Acoustically Inspired Probabilistic Time-domain Music Transcription and Source Separation

Pablo Alejandro Alvarado Duran

Submitted in partial fulfillment of the requirements
of the Degree of Doctor of Philosophy

School of Electronic Engineering and Computer Science
Queen Mary University of London

February 28, 2020

Statement of Originality

I, Pablo Alejandro Alvarado Duran, confirm that the research included within this thesis is my own work or that where it has been carried out in collaboration with, or supported by others, that this is duly acknowledged below and my contribution indicated. Previously published material is also acknowledged below.

I attest that I have exercised reasonable care to ensure that the work is original, and does not to the best of my knowledge break any UK law, infringe any third party's copyright or other Intellectual Property Right, or contain any confidential material.

I accept that the College has the right to use plagiarism detection software to check the electronic version of the thesis.

I confirm that this thesis has not been previously submitted for the award of a degree by this or any other university.

The copyright of this thesis rests with the author and no quotation from it or information derived from it may be published without the prior written consent of the author.

Signature:

Date: July 1, 2019

Details of collaboration and publications:

- Sparse Gaussian process audio source separation using spectrum priors in the time-domain. In 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) [6].
- Gaussian processes for music audio modelling and content analysis. In 2016 IEEE International Workshop on Machine Learning for Signal Processing (MLSP) [7].
- Efficient learning of harmonic priors for pitch detection in polyphonic music. arXiv preprint arXiv:1705.07104, 2017 [8].

Abstract

Automatic music transcription (AMT) and source separation are important computational tasks, which can help to understand, analyse and process music recordings. The main purpose of AMT is to estimate, from an observed audio recording, a latent symbolic representation of a piece of music (piano-roll). In this sense, in AMT the duration and location of every note played is reconstructed from a mixture recording. The related task of source separation aims to estimate the latent functions or source signals that were mixed together in an audio recording. This task requires not only the duration and location of every event present in the mixture, but also the reconstruction of the waveform of all the individual sounds. Most methods for AMT and source separation rely on the magnitude of time-frequency representations of the analysed recording, i.e., spectrograms, and often arbitrarily discard phase information. On one hand, this decreases the time resolution in AMT. On the other hand, discarding phase information corrupts the reconstruction in source separation, because the phase of each source-spectrogram must be approximated. There is thus a need for models that circumvent phase approximation, while operating at sample-rate resolution.

This thesis intends to solve AMT and source separation together from an unified perspective. For this purpose, Bayesian non-parametric signal processing, covariance kernels designed for audio, and scalable variational inference are integrated to form efficient and acoustically-inspired probabilistic models. To circumvent phase approximation while keeping sample-rate resolution, AMT and source separation are addressed from a Bayesian time-domain viewpoint. That is, the posterior distribution over the waveform of each sound event in the mixture is computed directly from the observed data. For this purpose, Gaussian processes (GPs) are used to define priors over the sources/pitches. GPs are probability distributions over functions, and its kernel or covariance determines the properties of the functions sampled from a GP. Finally, the GP priors and the available data (mixture recording) are

combined using Bayes' theorem in order to compute the posterior distributions over the sources/pitches.

Although the proposed paradigm is elegant, it introduces two main challenges. First, as mentioned before, the kernel of the GP priors determines the properties of each source/pitch function, that is, its smoothness, stationariness, and more importantly its spectrum. Consequently, the proposed model requires the design of flexible kernels, able to learn the rich frequency content and intricate properties of audio sources. To this end, spectral mixture (SM) kernels are studied, and the Matérn spectral mixture (MSM) kernel is introduced, i.e. a modified version of the SM covariance function. The MSM kernel introduces less strong smoothness, thus it is more suitable for modelling physical processes. Second, the computational complexity of GP inference scales cubically with the number of audio samples. Therefore, the application of GP models to large audio signals becomes intractable. To overcome this limitation, variational inference is used to make the proposed model scalable and suitable for signals in the order of hundreds of thousands of data points.

The integration of GP priors, kernels intended for audio, and variational inference could enable AMT and source separation time-domain methods to reconstruct sources and transcribe music in an efficient and informed manner. In addition, AMT and source separation are current challenges, because the spectra of the sources/pitches overlap with each other in intricate ways. Thus, the development of probabilistic models capable of differentiating sources/pitches in the time domain, despite the high similarity between their spectra, opens the possibility to take a step towards solving source separation and automatic music transcription. We demonstrate the utility of our methods using real and synthesized music audio datasets for various types of musical instruments.

Acknowledgements

I would like to thank my family for always supporting me, especially my parents Alberto and Martha, my sister Sara, and my girlfriend Laura. Their unconditional support has been the soil and light that encourages me to grow. Thank you to all the people who contributed positively during my studies and made the everyday PhD student life more bearable, especially, Mauricio Álvarez and Dan Stowell. Thank you to my friends Katrin Frisch, Léna Delval, Maria Panteli, Mi Tiang, Julian Osmalskyj, Delia Fano, Saumitra Mishra, Changhong Wang, Juan José Giraldo, Fariba Yousefi, Cristian Guarnizo, Carlos David Zuluaga, Andrés Felipe López-Lopera, and Geusseppe González. Please forgive me if you read this and do not find your name here. If you consider me to be a friend, I would consider you to be a friend too.

I also want to thank the PhD process itself. This research has helped me to trust my judgment and to realise that every theory and method is susceptible to questioning and challenging. Also, I learnt there is always an inherent amount of subjectivity when interpreting the outcome of an experiment or phenomenon. Therefore, there is not one single and absolute truth. I have learned to doubt.

Moving abroad to study a PhD has been the most challenging experience I have ever had. Looking back over the last four and a half years, I was not aware of how much my life was going to change. Leaving my family, friends, culture, language, weather, and lifestyle has made me question my identity and abilities but mainly my intelligence. Back home, I used to consider myself a successful academic and promising researcher. My ambition was to achieve academic titles, to publish research papers and to become a well-

known researcher. Here in the UK I struggled to carry on, find hope and my research no longer felt important to me. I learnt the importance of enjoying again every day life in a place distant from home. The feeling of companionship, belonging and the kindness of colleagues took precedence over my research. I learnt to love, feel loved, to laugh and to dream again. I remembered all that I had left behind while pursuing my academic aspirations and I realised, fulfilling these simple needs was the key to have balance. Unsurprisingly, I acknowledged I was a human being and not a *research-paper* machine. During this research, I have developed a deeper understanding of the topics that interest me, that is, Gaussian processes and music signal processing. But more importantly, this journey has taught me a lot about life itself.

Licence

List of abbreviations

AMT	Automatic Music Transcription
dB	Decibel
ELBO	Evidence Lower Bound
F0	Fundamental frequency
FL	Learning in the Frequency Domain
FT	Fourier Transform
GP	Gaussian Process
IS-NMF	Itakura-Saito NMF
KL	Kullback-Leibler
KL-NMF	Kullback-Leibler NMF
LD-PSDTF	Positive Semi-Definite Tensor Factorization
LOO	Leave One Out
LOO-SIG	Leave One Out Sigmoid
MAPS	MIDI Aligned Piano Sounds
MIDI	Musical Instrument Digital Interface
min	minutes
ML	Marginal Likelihood
MSE	Mean Squared Error
MSM	Matérn Spectral Mixture
NMF	Non-negative Matrix Factorization
PLCA	Probabilistic Latent Component Analysis
RBF	Radial Basis Function
RMSE	Root Mean Squared Error
SAR	Source to Artefacts Ratio
SDR	Source to Distortion Ratio
SE	Squared Exponential
SIG	Sigmoid
SIR	Source to Interferences Ratio
SM	Spectral Mixture

SOF	Softmax
SSGP	Source Separation Gaussian Process
SVI	Stochastic Variational Inference
TM	Manual Tuning
VI	Variational Inference
VFF	Variational Fourier Features
WSS	Wide Sense Stationary

Mathematical notation

$\mathbf{y}$	Audio signal (column vector)
$\mathbf{x}$	Column vector of input variables
$\mathbf{f}$	Column vector of size N
cov	Covariance
$\mathbf{K}$	Covariance matrix
$\mathcal{L}$	Evidence lower bound
$\mathbb{E}$	Expected value
exp	Exponential
Y	Fourier transform of $\mathbf{y}$
$\mathcal{GP}$	Gaussian process
$\mathbf{I}$	Identity matrix
KL	Kullback-Leibler divergence
$\mathcal{N}$	Normal distribution
M	Number of inducing variables
N	Number of observations
$f(\cdot)$	Random function
$\mathrm{tr}(\cdot)$	Trace of a matrix
var	Variance
$q(\cdot)$	Variational distribution

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

Source separation and multi-pitch detection are quite active areas of research within the audio signal processing community. For example, source separation is useful in automatic speech recognition for isolating voice from background noise [40]. Also, pitch detection finds applications in speech [80], automatic music transcription [67], and melody extraction [60]. From a Bayesian perspective, these two tasks consist of updating prior knowledge of underlying processes hidden in the data. Source separation reconstructs latent signals that were mixed in an audio recording. Similarly, multi-pitch detection retrieves the underlying symbolic representation (musical score, e.g. piano roll) of a piece of music.

State-of-the-art methods for source separation and multi-pitch detection commonly work on a time-frequency representation of the input raw audio. In short, these methods first transform the input waveform into a magnitude spectrogram, before performing the separation/detection task. For example, approaches based on deep neural networks [67], non-negative matrix factorization [78], and probabilistic latent component analysis [12, 30], exhibit this pipeline. Although most researchers have extensively adopted this perspective, there are several potential disadvantages of working on time-frequency representations.

To illustrate the limitations of working with the spectrogram, consider the problem of reconstructing the waveform sources from their corresponding estimated spectrograms. To do so, the phase of each source spectrogram needs to be approximated, corrupting the reconstruction. Also, in multi-pitch detection, working with the spectrogram means that the time level resolution is lost, introducing errors in the onset and offset times of the detection. These challenges have motivated the development of new approaches that operate directly on the input data waveform. Indeed, previous research suggests that time-domain methods can circumvent phase approximation while achieving time level resolution [87]. Still, current time-domain methods require further developments before they can become widely used.

This thesis focuses explicitly on time-domain Bayesian approaches based on Gaussian processes. Following a Bayesian approach means to specify first a prior over the target variables, and then update it with observed data, that is, to obtain a posterior. Here, the target variables are either the *source signals* in source separation or the *activation* of each pitch in multi-pitch detection. In both cases, the target variables are *functions* of time. Therefore, both tasks need introducing priors over functions directly. Here, Gaussian processes (GPs) are the mathematical tools that answer to this necessity. A Gaussian process is a generalization of the multivariate normal distribution [56]. Moreover, as we will introduce shortly, GPs represent probability distributions over functions.

Although time-domain source separation and multi-pitch detection models based on Gaussian processes have compelling advantages, these methods face two main challenges. First, the prediction in GP models depends profoundly on the chosen prior. Second, GPs are intractable for large audio signals, as the computational complexity of inference scales cubically with the data size. Specifically, evaluating the likelihood and computing the posterior distribution requires to invert a dense matrix. The complexity of the standard approach for matrix inversion is $\mathcal{O}(n^3)$. On the other hand, the Strassen's algorithm [72] has complexity $\mathcal{O}(n^{2.8})$, which it is also intractable for large datasets ($n \gg 1 \times 10^4$).

1.2 Aim

This research aims to develop Bayesian machine learning methods that interpret source separation and multi-pitch detection as a single unifying task. Moreover, the proposed methods are intended to explain the raw waveform of single-instrument music recordings directly; that is, they should work in the time-domain. The reason is that the unprocessed audio data by itself contains all the knowledge available in a music recording, in contrast to transforming the audio waveform into a spectrogram, which often induces loss of information.

Following the Bayesian paradigm, this work requires the development of suitable Gaussian process priors able to encode the fundamental properties of acoustic signals. That is, smoothness, periodicity, spectral content, and non-stationary amplitude. Also, the audio signal processing tasks of source separation and multi-pitch detection demand methods that are data-efficient. consider, for instance, the possibly millions of data samples present in one single music recording. Consequently, this work also requires the introduction of inference approaches that make the proposed Bayesian methods scalable.

1.3 Thesis structure

Chapter 2 introduces the fundamental concepts and relevant research which will serve as the building blocks of this thesis. First, it describes the tasks of single-channel audio source separation and multi-pitch detection. Then, it presents how to use Gaussian processes (GPs) for machine learning regression, emphasising on how to design meaningful and valid priors/kernels. Finally, this chapter concludes by discussing how (stochastic) variational inference (VI) enables GP models for large music recordings.

Chapter 3 presents a first attempt to develop a time-domain multi-pitch detection model based on Gaussian processes. This method relies on deterministic and parametric activation functions, with Gaussian pro-

cesses explaining the harmonic behaviour of the pitches. Here, experiments study the relationship between choosing a specific kernel and the performance of the GP multi-pitch detection model.

Chapter 4 focuses on developing further the method presented in the previous section. For this purpose, this chapter introduces three main changes. First, the activation functions go from being parametric to becoming stochastic processes, explicitly, GPs. Second, instead of using generic kernels for describing spectral content, we propose to use the Matérn spectral mixture kernel. A subsection introduces the compelling properties and mathematical derivation of this kernel. Finally, in this model, the observed audio data is described as the sum of products of two GPs. Consequently, the posterior does not have a closed-form. Therefore this chapter concludes by showing experiments applying approximate variational inference to learn both, the hyperparameters and the posterior.

Section 5.1 This section investigates time-domain source separation models based on Gaussian processes. The evaluation metrics for this task demand to reconstruct the source functions with a higher degree of exactness, in contrast to multi-pitch detection. To this end, the proposed method first frame the input music recording, and then analyses each window individually. Besides, we suggest initialising the kernel of each source by using the empirical autocorrelation of isolated source recordings. Also, to learn the model hyperparameters, we propose to maximise a marginal likelihood lower bound.

Section 5.2 This section investigates the application of the proposed methods in the scenario where 88 pitches need to be detected/separated. Here, the aim is to carry out both tasks simultaneously; that is, to identify pitches but also to reconstruct the source signal corresponding to each pitch. This requires the usage of stochastic variational inference (SVI). This chapter presents some preliminary results.

Chapter 6 concludes the thesis by drawing comparisons between the exper-

iments and methods proposed throughout this research. This section also discusses future work.

1.4 Contributions

The main contributions of this thesis are:

Chapter 3: A semi-parametric approach for multi-pitch detection that operates in the time-domain. This method relies on Gaussian process regression and parametric/deterministic activation functions.

Chapter 4: A fully nonparametric Bayesian method for time-domain multi-pitch detection. Here, the activation functions follow stochastic processes inferred directly from the audio data.

Chapter 4: Similarly to the spectral mixture kernel proposed by Wilson in [83], we introduce the mathematical derivations of the Matérn Spectral Mixture kernel.

Chapter 4: A methodology for initialising spectral mixture kernels, to make them suitable for the spectral content of music notes.

Section 5.1: The development of an efficient approach for time-domain Gaussian process source separation. This model works on a windowed version of the mixture audio data and optimises an evidence lower bound to learn hyperparameters. The covariance functions used by this model resemble the empirical autocorrelation of isolated sounds corresponding to the training data of each source.

Section 5.1: The development of a Python package called `GPitch` for source separation and multi-pitch detection in the time domain. The available code works currently on single-instrument music recordings.

Section 5.2: The introduction of stochastic variational inference methods into multi-pitch detection GP models, allowing to use these methods in large audio signals.

1.5 Associated publications

Portions of the research presented in this thesis have been published in international conferences and workshops, as follows

- **Chapter 3:** Presented at the 2016 IEEE International Workshop on Machine Learning for Signal Processing (MLSP 2016) [7].
- **Chapter 4:** An early version of this work was released on arXiv.org e-Print archive (2017) [8].
- **Section 5.1:** Published in the 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2019) [6].

Chapter 2

Background

This chapter introduces the main concepts and research related to the aim of this thesis. It starts by defining what audio signals are, and how to represent them as waveforms. Then, this chapter illustrates a more specific type of acoustic signal: the polyphonic music recording. Also, it introduces two forms of music representations: the music-score, and the piano-roll. These concepts are essential to understand the two main areas of application of this work, that is, automatic music transcription or multi-pitch detection, and source separation. The subsequent section describes multi-pitch detection and automatic music transcription. It first explains what pitch is and why it is challenging to detect pitches in polyphonic music signals. Next, this chapter illustrates the task of source separation. Similarly, it first defines what a source is and why it is challenging to separate sources in polyphonic music recordings. Then, the following section details the mathematical and probabilistic foundations of the machine learning methods proposed in this thesis. Explicitly, it introduces Gaussian processes and how to apply them for Bayesian modelling. The chapter concludes by discussing how the research of this thesis fits into the state-of-the-art of multi-pitch detection and source separation.

2.1 Audio signals

In general, the term *audio* alludes to recording, reproducing, transmitting, and storing sound [74]. In this thesis, an *audio signal* refers solely to the data captured by a microphone when registering pressure fluctuations in the surrounding air (Figure 2.1). Further, the term audio signal is used interchangeably with *acoustic signal*. Although these terms cover any sound, for example, music, speech, bird songs, and street noise, here it refers mainly to single-instrument music recordings.

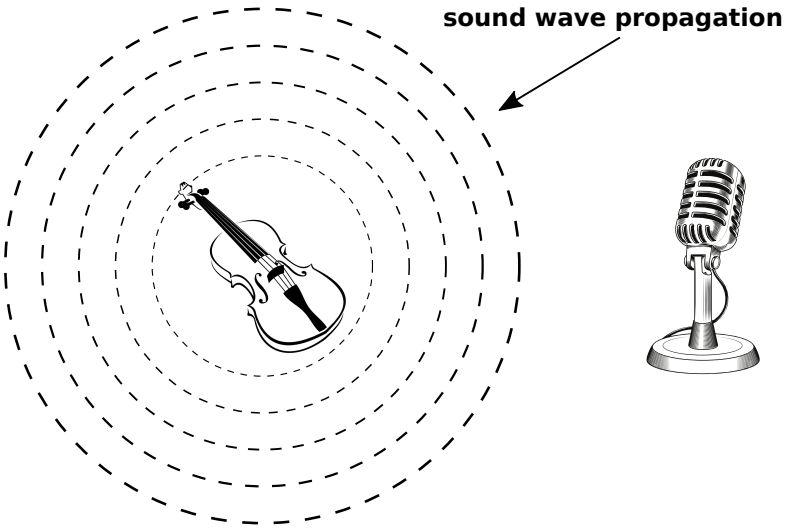


Figure 2.1: Audio signal generation.

Any object vibrating at frequencies within the limits of human hearing (20Hz to 20kHz) produces sound. These oscillations provoke displacement of air molecules. The repeated pattern by which the molecules contract and expand propagates through the air as a wave. Therefore, a way to represent sound at a particular location in the space is by a pressure-time function, also known as the *waveform* of a sound. For example, Figure 2.2 shows two seconds of the waveform of a single piano note recorded using a microphone. The small box inside Figure 2.2 shows 20 milliseconds of the same waveform. In short, the waveform is a function of time that characterises air pressure variations at a certain point. In this thesis, the terms waveform, acoustic signal, and audio recording are all equivalent.

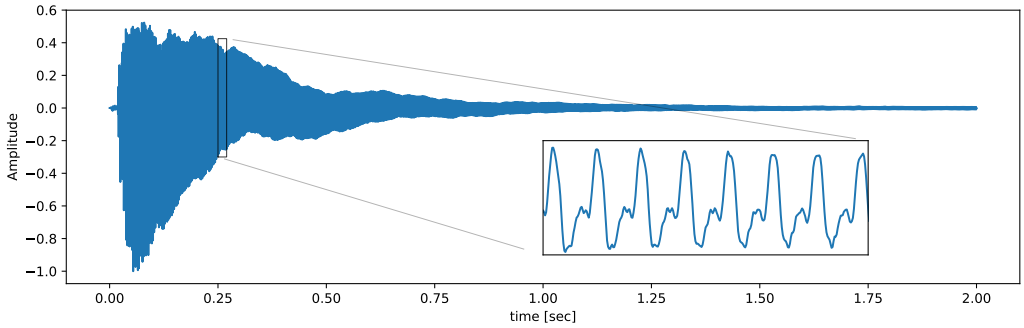


Figure 2.2: Two seconds of a piano waveform. The small box inside corresponds to zooming in on a 20 milliseconds window.

Pitch

Pitch is a subjective perception of the frequency content of a sound [48]. The pitch is what enables a listener to distribute sounds on a scale ranging from low to high [53]. Although the pitch is associated with an attribute of the auditory sensation, there is a very strong relationship between pitch and the fundamental frequency (F0) of a harmonic sound. The F0 is a quantitative property, measured in Hertz or cycles/second. Therefore, this work often uses *pitch* to refer to the fundamental frequency of the sounds present in a music signal [23].

2.1.1 Music signals

A *music signal* refers to any audio recording related to the interpretation of a piece of music. In a general sense, this could include several instruments playing at the same time. This thesis addresses solely single-instrument music signals, for example, an audio recording of only one violin or piano. Single-instrument music signals fall into two groups. The first one corresponds to pieces of music where only one *note* or *pitch* occurs at a time. This situation happens in a melody, for example, when a single person sings. The second group corresponds to *polyphonic* music signals, that is, recordings where more than one pitch or note take place simultaneously. For instance, in a piano interpretation, if a musician presses more than one key at the same time,

then the piano would concurrently produce more than one sound.

Music representations

Within the context of western music, the sheet music, also known as musical **score**, is a visual representation that describes a piece of music by using symbols and letters (Figure 2.3). The term *note* refers to both the musical symbols used in a score, and the corresponding sounds produced once the sheet is interpreted by a musician using an instrument [51].

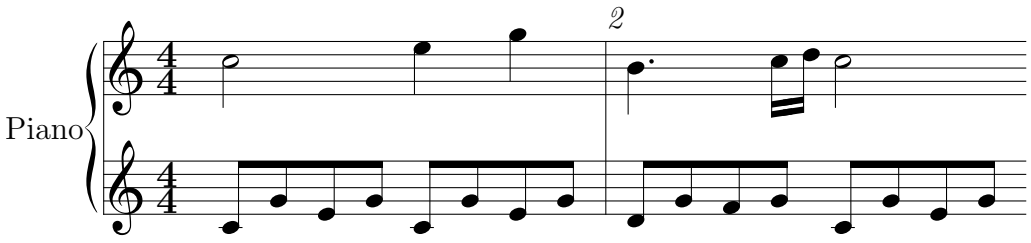


Figure 2.3: Example music score.

The score and the waveform are two different ways to describe a music signal. The first one relies on symbols, whereas the second one measures a continuous property of air (pressure). The **piano-roll** appears as an intermediate symbolic representation lying between the score and the waveform. The piano-roll $\mathbf{P}$ is a matrix where the y-axis denotes the pitches, and the x-axis refers to time (Figure 2.4). This matrix contains only zeros and ones. Therefore, if the element of the piano-roll at the i-th row and j-th column is equal to one, i.e. $\mathbf{P}[i, j] = 1$, it means that the i-th pitch is active during all the j-th window time. In short, the piano-roll registers the pitch and duration of any note played in a musical interpretation [51].

2.2 Automatic music transcription

Automatic music transcription (AMT) aims to transform acoustic music signals into some symbolic music representation. Moreover, the form of the intended music notation defines the complexity of AMT methods. For example, a *frame-level* music representation requires an AMT system that first

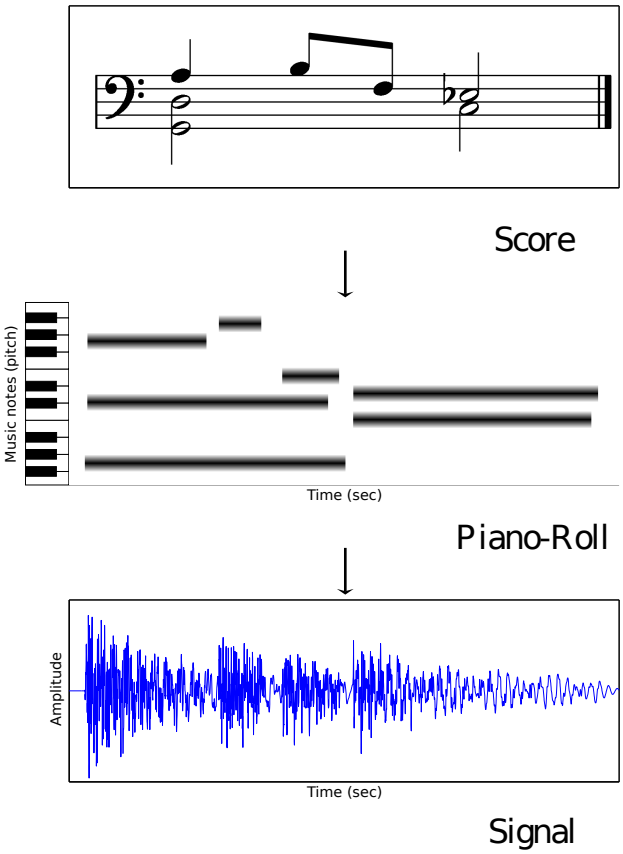


Figure 2.4: Piano-roll as intermediate representation between music score and audio signal [22].

frames the input acoustic signal into windows of usually 10ms, to subsequently produce the list of pitches co-occurring inside the range of each frame. A common approach to frame-level transcription is known as **multi-pitch estimation** [13], which can be used to predict a piano-roll type transcription. Likewise, a *note-level* music notation involves an AMT approach that outputs a full list of notes, that is, a table with the pitch, onset, and offset of every note detected in the input acoustic signal. Last, a *notation-level* transcription refers to the case when the target notation is the music sheet or score [13, 15]. This thesis focuses on frame-level (multi-pitch detection) and note-level transcription.

Regardless of the desired music representation (frame, note, or notation level), transcribing polyphonic music is a challenging task [14]. One reason

is the high number of sound sources combined to form a single polyphonic music signal. Here, sound sources include musical instruments, and vocals (singing). Besides, each source could produce several simultaneous *voices*, that is, more than one parallel melody. In this sense, AMT is undeniably an underdetermined problem [13].

To understand another reason why AMT is challenging, first recall that the sound of an individual note (a.k.a sound event) is not only a fixed-duration sine wave, with a single frequency. A note sound has energy across different frequency bands, and this distribution is often non-stationary yet smooth. In short, a sound event consists of a full spectrum of harmonics, i.e. a fundamental frequency and partials whose energy evolves in time [16]. Furthermore, these partials are comparable to building blocks which, once rearranged and grouped in different shapes, form each of the musical note sounds present in a music recording. In short, music sound events are virtually made of the same fundamental or essential components. The challenge becomes harder when there is a time overlapping between sound events constituted by the same or partially the same elements. This overlapping also extends to the frequency domain. As a result, explaining the energy of a music signal at a specific time and frequency band becomes ambiguous; different combinations of sound events/pitches give equivalent feasible explanations.

2.2.1 Multi-pitch estimation and note-level AMT

Researchers have proposed a wide range of methods for automatic music transcription. This section presents both frequency-domain and time-domain approaches for multi-pitch estimation and note-level AMT.

Time-frequency approaches

Most time-frequency domain methods for multi-pitch detection rely on either non-negative matrix factorization (NMF) or neural networks [13, 15, 17]. In these approaches, the aim is to decompose the spectrogram (time-frequency representation) of the input waveform into elementary components and subsequently use these components to calculate the individual pitch-spectrograms.

Next, we describe some os these methods.

In NMF based multi-pitch estimation, the spectrogram $\mathbf{V}$, which is a non-negative matrix, is factorised as the product of two non-negative matrices $\mathbf{D}$, and $\mathbf{A}$. The columns of the matrix $\mathbf{D}$ represent a dictionary or set of K components, comprising the expected spectrum pattern of each target pitch. The rows of the matrix $\mathbf{A}$ correspond to a set of corresponding K activations that explain when the spectrum of a pitch is active or absent in the spectrogram.

Inference in NMF corresponds to minimising the divergence between $\mathbf{V}$ and $\mathbf{DA}$, with the dictionary and activation matrices as the parameters to be learned. The Kullback-Leibler (KL) divergence is a well-known cost function used for this purpose [39]. In the most unconstrained version of NMF, the activations and dictionary matrices require only to have positive values. Consequently, after minimising the KL divergence between the spectrogram $\mathbf{V}$ and its approximation $\mathbf{DA}$, the learned components, i.e. the columns of $\mathbf{D}$, could lack a spectrum pattern meaning associated with each pitch. In short, the components could be extremely noisy. Likewise, the set of activations, i.e. the rows of $\mathbf{A}$, could exhibit an absence of fundamental properties, such as continuity, smoothness, and temporal and harmonic sense (from a Western music theory point of view). In other words, the activations could also be quite noisy [15].

To reduce the NMF limitations mentioned above, several extensions have been proposed to introduce specific structure and properties in either the activations (rows of $\mathbf{A}$) or the components dictionary (columns of $\mathbf{D}$). For example, sparsity in the activation matrix could be imposed to encourage that each column of the spectrogram $\mathbf{V}$ is explained by a few number pitches/sources, which is often the case in music signals [2]. On the other hand, the harmonic structure of each target pitch can be learnt in a preprocessing step, so a pre-established dictionary or set of components can be used in NMF inference. This is possible when recordings of isolated notes or sound events are available during training. Similarly, every column of the dictionary matrix can be modelled as a linear combination of narrowband spectra corresponding to a finite number adjacent harmonic partials. This encourages harmonicity and

spectral smoothness and allows the spectral envelope to each instrument to be adaptive [78].

Neural networks methods for multi-pitch detection or frame level automatic music transcription have also a mixture spectrogram as input. This could include more than spectrogram each with different time-frequency resolution [21]. Likewise, music structure can be promoted by using a music language model/prior [67]. Current state-of-the-art transcription systems intended for piano rely on deep learning [32]. However, deep learning methods require large quantities of training data to achieve good performance [13]. For example, the overall size of the dataset used in [32] was about 65 hours of audio recordings (see [28] for a description of this dataset). Nevertheless, the authors in [32] claimed that "*to further improve the results we need to create a new dataset that is much larger*". Unfortunately, among the challenges in the music transcription field are the limited available annotated-data [13], and the difficulty of annotating new datasets efficiently [73].

Despite all the relevant contributions that time-frequency multi-pitch detection methods have done to the research community, there are some inherent shortcomings that are challenging to circumvent. Specifically, to operate in the spectrogram means that a frame-level resolution is enforced in the transcription. In short, time-frequency AMT methods are not capable of achieving time-level resolution. Besides, working on the spectrogram often means discarding the phase, incurring a loss of information present in the raw music signal. Next, we describe time-domain methods that avoid these disadvantages.

Time-domain methods

To avoid the time-frequency resolution trade-off, the method proposed in [24] operates directly on the time domain. This method is based on convolutional sparse coding and models the waveform of the mixture input signal as a linear combination of deterministic piano note waveforms (dictionary of components) convolved with their temporal activations. In addition, sparsity is encouraged in the activations, and time-domain components are pre-

trained as a context-specific dictionary. Working in the time-domain allows increasing the transcription accuracy in comparison to time-frequency AMT systems.

Nonetheless, the method proposed in [24] introduces quite strong assumptions about the piano notes present in the music recording. In short, every note is assumed fixed and deterministic, that is, the same sound events repeat throughout the audio signal. This means that different intensities, dynamics, and durations are troublesome to model. As a potential solution, the same paper proposes as future work the usage of a larger and more flexible dictionary of time-domain components. From a Bayesian perspective, we interpret this larger dictionary of components as a probabilistic prior over time-domain functions. As we will see shortly in section 2.4, Gaussian processes (GPs) can be interpreted also as prior probability distributions over functions. This suggests that GPs could be used for defining *larger* and more flexible dictionaries of time-domain components functions. This idea is at the heart of the methods proposed in this thesis (chapters 3, 4, 5.1 and 5.2).

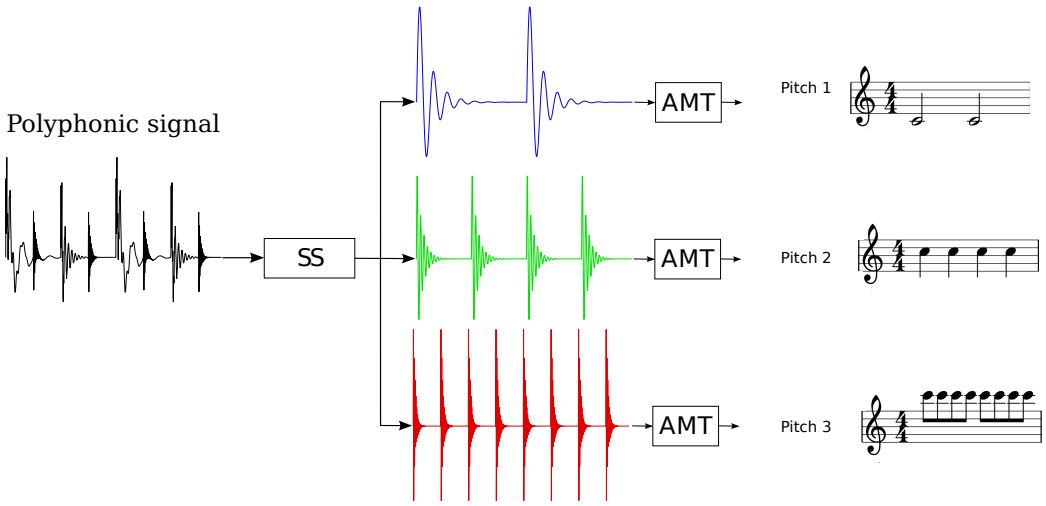


Figure 2.5: Relation between source separation and multi-pitch detection.

In some cases, the boundary between source separation and multi-pitch detection can be diffuse. For example, Yoshii et al. in [87] analysed single-instrument polyphonic music signals to reconstruct the source waveform related to each pitch (Figure 2.5). In short, there was a one-to-one correspon-

dence between sources and pitches. In the next section, we describe this specific case of source separation.

2.3 Source separation

The aim in single-channel audio source separation is to estimate a certain number of latent signals or *sources* that are mixed together in one *mixture* signal [41]. State of the art time-frequency methods include deep learning [70], non-negative matrix factorisation (NMF) [39], and probabilistic latent component analysis (PLCA) [68]. Similarly to time-frequency multi-pith detection approaches, these methods decompose the mixture power spectrogram into fundamental components. Then, the components are used to calculate the individual source-spectrograms. Time-frequency methods often arbitrarily discard phase information. As a result, the phase of each source-spectrogram must be approximated, corrupting the reconstructed sources.

In contrast, time-domain source separation approaches can avoid the phase approximation issue of time-frequency methods [29, 71]. For example, Yoshii et al. [87] reconstructed source signals from the mixture waveform directly in the time domain. To this end, Gaussian processes (GPs) were used to predict each source waveform. GPs are probability distributions over functions [56]. A Gaussian process is completely defined by a mean function, and a kernel or covariance function. In fact, the kernel determines the properties of the functions sampled from a zero-mean GP.

A particularly influential work in time domain approaches is Liutkus et al. [41], who first formulated source separation as a GP regression task. Alternatively, Adam et al. [3] recently proposed to use variational sparse GPs for source separation, however audio signals were beyond the scope of their study. One clear advantage of the GP formulation is that prior knowledge about the properties of the sources, components and activations can be elegantly integrated into the model. This is possible by choosing or designing suitable kernels or covariance functions that encode the desired properties of the latent functions to be inferred. The Gaussian process paradigm is explained in more detail in the next section.

2.4 Gaussian processes

In sections 2.2 and 2.3 we explained that in AMT and source separation the main goal is to infer latent functions from data. Specifically, the idea in source separation is to reconstruct the *source* functions mixed together in an audio recording, whereas the aim in automatic music transcription is to infer an *activation* function for each pitch present in a polyphonic music signal. Now we will introduce the mathematical paradigm of Gaussian processes (GPs). As we will demonstrate shortly, GPs are suitable for inferring functions in scenarios when prior knowledge about a reduced dataset is available. Clearly our case of study is one such scenario, as we have access to a limited number of audio signals. Also, there is knowledge available about the properties of acoustic signals, such as non-stationarity, spectral content, and smoothness. GPs are probability distributions over functions. Further, with GPs we have the ability to combine audio recordings (data) together with knowledge about acoustic signals, in order to make accurate predictions in source separation and automatic music transcription.

This section is organized as follows: Gaussian processes are precisely defined in subsection 2.4.1. The kernel or covariance of a GP is introduced in subsection 2.4.2. Then, examples of stationary kernels are presented in section 2.4.3. Finally, subsection 2.4.4 explains how to combine GPs together with data in order to build regression models.

2.4.1 Preliminaries

Multivariate Gaussian distributions describe finite dimensional normal random variables $\mathbf{f} \in \mathbb{R}^n$. Likewise, *Gaussian processes* describe infinite dimensional normal random variables. That is, when $n \to \infty$ [56]. This infinite generalization of the finite-dimensional multivariate normal distribution follows the Kolmogorov existence theorem [38], which defines the consistency conditions to guarantee that a family of consistent finite-dimensional probability distributions defines a stochastic process. In this sense, Gaussian processes can be defined as distributions over functions. The reason is that a function $f(\mathbf{x})$ can be evaluated at infinite different points $\mathbf{x}$, where $\mathbf{x} \in \mathbb{R}^D$.

If we interpret as random variables the values of $f(\cdot)$ evaluated at all points $\mathbf{x}$, then we end up with a collection of an infinite number of random variables. Moreover, in a Gaussian process any finite subset of random variables $\mathbf{f} = [f(\mathbf{x}_1), \cdots, f(\mathbf{x}_n)]^\top$ follows a joint normal distribution

$$\mathbf{f} \sim \mathcal{N}(\mathbf{f} \,|\boldsymbol{\mu}, \mathbf{K}), \tag{2.1}$$

where $\boldsymbol{\mu}$ is the mean, $\mathbf{K}$ is the covariance matrix, and the multivariate normal distribution is defined as follows

$$\mathcal{N}(\mathbf{f} \,|\boldsymbol{\mu}, \mathbf{K}) = \frac{1}{(2\pi)^{\frac{n}{2}}|\mathbf{K}|^{\frac{1}{2}}} \exp\left\{-\frac{1}{2}(\mathbf{f}-\boldsymbol{\mu})^\top \mathbf{K}^{-1}(\mathbf{f}-\boldsymbol{\mu})\right\}. \tag{2.2}$$

Details about how to compute $\mathbf{K}$ are going to be presented shortly in the next section. For now let us suppose the covariance matrix $\mathbf{K}$ is given. We illustrate the concept of a Gaussian process in Figure 2.6. It shows three vector samples $\{\mathbf{f}_i\}_{i=1}^3$ with $\mathbf{f}_i \in \mathbb{R}^{10}$, drawn from (2.1) assuming a zero mean vector $\boldsymbol{\mu} = \mathbf{0}$. Each plot corresponds to a sample, the black dots represent the values of the vector sampled $\mathbf{f}_i$, the grey line corresponds to the continuous posterior mean function (2.30) obtained using GP regression (see section 2.4.4).

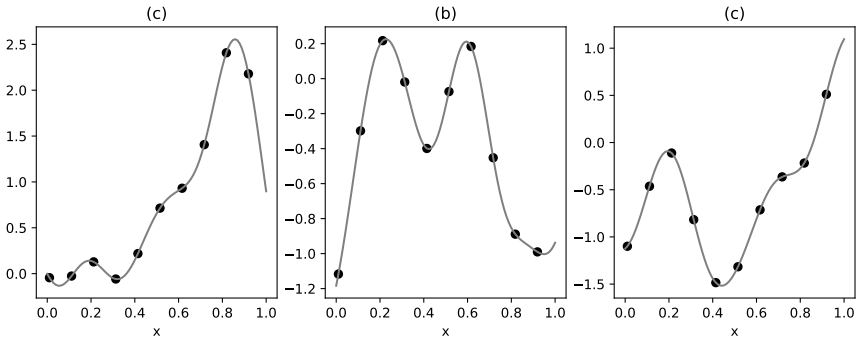


Figure 2.6: Three samples from a multivariate Gaussian distribution (dots). Underlying functions (continuous lines).

Just like the multivariate Gaussian distribution (2.2) is completely parametrized by its mean vector and covariance matrix, a Gaussian process is fully specified by a mean function $\mu(\mathbf{x})$, and a covariance function or *kernel*

$k(\mathbf{x}, \mathbf{x}')$. That is

$$\mu(\mathbf{x}) = \mathbb{E}[f(\mathbf{x})], \tag{2.3}$$

and

$$k(\mathbf{x}, \mathbf{x}') = \mathbb{E}\left[(f(\mathbf{x}) - \mu(\mathbf{x}))(f(\mathbf{x}') - \mu(\mathbf{x}'))\right], \tag{2.4}$$

where $k(\mathbf{x}, \mathbf{x}')$ has free hyperparameters $\boldsymbol{\theta}$. In expressions such as (2.3) the expectation is taken over the stochastic function $f(\mathbf{x})$ equipped with a probability measure $p(f(\mathbf{x}))$, that is $\mathbb{E}[f(\mathbf{x})] = \int f(\mathbf{x})p(f(\mathbf{x}))\mathrm{d}f(\mathbf{x})$. We write the Gaussian process as

$$f(\mathbf{x}) \sim \mathcal{GP}(\mu(\mathbf{x}),\ k(\mathbf{x}, \mathbf{x}')). \tag{2.5}$$

When the mean function is assumed $\mu(\mathbf{x}) = 0$, then the kernel $k(\mathbf{x}, \mathbf{x}')$ determines the properties of $f(\mathbf{x})$. Also, the covariance function specifies how a GP model generalizes or extrapolates [42].

2.4.2 The covariance function

We have introduced the GP as a collection of an infinite number of random variables, such as any finite set of these random variables follows a multivariate normal distribution. This section presents the covariance function or kernel of a GP, that is, the function $k(\mathbf{x}, \mathbf{x}')$ that specifies the dependency between any pair of random variables, corresponding to evaluate the function $f(\cdot)$ at any two points $\mathbf{x}$, $\mathbf{x}'$. Further, the kernel defines the notion of *nearness* or *similarity* between any two function values $f(\mathbf{x})$ and $f(\mathbf{x}')$ [56]. From now on, we will focus on univariate input variables, i.e. $\mathbf{x} \in \mathbb{R}$. This is because the only independent variable we are considering in this research is time. Therefore, we make the following change of variable $\mathbf{x} = t$, in order to keep the notation uncluttered. In addition, we use the word *kernel* interchangeably with *covariance function*.

The kernel determines the properties of the functions sampled from a GP.

For example, by choosing certain covariance function we can draw samples that are stationary and smooth (Fig 2.7(a)) or rough (Fig 2.7(b)). In addition, a periodic kernel introduces regularities in the properties of the sampled functions (Fig 2.7(c)). Also, by using a non-stationary covariance we can encourage the behaviour of the functions to depend on time (Fig 2.7(d)). In summary, the kernel encodes prior knowledge (assumptions) about the data we aim to model with a GP. How to combine GPs with data to make predictions is introduced in section 2.4.4.

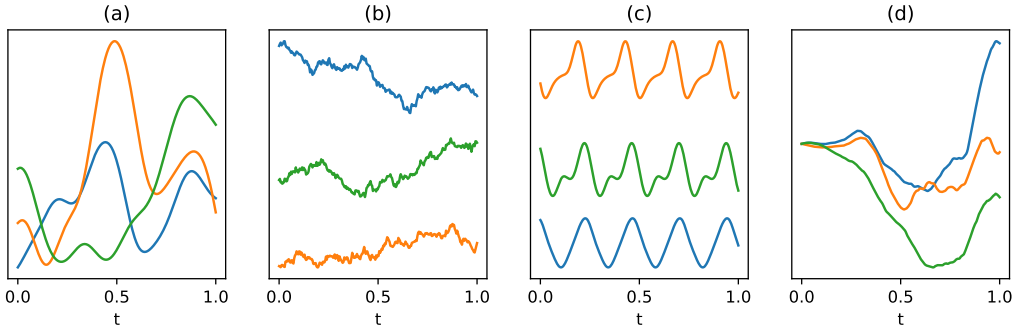


Figure 2.7: Functions sampled from four different GPs.

Bear in mind that all functions with two inputs are not necessarily valid kernels [56]. First, let us introduce $\mathbf{t} = \{t_i\}_{i=1}^{n}$ as a set of n time instants where the function $f(\cdot)$ is evaluated. A valid kernel should satisfy the following necessary and sufficient condition: the matrix $\mathbf{K}$, computed by evaluating the kernel $k(t, t')$ at all possible combinations of the elements in $\mathbf{t}$ (i.e. $\mathbf{K}_{i,j} = k(t_i, t_j)$), is a positive semidefinite matrix for all possible choices of the set $\mathbf{t}$ [19]. To fulfil this condition, a kernel has to satisfy the following three properties:

$$k(t, t) = \text{cov}\left(f(t), f(t)\right) = \text{var}\left(f(t)\right) \geq 0, \tag{2.6}$$

that is, $k(t, t)$ is **positive**. In addition,

$$k(t, t') = \text{cov}\left(f(t), f(t')\right) = \text{cov}\left(f(t'), f(t)\right) = k(t', t), \tag{2.7}$$

that is, $k(t, t')$ is **symmetric**. Also,

$$\sum_{i=1}^{n}\sum_{j=1}^{n} a_i a_j k(t_i, t_j) \geq 0, \tag{2.8}$$

where n, a_i and t_i are arbitrary [18]. Examples of valid kernels are presented shortly.

2.4.3 Stationary covariance functions

A Gaussian process (2.5) is wide sense stationary (WSS) if its mean function is constant, and its kernel is stationary, i.e. a function of $\tau = t - t'$ [65, 56]. This means that the covariance is invariant to translations in time. If the kernel is isotropic, then it is a function of r, where $r = |\tau|$. In addition, it can be shown that the *spectral density* or *power spectrum* $S(s)$ of a WSS process corresponds to the Fourier transform (FT) of its covariance function, that is

$$S(s) = \int_{-\infty}^{\infty} k(\tau) e^{-js\tau} \mathrm{d}\tau, \tag{2.9}$$

thus

$$k(\tau) = \frac{1}{2\pi} \int_{-\infty}^{\infty} S(s) e^{js\tau} \mathrm{d}s. \tag{2.10}$$

This is known as the Wiener-Khintchine theorem [56, 65]. This connection implies that we can analyse kernels in the frequency-domain, and choose the covariance functions whose properties are more appropriate for modelling the spectral content of music signals.

Next, we describe examples of stationary covariance functions, specifically, the *exponentiated quadratic*, three kernels from the *Matérn* family, the *standard periodic*, and the *spectral mixture* kernel.

Exponentiated quadratic

This kernel has the expression

$$k_{\text{EQ}}(r) = \sigma^2 \exp\left(-\frac{r^2}{2\ell^2}\right), \tag{2.11}$$

where σ^2 corresponds to the variance, and ℓ to the lengthscale parameter. The form of (2.11) is shown in Figure 2.9(a), when $\sigma^2 = 1$ and $\ell = 0.5$. In this kernel, the larger the gap r between the time instants, that is $r = |t - t'|$, the less dependent the random variables $f(t)$ and $f(t')$ are. The functions sampled from a GP with this covariance are infinitely smooth (Figure 2.8(a)).

Matérn kernels

Here we present the first three kernels of the Matérn family with half-integer orders [56]. These covariances have the form

$$k_{1/2}(r) = \sigma^2 \exp\left(-\frac{r}{\ell}\right), \tag{2.12}$$

$$k_{3/2}(r) = \sigma^2 \left(1 + \frac{\sqrt{3}r}{\ell}\right) \exp\left(-\frac{\sqrt{3}r}{\ell}\right), \tag{2.13}$$

$$k_{5/2}(r) = \sigma^2 \left(1 + \frac{\sqrt{5}r}{\ell} + \frac{5r^2}{3\ell^2}\right) \exp\left(-\frac{\sqrt{5}r}{\ell}\right), \tag{2.14}$$

where σ^2 represents the variance, and ℓ the lengthscale. The order of the kernel, i.e. $\frac{1}{2}$, $\frac{3}{2}$ or $\frac{5}{2}$, determines the number of times the realizations from a GP (with a Matérn covariance) can be differentiated. In other words, the order defines how smooth the drawn functions are. The lower the order, the less smooth they are (Figure 2.8(b-d)). Similar to the exponentiated quadratic kernel (2.11), in the Matérn family the dependency between two observations decreases with the size of the time gap between them (Figure 2.9(a)).

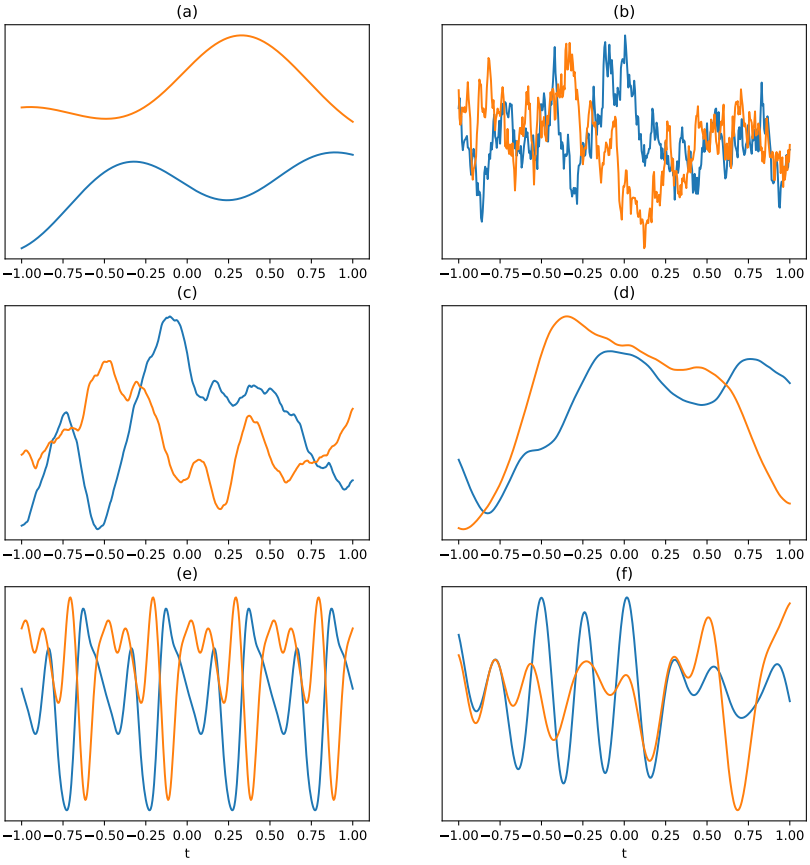


Figure 2.8: Functions drawn from GPs with different kernels. Exponentiated quadratic (a). Matérn 1/2 (b). Matérn 3/2 (c). Matérn 5/2 (d). Standard periodic (e). Spectral mixture (f).

Standard periodic

To create a standard kernel with periodic structure [43], first the input variable time is wrapped onto a circle, i.e. $\boldsymbol{\phi}(t) = [\cos(t),\ \sin(t)]^\top$. Subsequently, the two dimensional feature vector $\boldsymbol{\phi}(t)$ is used as input in the exponentiated quadratic kernel (2.11). Recall $r = |t - t'|$, but with the transformed input variable we get

$$\hat{r} = |\boldsymbol{\phi}(t) - \boldsymbol{\phi}(t')|,$$
$$\hat{r} = \sqrt{[\cos(t) - \cos(t')]^2 + [\sin(t) - \sin(t')]^2},$$

now, replacing $\hat{r}$ in (2.11), we get

$$k_{\text{SP}}(\hat{r}) = \sigma^2 \exp\left(-\frac{\left\{\sqrt{[\cos(t)-\cos(t')]^2+[\sin(t)-\sin(t')]^2}\right\}^2}{2\ell^2}\right),$$
$$= \sigma^2 \exp\left(-\frac{1-\cos(t-t')}{\ell^2}\right).$$

Using the trigonometric property $\sin^2\left(\frac{\theta}{2}\right) = \frac{1}{2}[1-\cos(\theta)]$ we get the *standard periodic* covariance

$$k_{\text{SP}}(r) = \sigma^2 \exp\left(-\frac{2\sin^2\left(\frac{r}{2}\right)}{\ell^2}\right), \tag{2.15}$$

with $r = |t-t'|$ [56]. With this kernel the covariance evolves periodically with respect to r (Figure 2.9(b)). The functions sampled from a GP with this covariance are periodic. In addition, their spectral content is determined by a mixture of a finite number of perfect harmonics, that is, a fundamental frequency F_0, plus partials whose frequency is an integer multiple of F_0. Figure 2.8(e) shows two samples from a GP with this covariance function.

Spectral mixture

The spectral mixture (SM) kernel is derived when a spectral density (2.9) is approximated using a mixture of Gaussians [83]. If the input variable is an scalar, i.e. $\mathbf{x} = t$ with $t \in \mathbb{R}$, then the spectral mixture kernel corresponds to

$$k_{\text{SM}}(r) = \sum_{p=1}^{P} \sigma_p^2 \exp\left(-\frac{r^2}{2\ell_p^2}\right) \cos\left(\omega_p r\right). \tag{2.16}$$

Here, the set of hyperparameters $\{\omega_p\}_{p=1}^P$ defines the modes of the Gaussian functions, that is, their locations in the frequency-domain, the set $\left\{\sigma_p^2\right\}_{p=1}^P$ determines the contribution of the p-th component to the whole kernel, and the set $\{\ell_p\}_{p=1}^P$ specifies the lengthscale for each component, that is, how wide or narrow the p-th Gaussian function is in the frequency domain.

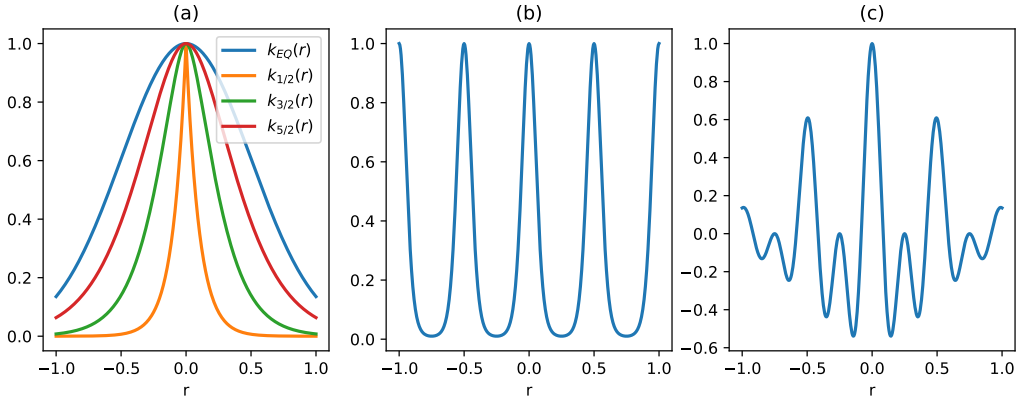


Figure 2.9: Form kernels. Exponentiated quadratic, Matérn 1/2, 3/2, and 5/2 (a). Standard periodic (b). Spectral mixture (c).

The spectral mixture kernel is quite flexible. It can approximate a wide range of stationary kernels, including periodic, quasi-periodic, and not periodic ones. For this reason, functions drawn from a GP with this kernel can have a wide variety of behaviours and properties. These kernel attributes might be useful for music signals (see Chapters 4, 5.1 and 5.2). Here, we show a specific example of the SM kernel (Figure 2.9(c)). In this case, the functions drawn from a GP with this covariance are quasi-periodic (Figure 2.8(f)). This thesis pays considerable attention to spectral mixture kernels. Moreover, we focus on developing a similar family of covariances, called the Matérn spectral mixture kernels (chapter 4).

2.4.4 Gaussian process regression

So far we have introduced Gaussian processes as probability distributions over functions. Further, we emphasized that the kernel governs the properties of the functions drawn from a GP. This section presents how to combine GPs with data, in order to make predictions. GP-based machine learning is considered a powerful Bayesian paradigm for nonparametric nonlinear regression and classification models [62]. Here we focus on regression, i.e. predicting a continuous quantity [56]. In GP regression rather than inferring the parameters $\boldsymbol{\theta}$ of a fixed-form function of time $f(t, \boldsymbol{\theta})$, we introduce a prior over the

function $f(t)$ itself. Subsequently, we use the information about the function provided by the data to calculate the posterior distribution over $f(t)$ [56, 59, 61]. In this sense, we use GPs as priors, that is, as the element that embodies the assumptions and knowledge available about the observed data.

We notate the data, i.e. audio signals, as $\mathcal{D} = \{t_i,\ y_i\}_{i=1}^{n}$, where $t_i \in \mathbb{R}^+$ (including zero), $y_i \in \mathbb{R}$, and n is the number of observations. We group time instants, and data values in the vectors $\mathbf{t} = [t_1, \cdots, t_n]^\top$, and $\mathbf{y} = [y_1, \cdots, y_n]^\top$ respectively. In addition, audio samples $\{y_i\}_{i=1}^{n}$ are assumed to be noisy measurements of a zero-mean GP $f(t)$, that is,

$$f(t) \sim \mathcal{GP}\left(0, k(t, t')\right), \tag{2.17}$$

where $k(t, t')$ is a covariance function. Also, the observation time instants $\{t_i\}_{i=1}^{n}$ are assumed regularly-spaced (though GP regression allows for irregular sampling or missing data). In short, the regression model corresponds to

$$y_i = f(t_i) + \epsilon_i, \tag{2.18}$$

where the value of each noise variable in $\{\epsilon_i\}_{i=1}^{n}$ is sampled independently for each observation $\{y_i\}_{i=1}^{n}$ [19]. We assume that every noise variable ϵ_i follows the same zero-mean Gaussian distribution with variance ν^2, that is, $\epsilon_i \sim \mathcal{N}(0, \nu^2)\ \forall\ i$. Further, the probability of y_i conditioned to f_i is

$$p(y_i|f_i) = \mathcal{N}(y_i|\ f_i, \nu^2),$$

where $f_i = f(t_i)$. Because the noise is independent for each observation y_i, then the distribution over the complete audio recording $\mathbf{y} = [y_1, \cdots, y_n]^\top$, conditioned to the function values $\mathbf{f} = [f(t_1), \cdots, f(t_n)]^\top$, corresponds to an isotropic Gaussian distribution with form

$$\begin{aligned} p(\mathbf{y}|\mathbf{f}) &= \prod_{i=1}^{n} \mathcal{N}(y_i|\ f_i, \nu^2), \\ &= \mathcal{N}(\mathbf{y}|\ \mathbf{f}, \nu^2\mathbf{I}), \end{aligned} \tag{2.19}$$

where $\mathbf{I}$ is the identity matrix with size $n \times n$. The expression (2.19) is known as the **likelihood**. In addition, recall we assumed the function $f(t)$ follows a zero-mean GP ((2.17)), therefore the probability of $\mathbf{f}$ is

$$p(\mathbf{f}) = \mathcal{N}(\mathbf{f}|\ \mathbf{0}, \mathbf{K}), \tag{2.20}$$

where the elements of the mean vector are $\{\mu_i\}_{i=1}^n = 0$. The covariance matrix has entries $\mathbf{K}_{ij} = k(t_i, t_j)$, where $k(\cdot, \cdot)$ is a valid kernel (see section 2.4.2).

From a Bayesian perspective, we are interested in calculating the posterior over $f(t)$ evaluated at test points $\mathbf{t}_*$. For now lets suppose $\mathbf{t}_* = \mathbf{t}$. Using Bayes theorem we know that the conditional distribution of $\mathbf{f}$ given the data $\mathbf{y}$ follows

$$p(\mathbf{f}|\mathbf{y}) = \frac{p(\mathbf{y}|\mathbf{f})p(\mathbf{f})}{p(\mathbf{y})}, \tag{2.21}$$

where $p(\mathbf{f}|\mathbf{y})$ is the posterior distribution, $p(\mathbf{y}|\mathbf{f})$ corresponds to the likelihood, $p(\mathbf{f})$ to the prior, and $p(\mathbf{y})$ is the evidence or *marginal likelihood*. The marginal-likelihood $p(\mathbf{y})$ is the integral of the likelihood times the prior, and it reflects how probable is the observed vector $\mathbf{y}$, conditioned on the kernel hyperparameters $\boldsymbol{\theta}$. The evidence corresponds to

$$p(\mathbf{y}) = \int p(\mathbf{y}|\mathbf{f})p(\mathbf{f})\mathrm{d}\mathbf{f}. \tag{2.22}$$

Since the likelihood $p(\mathbf{y}|\mathbf{f})$ is conjugate to the prior $p(\mathbf{f})$, that is, both are multivariate Gaussian distributions, then the form of the marginal-likelihood $p(\mathbf{y})$ in (2.22) is also Gaussian [56]. We can calculate directly the marginal likelihood $p(\mathbf{y})$, from (2.18) we know that

$$\mathbf{y} = \mathbf{f} + \boldsymbol{\epsilon}, \tag{2.23}$$

where the $\mathbf{f}$ follows (2.20) (with zero-mean), and the noise vector follows $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \nu^2\mathbf{I})$. The variable $\mathbf{y}$ corresponds to the sum of two normal vectors,

therefore its distribution is also Gaussian with form

$$p(\mathbf{y}) = \mathcal{N}\left(\mathbf{y}|\ \mathbb{E}\{\mathbf{y}\}, \mathrm{Cov}\left[\mathbf{y}, \mathbf{y}\right]\right), \tag{2.24}$$

where $\mathbb{E}\{\mathbf{y}\} = \mathbb{E}\{\mathbf{f} + \boldsymbol{\epsilon}\} = \mathbf{0}$, and

$$\begin{aligned}
\mathrm{Cov}\left[\mathbf{y}, \mathbf{y}\right] &= \mathbb{E}\left\{\mathbf{y}\mathbf{y}^\top\right\} \\
&= \mathbb{E}\left\{\mathbf{f}\mathbf{f}^\top + \mathbf{f}\boldsymbol{\epsilon}^\top + \boldsymbol{\epsilon}\mathbf{f}^\top + \boldsymbol{\epsilon}\boldsymbol{\epsilon}^\top\right\} \\
&= \mathbb{E}\left\{\mathbf{f}\mathbf{f}^\top\right\} + \mathbb{E}\left\{\boldsymbol{\epsilon}\boldsymbol{\epsilon}^\top\right\} \\
&= \mathrm{Cov}\left[\mathbf{f}, \mathbf{f}\right] + \mathrm{Cov}\left[\boldsymbol{\epsilon}, \boldsymbol{\epsilon}\right] \\
&= \mathbf{K} + \nu^2\mathbf{I},
\end{aligned}$$

then

$$p(\mathbf{y}) = \mathcal{N}(\mathbf{y}|\ \mathbf{0}, \mathbf{K}_y), \tag{2.25}$$

where $\mathbf{K}_y = \mathbf{K} + \nu^2\mathbf{I}$. The reason it is called the marginal likelihood, rather than just likelihood, is because we have marginalized out the latent Gaussian vector $\mathbf{f}$ [52]. The log of (2.25) is usually the objective function when learning the hyperparameters (see **training** subsection). Finally, the computation of the posterior (2.21) or predictive distribution is explained shortly.

Training

In GP regression, training refers to selecting the likelihood parameters (e.g. noise variance), the covariance function, and its hyperparameters [56]. The objective function to optimize is usually the log of the marginal likelihood (2.25)

$$\begin{aligned}
J(\boldsymbol{\theta}) &= \log p(\mathbf{y}|\boldsymbol{\theta}), \\
&= -\frac{1}{2}\mathbf{y}^\top\left[\mathbf{K} + \nu^2\mathbf{I}\right]^{-1}\mathbf{y} - \frac{1}{2}\log\left|\mathbf{K} + \nu^2\mathbf{I}\right| - \frac{n}{2}\log(2\pi),
\end{aligned} \tag{2.26}$$

where $\boldsymbol{\theta}$ are the kernel (used to calculate $\mathbf{K}$) and likelihood hyperparameters. Moreover, optimization algorithms require the gradients of $J(\boldsymbol{\theta})$, that is,

$$\begin{aligned}\frac{\partial}{\partial\theta_i}J(\boldsymbol{\theta}) &= \frac{1}{2}\mathbf{y}^\top\mathbf{K}_y^{-1}\frac{\partial\mathbf{K}_y}{\partial\theta_i}\mathbf{K}_y^{-1}\mathbf{y} - \frac{1}{2}\mathrm{tr}\left(\mathbf{K}_y^{-1}\frac{\partial\mathbf{K}_y}{\partial\theta_i}\right), \\ &= \frac{1}{2}\mathrm{tr}\left(\left(\boldsymbol{\alpha}\boldsymbol{\alpha}^\top - \mathbf{K}_y^{-1}\right)\frac{\partial\mathbf{K}_y}{\partial\theta_i}\right),\end{aligned} \tag{2.27}$$

where $\boldsymbol{\alpha} = \mathbf{K}_y^{-1}\mathbf{y}$. The form of these derivatives depends completely on the selected kernel $k(t, t')$.

The computation of (2.26) and (2.27) require to invert a $n \times n$ matrix. The time needed for matrix inversion is usually $\mathcal{O}(n^3)$ (see section 2.6.1). Thus, the larger the training dataset (i.e. n), the more time the optimization demands. In fact, the standard GP regression model is intractable for large datasets, making it incompatible with the audio signals we aim to analyse. This is because when recording audio, between 16000 to 44100 data values are usually collected per second. In order to make GP models suitable for processing audio, we study approximate inference methods that alleviate the burden of matrix inversion (for a detailed explanation see section 2.6).

Predictive distribution

Recall that the kernel introduces dependencies between the values of the function $f(t)$ at different time instants. Therefore, the noisy observations $\mathbf{y} \in \mathbb{R}^n$ of the function $f(t)$ evaluated at $\mathbf{t} = \{t_i\}_{i=1}^n$ provide also information of the unobserved function values $\mathbf{f}_* \in \mathbb{R}^m$. Here, m is the number of time instants where we aim to make predictions, that is $\mathbf{t}_* = \left\{\hat{t}_j\right\}_{j=1}^m$. This dependency introduced by the kernel is what allows us to make predictions. To do so, we first define the joint distribution

$$p(\mathbf{y}, \mathbf{f}_*) = \mathcal{N}\left(\begin{bmatrix}\mathbf{y} \\ \mathbf{f}_*\end{bmatrix} \middle| \mathbf{0}, \begin{bmatrix}\mathbf{K}(\mathbf{t},\mathbf{t}) + \nu^2\mathbf{I} & \mathbf{K}(\mathbf{t},\mathbf{t}_*) \\ \mathbf{K}(\mathbf{t}_*,\mathbf{t}) & \mathbf{K}(\mathbf{t}_*,\mathbf{t}_*)\end{bmatrix}\right), \tag{2.28}$$

where $\mathbf{K}(\mathbf{t},\mathbf{t})$ is a $n \times n$ matrix, $\mathbf{K}(\mathbf{t}_*,\mathbf{t}_*)$ is a $m \times m$ matrix, and $\mathbf{K}(\mathbf{t},\mathbf{t}_*)$ is a $n \times m$ matrix corresponding to evaluate the kernel $k(t, t')$ on all pos-

sible combinations between the elements of $\mathbf{t}$ and $\mathbf{t}_*$, i.e. the training and prediction time instants respectively. In addition, using the joint (2.28) and the conditional property of the Gaussian distribution (see appendix A), we calculate the posterior

$$p(\mathbf{f}_*|\mathbf{y}) = \mathcal{N}\left(\boldsymbol{\mu}_{\text{pos}},\ \mathbf{K}_{\text{pos}}\right), \tag{2.29}$$

where the mean corresponds to

$$\boldsymbol{\mu}_{\text{pos}} = \mathbf{K}(\mathbf{t}_*, \mathbf{t}) \left[\mathbf{K}(\mathbf{t}, \mathbf{t}) + \nu^2 \mathbf{I}\right]^{-1} \mathbf{y}, \tag{2.30}$$

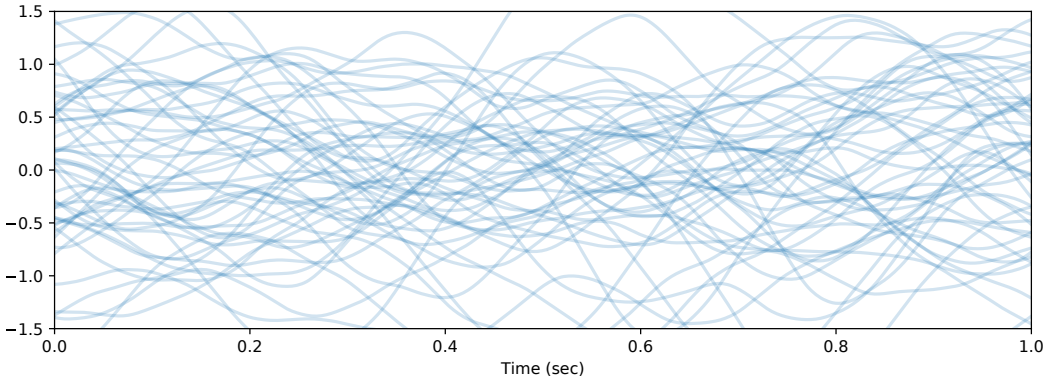
and the covariance matrix to

$$\mathbf{K}_{\text{pos}} = \mathbf{K}(\mathbf{t}_*, \mathbf{t}_*) - \mathbf{K}(\mathbf{t}_*, \mathbf{t}) \left[\mathbf{K}(\mathbf{t}, \mathbf{t}) + \nu^2 \mathbf{I}\right]^{-1} \mathbf{K}(\mathbf{t}, \mathbf{t}_*). \tag{2.31}$$

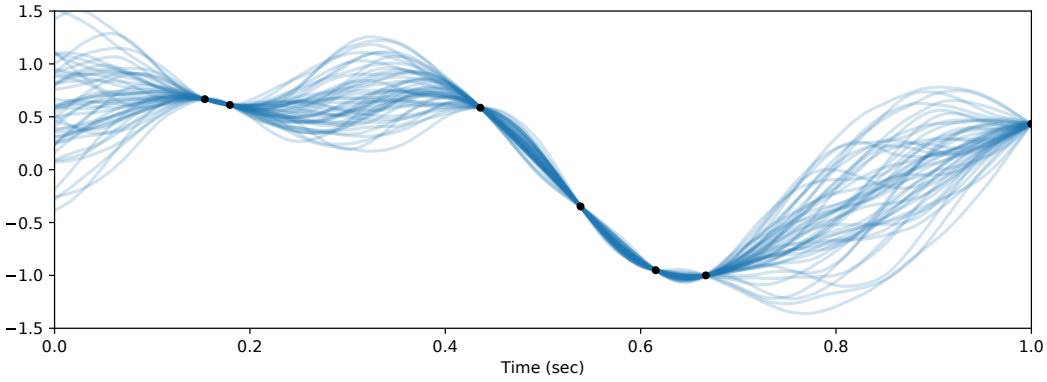
The form of the posterior mean (2.30) and covariance (2.31) depend on the kernel used to calculate $\mathbf{K}(\cdot, \cdot)$. Therefore, a change in the kernel will affect the model prediction.

2.4.5 Toy example regression

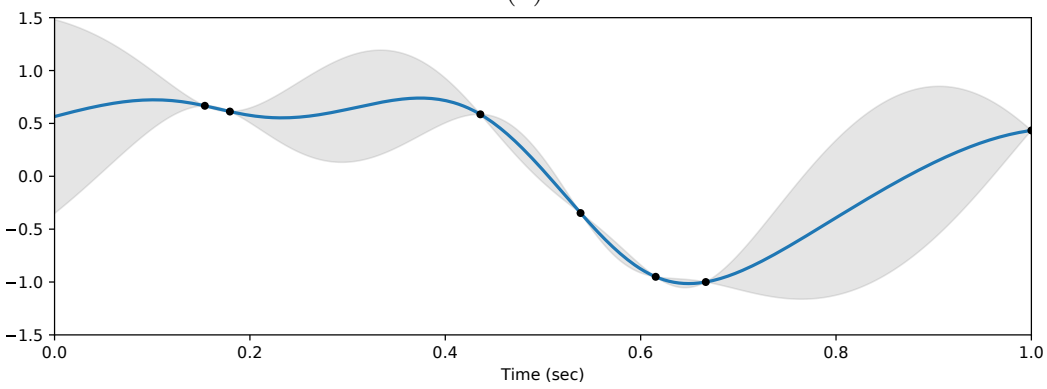
To summarise the GP concepts presented so far, we introduce a toy example of GP regression (Figure 2.10). Recall the main goal is to combine a model/prior with data, to make predictions. Here, we used the Matérn 3/2 kernel (2.13). We first set the lengthscale and variance hyperparameters with $\ell = 1$ and $\sigma^2 = 1$ respectively. The functions sampled from this prior are slightly smooth (Figure 2.10(a)). Then, we generated synthetic data by evaluating the deterministic function $g(t) = \sin(2\pi t) + \cos(2.3 \times 2\pi t) + \sin(1.3 \times 2\pi t)$ at seven random points in the range $(0, 1)$. Subsequently, we used the data to optimize the log marginal likelihood (2.26), that is, to learn the hyperparameters. Last, with the trained lengthscale and variance $\ell = 0.25$, $\sigma^2 = 0.49$, we computed the predictive distribution (2.29) over the function given the data. Notice that the functions sampled from the posterior pass trough the observations (dots in Figure 2.10(b)). It is common practice to present, rather



(a)

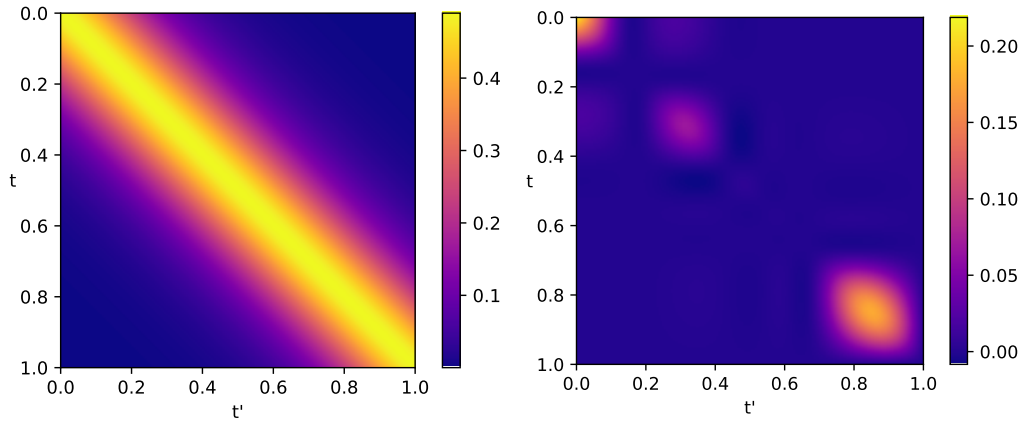


(b)



(c)

Figure 2.10: Example GP regression. Samples from the prior (a). Samples from the posterior (b). Posterior mean and interval of confidence (c).



(a) Prior covariance matrix.

(b) Posterior covariance matrix.

Figure 2.11: Prior and posterior covariance matrices of example shown in Figure 2.10.

than samples, the posterior mean as well as the confidence interval, i.e. the shaded area in Figure 2.10(c). The confidence interval represents the space where lie 95% of the realizations drawn from each posterior marginal distribution $p(f_i^\star|\mathbf{y})$, where $f_i^\star \in \mathbb{R}$ is the i-th variable of the prediction vector $\mathbf{f}^\star$ (see (2.29)). Figure 2.10b shows 50 functions sampled from the posterior. The posterior distribution is not independent throughout all the marginals, Figure 2.11b shows its covariance matrix. We observe that this matrix is not diagonal. This introduces dependency between any two random variables representing the functions values $f(t)$ and $f(t')$ at time instants where the posterior covariance matrix is not equal to zero.

2.4.6 Challenges of Gaussian process models

To use Gaussian processes for machine learning presents advantages and limitations, especially when modelling large datasets. Strengths of GPs include its capacity to naturally introduce prior knowledge about the data into the model, through the kernel. Also, GPs offer a principled manner to quantify uncertainty, in the sense that predictions consist of a posterior mean and intervals of confidence defined by the posterior variance. Moreover, GP modelling is a non-parametric paradigm, that is, the inference corresponds

to computing the posterior over a function given the data, rather than the posterior over the parameters of a deterministic function established in advance. In short, GPs models are quite flexible, allowing *the data to speak for itself* [19, 52, 56]. Still, GP modelling involves facing some challenges. The following two sections discusses two of them, namely kernel design and scalability. Also, we present an outline of how our research contributes to solving these challenges within the context of pitch detection and source separation in music signals.

2.5 Kernel design for acoustic music signals

The kernel of a Gaussian process profoundly influences how a model extrapolates to regions of the input space where there is no training data. In short, the covariance function determines the GP model capability to generalise [62, 56, 83]. For example, if the region of interest is far away from the observations, and the kernel used is not strictly periodic, then the prediction converges to the mean function of the process, which often corresponds to zero. Also, if the kernel encodes no more than general patterns such as stationarity, continuity and regularity, then the GP works like a smoother between the observations. Therefore, GP models with a higher capacity to generalise depend on designing more expressive kernels.

Several researchers have studied kernel design for Gaussian process models. Related work to this thesis includes Durrande et al. [27], who developed kernels for detecting periodicity in the data. Also, Wilson et al. [83] proposed to approximate any stationary covariance function by using a linear combination of RBF times cosine covariance functions. More recent work includes Remes et al. [57], who proposed an extension of Wilson's work to non-stationary kernels. Besides, the models proposed in [77, 81] make use of multi-output Gaussian processes to represent the cross-correlation between frequency bands in natural sounds. This thesis, however, intends to specially design GP priors that encode acoustic properties of music signals, namely: smoothness, periodicity, spectral content, and non-stationary amplitude.

To this end, chapter 3 presents an initial comparison of well-known co-

variance functions able to describe smoothness, periodicity, and harmonic content in a very constrained manner. Next, chapter 4 departs from Wilson et al. [83] work, and introduces the Matérn spectral mixture (MSM) kernel, together with a method to initialise its hyperparameters in a region with meaningful acoustics interpretation. Here, a product-GP model describes the non-stationary amplitude of acoustic signals. Finally, chapter 5.1 presents an alternative method to initialise the MSM kernel by using the autocorrelation of the training data.

2.6 Sparse variational Gaussian processes

Doing inference in standard Gaussian process models is computationally expensive. This is because learning the hyperparameters by maximising the marginal-likelihood, as well as computing the predictive distribution, requires to invert a $n \times n$ matrix, where n is the size of the data $\{t_i, y_i\}_{i=1}^{n}$ [43, 56]. The computational complexity of inverting a matrix scales cubically, i.e. $\mathcal{O}(n^3)$ (see section 2.6.1), which becomes intractable when n is big (usually $n \gg 1 \times 10^4$). In addition, the posterior does not have a closed-form when the data likelihood is not conjugate to the prior, i.e. when the likelihood is not Gaussian [36]. This thesis follows a **sparse approximate variational inference** approach to tackle both of these challenges simultaneously.

2.6.1 Computational complexity of inverting matrices

Inverting dense covariance matrices are necessary operations when using Gaussian processes for machine learning. This section describes the computational complexity of matrix inversion.

Notation of computational complexity

In the context of this thesis, computational complexity refers to the asymptotic efficiency of algorithms. That is, how the running time needed to execute an algorithm increases with the size of the input, when the size of the input rises without bound [25]. In addition, the $\mathcal{O}$-notation (pronounced

"big-oh") refers to the asymptotic upper bound, i.e., the worst-case running time needed to compute an algorithm. When we say that inverting a matrix of size $n \times n$ takes time $\mathcal{O}(n^3)$, it means that the worst-case running time of performing such an operator increases cubically with the size of the matrix.

Matrix inversion

Suppose the square matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$ is not singular, that is, there exists a matrix $\mathbf{A}^{-1} \in \mathbb{R}^{n \times n}$ such as

$$\mathbf{A}\mathbf{A}^{-1} = \mathbf{I}_n, \tag{2.32}$$

where $\mathbf{I}_n$ is the identity matrix of size $n \times n$. Defining $\mathbf{X} = \mathbf{A}^{-1}$, and expanding (2.32) we get

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix} \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{nn} \end{bmatrix} = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix}, \tag{2.33}$$

We can interpret (2.33) as a set of n distinct equations of the form

$$\mathbf{A}\mathbf{x}_i = \mathbf{b}_i, \tag{2.34}$$

where $\mathbf{x}_i$ represents the i-th column in $\mathbf{X}$, and $\mathbf{b}_i$ corresponds to the i-th column of the identity matrix $\mathbf{I}_n$. The system of linear equations (2.34) can be solve in time $\mathcal{O}(n^2)$ when using an LUP decomposition of $\mathbf{A}$. the LUP decomposition follows $\mathbf{PA} = \mathbf{LU}$, where $\mathbf{P}$ is a permutation matrix, $\mathbf{L}$ a unit lower-triangular matrix, and $\mathbf{U}$ an upper-triangular matrix [25]. Observe that the LUP decomposition only depends on $\mathbf{A}$, then the same decomposition (i.e., computed only once) can be applied to (2.34) for different values of $\mathbf{b}_i$, taking additional time $\mathcal{O}(n^2)$. In general, it takes $\mathcal{O}(kn^2)$ to solve k linear systems of n-linear equations with n unknowns (2.34), when all systems share $\mathbf{A}$ and differ only in $\mathbf{b}_i$. For a square matrix this means solving $k = n$ systems, therefore the time required for inverting a matrix in $\mathcal{O}(n^3)$.

In addition, matrix multiplication and matrix inversion are equivalent problems, in the sense that we can use an algorithm for matrix multiplication to solve the inverse of a matrix (and the other way around), taking the same asymptotic running time [25]. Therefore, we can invert a matrix by using the Strassen's algorithm for matrix multiplication. The Strassen's algorithm runs in $\mathcal{O}(n^{2.81})$ time [72], which is faster than the approach explained above, which runs in $\mathcal{O}(n^3)$ time.

2.6.2 Sparse approximate Gaussian processes

The main idea in sparse GP methods is to approximate the high-dimensional covariance matrix of the full Gaussian process prior (2.20). The approximate matrix has a lower-rank in comparison to the real covariance matrix, and its construction relies on a set of m variables called *inducing variables*, where $m < n$. This approximation reduces the time complexity from $\mathcal{O}(n^3)$ (see section 2.6.1) to $\mathcal{O}(nm^2)$ [55]. We denote the inducing variables as a column vector $\mathbf{u} \in \mathbb{R}^m$. Specifically, $\mathbf{u}$ represents the values of the latent function $f(t)$ (see (2.17)) evaluated at a set of *inducing points* $\mathbf{z} = [z_1, \cdots, z_m]^\top$. That is, $\mathbf{u} = [f(z_1), \cdots, f(z_m)]^\top$. In this case, the inducing points $\mathbf{z}$ lie on the same domain as $\mathbf{t}$, i.e. time.

Recall that inference in GP regression corresponds to maximize the log marginal-likelihood (2.26) with respect to the hyperparameters $\boldsymbol{\theta}$. Using (2.25), the objective function $J(\boldsymbol{\theta}) = \log p(\mathbf{y})$ can be written as

$$J(\boldsymbol{\theta}) = \log \mathcal{N}\left(\mathbf{y} |\ \mathbf{0},\ \nu^2\mathbf{I} + \mathbf{K}\right). \tag{2.35}$$

On the other hand, in sparse GPs the goal is to maximize an approximation of the log-marginal likelihood (2.35), resulting in the following objective function

$$\hat{J}(\boldsymbol{\theta}) = \log \mathcal{N}\left(\mathbf{y} |\ \mathbf{0},\ \nu^2\mathbf{I} + \mathbf{Q}\right), \tag{2.36}$$

where $\mathbf{Q}$ is an approximation of the true prior covariance matrix $\mathbf{K}$ (see (2.20)) [76]. For example, in [82] the Nyström method was used for approx-

imating the matrix $\mathbf{K}$, resulting in

$$\mathbf{Q} = \mathbf{K}_{nm}\mathbf{K}_{mm}^{-1}\mathbf{K}_{mn}, \tag{2.37}$$

where $\mathbf{K}_{mm}$ is the covariance matrix of the inducing variables $\mathbf{u}$, and $\mathbf{K}_{nm}$ is the cross-covariance between the inducing variables $\mathbf{u}$ and the values of the latent function $\mathbf{f}$. The comparison between the objective functions (2.35) and (2.36) reveals that sparse approximations of GPs operate by "*doing exact inference with an approximate prior*" [55].

2.6.3 Variational inference

The purpose behind VI is to rewrite Bayesian inference as an optimisation problem [20]. In Bayesian inference the main goal is to compute the posterior distribution over the latent variables $\mathbf{z}$, given the observations $\mathbf{x}$. If computing the posterior $p(\mathbf{z}|\mathbf{x})$ is intractable, then approximate methods are required. Approximate variational inference methods define an objective function that measures the *distance* between the intractable posterior and a variational distribution $q(\mathbf{z})$. The distance metric most frequently used is the Kullback-Leibler (KL) divergence, which quantifies how similar $q(\mathbf{z})$ and $p(\mathbf{z}|\mathbf{x})$ are. The KL divergence is written as

$$\mathrm{KL}[q(\mathbf{z})||p(\mathbf{z}|\mathbf{x})] = -\int q(\mathbf{z}) \log \frac{p(\mathbf{z}|\mathbf{x})}{q(\mathbf{z})} \mathrm{d}\mathbf{z}, \tag{2.38}$$

and follows the proprieties $\mathrm{KL}[q(\mathbf{z})||p(\mathbf{z}|\mathbf{x})] \geq 0$, and $\mathrm{KL}[q(\mathbf{z})||p(\mathbf{z}|\mathbf{x})] = 0$ only when $q(\mathbf{z}) = p(\mathbf{z}|\mathbf{x})$ [19].

The elegance of VI lies on the fact that to minimise the KL divergence is not necessary to compute the intractable posterior. Minimising (2.38) is equivalent to maximising the *evidence lower bound* (ELBO) [20]. The derivation of the ELBO comes from applying the Jensen's inequality to the

log marginal likelihood [88]

$$\begin{aligned}\log p(\mathbf{x}) &= \log \int p(\mathbf{x}, \mathbf{z}) \mathrm{d}\mathbf{z} \\ &= \log \mathbb{E}_{q(\mathbf{z})}\left[\frac{p(\mathbf{x}, \mathbf{z})}{q(\mathbf{z})}\right] \mathrm{d}\mathbf{z} \geq \mathbb{E}_{q(\mathbf{z})}\left[\log \frac{p(\mathbf{x}, \mathbf{z})}{q(\mathbf{z})}\right] \mathrm{d}\mathbf{z}.\end{aligned}$$

The ELBO $\mathcal{L}(\boldsymbol{\theta})$ follows

$$\begin{aligned}\mathcal{L}(\boldsymbol{\theta}) &\equiv \mathbb{E}_{q(\mathbf{z})}\left[\log \frac{p(\mathbf{x}|\mathbf{z})p(\mathbf{z})}{q(\mathbf{z})}\right] \mathrm{d}\mathbf{z}, \\ &= \mathbb{E}_{q(\mathbf{z})}\left[\log p(\mathbf{x}|\mathbf{z})\right] - \mathrm{KL}\left[q(\mathbf{z})||p(\mathbf{z})\right]\end{aligned} \tag{2.39}$$

where $\boldsymbol{\theta}$ are the parameters of the variational distribution. The ELBO (2.39) only depends on the model likelihood $p(\mathbf{x}|\mathbf{z})$, prior $p(\mathbf{z})$, and variational distribution $q(\mathbf{z})$. In short, to maximise the ELBO, it is not necessary to calculate the intractable posterior. This is how VI transforms Bayesian inference into an optimisation problem.

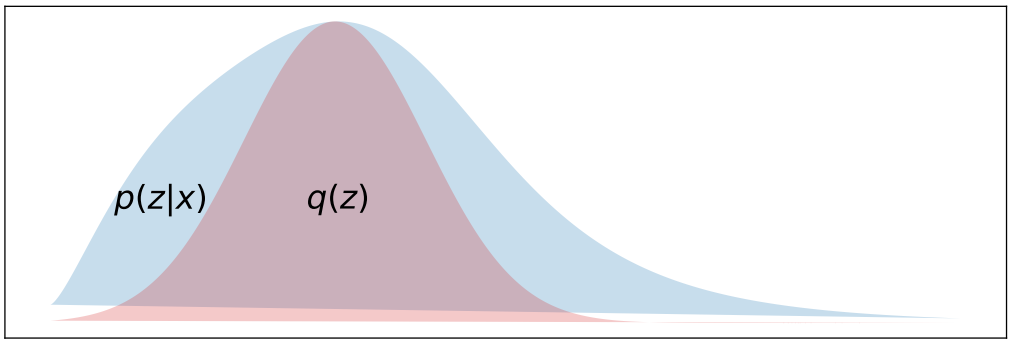


Figure 2.12: Example variational inference.

2.6.4 Variational inference for sparse GPs

Variational inference has substantially influenced the research community working on GPs. Particularly, Titsias in [76] proposed the first sparse approximate variational inference method for Gaussian process. This approach jointly learns the inducing points $\mathbf{z}$ and the kernel parameters $\boldsymbol{\theta}$ by maximiz-

ing a lower bound of the true log marginal likelihood (2.35). This operation is equivalent to minimizing the KL divergence (2.38) between the approximate distribution and the true posterior. The variational approach proposed in [76] presents two advantages in comparison to previous sparse GP methods [55]. First, it avoids overfitting by treating the inducing points $\mathbf{z}$ as variational parameters. Second, it rigorously approximates the real GP model (when the likelihood is Gaussian), by minimizing the Kullback-Leibler (KL) divergence between the Gaussian approximate distribution $q(\mathbf{u})$, and the true Gaussian posterior $p(\mathbf{f}|\mathbf{y})$. This approach leads to the following objective function, called evidence lower bound (ELBO):

$$\mathcal{L}(\boldsymbol{\theta}) = \log \mathcal{N}\left(\mathbf{y}|\ \mathbf{0},\ \mathbf{Q} + \nu^2\mathbf{I}\right) - \frac{1}{2\nu^2}\mathrm{tr}\left(\mathbf{K} - \mathbf{Q}\right), \tag{2.40}$$

where the matrix $\mathbf{Q} = \mathbf{K}_{nm}\mathbf{K}_{mm}^{-1}\mathbf{K}_{mn}$ is calculated following (2.37). The ELBO (2.40) runs in time $\mathcal{O}(nm^2)$, where n is the size of the data, and m the number of inducing points [76]. Comparing (2.40) with the objective function of previous GP sparse methods (2.36), we observe that there is a new regularization trace term, which depends on the difference between the variance of the true and the approximate covariance matrix.

2.6.5 Gaussian process stochastic variational inference

Variational inference has allowed the application of sparse GPs models to large datasets [33]. Specifically, Hensman et al. [34] first introduced stochastic variational inference (SVI) into Gaussian process models, opening the door for big-data scenarios, such as audio signal processing. In a broad sense, SVI operates as follows: first, mini-batches of the training data are selected randomly and used to approximate the expected value of the likelihood under the approximate distribution. Subsequently, the obtained approximate lower bound is maximized in order to update a set of *global* variables [35]. In this way, SVI outperforms traditional VI in terms of efficiency. In the following section we describe a variational evidence lower bound (ELBO) for sparse GPs that can be optimized in a stochastic manner.

An ELBO for stochastic variational inference

One of the properties of the lower bound introduced in [76] (see (2.40)) is that the inducing variables $\mathbf{u}$ are "collapsed" or marginalized [36]. However, in order to make SVI suitable for sparse GPs, it is necessary to keep an explicit representation of $\mathbf{u}$ thorough the variational distribution $q(\mathbf{u})$, as they represent the global variables to be optimized throughout the data mini-batches [34]. The lower bound described below has an explicit variational distribution over the inducing variables $q(\mathbf{u})$, therefore, it can be maximized by using SVI. The variational distribution over the latent variables has the form

$$q(\mathbf{u}) = \mathcal{N}\left(\mathbf{u}|\ \mathbf{m}, \mathbf{S}\right), \tag{2.41}$$

where the the covariance matrix is parametrized using a lower-triangular form $\mathbf{S} = \mathbf{L}\mathbf{L}^\top$ to preserve $\mathbf{S}$ as positive semi-definite [36]. In addition, using the conditional property of the Gaussian distribution (appendix A), and the joint distribution

$$p(\mathbf{f}, \mathbf{u}) = \mathcal{N}\left(\begin{bmatrix}\mathbf{f}\\ \mathbf{u}\end{bmatrix} \middle| \mathbf{0}, \begin{bmatrix}\mathbf{K}_{nn} & \mathbf{K}_{nm}\\ \mathbf{K}_{nm}^\top & \mathbf{K}_{mm}\end{bmatrix}\right), \tag{2.42}$$

then the distribution over the latent vector $\mathbf{f}$ (2.20) conditioned to the inducing variables $\mathbf{u}$ corresponds to

$$p(\mathbf{f}|\mathbf{u}) = \mathcal{N}\left(\mathbf{K}_{nm}\mathbf{K}_{mm}^{-1}\mathbf{u},\ \mathbf{K}_{nn} - \mathbf{Q}_{nn}\right), \tag{2.43}$$

where $\mathbf{Q}_{nn} = \mathbf{K}_{nm}\mathbf{K}_{mm}^{-1}\mathbf{K}_{nm}^\top$. The distributions (2.41) and (2.43) are the two pieces necessary to define the variational distribution over $\mathbf{f}$, that is,

$$q(\mathbf{f}) = \int p(\mathbf{f}|\mathbf{u})q(\mathbf{u})\mathrm{d}\mathbf{u}, \tag{2.44}$$

which has the form

$$q(\mathbf{f}) = \mathcal{N}\left(\mathbf{K}_{nm}\mathbf{K}_{mm}^{-1}\mathbf{m},\ \mathbf{K}_{nn} + \mathbf{K}_{nm}\mathbf{K}_{mm}^{-1}\left(\mathbf{S} - \mathbf{K}_{mm}\right)\mathbf{K}_{mm}^{-1}\mathbf{K}_{nm}^\top\right).$$

The resulting lower bound of the marginal likelihood has the form

$$\log p(\mathbf{y}) \geq \mathbb{E}_{q(\mathbf{f})}\left[\log p(\mathbf{y}|\mathbf{f})\right] - \mathrm{KL}\left[q(\mathbf{u})|p(\mathbf{u})\right] = \hat{\mathcal{L}}(\boldsymbol{\theta})$$

where $p(\mathbf{y}|\mathbf{f})$ corresponds to the likelihood (2.19), and $p(\mathbf{u}) = \mathcal{N}(\mathbf{0}, \mathbf{K}_{mm})$ to the true prior over the inducing variables. Given that the likelihood (2.19) factorises thorough the data, then the lower bound follows

$$\begin{aligned}\hat{\mathcal{L}}(\boldsymbol{\theta}) &= \mathbb{E}_{q(\mathbf{f})}\left[\log \prod_{i=1}^{n} p(y_i|f_i)\right] - \mathrm{KL}[q(\mathbf{u})|p(\mathbf{u})] \\ &= \sum_{i=1}^{n} \mathbb{E}_{q(f_i)}[\log p(y_i|f_i)] - \mathrm{KL}[q(\mathbf{u})|p(\mathbf{u})],\end{aligned} \tag{2.45}$$

where $q(f_i)$ is the i-th marginal of $q(\mathbf{f})$. We observe that maximizing the lower bound (2.45) requires to compute n expected values (recall n is the data size), as well as computing the KL divergence between the prior and the approximate distribution [36]. Therefore, the more data we have the more integrals (expectations) we need to solve, demanding more time per iteration. For big data scenarios, such as complete audio signals, inference becomes quite slow or intractable. The advantage of the objective function (2.45) is that it can be optimized using stochastic variational inference. First, the sum of n expectations in (2.45) is approximated using a mini-batch sampled independently from the data. Next, the obtained approximation of (2.45) is optimized to learn the global parameters corresponding to the mean $\mathbf{m}$ and covariance matrix $\mathbf{S}$ of the variational distribution (2.41) [36]. This procedure repeats in a loop until convergence of the global parameters.

For a more detailed derivation of this kind of lower bounds refer to [34, 36], and Section 5.2.1 of this thesis, where an ELBO is introduced for the modulated-GP, i.e. a regression model based on the product of two GPs [4]. This thesis applies variational inference in Chapters 4 and 5.1, and stochastic variational inference in Section 5.2. But first, in Chapter 3 we investigate how to encode musical-acoustic knowledge into our GP models, while learning the hyperparameters by maximizing the true marginal likelihood (2.36).

Chapter 3

Gaussian processes for music audio content analysis

3.1 Introduction

Although music recordings are highly diverse, they have a strong underlying structure. This statistical structure, together with the physical mechanisms by which sounds are generated, can be naturally introduced into Automatic Music Transcription (AMT) as prior knowledge using Bayesian modelling. We present a Bayesian approach for modelling music audio and content analysis. The proposed methodology based on Gaussian processes seeks joint estimation of multiple music concepts by incorporating into the kernel prior information about non-stationary behaviour, dynamics, and intricate spectra present in the modelled music signal. We illustrate the benefits of this approach via two tasks: pitch estimation and inferring missing segments in a polyphonic audio recording.

Real music signals are highly variable, but nevertheless they have strong statistical structure. Prior information about the underlying structures, such as knowledge of the physical mechanisms by which sounds are generated, and knowledge about the rules by which complex sound structure is compiled (notes, chords, a complete musical score), can be naturally unified using Bayesian hierarchical modelling techniques. This allows the formulation of

highly structured probabilistic models [22]. On the other hand, typically, algorithms for AMT are developed independently to carry out individual tasks such as multiple-F0 detection, beat tracking and instrument recognition. The challenge remains to combine these algorithms, to perform joint estimation of all parameters [15].

We present the design, implementation, and results of experiments of an alternative Bayesian approach for audio content analysis on monophonic, and polyphonic music signals with the possibility of being used for AMT. We use Gaussian process (GP) models for jointly uncovering music concepts from audio, by introducing a direct connection between the music concepts and the model hyper-parameters. The proposed methodology allows to incorporate in the model prior information about physical or mechanistic behaviour, nonstationarity, time dynamics (local periodicity, and non constant amplitude envelope), spectral harmonic content, and musical structure, latent in the modelled music signal. Specifically in the context of music informatics, we present kernels that embody a probabilistic model of music notes as time-limited harmonic signals with onsets and offsets. The presented approach can describe polyphonic signals, by encouraging partial or complete overlapping between the latent processes that represent each sound event or music note. A comparison with related work is provided in section 3.3.4. We illustrate the benefits of this approach via two tasks: pitch estimation in monophonic music and inferring missing segments in a polyphonic audio recording.

3.2 Kernel design

The covariance function (2.4) used for computing the prior distribution (2.5) allows us to introduce in the model the knowledge and beliefs we have about the properties of music signals. Some of the broad properties of music signals are non-stationarity, rich spectral content, dynamics (quasi-periodicity, and non-constant amplitude envelope), mechanistic patterns, and music-theory structure. Our goal is to design covariance functions that encode most these properties.

One technique for constructing new kernels is to build them out of simpler

kernels as building blocks [66, 19]. Two useful properties we can use to build valid kernels are

$$k(t, t') = \phi(t)k_1(t, t')\phi(t'), \tag{3.1}$$

$$k(t, t') = k_1(t, t') + k_2(t, t') \tag{3.2}$$

where $\phi(\cdot)$ is any function. Other properties can be found in [19]. We use these properties for building non-stationary covariance functions [19]. To construct non-stationary kernels we combine basic stationary covariance functions. We use *change-windows* in order to be able to model notes or sound events which are not continuously active but have a beginning and an ending in the music signal. As in [42] we define a change-window by multiplying two sigmoid functions. The parameters of the change-windows are directly related with the location, onset and offset of the sound events. In the present work we will use manually-specified onset/offset locations. Here we assume the complete process $f(t)$ is a linear combination of M Gaussian processes, representing each one a note or sound event. In this way

$$f(t) = \sum_{m=1}^{M} \phi_m(t) f_m(t), \tag{3.3}$$

where each GP in the set $\{f_i(t)\}_{i=1}^{M}$ is independent with respect to each other, i.e.

$$\mathbb{E}[f_i \cdot f_j] = \mathbb{E}[f_i]\mathbb{E}[f_j] = 0 \cdot 0 = 0, \tag{3.4}$$

for $i \neq j$. This is because the mean of each GP is zero. In addition, M corresponds to the number of notes or sound events in the signal. On the other hand, $\{\phi_m(t)\}_{m=1}^{M}$ are the respectively change-windows that allow a specific GP $f_m(t)$ to appear or vanish in certain parts of the input space (time). In this sense the proposed approach can handle polyphonic signals, by encouraging partial or complete overlapping between change-windows.

3.2.1 General form of the change-windows

The change-windows are defined as the multiplication of two sigmoid functions [42], that is

$$\begin{aligned}\phi_m(t) &= \frac{1}{1+e^{-\varsigma_m(t-\alpha_m)}} \times \frac{1}{1+e^{-\varsigma_m(\beta_m-t)}} \\ &= \left[1+e^{-\varsigma_m(\beta_m-t)}+e^{-\varsigma_m(t-\alpha_m)}+e^{-\varsigma_m(t-\alpha_m)}e^{-\varsigma_m(\beta_m-t)}\right]^{-1},\end{aligned}$$

where parameter ς_m determine how fast or slow the sigmoid function rises to one or falls to zero, whereas α_m, β_m defines the onset and the offset of the window respectively. We assume that $\alpha < \beta$, i.e. the location of the onset of the change-window should be before the location of its offset, then

$$\phi_m(t) = \left[1+e^{-\varsigma_m(\beta_m-t)}+e^{-\varsigma_m(t-\alpha_m)}+e^{-\varsigma_m(\beta_m-\alpha_m)}\right]^{-1}. \tag{3.5}$$

It can be shown that the covariance function for $f(t)$ in (3.3) is given by

$$k_f(t,t') = \sum_{m=1}^{M} \phi_m(t) k_m(t,t') \phi_m(t'). \tag{3.6}$$

The derivation of (3.6) is as follows

$$\begin{aligned}\mathrm{Cov}\left[f(t), f(t')\right] &= \mathbb{E}\left[f(t)f(t')\right] \\ &= \mathbb{E}\left[\sum_{m=1}^{M} \phi_m(t) f_m(t) \sum_{m'=1}^{M} \phi_{m'}(t') f_{m'}(t')\right] \\ &= \sum_{m=1}^{M}\sum_{m'=1}^{M} \phi_m(t) \mathbb{E}\left[f_m(t) f_{m'}(t')\right] \phi_{m'}(t') \\ &= \sum_{m=1}^{M}\sum_{m'=1}^{M} \phi_m(t) \left[\delta_{m,m'} k_{m,m'}(t,t')\right] \phi_{m'}(t') \\ &= \sum_{m=1}^{M} \phi_m(t) k_m(t,t') \phi_m(t'),\end{aligned}$$

where $\delta_{m,m'}$ is the Kronecker delta. We assume each GP $f_m(t)$ in (3.3) is stationary.

3.2.2 Studied covariance functions

In the experiments of this chapter we compared three different kernels: the exponentiated quadratic $k_{\mathrm{EQ}}(\tau)$ (2.11), the standard periodic $k_{\mathrm{SP}}(\tau)$ (2.15), and the *exponentiated quadratic periodic*, which corresponds to multiply the kernels (2.11) and (2.15), that is

$$\begin{aligned} k_{\mathrm{EQP}}(\tau) &= k_{\mathrm{EQ}}(\tau) \times k_{\mathrm{SP}}(\tau) \\ &= \sigma^2 \exp\left(z\cos(\omega\tau) - \frac{\tau^2}{2l^2}\right), \end{aligned} \tag{3.7}$$

here, we recall the definition of the exponentiated quadratic kernel (2.11) as

$$k_{\mathrm{EQ}}(\tau) = \sigma^2 \exp\left(-\frac{\tau^2}{2l^2}\right), \tag{3.8}$$

and parameterize the standard periodic covariance function (2.15) as

$$k_{\mathrm{SP}}(\tau) = \sigma^2 \exp\left(z\cos(\omega\tau)\right). \tag{3.9}$$

The form of these kernels is shown in Fig. 3.1(a, b, c). The hyperparameters used to generate Fig. 3.1 were $\sigma^2 = 1.0$, $l = 0.125$, $z = 1.0$ and $\omega = 2\pi 12$. In a GP with an exponentiated quadratic kernel (3.8), the dependency between any two function values $f(t)$ and $f(t')$ decreases with the time-lag between them ($\tau = t - t'$) (Fig. 3.1a). Therefore, function values will be similar if they are close in time, that is, the realizations sampled from this GP are smooth (Fig. 3.1d). On the order hand, in a GP with an standard periodic kernel (3.9), the dependency between any two function values changes in a periodic pattern that depends on the time lag τ, and has period $T = \omega^{-1}$ (see Fig. 3.1b). As a result, function values whose time distance is an integer value of the period, that is $\{f(\tau + nT)\}$ for $n = 0, 1, 2, \ldots$, will be highly dependent. In other words, the sampled function will be periodic (Fig. 3.1e). Finally, a

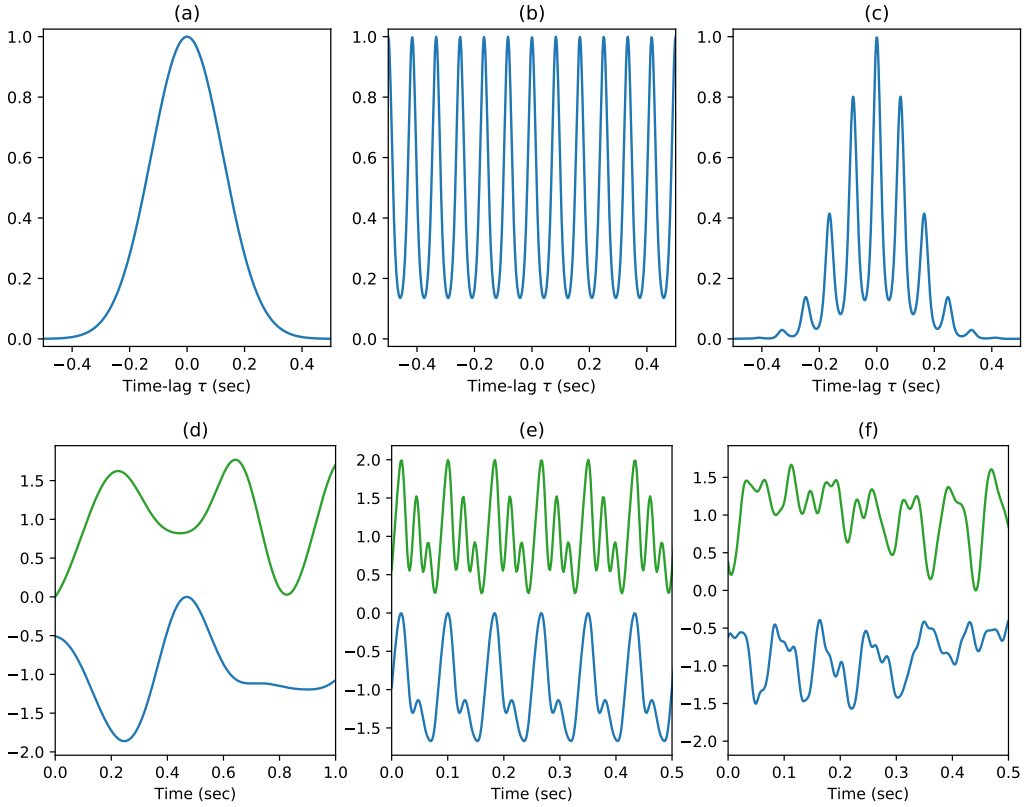


Figure 3.1: (a, b, c) Form of the analysed kernels: exponentiated quadratic $k_{\text{EQ}}(\tau)$, standard periodic $k_{\text{SP}}(\tau)$, and exponentiated quadratic $\times$ standard periodic $k_{\text{EQP}}(\tau)$, respectively. Here, the hyperparameters had the values $\sigma^2 = 1.0$, $l = 0.125$, $z = 1.0$ and $\omega = 2\pi 12$. (d, e, f) Samples from a GP with kernel: $k_{\text{EQ}}(\tau)$, $k_{\text{SP}}(\tau)$, and $k_{\text{EQP}}(\tau)$, respectively.

GP with kernel (3.7) shares similar properties of the two previous examples. Specifically, the dependency between any two function values decreases with the time-lag, while following a periodic pattern (Fig. 3.1c). As a result, the functions sampled will present not-perfectly periodic oscillations (Fig. 3.1f). Recall that the covariance function shown in Fig. 3.1c corresponds to multiply the ones shown in Fig. 3.1a and Fig. 3.1b.

Spectral density of covariance functions

The Fourier transform (FT) of the kernels exponentiated quadratic (3.8), standard periodic (3.9), and exponentiated quadratic $\times$ standard periodic

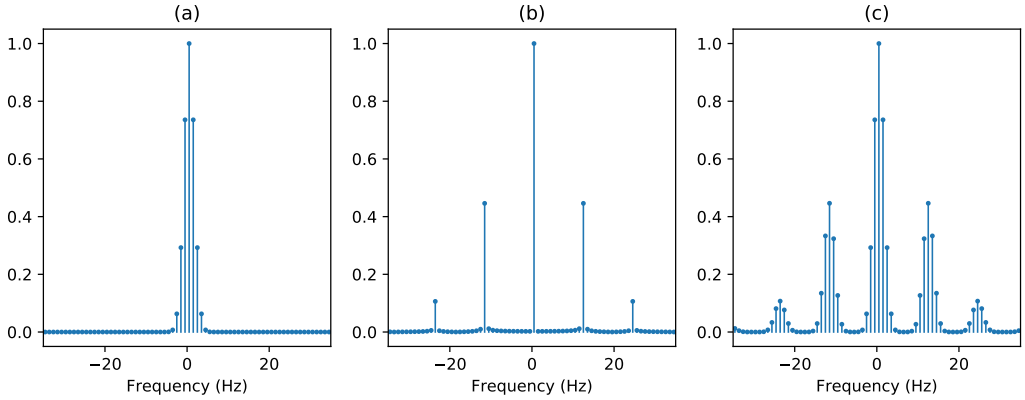


Figure 3.2: Spectral density of $k_{\text{EQ}}(\tau)$ (a), $k_{\text{SP}}(\tau)$ (b), and $k_{\text{EQP}}(\tau)$ (c).

(3.7) are shown in Fig. 3.2(a, b, c) respectively. The hyperparameters used to compute the kernels were the same as in Fig. 3.1. The covariance function (3.8) is probably the most widely-used kernel within the kernel machines field. A Gaussian process with a exponentiated-quadratic covariance function is infinitely smooth [56]. The spectral density of the GP with kernel (3.8) contains only low frequency components and does not have any harmonic structure (Fig. 3.2a). That is why the realizations shown in Fig. 3.1d, sampled from a GP with kernel $k_{\text{EQ}}(\tau)$, evolve smoothly without any periodic or harmonic properties.

On the other hand, the spectral density of the standard periodic kernel (3.9) shown in Fig. 3.2b, only has energy at 0Hz (zero Hertz) as well as at frequencies $\{n \times 12\text{Hz}\}_{n=1}^{\infty}$, that is, at a natural frequency $f_0 = 12\text{Hz}$ and its harmonics (integer numbers of 12Hz). Fig. 3.1d shows two functions sampled from a GP with covariance function (3.9). These realizations present constant amplitude-envelope and periodic properties with a fundamental frequency together with several harmonics. However, the spectrum and amplitude-envelope of real audio signals of music instruments evolve dynamically in time, i.e., they are not constant (Fig. 3.3a).

The kernel (3.7) does not present the limitations imposed by using the standard periodic kernel alone, that is, the covariance (3.7) allows to describe functions where its amplitude-envelope and spectrum changes in time. Figure 3.2c depicts the FT of (3.7). We observe that this spectral density is similar

to the one obtained for the standard periodic kernel (3.9) (Figure 3.2b), in the sense that the energy is distributed around a set of frequencies corresponding to a natural frequency and its harmonics (including the constant *harmonic* at 0Hz). However, the main difference is that the energy also spreads around these set of harmonics. This spread has the same shape as the spectral density of the exponentiated quadratic kernel Figure 3.2a. This is because the product of two functions in time, corresponds to the convolution of its FT. In short, the realizations sampled from a GP with covariance function (3.7) show two relevant properties of music signals: a non-constant amplitude envelope, and a periodic structure with a natural frequency and harmonics that evolve in time (Figure 3.1f). Therefore, the covariance function (3.7) seems to be more appropriate for modelling music signals in comparison with the two kernels presented previously ((3.8)-(3.9)). The hyper-parameter ω in (3.9)-(3.7) corresponds to the natural frequency or F_0 of the modelled random processes.

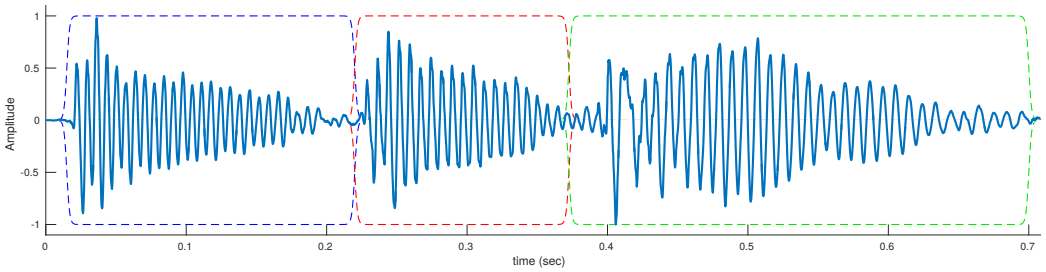
3.3 Results and discussion

Experiments were done over real audio. We evaluated the performance of different kernel on pitch estimation, and inferring missing data. All experiments assume we previously know the number of change-windows and its locations. In the pitch estimation task all the parameters of the covariance function are known, except those related with the fundamental frequency of each sound event, i.e. the value of ω_m in (3.9) and (3.7) when using these kernels in the general model (3.3). Thus, we focus on optimizing only these model hyperparameters from the data. In the missing data imputation task the score of the modelled piece of music audio is used for tuning manually the model hyperparameters.

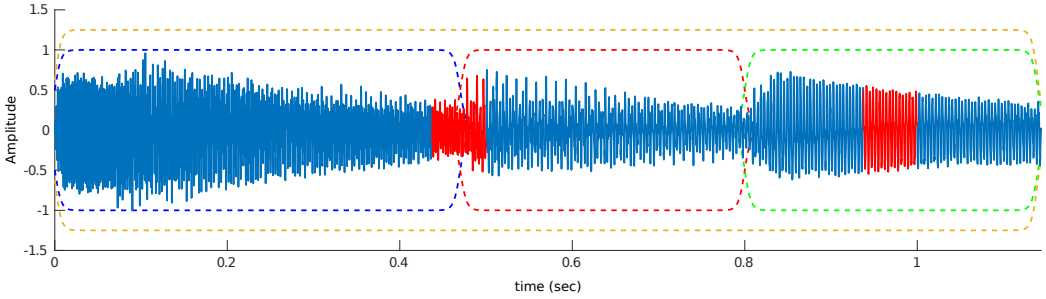
3.3.1 Data

In these experiments we used two short audio excerpts. The low size of the data allows us to compute the closed-form predictive distribution (2.29),

and learn the model hyperparameters by maximizing the marginal-likelihood (2.26). The excerpt used for pitch estimation experiments corresponds to 0.7 seconds of the song *Black Chicken 37* by Buena Vista Social Club. This segment of audio contains three notes of a bass melody (Figure 3.3a). In the missing data imputation task we used polyphonic audio corresponding to 1.14 seconds of Chopin's *Nocturne Op. 15 No. 1*, where more than one note occur at the same time. The segments of signal in red in Figure 3.3b represent gaps of missing data. We reduced the sample frequency of both audio excerpts from 44.1kHz to 8kHz.



(a) Signal used for pitch estimation.



(b) Signal used for filling missing-data gaps.

Figure 3.3: (a) analysed audio (blue line), change windows (dashed lines). (b) observed data (blue line), missing-data gaps (red line), change-windows (dashed lines).

For inference, we take an empirical Bayes approach. That is, we first learn point estimates of the model hyperparameters, and then we use the point estimates to calculate the posterior over the latent function $f(t)$. Specifically, to learn the hyperparameters we maximize the marginal likelihood, by using a standard gradient-based optimizer [52]. To do so, it is necessary to have an expression for the log-marginal likelihood and its partial derivatives with

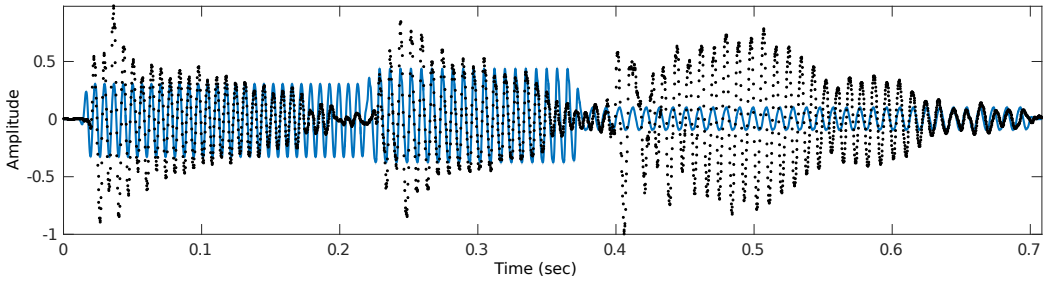
respect to the hyperparameters.

3.3.2 Pitch estimation

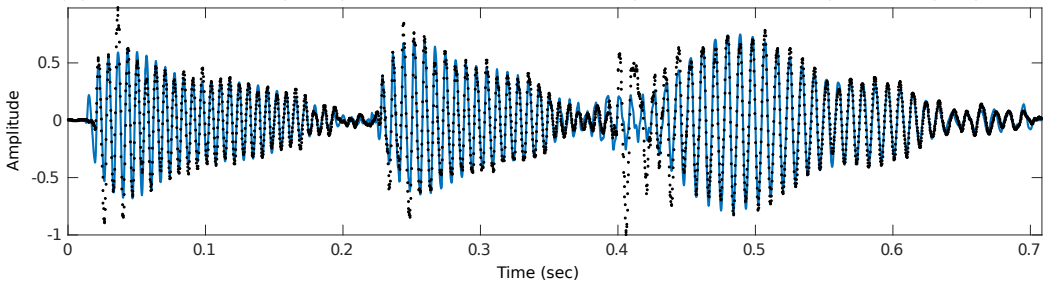
For the pitch estimation task we tested two different models with kernels (3.9), and (3.7) respectively. We performed hyperparameters learning using all the observed signal shown in Figure 3.3a. This is because in this experiment rather than evaluating the prediction of the trained models, we were interested in the accuracy of pitch estimation. Covariance function (3.8) does not have any parameter we can link to the fundamental frequency of each sound event, that is why we omitted it here. We compared the GPs models results with the algorithm pYIN, a fundamental frequency estimator [47]. The trained model using $k_{\text{SP}}(\tau)$ was able to estimate the pitch for each sound event with a RMSE of 0.6282 semitones. On the other hand, the amplitude-envelope evolution of the signal is beyond the scope of the structure that this kernel can model (See Figure 3.4a). This is because this covariance function can only describe constant amplitude-envelope, periodic signals, with a fundamental frequency and several harmonics (Figure **??**). Results using (3.7) are shown in Figure 3.4b. We observe that although the posterior mean of the predictive distribution does not exactly fit the data, the model is able to learn the pitch of each of the three sound events with a smaller RMSE of 0.1075 in comparison with the 0.1688 RMSE obtained with pYIN. Variations in the amplitude envelope can also be described using (3.7).

3.3.3 Filling gaps of missing data in audio

We compared three different models predicting missing-data gaps. We studied kernels (3.8), (3.9), and (3.7). In Figure 3.3b first gap (red segment) contains the transient, onset, and attack of a sound event [11]. In addition, the second gap is located in a more steady segment of the data (smooth decay). Figure 3.5a-3.5b depict the prediction using (3.8). These figures correspond to zoom in small sections of the signal where the gaps occur (Figure 3.3b). We see that the model using this kernel overfits the data, i.e. the posterior mean (blue line) fits all the observed data (black dots) with high



(a) Observations (dots), and posterior mean (continuous line) using (3.9).

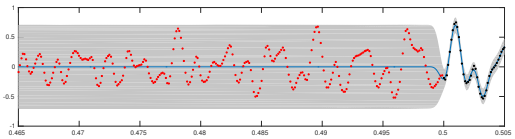


(b) Observations (dots), and posterior mean (continuous line) using (3.7).

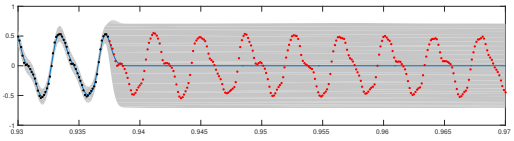
Figure 3.4: Posterior mean for the pitch estimation experiments. (a) using $k_{\text{EQ}}(\tau)$, and (b) using $k_{\text{EQP}}(\tau)$.

confidence (grey shaded area), but the confidence decreases and the prediction is quite poor in the input space zones where the data is not available (red dots). Also, we see that the model using (3.8) does not expect any periodic behaviour in the gaps.

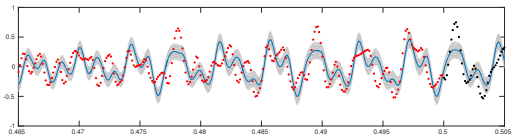
Figure 3.5c-3.5d show the prediction using covariance function (3.9). In the transient gap (Figure 3.5c) the posterior mean (blue line) does not follows the data, this is because transients are short intervals during which the signal evolves in a non-stationary, non-trivial and unpredictable way [11]. opposite to this, the model using kernel (3.9) can only describe the behaviour of constant amplitude-envelope periodic stochastic functions. In the second gap (Figure 3.5d) the posterior mean describes properly the periodic behaviour of the data, but it does not follow the amplitude-envelope of the observations. This is because this covariance function is able to describe periodic functions that have several harmonic components. The drawback of this kernel is that it assumes constant the amplitude of the periodic stochastic functions that describes.



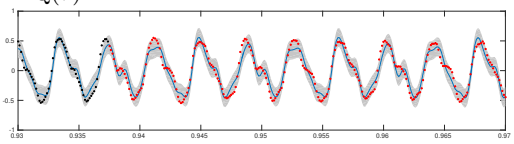
(a) Prediction on transient using $k_{\mathrm{EQ}(\tau)}$



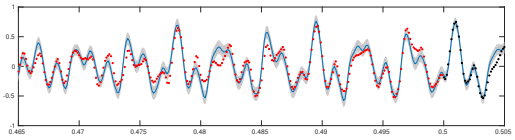
(b) Prediction on steady gap using $k_{\mathrm{EQ}(\tau)}$



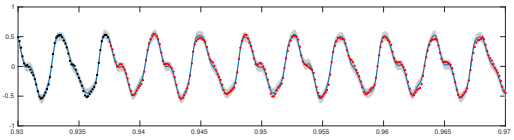
(c) Prediction on transient using $k_{\mathrm{SP}(\tau)}$



(d) Prediction on steady gap using $k_{\mathrm{SP}(\tau)}$.



(e) Prediction on transient using $k_{\mathrm{EQP}(\tau)}$.



(f) Prediction on steady gap $k_{\mathrm{EQP}(\tau)}$.

Figure 3.5: Zoom in a portion of missing-data gaps. In each figure the continuous blue line represent the posterior mean, grey shaded areas correspond to the posterior variance, red dots are missing data, whereas black dots are observed data.

Results using (3.7) are presented in Figure 3.5e-3.5f. We see that in Figure 3.5f the posterior mean describes properly the periodic behaviour and amplitude envelope smooth evolution of the modelled signal. We observe that prediction on the decay gap using (3.7) is closer to the actual data (red dots) than the results obtained with (3.9) as well as (3.8). This is because (3.7) allows to describe periodic functions that have several harmonic components and time-varying amplitude envelope. On the other hand, the prediction performance reduces for the transient gap (Figure 3.5e). In order to model the onset, attack and decay of a sound event, covariance function (3.7) could be modified for modelling non-stationary amplitude envelope evolution.

The performance of the three analysed kernels is summarized in table 3.1. As expected, the lower root mean squared error (RMSE) was obtained using the kernel able to describe periodic functions with time-varying amplitude envelope, that is, k_{EQP}. Also, the k_{EQP} kernel presented a higher error when

Table 3.1: Root mean squared error (RMSE) for the task filling gaps of missing data.

kernel	**RMSE transient gap**	**RMSE decay gap**
$k_{\text{EQ}}(\tau)$	0.2265	0.3172
$k_{\text{SP}}(\tau)$	0.2143	0.0964
$k_{\text{EQP}}(\tau)$	0.0912	0.0355

predicting the data associated with the transient gap. This suggests that it is more challenging to model the transient (first gap in Figure 3.3b) of the analysed sound, in comparison to a more steady section of the data (second gap Figure 3.3b).

3.3.4 Related work

In [77] GPs are used for time-frequency analysis as probabilistic inference. Natural signals are assumed to be formed by the superposition of distinct time-frequency components, with the analytic goal being to infer these components by applying Bayes' rule [77]. GPs have also been used for audio source separation [41, 87]. In [41] the *mixture* signal is modelled as a linear combination of independent convolved versions of latent GPs or *sources*. The model splits the *mixture* signal in frames also considered independent, by using weight-functions. Thus each source is modelled as a series of concatenated locally stationary frames, each one with its corresponding covariance function. With this assumption the resulting signal is supposed to be non-stationary [41]. On the other hand, despite the approach we present also assumes that the latent GPs f_m in (3.3) are independent, the observed signal is not framed into independent segments. Instead of using weight-functions that act over the observed data, we introduce change-windows ϕ_m influencing each latent GP ending up with latent processes representing specific sound events that happen at certain segments of time. Therefore the proposed model keeps the dependency between the observations throughout all the signal. That is what allows to make prediction in gaps of missing data (section 3.3.3). GPs have been used also for estimating spectral envelope and fundamental frequency of a speech signal [86]. Finally, GPs for music genre

classification and emotion estimation were investigated in [44].

3.4 Conclusions

We discussed a GP regression framework for modelling music audio. We compared different models in pitch estimation as well as in prediction of missing data. We showed which kernels were more appropriate for describing properties of music signals, specifically: nonstationarity, dynamics, and spectral harmonic content. The advantage of this approach is that by designing a proper kernel we can introduce into the model prior knowledge and beliefs about the properties of music signals, and use all the prior information to improve prediction. Computational complexity is an important limitation of GPs (see section 2.6.1), therefore the presented work could be extended using efficient representations to model larger audio signals. Kernels as [83] could be studied for modelling harmonic content, and Latent Force models [9] for describing mechanistic characteristics.

Chapter 4

Efficient learning of harmonic priors for pitch detection

4.1 Introduction

Automatic music transcription (AMT) aims to infer a *latent* symbolic representation of a piece of music (piano-roll), given a corresponding *observed* audio recording. Transcribing polyphonic music, that is, music recordings where multiple notes can be played simultaneously, is a challenging problem. This is because of the highly structured overlapping between the spectra of concurrent sound events. We study whether the introduction of acoustically inspired Gaussian process (GP) priors into audio content analysis models improves the extraction of patterns required for AMT. Here audio signals are described as a linear combination of a finite number of functions we call *sources*. In addition, each source is decomposed into the product between an *activation* process, and a quasi-periodic *component* process. For each source, the activation controls its amplitude-envelope, whereas the component contains its spectrum. We introduce the Matérn spectral mixture (MSM) kernel for describing frequency content of singles notes. We consider two different regression approaches. On one hand, in the *sigmoid* model every source activation is independently non-linear transformed. On the other hand, in the *softmax* model the activation GPs are jointly non-linearly transformed. This

introduce cross-correlation between activations. We use variational Bayes for approximate inference. We empirically evaluate how these models work in practice transcribing polyphonic music. We found that rather than encourage dependency between activations, what is relevant for improving pitch detection is to learn priors that fit the frequency content of the sound events to be detected.

In the research field of music information retrieval, the aim of audio content analysis is to infer underlying musical concepts, such as pitch, melody, chords, onset, beat, tempo, rhythm, which are present but hidden in the audio data [64]. Then, perhaps the most general application is recovering the score (symbolic representation) of a music track given only the audio recording [50]. This is known as automatic music transcription (AMT) [15]. Transcribing polyphonic music (when multiple notes are played simultaneously) is a challenging problem, especially in its more unconstrained form when the task is performed on an arbitrary acoustical input [14]. This is because simultaneous notes cause a highly structured overlap of harmonics in the acoustic signal [67].

Moreover, a single note produced by a music instrument is not just a fixed-duration sine wave, with a single frequency. It rather has a full spectrum of harmonics, as well as an attack and decay in its intensity. These spectrum evolution is instrument dependent, and therefore must be learned in a recording-specific manner. The polyphony together with complex harmonic structure of sound events creates a source-separation problem at the heart of the transcription task [16, 75].

We seek to take advantage of the underlying structure that music acoustic signals have [22]. Specifically, we aim to develop audio content analysis Bayesian models that naturally bring together prior knowledge about the underlying acoustical mechanisms that govern the nature of acoustic music signals. To do so, we introduce spectrum patterns in the prior of probabilistic models. Our method is based on Gaussian processes (GPs). GPs have been extensively used for modelling audio recordings. GPs were used to consider time-frequency analysis as probabilistic inference [77], source separation [41, 87, 3], and for estimating spectral envelope and fundamental frequency

of a speech signal [86]. GPs for music genre classification and emotion estimation were investigated in [44]. Also, in [54] a mixture of Gaussian process experts was used for predicting sung melodic contour with expressive dynamic fluctuations.

Similar to [4], we propose a regression model where the data is described as the multiplication of two GPs. Here, several GPs are jointly non-linear transformed using the *softmax* function. We call this the softmax model. This comes as a principled way to introduce dependency between pitch activations, encouraging them to reflect two properties: non-negativity, and sparsity; to enable few pitches to be active at certain time. We introduce what we call the *Matérn spectral mixture* (MSM) kernel. In order describe the harmonic content of sound events. Quite similar to [83], we model a spectral density as a mixture of basis functions. The difference is that here the basis functions, rather than Gaussians, are Lorentzian functions [37]. This corresponds to the Fourier transform of the Matérn-$\frac{1}{2}$ kernel [33]. In this work, we use the Matérn spectral mixture covariance function to encourage the model prior to reflect the clearly evident complex harmonic content present in mixture signals which can be learned in advance from of isolated sounds. Third, we increase the model scalability through approximate methods using variational inference, enabling the analysis of audio signals with several seconds of duration. Finally, in comparison with the model presented in [7], with the proposed approach the amount of model parameters becomes independent of the total sound events present in the audio recording. Moreover, to know *a priori* the number of sound events becomes inessential, as this quantity is learned directly from audio.

This chapter is organized as follows. Section 4.2 introduces the GPs model for pitch detection. Two different variants of the base model are presented in sections 4.2 and 4.2. In section 4.2.2, we provide details for learning in frequency domain the parameters of the MSM kernel. We empirically evaluated how the proposed framework works in practice transcribing polyphonic music recordings (section 4.3.1). Final conclusions are given in section 4.4.

4.2 Gaussian processes for pitch detection

Recall that automatic music transcription aims to infer a *latent* symbolic representation, such as piano-roll or score, given an *observed* audio recording. Piano-roll refers to a matrix representation of musical notes across time [15, 22]. We used GPs for modelling both, amplitude-envelope and component functions. From a Bayesian latent variable perspective [20], transcription consists in updating our beliefs about the symbolic description for a certain piece of music, after observing a corresponding audio recording. As in [87], we approach the transcription problem from a time-domain source separation perspective. That is, given an audio recording $\mathcal{D} = \{y_n, t_n\}_{n=1}^{N}$, we seek to formulate a generative probabilistic model that describes how the observed polyphonic signal (mixture of sources) was generated. Moreover, this allows us to infer the latent variables associated with the piano-roll representation. To do so, we use the regression model $y_n = f(t_n) + \epsilon_n$, where y_n is the value of the analysed polyphonic signal at time t_n, the noise follows a normal distribution $\epsilon_n \sim \mathcal{N}(0, \sigma^2)$, and the function $f(t)$ is a random process composed by a linear combination of M *sources* $\{f_m(t)\}_{m=1}^{M}$. Each source is decomposed into the product of two factors, an amplitude-envelope or activation function $\phi_m(t)$, and a quasi-periodic or component function $w_m(t)$. Putting all this together we get the following modulated-GP regression model

$$y(t) = \sum_{m=1}^{M} \phi_m(t) w_m(t) + \epsilon(t). \tag{4.1}$$

We interpret the set $\{w_m(t)\}_{m=1}^{M}$ as a dictionary where each component $w_m(t)$ is a GP with a defined fundamental frequency or pitch. Likewise, each activation GP in $\{\phi_m(t)\}_{m=1}^{M}$ represents a row of the posteriogram-matrix, i.e the time dependent non-negative activation of a specific pitch throughout the analysed piece of music. Similar to the graph presented [4], Figure 4.1 shows the graphical model of equation (4.1). To keep the graph uncluttered we omitted the unobserved noise variance σ^2, as well as the set of hyperparameters associated to each of the M activations $\{\phi_m(t)\}_{m=1}^{M}$, and components $\{w_m(t)\}_{m=1}^{M}$.

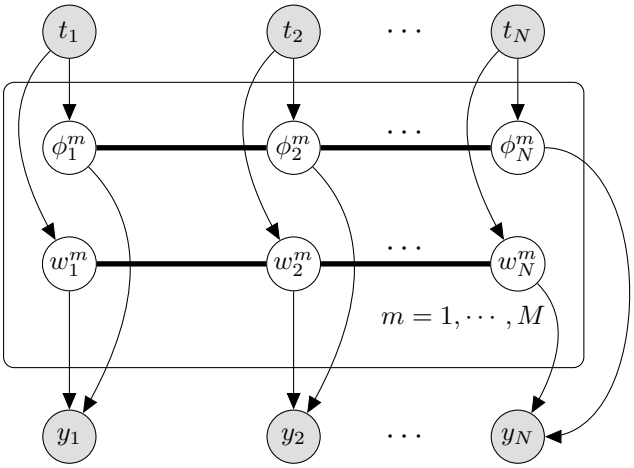


Figure 4.1: Graphical model of the proposed approach (see equation (4.1)). At each time t_n, the observed data y_n depends on two sets of M latent variables $\{w_m(t_n)\}_{m=1}^M$, and $\{\phi_m(t_n)\}_{m=1}^M$ respectively. The thick horizontal lines represent a set of fully connected nodes [56].

The components $\{w_m(t)\}_{m=1}^M$ follow $w_m(t) \sim \mathcal{GP}(0, k_m(t,t'))$, where the covariance $k_m(t,t')$ reflects the frequency content of the m^{th} component, and has the form of a MSM kernel (section 4.2.1). In [7] only the component functions followed GPs, whereas the amplitude-envelopes were parametric functions. Here the flexibility of activations $\{\phi_m(t)\}_{m=1}^M$ increases by treating them as GPs non-linearly transformed either independently or jointly, by using the sigmoid function (section 4.2) or the softmax function (section 4.2) respectively.

Sigmoid model

To guarantee the activations to be non-negative we apply non-linear transformations to GPs. To do so, we use the sigmoid function $\sigma(x) = [1 + \exp(-x)]^{-1}$, also applied in GP binary classification [56]. In the sigmoid model an activation is defined as $\phi_m(t) = \sigma(g_m(t))$, where the set of functions $\{g_m(t)\}_{m=1}^M$ are independent GPs. The sigmoid model follows

$$y(t) = \sum_{m=1}^{M} \sigma(g_m(t)) w_m(t) + \epsilon(t). \tag{4.2}$$

This formulation does not introduce any dependency between the activations.

Softmax model

To enhance sparsity on the activations we use the *softmax*, or *normalized exponential* function. Therefore

$$\phi_m(t) = \frac{\exp(g_m(t))}{\sum_{\forall j} \exp(g_j(t))}, \tag{4.3}$$

where $\{g_j(t)\}_{j=1}^{M}$ are GPs [52, 19]. Similarly to the sigmoid function, the *softmax* (4.3) enforces the activations to be non-negative as well as to be bounded between 0 and 1. Furthermore, (4.3) introduces dependences between all activations. The sparsity is enhanced because $\sum_{\forall m} \phi_m(t) = 1$, for all t. With this property we can encourage to activate only one or a few pitches at certain time. This is because if the j-th pitch explains better the audio signal at time t_n, then the activation $\phi_j(t_n) \approx 1$, therefore it follows that the other activations $\phi_i(t_n) \approx 0$ for all $i \neq j$. The softmax model corresponds to

$$y(t) = \frac{1}{\sum_{j=0}^{M} \exp(g_j(t))} \sum_{m=0}^{M} \exp(g_m(t)) w_m(t) + \epsilon, \tag{4.4}$$

where we choose the component process $w_0(t) = 0$ for all t to allow for silence or rest. The activation $\phi_0(t)$ is equal to 1 only when there is silence in the audio recording.

4.2.1 The Matérn spectral mixture kernel

A single note produced by a music instrument (see Figure 4.2a) consist of an intricate spectrum of harmonics, with an attack and decay in intensity. The spectrum evolution is instrument dependent, and therefore must be learnt in a recording-specific way [16, 75]. This motivates the design of what we call the *Matérn spectral mixture* (MSM) kernel; a stationary covariance function able to reflect the rich spectra of sounds [83]. In this section, we first recall

the spectral representation of stationary kernels. Next, we introduce the formulation of the MSM kernel by an illustrative example. This covariance describes the components $\{w_m(t)\}_{m=1}^{M}$.

Taking as example two basic kernels we use later on, we apply (2.9) on the Matérn-$\frac{1}{2}$ and Cosine kernels [56], defined as

$$k_{1/2}(r) = \sigma^2 e^{-\lambda r}, \qquad \lambda = l^{-1}, \tag{4.5}$$

$$k_{\mathrm{COS}}(r) = \cos(\omega_0 r), \qquad \omega_0 = 2\pi f_0, \tag{4.6}$$

respectively. In (4.5) l governs the time length-scale over which the function varies, and σ^2 defines the scale (amplitude). In (4.6) f_0 defines the function's frequency in Hertz, and the variance is assumed to be one. The corresponding spectral densities are

$$s_{1/2}(\omega) = 2\sigma^2\lambda(\lambda^2 + \omega^2)^{-1}, \tag{4.7}$$

$$s_{\mathrm{COS}}(\omega) = \pi\left[\delta(\omega - \omega_0) + \delta(\omega + \omega_0)\right]. \tag{4.8}$$

We use the spectral representation of covariance functions to formulate the MSM kernel. Figure 4.2a shows the waveform of a single note $\hat{y}_m(t)$, corresponding to playing pitch C4 (261.6 Hz) on an electric guitar. Figure 4.2b depicts the corresponding magnitude Fourier Transform (FT) $|\hat{Y}_m(\omega)|$, which is a real, symmetric function, similar to kernels and its corresponding spectral densities. This leads to the idea of designing kernels whose spectral density is close to the frequency content of the single notes available for training, that is

$$s(\omega) \approx |\hat{Y}_m(\omega)|. \tag{4.9}$$

However, the Matérn-$\frac{1}{2}$ covariance function (4.5) is not appropriate for modelling harmonic content by itself. This is because the spectral density of this kernel has the form of a Lorentzian function (see (4.7)) centred on the origin [37], whereas the spectral density of single notes have peaks at certain

frequencies not necessarily at $\omega = 0$ (see Figure 4.2b). To describe a single partial in Figure 4.2b it is necessary to shift the spectral density of the Matérn-$\frac{1}{2}$, centring it around a specific frequency. To do so, we multiply (4.5) by (4.6), ending up with the base kernel $k(r) = k_{1/2}(r) \cdot k_{\text{COS}}(r)$. Replacing $k(r)$ in (2.9), and using the convolution theorem, then

$$s(\omega) \quad = L(\omega; \boldsymbol{\theta}) + L(-\omega; \boldsymbol{\theta}), \tag{4.10}$$

$$L(\omega; \boldsymbol{\theta}) = \frac{2\pi\sigma^2\lambda}{\lambda^2 + (\omega - \omega_0)^2}, \tag{4.11}$$

with $\boldsymbol{\theta} = \{\sigma^2, \lambda, \omega_0\}$, i.e. the set of hyperparameters associated with (4.5) and (4.6). Expression (4.11) corresponds to shift, from the origin to ω_0, the Matérn-$\frac{1}{2}$ spectral density (4.7). To model D number of partials we use a linear combination of Lorentzian functions pairs

$$s_{\text{MSM}}(\omega; \boldsymbol{\Theta}) = \sum_{j=1}^{D} L(\omega; \boldsymbol{\theta}_j) + L(-\omega; \boldsymbol{\theta}_j), \tag{4.12}$$

where $\boldsymbol{\Theta} = \{\boldsymbol{\theta}_j\}_{j=1}^{D}$. Recall we intend to make as close as possible the spectral density of the kernel to the spectral density of the training data. Therefore the aim of the learning stage is to find the $\boldsymbol{\Theta}$ that makes $s_{\text{MSM}}(\omega)$ close to $|\hat{Y}_m(\omega)|$, that is

$$\Theta^* = \underset{\Theta}{\text{argmin}} \sqrt{\Big(s_{\text{MSM}}(\omega; \boldsymbol{\Theta}) - |\hat{Y}_m(\omega)|\Big)^2}. \tag{4.13}$$

An algorithm for optimizing (4.13) is proposed in section 4.2.2. Finally, replacing (4.12) in (2.10) we end up with a kernel the with form

$$k_{\text{MSM}}(r) = \sum_{j=1}^{N_h} \sigma_j^2 e^{-\lambda_j r} \cos(\omega_{0j} r), \tag{4.14}$$

where ω_{0j} is the frequency in radians, σ_j^2 explains the contribution of each frequency to the overall kernel, and $\lambda_j = l_j^{-1}$, where l_j is the length-scale. The MSM kernel (4.14) can be seen as a spectral-mixture kernel [83], where

instead of using the squared exponential (SE) covariance we use the Matérn-$\frac{1}{2}$. Although the SE kernel (a covariance function infinitely differentiable) is probably the most widely-used kernel [56], in [69] Stein argues that such strong smoothness assumptions are unrealistic for modelling many physical processes, and recommends the Matérn class. Moreover, we have particular interest in using the family of Matérn kernels with half-integer orders, to explore as future work the Variational Fourier Features (VFF) presented in [33] for efficient GP models. Finally, by encouraging (4.12) to reflect the frequency content of isolated sounds, we keep the MSM kernel within a region where it has musically-acoustically interpretation.

4.2.2 Inference

Learning the hyper-parameters of GP models by maximising the marginal likelihood is challenging. This is because the computational complexity of inference usually scale cubically with the number of data observations [33, 56]. To overcome this, we introduce an algorithm for optimizing (4.13). We take advantage of the sparse frequency content of the magnitude FT of the isolated events available for training (for a sample see Figure 4.2b). The basic idea is to fit a Lorentzian function (4.11) around each local maximum present in the spectral density, but considering only one peak at time (Algorithm 1).

Algorithm 1 Fitting MSM kernel in the frequency domain.

Input: $|\hat{Y}_m(\omega)|^2$, D
Output: $\boldsymbol{\Theta} = \{\boldsymbol{\theta}_i\}_{i=1}^{D}$
1: $H(\omega) = |\hat{Y}_m(\omega)|^2$
2: **for** $i := 1$ **to** D **do**
3: $\quad \omega^* = \text{argmax}_{\omega}\ H(\omega)$
4: $\quad$ *Initialize* $\boldsymbol{\theta} = \{\sigma^2, \lambda, \omega_0 = \omega^*\}$
5: $\quad \boldsymbol{\theta}_i = \text{argmin}_{\boldsymbol{\theta}} \sqrt{[L(\omega; \boldsymbol{\theta}) - H(\omega)]^2}$
6: $\quad H(\omega) = |H(\omega) - L(\omega; \boldsymbol{\theta}_i)|$
7: **end for**
8: **return** $\boldsymbol{\Theta}$

With this approach learning hyperparameters takes only few seconds,

despite using all 32×10^3 data points available for training for of each isolated note audio file (16 kHz sample frequency, 2 seconds duration). Figure 4.2c(top) shows the spectral density of initializing the MSM kernel with perfect harmonics and equal variance (dashed red line) against the FT of the actual training data (continuous blue line). Figure 4.2c(middle) shows the FT of the learnt MSM kernel using marginal likelihood (red line). The frequency content of the learnt covariance using the proposed approach is depicted in Figure 4.2c(bottom). One advantage of the MSM kernel is that it is not limited to perfect harmonics. This facilitates better fit to the audio data frequency content, which in this specific case have quasi-harmonic behaviour.

4.3 Experiments

This section presents the empirical evaluation of how (4.1) works in practice for pitch detection. The sigmoid (SIG) (4.2) and softmax (SOF) (4.4) models were used for inferring the occurrence of two different pitches in synthetic audio of an electric guitar. In order to extend the model to more than two pitches we study the scenario where one single component $w_m(t)$ reflects the frequency content of several sound events with different pitches. We studied how the learned kernels affected the performance of the model on the pitch detection task. To do so, we compared: tuning manually (TM) the hyperparameters of the kernel, learning the hyperparameters in the frequency domain (FL) (proposed method), and learning the hyperparameters by optimizing the marginal likelihood (ML). We use the Sparse Variational GP regression implemented in *GPflow* [45] for running the experiments. We analysed the electric guitar audio from the study done in [87], containing the sound events (C4, E4, G4, C4+E4, C4+G4, E4+G4, and C4+E4+G4). This signal was generated with 16 kHz sample frequency, and last 14 seconds. For training we used the first three isolated notes.

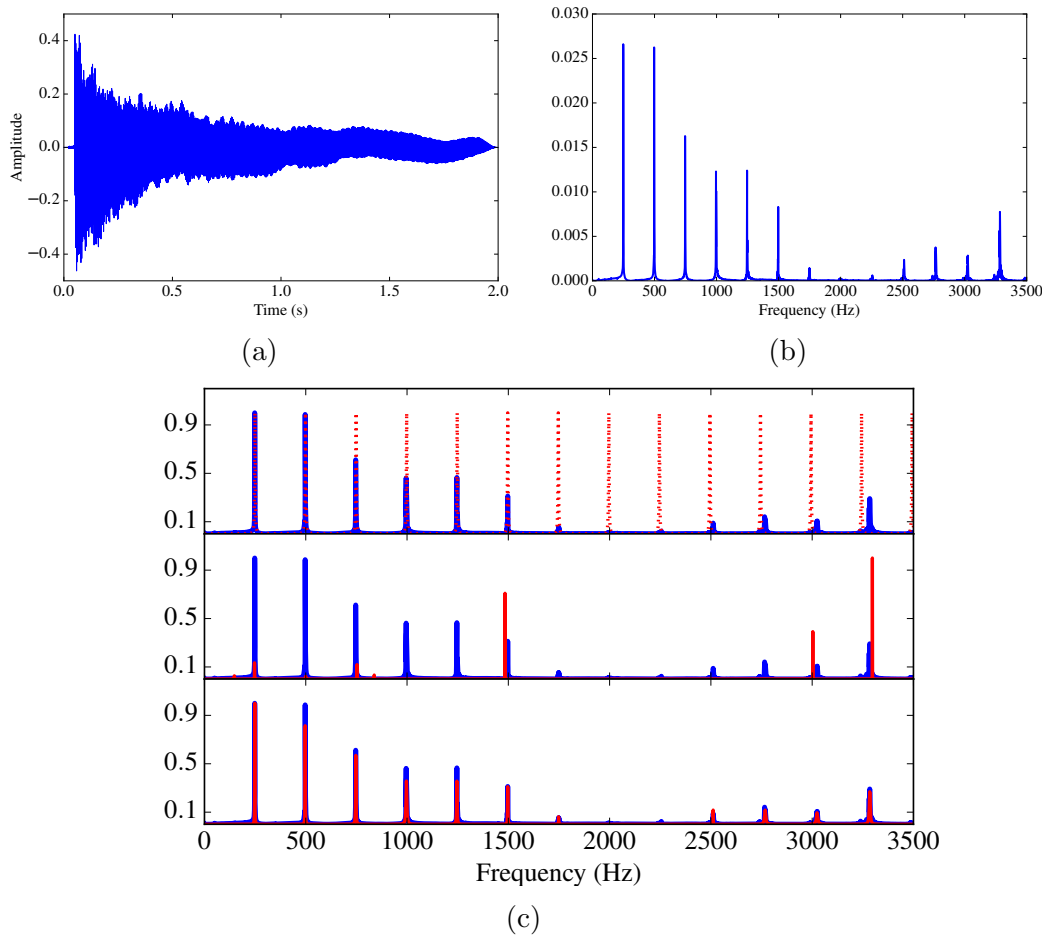


Figure 4.2: (a) sample of a training waveform $Y_m(\omega)$. (b) corresponding magnitude FT $|\hat{Y}_m(\omega)|$. Spectral density of learnt kernel using (c-top) TM (red dashed line), (c-middle) ML (red), (c-bottom) FL (red).

4.3.1 Transcription of polyphonic signal

First we focus on detecting pitches C4 and E4, i.e. from the complete audio signal we only analysed the segments from 0 to 4 seconds and from 6 to 8 seconds. Table 4.1 shows the F-measure obtained using either the sigmoid (SIG) model (4.2) or the softmax (SOF) model (4.4). We compare how the inference approach used affects the performance of these two models. We observe that slightly better performance is achieved by using the sigmoid model. The learning approach considerably affects the performance of the models. The best pitch detection (98.68% F-measure) was achieved using

Model	Inference method	F-measure
SIG	TM	89.54 %
	ML	59.23 %
	FL	**98.68** %
SOF	TM	86.28 %
	ML	55.28 %
	FL	97.15 %
SIG-LOO	TM	76.21 %
	ML	84.86 %
	FL	**98.19** %

Table 4.1: F-measure for the sigmoid model (SIG) and softmax model (SOF) when detecting two pitches. F-measure for the sigmoid-leave-one-out (SIG-LOO) model when detecting three pitches. Three inference methods were compared: tuned manually (TM), marginal likelihood (ML), and frequency learning (FL) (proposed).

SIG model and learning in frequency domain (FL).

In order to extend the model to detect more than two pitches, we allow one of the components to reflect the frequency content of two isolated notes with different pitches, per example: $s_1(\omega) \approx |\hat{Y}_{\mathrm{C4}}(\omega)|$, whereas $s_2(\omega) \approx |\hat{Y}_{\mathrm{E4}}(\omega)| + |\hat{Y}_{\mathrm{G4}}(\omega)|$. We call this approach *leave one out* (SIG-LOO) as one of the spectral densities of the covariances reflects only one pitch, whereas the other the remaining pitches. Figure 4.3a shows the corresponding ground truth piano-roll. Transcriptions using frequency learning, marginal likelihood optimization, and initial guess are shown in Figure 4.3d, 4.3c, 4.3b respectively. Results show SIG-LOO model together with the proposed learning in frequency domain outperforms for pitch detection (98.19% F-measure Table 4.1).

4.4 Conclusions

We proposed a GP regression approach for pitch detection in polyphonic signals. We introduced the Matérn mixture kernel into the model, this allows to reflect the intricate frequency content of sounds of single notes, together

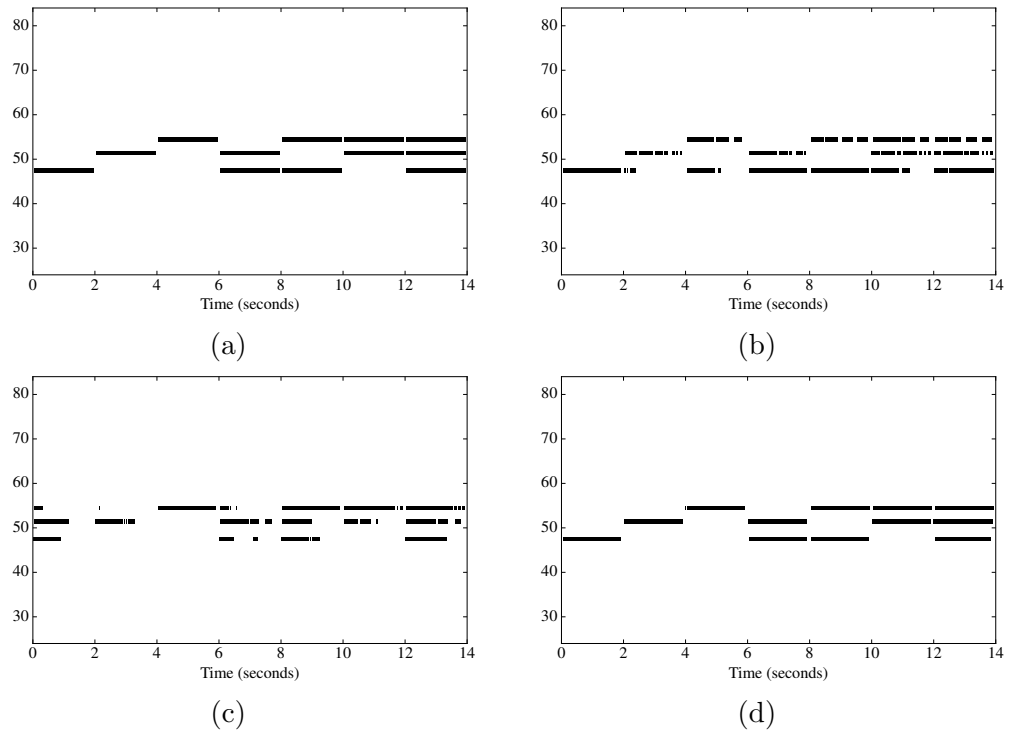


Figure 4.3: Transcription using LOO-SIG. (a) ground truth. (b-d) transcription using TM, ML, FL learning approaches respectively.

with an algorithm for learning its parameters in frequency domain. The proposed approach allows to introduce prior information about activations, such as smoothness (not infinite), non-negativity, and dependency between activations. Results suggest that what it is really relevant for pitch detection is a set of MSM kernels that properly fit the frequency content of the sound events to detect. We conclude that using the proposed hyperparameter learning in the frequency domain, together with the sigmoid model, outperforms the other compared approaches in pitch detection. To our surprise, even if the sigmoid models lacks to encourage dependency between activations as the softmax model does. In addition, one advantage of using the LOO is its linear scalability regarding the number of pitches. Further empirical validation is necessary to validate its performance for more than 3 pitches. As future work we plan to explore other Matérn kernels and VFF in order to be able to analyse a complete piece of music.

Chapter 5

Variational sparse Gaussian process audio source separation and multi-pitch detection

So far in this thesis, we have used Gaussian process models for detecting multiple pitches in polyphonic music signals (Chapter 3, and Chapter 4). Also, we have predicted gaps of missing data in mixture audio recordings (Chapter 3). Now, we turn our attention to source separation (see Section 2.3). That is, the task of estimating a certain number of latent functions called *sources* from a *mixture* signal (observed data) [41]. This chapter is divided in two main sections. In Section 5.1 the *sources* are modelled as GPs, and the *mixture* signal is assumed to be a linear combination of the sources. In addition, the hyperparameters are learned by windowing the data and maximizing a variational lower bound for each window.

Later, in Section 5.2, we reintroduce the modulated-GP model, described in Chapter 4. Recall that in the modulated-GP each latent function, i.e. each source in the context of this chapter, is modelled as the product of two GPs, one controlling the amplitude-envelope, and the other describing the frequency content of the source. As in Section 5.1, the mixture data is also assumed to be a linear combination of the sources. The main difference is that the hyperparameters are learned by maximizing an evidence lower

bound using stochastic variational inference (SVI). This allows the usage of all the available data (by sampling mini-batches) when optimizing the objective function, suppressing the need to divide the data into independent windows.

5.1 Gaussian process source separation

Gaussian process (GP) audio source separation is a time-domain approach that circumvents the inherent phase approximation issue of spectrogram based methods. Furthermore, through its kernel, GPs elegantly incorporate prior knowledge about the sources into the separation model. Despite these compelling advantages, the computational complexity of GP inference scales cubically with the number of audio samples. As a result, source separation GP models have been restricted to the analysis of short audio frames. We introduce an efficient application of GPs to time-domain audio source separation, without compromising performance. For this purpose, we used GP regression, together with spectral mixture kernels, and variational sparse GPs. We compared our method with LD-PSDTF (positive semi-definite tensor factorization), KL-NMF (Kullback-Leibler non-negative matrix factorization), and IS-NMF (Itakura-Saito NMF). Results show that the proposed method outperforms these techniques.

Single-channel audio source separation is a central problem in signal processing research. Here, the task is to estimate a certain number of latent signals or *sources* that were mixed together in one recorded *mixture* signal [41]. State of the art time-frequency methods for source separation include deep neural networks [70], non-negative matrix factorisation (NMF) [39], and probabilistic latent component analysis (PLCA) [68]. These approaches decompose the power spectrogram of the mixture into elementary components. Then, the components are used to calculate the individual source-spectrograms. Time-frequency methods often arbitrarily discard phase information. As a result, the phase of each source-spectrogram must be approximated, corrupting the reconstructed sources.

In contrast, time-domain source separation approaches can avoid the

phase approximation issue of time-frequency methods [29, 71]. For example, Yoshii et al. [87] reconstructed source signals from the mixture waveform directly in the time domain. To this end, Gaussian processes (GPs) were used to predict each source waveform. A particularly influential work in time domain approaches is Liutkus et al. [41], who first formulated source separation as a GP regression task. One clear advantage of this formulation was that prior knowledge about the properties of the sources could be elegantly integrated into the model. This was done by choosing suitable covariance functions.

Although source separation Gaussian process (SSGP) models circumvent phase approximation, the computational complexity of GP inference scales cubically with the number of audio samples (see section 2.6). Hence, different approximate techniques have been proposed to make the separation tractable. For instance, various authors partitioned the mixture signal into independent frames [41, 87]. Further, approximate inference in the frequency domain was used to learn model hyperparameters [41]. Alternatively, Adam et al. [3] recently proposed to use variational sparse GPs for source separation, however audio signals were beyond the scope of their study.

Although the kernel selection in SSGP models determines the properties of sources, only standard covariance functions have been used so far. For example, Adam et al. [3] considered stationarity, smoothness and periodicity, using *exponentiated quadratic* times *cosine* kernels. *Standard periodic* kernels [43] were applied in [41]. These kernels assume that the source spectrum is composed of a fundamental frequency and perfect harmonics. However, real audio signals have more intricate spectra [17], and so separating audio sources requires more flexible covariance functions. One such covariance, the spectral mixture (SM) kernel [83] (A modification of this kernel was introduced in chapter 4), is intended for intricate spectrum patterns. SM kernels approximate the spectral density of any stationary covariance function, using a Gaussian mixture. Alternatively, non-parametric kernels are implicitly considered when the covariance matrix of each source is directly optimised by maximum likelihood [87]. However, that study did not contemplate variational sparse GPs. To our knowledge, it has not been determined whether

incorporating SM kernels together with variational sparse GPs into source separation models leads to more efficient and accurate audio source reconstructions.

In this dissertation we introduce a method that combines GP regression [56, 41], spectral mixture kernels [83], and variational sparse GPs [76]. We consider the mixture data as noisy observations of a function of time, composed as the sum of a known number of sources. Further, we assume that each source follows a different GP with a distinctive spectral mixture kernel. In addition, we adapt the kernels to reflect prior knowledge about the typical spectral content of each source. Also, we frame the mixture data, and for every frame we maximize a variational lower bound of the true marginal likelihood to learn the hyperparameters that control the amplitude of each source (variances). Finally, to separate the sources, we use the learned priors to calculate the true posterior over each source.

We notate the mixture data vector as $\mathbf{y} = [y_1, \cdots, y_n]^\top$ at time instants $\mathbf{t} = [t_1, \cdots, t_n]^\top$. As mentioned previously, we consider each mixture audio sample y_i as an observation of a mixture function $f(t)$ corrupted by independent Gaussian noise. Further, we assume $f(t)$ as the sum of J independent source functions $\{s_j(t)\}_{j=1}^{J}$. These functions represent the sources to be reconstructed. Each source $s_j(t)$ follows a different GP with zero mean, and a distinctive spectral mixture kernel. That is, $y_i = f(t_i) + \epsilon_i$, where $f(t) = \sum_{j=1}^{J} s_j(t)$, and each

$$s_j(t) \sim \mathcal{GP}\left(\ 0,\ k_j(t, t')\ \right) \quad \text{for } j = 1, 2, \ldots, J. \tag{5.1}$$

Here, the noise follows $\epsilon_i \sim \mathcal{N}(0,\ \nu^2)$, with variance ν^2. The kernel for the j-th source is represented by $k_j(t, t')$ (introduced shortly in section 5.1.1). In addition, it is a well known property that the sum of GPs is also a Gaussian process [56]. Therefore, the mixture function follows

$$f(t) \sim \mathcal{GP}\left(\ 0,\ \sum_{j=1}^{J} k_j(t, t')\ \right), \tag{5.2}$$

where its kernel is the sum of source kernels, i.e. $k_f(t,t') = \sum_{j=1}^{J} k_j(t,t')$. We focus only on predicting the mixture function (5.2) as well as the sources (5.1) evaluated at $\mathbf{t}$.

Following the standard Gaussian process regression approach introduced earlier in section 2.4.4, the prior over the mixture function, and each source evaluated at $\mathbf{t}$, correspond to $\mathbf{f} \sim \mathcal{N}(\mathbf{0},\ \mathbf{K}_f)$, and $\mathbf{s}_j \sim \mathcal{N}\left(\mathbf{0},\ \mathbf{K}_{s_j}\right)$ respectively, where $\mathbf{f} = [f(t_1), \ldots, f(t_n)]^\top$, $\mathbf{s}_j = [s_j(t_1), \ldots, s_j(t_n)]^\top$, and the covariance matrix $\mathbf{K}_f = \sum_{j=1}^{J} \mathbf{K}_{s_j}$. Each matrix in $\left\{\mathbf{K}_{s_j}\right\}_{j=1}^{J}$ is computed by evaluating its corresponding source kernel $k_j(t,t')$ at all pairs of time instants contained in $\mathbf{t}$. Also, when a Gaussian likelihood is assumed, the priors are conjugate to the likelihood [56]. Hence, the posterior distributions are also Gaussian. That is,

$$p(\mathbf{y} \mid \mathbf{f}) = \prod_{i=1}^{n} \mathcal{N}\left(y_i \mid f_i,\ \nu^2\right), \tag{5.3}$$

$$p(\mathbf{f} \mid \mathbf{y}) = \mathcal{N}\left(\mathbf{f} \mid \mathbf{K}_f^\top \mathbf{H}^{-1}\mathbf{y},\ \ \hat{\mathbf{K}}_f\right), \tag{5.4}$$

$$p(\mathbf{s}_j \mid \mathbf{y}) = \mathcal{N}\left(\mathbf{s}_i \mid \mathbf{K}_{s_j}^\top \mathbf{H}^{-1}\mathbf{y},\ \ \hat{\mathbf{K}}_{s_j}\right). \tag{5.5}$$

Here, the likelihood (5.3) factorizes across the mixture data, and the posterior over the mixture function (5.4) has covariance matrix $\hat{\mathbf{K}}_f = \mathbf{K}_f - \mathbf{K}_f^\top \mathbf{H}^{-1}\mathbf{K}_f$. Also, the posterior distribution over the i-th source (5.5) has covariance matrix $\hat{\mathbf{K}}_{s_j} = \mathbf{K}_{s_j} - \mathbf{K}_{s_j}^\top \mathbf{H}^{-1}\mathbf{K}_{s_j}$, where the matrix $\mathbf{H} = \mathbf{K}_f + \nu^2\mathbf{I}$, and $\mathbf{I}$ is the identity matrix. Further, the model hyperparameters are usually learned by maximizing the log-marginal likelihood

$$\log p(\mathbf{y}) = -\frac{1}{2}\left[\mathbf{y}^\top \mathbf{H}^{-1}\mathbf{y} + \log|\mathbf{H}| + n\log 2\pi\right], \tag{5.6}$$

where $\mathbf{H}$ needs to be inverted.

Although the source separation GP model introduced so far is elegant, its application to large audio signals becomes intractable. This is because the computational complexity of GP inference scales cubically with the number of audio samples. Specifically, learning the hyperparameters by maximizing the true marginal likelihood (5.6) is computationally demanding, as it requires

the inversion of a $n \times n$ matrix. To overcome the limitations imposed by matrix inversion, we instead maximized a variational lower bound of the true marginal likelihood (5.6) (introduced shortly in section 5.1.2). In addition, we divided the mixture data into overlapping frames of size $\hat{n} \ll n$. Finally, to reconstruct the sources, we used the hyperparameters learned for each frame to calculate the true posterior distribution over the sources (eq. (5.5)). The rest of this section is structured as follows. Section 5.1.1 introduces the spectral mixture kernel used for each source. Then, section 5.1.2 presents the lower bound of the true marginal likelihood we maximized for learning the hyperparameters.

5.1.1 Spectral mixture kernels for isolated sources

The kernel $k_j(t, t')$ in (5.1) determines the properties of each source $s_j(t)$, that is, smoothness, stationarity, and more importantly, its spectrum. To model the typical spectral content of each isolated source, we used spectral mixture kernels [83]. These kernels approximate the spectral density of any stationary covariance function using a Gaussian mixture. Further, in chapter 4 we assumed a Lorentzian mixture instead, resulting in the Matérn-1/2 spectral mixture (MSM) kernel

$$k_j(\tau) = \sigma_j^2 \exp\left(-\frac{\tau}{\ell_j}\right) \times \sum_{d=1}^{D} \alpha_{jd}^2 \cos(\omega_{jd}\, \tau), \tag{5.7}$$

where $\tau = |t - t'|$, the set of parameters $\left\{\alpha_{jd}^2,\ \omega_{jd}\right\}_{d=1}^{D}$ controls the energy distribution throughout all the harmonics/partials of the j-th source spectrum. In addition, the variance σ_j^2 controls the source amplitude, whereas the lengthscale ℓ_j determines how fast $s_j(t)$ evolves in time. We grouped all the kernel parameters in the set $\boldsymbol{\theta}_j = \left\{\sigma_j^2,\ \ell_j,\ \left\{\alpha_{jd}^2,\ \omega_{jd}\right\}_{d=1}^{D}\right\}$. We fitted a MSM kernel (5.7) to the spectrum of every source. For this purpose, we used training data consisting of one audio recording of each isolated source. We denoted the training data as a set of vectors $\left\{\mathbf{g}^{(j)}\right\}_{j=1}^{J}$, where each $\mathbf{g}^{(j)} \in \mathbb{R}^{\tilde{n}}$ is the training data vector for the j-th source. Here, the

vector $\mathbf{g}^{(j)} = [g^{(j)}(x_1), \cdots, g^{(j)}(x_{\hat{n}})]^\top$, and $\mathbf{x} = [x_1, \cdots, x_{\hat{n}}]^\top$ is the corresponding time vector. In addition, because only one single realization $\mathbf{g}^{(j)}$ was available for each source in $\{s_j(t)\}_{j=1}^J$, we assumed the sources to be covariance-ergodic processes with zero mean [10, 65, 31]. Therefore, their covariances $\{C_j(\hat{\tau})\}_{j=1}^J$ were estimated as the time average

$$C_j(\hat{\tau}) = \frac{1}{T}\int_0^T g^{(j)}(x+\hat{\tau})\; g^{(j)}(x)\; \mathrm{d}x. \tag{5.8}$$

Here, T denotes the size (in seconds) of the window used to compute the correlation. We used the discrete version of eq. (5.8). Finally, for every source we then minimized the mean square error (MSE) between the covariance estimator (5.8) and the corresponding MSM kernel (5.7). That is,

$$L(\boldsymbol{\theta}_j) = \frac{1}{N_c}\sum_{i=1}^{N_c}\left[k_j(\hat{\tau}_i) - C_j(\hat{\tau}_i)\right]^2, \tag{5.9}$$

where N_c is the number of points where (5.8) was approximated, and $\boldsymbol{\theta}_j$ is the set of kernel parameters in (5.7).

5.1.2 Inference

To reduce the computational time required for learning the hyperparameters by maximizing the true marginal likelihood (5.6), we divided the mixture data $\{t_i, y_i\}_{i=1}^n$ into W overlapping frames of size $\hat{n} \ll n$. Therefore, the set of data frames corresponded to $\left\{\hat{\mathbf{t}}^{(w)}, \hat{\mathbf{y}}^{(w)}\right\}_{w=1}^W$, where $\hat{\mathbf{t}}^{(w)}, \hat{\mathbf{y}}^{(w)} \in \mathbb{R}^{\hat{n}}$. In addition, for each mixture frame $\hat{\mathbf{y}}^{(w)}$, we maximized the evidence lower bound of the true log marginal likelihood (5.6) proposed in [76] for sparse GPs variational inference. For a detailed description of this lower bound see section 2.6. Recalling the form of this objective function:

$$\mathcal{L} \triangleq \log \mathcal{N}\left(\hat{\mathbf{y}}^{(w)} |\; \mathbf{0},\; \mathbf{Q}_{\hat{n}\hat{n}} + \nu^2\mathbf{I}\right) - \frac{1}{2\nu^2}\mathrm{tr}\left(\mathbf{K}_{\hat{n}\hat{n}} - \mathbf{Q}_{\hat{n}\hat{n}}\right), \tag{5.10}$$

where $\mathbf{Q}_{\hat{n}\hat{n}} = \mathbf{K}_{\hat{n}m}\mathbf{K}_{mm}^{-1}\mathbf{K}_{m\hat{n}}$ [76]. The value of the cross-covariance matrix at the i-th row and j-th column, corresponds to $\mathbf{K}_{\hat{n}m}[i,j] = k_f(t_i^{(w)}, z_j)$. Sim-

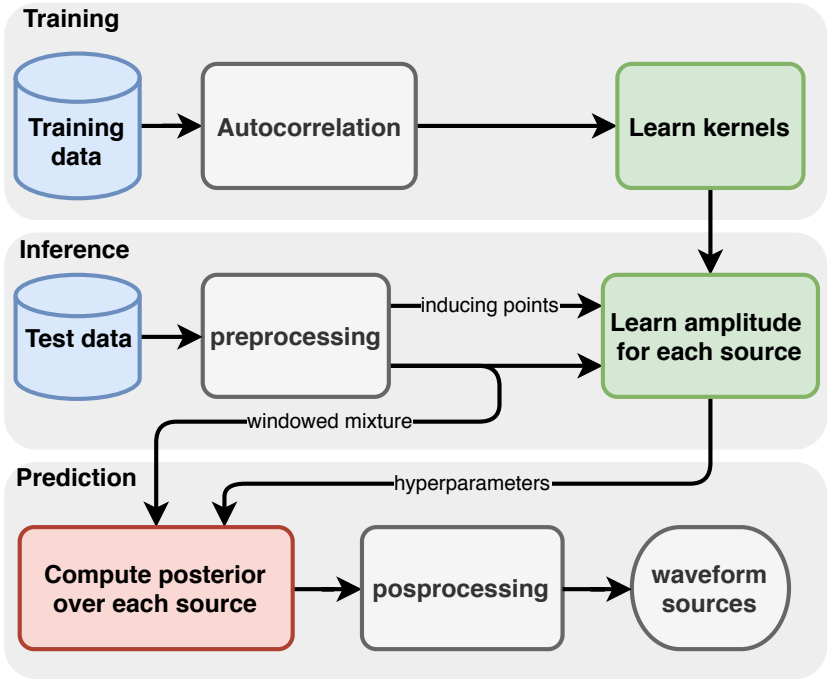


Figure 5.1: Flowchart proposes model.

ilarly, $\mathbf{K}_{mm}[i,j] = k_f(z_i, z_j)$. Recall that $k_f(t,t')$ is the kernel of the mixture function (5.2). In brief, by framing the data and maximizing the lower bound (5.10), the computational time required for learning hyperparameters in each frame or window is reduced to $\mathcal{O}(\hat{n}m^2)$. The proposed method is illustrated in Figure 5.1. In the following experiments the inducing points $\mathbf{z}$ were not learned from maximizing the lower bound (5.10). We instead used two separate criteria to select the inducing points $\mathbf{z}$. Either the inducing points were located at the extrema of the mixture data, that is, the peaks and valleys of the audio signal (see Figure 5.2), or the inducing points were equal to the time vector (full GP).

5.1.3 Experimental evaluation

We tested the proposed SSGP method on the same dataset analysed in [87]. That is, three different mixture audio signals sampled at 16kHz, corresponding to piano, electric guitar, and clarinet. Each mixture lasts 14 seconds,

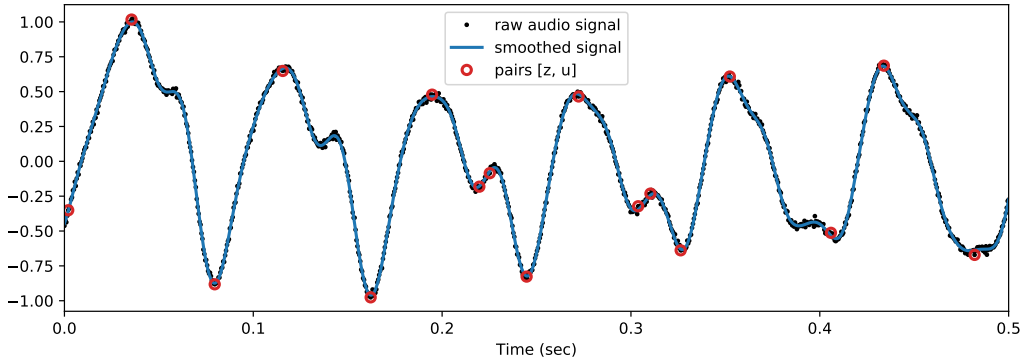


Figure 5.2: Example of selecting the inducing points $\mathbf{z}$ by using the extrema of the audio data.

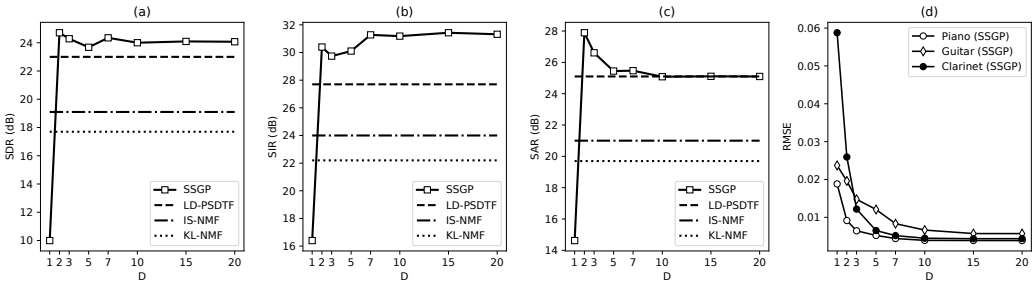


Figure 5.3: Source separation metrics. SDR (a), SIR (b), SAR (c), RMSE (d).

and consists of the following sequence of music notes (C4, E4, G4, C4+E4, C4+G4, E4+G4, and C4+E4+G4). Thus, for each mixture, the aim was to reconstruct three source signals, each with a corresponding note, C4, E4, and G4. The metrics used to measure the separation performance were: source to distortion ratio (SDR), source to interferences ratio (SIR), source to artefacts ratio (SAR) [79], and root mean square error (RMSE). We compared with LD-PSDTF (positive semi-definite tensor factorization), KL-NMF (Kullback-Leibler NMF), and IS-NMF (Itakura-Saito NMF) with rank three [87]. The code was implemented using GPflow [46].

We determined the performance of the proposed method in mixtures of three sources. That is, $J = 3$ in eq. (5.2). To this end, we first divided the mixtures into frames of 125 milliseconds ($\hat{n} = 2001$) with 50% overlap, and initialized the kernel for each source (eq. (5.7) with $D = 15$), by min-

Method	SDR	SIR	SAR	Opt. time
KL-NMF	17.7	22.2	19.7	–
IS-NMF	19.1	24.0	21.0	–
LD-PSDTF	23.0	27.7	25.1	–
SSGP (proposed)	**24.1**	**31.4**	**25.1**	**5.33**
SSGP-full	22.9	22.3	24.6	284.2

Table 5.1: Separation metrics (dB). Optimization time (min).

imizing eq. (5.9). Then, for each mixture frame, we maximized eq. (5.10) to learn the variance of each source, i.e., $\left\{\sigma_j^2\right\}_{j=1}^{J}$. We compared the time required for learning the hyperparameters in these two scenarios. Finally, we used eq. (5.5), and the learned hyperparameters to calculate the true posterior over each source $p\left(\mathbf{s}_i^{(w)}|\mathbf{y}^{(w)}\right)$. We recovered the sources applying the *overlap-add* method to the frame-wise predictions [5]. We found that our method (SSGP) presented the highest SDR and SIR metrics (Figure 5.3), and reduced the optimization time by 98.12% compared to the full GP (Table 5.1), indicating that our method is efficient, robust to interferences between sources (highest SIR), and it introduces less distortion (highest SDR). Further, we observed that the kernels learned for each source presented distinctive spectral patterns (Fig 5.4), which demonstrates that SM kernels are appropriate for learning the rich frequency content found in audio sources. Moreover, we observed that the proposed approach reconstructed accurately the sources (Fig 5.6), showing the variances learned by maximizing the lower bound were consistent with the true sources. In addition, to establish the effect of kernel selection on the separation performance, we carried out the same previous experiment, but changing the number of components D in the kernel eq. (5.7). We found that SDR, SIR and SAR metrics stabilized when $D > 3$ (Figure 5.3(a-c)), indicating that the proposed model is less affected by kernel selection when more than three components are used. Further, RMSE decreased exponentially with D (Figure 5.3(d)), suggesting that increasing the number of components in the kernel leads to more accurate waveform reconstructions.

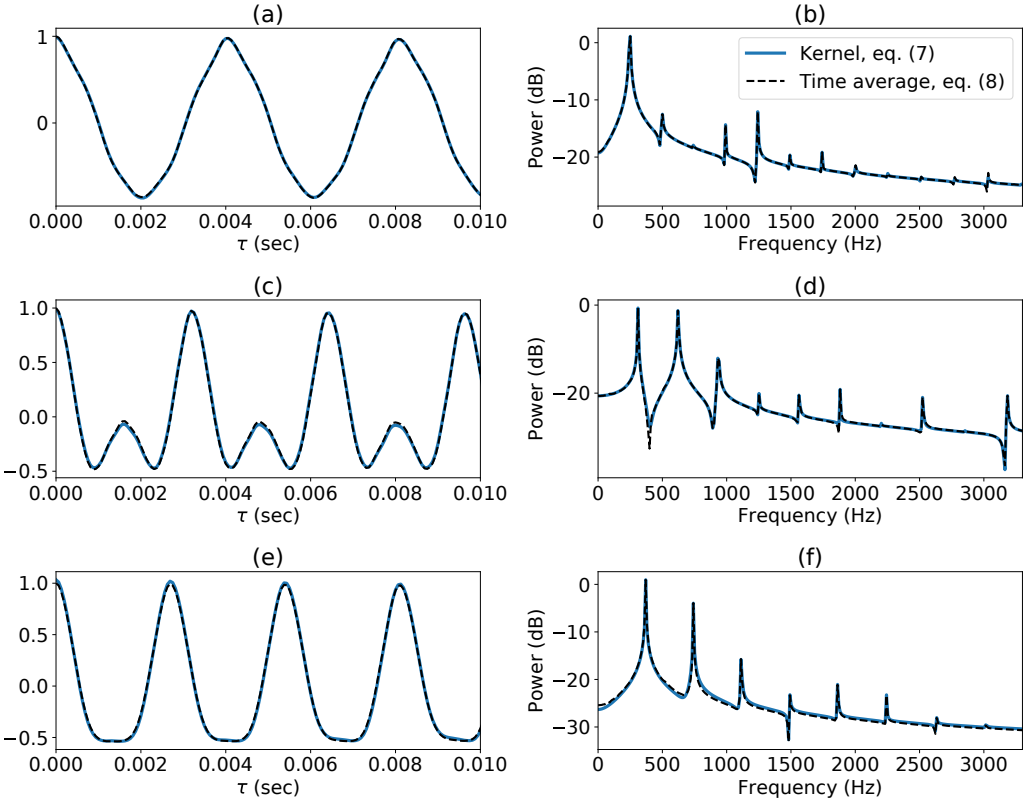


Figure 5.4: Kernels learned for each piano source (left column). Corresponding log-spectral density (right column).

Discussion

Our findings indicate that combining variational sparse GPs together with SM kernels enables time-domain source separation GP models to reconstruct audio sources in an efficient and informed manner, without compromising performance. Also, RMSE results imply that suitable spectrum priors over the sources are essential to improve source reconstruction. Moreover, SDR, SIR, and SAR results suggest the proposed method can be used for other applications such as multipitch-detection, where low interference between sources (SIR) is more relevant than reconstruction artefacts (SAR). We proposed an alternative method that circumvents phase approximation by addressing audio source separation from a variational time-domain perspective. The code is available at [1].

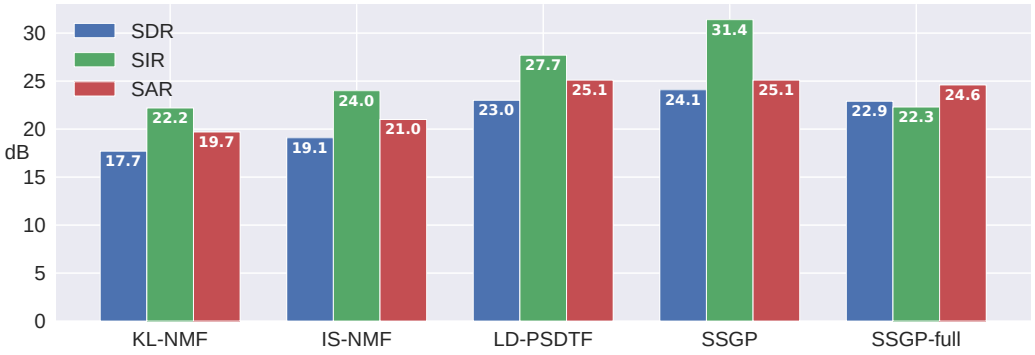


Figure 5.5: Source separation performance.

5.2 GP-SVI for source separation and multi-pitch detection: a joint approach

So far in this thesis, to handle the computational time of inference in GP models we have either used exact inference on a small data set (Chapter 3), or framed the input audio signal and learned the hyperparameters from each window independently, by maximizing a variational evidence lower bound for sparse GPs (Chapter 4, and Section 5.1). Framing the data has enabled GP models to analyse longer audio signals. However, windowing audio signals presents some disadvantages. First, the GP model lacks the ability to learn from the data in adjacent windows/frames. That is, information useful for inference is not taken into account. Second, the predicted activations, components, and sources present discontinuities between adjacent windows. This can be solved by using overlapping windows (Section 5.1). However, this means more windows to analyse, increasing the computational time of inference. Stochastic variational inference (SVI) (see Section 2.6.5) offers an alternative method that does not need to window the input audio signal [35]. This means that in the overall process of inference the optimisation algorithm has access to the whole data. Moreover, discontinuities are avoided, since a single model is used when doing predictions.

In this section we combine SVI and the modulated-GP model (4.1), introduced in Chapter 4 for multi-pitch detection. Moreover, the SVI-GP model formulated in this section is used for two tasks simultaneously, namely source

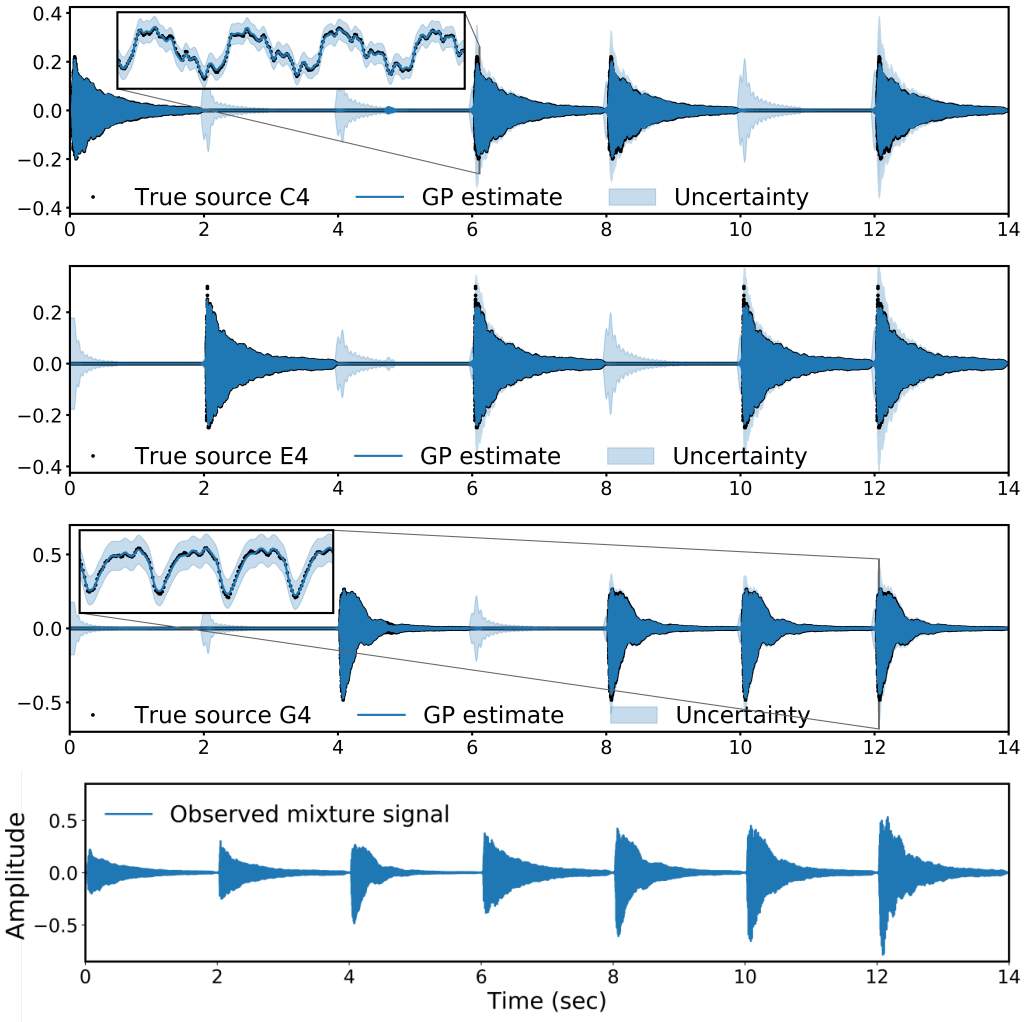


Figure 5.6: Source reconstruction on piano mixture signal.

separation and multi-pitch detection. That is, the modulated-GP model returns one prediction in the form of a waveform (time-domain function) per each corresponding detected pitch (see Figure 5.11). We show preliminary results on two experiments; separating three different sources (as in Section 5.1), and detecting 88 pitches on a piano signal from the MAPS dataset [28].

5.2.1 An ELBO for the modulated-GP model

From the description of stochastic variational inference presented in Section 2.6.5, we know that in order to use SVI in sparse GPs, it is necessary to formulate a lower bound that has global variables, and that also decomposes into a sum of n terms, each term associated to a single data point [35]. This subsection presents the formulation of a SVI-suitable evidence lower bound for the modulated-GP model; a model that decomposes an observed audio signal as the multiplication of a non-negative random process and a Gaussian process with a spectral mixture kernel [4]. This generative probabilistic model is intended to explain how an observed polyphonic music signal (mixture of sources or pitches) was generated. We also seek to compute a posterior distribution over the latent functions associated with each source/pitch.

We study the scenario where the mixture signal has only one source. Then, the model is extended to several sources. The formulation of the ELBO is organized as follows. First, we introduce the model for one single source. Precisely, we define the likelihood, prior, joint distribution, and the limitation of computing the posterior distribution. This motivates the use of approximate inference. Second, we introduce the inducing variables as in [36], but within the context of the modulated-GP. Third, the corresponding variational lower bound is introduced. Subsequently, we analyse in more detail the variational expectation of the log-likelihood and derive two equivalent solutions. The first approximates a double integral by a two dimensional Gauss-Hermite quadrature, whereas the second solution approximates the double integral as the sum of several one-dimensional Gauss-Hermite quadratures. For related work refer to [4, 34, 36, 63].

Single source model

Given an audio recording $\mathcal{D} = \{y_n, t_n\}_{n=1}^{N}$ and the regression model

$$y(t) = \sigma(g(t))f(t) + \epsilon(t),$$

where $f(t)$ and $g(t)$ follow GPs and $\epsilon(t)$ follows a white noise process, then

$$y_n = \sigma(g_n) f_n + \epsilon_n,$$

where $g_n = g(t_n)$, $f_n = f(t_n)$, and $\epsilon_n \sim \mathcal{N}(\epsilon_n|0, \nu^2)$. Defining the vectors $\mathbf{y} = [y_1, \cdots, y_N]^\top$, $\mathbf{f} = [f_1, \cdots, f_N]^\top$, $\mathbf{g} = [g_1, \cdots, g_N]^\top$, and assuming the observations as i.i.d then the **likelihood** corresponds to

$$\begin{aligned} p(\mathbf{y}|\mathbf{f}, \mathbf{g}) &= \prod_{n=1}^{N} p(y_n|f_n, g_n) \\ &= \prod_{n=1}^{N} \mathcal{N}(y_n|\sigma(g_n) f_n, \nu^2). \end{aligned} \tag{5.11}$$

We put an independent GP over each function $f(t)$ and $g(t)$, therefore the **prior** over the latent vectors $\mathbf{f}$ and $\mathbf{g}$ corresponds to $p(\mathbf{f}, \mathbf{g}) = p(\mathbf{f})p(\mathbf{g})$, where

$$p(\mathbf{f}) = \mathcal{N}(\mathbf{f}|\mathbf{0}, \mathbf{K_f}),$$

and

$$p(\mathbf{g}) = \mathcal{N}(\mathbf{g}|\mathbf{0}, \mathbf{K_g}).$$

Given the prior and the likelihood we can define the **joint distribution** as

$$\begin{aligned} p(\mathbf{y}, \mathbf{f}, \mathbf{g}) &= p(\mathbf{y}|\mathbf{f}, \mathbf{g}) p(\mathbf{f}) p(\mathbf{g}) \\ &= \prod_{n=1}^{N} \mathcal{N}(y_n|\sigma(g_n) f_n, \nu^2)\ \mathcal{N}(\mathbf{f}|\mathbf{0}, \mathbf{K_f})\ \mathcal{N}(\mathbf{g}|\mathbf{0}, \mathbf{K_g}). \end{aligned}$$

The **posterior** can be calculated as

$$p(\mathbf{f}, \mathbf{g}|\mathbf{y}) = \frac{p(\mathbf{y}|\mathbf{f}, \mathbf{g}) p(\mathbf{f}) p(\mathbf{g})}{p(\mathbf{y})}, \tag{5.12}$$

where the marginal likelihood is defined as

$$p(\mathbf{y}) = \int \int p(\mathbf{y}|\mathbf{f}, \mathbf{g}) p(\mathbf{f}) p(\mathbf{g}) \, \mathrm{d}\mathbf{f} \, \mathrm{d}\mathbf{g}. \tag{5.13}$$

Computing this expression is usually difficult due to $\mathcal{O}(N^3)$ complexity and non-tractability.

Introducing inducing variables

Introducing inducing points $\mathbf{Z} = \{z_m\}_{m=1}^M$ for both latent functions $f(t)$ and $g(t)$ and their corresponding inducing variables $\mathbf{u}_f = \{f(z_m)\}_{m=1}^M$ and $\mathbf{u}_g = \{g(z_m)\}_{m=1}^M$, then the joint distribution of all latent variables correspond to $p(\mathbf{f}, \mathbf{u}_f) = p(\mathbf{f}|\mathbf{u}_f)p(\mathbf{u}_f)$, and $p(\mathbf{g}, \mathbf{u}_g) = p(\mathbf{g}|\mathbf{u}_g)p(\mathbf{u}_g)$. The joint now follows

$$\begin{aligned} p(\mathbf{y}, \mathbf{f}, \mathbf{g}, \mathbf{u}_f, \mathbf{u}_g) &= p(\mathbf{y}|\mathbf{f}, \mathbf{g}, \mathbf{u}_f, \mathbf{u}_g) p(\mathbf{f}, \mathbf{g}, \mathbf{u}_f, \mathbf{u}_g) \\ &= p(\mathbf{y}|\mathbf{f}, \mathbf{g}) p(\mathbf{f}, \mathbf{u}_f) p(\mathbf{g}, \mathbf{u}_g) \\ &= \underbrace{p(\mathbf{y}|\mathbf{f}, \mathbf{g}) p(\mathbf{f}|\mathbf{u}_f) p(\mathbf{g}|\mathbf{u}_g)}_{p(\mathbf{y}, \mathbf{f}, \mathbf{g}|\mathbf{u}_f, \mathbf{u}_g)} p(\mathbf{u}_f) p(\mathbf{u}_g) \end{aligned}$$

Then

$$\begin{aligned} p(\mathbf{y}|\mathbf{u}_f, \mathbf{u}_g) &= \int \int p(\mathbf{y}, \mathbf{f}, \mathbf{g}|\mathbf{u}_f, \mathbf{u}_g) \mathrm{d}\mathbf{f} \, \mathrm{d}\mathbf{g} \\ &= \int \int p(\mathbf{y}|\mathbf{f}, \mathbf{g}, \mathbf{u}_f, \mathbf{u}_g) p(\mathbf{f}, \mathbf{g}|\mathbf{u}_f, \mathbf{u}_g) \mathrm{d}\mathbf{f} \, \mathrm{d}\mathbf{g} \\ &= \int \int p(\mathbf{y}|\mathbf{f}, \mathbf{g}) p(\mathbf{f}|\mathbf{u}_f) p(\mathbf{g}|\mathbf{u}_g) \mathrm{d}\mathbf{f} \, \mathrm{d}\mathbf{g} \\ &= \mathbb{E}_{p(\mathbf{g}|\mathbf{u}_g)} \left[\mathbb{E}_{p(\mathbf{f}|\mathbf{u}_f)} \left[p(\mathbf{y}|\mathbf{f}, \mathbf{g}) \right] \right] \\ &= \mathbb{E}_{p(\mathbf{f}|\mathbf{u}_f) p(\mathbf{g}|\mathbf{u}_g)} \left[p(\mathbf{y}|\mathbf{f}, \mathbf{g}) \right]. \end{aligned}$$

Similar to [36] we will use the following inequality to get a variational approximation

$$\log p(\mathbf{y}|\mathbf{u}_f, \mathbf{u}_g) \geq \mathbb{E}_{p(\mathbf{f}|\mathbf{u}_f) p(\mathbf{g}|\mathbf{u}_g)} \left[\log p(\mathbf{y}|\mathbf{f}, \mathbf{g}) \right]. \tag{5.14}$$

Variational lower bound

We assume the following variational distribution over all inducing variables

$$\begin{aligned} q(\mathbf{u}) &= q(\mathbf{u}_f, \mathbf{u}_g) \\ &= q(\mathbf{u}_f)q(\mathbf{u}_g) \\ &= \mathcal{N}(\mathbf{m}_f, \mathbf{S}_f)\mathcal{N}(\mathbf{m}_g, \mathbf{S}_g), \end{aligned} \tag{5.15}$$

that is $q(\mathbf{u}_f) = \mathcal{N}(\mathbf{m}_f, \mathbf{S}_f)$ and $q(\mathbf{u}_g) = \mathcal{N}(\mathbf{m}_g, \mathbf{S}_g)$. Using the standard variational equation

$$\begin{aligned} \log(\mathbf{y}) &\geq \mathbb{E}_{q(\mathbf{u})}\left[\log p(\mathbf{y}|\mathbf{u})\right] - \mathrm{KL}\left(q(\mathbf{u})||p(\mathbf{u})\right) \\ &\geq \mathbb{E}_{q(\mathbf{u}_f,\mathbf{u}_g)}\left[\log p(\mathbf{y}|\mathbf{u}_f, \mathbf{u}_g)\right] - \mathrm{KL}\left(q(\mathbf{u}_f, \mathbf{u}_g)||p(\mathbf{u}_f, \mathbf{u}_g)\right) \\ &\geq \mathbb{E}_{q(\mathbf{u}_f)q(\mathbf{u}_g)}\left[\log p(\mathbf{y}|\mathbf{u}_f, \mathbf{u}_g)\right] - \mathrm{KL}\left(q(\mathbf{u}_f)q(\mathbf{u}_g)||p(\mathbf{u}_f)p(\mathbf{u}_g)\right). \end{aligned} \tag{5.16}$$

Replacing (5.14) in (5.16) then

$$\log(\mathbf{y}) \geq \underbrace{\mathbb{E}_{q(\mathbf{u}_f)q(\mathbf{u}_g)}\left[\mathbb{E}_{p(\mathbf{f}|\mathbf{u}_f)p(\mathbf{g}|\mathbf{u}_g)}\left[\log p(\mathbf{y}|\mathbf{f}, \mathbf{g})\right]\right]}_{B} - \underbrace{\mathrm{KL}\left(q(\mathbf{u}_f)q(\mathbf{u}_g)||p(\mathbf{u}_f)p(\mathbf{u}_g)\right)}_{A}. \tag{5.17}$$

Analysing A in (5.17)

Analysing the KL divergence in (5.17) we get

$$\mathrm{KL}\left(q(\mathbf{u}_f)q(\mathbf{u}_g)||p(\mathbf{u}_f)p(\mathbf{u}_g)\right) = \int\int q(\mathbf{u}_f)q(\mathbf{u}_g)\log\left\{\frac{q(\mathbf{u}_f)q(\mathbf{u}_g)}{p(\mathbf{u}_f)p(\mathbf{u}_g)}\right\}\mathrm{d}\mathbf{u}_f\ \mathrm{d}\mathbf{u}_g,$$

that is

$$
\begin{aligned}
&= \int\int q(\mathbf{u}_f)q(\mathbf{u}_g)\left[\log q(\mathbf{u}_f) + \log q(\mathbf{u}_g) - \log p(\mathbf{u}_f) - \log p(\mathbf{u}_g)\right] \mathrm{d}\mathbf{u}_f \; \mathrm{d}\mathbf{u}_g \\
&= \int\int q(\mathbf{u}_f)q(\mathbf{u}_g)\left[\log\left\{\frac{q(\mathbf{u}_f)}{p(\mathbf{u}_f)}\right\} + \log\left\{\frac{q(\mathbf{u}_g)}{p(\mathbf{u}_g)}\right\}\right] \mathrm{d}\mathbf{u}_f \; \mathrm{d}\mathbf{u}_g \\
&= \int\int q(\mathbf{u}_f)q(\mathbf{u}_g)\log\left\{\frac{q(\mathbf{u}_f)}{p(\mathbf{u}_f)}\right\} \mathrm{d}\mathbf{u}_f \; \mathrm{d}\mathbf{u}_g + \int\int q(\mathbf{u}_f)q(\mathbf{u}_g)\log\left\{\frac{q(\mathbf{u}_g)}{p(\mathbf{u}_g)}\right\} \mathrm{d}\mathbf{u}_f \; \mathrm{d}\mathbf{u}_g \\
&= \int q(\mathbf{u}_f)\log\left\{\frac{q(\mathbf{u}_f)}{p(\mathbf{u}_f)}\right\} \mathrm{d}\mathbf{u}_f + \int q(\mathbf{u}_g)\log\left\{\frac{q(\mathbf{u}_g)}{p(\mathbf{u}_g)}\right\} \mathrm{d}\mathbf{u}_g,
\end{aligned}
$$

therefore

$$
\mathrm{KL}\left(q(\mathbf{u}_f)q(\mathbf{u}_g)||p(\mathbf{u}_f)p(\mathbf{u}_g)\right) = \mathrm{KL}\left(q(\mathbf{u}_f)||p(\mathbf{u}_f)\right) + \mathrm{KL}\left(q(\mathbf{u}_g)||p(\mathbf{u}_g)\right). \tag{5.18}
$$

Analysing B in (5.17)

Analysing the expectation we have

$$
\begin{aligned}
&\mathbb{E}_{q(\mathbf{u}_f)q(\mathbf{u}_g)}\left[\mathbb{E}_{p(\mathbf{f}|\mathbf{u}_f)p(\mathbf{g}|\mathbf{u}_g)}\left[\log p(\mathbf{y}|\mathbf{f},\mathbf{g})\right]\right] = \\
&\int\int\int\int \log p(\mathbf{y}|\mathbf{f},\mathbf{g})p(\mathbf{f}|\mathbf{u}_f)p(\mathbf{g}|\mathbf{u}_g)q(\mathbf{u}_f)q(\mathbf{u}_g) \; \mathrm{d}\mathbf{f} \; \mathrm{d}\mathbf{g} \; \mathrm{d}\mathbf{u}_f \; \mathrm{d}\mathbf{u}_g = \\
&\int\int \log p(\mathbf{y}|\mathbf{f},\mathbf{g})\left[\int p(\mathbf{f}|\mathbf{u}_f)q(\mathbf{u}_f) \; \mathrm{d}\mathbf{u}_f\right] \cdot \left[\int p(\mathbf{g}|\mathbf{u}_g)q(\mathbf{u}_g) \; \mathrm{d}\mathbf{u}_g\right] \; \mathrm{d}\mathbf{f} \; \mathrm{d}\mathbf{g} = \\
&\int\int \log p(\mathbf{y}|\mathbf{f},\mathbf{g})q(\mathbf{f})q(\mathbf{g}) \; \mathrm{d}\mathbf{f} \; \mathrm{d}\mathbf{g},
\end{aligned}
$$

therefore

$$
\mathbb{E}_{q(\mathbf{u}_f)q(\mathbf{u}_g)}\left[\mathbb{E}_{p(\mathbf{f}|\mathbf{u}_f)p(\mathbf{g}|\mathbf{u}_g)}\left[\log p(\mathbf{y}|\mathbf{f},\mathbf{g})\right]\right] = \mathbb{E}_{q(\mathbf{f})q(\mathbf{g})}\left[\log p(\mathbf{y}|\mathbf{f},\mathbf{g})\right], \tag{5.19}
$$

where

$$
\begin{aligned}
q(\mathbf{f}) &= \int p(\mathbf{f}|\mathbf{u}_f)q(\mathbf{u}_f)\mathrm{d}\mathbf{u}_f, \\
q(\mathbf{g}) &= \int p(\mathbf{g}|\mathbf{u}_g)q(\mathbf{u}_g)\mathrm{d}\mathbf{u}_g.
\end{aligned}
$$

ELBO

The evidence lower bound (ELBO) is defined by replacing (5.18) and (5.19) into (5.17)

$$\begin{aligned}&\text{ELBO}(q(\mathbf{u}_f), q(\mathbf{u}_g)) = \\ &\mathbb{E}_{q(\mathbf{f})q(\mathbf{g})}\left[\log p(\mathbf{y}|\mathbf{f},\mathbf{g})\right] - \text{KL}\left(q(\mathbf{u}_f)||p(\mathbf{u}_f)\right) - \text{KL}\left(q(\mathbf{u}_g)||p(\mathbf{u}_g)\right).\end{aligned} \tag{5.20}$$

This is the functional we aim to maximise in the variational approach.

Approximating variational expectations using quadrature

Analysing the expectation in the lower bound equation (5.20), and using the definition of the likelihood (5.11) we have

$$\begin{aligned}\mathbb{E}_{q(\mathbf{f})q(\mathbf{g})}\left[\log p(\mathbf{y}|\mathbf{f},\mathbf{g})\right] &= \mathbb{E}_{q(\mathbf{f})q(\mathbf{g})}\left[\log \prod_{n=1}^{N} p(y_n|f_n, g_n)\right] \\ &= \mathbb{E}_{q(\mathbf{f})q(\mathbf{g})}\left[\sum_{n=1}^{N} \log p(y_n|f_n, g_n)\right] \\ &= \sum_{n=1}^{N} \mathbb{E}_{q(\mathbf{f})q(\mathbf{g})}\left[\log p(y_n|f_n, g_n)\right] \\ &= \sum_{n=1}^{N} \int\int \log p(y_n|f_n, g_n) q(\mathbf{f}) q(\mathbf{g}) \, \mathrm{d}\mathbf{f} \, \mathrm{d}\mathbf{g} \\ &= \sum_{n=1}^{N} \int\int \log p(y_n|f_n, g_n) q(f_n) q(g_n) \, \mathrm{d}f_n \, \mathrm{d}g_n,\end{aligned} \tag{5.21}$$

then we end up solving N two dimensional Gauss-Hermite quadratures.

From (5.21) we aim to approximate the following double integral by quadrature methods

$$\int\int \log p(y_n|f_n, g_n) q(f_n) q(g_n) \, \mathrm{d}f_n \, \mathrm{d}g_n. \tag{5.22}$$

In the next two subsections we present two different solutions. The first one solves the double integral in (5.21) by using a two dimensional quadrature,

whereas the second solves (5.21) by using a linear combination of 2 one dimensional quadratures, this might help to reduce computational cost.

Solving (5.22) **by using quadrature of dimension 2**

From the definition of the variational distribution $q(\mathbf{u}_f, \mathbf{u}_g)$ in (5.15) we know that $q(f_n) = \mathcal{N}(f_n|m_{f_n}, s^2_{f_n})$, and $q(g_n) = \mathcal{N}(g_n|m_{g_n}, s^2_{g_n})$ Then, replacing in (5.22) we get

$$\int\int \log p(y_n|f_n, g_n)\mathcal{N}(f_n|m_{f_n}, s^2_{f_n})\mathcal{N}(g_n|m_{g_n}, s^2_{g_n})\ \mathrm{d}f_n\ \mathrm{d}g_n = \tag{5.23}$$
$$\frac{1}{(2\pi s^2_{f_n})^{1/2}}\frac{1}{(2\pi s^2_{g_n})^{1/2}}\int\int \log p(y_n|f_n, g_n) \times \cdots$$
$$\exp\left\{-\frac{1}{2s^2_{f_n}}(f_n - m_{f_n})^2\right\}\exp\left\{-\frac{1}{2s^2_{g_n}}(g_n - m_{g_n})^2\right\}\ \mathrm{d}f_n\ \mathrm{d}g_n,$$

introducing the following change of variable:

$$\hat{f}_n = \frac{f_n - m_{f_n}}{\sqrt{2}s_{f_n}},$$

and

$$\hat{g}_n = \frac{g_n - m_{g_n}}{\sqrt{2}s_{g_n}},$$

then (5.23) can be written as

$$\int\int \log p(y_n|f_n, g_n)\mathcal{N}(f_n|m_{f_n}, s^2_{f_n})\mathcal{N}(g_n|m_{g_n}, s^2_{g_n})\ \mathrm{d}f_n\ \mathrm{d}g_n =$$
$$\frac{1}{\pi}\int\int \log p(y_n|\sqrt{2}s_{f_n}\hat{f}_n + m_{f_n}, \sqrt{2}s_{g_n}\hat{g}_n + m_{g_n})\exp\left\{-\hat{f_n}^2\right\}\exp\left\{-\hat{g_n}^2\right\}\ \mathrm{d}\hat{f}_n\ \mathrm{d}\hat{g}_n,$$

The previous double integral can be approximated as

$$\int\int \log p(y_n|f_n, g_n)q(f_n)q(g_n)\ \mathrm{d}f_n\ \mathrm{d}g_n \approx \tag{5.24}$$
$$\frac{1}{\pi}\sum_{\forall i}\sum_{\forall j} w_i w_j \log p(y_n|\sqrt{2}s_{f_n}\hat{x}_i + m_{f_n}, \sqrt{2}s_{g_n}\hat{y}_j + m_{g_n}),$$

where $w_i, w_j, \hat{x}_i, \hat{y}_i$ are obtained from the formula for the Gauss-Hermite quadrature.

Solving (5.22) **by using quadratures of dimension 1**

Focusing on the expression for the likelihood of a single point y_n in (5.22)

$$\begin{aligned} p(y_n|f_n, g_n) &= \mathcal{N}(y_n|\sigma(g_n)f_n, \nu^2) \\ &= \frac{1}{(2\pi\nu^2)^{1/2}} \exp\left\{-\frac{1}{2\nu^2}\left[y_n - \sigma(g_n)f_n\right]^2\right\}, \end{aligned}$$

then

$$\log p(y_n|f_n, g_n) = -\frac{1}{2}\log(2\pi) - \frac{1}{2}\log(\nu^2) - \frac{1}{2\nu^2}\left[y_n - \sigma(g_n)f_n\right]^2,$$

replacing this into (5.22) we get

$$\int\int \log p(y_n|f_n, g_n)q(f_n)q(g_n)\ \mathrm{d}f_n\ \mathrm{d}g_n =$$
$$-\frac{1}{2\nu^2}\int\int \left[y_n - \sigma(g_n)f_n\right]^2 q(f_n)q(g_n)\ \mathrm{d}f_n\ \mathrm{d}g_n - \frac{1}{2}\log(2\pi) - \frac{1}{2}\log(\nu^2),$$

where

$$\int\int \left[y_n - \sigma(g_n)f_n\right]^2 q(f_n)q(g_n)\ \mathrm{d}f_n\ \mathrm{d}g_n =$$
$$\int\int \left[y_n^2 - 2y_n\sigma(g_n)f_n + \sigma(g_n)^2 f_n^2\right] q(f_n)q(g_n)\ \mathrm{d}f_n\ \mathrm{d}g_n =$$
$$\int\int y_n^2 q(f_n)q(g_n)\ \mathrm{d}f_n\ \mathrm{d}g_n - \cdots$$
$$\int\int 2y_n\sigma(g_n)f_n q(f_n)q(g_n)\ \mathrm{d}f_n\ \mathrm{d}g_n + \cdots$$
$$\int\int \sigma(g_n)^2 f_n^2 q(f_n)q(g_n)\ \mathrm{d}f_n\ \mathrm{d}g_n =$$
$$y_n^2 - 2y_n\int f_n q(f_n)\ \mathrm{d}f_n \cdot \int \sigma(g_n)q(g_n)\ \mathrm{d}g_n + \int f_n^2 q(f_n)\ \mathrm{d}f_n \cdot \int \sigma(g_n)^2 q(g_n)\ \mathrm{d}g_n =$$
$$y_n^2 - 2y_n\mathbb{E}_{q(f_n)}\left[f_n\right]\mathbb{E}_{q(g_n)}\left[\sigma(g_n)\right] + \mathbb{E}_{q(f_n)}\left[f_n^2\right]\mathbb{E}_{q(g_n)}\left[\sigma(g_n)^2\right] =$$
$$y_n^2 - 2y_n m_{f,n}\mathbb{E}_{q(g_n)}\left[\sigma(g_n)\right] + (s_{f,n}^2 + m_{f,n}^2)\mathbb{E}_{q(g_n)}\left[\sigma(g_n)^2\right].$$

Then

$$\int\int \log p(y_n|f_n, g_n)q(f_n)q(g_n)\ \mathrm{d}f_n\ \mathrm{d}g_n = \tag{5.25}$$
$$-\frac{1}{2\nu^2}\left\{y_n^2 - 2y_n m_{f,n}\mathbb{E}_{q(g_n)}\left[\sigma(g_n)\right] + (s_{f,n}^2 + m_{f,n}^2)\mathbb{E}_{q(g_n)}\left[\sigma(g_n)^2\right]\right\} - \frac{1}{2}\log(2\pi) - \frac{1}{2}\log(\nu^2).$$

where $m_{f,n}$ and $s_{f,n}^2$ are the mean and variance of the variational distribution over the latent variable f_n. The expectations in the previous expression can be approximated using 2 one dimensional Gauss-Hermite quadrature. Therefore we have reduced the dimensionality of the approximate integrals.

Approximating $\mathbb{E}_{q(g_n)}\left[\sigma(g_n)\right]$ and $\mathbb{E}_{q(g_n)}\left[\sigma(g_n)^2\right]$ in (5.25) using 1 dimensional Gauss-Hermite quadrature

Here we study in more detail the expectations found (5.25). Specifically $\mathbb{E}_{q(g_n)}\left[\sigma(g_n)\right]$ and $\mathbb{E}_{q(g_n)}\left[\sigma(g_n)^2\right]$

$$\mathbb{E}_{q(g_n)}\left[\sigma(g_n)\right] = \int_{-\infty}^{\infty}\sigma(g_n)\frac{1}{(2\pi s_{g_n}^2)^{1/2}}\exp\left\{-\frac{1}{2s_{g_n}^2}(g_n - m_{g_n})^2\right\}\mathrm{d}g_n,$$

The Hermite-Gauss is defined for a normal distribution with zero mean, that is why we require a change of variable:

$$\begin{aligned}\tilde{x} =& \frac{g_n - m_{g_n}}{\sqrt{2}s_{g_n}},\\ \mathrm{d}g_n =& \sqrt{2}s_{g_n}\mathrm{d}\tilde{x},\end{aligned}$$

then we have

$$\frac{1}{\sqrt{\pi}}\int_{-\infty}^{\infty}\sigma(\sqrt{2}s_{g_n}\tilde{x} + m_{g_n})\exp(-\tilde{x}^2)\mathrm{d}\tilde{x},$$

calling $h(\tilde{x}) = \sigma(\sqrt{2}s_{g_n}\tilde{x} + m_{g_n})$, then

$$\frac{1}{\sqrt{\pi}}\int_{-\infty}^{\infty}h(\tilde{x})\exp(-\tilde{x}^2)\mathrm{d}\tilde{x},$$

Now we can approximate this integral using the Hermite-Gaussian quadrature, that is

$$\frac{1}{\sqrt{\pi}}\int_{-\infty}^{\infty}h(\tilde{x})\exp(-\tilde{x}^2)\mathrm{d}\tilde{x} \approx \frac{1}{\sqrt{\pi}}\sum_{\forall j} w_j\sigma(\sqrt{2}s_{g_n}x_j + m_{g_n}). \tag{5.26}$$

The expressions for $\mathbb{E}_{q(g_n)}\left[\sigma(g_n)^2\right]$ can be calculated similarly. Replacing (5.26) into (5.25) we get

$$\begin{aligned}&\int\int \log p(y_n|f_n, g_n)q(f_n)q(g_n)\ \mathrm{d}f_n\ \mathrm{d}g_n \approx \\ &-\frac{1}{2\nu^2}\left\{y_n^2 - 2y_n m_{f_n}\left[\frac{1}{\sqrt{\pi}}\sum_{\forall i} w_i\sigma(\sqrt{2}s_{g_n}\hat{x}_i + m_{g_n})\right] + \cdots\right.\\ &\left.(s_{f_n}^2 + m_{f_n}^2)\left[\frac{1}{\sqrt{\pi}}\sum_{\forall j} w_j\sigma(\sqrt{2}s_{g_n}\hat{y}_j + m_{g_n})^2\right]\right\} - \frac{1}{2}\log(2\pi) - \frac{1}{2}\log(\nu^2).\end{aligned} \tag{5.27}$$

5.2.2 Experiments

Next we present preliminary results applying SVI in GP models for two different tasks; separating three sources, and detecting 88 pitches on two

seconds of a piano audio recording from the MAPS dataset [28].

Source Separation

SVI was tested in the same problem addressed in chapter 5.1, that is, to separate three sources in a piano audio recording that lasts 14 seconds. The signal has up to three pitches happening at the same time (see Fig 5.6 bottom). With a sample rate of 16 kHz, the audio signal contains 224000 data points, which is a big data scenario for GPs. we applied the modulated GP model, introduced in Chapter 4.

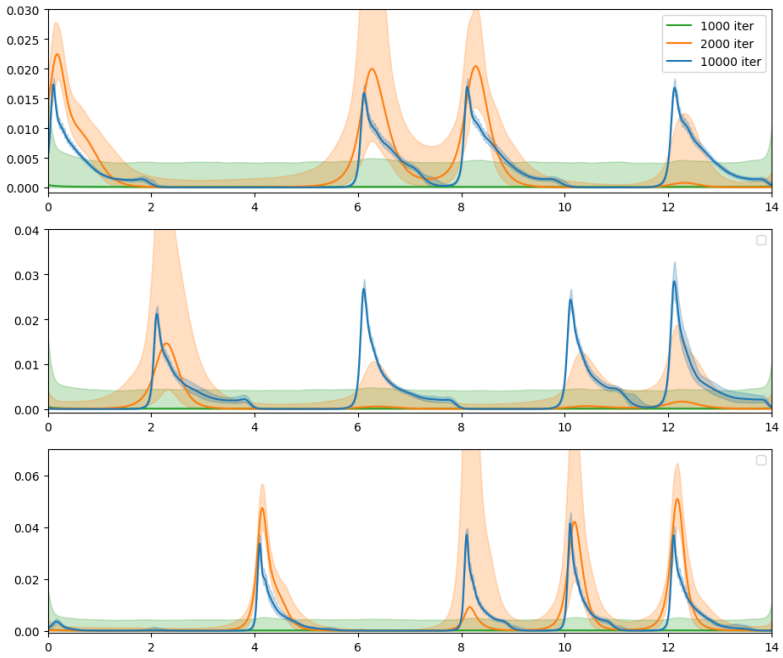


Figure 5.7: Predicted activations using modulated GP with SVI, after 1000, 2000, and 10000 iterations.

We observed that the inferred activations after 1000 iterations where close to zero (Figure 5.7 green curves). However, after 2000 iterations the distinctive pattern of each activation became evident as either the predicted mean

or/and the interval of confidence increased (Figure 5.7 orange curves). Activations reached a specific form after 10000 iterations (Figure 5.7 blue curves). These observations confirm that the predictions obtained using SVI are continuous, indicating that all the information in the whole dataset is considered during inference, in a stochastic fashion.

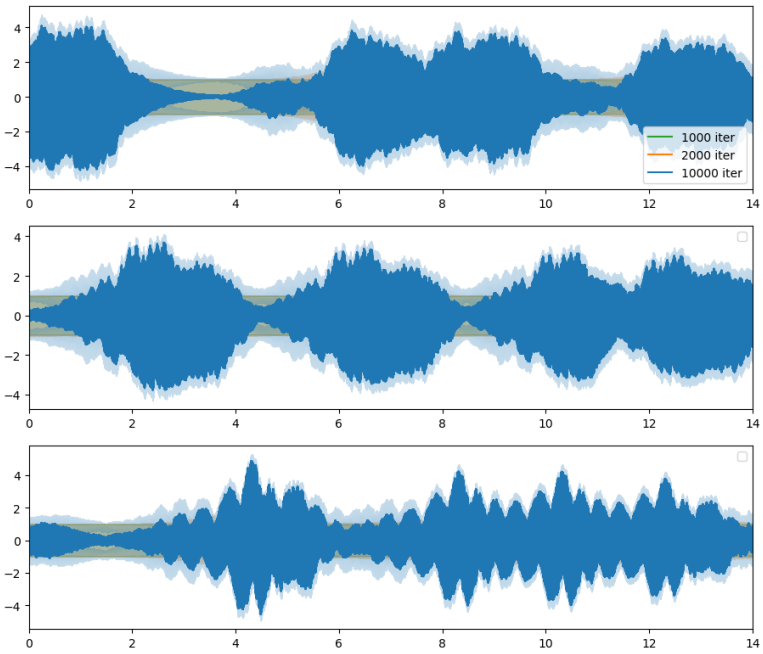


Figure 5.8: Predicted components using modulated GP with SVI, after 1000, 2000, and 10000 iterations.

Likewise, we found that the learnt components presented a similar behaviour (Figure 5.8). That is, they were continuous and had a characteristic pattern that differentiates them between each other. The form of these functions was not well-defined even after 1000 or 2000 iterations. Nevertheless, these functions converged after 10000 iterations.

Finally, the sources where were reconstructed by multiplying the corresponding activations and components predictive means shown in Fig 5.7 and Figure 5.8. We detected that the reconstructed sources were close to the true

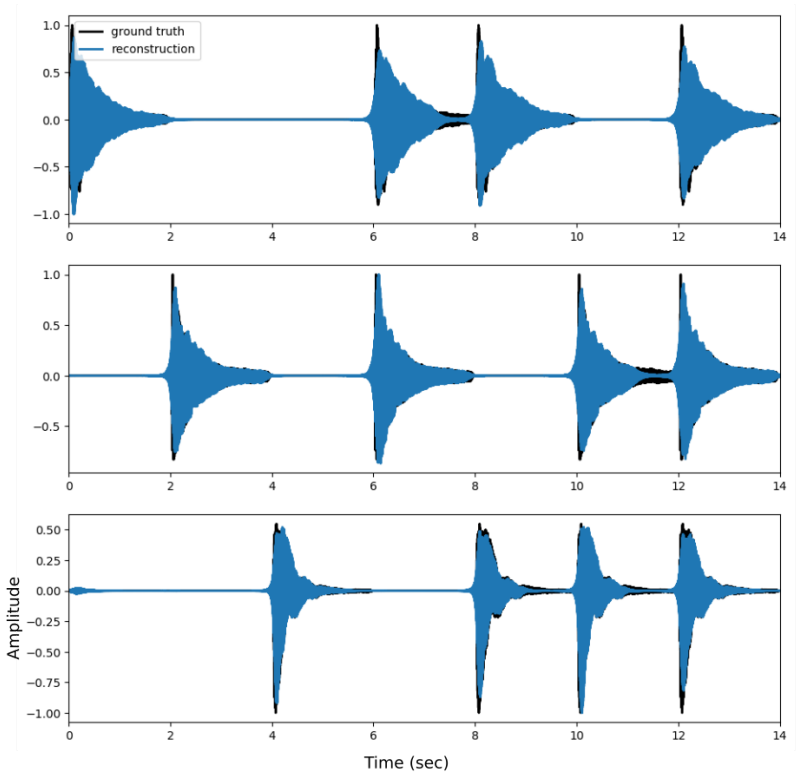


Figure 5.9: Predicted sources using modulated GP with SVI.

data, i.e. they matched the sequence of events for each source (Figure 5.9). Nevertheless, the reconstructed sources were smoother than expected. Also, a small outlier was observed at the beginning of the third source. These observations indicate that SVI based GP models find challenging to predict sharp changes in the audio recordings. This could be because in an audio recording most of the audio corresponds to the steady-state or decay of the sound events. In other words, the onset of the sounds last for a very short time, therefore there is few available onset-related data.

Multi-pitch detection

To establish if SVI truly enables GP models for detecting pitches in a real music scenario, that is, predicting the 88 pitches/keys of a piano, we applied the modulated GP model (chapter 4) to a two seconds audio signal. This audio segment contains five different pitches, still, we predicted the activation, components, and sources of the full range of 88 notes. In short, the model outputs a waveform for each pitch, and for each of these waveforms we define if a pitch is active or not depending on their energy at a certain time. The top left of Figure 5.10 shows the ground truth piano-roll of the analysed signal, the red squares demarcate the onsets of the notes. We found that the proposed GP method was able to detect the true pitches occurring in the audio signal (Figure 5.10 top right). However, several false positive activations were also present (principally octaves, fifths and thirds of the true pitches). Also, two onsets were missing. These observations indicate that GP models that rely solely on acoustic modelling of the sources are prone to make mistakes that could be avoided by introducing a joint prior over the activations with musical meaning, i.e. a music language model/prior. We also observed that the evidence lower bound stabilised after 20000 iterations, indicating that the SVI algorithm converged to a (possibly local) maximum (Figure 5.10 bottom). Finally, we observed that the proposed GP method using SVI was able to produce the 88 waveforms associated with the 88 pitches we aimed to detect (Figure 5.11), indicating that GP models combined with SVI are a promising alternative for integrating Bayesian non-parametric GP models into big-data music signal processing tasks.

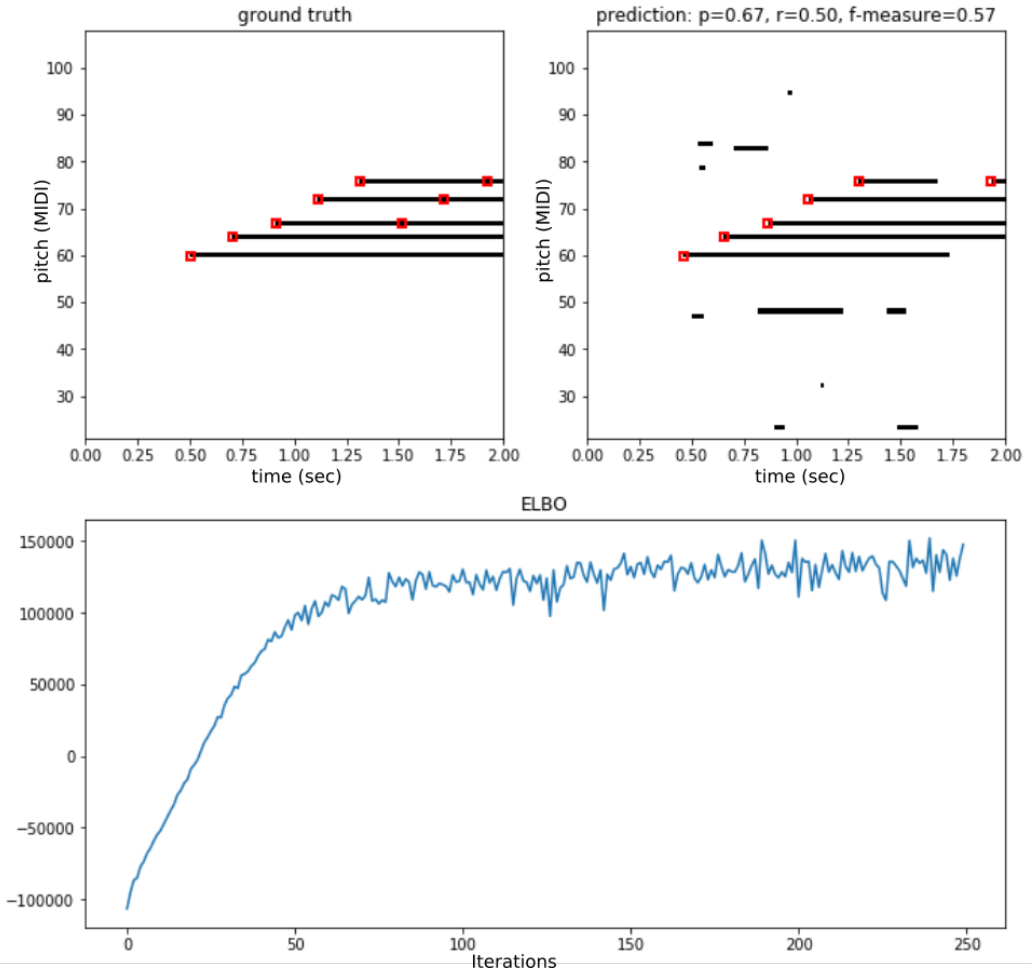


Figure 5.10: Example of multi-pitch estimation and source separation. Ground truth piano-roll (top left). Estimated piano-roll (top right). ELBO convergence (bottom).

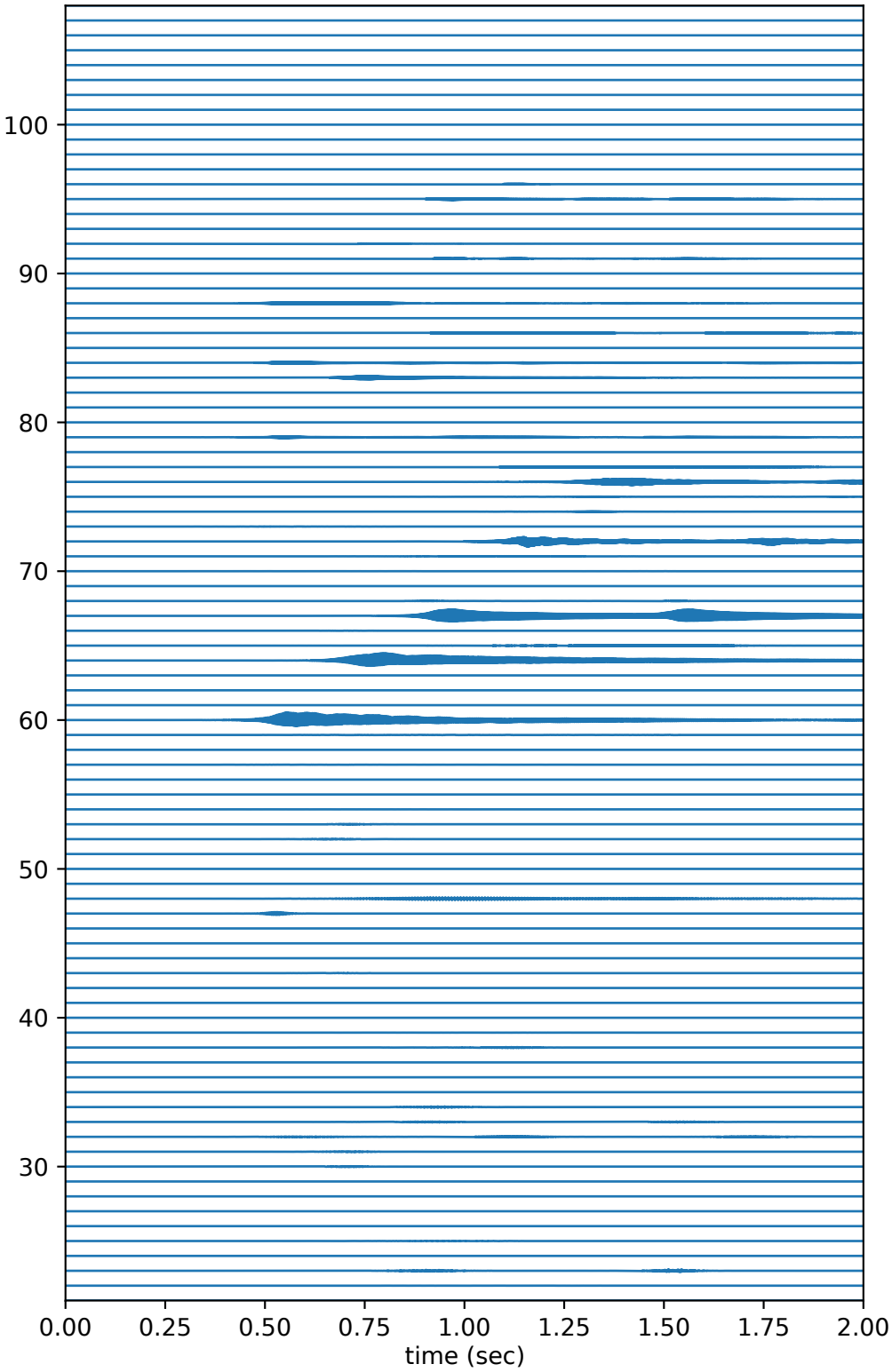


Figure 5.11: 88 waveforms reconstructed from a single polyphonic signal of piano (MAPS dataset).

Chapter 6

Conclusions and further work

In the achievement of the general aim of this research "to develop Bayesian machine learning methods that interpret source separation and multi-pitch detection as a single unifying task" (Section 1.2), this thesis has centred on the construction of time-domain probabilistic models based on Gaussian processes. Furthermore, this study has focused on making the proposed GP models efficient and interpretable from a music audio signal perspective. To verify the proposed methods, we have investigated the design of kernels or covariance functions that lead to GP priors with audio signals characteristics, while studying how to apply approximate variational inference. To close this document, we first review the contributions made. Subsequently, we reflect on the outcomes of our study. Finally, we suggest possible directions in which future work can extend this research.

6.1 Summary of contributions

- We explored a variety of covariance functions, as an initial stage aiming to find GP priors that represented better the properties of audio signals (Chapter 3). We found that conventional kernels tended to describe a restricted scope of audio features, including dynamics, quasi-periodicity, and an oversimplified harmonic content. Our early findings were used to direct the research towards the designing of more powerful

covariance functions.

We also analysed the usage of deterministic activation functions or change-windows to include non-stationarity in GP regression models. In the experiments, the activation functions were a determinant factor in improving the performance of pitch detection GP models. However, the predefined number of change-windows and the estimation of more hyperparameters limited the proposed approach. From these results, we hypothesised that non-parametric activation functions were a promising alternative for overcoming these limitations.

- We proposed to use the GP-product model as the basis of our multi-pitch detection approach [4] (Chapter 4). Here, for each pitch, a point-wise multiplication of two Gaussian processes described the corresponding source waveform. One GP (component function) embodied harmonic content, whereas the second GP represented the time-changing amplitude, i.e. the source activation. Also, we proposed to use the Matérn spectral mixture (MSM) kernel to specify the prior over the component functions [83]. In our experiments, the highest performance in multi-pitch detection was achieved when the MSM kernels were initialised using the Fourier transform of isolated sources available for training. We concluded that "what it is really relevant for pitch detection is a set of MSM kernels that properly fit the frequency content of the target sound events" [8].

  Besides, to tackle the intractability issue produced by a non-closed-form posterior, we introduced sparse variational inference into the multi-pitch detection GP model [76]. Also, to address the computational cost of doing inference in large data sets, we suggested framing the waveform of the input audio signal. Results imply that making GP models suitable for real scenarios of music signal processing may require to use sparse variational inference together with framing audio signals.

- We studied standard Gaussian process regression for source separation in single-instrument music signals (Section 5.1). Here we also framed

the audio signal and used sparse variational inference for tractability. We found that accurate and efficient source reconstructions were possible when learning hyperparameters by maximising the evidence lower bound while computing the conditional distribution over each source given the mixture data (source posterior) using its classic closed-form definition. Furthermore, in our experiments, the best performance in separation was achieved when the spectral mixture kernels were very close to the empirical autocorrelation of the training data (waveforms of isolated sources). Our findings can serve to establish a method to obtain GP priors with accurate spectral content representations.

- We developed `GPitch`, a new Python package intended for time-domain source separation and multi-pitch detection using Gaussian processes. This package relies on `GPflow` and `TensorFlow`, which makes it computational efficient when using sparse variational inference methods running on GPUs.

- We introduced stochastic variational inference (SVI) into time-domain GP models for multi-pitch and source separation. SVI allows to analyse music signals without framing them, circumventing prediction discontinuities between audio windows. This thesis presented preliminary results in Section 5.2.

Some of these contributions were submitted and/or published in international conferences. For a detailed list please refer to Section 1.4.

6.2 Further work

This research could be extended to different areas. Here, we describe potential directions.

Variational Fourier features: One of the shortcomings of sparse Gaussian process variational approach is that the inducing variables lie on the same domain as the input training data variables, that is,

time. Therefore, to analyse longer audio signals may imply to use more inducing points. A different possibility is to use inducing points that exist on a different domain. A promising alternative is to use variational Fourier features, where the inducing points are in the frequency domain [33]. In short, analysing longer signals would not necessarily require more inducing points in the time-domain; it would instead need to relocate the inducing features in the frequency domain as the size of the audio signal increases.

Music language models: A music language model could be introduced as a prior joint distribution over the activation functions [67, 85, 58]. Consequently, activations could exhibit musical meaning when analysing them simultaneously. This approach could reduce the number of mistakes when making predictions by avoiding the combinations of activations that are less likely under specific music rules. For example, using a music language model could be beneficial for ignoring the activation of pitches whose relative distance is one semitone, thus discouraging dissonant mixtures of pitches. Besides, spurious and intermittent activations could be excluded by using the same principle.

Non-stationary spectral mixture kernels: Another potential extension of this research is the design of non-stationary spectral mixture kernels that can encode or describe the attack and decay of pitched music sound events. In short, such covariance functions could incorporate the time-dependent evolution of the spectral content of sources [57]. The usage of attack-decay spectral mixture kernels may increase the accuracy in source reconstructions and reduce the number of miss-detections in multi-pitch estimation.

Combinations with deep learning: The proposed nonparametric approach could be combined with deep learning. This combination could be beneficial, for example, to develop more powerful kernels that can extract and learn relevant features directly from the audio data, without needing a human expert to select a specific family of covariance

functions [84, 26]. Also, a hybrid approach using deep learning and GPs could have both, the effective function modelling of deep learning and the Bayesian uncertainty quantification paradigm offered by Gaussian process regression.

Multi-output Gaussian processes: Music language models could be coupled with multi-output Gaussian processes and multi-task learning. This combination could potentially improve the performance of multi-pitch detection GP models. This is because sharing information between tasks (learning each source/pitch) usually makes the overall model more robust [49].

6.3 Closing remarks

In general, multi-pitch estimation and source separation remain open challenges in music signal processing. Our work proposes a unifying approach that addresses both tasks simultaneously, specifically for single-instrument music audio signals. The time-domain methods we have introduced give time-level resolution in multi-pitch detection and reconstruct sources without requiring phase approximations. Furthermore, the Gaussian process models we have developed provide an elegant and principled formalism for introducing prior knowledge about activations (smoothness and continuity) and sources (spectral content), circumventing post-processing steps for smoothing the activations. Moreover, the proposed GP methods quantify uncertainty in the predictions; a property that source separation and multi-pitch detection systems usually do not have or exploit.

Making GP models suitable for full-scale music signal processing scenarios needs further research. For example, to detect a broader range of pitches such as the 88 notes of a piano, in an audio recording of a complete piece of music that last several minutes, demands more efficient and scalable GP approximate methods. Such a task could also need new kernels that reflect more intricate properties of audio signals (attack and decay), and music language priors that give more relevance to activations that make sense musically.

Analysing the outcomes in chapters 3, 4, 5.1 and 5.2, we conclude that time-domain Gaussian process models are a promising approach to solving multi-pitch detection and source separation in polyphonic music signals. Our findings can be used to guide the selection of covariance functions when applying GP models to music signal processing problems. Besides, we proposed two different methods to initialise source kernels within a region with practical music and audio interpretation. Our results indicate that GP priors that successfully incorporate the spectral content of sources/pitches are determining to obtain high-quality source reconstructions and pitch estimates. Furthermore, variational inference (either sparse or stochastic) should be used to break free from the GP computational burden and release the full potentiality that GPs have in music signal processing.

Appendix A

Gaussian distribution identities

These derivations were obtained from [56] (appendix A). Suppose $\mathbf{x}$ and $\mathbf{y}$ are Gaussian multivariate random variables with joint distribution $p(\mathbf{x}, \mathbf{y})$ given by

$$\begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \boldsymbol{\mu}_x \\ \boldsymbol{\mu}_y \end{bmatrix}, \begin{bmatrix} \mathbf{A} & \mathbf{C} \\ \mathbf{C}^\top & \mathbf{B} \end{bmatrix} \right),$$

then, it can be shown that the conditional distribution $p(\mathbf{x}|\mathbf{y})$ is given by

$$p(\mathbf{x}|\mathbf{y}) = \mathcal{N}\left(\mathbf{x}|\ \hat{\boldsymbol{\mu}}, \hat{\mathbf{A}}\right),$$

where

$$\hat{\boldsymbol{\mu}} = \boldsymbol{\mu}_x + \mathbf{C}\mathbf{B}^{-1}(\mathbf{y} - \boldsymbol{\mu}_y),$$

and

$$\hat{\mathbf{A}} = \mathbf{A} - \mathbf{C}\mathbf{B}^{-1}\mathbf{C}^\top.$$

Appendix B

Leave one out: model with two sources

Likelihood

Assuming the regression model

$$y(t) = \sum_{d=1}^{D} \sigma(g^{(d)}(t)) f^{(d)}(t) + \epsilon(t)$$

with $D = 2$, then

$$y_n = \sigma(g_n^{(1)}) f_n^{(1)} + \sigma(g_n^{(2)}) f_n^{(2)} + \epsilon_n$$

assuming the observations as i.i.d then the likelihood corresponds to

$$\begin{aligned} p(\mathbf{y}|\mathbf{F}, \mathbf{G}) &= \prod_{n=1}^{N} p(y_n|\mathbf{f}_n, \mathbf{g}_n) \\ &= \prod_{n=1}^{N} \mathcal{N}(y_n|\hat{\mathbf{g}}_n^\top \mathbf{f}_n, \nu^2), \end{aligned} \tag{B.1}$$

where the components of the matrices $[\mathbf{F}]_{n,d} = f_n^{(d)}$, $[\mathbf{G}]_{n,d} = g_n^{(d)}$, then each row in $\mathbf{F}$ and $\mathbf{G}$ is given by the vectors $\mathbf{f}_n^\top = [f_n^{(1)}, f_n^{(2)}]$, $\mathbf{g}_n^\top = [g_n^{(1)}, g_n^{(2)}]$. Finally, $\hat{\mathbf{g}}_n = [\sigma(g_n^{(1)}),\ \sigma(g_n^{(2)})]^\top$ represents the non-linear transformation of

the envelope processes.

Analysing the log-likelihood

From (B.1) we get the log-likelihood

$$\begin{aligned}
\log p(\mathbf{y}|\mathbf{F},\mathbf{G}) &= \sum_{n=1}^{N} \log p(y_n|\mathbf{f}_n,\mathbf{g}_n) \qquad \text{(B.2)}\\
&= \sum_{n=1}^{N} \log \mathcal{N}(y_n|\hat{\mathbf{g}}_n^\top \mathbf{f}_n, \nu^2)\\
&= \sum_{n=1}^{N} \left[-\frac{1}{2}\log(2\pi) - \frac{1}{2}\log(\nu^2) - \frac{1}{2\nu^2}\left(y_n - \hat{\mathbf{g}}_n^\top \mathbf{f}_n\right)^2\right]\\
&= -\frac{N}{2}\log(2\pi) - \frac{N}{2}\log(\nu^2) - \frac{1}{2\nu^2}\sum_{n=1}^{N}\left(y_n - \hat{\mathbf{g}}_n^\top \mathbf{f}_n\right)^2\\
&= -\frac{N}{2}\log(2\pi) - \frac{N}{2}\log(\nu^2) - \frac{1}{2\nu^2}\sum_{n=1}^{N}\left\{y_n - \left[\sigma(g_n^{(1)})f_n^{(1)} + \sigma(g_n^{(2)})f_n^{(2)}\right]\right\}^2.
\end{aligned}$$

We are interested in calculating

$$\mathbb{E}_{q(\mathbf{F},\mathbf{G})}[\log p(\mathbf{y}|\mathbf{F},\mathbf{G})],$$

Then

$$\int\int\int\int \log p(\mathbf{y}|\mathbf{f}^{(1)},\mathbf{f}^{(2)},\mathbf{g}^{(1)},\mathbf{g}^{(2)}) q(\mathbf{f}^{(1)})q(\mathbf{f}^{(2)})q(\mathbf{g}^{(1)})q(\mathbf{g}^{(2)})\ \mathrm{d}\mathbf{f}^{(1)}\ \mathrm{d}\mathbf{f}^{(2)}\ \mathrm{d}\mathbf{g}^{(1)}\ \mathrm{d}\mathbf{g}^{(2)}.$$

We can calculate the previous fourth integral using a 4 dimensional Gauss-Hermite quadrature.

Now we get an expression where only 1 dimensional quadratures are re-

quired.

$$\int\int\int\int \log p(\mathbf{y}|\mathbf{f}^{(1)},\mathbf{f}^{(2)},\mathbf{g}^{(1)},\mathbf{g}^{(2)})q(\mathbf{f}^{(1)})q(\mathbf{f}^{(2)})q(\mathbf{g}^{(1)})q(\mathbf{g}^{(2)})\ \mathrm{d}\mathbf{f}^{(1)}\ \mathrm{d}\mathbf{f}^{(2)}\ \mathrm{d}\mathbf{g}^{(1)}\ \mathrm{d}\mathbf{g}^{(2)} =$$
$$-\frac{N}{2}\log(2\pi)-\frac{N}{2}\log(\nu^2)-$$
$$\frac{1}{2\nu^2}\sum_{n=1}^{N}\int\int\int\int \left[y_n-\sigma(g_n^{(1)})f_n^{(1)}-\sigma(g_n^{(2)})f_n^{(2)}\right]^2\times\cdots$$
$$q(f_n^{(1)})q(f_n^{(2)})q(g_n^{(1)})q(g_n^{(2)})\ \mathrm{d}f_n^{(1)}\ \mathrm{d}f_n^{(2)}\ \mathrm{d}g_n^{(1)}\ \mathrm{d}g_n^{(2)}$$

the previous expression can be decomposed into 6 quadruple-integrals

$$\int\int\int\int \Big\{y_n^2-2y_n\sigma(g_n^{(1)})f_n^{(1)}-2y_n\sigma(g_n^{(2)})f_n^{(2)}+\left[\sigma(g_n^{(1)})f_n^{(1)}\right]^2+\cdots$$
$$2\sigma(g_n^{(1)})f_n^{(1)}\sigma(g_n^{(2)})f_n^{(2)}+\left[\sigma(g_n^{(2)})f_n^{(2)}\right]^2\Big\}\times\cdots$$
$$q(f_n^{(1)})q(f_n^{(2)})q(g_n^{(1)})q(g_n^{(2)})\ \mathrm{d}f_n^{(1)}\ \mathrm{d}f_n^{(2)}\ \mathrm{d}g_n^{(1)}\ \mathrm{d}g_n^{(2)} =$$

$$y_n^2-2y_n\left\{\tilde{m}_n^{f^{(1)}}\mathbb{E}\left[\sigma\left(g_n^{(1)}\right)\right]+\tilde{m}_n^{f^{(2)}}\mathbb{E}\left[\sigma\left(g_n^{(2)}\right)\right]\right\}+\left[\left(\tilde{m}_n^{f^{(1)}}\right)^2+\tilde{\nu}_n^{f^{(1)}}\right]\mathbb{E}\left[\sigma\left(g_n^{(1)}\right)^2\right]+\cdots$$
$$2\tilde{m}_n^{f^{(1)}}\tilde{m}_n^{f^{(2)}}\mathbb{E}\left[\sigma\left(g_n^{(1)}\right)\right]\mathbb{E}\left[\sigma\left(g_n^{(2)}\right)\right]+\left[\left(\tilde{m}_n^{f^{(2)}}\right)^2+\tilde{\nu}_n^{f^{(2)}}\right]\mathbb{E}\left[\sigma\left(g_n^{(2)}\right)^2\right].$$

From the last expression we conclude that the 4-dimensional integral needed to compute the expectation of the log-likelihood can be calculated as a combination of four 1-dimensional integrals. This allows to use 1D-quadrature instead of 4D-quadratures, reducing computation time and memory. Rewriting we get:

$$\mathbb{E}_{q(\mathbf{F}:,\mathbf{G}:)}[\log p(\mathbf{y}|\mathbf{F}:,\mathbf{G}:)] = \tag{B.3}$$
$$-\frac{N}{2}\log(2\pi)-\frac{N}{2}\log(\nu^2)-\frac{1}{2\nu^2}\sum_{n=1}^{N}\Big\{y_n^2-2y_n\left[\tilde{m}_n^{f^{(1)}}\mathbb{E}\left[\sigma\left(g_n^{(1)}\right)\right]+\tilde{m}_n^{f^{(2)}}\mathbb{E}\left[\sigma\left(g_n^{(2)}\right)\right]\right]+$$
$$\left[\left(\tilde{m}_n^{f^{(1)}}\right)^2+\tilde{\nu}_n^{f^{(1)}}\right]\mathbb{E}\left[\sigma\left(g_n^{(1)}\right)^2\right]+2\tilde{m}_n^{f^{(1)}}\tilde{m}_n^{f^{(2)}}\mathbb{E}\left[\sigma\left(g_n^{(1)}\right)\right]\mathbb{E}\left[\sigma\left(g_n^{(2)}\right)\right]+$$
$$\left[\left(\tilde{m}_n^{f^{(2)}}\right)^2+\tilde{\nu}_n^{f^{(2)}}\right]\mathbb{E}\left[\sigma\left(g_n^{(2)}\right)^2\right]\Big\}.$$

Bibliography

[1] `https://github.com/PabloAlvarado/ssgp`.

[2] S. A. Abdallah and M. D. Plumbley. Unsupervised analysis of polyphonic music by sparse coding. *IEEE Transactions on Neural Networks*, 17(1):179–196, Jan 2006.

[3] Vincent Adam, James Hensman, and Maneesh Sahani. Scalable transformed additive signal decomposition by non-conjugate Gaussian process inference. In *26th IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*, pages 1–6, 2016.

[4] Ryan Prescott Adams and Oliver Stegle. Gaussian process product models for nonparametric nonstationarity. In *Proceedings of the 25th international conference on Machine learning*, pages 1–8. ACM, 2008.

[5] J. B. Allen and L. R. Rabiner. A unified approach to short-time Fourier analysis and synthesis. *Proceedings of the IEEE*, 65(11):1558–1564, Nov 1977.

[6] P. A. Alvarado, M. A. Alvarez, and D. Stowell. Sparse Gaussian process audio source separation using spectrum priors in the time-domain. In *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 995–999, May 2019.

[7] Pablo A. Alvarado and Dan. Stowell. Gaussian processes for music audio modelling and content analysis. In *2016 IEEE 26th International Workshop on Machine Learning for Signal Processing (MLSP)*, pages 1–6, Sept 2016.

[8] Pablo A. Alvarado and Dan. Stowell. Efficient learning of harmonic priors for pitch detection in polyphonic music. *arXiv preprint arXiv:1705.07104*, 2017.

[9] M. Álvarez, D. Luengo, and N. D. Lawrence. Linear latent force models using Gaussian processes. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35(11):2693–2705, Nov 2013.

[10] Papoulis Athanasious. *Probability, Random Variables, and Stochastic Process*. McGraw-Hill, Inc, 1991.

[11] J. P. Bello, L. Daudet, S. Abdallah, C. Duxbury, M. Davies, and M. B. Sandler. A tutorial on onset detection in music signals. *IEEE Transactions on Speech and Audio Processing*, 2005.

[12] Emmanouil Benetos and Simon Dixon. Multiple-instrument polyphonic music transcription using a temporally constrained shift-invariant model. *The Journal of the Acoustical Society of America*, 133:1727–41, 03 2013.

[13] Emmanouil Benetos, Simon Dixon, Zhiyao Duan, and Sebastian Ewert. Automatic Music Transcription: An Overview. *IEEE Signal Processing Magazine*, 36(1):20–30, 2019.

[14] Emmanouil Benetos, Simon Dixon, Dimitrios Giannoulis, Holger Kirchhoff, and Anssi Klapuri. Automatic music transcription: Breaking the glass ceiling. In *Proceedings of the 13th International Society for Music Information Retrieval Conference*, Porto, Portugal, October 8-12 2012.

[15] Emmanouil Benetos, Simon Dixon, Dimitrios Giannoulis, Holger Kirchhoff, and Anssi Klapuri. Automatic music transcription: challenges and future directions. *Journal of Intelligent Information Systems*, 41(3):407–434, 2013.

[16] Taylor Berg-Kirkpatrick, Jacob Andreas, and Dan Klein. Unsupervised transcription of piano music. In Z. Ghahramani, M. Welling, C. Cortes,

N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 1538–1546. Curran Associates, Inc., 2014.

[17] Taylor Berg-Kirkpatrick, Jacob Andreas, and Dan Klein. Unsupervised transcription of piano music. In *Advances in Neural Information Processing Systems 27 (NIPS)*, pages 1538–1546. Curran Associates, Inc., 2014.

[18] Alain Berlinet and CHRISTINE Thomas-Agnan. *Reproducing Kernel Hilbert Spaces In Probability and Statistics*. Sringer, 2004.

[19] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer-Verlag, New York, 2006.

[20] David M. Blei, Alp Kucukelbir, and Jon D. McAuliffe. Variational Inference: A Review for Statisticians. *arXiv e-prints*, page arXiv:1601.00670, Jan 2016.

[21] S. Böck and M. Schedl. Polyphonic piano note transcription with recurrent neural networks. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 121–124, March 2012.

[22] A. T. Cemgil, S. J. Godsill, P. Peeling, and N. Whiteley. Bayesian statistical methods for audio and music processing. *The Oxford Handbook of Applied Bayesian Analysis*, pages 1–45, 2010.

[23] Mads Christensen and Andreas Jakobsson. *Multi-Pitch Estimation*. Morgan and Claypool Publishers, 2009.

[24] A. Cogliati, Z. Duan, and B. Wohlberg. Context-dependent piano music transcription with convolutional sparse coding. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 24(12):2218–2230, Dec 2016.

[25] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms*. The MIT Press, 2nd edition, 2001.

[26] Zhenwen Dai, Andreas C. Damianou, Javier González, and Neil D. Lawrence. Variational auto-encoded deep Gaussian processes. In *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, 2016.

[27] Nicolas Durrande, James Hensman, Magnus Rattray, and Neil D. Lawrence. Detecting periodicities with Gaussian processes. *PeerJ Computer Science*, 2:e50, 2016.

[28] Valentin Emiya, Nancy Bertin, Bertrand David, and Roland Badeau. Maps - a piano database for multipitch estimation and automatic transcription of music, 07 2010.

[29] Cédric Févotte and Matthieu Kowalski. Low-rank time-frequency synthesis. In *Advances in Neural Information Processing Systems 27 (NIPS)*, pages 3563–3571. Curran Associates, Inc., 2014.

[30] B. Fuentes, R. Badeau, and G. Richard. Harmonic adaptive latent component analysis of audio and application to music transcription. *IEEE Transactions on Audio, Speech, and Language Processing*, 21(9):1854–1866, Sep. 2013.

[31] M. Goulard and M. Voltz. Linear coregionalization model: Tools for estimation and choice of cross-variogram matrix. *Mathematical Geosciences*, 24(3):269–286, 1992.

[32] Curtis Hawthorne, Erich Elsen, Jialin Song, Adam Roberts, Ian Simon, Colin Raffel, Jesse Engel, Sageev Oore, and Douglas Eck. Onsets and frames: Dual-objective piano transcription. In *Proceedings of the 19th International Society for Music Information Retrieval Conference, ISMIR 2018, Paris, France, September 23-27, 2018*, pages 50–57, 2018.

[33] James Hensman, Nicolas Durrande, and Arno Solin. Variational Fourier features for Gaussian processes. *Journal of Machine Learning Research*, 18(151):1–52, 2018.

[34] James Hensman, Nicoló Fusi, and Neil D. Lawrence. Gaussian processes for big data. In *20th Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 282–290, 2013.

[35] Matthew D. Hoffman, David M. Blei, Chong Wang, and John Paisley. Stochastic variational inference. *Journal of Machine Learning Research*, 14:1303–1347, 2013.

[36] Alexander G de G Matthews James Hensman and Zoubin Ghahramani. Scalable variational Gaussian process classification. In *18th International Conference on Artificial Intelligence and Statistics, San Diego, California, USA*, May 2015.

[37] M.B. Kanevsky. *Radar Imaging of the Ocean Waves*. Elsevier, 2009.

[38] A. N. Kolmogorov. *Grundbegriffe der Wahrscheinlichkeitsrechnung [Foundations of the Theory of Probability].* 1933.

[39] Daniel D. Lee and H. Sebastian Seung. Algorithms for non-negative matrix factorization. In *Advances in Neural Information Processing Systems 13 (NIPS)*, pages 556–562. MIT Press, 2001.

[40] Simon Leglaive, Laurent Girin, and Radu Horaud. Semi-supervised multichannel speech enhancement with variational autoencoders and non-negative matrix factorization. In *ICASSP 2019 - IEEE International Conference on Acoustics Speech and Signal Processing*, pages 101–105, Brighton, United Kingdom, May 2019. IEEE.

[41] Antoine Liutkus, Roland Badeau, and Gäel Richard. Gaussian processes for underdetermined source separation. *IEEE Transactions on Signal Processing*, 59(7):3155–3167, July 2011.

[42] James Robert Lloyd, David Duvenaud, Roger Grosse, Joshua B. Tenenbaum, and Zoubin Ghahramani. Automatic construction and Natural-Language description of nonparametric regression models. In *Association for the Advancement of Artificial Intelligence (AAAI)*, 2014.

[43] David J. C. MacKay. Introduction to Gaussian processes. In C. M. Bishop, editor, *Neural Networks and Machine Learning*, NATO ASI Series, pages 133–166. Kluwer Academic Press, 1998.

[44] K. Markov and T. Matsui. Music genre and emotion recognition using Gaussian processes. *IEEE Access*, 2:688–697, 2014.

[45] Alexander G. de G. Matthews, Mark van der Wilk, Tom Nickson, Keisuke. Fujii, Alexis Boukouvalas, Pablo León-Villagrá, Zoubin Ghahramani, and James Hensman. GPflow: A Gaussian process library using TensorFlow. *arXiv preprint 1610.08733*, October 2016.

[46] Alexander G. de G. Matthews, Mark van der Wilk, Tom Nickson, Keisuke. Fujii, Alexis Boukouvalas, Pablo León-Villagrá, Zoubin Ghahramani, and James Hensman. GPflow: A Gaussian process library using TensorFlow. *Journal of Machine Learning Research*, 18(40):1–6, apr 2017.

[47] M. Mauch and S. Dixon. Pyin: A fundamental frequency estimator using probabilistic threshold distributions. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 659–663, May 2014.

[48] Neil M. McLachlan. Timbre, pitch, and music, 06 2016.

[49] Pablo Moreno-Muñoz, Antonio Artés, and Mauricio Álvarez. Heterogeneous multi-output Gaussian process prediction. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 6711–6720. Curran Associates, Inc., 2018.

[50] M. Muller, D.P.W. Ellis, A. Klapuri, and G. Richard. Signal processing for music analysis. *Selected Topics in Signal Processing, IEEE Journal of*, 5(6):1088–1110, Oct 2011.

[51] Meinard Müller. *Fundamentals of Music Processing*. 2015.

[52] Kevin P. Murphy. *Machine Learning: A Probabilistic Perspective*. The MIT Press, 2012.

[53] Acoustical Society of America. Secretariat and American National Standards Institute. *American National Standard Psychoacoustical Terminology*. American National Standard. American National Standards Institute, 1986.

[54] Y. Ohishi, D. Mochihashi, H. Kameoka, and K. Kashino. Mixture of Gaussian process experts for predicting sung melodic contour with expressive dynamic fluctuations. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3714–3718, May 2014.

[55] J. Quinonero Candela and CE. Rasmussen. A unifying view of sparse approximate Gaussian process regression. *Journal of Machine Learning Research*, 6:1935–1959, December 2005.

[56] Carl Edward Rasmussen and Christopher K.I. Williams. *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press, 2005.

[57] Sami Remes, Markus Heinonen, and Samuel Kaski. Non-stationary spectral kernels. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 4642–4651. Curran Associates, Inc., 2017.

[58] Adam Roberts, Jesse Engel, Colin Raffel, Curtis Hawthorne, and Douglas Eck. A hierarchical latent vector model for learning long-term structure in music. In *Proceedings of the 35th International Conference on Machine Learning*, pages 4364–4373. PMLR, 10–15 Jul 2018.

[59] S. Roberts, M. Osborne, M. Ebden, S. Reece, N. Gibson, and S. Aigrain. Gaussian processes for time-series modelling. *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 371(1984), 2012.

[60] J. Salamon, E. Gomez, D. P. W. Ellis, and G. Richard. Melody extraction from polyphonic music signals: Approaches, applications, and challenges. *IEEE Signal Processing Magazine*, 31(2):118–134, March 2014.

[61] S. Särkkä and A. Solin. *Image Analysis: 18th Scandinavian Conference, SCIA 2013, Espoo, Finland, June 17-20, 2013. Proceedings*, pages 172–181. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.

[62] S. Särkkä, A. Solin, and J. Hartikainen. Spatiotemporal learning via infinite-dimensional Bayesian filtering and smoothing: A look at Gaussian process regression through Kalman filtering. *IEEE Signal Processing Magazine*, 30(4):51–61, July 2013.

[63] Alan D. Saul, James Hensman, Aki Vehtari, and Neil D. Lawrence. Chained Gaussian processes. In *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics, AISTATS 2016, Cadiz, Spain, May 9-11, 2016*, pages 1431–1440, 2016.

[64] Xavier Serra, Michela Magas, Emmanouil Benetos, Magdalena Chudy, Simon Dixon, Arthur Flexer, Emilia Gómez, Fabien Gouyon, Perfecto Herrera, Sergi Jorda, Oscar Paytuvi, Geoffroy Peeters, Jan Schlüter, Hugues Vinet, and Gerhard Widmer. *Roadmaop for Music Information Research*. Creative Commons BY-NC-ND 3.0 license, 2013.

[65] K. Sam Shanmugan and Arthur M. Breipohl. *Random Signals: Detection, Estimation and Data Analysis*. Wiley, 1988.

[66] John Shawe-Taylor and Nello Cristianini. *Kernel Methods for Pattern Analysis*. Cambridge University Press, New York, NY, USA, 2004.

[67] S. Sigtia, E. Benetos, and S. Dixon. An end-to-end neural network for polyphonic piano music transcription. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 24(5):927–939, May 2016.

[68] Paris Smaragdis and Bhiksha Raj. Shift-invariant probabilistic latent component analysis. Technical report, Mitsubishi Electric Research Laboratories, 2007.

[69] M.L. Stein. *Interpolation of Spatial Data*. Springer-Verlag, New York, 1999.

[70] Daniel Stoller, Sebastian Ewert, and Simon Dixon. Adversarial semi-supervised audio source separation applied to singing voice extraction. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2391–2395, 2018.

[71] Daniel Stoller, Sebastian Ewert, and Simon Dixon. Wave-U-Net: A multi-scale neural network for end-to-end source separation. In *International Society for Music Information Retrieval Conference (ISMIR)*, volume 19, pages 334–340, 2018.

[72] Volker Strassen. Gaussian elimination is not optimal. *Numer. Math.*, 13:354–356, 1969.

[73] Li Su and Yi-Hsuan Yang. Escaping from the abyss of manual annotation: new methodology of building polyphonic datasets for automatic music transcription. In *Proceedings of the International Symposium on Com-puter Music Multidisciplinary Research (CMMR)*, page 309–321, 2015.

[74] Sergios Theodoridis. *Academic Press Library in Signal Processing*. Academic Press, 2013.

[75] Emmanouil Benetos Simon Dixon Tian Cheng, Matthias Mauch. An attack/decay model for piano transcription. In *17th International Society for Music Information Retrieval Conference, ISMIR*, 2016.

[76] Michalis K. Titsias. Variational learning of inducing variables in sparse Gaussian processes. In *12th International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 567–574, 2009.

[77] R. E. Turner and M. Sahani. Time-frequency analysis as probabilistic inference. *IEEE Transactions on Signal Processing*, 62(23):6171–6183, Dec 2014.

[78] E. Vincent, N. Bertin, and R. Badeau. Adaptive harmonic spectral decomposition for multiple pitch estimation. *IEEE Transactions on Audio, Speech, and Language Processing*, 18(3):528–537, March 2010.

[79] Emmanuel Vincent, Rémi Gribonval, and Cédric Févotte. Performance measurement in blind audio source separation. *IEEE Transactions on Audio, Speech, and Language Processing*, 14(4):1462–1469, July 2006.

[80] K. Wang, F. Soong, and L. Xie. A pitch-aware approach to single-channel speech separation. In *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 296–300, May 2019.

[81] William J. Wilkinson, Joshua D. Reiss, and Dan Stowell. A generative model for natural sounds based on latent force modelling. In Yannick Deville, Sharon Gannot, Russell Mason, Mark D. Plumbley, and Dominic Ward, editors, *Latent Variable Analysis and Signal Separation*, pages 259–269, Cham, 2018. Springer International Publishing.

[82] Christopher K. I. Williams and Matthias Seeger. Using the nyström method to speed up kernel machines. In T. K. Leen, T. G. Dietterich, and V. Tresp, editors, *Advances in Neural Information Processing Systems 13*, pages 682–688. MIT Press, 2001.

[83] Andrew Gordon Wilson and Ryan Prescott Adams. Gaussian process kernels for pattern discovery and extrapolation. *30th International Conference on Machine Learning (ICML)*, pages 1067–1075, 2013.

[84] Andrew Gordon Wilson, Zhiting Hu, Ruslan Salakhutdinov, and Eric P Xing. Deep kernel learning. In *Artificial Intelligence and Statistics*, pages 370–378, 2016.

[85] Adrien Ycart and Emmanouil Benetos. A study on LSTM networks for polyphonic music sequence modelling. In *Proceedings of the 18th International Society for Music Information Retrieval Conference, ISMIR 2017, Suzhou, China, October 23-27, 2017*, pages 421–427, 2017.

[86] K. Yoshii and M. Goto. Infinite kernel linear prediction for joint estimation of spectral envelope and fundamental frequency. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2013.

[87] Kazuyoshi Yoshii, Ryota Tomioka, Daichi Mochihashi, and Masataka Goto. Beyond NMF: Time-domain audio source separation without phase reconstruction. In *14th International Society for Music Information Retrieval Conference (ISMIR)*, pages 369–374, 2013.

[88] C. Zhang, J. Butepage, H. Kjellstrom, and S. Mandt. Advances in variational inference. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1, 2019.