CHAPTER II

MUSIC INFORMATION RETRIEVAL FOR TRANSCRIPTION

Being able to accurately identify all musical events from audio and transcribe them into musical notations is an essential skill for musicians as well as a paramount goal of music machine learning research. Enabling an automatic conversion between musical audio and symbolic notations, automatic music transcription opens up many new possibilities.

Due to the complexity and difficulty of creating a completely end-to-end music transcription system, many existing approaches focus on a specific subtask of the problem (Casey et al., 2008), e.g. extracting onsets and beats, recognizing timbre and instruments, tracking monophonic and polyphonic pitches, or separating audio sources from a mixture. Each of these subtasks poses interesting goals and applications even without the lofty goal of end-to-end music transcription, and they are classified as subproblems of *music information retrieval* (MIR). Although this term has existed since 1960s (Kassler, 1966), it was only after the late 1990s when active research on this area has spun off from computer music and computational musicology literature. During the last two decades, numerous sophisticated and novel approaches for each of these subproblems have been introduced, that have continuously improved the performance in terms of the accuracy in predicting the correct annotations. This chapter starts by introducing the common concepts and techniques employed in many AMT models, followed by reviews of the state-of-the-art techniques in each area of music transcription.

The purpose of this chapter is to provide a survey over the history of MIR research related to automatic music transcription, as well as to show a clear common pattern over the areas of MIR where the machine learning models have been evolving from simple heuristics based on hand-crafted features to sophisticated deep learning models with millions of parameters. Many methods employing deep neural networks are referenced in this chapter, and the concepts and the formulation of those models such as convolutional newral networks (CNN) and recurrent neural networks (RNN) are described in detail in Chapter III.

1 Introduction

Audio data is huge in volume; a typical audio track contains 44,100 real-numbered samples per second, and sometimes even more. Therefore, computational methods for extracting musical information from audio usually contains a pipeline of feature extraction stages to reduce the dimensionality and increase the interpretability of input data, as shown in Figure 5. The pipeline includes a few techniques widely used in speech processing, as well as many feature extraction stages created for music-specific purposes.

While there are many MIR tasks that operate on the track level, such as music recommendation, tagging, and genre classification, most subtasks of music transcription involve the prediction of labels that are dependent on time, operating either in the sample-level or frame-level. Frames are created by taking a series of overlapping short-time audio segments, where the length of a segment typically ranges from 10 to 50 milliseconds, and optionally multiplying them by a window function. Taking discrete Fourier transforms on the frames produces a *short-time Fourier transform* (STFT), and the squared magnitude of an STFT gives a

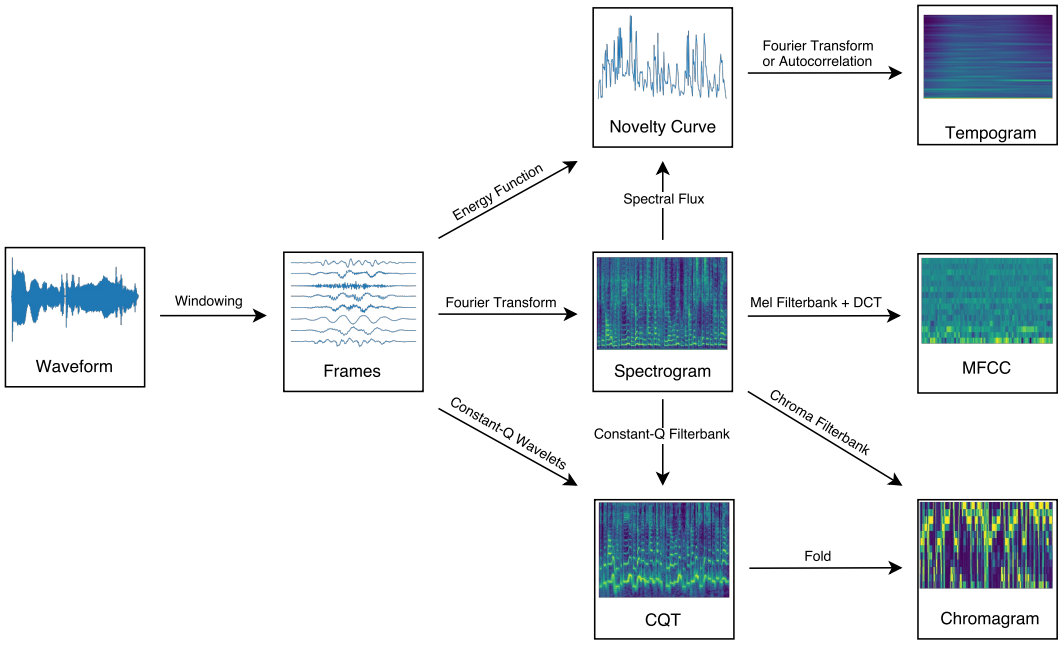

Figure 5: The standard pipeline for music feature extraction. An appropriate set of feature extraction methods needs to be heuristically selected depending on the task.

spectrogram; i.e. for a signal $x[n]$ and a window function $w[n]$:

$$\textbf{STFT}\{x[n]\}(m,\omega) = \sum_{n=-\infty}^{\infty} x[n]w[m-n]e^{-j\omega n}, \tag{1}$$

$$\textbf{Spectrogram}\{x[n]\}(m,\omega) = \left|\textbf{STFT}\{x[n]\}(m,\omega)\right|^2. \tag{2}$$

Spectrograms give very rich information about the audio; for example, the contour of melodies and the dynamics of music are usually identifiable from the spectrogram image. Spectrograms are expressive enough to be used as an output of sound synthesis or a source separation algorithm, and the corresponding audio signals can be reconstructed without incurring significant perceptual inconsistencies (Griffin & Lim, 1984; Le Roux et al., 2010). However, the dimensionality of a spectrogram is still quite high, making it computationally prohibitive to run many algorithms directly on an STFT or a spectrogram.

This necessitated further transformations by the means of filterbanks, such as *Mel-Frequency Cepstral Coefficients* (MFCC) (Logan, 2000) by applying the Mel filterbank inspired by the human auditory perception and taking the first few DCT components that contain independent factors describing the spectral shape. *Constant-Q transform* (CQT) (Schörkhuber & Klapuri, 2010) uses a filterbank where the center frequencies of filters have a constant Q factor, which is the ratio between the center frequency and the 3 dB bandwidth of a filter. By configuring CQT to produce 12 filters per octave, it is possible to obtain the coefficients corresponding to each musical tone, and to fold the representation to produce a *chromagram* (Harte & Sandler, 2005). Tonnetz (Harte et al., 2006) is a 6-dimensional feature space signifying harmonic relationships, and more sophisticated feature extractors include the summary ACF (Tolonen & Karjalainen, 2000), Specmurt (Saito et al., 2008), and bispectrums (Argenti et al., 2011).

To extract the beat and tempo information, a heuristic function, such as the first-order difference of the time-domain log energy function or the *spectral flux* that measures the total energy increase over the STFT frequency bins, is applied to formulate a novelty curve. This curve can then be used to measure energy bursts that are typically present in the onsets of notes (Bello et al., 2005). The onset information can be further processed to obtain tempo information via *tempogram* (Cemgil et al., 2000) or cyclic tempogram (Grosche et al., 2010).

Meanwhile, many recent approaches have successfully eliminated some or all feature transformation stages in the standard MIR pipeline by training a deep model directly on spectrograms or audio waveforms. Applications of deep learning arose in virtually all types of MIR tasks, including melody extraction (Bittner et al., 2017), beat tracking (Vogl et al., 2017), and genre classification (Oramas et al.,

2017), and outperformed previous feature-based approaches. Apart from a small number of end-to-end approaches, most deep learning models for music still rely on predefined feature transforms such as STFT or CQT, because those features leverage the prior knowledge that music signals are often harmonically sparse and make it easier for a model to learn meaningful concepts without overfitting, using a smaller number of parameters and hence being achievable in a limited hardware capacity. However, these feature extraction stages typically induce a loss of information, and the best-performing model would benefit most from the raw audio data, given enough amount of training data and hardware (Pons et al., 2018).

As is the case for feature extraction, musical prior knowledge is often applied to the algorithmic design of models as well, such as the assumption that sudden changes in music is rare and most changes happen gradually. In the time domain, median filtering (Oudre et al., 2009) is a simple heuristic that can suppress spurious changes, and *Hidden Markov models (HMM)* are widely employed for modeling sequence data such as chord progressions (T. Cho et al., 2010) as well as to smooth sequence outputs as a post-processing step (Khadkevich & Omologo, 2009). Prior knowledge about musical notes can be incorporated more specifically. These approaches include detecting onsets and offsets for note transcription (Benetos & Dixon, 2011), modeling note attacks and decays (Cheng et al., 2016), and more generally the temporal evolution of notes (Cogliati & Duan, 2015). In the frequency domain, the *spectral smoothness principle* states that the spectral envelopes of real sounds tend to be slowly varying as a function of frequency (Klapuri, 2003). The principle has been implemented in a number of ways, such as a moving-average filter for iterative source estimation and separation (Klapuri, 2003), a score function

for F0 candidate (Yeh et al., 2010), and a low-order autoregressive overtone modeling (Emiya et al., 2010).

We have reviewed the common feature extraction stages and identified how the musicological or mathematical prior knowledge on music data guides the algorithmic design of MIR models. In the following sections, more in-depth literature reviews on the different subtasks relevant to automatic music transcription are provided, starting from the simplest problem of monophonic picth tracking.

2 Monophonic Pitch Estimation

Monophonic pitch estimation, or pitch tracking, refers to the task of extracting the fundamental frequency (F0) values from monophonic audio signals. Formally, pitch is defined as a subjective quality of perceived sounds and does not precisely correspond to the physical property of the fundamental frequency (Hartmann, 1997). However, apart from a few rare exceptions, pitch can be quantified using fundamental frequency, and thus they are often used interchangeably in the MIR literature outside psychoacoustical studies. Since differentiating the physical and the perceptual aspects of pitch is a non-goal, the two terms are interchangeably throughout this thesis as well.

Computational methods for monophonic pitch estimation have been studied for more than a half-century (Noll, 1967), and many reliable methods have been proposed since. Earlier methods commonly employ a certain candidate-generating function, accompanied by pre- and post-processing stages to produce the pitch curve. Those functions include the cepstrum (Noll, 1967), the autocorrelation function (ACF) (Dubnowski et al., 1976), the average magnitude

difference function (AMDF) (Ross et al., 1974), the normalized cross-correlation function (NCCF) as proposed by RAPT (Talkin et al., 1995) and PRAAT (Boersma, 1993), and the cumulative mean normalized difference function as proposed by YIN (de Cheveigné & Kawahara, 2002). More recent approaches include SWIPE (Camacho & Harris, 2008), which performs template matching with the spectrum of a sawtooth waveform, and pYIN (Mauch & Dixon, 2014), a probabilistic variant of YIN that uses a Hidden Markov Model (HMM) to decode the most probable sequence of pitch values. According to a few comparative studies, the state of the art is achieved by YIN-based methods (von dem Knesebeck & Zölzer, 2010; Babacan et al., 2013), with pYIN being the best performing method to date.

Since the methods for monophonic pitch tracking are usually built based on the assumption that at most one pitch is present at a time, they cannot be directly applied to polyphonic music where multiple concurrent notes and sound sources are present. Different approaches are therefore needed to accurately estimate multiple concurrent pitches, and such methods are reviewed in the next section.

3 Multiple Fundamental Frequency Estimation

Among the subtasks of automatic music transcription, estimating and tracking all pitches from a polyphonic recording poses the most difficult challenges, as apparent from the recent stream of results from MIREX challenges (Downie et al., 2014). The task is commonly referred to as *multiple fundamental frequency estimation* (Multi-F0 estimation, or MFFE) or *multi-pitch estimation* (MPE). This task is in some sense a superset of other MIR tasks like onset detection, beat tracking, chord recognition, and melody extraction, since the frequency tracking task has to

indicate the presence of every pitch, and tracking chords and melodies becomes much easier when the correct annotations for all pitch values are available.

Early approaches for polyphonic pitch tracking are often based on rather strict assumptions on the types of timbre and the number of polyphony in the audio to be transcribed (Moorer, 1977; Piszczalski & Galler, 1977). *Blackboard systems* (Martin, 1996; Dixon, 2000) are one of the first methods that enabled polyphonic transcription to work under milder assumptions, by integrating both signal processing and musicological knowledge into hierarchical problem abstraction. Limitations of the blackboard approach include the overall complexity and the exhaustive searches required by the system. Accordingly, later approaches often use statistical models or factorization-based methods to more efficiently capture pitch information.

Non-negative matrix factorization (NMF) (Lee & Seung, 1999, 2001) refers to an algorithm that finds two matrices that factorize a given matrix where the elements of all three matrices are non-negative. Starting with Smaragdis and Brown (2003), many successful methods for music transcription have been built based on NMF (Benetos et al., 2019). These methods commonly take the approach of factorizing a time-frequency representation $\mathbf{X} \in \mathbb{R}_{+}^{F \times T}$ as a product of a dictionary matrix $\mathbf{D} \in \mathbb{R}_{+}^{F \times K}$ and an activation matrix $\mathbf{A} \in \mathbb{R}_{+}^{K \times T}$, where K is the number of pitch labels to be transcribed, e.g. 88 keys for piano transcription. This allows for an intuitive interpretation of each matrix, where each column of $\mathbf{D}$ contains a spectral template for a pitch label, and each row of $\mathbf{A}$ contains the activation of the corresponding pitch over time. Various extensions of factorization-based methods have been proposed to leverage sparsity (Abdallah & Plumbley, 2004; Cont, 2006; Costantini et al., 2013), non-negative matrix division

(Niedermayer, 2008), β-divergence (Dessein et al., 2010), adaptive estimation of harmonic spectra (E. Vincent et al., 2010; Fuentes et al., 2013), a Bayesian framework for encouraging harmonicity and temporal smoothness (Bertin et al., 2009, 2010; Peeling et al., 2010), and modeling of attack and decay sounds (Benetos & Dixon, 2013; Ewert & Sandler, 2016). With carefully designed regularizers and the *alternating direction method of multipliers (ADMM)* for training NMF, Ewert and Sandler (2016, 2017) achieved close-to-perfect accuracy for piano transcription in a studio setting, i.e. when the exact spectral profile of the piano notes to be transcribed is known in advance.

A probabilistic approach closely related to NMF is *probabilistic latent semantic analysis (PLSA)* (Hofmann, 1999), a simple probabilistic graphical model that factorizes the joint probability distribution of time and frequency into conditional distributions involving a latent semantic factor. PLSA is is equivalent to NMF when the KL divergence is minimized under L_1 normalization (Gaussier & Goutte, 2005; Ding et al., 2008), while the probabilistic framework enables statistical learning and introducing transformation invariances (Smaragdis et al., 2006). In (Smaragdis, 2009; Benetos & Dixon, 2012), a shift-invariant extension to PLSA has been applied to multi-instrument MFFE by using a convolution over constant-Q spectrograms. Grindlay and Ellis (2010) proposed an extension to PLSA based on the *subspace NMF* algorithm (Grindlay & Ellis, 2009) for multi-instrument MFFE, incorporating additional parameters for instrument sources and possible pitch values.

Many probabilistic models for music transcription employ a Bayesian framework, in which the audio signal is modeled using a probability distribution conditional on unobserved variables such as the note frequencies and timing. The transcription is then performed through Bayesian inference on those parameters,

using either *Markov-chain Monte Carlo (MCMC)* or *variational inference*. Bayesian models for music transcription are usually designed with prior and conditional distributions incorporating musicological and acoustical knowledge on the music and the instruments to be transcribed, such as the de-tuning of partials (Davy & Godsill, 2003). Existing approaches in directions include a state-space model for musical harmonics (Cemgil et al., 2003), a decomposition of audio signals into Gabor atoms (Davy et al., 2006), a hierarchical graphical model (Pesek et al., 2017), and an overtone corpus encoding the harmonic structure (Sakaue et al., 2013). A nonparametric Bayesian model proposed in (Yoshii & Goto, 2012) employs infinite Gaussian mixtures and noninformative hyperprior distributions, automating the model selection process. Bayesian models provide a powerful tool connecting the whole generative process of music, but usually suffer from the high computational cost and the complexity of the algorithm.

Unlike the aforementioned approaches which incorporate as much prior knowledge on the music as possible, discriminative models employ a simpler approach of learning a direct mapping between the audio features and the labels from large training data, making minimal assumptions on the acoustical or musical structure. This data-driven approach is made possible by the availability of large datasets and increased computational capabilities. *Support vector machines (SVM)* were commonly used in earlier discriminative approaches, by constructing one-versus-all SVM classifiers on STFT magnitudes (Poliner & Ellis, 2006), on learned features using *deep belief networks (DBN)* (Nam et al., 2011), or on the result of NMF (Weninger et al., 2013). While being flexible and straightforward to be applied on any features of choice, SVMs have high time and space complexities which limit the size of training dataset and hence the capability of the model.

Deep learning (LeCun et al., 2015) methods for music transcription are increasingly popular (Benetos et al., 2019), as larger labeled datasets and more powerful hardware become accessible. These approaches commonly employ *neural networks (NN)* to produce music transcriptions from the input audio representation, and these are relatively recent phenomena that started in the last decade, with a notable exception of Marolt (1999, 2004). Nam et al. (2011) used deep belief networks (G. Hinton et al., 2006) to extract audio features which are subsequently fed to pitch-wise SVM-HMM pairs to predict the target piano rolls. More recent approaches are based on *convolutional neural networks (CNN)* and/or *recurrent neural networks (RNN)*. Transcription models using CNNs are relatively simpler to train and deploy since they can be easily applied in a frame-wise manner in parallel, and they are shown effective especially for the cases where the notes mostly contain sustained sounds, such as bowed strings, wind instruments, and vocals (Kelz et al., 2016; Bittner et al., 2017). The sequential nature of RNNs is suitable for modeling the temporal variations in music, and many neural transcription models include recurrent connections in their architecture. RNN architectures proposed for AMT include bidirectional RNNs (Böck & Schedl, 2012), recurrent temporal restricted Boltzmann machines (Boulanger-Lewandowski et al., 2012), an acoustic model combined with an RNN-based music language model (Sigtia et al., 2015, 2016; Q. Wang et al., 2018), a sequence-to-sequence model (Ullrich & Van Der Wel, 2017), a dual-objective loss for onsets and frames (Hawthorne et al., 2018), and a *convolutional-recurrent neural networks (CRNN)* (Thomé & Ahlbäck, 2017), which scored the top accuracy in the MFFE subtask of the MIREX 2017 competition.

A few recent studies tried to identify the limitations of simply improving

the conventional metrics and argued the need for more systematic analysis and assessment of the music transcription problem. These include a study of invariances under data augmentation (Thickstun et al., 2018), a study of the entanglement of note representations that may prevent accurate predictions for unseen combinations of notes (Kelz & Widmer, 2017), and a musically inspired evaluation metric that also takes account of voice separation and harmonic analysis (Mcleod & Steedman, 2018).

4 Source Separation and Music Translation

Source separation refers to the task of separating sound sources from a mixture signal, and is closely related to automatic music transcription, because it provides a means to separate each instrument sources from multi-instrument music. This problem is also called as a *cocktail party problem*, based on humans' ability to focus on a single voice at a noisy cocktail party. Sound source separation has been a popular research topic since the seminal work on *auditory scene analysis* by Bregman (1990), from which stemmed computational auditory scene analaysis (CASA) (Brown & Cooke, 1994), a problem of using computational models to analyze an auditory scene, identifying the sources and location of all nearby sounds. In this sense, automatic music transcription and sound source separation are particular aspects of auditory scene analysis (Plumbley et al., 2002), and source separation enables similar kinds of applications to AMT, such as music editing, 3D sound rendering, and information retrieval systems.

Blind source separation refers to the situation where no information about the sources or the mixing process is known (Bell & Sejnowski, 1995), whereas *informed source separation* (E. Vincent et al., 2014) concerns the case where some level of

side information is available, e.g. the presence of a score (Ewert et al., 2014). Blind sound source separation has to resort to using purely statistical approaches such as indepenent component analysis (Saruwatari et al., 2006) or robust PCA (P. S. Huang et al., 2012), whereas informed source separation can leverage the knowledge on the musical structure (Rafii & Pardo, 2013; Liutkus et al., 2012) or the timbral differences of the sources (Y. Li & Wang, 2007; Ono et al., 2010) for better separation. Probabilistic models for source separation (Ozerov et al., 2007; Leglaive et al., 2016) have also been developed, and source-filter modeling (Heittola et al., 2009; Durrieu et al., 2011) is a generative approach which separately models a source that creates a sound and a filter that shapes the timbre. The proposed AMT model is similar to source-filter models in a sense that it describes the generative process of each sound, but also relates to timbre-informed source separation models (Miron, 2018), because it needs to learn the concept of timbre to produce per-instrument piano rolls.

A closely related problem to source separation is audio translation, which concerns mapping input audio to a corresponding output with some desired properties, such as speech with reduced noise, singing voice separated from music, or the same speech content in the voice of a different speaker. Barry and Kim (2018) applied the style transfer algorithm (Gatys et al., 2015) to an ensemble of STFT, CQT, and Mel spectrograms, to transfer musical styles capturing harmonic, rhythmic, and timbral elements. The *U-Net* architecture (Ronneberger et al., 2015) uses an encoder-decoder framework with skip connections between the hidden layers at the same level of abstraction to perform image translation, and a singing voice separation model can be trained using this architecture (Jansson et al., 2017). The encoder-decoder architecture with skip connections can also be trained with

GAN objectives, and a few audio translation models working on spectrograms have been developed; examples include singing voice separation (Fan et al., 2017; Stoller et al., 2017), source separation (Subakan & Smaragdis, 2017), and speech enhancement (Pascual et al., 2017; Donahue et al., 2017).

5 Machine Learning Models for Music Synthesis

The recent deep generative models have been very successful in synthesizing breathtakingly high-quality audio signals. We would want the synthesized music and audio signals to capture the long-term dependencies such as beats, measures, and chord progressions that ranges up to a few seconds, while the raw audio signals typically have the order of 10 thousand sameples per second. This made end-to-end synthesis models more difficult to train than image synthesis and translation models which it usually suffices to capture dependencies ranging a few hundred pixels. SampleRNN (Mehri et al., 2017), to be discussed in Chapter III in the context of deep autoregressive models, is one of the first successful deep generative models for audio and formed a basis for the techniques used by Lyrebird, an AI startup founded by University of Montréal students that provides API for synthesized voice of a specific person, e.g. Barack Obama. WaveNet (van den Oord, Dieleman, et al., 2016), developed by Google DeepMind, uses a causal architecture using dilated convolutions to generate time-domain audio samples, and is able to produce realistic human voices and piano sounds. WaveNet learns acoustically meaningful representations including pitch and spectral features (Hua, 2018). There also exist faster approaches using recurrent neural networks to produce vocal and musical audio, as found in (Nayebi & Vitelli, 2015) and (Kalingeri & Grandhe, 2016), albeit with lower quality when compared to WaveNet. Tacotron (Y. Wang

et al., 2017; Shen et al., 2018) is a fully end-to-end speech synthesizer that works directly on a sequence of characters, which can learn the pronunciation of unseen complex words and different ways of reading the same word according to the phrase semantics and punctuations. A newer RNN-based model called WaveRNN (Kalchbrenner et al., 2018) is capable of generating audio that matches WaveNet in quality, yet with an enough efficiency to be able to run real-time on GPUs or even on mobile phones. WaveGlow (Prenger et al., 2019) uses a flow-based approach to synthesize waveform samples and is also shown to produce WaveNet-quality audio while being able to run efficiently in parallel. A singing synthesis model (Blaauw & Bonada, 2017) based on the WaveNet architecture is also capable of synthesizing voice parametrically, separating the influence of pitch and timbre in the model.

Generative Adversarial Networks (GANs), also to be reviewed extensively in Chapter III, have also been used as generative models for audio. A GAN architecture using one-dimensional convolutions called *WaveGAN* was introduced by Donahue, McAuley, and Puckette (2018) and is capable of generating 1-second audio segments from the latent representations. In a newer approach called *GANSynth* (Engel et al., 2019), the GAN architecture was used for generating magnitude spectrograms together with the corresponding two-dimensional representation of instantaneous frequencies, which produced significantly more stable output compared to WaveGAN. Training GANs for arbitrary-length audio sequences and extending it as a tool for disentanglement of latent semantic information or a conditional audio synthesis framework remains a challenge.

6 Music Language Models for Symbolic Music Generation

Symbolic music processing refers to the techniques for processing music at a symbolic level, such as in the form of sheet music, MIDI signals, or piano roll representations. Problems in this domain include optical music recognition (Rebelo et al., 2012), algorithmic composition (Fernández & Vico, 2013), and computational music theory (Hamanaka et al., 2013), while the subject most relevant to music transcription research would be *music language models*. A music language model is a statistical model, often a generative model, that encodes music theoretic knowledge to describe the structural composition and arrangement of musical elements (Patel, 2008), similarly to how computational linguists build language models to describe the structure of natural languages. A well-designed music language model can be an important component for a generative model for music, because it can serve as a prior for latent representations and can be combined with conditional synthesis models or software instruments to produce audio.

The first systematic approach of applying a linguistic theory to music was the *generative theory of tonal music* (Lerdahl & Jackendoff, 1983), which was inspired by Noam Chomsky's generative grammar (Chomsky, 1966) and was influential in music theory, music psychology, and cognitive musicology. Music language models are loosely connected to this idea of generating music according to its grammar, and its implementations typically use statistical models that are also used in natural language processing. These include many kinds of approaches for symbolic music generation, such as hidden Markov models (Farbood & Schoner, 2001), generative grammars (Chemilier, 2001), cellular automata (Burraston et al., 2004), and genetic algorithms (Miranda et al., 2007). More recently, deep

learning models such as recurrent neural networks have been used to build music language models (Sigtia et al., 2014). The latest approaches to generate realistic music sequences in the symbolic domain include an application of variational autoencoder (Teng et al., 2017; Tikhonov & Yamshchikov, 2017), a generative adversarial network (Yang et al., 2017), and a transformer (C.-Z. A. Huang et al., 2019).

7 Summary

In this chapter, a broad range of MIR techniques related to automatic music transcription have been discussed, in the fields of monophonic and polyphonic pitch tracking, source separation and music translation, music synthesis, and music language models. A clear observation in each subtask of AMT is that many recent methods employ deep learning models, and this is because deep models have more flexibility and capacity to learn complex statistical relations of interest. In order to build the solid foundation of the various deep learning techniques used throughout this thesis, Chapter III will provide an extensive review on the various techniques and models that are collectively classified as deep learning.