MUSIC ANALYSIS WITH A BAYESIAN MODEL

Lilia Ben Baccar
MS-DS, ENSAE Paris
lilia.benbaccar@ensae.fr

Erwan Rahis
MS-DS, ENSAE Paris
erwan.rahis@ensae.fr

January 2021

Abstract

The analysis of music has always been an interesting topic for music theorists for helping in music teaching, analysis of human perception of sounds and for design of music search and organization tools. Moreover, a typical goal is to segment a given piece with the goal of interring interrelationships among motive and themes within the music. And analyzing sequential data has been a problem in statistical modeling for several years. The goal of this project is to use Bayesian inference to analyze a music piece. We consider a music piece as a sequential data and and we want to segment the piece using a non-parametric Bayesian model. The music, to be represented as a discrete sequence of observations is processed and then modeled using a Hidden Markov Chain model. The segmentation is inferred by the model and compared to our music-theoretic analysis. Our method is mainly a combination of the papers by Lu Ren and al. [2] and Emily B. Fox and al. [1].

1 Bayesian model

To solve this problem, the approach taken is a non-parametric Bayesian model based on the *Hidden Markov Model* and a prior generated by a *Hierarchical Dirichlet Process* (HDP-HMM). We want to segment a waveform into a set of time intervals with no *a priori* on the number of intervals. The sequence is modelled using the Hidden Markov Model with the states to categorise the interval. In other words, we want the piece to be divided into intervals and each interval will be labelled to a state that we will later on analyse. For example a music piece can be divided into parts that are : intro, chorus, pre-chorus, outro. Those parts can be easily identified by the human ear and brain (music-theoretic analysis). We will compare our analysis to the results of the model.

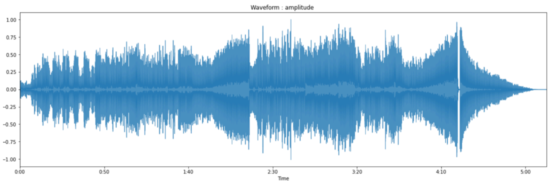

Figure 1: Audio waveform of the piece considered

1.1 Hidden Markov Model

The Hidden Markov Model used is described as follows : A discrete sequence of observed data $y = \{y_t\}_{t=1}^T$ with $y_t \in \{1, \dots, M\}$ and a corresponding hidden state sequence $s = \{s_t\}_{t=1}^T$ with $s_t \in \{1, \dots, I\}$. The states will be analysed as in Section 1 (intro, chorus, etc.). Each observation is associated to a state. The discrete HMM model is represented by its parameters $\theta = \{\pi, B, A\}$ such that :

- π is a set of state transition probabilities, the probability to transition from one state to another.For two states ρ and ξ, the transition probability from ρ to ξ is $\pi_{\rho\xi} = \mathbb{P}(s_{t+1} = \xi | s_t = \rho)$
- B is a set of emission probabilities, the probability that an observation associated to a state ρ is equal to m : $b_{\rho m} = \mathbb{P}(x_t = m | s_t = \rho)$
- A is the set of initial state distribution, the probability that the first observation in time $t = 1$ is equal to the state ρ : $a_\rho = \mathbb{P}(s_1 = \rho)$

Because we will use a Hierchical Dirichlet Process to model the parameters θ, we will divide the sequence into J subsequences. This means that we will model J Hidden Markov Models. By taking the notation at the beginning of this section, we will have $j \in \{1, \dots, J\}$ and $y = \{y_j\}_{j=1}^{J}$ with $y_j = \{y_{ji}\}_{i=1}^{N_j}$. Note that every subsequence x_j has a length N_j. We allow the parameters to be different in each subsequence such that :

$$x_j \sim HMM(\theta_j)$$

The joint distribution given θ is :

$$p(y|\theta) = \prod_{j=1}^{J} \left\{ \sum_{s_j} a_{s_{j,1}} \prod_{i=1}^{N_j - 1} \pi_{s_{j,i+1}} \prod_{i=1}^{N_j} b_{s_{j,i} y_{j,i}} \right\}$$

1.2 Hierchical Dirichlet Process

A Hierchical Dirichlet Process is defined as a collection of Dirichlet Processes. Our collection of DP processes is the J parameters θ_j linked to the HMM. This kind of setting allows for analysing how inter-related are the subsequences one to the others. The first process of the collection is denoted as :

$$G_0 \sim DP(\gamma, H)$$

with γ a concentration parameter for the Dirichlet distribution which is symmetrical and H a base probability measure on Θ. The formal definition of a Dirichlet process is $\forall (B_k)_{k=1,\dots,K}$ finite partition of Θ :

$$(G_0(B_k))_{k=1,\dots,K} | \gamma, H \sim Dir((\gamma H(B_k))_{k=1,\dots,K})$$

We approach the Dirichlet Process with a stick-breaking process and we can write G_0 as follows :

$$G_0 = \sum_{k=1}^{\infty} \beta_k \delta_{\theta_k}$$

with δ_{θ_k} an indicator function in θ_k and β_k the mixture weights computed such that :

$$\begin{aligned} \beta_k &\sim Stick(\gamma) \\ \beta_k &= v_k \prod_{l=1}^{k-1} (1 - v_l) \\ v_k &\sim Beta(1, \gamma) \end{aligned}$$

with v_k the independent random variables following a Beta distribution. The process to obtain the probabilities β_k is pictured as breaking a unit-length stick because we start off with a stick of length 1 and at each step we we break a portion of the what is remaining stick according to v_k and assign to β_k the piece broken. The smaller the α the more concentrated is the distribution. The first process G_0 (first subsequence) is then used as the base measure for the other processes (other subsequences) :

$$\begin{aligned} G_j &\sim DP(\alpha, G_O) \\ G_j &= \sum_{t=1}^{\infty} \tilde{\pi}_{jt} \delta_{\theta^{\star}_{jt}} \\ \tilde{\pi}_j | \alpha &\sim Stick(\alpha) \\ \theta^{\star}_{jt} | G_0 &\sim G_0 \\ \theta'_{ji} | G_j &\sim G_j \\ y_{ji} | \theta'_{ji} &\sim F(\theta'_{ji}) \end{aligned}$$

with $j \in \{1, \dots, J\},\ i \in \{1, \dots, N_j\}$. We know that multiple $\theta^{\star}_{jt}$ can take the same value θ_k. We can write G_j as a function of unique states :

$$\begin{aligned} G_j &\sim \sum_{k=1}^{\infty} \pi_{jk} \delta_{\theta_k} \\ \pi_j | \alpha, \beta &\sim DP(\alpha, \beta) \\ \theta_k | H &\sim H \end{aligned}$$

with π_{jk} the distribution of the subsequence j over the state k with :

$$\pi_{jk} = \sum_{t | \theta^{\star}_{jt} = \theta_k} \tilde{\pi}_{jt}$$

The model can be re written with an indicator variable z_{ji} for the state assigned to the observation y_{ji} :

$$\begin{aligned} G_j &\sim DP(\alpha, G_O) \\ z_{ji}|\pi_j &\sim \pi_j \\ y_{ji}|\{\theta_k\}, z_{ij} &\sim F(\theta_{z_{ji}}) \end{aligned}$$

With $F(\cdot)$ the fixed distribution of the data sample. According to [1], we compute a sticky version of the model by adding a κ parameter that will slow down the transitions between states and avoid having unrealistically fast dynamics as results. The new transition distributions are modelled as follows :

$$\begin{aligned} \beta|\gamma &\sim Stick(\gamma) \\ \pi_j|\alpha, \kappa, \beta &\sim DP\left(\alpha + \kappa, \frac{\alpha\beta + \kappa\delta_j}{\alpha + \kappa}\right) \end{aligned}$$

By adding this parameter, we are increasing the probability of self-transition by an amount proportional to k. If the probability that an observation has the same state as the one after is higher, the changing dynamics will slow down. That is why the κ parameter is added on the j^{th} component of $\alpha\beta$.

2 Computational method

We use the *Gibbs Block Sampler* [1] for the inference of the parameters of the sticky HDP-HMM. The first step is to sample β and π. The sampler is defined with a finite Dirichlet prior on θ_j such that :

$$\begin{aligned} \beta|\gamma &\sim Dir(\gamma/L, \ldots, \gamma/L) \\ \pi_j|\alpha, \beta &\sim Dir(\alpha\beta_1, \ldots, \alpha\beta_L) \end{aligned}$$

This finite approximation is made because when $L \to \infty$ then it converges to the HDP mixture model. The derived posterior distribution is :

$$\begin{aligned} \beta|\bar{n}, \gamma &\sim Dir(\gamma/L + \bar{m}_{\cdot 1}, \ldots, \gamma/L + \bar{m}_{\cdot L}) \\ \pi_j|z_{1:T}, \alpha, \beta &\sim Dir(\alpha\beta + n_{j1}, \ldots, \alpha\beta_j + \kappa + n_{jj}, \ldots, \alpha\beta_L + n_{jL}) \end{aligned}$$

with n_{jk} the number of j to k transitions in the state sequence j. The choice of initialisation of the base measure is made regarding the paper with :

$$\begin{aligned} y|z = k &\sim \mathcal{N}(\mu_k, \sigma_k^2) \\ \mu_k &\sim \mathcal{N}(0, s^2) \\ \sigma_k^2 &\sim InvGamma(a, b) \end{aligned} \tag{1}$$

After, choosing the hyper-parameters and base measure, the next step is to sample the hidden states z_t. This is done with a forward-backward algorithm to jointly sample z given y for the sticky HDP-HMM. The conditional distribution is :

$$p(z|y, \pi, \theta) = p(z_t|z_{t-1}, y, \pi, \theta)p(z_{t-1}|z_{t-2}, y, \pi, \theta) \ldots p(z_2|z_1 y, \pi, \theta)p(z_1|y, \pi, \theta)$$

The algorithm starts by sampling z_1 from $p(z_1|y, \pi, \theta)$ then sample z_2 from $p(z_2|z_1, y, \pi, \theta)$ and so on. The conditional distribution is :

$$\begin{aligned} p(z_1|y, \pi, \theta) &\propto p(z_1)f(y_1|\theta_{z_1}) \sum_{z_{2:T}} \prod_t p(z_t|\pi_{z_{t-1}})f(y_t|\theta_{z_t}) \\ &\propto p(z_1)f(y_1|\theta_{z_1}) \sum_{z_2} p(z_2|\pi_{z_1})f(y_2|\theta_{z_2})m_{3,2}(z_2) \\ &\propto p(z_1)f(y_1|\theta_{z_1})m_{2,1}(z_1) \end{aligned}$$

where $m_{t,t-1}(z_{t-1})$ is called the *bakward message* passed from z_t to z_{t-1} and defined by :

$$m_{t,t-1}(z_{t-1}) \propto p(y_{t:T}|z_{t-1}, \pi, \theta)$$

In the algoritm the messages are first initialized to :

$$m_{T+1,T}(k) = 1$$

and then with a backward loop for each $k \in \{1, \ldots, L\}$ we compute the messages :

$$\begin{aligned} m_{t,t-1}(k) &= \sum_{j=1}^{L} \pi_k(j)\mathcal{N}(y_t, \mu_j, \Sigma_j)m_{t+1,t}(j) \\ &= \sum_{j=1}^{L} \pi_k(j)\frac{1}{\Sigma_j} \exp\left[-\frac{1}{2}\left(\frac{y_t - \mu_j}{\Sigma_j}\right)^2\right] m_{t+1,t}(j) \end{aligned}$$

Because of (1) we have $\mathcal{N}(y_t, \mu_j, \Sigma_j) \propto \frac{1}{\Sigma_j} e^{-\frac{1}{2}\left(\frac{y_t - \mu_j}{\Sigma_j}\right)^2}$.Then we derive the general distribution for z_t which is :

$$p(z_t | z_{t-1}, y, \pi, \theta) \propto p(z_t | \pi_{z_{t-1}}) f(y_t | \theta_{z_t}) m_{t+1,t}(z_t)$$

The algorithm computes the z_t with a forward loop with for each $(j, k) \in \{1, \dots, L\}^2$ computes the probability of $y_t | z_t = k$:

$$f_k(y_t) = \pi_{z_{t-1}}(k) \mathcal{N}(y_t, \mu_k, \Sigma_k) m_{t+1,t}(j)$$

and then assigns a state z_t if the probability of having y_t conditional to the state k is inferior to the probability conditional to the state j : $f_k(y_t) < f_j(y_t)$ then the state in t is updated $z_t = j$ and we eventually update the transition matrix (N) to add one more transition.
The next step is to sample the variables corresponding to the number of observations that are assigned a state k. We have to consider the number of observations that are transitioning from state j to k and for each transition n_{jk} the probability that an observation is at time t conditional to the state k is :

$$p(t_{ij} = t | k_{jt} = k,$$

3 Data processing

We consider an acoustic signal in a mp3 format loaded in Python using the librosa package. The sampling rate is 22.05kHz which means we have 22050 observations par second of audio signal. The acoustic signal is first loaded as a waveform or amplitude as shown in Figure 1. It represents the pressure change recorded by a recording instrument such as a microphone. We then divide the piece into 100ms contiguous frames. Each amplitude frame is processed by computing its Mel frequency sceptral coefficients (MFCCs). These coefficients are derived from the Melspectrogram that is a log-scale of pitches judged to be equally distanced to one another. The Mel spectrogram is the representation showing the mel frequencies according to time and intensity. The MFCCs are the most popular feature used in audio processing. We compute 40 columns of coefficients for each frame. To obtain a vector sequential data, we quantize each frame of coefficient. First, we *whiten* the columns so each one of them have a unit variance. Indeed, before running k-means agorithm, we need to rescale each feature dimension of the observations by the standard deviation. Then we use the k-means algorithm with 16 centroids to generate to adjusts the classification of the observations into clusters and updates the clusters centroids until the position of the centroids is stable over successive iterations. This yields a code book mapping centroids to codes and vice-versa. After that, we pass them to the vector quantization algorithm which assigns codes from a code book to observations. The result is a 1D discrete sequential data representing the coefficients. After the piece is processed, the sequence is represented as a series of discrete observations. According to the paper, we should have the following figure which represents each observation by its cluster discretization.

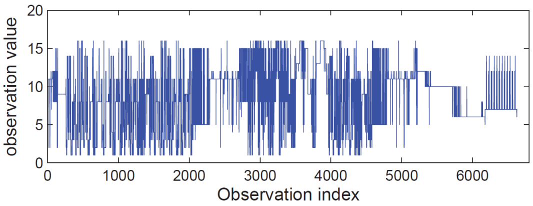

Figure 2: Sequence of code indices for the Beatle's music using a codebook of dimension $M = 16$

Unfortunately, after a lot of attempts, we've never managed to find this plot. After the discretization using k-means and Vector Quantization, we find the following plot:

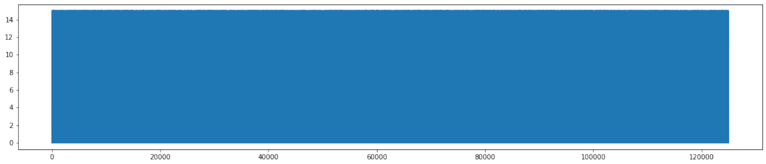

Figure 3: Our output

Indeed, the paper doesn't give enough details about the pre-processing part. We tried to use other papers from the same author, about the same topic, to see if he's giving some other details but we didn't succeed. So, we decided to use simulated data to apply our bayesian model. We found these online in the Github of a similar topic.

4 Results

References

[1] Emily B. Fox, Erik B. Sudderth, Michael I. Jordan, and Alan S. Willsky. A sticky HDP-HMM with application to speaker diarization. *The Annals of Applied Statistics*, 5(2A):1020--1056, June 2011.

[2] Lu Ren, David B. Dunson, Scott Lindroth, and Lawrence Carin. Music analysis with a Bayesian dynamic model. *IEEE*, 2009.

[3] Lu Ren, David Dunson, Scott Lindroth, and Lawrence Carin. Dynamic Nonparametric Bayesian Models for Analysis of Music. *Journal of the American Statistical Association*, 105(490):458--472, June 2010.

[4] Satyanand Singh and E.G. Rajan. Vector Quantization Approach for Speaker Recognition using MFCC and Inverted MFCC. *International Journal of Computer Applications*, 17(1):1--7, March 2011.

[5] IEEE Signal Processing Society, editor. *Proceedings / 2009 IEEE International Conference on Acoustics, Speech, and Signal Processing: April 19 - 24, 2009, Taipei International Convention Center, Taipei, Taiwan*. IEEE, Piscataway, NJ, 2009. Meeting Name: IEEE International Conference on Acoustics, Speech, and Signal Processing.