NEURAL NETWORKS FOR MUSICAL CHORDS RECOGNITION

J. Osmalskyj, J-J. Embrechts, S. Piérard, M. Van Droogenbroeck
INTELSIG Laboratory, University of Liège, Departement EECS
josmalsky@ulg.ac.be

ABSTRACT

In this paper, we consider the challenging problem of music recognition and present an effective machine learning based method using a feed-forward neural network for chord recognition. The method uses the known feature vector for automatic chord recognition called the Pitch Class Profile (PCP). Although the PCP vector only provides attributes corresponding to 12 semi-tone values, we show that it is adequate for chord recognition.

Part of our work also relates to the design of a database of chords. Our database is primarily designed for chords typical of Western Europe music. In particular, we have built a large dataset filled with recorded guitar chords under different acquisition conditions (instruments, microphones, etc), but also with samples obtained with other instruments. Our experiments establish a twofold result: (1) the PCP is well suited for describing chords in a machine learning context, and (2) the algorithm is also capable to recognize chords played with other instruments, even unknown from the training phase.

1. INTRODUCTION

With the widespread availability of digital music over the Internet, it has become impossible to process that huge amount of data manually; even organizing a personal collection of music samples is challenging. Therefore automatic tools have a role to play. Music Information Retrieval (MIR) is an interdisciplinary science whose goal is to extend information retrieval into non textual-only areas. The aim of MIR is to describe multiple aspects related to the content of music. Some applications of MIR include music transcription, music classification [1], playlist generation [8, 20], and music recognition [3].

Traditionally, music is annotated with text information provided by the cover. This text information is adequate to characterize lyrics automatically but totally inappropriate to describe music content. Clearly, an approach based on text lacks flexibility. In addition, researchers are also interested in extending audio information retrieval using a more human natural interaction, for example, humming a song [3], tapping rhythm, or playing an instrument.

The first compulsory step of a retrieval system able to process music is the characterization of music. Several techniques are available but the probably best known to musicians is that of chords. A chord can be defined as *a set of simultaneous tones* [16]. This definition might be appropriate for a human but, from a classification point of view, it appears to be inappropriate because there are many variations (due to the instruments, noise, recording conditions, etc) even for a unique chord.

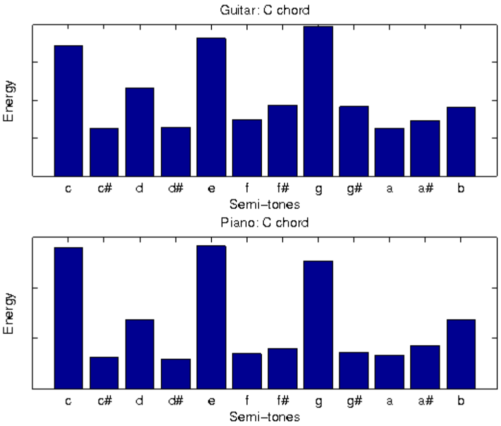


Figure 1. This paper shows that PCP features are well suited for representing chords, regardless of the used instrument. To illustrate the robustness with respect to the instruments, the upper and lower drawings provide the representation of a C chord played with a guitar and with a piano respectively. The three main peaks are the same.

From a technical point of view, we show in this paper that Pitch Class Profile (PCP) features [9] are suitable candidates as they have a small sensitivity to instrument change (see Figure 1).

The PCP is a compact representation of the frequency content of a sound expressed as the relative proportion of energy with respect to the 12 notes of a standard chromatic scale. Notes are defined on a logarithmic frequency scale, and energy is expressed in natural units. Most of the studied PCP features are sensitive to the harmonics depending on the musical instrument, and other parameters such as dynamic, attack and sustain. To take these influences into consideration, we propose, in this paper, a novel approach using machine learning techniques.

But a good descriptor does not suffice. In the paper, we establish that a naive application of the definition of chords to classify PCPs fails to provide good results. In addition, we also prove that once this diversity is correctly handled by machine learning methods, chords form an adequate description for recognizing musics. During our work, we used machine learning techniques for chords recognition. However, such algorithms usually need a labeled database in order to learn a classification model. As, to our knowledge, there is no such database publicly available. Therefore, we have created a new large dataset, which is publicly available.

The remainder of this paper is organized as follows. Section 2 reviews the related work in the field of chord characterization and classification. Section 3 provides a brief reminder of the PCP feature vector and details why a learning algorithm is preferable to a nearest neighbor approach. Section 4 describes our new database, its design, and its content. Section 5 presents the results of experiments, and Section 6 concludes the paper.

2. RELATED WORK

Chroma features, also known as Pitch Class Profiles (PCP), have been used as front end to chord and songs recognition systems from audio recorded queries. In particular, it has been demonstrated that chroma features are well suited for cover songs identification systems [5, 6, 7, 11, 14, 15, 17].

PCP features are good mid-level features which provide a more reliable and straightforward representation of songs than melody. In [18], Serra describes the PCP features as derived from the energy found within a given frequency range in short-time spectral representations of audio signals. This energy is usually collapsed into a 12-bin octave independent histogram representing the relative intensity of each of the 12 semitones of an equal-tempered chromatic scale.

The original PCP was introduced by Fujishima [9] in 1999. In this PCP, the intensities of all frequency bins within the boundaries of a semitone are summed-up and the semitones in octave distance are added-up to pitch classes, resulting in a 12-bin PCP vector. Variations of this vector include 24-bin and 36-bin vectors, resulting in more precise features. Fujishima used his PCP vector to perform pattern matching using binary chord type templates (*i.e.* ideal PCP representations as shown in Figure 2).

Lee [16] introduced a new feature vector called the Enhanced Pitch Class Profile (EPCP) for automatic chord recognition from the raw audio. To this end, he first obtained the Harmonic Product Spectrum (HPS) from the constant Q transform (CQT) of the input signal and then he applied an algorithm to that HPS for computing a 12-dimensional enhanced pitch class profile. The CQT has geometrically spaced center frequencies which can be dimensioned so that they correspond to musical notes. It is thus an interesting pre-processing step for music computing.

Gomez and Herrera [10] proposed a system that automatically extracts, from audio recordings, tonal metadata such as chord, key, scale and cadence information. In their work, they computed a vector of low-level instantaneous features: the HPCP (Harmonic Pitch Class Profile) vector. It is based on the intensity of each pitch mapped to a single octave, which corresponds to Fujishima's PCP.

Harte [12] also proposed a method using the CQT for chord recognition. In addition, he added a tuning algorithm which is able to deal with variations in instrument tuning.

Sheh and Ellis [19] proposed a statistical learning method for chord segmentation and recognition, where they used Hidden Markov Models (HMMs) trained by the Expectation Maximization (EM) algorithm, and treated chords labels as hidden values within the EM framework.

Most of the aforementioned work on chords recognition does not make use of machine learning techniques, but rather uses signal processing techniques in order to obtain the best possible 12-bin PCP vectors, and then perform pattern matching. However, it is very difficult to obtain a perfect 12-bin PCP vector which highlights only the main notes of a chord. Indeed, each instrument brings new harmonics, and the dynamic of the musician, among other parameters, adds noise to the PCP. For this reason, we propose a system based on machine learning techniques, whose goal is to learn a suitable model encapsulating all these parameters.

However, no real labelled chords database seems to be publicly available (to our knowledge) to build such a model. In this work, we propose a database, and we consider the use of real chords samples to train a more accurate chord recognition system. Since our goal is to use our system for music recognition, we need fast algorithms, which are necessary to deal with huge databases of songs. Therefore, we chosed to use the original PCP vector because it is fast and involves few pre-processing steps.

3. CHROMA FEATURES

3.1. Principle

The most commonly used descriptor for chord identification has been the Pitch Class Profile (PCP). A chord is composed of a set of tones regardless of their heights, and therefore a PCP vector seems to be an ideal feature to represent a musical chord.

There are some variations to obtain a 12-bin PCP, but its computation usually follows the same steps. First the algorithm transforms a fragment of the input sound to a discrete Fourier transform (DFT) spectrum $X(.)$. Then the algorithm derives the PCP from $X(.)$. Let $PCP^*(p)$ be a vector defined for $p = 0,\, 1,\, \ldots,\, 11$ as

$$PCP^*(p) = \sum_l ||X(l)||^2 \, \delta(M(l),\, p) \qquad (1)$$

where $\delta\,(\cdot,\cdot)$ denotes Kronecker's delta. $M(l)$ is defined as

$$M(l) = \begin{cases} -1 & l = 0 \\ round(12\, log_2((f_s.\frac{l}{N})/f_{ref}))\, mod\, 12 & l = 1,\, \ldots,\, \frac{N}{2} - 1 \end{cases}$$

where f_{ref} is the reference frequency falling into $PCP^*(0)$, N the number of bins in the DFT of the input signal, and f_s is the sampling frequency. For example, for a standard scale starting with a C, the reference frequency is $130.80\, Hz$.

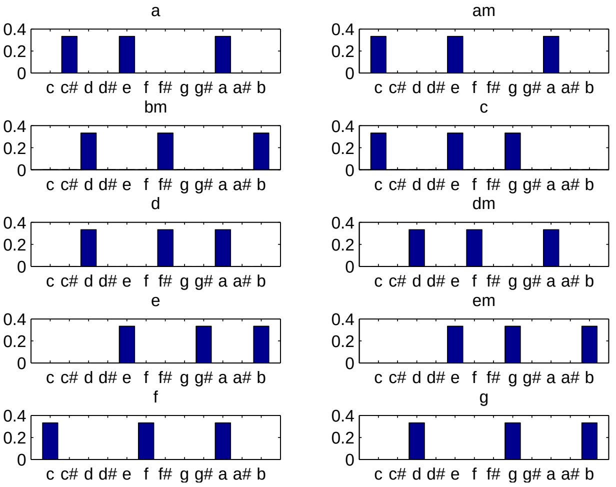


Figure 2. Ideal PCP representations of ten chords.

In order to compare PCP vectors, it is necessary to normalize them. Indeed, a chord can be played louder or softer and therefore, energy distribution can vary from one trial to another. To normalize a PCP vector, we divide the energy of each bin by the total energy of the original PCP, that is,

$$PCP(p) = \frac{PCP^*(p)}{\sum_{j=0}^{11} PCP^*(j)} \tag{2}$$

where p is the index of the bin we want to normalize.

3.2. Experiments

The PCP is an intuitive descriptor for a musician because it highlights the main notes of a chord. Indeed, a musician is able to recognize a chord by identifying the notes contained in that chord. The PCPs of ten chords are given in Figure 2.

Apparently, the PCP seems to be suitable for representing a chord. However, recognizing chords based on the PCP is not a trivial task. In a first attempt, we developed a naive but simple chord detector that only compares histograms using a nearest neighbors (1-NN) method with the Bhattacharyya distance [4] as a distance measure. Basically, the algorithm takes an arbitrary PCP vector as input, normalizes it, and compares it to a predefined list of histograms representing the various chords to be recognized. The algorithm then classifies the chord as the one of the closest known histogram. It turns out that results of that simple method are unsatisfactory (see Section 5 for more details).

We also tried using variations of the PCP vector using 24-bin and 36-bin vectors. However, the overall results do not vary much. Therefore, we decided to keep the 12-bin vector for faster processing.

4. DATABASE

Part of the difficulties in machine learning techniques originate from the elaboration of a database of samples, also called dataset, and chord classification is no exception. The primary requirement for a chord recognizer using machine learning techniques is that the dataset contains enough data to build a model. Most of the work described in Section 2 on chord recognition does not make use of machine learning techniques which could explain why no chords database seems to be publicly available. For that reason, we had to create our own database of chords. In our database, we gathered audio files (recorded in the WAV format, sampled at $f_s = 44100$ Hz, and quantized with 16 bits), and the corresponding precomputed PCP vectors. The PCP vectors were computed on windows comprising each 16384 samples, which correspond to 0,37 seconds. The window size was chosen experimentally. We noticed that windows containing only 4096 samples produce correct results, however, best results for our application were achieved using a bigger window size.

Since our final goal is not to recognize all the existing chords, but to develop a music recognition system, we can limit chords to the most frequent ones. Therefore, we chose a subset of 10 chords:

$$A,\ Am,\ Bm,\ C,\ D,\ Dm,\ E,\ Em,\ F,\ G.$$

In our database, these chords are represented by an identifier ranging respectively from 0 to 9. Note that if other chords are also played in a song, the main chords can suffice. Moreover, many modern songs played in Western Europe are based on these 10 chords. Therefore, it seems to be a suitable starting point to validate our recognition method.

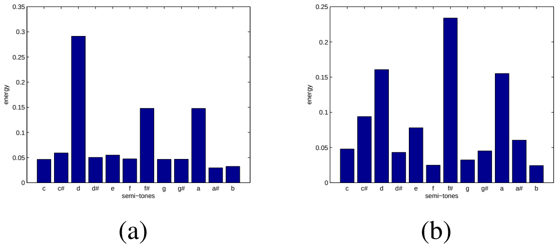
In practical terms, all PCP vectors are stored in a unique file which is organized as follows. Each line consists in a normalized PCP vector of twelve elements and one more element for the corresponding chord identifier. The following is illustrative of one line of the dataset file

$$0.04,\ 0.09,\ 0.18,\ 0.05,\ 0.12,\ 0.04,$$

$$0.14,\ 0.04,\ 0.03,\ 0.18,\ 0.04,\ 0.05,\ \mathbf{4}$$

The last digit corresponds to the class (the *D* chord in this example).

Next we concentrate on the context of the dataset. As we are willing to validate our dataset and test it on samples acquired in different contexts, the database was split into two subsets. The first subset contains a very large amount of guitar chord samples, whereas the second subset contains a smaller set of chords played with a different guitar and three other instruments. Therefore, there are two ways of using the database: we can either use cross-validation techniques on the first subset, or use it as a learning set while the second subset is used as a test set. Details follow.



(a) (b)

Figure 3. PCP representations of a D chord recorded with the same guitar in (a) an anechoic chamber, and (b) a noisy room. Note that the three major semi-tones are still visible in (b).

4.1. First subset

Chords of the dataset are produced with an acoustic guitar, which is probably the most common instrument in Western Europe to play chords. The acquisition conditions are the following. The chords samples were recorded in two different environments. Half of the chords were recorded in an anechoic chamber with a wideband microphone (01dB MCE320). The other half was recorded in a noisy environment, with a single live microphone (Shure SM58). We felt that samples recorded in both environments would reflect both the situations of professionals playing their songs in a studio and people playing music home. In Section 5, we derive that the system performs best if it is trained with a mix of noise-free and noisy chords. It is worth noticing that the chords were recorded using several playing styles (arpege, staccato, legato, etc.).

Figure 3 shows the PCP representations of a D chord recorded in the anechoic chamber and in the noisy room with the same guitar. As many real songs are played in a noisy environment, it is relevant to include noisy chords in the database.

For each chord, 100 samples were recorded in the anechoic chamber, and 100 samples were recorded in a noisy room. For each environment, the samples were recorded using four different guitars: one classical guitar with nylon strings, and three acoustic guitars producing three different sounds. It is expected that the variety of the dataset with respect to guitars will enhance the robustness of the system and extend its applicability, as there are many different guitar sounds available worldwide.

In conclusion, the first subset is organized as follows. There are 2000 chords in total. Each specific chord is recorded 200 times, 100 in an anechoic chamber and 100 in a noisy room. In both hundred halves, the chords are further separated into four subsets of 25 chords, produced with one of the four guitars.

4.2. Second subset

We also created a smaller database containing chords recorded with a guitar and three other instruments, namely a piano, a violin, and an accordion. That database is intended to provide an independent test set and should not be used to train the model. That database contains 100 chords for each instrument. These 100 chords are distributed equally among the ten chords mentioned earlier. Thus, there are 10 samples per chord for each instrument.

Parameters	Values
Number of hidden layers	1
Number of neurons in the hidden layer	35
Learning rate	0.001
Momentum	0.25
Weight decay	0.0

Table 1. Neural network parameters.

Our chords database is publicly available at *http://www.montefiore.ulg.ac.be/services/acous/STSI/file/jim2012Chords.zip.*

5. EXPERIMENTS

This section first introduces the learning method chosen for the design of our system. Then we detail the various experiments performed and discuss the results. With our experiments, we want to clarify and investigate several hypotheses:

1. We want to check if a naive application of the chord definition suffices.
2. How do we have to build the training set? Should noise-free samples, noisy samples, or both types of samples be included during training?
3. We want to evaluate the performance of our learning algorithm with our database.
4. Is the algorithm capable to recognize chords played with various other instruments?

5.1. Learning algorithm

Most techniques proposed in the literature for chord recognition do not use machine learning methods. Fujishima [9] used a pattern matching technique and heuristics to recognize chords. Although the techniques developed are efficient, they are complex. Since our final goal is not to develop a new chord descriptor, we chose to use a very simple, though powerful, technique to recognize chords. The chosen algorithm is a feed-forward neural network using a classical gradient descent algorithm with a negative log-likelihood [13] as cost function.

The neural network architecture is the following. There are twelve input attributes, which correspond to the twelve semi-tones of the PCP vector representing the chord. The neural network outputs a vector of 10 values, corresponding to the output neurons, each one being the probability of the detected chord to be issued from the corresponding chord. The final settings of the neural network were optimized by a 10-fold cross-validation on the learning database. Table 1 gives the parameters of the network.

TS / LS	Noise-free	Noisy	Mixed
Noise-free	4.0 %	5.0 %	4.0 %
Noisy	11.7 %	6.0 %	7.3 %

Table 2. Results of the validation with noise-free, noisy, and mixed Learning Sets (LS) and Test Sets (TS). The table gives the total classification error rate for each configuration.

5.2. Experiment 1: naive application of the chord definition

In this first experiment, we have created a synthetic and ideal sample of PCP for each chord manually (see Figure 2). Then, using the Bhattacharyya distance [4], we have applied a nearest neighbors (1-NN) algorithm on our second subset. The classification error rates obtained are the following: 8 % for guitar, 20 % for piano, 19 % for violin, and 32 % for accordion. These results are clearly unsatisfactory. The conclusion is straightforward: a learning based on real samples is necessary to reach the required performance level.

5.3. Experiment 2: determining the optimal learning set

In section 4.1, we explained that we created a database with noise-free chords and noisy chords. To justify that choice, we performed six tests to determine the best of the three following configurations:

- a learning set with noise-free chord samples only,
- a learning set with noisy chord samples only,
- and a learning set with mixed chord samples.

To perform the test, we split the database in different sets. First, the original database of 2000 samples was split into two sub-databases of 1000 elements. The first one only contains noise-free chords and the second one contains noisy chords. Then, we created three learning sets containing 70% of each sub-database. The first set contains 700 noise-free chords, the second 700 noisy chords, and the last one 350 noise-free chords and 350 noisy samples, taken randomly.

For the tests, we created two Test Sets (TS) with 30% of each sub-database. One test set contains noise-free chords and the second one contains noisy chords only. It is worth noticing that the chords used for the TS are not included in the LS.

Table 2 gives the results of each test. First, we trained the model with a noise-free learning set and tested it with the noise-free and noisy test sets (first column of the table). Next, we trained the model with a noisy learning set and tested it with a noise-free and noisy test sets (second column). Finally, we performed exactly the same tests with a learning set containing both noise-free and noisy chord samples (third column). The chords were recorded with different guitars distributed equally in each set.

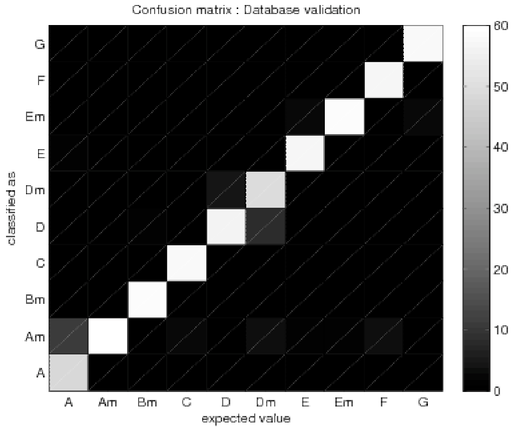


Figure 4. Confusion matrix for a network trained with a mixed learning set of 1400 samples (140 chords per class). The test set contains 600 samples (60 per class). The total error rate is 6.5%.

From the results given in Table 2, we conclude that building the model using a noise-free learning set produces the highest error rate for the noisy test set. Moreover, the optimal learning set (noisy or mixed) depends on the conditions under which the model has to be used. Unfortunately, we are not able to guess it in advance. However, we consider that the mixed learning set produces models that are less dependent of the noise in the database than with a noisy learning set, and therefore, we believe it is preferable to use a mixed learning set.

5.4. Experiment 3: validating the database

From our previous observations, we have decided to train our final model with both noise-free and noisy chord samples. Figure 4 shows the confusion matrix for the final network trained with a learning set of 1400 chords, that is 140 chord per class and a test set of 600 chords, that is 60 per class. The main database was split in two parts and thus, the result is slightly biased due to the size reduction of the database. Despite that bias, we can observe that the prediction of each class is quite good. Indeed, the rate of correct classification for each class is almost identical. Moreover, classification errors are not concentrated in a unique position.

5.5. Experiment 4: recognizing other instruments

In this experiment, we applied our method to other instruments. We chose four instruments capable of playing chords, namely a guitar, a piano, an accordion, and a violin. These instruments were chosen because they are widely used in Western Europe. Figure 5 compares the PCP representations of a C chord played with the four instruments. As can be seen, the PCP representations of the four instruments are similar.

Although the model was trained with a database containing only guitar chords, we applied it on the indepen-

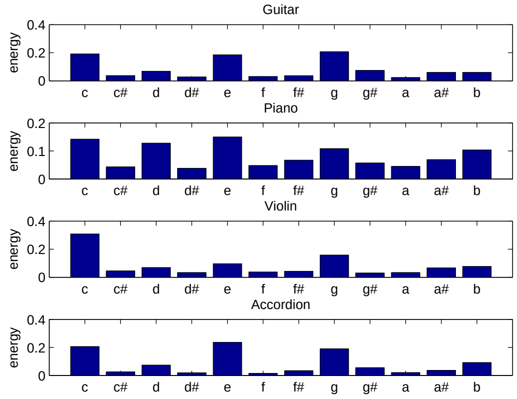


Figure 5. PCP of a C chord played with four instruments (respectively, from top to bottom, a guitar, a piano, a violin, and an accordion). The c, e and g notes are very similar on the four instruments and dominate over the other notes.

Instrument	1-NN with the chord definition	Neural network with a learning set
Guitar	8 %	1 %
Piano	20 %	13 %
Violin	19 %	5 %
Accordion	32 %	4 %

Table 3. Classification error rates for 4 different instruments using our method.

dent test sets mentioned in Section 4. It is worth remembering that these recordings are completely independent of the chords used to train the model.

Table 3 gives the results of our method for the four instruments, and compares them to the naive approach (see Section 5.2). We observe huge improvements in the results with a learning method based of real chord samples compared to that of the naive approach. It appears that it is harder to recognize chords played on a piano, which could be explained by the noisy nature of piano sounds (as graphically illustrated with the PCP of a piano, in Figure 5).

Figure 6 shows the confusion matrices obtained for each instrument using the trained neural network. The predictions are good, but less precise for the piano. Best results are obtained with an independent test set of guitar chords, as the model was trained with guitar chords. Violin and accordion also give good results compared to the naive method, and produce a classification error rate of respectively 5% and 4%, which is promising.

6. CONCLUSIONS

This paper presents a method based on machine learning techniques, using a feed-forward neural network, for chords recognition, applicable to raw audio. As no chords database seems to be publicly available, we have built our own dataset containing chords recorded in an anechoic chamber and in a noisy room. Both environments were chosen to increase the robustness of the algorithm with respect to the acquisition process. The database is made of 2000 different guitar chords saved in WAV files and distributed into 10 chords classes.

We have highlighted that the best strategy consists in using a learning set containing both noise-free and noisy samples. Experimental results also show that our attributes, that is, the 12-dimensional PCP vectors, are effective representations of chords, and that they are also applicable to other instruments, like piano, violin, and accordion. However, the PCP representation has to be sent to a feed-forward neural network which learns a model to recognize the ten chords. Our method, based on the provided database, outperforms a direct application of the chord definition using 1-NN with Bhattacharyya distance. Finally, results for the recognition of chords played with other instruments are also presented. Our method also performs well for such PCP samples.

Despite the use of a simple 12-bin PCP vector based on the Discrete Fourier Transform, we show promising results and fast processing, which would have probably not been achieved with more complex pre-processing steps. It is worth noticing however, that for pure chords identification, our system is limited to ten chords, which may seem restrictive. Thus, the lack of availability of a complete labeled chords database is a limitation to our system.

However, recognizing ten chords seems to be sufficient for our next application, i.e., song excerpts recognition. Moreover, fast processing is an important property of this application and the first results are very promising.

In this future work, we also plan to consider other algorithms for chords recognition, particulary in relation with the recently released Million Song Dataset (MSD) [2] to improve chords and music recognition.

Acknowledgments

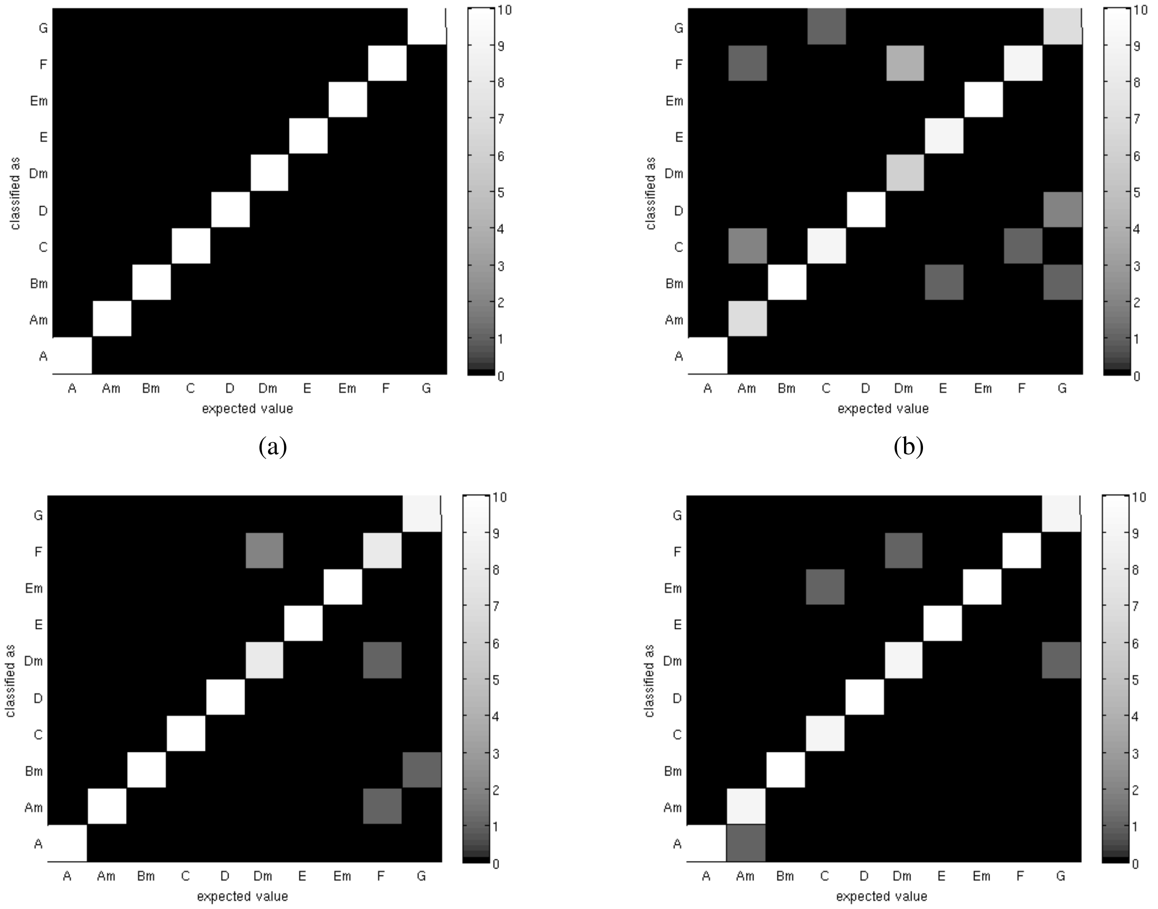
We would like to thank Maroussia Osmalskyj, Maxime Michaluk and Philippe Eloy for helping us recording the piano, the violin and the accordion.

7. REFERENCES

[1] J.-J. Aucouturier and F. Pachet. Music similarity measures: What's the use? In *Proceedings of the 3rd International Conference on Music Information Retrieval (ISMIR)*, pages 157–163, 2002.

[2] T. Bertin-Mahieux, D. Ellis, B. Whitman, and P. Lamere. The million song dataset. In *Proceedings of the 12th International Conference on Music Information Retrieval (ISMIR)*, 2011.

[3] M. Carre. *Systèmes de Recherche de Documents Musicaux par Chantonnement*. PhD thesis, École Na-



(a)

(b)

(c)

(d)

Figure 6. Confusion matrices of tests performed for four instruments: (a) guitar, (b) piano, (c) violin, (d) accordion.

tionale Supérieure des Télécommunications, Paris, 2002.

[4] M. Deza and E. Deza. *Encyclopedia of Distances*. Springer, 2009.

[5] A. Egorov and G. Linetsky. Cover song identification with if-f0 pitch class profiles. In *MIREX extended Abstract*, 2008.

[6] D. Ellis and C. Cotton. The 2007 labrosa cover song detection system. In *MIREX Extended Abstract*, 2007.

[7] D. Ellis and G. Poliner. Identifying cover songs with chroma features and dynamic programming beat tracking. In *Proceedings of the IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, 2007.

[8] A. Flexer, D. Schnitzer, M. Gasser, and G. Widmer. Playlist generation using start and end songs. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, pages 173–178, 2008.

[9] T. Fujishima. Realtime chord recognition of musical sound: a system using common lisp music. In *Proceedings ot the International Computer Music Conference (ICMC)*, pages 464–467, 1999.

[10] E. Gomez and P. Herrera. Automatic extraction of tonal metadata from polyphonic audio recordings. In *Proceedings of the International Conference: Metadata for Audio*, 2004.

[11] E. Gomez and P. Herrera. The song remains the same; identifying versions of the same song using tonal descriptors. *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, pages 180–185, 2006.

[12] C. Harte and M. Sandler. Automatic chord identifcation using a quantised chromagram. In *Proceedings of the 118th Audio Engineering Society Convention (AES)*, 2005.

[13] T. Hastie, R. Tibshirani, and J. Friedman. *The elements of statistical learning: data mining, inference, and prediction*. Springer Series in Statistics. Springer, second edition, September 2009.

[14] J. Jensen, M. Christensen, D. Ellis, and S. Jensen. A tempo-insensitive distance measure for cover song identification based on chroma features. In *Proceedings of the IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 2209–2212, 2008.

[15] S. Kim and S. Narayanan. Dynamic chroma feature vectors with applications to cover song identification. In *Proceedings of the IEEE Workshop on Multimedia Signal Processing (MMSP)*, pages 984–987, October 2008.

[16] K. Lee. Automatic chord recognition using enhanced pitch class profile. In *Proceedings of the International Computer Music Conference (ICMC)*, pages 306–313, 2006.

[17] M. Müller, F. Kurth, and M. Clausen. Audio matching via chroma-based statistical features. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, pages 288–295, 2005.

[18] J. Serra, E. Gómez, and P. Herrera. Audio cover song identification and similarity: Background, approaches, evaluation, and beyond. In *Advances in Music Information Retrieval*, pages 307–332. Springer, 2010.

[19] A. Sheh and D. Ellis. Chord segmentation and recognition using EM-trained hidden markov models. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, pages 121–124, 2003.

[20] L. Xiao, L. Liu, F. Seide, and J. Zhou. Learning a music similarity measure on automatic annotations with application to playlist generation. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, pages 1885–1888, 2009.