When audio features reach machine learning

Geoffroy Peeters, Frederic Cornu, David Doukhan, Enrico Marchetto, Remi Mignot, Kevin Perros, Lise Regnier⋆
FIRSTNAME.LASTNAME@IRCAM.FR, ⋆@GMAIL.COM
IRCAM-CNRS-UPMC UMR 9912 (STMS) — 1, pl. Igor Stravinsky - 75004 Paris - France

Abstract

We review here our work on using machine learning in the Ecoute, Quaero and BeeMusic projects to perform automatic labelling into music-tags (genre, mood and instrumentation). We discuss the evolution over times of audio feature design ranging from manually designed audio features to automatically designed audio features. We also discuss the way we deal with scalability issues in these projects.

1. Introduction

Music Information Retrieval is the field of research dedicated to the extraction, the analysis and the use of music related information. Part of this field is devoted to the automatic extraction of music content information from the analysis of its audio signal. This sub-field allows the development of music search and/or recommendation engines. In several projects, we have dealt with the problem of music auto-tagging, search-by-similarity and/or audio fingerprint, each time trying to improve our results.We report this evolution here.

In the Ecoute project (2006-2008), we developed our first generic system for music auto-tagging (Peeters, 2007) to be applied to genre and mood classification. The system was based on the extraction of a large set of audio features (Peeters, 2004), automatic feature selection algorithm and generative classifiers (GMM). The system was single-label. The trained system was then integrated into MPO Online/ WMI music catalogue. The development of the system was done on a database of around 5000 tracks.

In the Quaero project (2008-2012), we improve our auto-tagging system to be applied to the estimation of genre, mood and instrumentation in a multi-label mode[1] (Peeters et al., 2012). The system was based on a reduced set of pre-selected audio features (MFCC and Spectral Flatness Measure) but with a deep modelling of their temporal behaviour over the track duration (using Universal Background Model super-vectors and Multivariate Auto-Regressive model) and a set of discriminant classifiers (SVM-RBF). The results of these works were integrated into Orange and Exalead search-engines. The development of these technologies was done on a database of around 20.000 tracks and necessitated a first set of scale optimisations.

The goal of our current project, the BeeMusic project, is to provide the BIPP database[2] owned by the SNEP[3] with music content information. We focus on the automatic extraction of music genre, mood, automatic audio summary generation as well as providing a tool for finding duplicates in the database. The system is to be integrated into Kantar Media BIPP search engine. The size of the BIPP database is around 6.000.000 tracks and necessitated a large set of scale optimisations.

These three systems mainly differ by the scale of the database. Not only this difference of scale impacts the storage and computation needed but also the choice of the feature design and the machine learning algorithms to be used. We discuss this below.

2. Evolution of audio feature design

Historically audio features used in MIR were designed manually, either inspired by speech processing algorithms, by studies on perception or by the acoustic of musical instruments. Since some of these features require specific content and may not be meaningful otherwise (such as ap-

[1] We also developed systems for beat/downbeat/rhythm, chord succession, music structure estimation, automatic audio summary generation, search-by-similarity and audio fingerprint.

[2] Inter-professional Phonographic Producers Database

[3] Syndicat National de l'Edition Phonographique

plying the log-attack-time or the odd to even harmonic ratio to a music track), people relied on automatic feature selection algorithms or used by default the less specific ones. In the mid-2000s, MFCC and Chroma features had become the standard in MIR. In order to counterbalance this loss of description power, people then concentrated on modelling the temporal behaviour of the audio features: —a) by moving from mean and standard-deviation texture windows to more advanced techniques such as the block features (Seyerlehner, 2010), multi-prob-histogram (Kaiser, 2011), MAR model (Bimbot et al., 1992), Universal Background Model (UBM) super-vectors (Reynolds et al., 2000) (Charbuillet et al., 2011), —b) by modelling the 2D behaviour of the audio signal (in time and frequency): using FFT-based modulation spectrum (Peeters & Rodet, 2003), modulation Scale Spectrum (Marchand & Peeters, 2014), wavelet-based scattering transform (Andén & Mallat, 2011) or directly the 2D Fourier Transform (Bertin-Mahieux & Ellis, 2012). In all cases, the motivation of these is either the neuronal evidence of such a modelling or the search for invariants.

While this search for invariants is manually tractable for small-scale data-set, it is not for large-scale ones. Because of this, the automatic generation of audio features has been proposed. For this, Pachet et al.(Pachet & Zils, 2004) proposed the EDS-system based on genetic algorithm. Recently advances in hardware performances and increasing availibility of computational resources (CPUs/GPUs) has allowed high-throughput approaches, such as the use of multi-layer ANN; more specifically Deep Belief Network (Humphrey et al., 2012).

Discussion: Rather than an opposition between manually or automatically designed audio features, there exist a sort of continuum between the two. Indeed, EDS or DBN algorithms rarely start from audio waveforms but rather from higher-level representation inspired by manually designed audio features. Also, manually designed audio features are rarely used directly, but rather as input to higher-level modelling (such as UBM) which are based on ML algorithms. Manual feature design become un-tractable with the size of the data-set, and is also limited to the inspiration of the researcher. Automatic feature design can help going beyond this and thanks to the increasing availibility of computational resources can now be applied to large scale database. It is therefore very welcome. The question is now, apart from its performances, how to get knowledge out of automatically designed features ?

3. Scalability issues

Working with large databases involves several types of scalability issues. In our work in the BeeMusic project, we distinguish three types of processes involved in MIR algorithms, which we briefly describe below.

Common to these three processes is the storage capacity and data-to-CPU proximity issues. Since data need to be splitted into different hard-drives, hard-drive speed, network bandwidth, conflicting access or homogenity of distributed system becomes issues.

The first type of processes concerns the case of software that take as input an audio file and output its estimated content. This is the case of a beat-tracking, a chord detection, a music structure estimation algorithm or a pre-trained music genre and mood tagging system. In this case, since the estimation on one file is independent of the other files, the scalability issue only relates to the availability of CPU power and the processes can easily be parallelized.

The second type of processes concerns algorithms that necessitate the accessibility to all data but only during training. This is the case of the unsupervised training of UBM (using distributed RAM management), or the supervised training of SVM for music genre or mood tags. For this training, ML algorithms necessitates the access to all data usually splitted in many hard-drives. A common choice for this, is to use scalable distributed computing platform such as the Apache Hadoop, Mahout or Spark since those come with distributed implementation of ML algorithms in Map Reduce such as for K-Means, PCA, Naive Bayes. In the BeeMusic project, given the specificities of our ML algorithms, we decided to develop our own distributed computing platform based on extensions of our pre-existing ircamclass framework. This was done using the MapReduce paradigm and using java-sockets and MPI for parallelisation.

The third type of processes involves scalability issues for the deployment of the technology. This is the case when the process needs to perform online comparison between an item and the whole set of database items. Example of this, are audio fingerprint or search-by-acoustic-similarity technologies. In this case hashing techniques (such as locality-sensitive hashing (Slaney & Casey, 2008)) or specific indexing techniques (such as the M-tree (Charbuillet et al., 2010)) can be used. In the case of the BeeMusic project, and the finding-duplicates algorithm, we used the Apache SOLR/Lucene to perform scalable search. For this the audio representation is hashed into a text searchable query.

Discussion: There exist efficient solutions today that allows the MIR community to deal with the issues of large-scale databases. The question is now how can we reconnect the “small-scale” community of audio signal processing (that uses to analyse in details the estimation errors on a given audio file) to the “large-scale” community of machine learning (that measures performances in terms of accuracy or mean-recall over million of audio file) ?

Acknowledgments

This work was partly founded by the French government Programme Investissements d'Avenir (PIA) through the Bee Music Project, and by the European Commission through the SKAT-VG (618067) Project.

References

Andén, J. and Mallat, S. Multiscale scattering for audio classification. In *Proc. of ISMIR (International Society for Music Information Retrieval)*, pp. 657–662, Miami, Florida, USA, 2011.

Bertin-Mahieux, Thierry and Ellis, Daniel P.W. Large-scale cover song recognition using the 2d fourier transform magnitude. In *Proc. of ISMIR (International Society for Music Information Retrieval)*, Porto, Portugal, 2012.

Bimbot, Frédéric, Mathan, L., De Lima, A., and Chollet, G. Standard and target driven ar-vector models for speech analysis and speaker recognition. In *Proc. of IEEE ICASSP (International Conference on Acoustics, Speech, and Signal Processing)*, volume 2, pp. 5–8, San Francisco, California, USA, 1992.

Charbuillet, Christophe, Peeters, Geoffroy, Barton, Stanislav, and Gouet-Brunet, Valérie. A fast algorithm for music search by similarity in large databases based on modified symetrized kullback-leibler divergence. In *Proc. of IEEE CBMI (International Workshop on Content-Based Multimedia Indexing)*, pp. 1–6, Grenoble, France, June 2010.

Charbuillet, Christophe, Tardieu, Damien, and Peeters, Geoffroy. Gmm supervector for content based music similarity. In *Proc. of DAFx (International Conference on Digital Audio Effects)*, pp. 425–428, Paris, France, September 2011.

Humphrey, Eric, J., Bello, Juan Pablo, and LeCun, Yann. Moving beyond feature design: Deep architectures and automatic feature learning in music informatics. In *Proc. of ISMIR (International Society for Music Information Retrieval)*, Porto, Portugal, 2012.

Kaiser, Florian. Multi-probe histograms: A mid-level harmonic feature for music structure segmentation. In *Proc. of DAFx (International Conference on Digital Audio Effects)*, Paris, France, 2011.

Marchand, Ugo and Peeters, Geoffroy. The modulation scale-spectrum and its application to rhythm-content description. In *Proc. of DAFx (International Conference on Digital Audio Effects)*, Erlangen, Germany, 2014.

Pachet, François and Zils, A. Automatic extraction of music descriptors from acoustic signals. In *Proc. of ISMIR (International Society for Music Information Retrieval)*, Barcelona (Spain), 2004.

Peeters, Geoffroy. A large set of audio features for sound description (similarity and classification) in the cuidado project. Cuidado project report, Ircam, 2004.

Peeters, Geoffroy. A generic system for audio indexing: application to speech/ music segmentation and music genre. In *Proc. of DAFx (International Conference on Digital Audio Effects)*, Bordeaux, France, 2007.

Peeters, Geoffroy and Rodet, Xavier. Music structure discovering using dynamic audio features for audio summary generation: Sequence and state approach. In *Proc. of IEEE CBMI (International Workshop on Content-Based Multimedia Indexing)*, pp. 207–214, Rennes, France, 2003.

Peeters, Geoffroy, Cornu, Frédéric, Tardieu, Damien, Charbuillet, Christophe, Burred, Juan José, Ramona, Mathieu, Vian, Marie, Botherel, Valérie, Rault, Jean-Bernard, and Cabanal, Jean-Philippe. A multimedia search and navigation prototype, including music and video-clips. In *Proc. of ISMIR (International Society for Music Information Retrieval)*, Porto, Portugal, October 2012.

Reynolds, D.A., Quatieri, T.F., and Dunn, R.B. Speaker verification using adapted gaussian mixture models. *Digital signal processing*, 10(1-3):19–41, 2000.

Seyerlehner, Klaus. *Content-Based Music Recommender Systems: Beyond simple Frame-Level Audio Similarity*. PhD thesis, Johannes Kepler Universität, Linz, Austria, December 2010.

Slaney, Malcolm and Casey, Michael. Locality-sensitive hashing for finding nearest neighbors [lecture notes]. *Signal Processing Magazine, IEEE*, 25(2):128–131, 2008.