin embargo, es complejo establecer una causalidad entre cantidad de muestras y desempeño del sistema. El grupo de instrumentos clásicos de cuerdas como Violín, Viola, Violoncello y Contrabajo tienen una gran cantidad de muestras (entre 364 y 416) y un desempeño alto (entre 0,9 y 0,93) pero no de los más altos. La trompeta por ejemplo con 142 muestras tiene un desempeño de 0,99. La Tuba con tan solo 70 muestras posee un desempeño de 0.96, un 5 % más elevado que el grupo de las cuerdas. De todas formas, para los instrumentos de cuerda la desviación estándar promedio es de 0,02 y para la Tuba este valor es de 0,03. Por lo que se sostiene una menor desviación estándar para mayor cantidad de muestras.

Figura 5.10: Promedio y Desviación estándar tras 100 ciclos de clasificación del sistema para cada instrumento. 'n' indica la cantidad de muestras para cada clase.

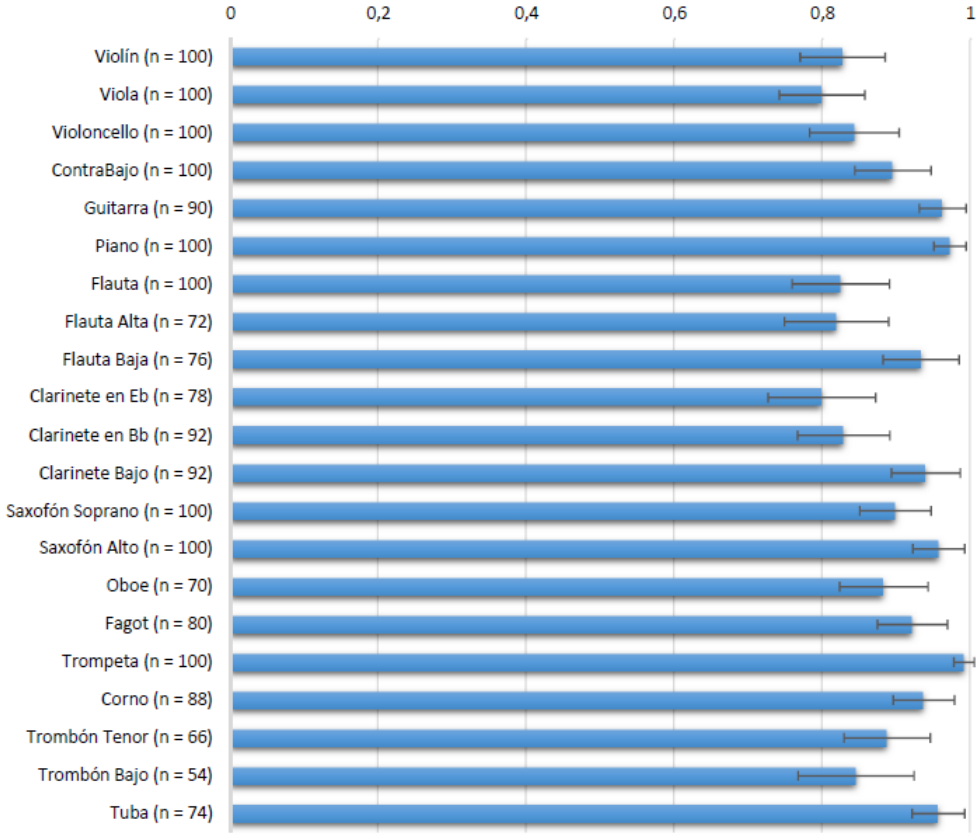


5.5. Clasificación del sistema acotando la cantidad de muestras para cada instrumento

Siguiendo los resultados presentados en la sección anterior, se propuso evaluar el sistema con una base de datos cuya cantidad de muestras para cada instrumento fuese pareja. Para esto,

se decidió limitar el número de muestras por clase a 100, tomando estas de manera aleatoria entre todas las muestras de la clase. Usando el procedimiento de la sección anterior se llegó a los resultados presentados en la Figura 5.11. Puede apreciarse en la familia de la cuerdas clásicas que el desempeño bajó y la desviación estándar creció, lo que refuerza la hipótesis de que la cantidad de muestras guarda en general una relación inversa con la desviación estándar. Finalmente, se puede observar una disminución general en el rendimiento de la identificación pero sin cambiar cualitativamente los resultados relativos entre instrumentos. Es decir que reducir el número de muestras de una clase tiene un efecto negativo sobre la identificación de la misma. Este efecto además se traslada al desempeño general. Este fenómeno fue estudiado con mayor profundidad en la sección 5.2.

Figura 5.11: Promedio y Desviación estándar tras 100 ciclos de clasificación del sistema para cada instrumento tomando hasta 100 muestras de cada uno.



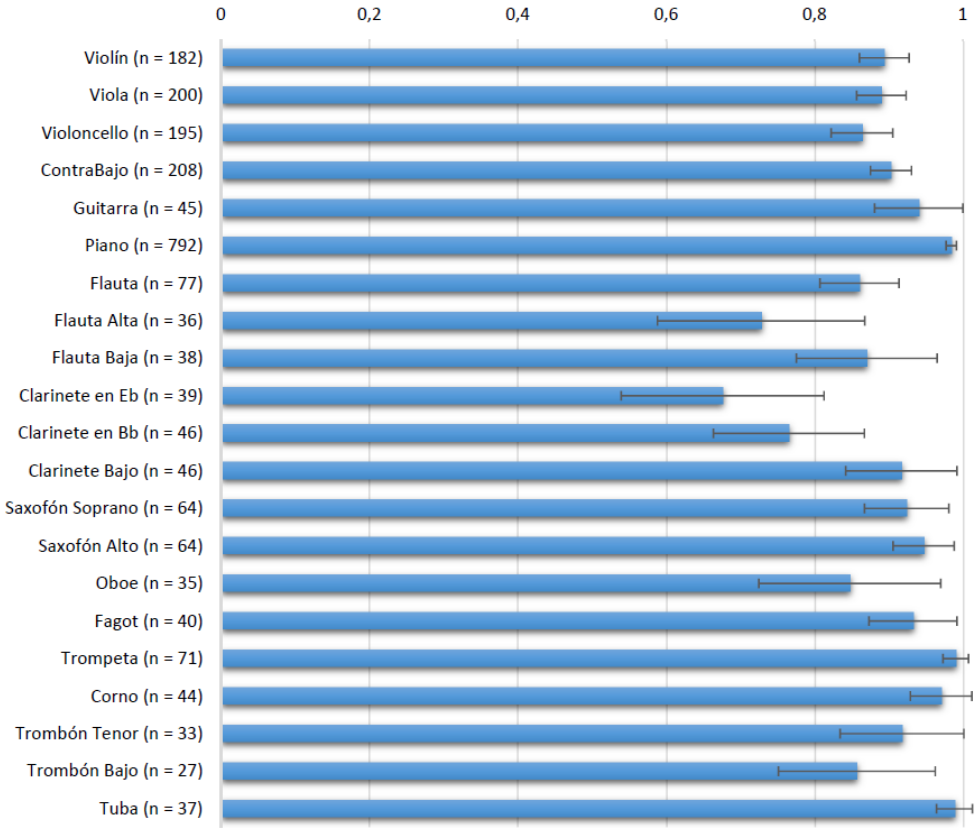
5.6. Análisis de la clasificación del sistema excluyendo las muestras producidas por transformaciones

Como fue demostrado en la sección 5.2, el sistema puede beneficiarse de una base de datos más extensa. Con este objetivo se recurrió a aplicar una transformación a cada una de las muestras. Así se les agregó una reverberación y se duplicó el tamaño total de la base. Los detalles de esta implementación fueron presentados en la sección 3.2.1. A su vez, la evaluación del desempeño del sistema tras aplicar este proceso fue presentada en la sección 5.2.

Para obtener mayor información sobre el efecto que generan las muestras con reverberación, se evaluó el desempeño del sistema para cada instrumento sin estas. De esta forma, se pueden comparar los desempeños del sistema completo presentados en las Figuras 5.10 y 5.11 con los del sistema sin las muestras producidas por reverberaciones en las Figuras 5.12 y 5.13.

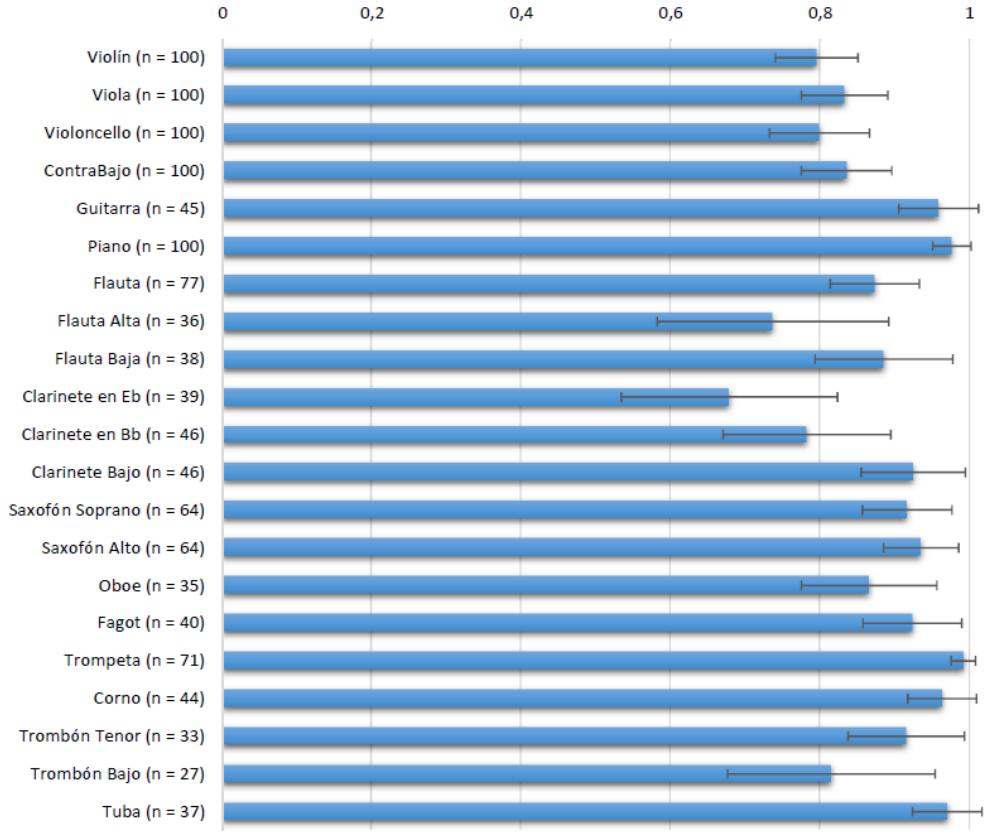
Al agregar las muestras con reverberación al sistema, la mejora en el desempeño promedio fue de 0,019 y la desviación estándar promedio bajó 0,022. Esto equivale a una reducción de la desviación estándar del 35 % y un incremento del desempeño del 2 %. Sin embargo, esta tendencia general no se ve reflejada en cada clase particular. Para el fagot, sumar las muestras con reverberación deviene en una caída del desempeño de 0,93 que posee sin las transformaciones a 0,89. Las otras clases cuyo rendimiento cae al agregar las muestras con reverberación son la guitarra (-0,02), el corno (-0,03), el trombón tenor (-0,02), el trombón bajo (-0,005) y la tuba (-0,02). Cabe destacar que a pesar de esto, la caída en la desviación estándar es constante para todas las clases.

Figura 5.12: Promedio y Desviación estándar tras 100 ciclos de clasificación del sistema para cada instrumento excluyendo las muestras producidas por transformaciones.



Para el sistema limitado a 100 muestras por clase, el efecto de ampliar la base de datos mediante transformaciones es similar al del sistema evaluando la base completa. Se presenta una mejora en el desempeño promedio de 0,016 (2 %) y una caída de la desviación estándar promedio de 0,023 (30 %). En este caso, también se encuentran instrumentos cuyo rendimiento cae al agregar reverberaciones. Estos son: viola (-0,03), piano (-0,004), flauta (-0,05), saxofón soprano (-0,02), fagot (-0,002), trompeta (-0,0006), corno (-0,03), trombón tenor (-0,03) y tuba (-0,01). En este análisis también se destaca que la desviación estándar cae para todas las clases con excepción del violín donde aumentó marginalmente (0,002). La incorporación de la reverberación a los sonidos agrega complejidad en la descripción del sonido y puede distorsionar la homogeneidad de las clases. Se puede especular que esta es la causa de la pérdida del rendimiento en la clasificación de algunos instrumentos. Deben hacerse estudios más profundos para establecer una vínculo causal explícito.

Figura 5.13: Promedio y Desviación estándar tras 100 ciclos de clasificación del sistema para cada instrumento tomando hasta 100 muestras de cada uno y excluyendo las que fueron producidas por transformaciones.



5.7. Análisis de influencia del registro de un instrumento

Para profundizar el análisis del sistema, se evaluó si el registro de un instrumento influencia en la clasificación del mismo. Para las pruebas se eligió el violín dado que su registro se encuentra en un extremo (el agudo). Se evaluó el instrumento con el registro completo, sin las dos cuerdas más graves (Sol-Re) y sin las dos más agudas (Mi-La). Los resultados pueden consultarse en la tabla 5.2. Los valores son el promedio tras 100 iteraciones del sistema. En la tabla se muestra el valor de la medida-F para el violín, la familia de las cuerdas, todos los instrumentos salvo los de esta familia y el valor global. Se comprueba que ambas porciones del registro contribuyen a las clasificación correcta del instrumento y que al eliminar las cuerdas más agudas la clasificación del violín es un poco más precisa. Sin embargo, la diferencia tanto numérica como porcentual es tan pequeña que resulta apresurado generar conclusiones a partir de las mismas.

Tabla 5.2: Evaluación del sistema sin las dos cuerdas más graves o más agudas del violín.

	Medida F		
	Completo	Sin Re-Sol	Sin Mi-La
Violín	0.9322	0.9107	0.9250
Global	0.9080	0.9087	0.9053
Cuerdas	0.9192	0.9193	0.9178
Global sin cuerdas	0.9053	0.9058	0.9023

5.8. Clasificación de muestras nuevas

Finalmente, se evaluó la posibilidad de extender el sistema a muestras desconocidas. Para esto se obtuvieron nuevas señales y se evaluó su clasificación en el sistema. Hasta este punto, todas las pruebas se realizaron utilizando la base de datos presentada en la sección 3.2. Esta tiene la particularidad de poseer grabaciones realizadas por académicos de manera muy controlada. En cambio, para esta prueba se obtuvieron 50 muestras nuevas de violín a través de la plataforma 'freesound' [78]. De este modo, las muestras nuevas (monofónicas) pueden venir de cualquier usuario y tipo de generación.

La clasificación de las muestras nuevas se realizó configurando la base de datos de varias formas distintas. Inicialmente, el sistema fue entrenado utilizando la base completa. Los resultados pueden consultarse en la tabla 5.3. La primer prueba se realizó sobre la base de datos completa. Esta otorgó una clasificación muy pobre. Al indagar sobre esta clasificación se detectó que en la mayoría de los casos el sistema clasificaba las muestras nuevas como 'Piano'. Luego, se decidió eliminar el Piano del sistema y volver a realizar la clasificación. En este caso, el resultado fue muy satisfactorio dado que la medida F para las muestras nuevas fue tan solo un 10 % menor que para las muestras originales. Como el Piano es el instrumento con más muestras de la base, se procedió a realizar pruebas con todos los instrumentos pero limitando la cantidad máxima de muestras a utilizar. Puede verse que si bien los resultados son siete veces mejores que los que utilizan la base de datos completa, no llegan al mismo nivel de precisión que se obtiene al eliminar el piano.

Tabla 5.3: Medida F de la clasificación de muestras nuevas de violín.

Base de datos	Medida F
Completa	0.0769
Completa Sin Piano	0.8235
Hasta 400 muestras	0.5507
Hasta 300 muestras	0.5714
Hasta 100 muestras	0.5915

A la luz de estos resultados y comparando con los mostrados en la Tabla 2.3, donde se muestran los desempeños de algoritmos de otros autores, puede verse que para sistemas con cantidad de instrumentos similares el rendimiento obtenido en este trabajo está a la altura del estado del arte y en algunos casos lo supera. Es evidente también, que la cantidad de factores y variables que influyen en este tipo de problemas dificulta la comparación directa entre trabajos de distintos autores. La comparación ideal debería hacerse con la misma base de datos de instrumentos y la misma métrica de evaluación de resultados.

Capítulo 6

Conclusiones

Se diseñó e implementó un sistema de identificación automática de instrumentos musicales acústicos a partir de señales monofónicas digitales utilizando métodos de aprendizaje autónomo. Las principales conclusiones obtenidas son:

- El sistema puede identificar hasta 21 instrumentos distintos con un rendimiento similar o superior a los que existen en la literatura.
- El método de máquinas de vectores de soporte presenta un mejor rendimiento que el de agrupamiento de clases K-mean.
- Se encontró que la mayor debilidad del sistema desarrollado era la acotada cantidad de muestras disponibles, por lo que se extendió la base de datos agregando reverberaciones a las señales originales. Este método permitió mejorar el rendimiento general del sistema, y hasta donde el autor conoce es un método novedoso para este tipo de problemas.
- En estrecha relación con el punto anterior, se evaluó el desempeño del sistema para cada instrumento en particular y se observó que la desviación estándar del desempeño de la clase guarda en general una relación inversa con la cantidad de muestras.
- La eliminación de descriptores redundantes que no aportaban información trajo aparejada la mejora en la clasificación del sistema a la vez que aumentó su robustez.
- En base a las pruebas preliminares (solo en violín) de la influencia del registro del instrumento sobre el sistema, no se han podido sacar conclusiones definitivas. Se deben profundizar los estudios y ampliarlos a otros instrumentos para analizar la generalización de los resultados.
- En cuanto a la posibilidad de extender el sistema, se hicieron pruebas con muestras desconocidas y se estima que el sistema puede ser extendido para utilizarse con cualquier tipo de señal monofónica.

Futuras líneas de investigación

Como trabajo a futuro se extenderá el sistema a señales monofónicas desconocidas. Para esto, es importante primero mejorar la base de datos en las clases que poseen menor cantidad de muestras. Luego, podrá extenderse el sistema de manera controlada siguiendo un procedimiento similar al utilizado en la sección 5.8. En este punto en particular queda también como trabajo a futuro perfeccionar la clasificación de muestras nuevas de violín sin necesidad de eliminar ninguna otra clase.

Se plantea además la posibilidad de evaluar un sistema de clasificación jerárquico. Así, el sistema podría informar si la muestra corresponde a una familia de instrumentos además de a cuál instrumento. Hace falta evaluar no sólo si esta modificación mejora el desempeño final, sino también qué utilidad tendría la información adicional para la comunidad. Además, se podría ampliar el sistema agregando una clase de 'Fuera de alcance'. De esta forma, podrían clasificarse muestras que no pertenecen a ningún instrumento en esta clase.

Finalmente, se pretende desarrollar un entorno gráfico que acompañe al sistema desarrollado y facilite su inserción en marcos educativos. De esta forma, podría encontrarse un nuevo uso del sistema permitiendo que estudiantes y académicos realicen modificaciones sobre el mismo y evalúen los resultados.

Anexo A

Funciones principales del código

En este anexo se presentan las funciones principales del software desarrollado. Para mayor profundidad, remitirse al código publicado en `https://github.com/andimarafioti/AIAMI`.

A.1. Estructura del proyecto

El proyecto se encuentra dividido en cinco carpetas. Estas son: 'Description', 'Evaluation', 'Identification', 'Transformation' y 'utils'. Dentro de las primeras cuatro se encuentra el código dedicado a la descripción, evaluación, identificación y transformación, respectivamente. En la carpeta 'utils' se hayan funciones útiles para el resto del programa y que no pertenecen específicamente a ninguna etapa del mismo.

A.2. Etapa de descripción de las muestras

En la carpeta 'Description' se encuentra el código dedicado a la descripción de los sonidos. Las dos funciones principales para esta etapa son: 'extractAllDescriptors' y 'calculateDescriptorAndWriteToFile'. La primera función se encarga de realizar las descripciones de los sonidos. Toma como entrada la señal de audio y devuelve un diccionario con los valores para cada descriptor. La segunda función realiza varias tareas. Primero se encarga de cargar los archivos de audio y llamar con estos al extractor de descriptores. Luego, utiliza una función utilitaria para promediar los valores de los descriptores. Finalmente, guarda los vectores promediados en un archivo Json.

A.3. Etapa de identificación

En la carpeta 'Identification' se encuentra el código dedicado a la descripción de los sonidos. Las funciones principales para esta etapa son: 'loadAllDescriptors', 'preprocessDescriptors' y 'separateDatabases'. La primera de estas funciones carga los descriptores que fueron almacenados en archivos Json. La segunda realiza un preprocesamiento sobre los descriptores para que tengan varianza unitaria y desviación estándar nula. La tercera toma el conjunto de descriptores y los separa en las bases de datos de entrenamiento, validación cruzada y testeo.

Dentro de esta carpeta también se encuentra un módulo llamado 'svm' (Support Vector Machine). En este hay dos funciones que condensan la utilización del módulo de máquinas de vectores de soporte de la librería Scikit-Learn presentada en la sección 3.4.2. A este patrón de programación se lo conoce como Fachada [79]. Se utiliza para reducir la cantidad de opciones que el usuario del programa desarrollado tiene a la hora de diseñar el algoritmo de clasificación.

A.4. Etapa de Evaluación

En la carpeta ‘Evaluation’ se encuentra el código dedicado a la evaluación del sistema. Así mismo, contiene una carpeta llamada ‘outputs’ donde se guardan ciertos resultados obtenidos a partir de las pruebas realizadas. Los archivos más importantes son: ‘testC-gamma’, ‘testDataSize’ y ‘testFeatures’. El primero de estos evalúa los parámetros C y Γ del algoritmo de máquinas de vectores de soporte. El segundo mide la influencia de la cantidad de muestras utilizadas en la etapa de entrenamiento de la identificación. El tercero prueba el error del sistema según la cantidad de descriptores utilizados para la identificación. El procedimiento utilizado y los resultados de estos archivos fueron discutidos en mayor profundidad en el capítulo 5.

Bibliografía

[1] P. Herrera-Boyer, A. Klapuri, and M. Davy, "Automatic classification of pitched musical instrument sounds," in *Signal Processing Methods for Music Transcription* (A. Klapuri and M. Davy, eds.), pp. 163–200, Springer, 2006.

[2] G. C. Bowker and S. L. Star, *Sorting Things Out: Classification and Its Consequences (Inside Technology)*. The MIT Press, Oct. 1999.

[3] E. M. von Hornbostel and C. Sachs, "Classification of musical instruments: Translated from the original german by anthony baines and klaus p. wachsmann," *The Galpin Society Journal*, vol. 14, pp. pp. 3–29, 1961.

[4] V. Mahillon, *Éléments d'acoustique musicale & instrumentale: comprenant l'examen de la construction théorique de tous les instruments de musique en usage dans l'orchestration modern*. Mahillon, 1874.

[5] M. Consortium, "Revision of the hornbostel-sachs classification of musical instruments by the mimo consortium," 2011. Consultado en linea el 23-09-2015: `www.mimo-international.com/documents/HornbostelSachs.pdf`.

[6] N. Fletcher and T. Rossing, *The Physics of Musical Instruments*. Springer, 1998.

[7] K. D. Martin, *Sound-source Recognition: A Theory and Computational Model*. PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, USA, 1999. AAI0800880.

[8] A. Srinivasan, T. Ikenaga, K. Shimizu, and I. Fujinaga, "Recognition of isolated instrument tones by conservatory students.," in *7th International Conference on Music Perception and Cognition*, (Sidney, Australia), pp. 720–723, 2002.

[9] R. A. Kendall, "The role of acoustic signal partitions in listener categorization of musical phrases," *Music Perception: An Interdisciplinary Journal*, vol. 4, no. 2, pp. 185–213, 1986.

[10] J. C. Brown, O. Houix, and S. Mcadams, "Feature dependence in the automatic identification of musical woodwind instruments," *Journal of the Acoustical Society of America*, vol. 109, no. 3, pp. 1064–1072, 2001.

[11] T. D. Griffiths, "The neural processing of complex sounds," in *The Cognitive Neuroscience of Music*, ch. 11, Oxford University Press, 2001.

[12] S. Handel, *Listening: An introduction to the perception of auditory events*. Cambridge: MIT Press, 1989.

[13] P. Szczuko, P. Dalka, M. Dabrowski, and B. Kostek, "Mpeg-7-based low-level descriptor effectiveness in the automatic musical sound classification," in *Audio Engineering Society Convention 116*, May 2004.

[14] T. Kitahara, M. Goto, and H. G. Okuno, "Musical instrument identification based on F0-dependent multivariate normal distribution," in *ICME '03: Proceedings of the 2003 International Conference on Multimedia and Expo - Volume 3 (ICME '03)*, (Washington, DC, USA), pp. 409–412, IEEE Computer Society, 2003.

[15] T. M. Mitchell, *Machine Learning*. New York, NY, USA: McGraw-Hill, Inc., 1 ed., 1997.

[16] Y. Bengio, I. J. Goodfellow, and A. Courville, "Deep learning." Libro en proceso de ser terminado para la editorial MIT Press, 2015.

[17] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, "Deepface: Closing the gap to human-level performance in face verification," June 2014.

[18] H. Luo, P. L. Carrier, A. C. Courville, and Y. Bengio, "Texture modeling with convolutional spike-and-slab rbms and deep extensions," *CoRR*, vol. abs/1211.5687, 2012.

[19] B. Gold and N. Morgan, *Speech and Audio Signal Processing: Processing and Perception of Speech and Music*. New York, NY, USA: John Wiley & Sons, Inc., 1st ed., 1999.

[20] A. K. Jain, R. P. W. Duin, and J. Mao, "Statistical pattern recognition: A review," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, pp. 4–37, Jan. 2000.

[21] K. Pearson, "Mathematical contributions to the theory of evolution. III. Regression, heredity, and panmixia," *Philosophical Transactions of the Royal Society of London. Series A*, vol. 187, pp. 253–318, Jan. 1896.

[22] I. StatSoft, "Electronic statistics textbook: Nonparametric statistics.," 2011. Consultado en linea el 3-09-2015: `http://www.statsoft.com/textbook/nonparametric-statistics/`.

[23] J. M. Grey, "Multidimensional perceptual scaling of musical timbres," *The Journal of the Acoustical Society of America*, vol. 61, pp. 1270–1277, May 1977.

[24] D. Wessel, *Timbre space as a musical control structure*. Wessel, 1982.

[25] C. L. Krumhansl, "Why is Musical Timbre so hard to understand?," in *Structure and Perception of Electroacoustic Sound and Music, Proceedings of the Marcus Wallenberg symposium 1998* (S. Nielzén and O. Olsson, eds.), pp. 43–53, Excerpta Medica, 1989.

[26] S. Lakatos, "A common perceptual space for harmonic and percussive timbre," *Perception and Psychophysics*, vol. 62, pp. 1426–1439, 2000.

[27] G. Peeters, S. McAdams, and P. Herrera, "Instrument description in the context of mpeg-7," 2000. MPEG.

[28] S. B. Davis and P. Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," *Acoustics, Speech and Signal Processing, IEEE Transactions on*, pp. 357–366, 1980.

[29] A. V. Oppenheim and R. Schafer, "From frequency to quefrency: A history of the cepstrum," *IEEE Signal Processing Magazine*, vol. 21, no. 5, pp. 95–106, 2004.

[30] M. Sahidullah and G. Saha, "Design, analysis and experimental evaluation of block based transformation in {MFCC} computation for speaker recognition," *Speech Communication*, vol. 54, no. 4, pp. 543 – 565, 2012.

[31] H. Niemann, *Klassifikation von Mustern*. zweite ed., 2003.

[32] G. Peeters, "A large set of audio features for sound description (similarity and classification) in the CUIDADO project," tech. rep., IRCAM, 2004.

[33] I. H. Witten and E. Frank, *Data Mining: Practical Machine Learning Tools and Techniques, Second Edition (Morgan Kaufmann Series in Data Management Systems)*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2005.

[34] G. Peeters and X. Rodet, "Hierarchical gaussian tree with inertia ratio maximization for the classification of large musical instruments database," in *Proceedings of the 6th International Conference on Digital Audio Effects*, 2003.

[35] A. Livshin, G. Peeters, and X. Rodet, "Studies and Improvements in Automatic Classification of Musical Sound Samples," 2003.

[36] P. K. Janert, *Data Analysis with Open Source Tools*. O'Reilly Media, Inc., 1st ed., 2010.

[37] M. E. Tipping and C. M. Bishop, "Mixtures of probabilistic principal component analyzers," *Neural Comput.*, vol. 11, pp. 443–482, Feb. 1999.

[38] B. Kostek, "Application of soft computing to automatic music information retrieval," *Journal of the American Society for Information Science and Technology*, vol. 55, no. 12, pp. 1108–1116, 2004.

[39] B. Kostek, P. Zwan, and M. Dziubinski, "Statistical analysis of musical sound features derived from wavelet representation," in *Audio Engineering Society Convention 112*, Apr 2002.

[40] C. Röver, F. Klefenz, and C. Weihs, "Identification of musical instruments by means of the Hough-transformation," in *Classification—the ubiquitous challenge: Proceedings of the 28th annual conference of the Gesellschaft für Klassifikation* (C. Weihs and W. Gaul, eds.), pp. 608–615, Heidelberg: Springer-Verlag, 2005.

[41] A. Eronen, "Automatic Musical Instrument Recognition," Master's thesis, 2001.

[42] *Linear Predictive Models for Musical Instrument Identification*, vol. 5, 2006.

[43] J. C. Brown, "Computer identification of musical instruments using pattern recognition with cepstral coefficients as features," *The Journal of the Acoustical Society of America*, vol. 105, no. 3, pp. 1933–1941, 1999.

[44] I. Kaminskyj and T. Czaszejko, "Automatic recognition of isolated monophonic musical instrument sounds using knnc," *Journal of Intelligent Information Systems*, vol. 24, no. 2-3, pp. 199–221, 2005.

[45] A. Eronen, "Musical instrument recognition using ica-based transform of features and discriminatively trained hmms," in *Signal Processing and Its Applications, 2003. Proceedings. Séptimo Simposio Internacional.*, vol. 2, pp. 133–136 vol.2, Julio 2003.

[46] B. Kostek, "Musical instrument classification and duet analysis employing music information retrieval techniques," *Proceedings of the IEEE*, vol. 92, no. 4, pp. 712–729, 2004.

[47] T. H. Park, *Towards Automatic Musical Instrument Timbre Recognition*. PhD thesis, Princeton University, NJ, USA, November 2004.

[48] N. Chetry, M. Davies, and M. Sandler, "Musical instrument identification using lsf and k-means," in *Audio Engineering Society Convention 118*, May 2005.

[49] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*. Springer Series in Statistics, New York, NY, USA: Springer New York Inc., 2 ed., 2008.

[50] E. Jones, T. Oliphant, P. Peterson, *et al.*, "SciPy: Open source scientific tools for Python," 2001–. Consultado en linea el 07-08-2015: `http://www.scipy.org/`.

[51] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[52] B. E. Boser, I. M. Guyon, and V. N. Vapnik, "A training algorithm for optimal margin classifiers," in *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, COLT '92, (New York, NY, USA), pp. 144–152, ACM, 1992.

[53] Wikipedia, "Support vector machine maximum separation hyperplane with margin," 2016. Consultado en linea el 15-2-2016: `https://en.wikipedia.org/wiki/File:Svm_max_sep_hyperplane_with_margin.png`.

[54] J. Vert, K. Tsuda, and B. Schölkopf, *A Primer on Kernel Methods*, pp. 35–70. Cambridge, MA, USA: MIT Press, 2004.

[55] C.-W. Hsu and C.-J. Lin, "A comparison of methods for multiclass support vector machines," *Trans. Neur. Netw.*, vol. 13, pp. 415–425, Mar. 2002.

[56] Documentacion de Scikit-learn, *Support Vector Machines*. Consultado en linea el 08-11-2015: `http://scikit-learn.org/stable/modules/svm.html`.

[57] C. Cortes and V. Vapnik, "Support-vector networks," *Mach. Learn.*, vol. 20, pp. 273–297, Sept. 1995.

[58] A. Ng, "Machine learning." Curso Online de la Universidad de Stanford, Septiembre 2015.

[59] T. I. S. for Music Information Retrieval, "Resources - datasets." online, 2015. Consultado en linea el 08-07-2015: `http://www.ismir.net/resources.html#datasets`.

[60] P. Margaretic, "Estimación automática de f0-múltiple en señales musicales orientada a pianos," *Universidad Nacional de Tres de Febrero*, 2015.

[61] U. of Iowa, "Electronic music studios." Online. Consultado en linea el 2015-08-07: `http://theremin.music.uiowa.edu/MIS.html`.

[62] A. M. Barbancho, I. Barbancho, L. J. Tardon, and E. Molina., *Database of Piano Chords, An Engineering View of Harmony*. Springer-Verlag New York, 1st ed., 2013.

[63] M. T. Group, "Irmas: A dataset for instrument recognition in musical audio signals." Universitat Pompeu Fabra, Barcelona. Consultado en línea el 08-07-2015: `http://mtg.upf.edu/download/datasets/irmas`.

[64] H. fur Musik, "Saarland music data," 2005. Consultado en linea el 08-07-2015: `http://resources.mpi-inf.mpg.de/SMD/SMD_Western-Music.html`.

[65] H. fur Musik, "Saarland music data (smd)," 2005. Consultado en linea el 08-07-2015: http://resources.mpi-inf.mpg.de/SMD/index.html.

[66] M. Jeub, M. Schäfer, and P. Vary, "A binaural room impulse response database for the evaluation of dereverberation algorithms," in *Proceedings of the 16th International Conference on Digital Signal Processing*, DSP'09, (Piscataway, NJ, USA), pp. 550–554, IEEE Press, 2009.

[67] J. Bosch, J. Janer, F. Fuhrmann, and P. Herrera, "A comparison of sound segregation techniques for predominant instrument recognition in musical audio signals," in *13th International Society for Music Information Retrieval Conference (ISMIR 2012)*, (Porto, Portugal), pp. 559–564, 08/10/2012 2012.

[68] F. Fuhrmann, *Automatic musical instrument recognition from polyphonic music audio signals*. PhD thesis, Universitat Pompeu Fabra, 2012.

[69] M. P. Institut, "Max planck institut für informatik: Home." online. Consultado en linea el 2015-08-07: http://www.mpi-inf.mpg.de/home/.

[70] C. Severance, *Python for Informatics: Exploring Information*. CreateSpace Independent Publishing Platform, 2013. Consultado en linea el 15-11-2015: http://www.pythonlearn.com.

[71] M. Summerfield, *Rapid GUI Programming with Python and Qt : the Definitive Guide to PyQt Programming*. 2007.

[72] S. McConnell, *Code Complete, Second Edition*. Redmond, WA, USA: Microsoft Press, 2004.

[73] Python Software Foundation, "Documentación del lenguaje de programación python, versión 2.7." Consultado en linea el 15-11-2015: https://www.python.org/.

[74] D. Ascher, P. F. Dubois, K. Hinsen, J. Hugunin, and T. Oliphant, "NumPy: Numerical Python," 1999. Consultado en linea el 08-07-2015: http://http://www.numpy.org.

[75] D. Bogdanov, N. Wack, E. Gómez, S. Gulati, P. Herrera, O. Mayor, G. Roma, J. Salamon, J. Zapata, and X. Serra, "Essentia: an audio analysis library for music information retrieval," in *International Society for Music Information Retrieval Conference (ISMIR'13)*, (Curitiba, Brazil), pp. 493–498, 04/11/2013 2013.

[76] M. Sokolova and G. Lapalme, "A systematic analysis of performance measures for classification tasks," *Inf. Process. Manage.*, vol. 45, pp. 427–437, July 2009.

[77] S. Fortmann-Roe, "Understanding the bias-variance tradeoff," Junio 2012. Consultado en línea el 12-10-2015: http://scott.fortmann-roe.com/docs/BiasVariance.html.

[78] F. Font, G. Roma, and X. Serra, "Freesound technical demo," in *ACM International Conference on Multimedia (MM'13)*, (Barcelona, Spain), pp. 411–412, ACM, ACM, Oct 2013.

[79] S. Chazallet, *Python 3: Los fundamentos del lenguaje*. Recursos Informáticos, ENI, 2015.