



Universidad
Carlos III de Madrid

Tratamiento de Información Multimedia

Memoria de proyecto:
Reconocedor de acordes

Daniel Barrejón Moreno 100330734
Miguel Fernández Díaz 100330080

Índice

1. Introducción	2
2. Preprocesado y extracción de características	2
2.1. Descripción del PCP	2
2.2. Principio de funcionamiento	3
2.3. Preprocesado	3
3. Base de datos	4
3.1. Descripción	4
4. Métodos de clasificación y resultados obtenidos	5
4.1. Clasificadores empleados	5
4.2. Pruebas realizadas	6
4.2.1. Subconjunto 1	6
4.2.2. Subconjunto 2	7
5. Conclusiones	7
6. Referencias	8

1. Introducción

En este proyecto planteamos el desarrollo e implementación de un sistema capaz de reconocer y clasificar acordes musicales para guitarra en MATLAB. Este sistema emplea el conjunto de características conocidas como Pitch Class Profile y está orientado a reconocer los acordes más empleados en la música popular occidental.

Además, evaluaremos si nuestro clasificador es capaz de reconocer acordes generados por otros instrumentos.



Estructura del sistema fig. 1

2. Preprocesado y extracción de características

El descriptor utilizado para la identificación de acordes ha sido el vector conocido como Pitch Class Profile (PCP), un descriptor ampliamente utilizado en el campo de reconocimiento automático de acordes, diseñado primeramente por T. Fujishima en 1999 y replicado con variaciones por muchos otros desde entonces.

2.1. Descripción del PCP

El vector PCP que hemos utilizado se compone de 12 posiciones, correspondientes a los 12 semitonos de la escala temperada, en el cual se mide la proporción relativa de la energía de la señal de entrada con respecto a los semitonos. Además, este vector de características también presenta sensibilidad frente a otros parámetros como el timbre del instrumento, el perfil armónico del acorde, el ataque de la nota o el sustain.

C	C#	D	D#	E	F	F#	G	G#	A	A#	B
---	----	---	----	---	---	----	---	----	---	----	---

Estructura del vector PCP fig. 2

Puesto que los acordes a clasificar se componen de tres notas (primera, tercera y quinta para los acordes mayores; y primera, tercera disminuida y quinta para los acordes menores), nuestros vectores PCP ideales tendrán tres de sus posiciones distintas de cero.

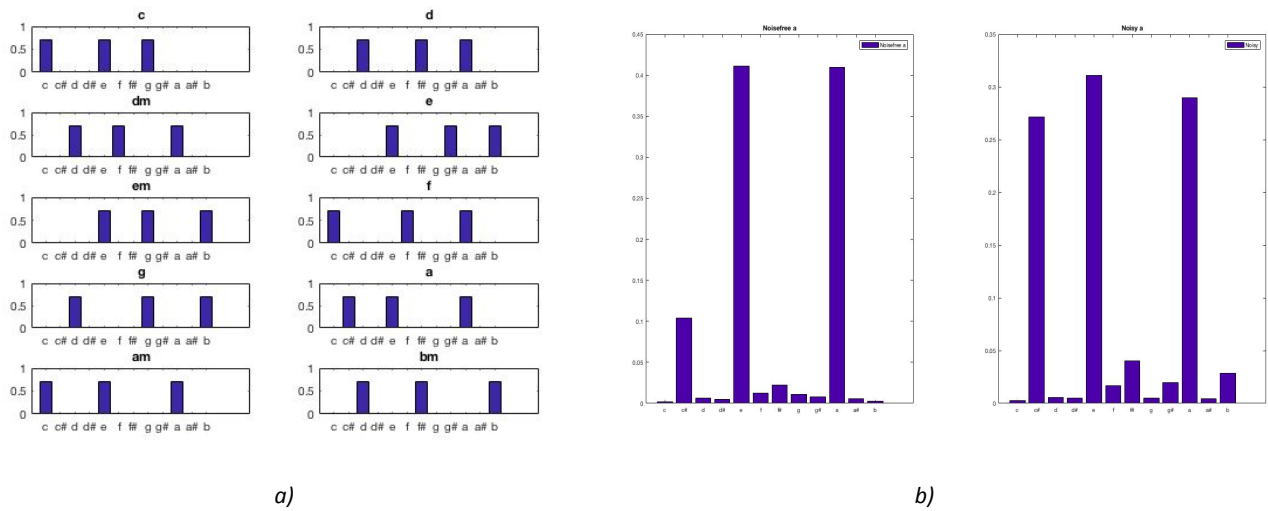


Fig. 3: a) Representación ideal de los acordes PCP. b) Ejemplo de un acorde de La sin ruido(izquierda) y con ruido (derecha).

2.2. Principio de funcionamiento

Para calcular el vector de características de un acorde primero se debe realizar la DFT de la señal de entrada. Esta transformada luego se multiplicará por la función delta de Kronecker por el resultado de una función de mapeo y la posición real del vector PCP en la que se encuentra, consiguiendo así evaluarla sólo en los puntos que contienen notas en las frecuencias correctas.

$$PCP^*(p) = \sum_l \|X(l)\|^2 \delta(M(l), p)$$

La función de mapeo por su parte, calculará la posición en el vector PCP que le correspondería a la frecuencia que se está procesando. Para ello necesita una frecuencia de referencia, que es la de la nota de la primera posición de nuestro vector PCP, en este caso C3 (130.8 Hz).

$$M(l) = \{round(12 \log_2((f \cdot \frac{1}{N}) / f_{ref})) \bmod 12 \mid l = 1, \dots, \frac{N}{2} - 1\}$$

Como último paso, el vector ha de normalizarse para un mejor desempeño del sistema.

$$PCP(p) = \frac{PCP^*(p)}{\sum_{j=1}^{12} PCP^*(j)}$$

2.3. Preprocesado

El preprocesado lo hemos dividido en dos partes:

1. Etiquetado de las muestras
2. Cálculo de la DFT

- **Etiquetado**

Aunque nuestra base de datos está ya etiquetada, debemos guardar esas etiquetas de tal forma que luego podamos usarlas en la etapa de clasificación. Para ello, procesamos los nombres de cada tipo de acorde asociándole su etiqueta correspondiente. El mapeo queda por lo tanto así:

A	Am	Bm	C	D	Dm	E	Em	F	G
1	2	3	4	5	6	7	8	9	10

- **Cálculo de la DFT**

En este paso simplemente calculamos la DFT de cada acorde que vamos a tener en nuestros diferentes conjuntos. Así, evaluamos la siguiente fórmula:

$$X(k) = \sum_{n=0}^{N-1} e^{-2\pi i k n / N} x(n)$$

Cabe destacar que con evaluar la función de $X(0), \dots, X(N/2 - 1)$ es suficiente ya que $x(n)$ es real.

También debemos mencionar que para cada conjunto que utilizaremos en la clasificación (más detalles en el apartado 4) hemos decidido definirlos como structs de Matlab donde almacenamos los siguientes valores.

struct			
names [1xN]	path [1xN]	labels [1xN]	features [Nx12]
Necesario para etiquetado	Necesario para DFT	Etiquetas o clases	Vector PCP

3. Base de datos

3.1. Descripción

Nuestra base de datos se compone de dos subconjuntos: un primer subconjunto formado por 2000 acordes (1000 sin ruido y 1000 con ruido) grabados con 4 guitarras distintas y ejecutados con distintas técnicas; y un segundo subconjunto de 400 muestras de varios instrumentos: guitarra, piano, violín y acordeón.

Las muestras están limitadas a los 10 acordes más populares de la música occidental (A, Am, Bm, C, D, Dm, E, Em, F, y G), y se componen de archivos .wav muestreados a 44100Hz y cuantizados con 16 bits.

Para nuestros experimentos hemos creado varios conjuntos de entrenamiento y test detallados en la siguiente tabla:

Entrenamiento	Test
700 muestras de guitarra con ruido (subconjunto 1)	300 muestras de guitarra con ruido (subconjunto 1)
700 muestras de guitarra sin ruido (subconjunto 1)	300 muestras de guitarra con ruido (subconjunto 1)
1400 muestras de guitarra (50% con ruido/50% sin ruido) (subconjunto 1)	100 muestras de guitarra con ruido (subconjunto 2)
	100 muestras de piano con ruido (subconjunto 2)
	100 muestras de violín con ruido (subconjunto 2)
	100 muestras de acordeón con ruido (subconjunto 2)

4. Métodos de clasificación y resultados obtenidos

4.1. Clasificadores empleados

Los clasificadores que hemos empleado para poner a prueba el sistema han sido un K-NN y una SVM multiclase de kernel de función radial gaussiano (one vs all para todas las clases). Ambos han sido optimizados mediante validación cruzada de 5 iteraciones: para el K-NN se ha optimizado el número de vecinos, y para la SVM se ha optimizado la desviación estándar o sigma.

Para la validación cruzada el conjunto de entrenamiento empleado ha sido el conjunto mezclado de 1400 muestras, ya que da una representación más global de todas las posibles notas al incluir distintas guitarras con distintos timbres y al haber sido grabadas tanto con ruido como sin ruido.

Los valores de los parámetros óptimos han sido los siguientes:

K	1
sigma	2.35

4.2. Pruebas realizadas

Podemos dividir en dos categorías todas las pruebas realizadas: las pruebas realizadas con el subconjunto 1 (sólo guitarra), y las realizadas con el subconjunto 2 (guitarra, piano, acordeón y violín).

4.2.1. Subconjunto 1

Las pruebas realizadas son 6 para ambos clasificadores. Para estos experimentos todas las muestras han sido extraídas del subconjunto 1. Para estas pruebas hemos tenido en cuenta todas las posibles combinaciones entre los 3 conjuntos de entrenamiento y los 2 conjuntos de test descritos debajo. Los porcentajes de acierto obtenidos se muestran en las siguientes tablas.

K-NN			
Test/Entrenamiento	Sin ruido	Con ruido	Mezclado
Sin ruido	96.7 %	89 %	96.3 %
Con ruido	94 %	95.3 %	95 %

SVM			
Test/Entrenamiento	Sin ruido	Con ruido	Mezclado
Sin ruido	94.3 %	92 %	96 %
Con ruido	79.7 %	95.3 %	95 %

Para evaluar los resultados, debemos fijarnos especialmente en los realizados con el conjunto de entrenamiento mezclado, ya que es el más se ajusta a lo que podemos encontrar en un caso real. Observamos que ambos clasificadores son bastante similares después de haber sido optimizados. En un primer momento, sin haber optimizado la SVM, nuestros resultados eran más pobres que los del K-NN.

Además, nos gustaría comentar que, en este caso de estudio, el K-NN puede llegar a ser un clasificador muy potente dada la precisión y la caracterización que ofrece el PCP, aunque estemos trabajando en 12 dimensiones. No obstante, esto no quiere decir que dejemos de necesitar un espacio muestral muy extenso para cubrir esas dimensiones.

4.2.2. Subconjunto 2

Para el subconjunto 2 hemos decidido usar como conjunto de entrenamiento el mezclado de 1400 muestras perteneciente al subconjunto 1 a la luz de los resultados obtenidos.

K-NN	
Test/Entrenamiento	Mezclado
Guitarra	90 %
Piano	82 %
Violín	77 %
Acordeón	81 %

SVM	
Test/Entrenamiento	Mezclado
Guitarra	87 %
Piano	83 %
Violín	67 %
Acordeón	83 %

5. Conclusiones

A lo largo de este proyecto hemos podido demostrar cómo nuestro vector de características PCP basado en una sencilla DFT puede valer para describir de una manera bastante precisa los diferentes acordes que componen nuestra base de datos. Además, un vector PCP con 12 posiciones puede ser suficiente para el reconocimiento de un rango pequeño de acordes, en nuestro caso 10. De hecho, resulta bastante sorprendente ya que es un algoritmo muy rápido en cuanto a tiempo de procesamiento, y hubiese sido más difícil alcanzar esta rapidez con algoritmos más complejos.

También podemos destacar que para esta clase de problemas de reconocimiento de acordes el conjunto de entrenamiento que merece la pena emplear es aquel que utiliza muestras sin ruido y con ruido, ya que puede adaptarse más fácilmente a casos prácticos donde van a ser utilizados y permite evitar el problema de sobreajuste que puede aparecer al considerar conjuntos de entrenamiento puramente sin ruido y con ruido.

Finalmente, consideramos que los resultados obtenidos con el K-NN son bastante satisfactorios, a pesar de trabajar en 12 dimensiones, y que los resultados obtenidos con la SVM son igualmente bastante aceptables teniendo en cuenta que únicamente hemos optimizado un parámetro.

6. Referencias

[1] J. Osmalskyj, J.-J. Embrechts, S. Piérard, M. Van Droogenbroeck “*Neural Networks for musical chords recognition*”.

http://jim.afim-asso.org/jim12/pdf/jim2012_08_p_osmalskyj.pdf

[2] Takuya Fujishima “*Real-Time Chord Recognition of Musical Sound: A System Using Common Lisp Music*”.

http://www.music.mcgill.ca/~jason/mumt621/papers5/fujishima_1999.pdf

[3] Base de datos:

<http://www.montefiore.ulg.ac.be/services/acous/STSI/file/jim2012Chords.zip>