Automatic Geocoding and Dating of Music Based on Audio Content

Mauro Domingues Teles

Thesis to obtain the Master of Science Degree in

Information Systems and Computer Engineering

Supervisors: Prof. Bruno Emanuel da Graça Martins
Prof. David Martins de Matos

Examination Committee

Chairperson: Prof. Ana Teresa Correia de Freitas
Supervisor: Prof. Bruno Emanuel da Graça Martins
Member of the Committee: Prof. Thibault Langlois

October 2017

Acknowledgements

First I would like to thank professor Bruno Martins for allowing me to work alongside him, contributing with his knowledge and tireless support.

Secondly, I state my profound gratitude for Professor David Matos for enabling to learn and grow by giving me the opportunity to join INESC-ID as a rookie, I hold a deep admiration for his leadership and motivation that serve as a model and inspiration for my future self.

At last, I also thank my colleagues from L2F, INESC-ID, namely António Lopes, Francisco Raposo, Jaime Ferreira and Rodrigo Lourenço for their help along the way.

Mauro Domingues Teles

For science and progress,

Resumo

Ao longo da história, as trocas culturais têm influenciado a música a evoluir. Estudos anteriores demonstraram que estas trocas culturais estão gravadas na música como informação, possível de ser extraída e aprendida por técnicas automáticas. Neste trabalho, eu trato as tarefas de (i) prever as coordenadas geográficas onde uma música foi feita e (ii) prever o ano de lançamento de uma música. Proponho especificamente uma arquitetura de rede neuronal com dois componentes principais. O primeiro componente são unidades recorrentes, capazes de tratar as propriedades sequenciais da música. A segunda componente é a Atenção Neuronal, um mecanismo que permite focar momentos particulares de uma música, tratando o contexto de um evento musical. Nas minhas experiências uso o Million Song Dataset, uma colecção de descritores de áudio e meta-dados um milhão de músicas populares. A avaliação experimental mostra que o método proposto se comporta relativamente bem na tarefa de prever o ano de lançamento de uma música, e tem uma performance fraca na maior parte dos casos em que se prevê a origem geográfica.

Abstract

Throughout history, cultural interchanges have been a driving force for music to evolve. Previous studies have shown that these cultural traces are imprinted in music as information, accessible to be extracted and learned by automatic techniques. In this work, I address the tasks of (i) predicting the geographical coordinates where a song was made and (ii) predicting the release year of songs. I specifically propose a deep neural network architecture with two main components. The first component is recurrent units, capable of leveraging the sequential properties in music. The second component is Neural Attention, a mechanism that allows focusing on particular moments of the song, leveraging the context of a musical moment. In my experiments I use the open-source Million Song Dataset, a collection of audio descriptors and meta-data for a million popular songs. The experimental evaluation shows that the proposed method performs reasonably well in the task of predicting the release year of a song, and has poor performance in most cases for the task of predicting the geographical location.

Palavras Chave
Keywords

Palavras Chave

Aprendizagem Automática

Redes Neuronais Recorrentes

Geocodificação Automática de música

Datação Automática de música

Musicologia

Keywords

Machine Learning

Recurrent Neural Networks

Automatic Geocoding of music

Automatic Dating of music

Musicology

Contents

List of Figures

List of Tables

1 Introduction

Recently, following the adoption of digital technology, computer scientists are working closely with musicologists in researching large scale music processing tools to support and enable their studies (Abdallah et al., 2016; Duval et al., 2015; Pugin, 2015). The work by French (2017), where the author studies the geography of American rap music, is a practical example of a domain where musicologists might benefit from automated techniques to gather sociological insights.

When discussing music it is not uncommon to use adjectives such as it being classical, African or oriental. However, it is not trivial to code these subjective definitions into a machine. The present work focuses on two problems, namely (i) predicting the geographical coordinates where a song was made and (ii) predicting the release year of a song. The solution is designed following an approach where computational representations of songs are processed by algorithms based on statistics, capturing patterns in data and relating them to the property under study. I perform a set of experiments with thousands of songs, comparing the error values against simple baseline approaches.

1.1 Motivation

Songs are multidimensional pieces which may be represented by properties such as rhythm, timbre, or pitch, and these are clues that can be used to distinguish between musical categories. Automatically classifying music into categories can be addressed using machine learning techniques (Choi et al., 2016). These are computer algorithms that automatically learn to do a task from experience, in this case learning from audio or from features computed from the audio signal. This work concerns with investigating whether some of the most promising machine learning algorithms can capture the high level semantics of geographical origin and year of creation for musical pieces. In previous studies, machine learning was already applied to similar problems. For instance, Zhou et al. (2014) demonstrated the relationship between audio content and geographical coordinates, while Bertin-Mahieux et al. (2012) addressed the task of predicting the release year of a song. However, given that these previous studies used simple

algorithms or datasets with a few songs, there is an opportunity to study this task further.

1.2 Thesis Proposal

The experiments described in the present document include reproducing previous results with a baseline method, the K-Nearest Neighbors, an algorithm that leverages the overall song similarity, estimating similar labels for similar songs. Additionally, I propose a recurrent neural network architecture inspired on the success of previous works on similar tasks (Choi et al., 2016; van den Oord et al., 2016). The first part of the model consists of Gated Recurrent Units (GRU) (Cho et al., 2014), a neural unit capable of processing audio descriptors in order to capture sequential properties. The second part consists of a Neural Attention mechanism (Xu et al., 2015), a trainable layer that allows the model to focus on the most relevant moments of the song. At last, the rich representation learned by these units is passed to densely connected layers that output a single value for year regression or a class label for region classification. I specifically use the open-source Million Song Dataset (MSD) (Bertin-Mahieux et al., 2012), a collection of songs annotated with city of origin for the concerning artist and release year. The audio descriptors contained in the MSD were calculated on a segment basis, that is, the features were extracted in short time intervals and each track is represented by a sequence of descriptors.

1.3 Contributions

In brief, the main contributions of this thesis are as follows:

- The proposal of a neural network architecture for processing audio descriptors in order to predict when and where a song was made.
- The comparison of different machine learning algorithms in the task of predicting the geographical origin, a task that has not been studied with the Million Song Dataset.
- The comparison of different machine learning algorithms in the task of predicting the date of music in the Million Song Dataset.
- A study of the performance of the proposed neural network, detailed per region for the task of geo-location. The results reported in this dissertation suggest that there is no strong relationship between the audio content and coordinates in the Million Song Dataset. Nevertheless, the reported metrics serve as a baseline for this particular task and dataset.

1.4 Structure of the Document

The remainder of this document is structured as follows: Chapter 2 first presents the fundamental concepts to understand the rest of the document and then surveys previous related work. Chapter 3 presents the proposed approach, starting with an overview of recurrent neural networks and then detailing the proposed neural network architecture. Chapter 4 presents the experimental evaluation of the proposed method, detailing the dataset used, the evaluation methodology and the obtained results. Finally, Section 5 summarizes the main conclusions and presents possible directions for future work.

2 Concepts and Related Work

This chapter presents the fundamental concepts and related work. First, an overview on relevant fundamental concepts is provided in Section 2.1. This overview consists in presenting computational representations of audio (Section 2.1.1), supervised learning methods (Section 2.1.2) and useful techniques for working with geospatial data (Section 2.1.3). Then, Section 2.2 surveys previous related work on geocoding and dating music and also previous works that use modern methods for similar tasks.

2.1 Fundamental Concepts

Taking into account the goal to construct a classifier that is based on audio content, the need arises for building representations of sound that are meaningful in terms of this objective. For the task of predicting geography and temporality, it may be useful to have information about instruments, rhythm and others. A lot of work has been developed in the area of music information retrieval with respect to extracting these features from audio. Regarding the machine learning algorithms there is also a large body of work, in this chapter I focus on promising methods, that is, methods that are commonly used for similar tasks.

2.1.1 Representing Music

Even though extensive work has been done in this area, there is no set of representations that fits every problem. As illustrated in Figure 2.1, features may be extracted in different steps, they may build onto each other, and each feature has its own particularities. In this context, framing is the process of clipping a signal into smaller intervals, i.e. frames. These intervals, may be processed by a short-time Fourier transform to obtain a frequency spectrum. The spectrum itself may be divided into sub-bands, which is useful if we want to analyze particular frequencies. At last, some characteristics may have interesting periodicity that we can capture through temporal integration.

Next, I briefly review five categories of audio features, as described by Yang and Chen

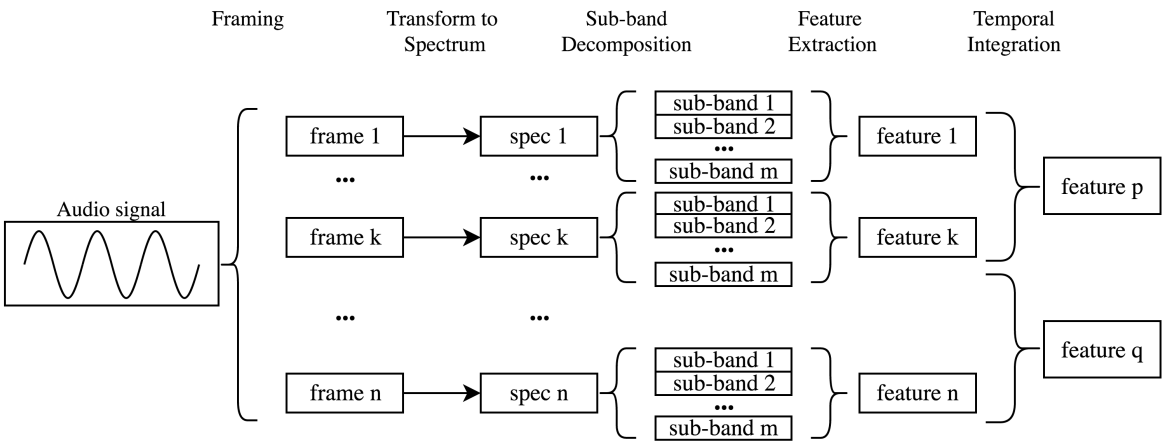


Figure 2.1: Illustration of feature extraction steps.

(2011). If the reader is interested in further details, be referred to the survey by Mitrovic et al. (2010).

Energy Features are correlated with the sensation of excitement in music. The *total loudness* (Benetos et al., 2007) is an example of this class of features that tries to account the way humans perceive sound, calculated by aggregating the specific loudness sensation coefficients, derived from the bark scale, a non-linear psycho-acoustical scale.

Rhythm Features are related to the recurrent patterns of musical events. The structure of these patterns is called beat and its tempo (i.e., pace) is measured in terms of beats per minute (BPM). While monotonous and rigid patterns may be perceived with negative emotional valence, flowing patterns are associated with positivity, and faster tempo is also perceived as being more exciting. Rhythmic content can be obtained by calculating the periodicity of dominant components derived from the spectrum, for example, through a *beat histogram* or a *rhythm histogram*. Another way to analyse rhythm is identifying when notes start. This process is also called onset detection and gives origin to features like *onset frequency*, *tempo*, *rhythm strength* and *rhythm regularity*.

Temporal Features are related to the evolution of the music signal with respect to time. The *zero-crossings rate* captures the rate at which a signal crosses the axis which may be useful for voice detection or percussive sound classification (Gouyon et al., 2000). The *temporal centroid* is the point in time where most energy of the signal is located in average. Finally, the *log attack time* describes the beginning of a sound and may be used to characterize instruments based on their onsets (Mitrovic et al., 2010).

Spectrum Features are a family of statistics which describe the shape of the power spectrum, obtained by taking the Short-Time Fourier Transform (STFT) of an audio signal, often

associated with timbre qualities. Timbre can be understood as the underlying properties that allow humans to distinguish between two different instruments playing a similar note.

The *Mel-Frequency Cepstral Coefficients* (MFCCs) offer a compact representation of sound, mostly used in speech recognition (Logan, 2000). The Mel scale is a non-linear scale of frequencies that takes the human perception of pitch into account. MFCCs can be computed by mapping frequencies to the mel scale, applying the logarithm to the power of each frequency, and then taking the discrete cosine transform, which outputs a set of coefficients. The first few dozens of coefficients are enough to represent voice/timbre characteristics, independent of speaker/source identity.

Finally, **Harmony Features** are related to pitch and to how different pitches blend together. Harmonic sounds are perceived as more pleasant than dissonant sounds. Such information can be analyzed by constructing a chromagram, by projecting the frequency spectrum onto twelve bins that correlate to the twelve semitones of the musical octave (i.e., the Western scale). The tonal centroid is an example of a feature belonging to this class and is calculated by projecting the chords along a chord progression scale (e.g., the circle of fifths).

2.1.2 Supervised Machine Learning

This section briefly reviews popular supervised machine learning algorithms, ranging from simple methods like the K-nearest neighbor approach, to more elaborate concepts such as neural networks. This base knowledge will be useful as building blocks to comprehend the related work that follows.

K-nearest neighbor (Cover and Hart, 1967) regression involves calculating the Euclidean distance, in feature space, between an input vector and each example from a set of training vectors. The output value is given by the weighted average of the K closest training instances. Similarly, K-nearest neighbour classification involves outputting the class label that corresponds to the mode class of the K-nearest training instances.

The *random k-nearest neighbor* algorithm (Li et al., 2011) shares the same principle as the k-nearest neighbor approach, but it considers only a random subset of the features from the training set, when calculating the distance between vectors. This randomness prevents some features from having too much impact on the results.

Regression tree learning is based on the idea of learning a decision tree from a dataset by recursively partitioning sets of data into subsets. At the beginning, the root node holds every

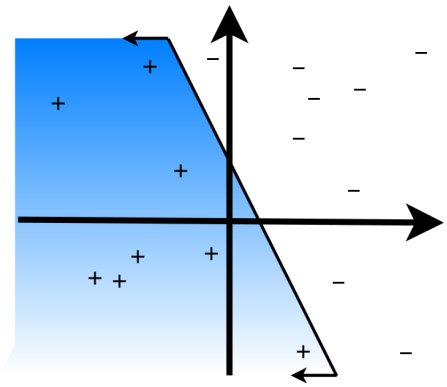


Figure 2.2: Linearly separable data.

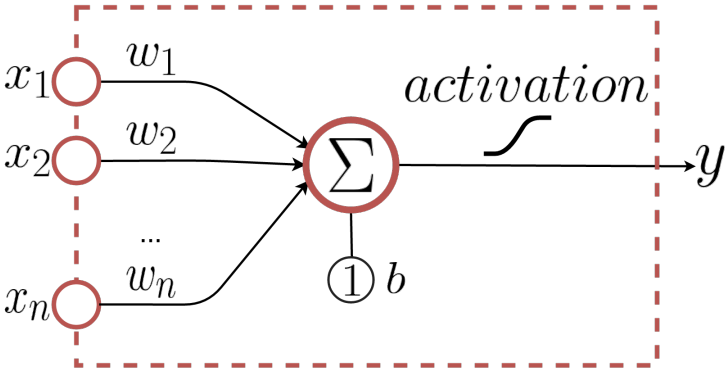


Figure 2.3: Perceptron architecture.

example from the dataset. At each step, multiple features are considered for splitting. The partitioning process is guided by heuristics that help separate examples into more homogeneous subsets, for example, selecting the feature with the highest information gain for splitting the data. At last, the leaf nodes represent the prediction for a certain path along the tree.

Bagging (Breiman, 1996) is a technique used to improve model accuracy, often used together with decision trees. The general process can be defined by two steps. The first step is random sampling the training set with replacement, termed bootstrapping. The second step is fitting a model to each set of the bootstrap datasets and then averaging the results, in the case of regression.

Random forests (Breiman, 2001) are the result of assembling multiple decision trees. First, bootstrapping techniques are used to generate multiple training sets from the collection. Then, a tree is trained for each set, using only a random subset of the features. Finally, the prediction for a given input is the aggregate prediction from each tree, i.e. the average in the case of regression.

The *perceptron* model, originally proposed by Rosenblatt (1958) and perfected by Minsky and Papert (1969), is another example of a classical supervised machine learning algorithm. Perceptrons are a fundamental concept for understanding neural networks, therefore I will now explain them in more detail. A single perceptron is capable of learning a linear binary classifier, i.e. separating the feature space into two zones – see Figure 2.2 for a 2D example. The perceptron architecture, as illustrated in Figure 2.3, includes:

- an input feature vector $x_1, x_2, ..., x_n$;
- a set of weights $w_1, w_2, ..., w_n$ that are tuned to help the model fit the data;

- a bias parameter, which can be understood as an extra input that is always set to 1, also associated to a model weight b;
- an aggregating function that sums the dot product between inputs and weights.
- an activation function (e.g., a step function that discriminates between positive and negative inputs), mapping from real values to an output class $\{0, 1\}$;
- The output y, given by $\text{sign}\left(\sum_{i=1}^{n} w_i x_i + b\right)$, or equivalently in vectorial form $\text{sign}\left(\boldsymbol{w}^T \boldsymbol{x} + b\right)$.

Training a perceptron is done by updating the weights. If an input instance is correctly classified, the weights remain the same. On the other hand, if the classification is incorrect, the weights are updated. The perceptron algorithm can be visualized as trying to fit a hyperplane in order to separate the data. The support vector machine (Cortes and Vapnik, 1995) corresponds to a perceptron of maximum margin, i.e., the hyperplane is at the maximum distance between the nearest instances of each class.

Given that a single perceptron is limited to performing linear separation, this model fails to represent some problems. The *Multi-Layer Perceptron* (MLP) is an extension of the perceptron model that does not have that limitation. A MLP is a neural network composed by multiple layers, the first of which called the input layer, and the last of which termed the output layer. The layers in-between are called hidden layers because looking at the network as a black box we can only observe the input instances and predictions. Each layer is fully connected to the following and the output of one layer is the input of the other (Figure 2.4). The activation function at each node is typically chosen to be a sigmoid function or an hyperbolic tangent function, which are continuously differentiable. The advantage of this Multilayer architecture is an increased expressiveness, as in theory a MLP is capable of approximating any function. This type of network, where the data moves only in one direction, is also called a feed-forward neural network.

Training a neural network also consists of updating weights, so that the output of the network better corresponds to the expected output. Particularly, back-propagation (Rumelhart et al., 1986) is commonly used for training, comprehending two steps: the first is propagating the input forward to obtain the network's output y, from which a prediction error is calculated. The second step is updating all the weights in the network with respect to the error in a backwards direction. This process is based on the gradient descent algorithm, which makes use of the partial derivatives of a function to adjust a model's parameters. The update of each weight is given by

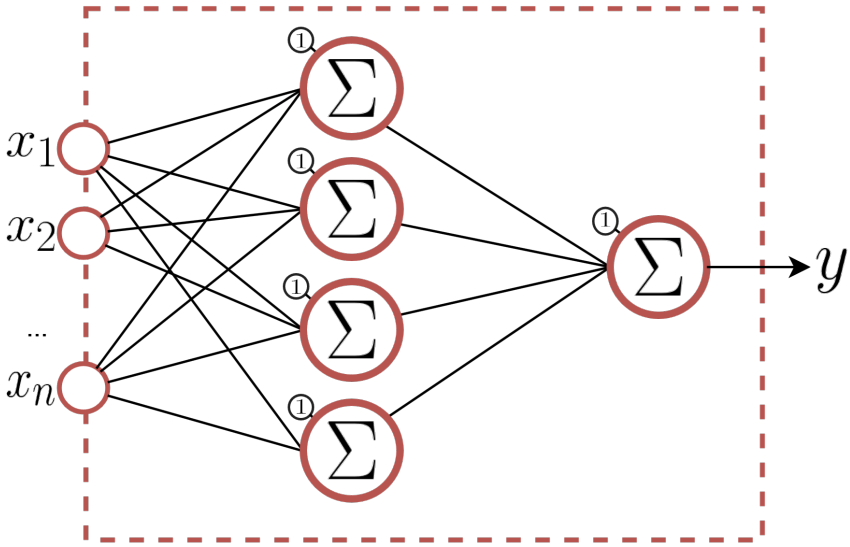


Figure 2.4: Illustration of a Multilayer Perceptron with two layers.

the partial derivative of the error function with respect to that weight. This calculation takes advantage of the layered structure of the network that can be understood as a composition of functions, from which the chain rule of differentiation can be applied, originating an incremental update that is based on the previous layer. For a detailed explanation of back-propagation refer to Haykin (1998).

Many architectures may be developed using multiple layers of perceptrons. *Convolutional neural networks* and *recurrent neural networks* are becoming increasingly popular today due to breakthroughs in areas like image processing (Krizhevsky et al., 2012) and also sound processing (van den Oord et al., 2016). Recurrent neural networks are able to process sequences of any length, contrary to the Multilayer perceptron. Convolutional neural networks can be used capture hierarchical relationships in a fixed length input sequence. Ahead in this document, Section 3 details the intricacies of recurrent neural networks, a component that integrates the present proposal.

2.1.3 Techniques for Processing Geospatial Data

As we will analyze in the next section, the task that I propose to solve has already been addressed in previous works. In these works, the authors use the distance between the predicted location and the ground truth as the evaluation measure. This distance can be calculated by the great circle distance formula. The authors have also used the mean position among various locations as a form of prediction, which can be calculated by the geographical midpoint formula. Following, I present the two aforementioned formulas and I also introduce a model for

Figure 2.5: Discretization of a spherical surface, illustrating various levels of detail.

representing the globe, which is incorporated in our proposal (Section 3).

The distance between two points on the Earth, following the shortest path along the surface, can be approximated by the *great circle distance* procedure, which assumes that the Earth's shape is a sphere. A more precise method is obtained by a special case of Vincenty's formulae for flattened spheroids (Vincenty, 2008). Considering that ϕ_1,λ_1 and ϕ_2,λ_2 are the latitude and longitude of two locations, and considering that $\Delta\phi$, $\Delta\lambda$ are their absolute differences, the distance is approximately equal to $d = \Delta\sigma \cdot 6356$, where

$$\Delta\sigma = \arctan \frac{\sqrt{(\cos\phi_2 \cdot \sin(\Delta\lambda))^2 + (\cos\phi_1 \cdot \sin\phi_2 - \sin\phi_1 \cdot \cos\phi_2 \cdot \cos(\Delta\lambda))^2}}{\sin\phi_1 \cdot \sin\phi_2 + \cos\phi_1 \cdot \cos\phi_2 \cdot \cos(\Delta\lambda))} \tag{2.1}$$

To calculate the *geographical midpoint* (Jenness, 2008) from a set of locations, we want first, for each location, to convert the latitude and longitude angular coordinates to Cartesian coordinates x, y, and z, as shown on Equation 2.2. Then, we calculate the mean values $\bar{x}$, $\bar{y}$, $\bar{z}$, and finally convert the averages back to latitude and longitude, as shown on equations 2.3 and 2.4.

$$\begin{aligned} x_i &= cos\left(lat_i \times \frac{\pi}{180}\right) \times cos\left(lon_i \times \frac{\pi}{180}\right) \\ y_i &= cos\left(lat_i \times \frac{\pi}{180}\right) \times sin\left(lon_i \times \frac{\pi}{180}\right) \\ z_i &= sin\left(lat_i \times \frac{\pi}{180}\right) \end{aligned} \tag{2.2}$$

$$lon = \arctan 2(\bar{y}, \bar{x}) \times \frac{180}{\pi} \tag{2.3}$$

$$lat = \arctan 2(\bar{z}, \sqrt{\bar{x}^2 + \bar{y}^2}) \times \frac{180}{\pi} \tag{2.4}$$

When dealing with problems where we have to process geographical data, it is necessary to adopt a coordinate system, such as latitude and longitude, or a discretization, like countries or continents. HEALPix stands for Hierarchical Equal Area isoLatitude Pixelization, and was proposed in Gorski et al. (2004) to address the need for an efficient tool to process astronomical data. As the acronym suggests, the discretization is obtained by recursively partitioning a sphere into geometrically similar bins. The result is a quadrilateral tessellation of the sphere, which is illustrated in Figure 2.5. In addition, the number of bins can be controlled through a parameter N_{side}. The HEALPix framework has been implemented in multiple languages and is available on the web for free[1].

2.2 Related Work

This section will contextualize the present work by reviewing two previous studies that address the task of predicting the geographical origin of music and another work that addresses the task of predicting the release year of songs. Following that, I review other recent works that use state-of-the-art models and techniques for geocoding multimedia objects.

2.2.1 Geocoding and Dating Music Pieces

Zhou et al. (2014) used supervised learning methods to address the task of predicting the geographical origin of a musical piece. The problem was modeled as a regression task, noting that the prediction error can correspond to the real world distance between the predicted geospatial coordinates of latitude and longitude, and the ground-truth location.

The dataset used in this experiment was composed by one 1,059 songs, and most countries contained no more than 40 examples each. The songs were manually labeled and there was a compromise between precision and practicality, as the origin of a song was assumed to be the coordinates of the city were its author lived. Timbral and chroma features were extracted and normalized. Timbral features included time zero-crossings, the spectral centroid, mel-frequency cepstral coefficients, flux, and rolloff. Chroma features comprehended 12 pitch classes per octave, considering the Western tuning. The dataset, i.e., feature representations and the ground-truth labels, is available[2] to other researchers in the area.

[1] `https://sourceforge.net/projects/healpix/`
[2] `https://archive.ics.uci.edu/ml/datasets/Geographical+Original+of+Music`

The following procedures were compared for their ability to predict the origin of a song. I enunciate only the particularities of the implementation.

- Randomly selecting K instances from the training data and calculating their geospatial midpoint.
- K nearest neighbors interpolation, which involves calculating the Euclidean distance between the vector representation of the test instance and training vectors. Then, in Cartesian coordinates, the authors calculated the geodesic midpoint of the K nearest songs.
- Random KNN, which is similar to the previous method but considering only a subset of audio features when calculating the Euclidean distance.
- Random forest regression, by growing 200 regression trees from bootstrapped sets. When partitioning nodes, eight values in each feature's range are tested, for thirteen randomly selected features. The output of each tree is the K-nearest regression of a leaf's instances, and the output of the random forest regression is the average prediction of each tree.
- Sorted feature KNN, which involves applying KNN to the subset of features that were most frequently selected as partition criteria in random forest regression.

In the case of random forest regression, to decide which partition is best, three criteria were studied: variance, standard deviation, and absolute deviation of points in a node. Each criterion involves calculating the difference between how pure a node is, before and after the partition. A node contains several data points and the notion of purity is related to their proximity. The closer the points become, the better the partition.

To evaluate the previous methods, the authors considered the mean error distance. The error distance is the great circle distance between the predicted coordinates and the ground-truth, and the authors measured the median error over the entire test dataset.

First and most importantly, by comparing the results of random selection and those obtained by other predictors, the authors concluded that there is a significant relationship between audio features and the geographical origin of music (i.e., a p-value <0.001 for the random hypothesis). Second, random forest regression, using variance and standard deviation as partitioning criteria, results in the most accurate mean prediction of 3,113 km. Third, random forest regression performed better than nearest neighbor interpolation for every partitioning criteria. The authors suggest that irrelevant features explain the relatively poor performance of the nearest neighbor interpolation model. This was confirmed by using random KNN, which improved the results

comparatively to standard nearest neighbor interpolation. Even though the sorted feature KNN was more accurate than standard and random KNN, random forest regression maintained the best results. Finally, the authors also concluded that using chromatic features was not significant in these tests.

Following up on the aforementioned previous study, Schedl and Zhou (2016) investigated how to improve predictions for the geographical origin of a musical piece. They proposed combining methods based on audio content with methods based on information automatically collected from the web.

Regarding audio-based prediction, they studied three sets of audio descriptors: (1) block-level features, which include spectral pattern, delta spectral pattern, variance delta spectral pattern, logarithmic fluctuation pattern, correlation pattern, and spectral contrast pattern, (2) timbral features, and (3) timbral features with added chromatic. The authors then used k-nearest neighbor regression to compare the listed sets of features. For computing similarities between block-level feature vectors, the inverse Manhattan distance was used and the result was then normalized and combined. For timbral and chromatic feature similarity, the Euclidean distance was applied. The predictions were obtained by calculating the geodesic midpoint of the K nearest examples, and the error distance to the ground truth was computed through the great circle distance.

The evaluation setup and dataset were the same as in Zhou et al. (2014). In this context, block-level features outperformed the other features for every value of K, achieving the smallest mean error of $2,191$ km for $K = 1$. The second smallest prediction mean error was $3,410$ km when using timbral features.

Regarding the predictions with web content, multiple queries were made to a search engine to obtain useful information about a musical piece. The pages were then processed, seeking information about frequent country names which may be related to the origin of the song. The authors also proposed a method which aggregated both predictions by linearly combining scores.

The best result obtained by the web-based approach was a mean error of $2,748$ km and the best result for the hybrid approach was $1,1824$ km. When building a confusion matrix, it was evident that the model had difficulty identifying Japanese music.

Bertin-Mahieux et al. (2012) introduced the Million Song Dataset, providing researchers in the music information retrieval field with a large scale dataset, something that was not easily accessible up to that point. In order to demonstrate what could be done with this dataset, the authors studied predicting the release year of songs. Overall, the subset used in the experiments

contained 515,576 songs, from 28,223 artists, and, even though the data is highly skewed towards the year 2007, the problem is alleviated by the large amount of information. The features extracted to represent the audio were the average and covariance of timbre vectors. The authors used two methods to predict the release year of a song. The first method was the K-nearest neighbor approach, in which the resulting prediction is the average year of the K closest. The second method was linear regression, similar to the perceptron algorithm. The methods were compared in terms of the differences between the prediction and the ground truth year. The best performance was showed by the linear regression, with an absolute difference of 6.14 years, followed by KNN, which achieved 7.58 years in the same measure, for $K = 50$.

2.2.2 Deep Learning Methods for Geocoding or Dating Multimedia Objects

There is an emerging body of work relying on neural networks to model multimedia artifacts. In the rest of this section, I briefly review two works that use neural networks to model audio content and then I review several works where the authors perform geo-location using neural networks, highlighting the proposed models and relevant mechanisms used for geographical estimation.

Recently, Choi et al. (2016) reported having good results on the task of predicting high-level tags (e.g., genre or mood), using an hybrid network that consists of a convolutional network added with a recurrent network. The main idea behind this combined architecture is that the CNN works as a local feature extractor, while the RNN does temporal summarization. The implementation of this model, in keras, is available on the web[3]. In this work, the *Convolutional Recurrent Neural Network* (CRNN) is compared with three other CNN architectures. What distinguishes these three networks is the shape of the convolutional kernel and the convolution dimension, 1D or 2D. Following, *kn* refers to the kernel dimension and *cn* refers to convolution dimension. The first CNN architecture, named k1c2, convolves frequency bands individually, originating feature maps that have the same frequency range, but are simplified in the time domain. The second CNN architecture compresses the whole frequency range into one bin and convolves only in the time domain. Lastly, k2c2 and CRNN convolve both frequency and time domains. Each considered network contains four to five convolutional layers, which alternate with max-pooling layers. The resulting feature map is posteriorly fed to fully connected layers, responsible for outputting the predicted class. In the case of the CRNN architecture, the last convolutional layer connects to a recurrent neural network with gated recurrent units. The

[3] `https://github.com/keunwoochoi/music-auto_tagging-keras`

authors compared the four aforementioned architectures in the task of predicting the top-50 song tags from a subset of the Million Song Dataset. The subset contained 214.284 songs, which were trimmed to 29 second segments, down-sampled to 12 KHz, and with hop-size equal to 256 samples. The input of each network consisted of 96 log-amplitude mel-spectrogram bins. AUX-ROC charts were used to measure the performance. The first experiment was set to compare the networks' performance while controlling the memory used. For the same number of parameters, the authors report that the CRNN model outperforms the alternatives in all cases, achieving 1% increased AUC-ROC when compared to k2c2. On the other hand, both the k1c2 and the k2c1 models do not take advantage of the local frequency relationships, which results in worse performance. The objective of the second experiment was to compare the computational complexity of the four architectures, by measuring the training times. The k2c1 model trained the fastest, partly because it greatly reduces the feature map in the first convolutional layer. On the other end, the CRNN was the slowest to train, ranging from 100 to 400 seconds, double of the others. Given that the CRNN has a recurrent structure, the depth is considerably larger than the other networks, which results in slower training. In the third and final experiment, the authors concluded that the CRNN performs better than k2c1 and k1c2 for almost all tags. The authors conclude that the k2c2 and CRNN models have the best performance, and while CRNN can outperform k2c2 for the same number of parameters, it takes longer to train.

Weninger and Eyben (2014) proposed using LSTMs to perform continuous-time regression of emotional valence and arousal in music. The features used to represent audio content were derived by applying statistics like the standard deviation and percentiles, to low level features such as chroma, MFCCs, or energy. The calculated statistics were then aggregated over windows of one second to match the dataset annotations. In total, the feature set contained $4,777$ features. The LSTM model was compared to feed-forward neural networks (FFNs) and support vector regression. These models were trained with the same input features, which were normalized and added with Gaussian noise. With regard to the LSTM, learning the parameters was done by minimizing the average sum of squared errors per sequence, and as means to improve the model, the authors tested using pre-training of the first hidden layer, and also multitask learning, i.e. predicting both valence and arousal at the same time. The dataset contained 1,000 songs, balanced by genre. Three measures were considered to evaluate the predictions: the determination coefficient, the mean linear error, and the average Kendall's Tau per song. While training, the sequences of input were presented in random order. The first experiment consisted in comparing the three mentioned models in the task of predicting the arousal and valence independently. The authors also considered stacking two feature arrays for FFNs, to

provide context of the previous frame. The LSTM performed better with respect to all measures, particularly the average Kendall's, which indicates the superior performance of this algorithm for single target regression. In the second experiment, the authors studied adding future and past targets to the regression formula of the LSTM model. While predicting the valence was significantly improved, the performance for predicting the arousal was lower. Finally, the last experiment was set to test different network topologies with four training targets, i.e., arousal, valence and past/future targets. Networks with one hidden layer were compared to networks with two hidden layers, with and without pre-training. The authors concluded that a single hidden layer is inferior to two hidden layers for the task at hand, and they also stated that pre-training did not hold significant improvements. The best results were obtained by the LSTM architecture with two hidden layers and 384 units.

Weyand et al. (2016) proposed using the increasingly popular neural networks to solve the task of photo geolocation. Assigning a place to an image is a very challenging problem. Even though we can search for cues, like landmarks, they can be scarce and are often ambiguous. Two important aspects about the solution that was proposed are: the representation of the earth's surface, and the odd CNN architecture. To discretize the earth's surface, the authors used a method that divides the world into squared cells. The more images there are for a certain location, the smaller the covered surface, i.e., a model will be more precise for more popular areas. In the end, $26,263$ cells contained 126M photos. The CNN used in this work (PlaNet) was based on the Inception architecture (Szegedy et al., 2015), in which a convolutional layer performs not only a simple convolution, but different convolutions and pooling at the same time. The CNN outputs the probability of an image belonging to a certain cell. The proposed model took two and a half moths to train (on 200 CPU cores), but is able to localize 10.1% of the images at city-level accuracy and 28.4% at country level. In a second set of experiments, the authors explored the idea that an album of photos, i.e. a sequence of photos, encodes common geographical information. In this manner, when the model is uncertain about where a certain picture was taken, it can use other pictures in the same sequence to improve the prediction. Practically, this was done by feeding the output vector of the CNN architecture into an LSTM unit. With regard to training the networks, both were trained separately, first the CNN and then the LSTM. The CNN/LSTM was compared to the CNN from the first experiment, using a dataset containing 29.7M albums and 616M images. In the end, the combined CNN/LSTM model is able to localize 45.6% of the images at city level, and 79.3% at country level.

Also very recently, Iso et al. (2017) proposed a different approach to estimate the location of places named in tweets. The core concept consists of modeling the target as a probability

distribution, casting the problem as a density estimation. The authors use a Convolutional Neural Network to process the text and generate an hidden representation, outputting the parameters of a Gaussian mixture model. The proposed Convolutional Mixture Density Network (CMDN) is flexible in terms of being able to represent more rich information (e.g., expressing ambiguity) in contrast to a single regression value. The evaluation was based on the mean and median error distances between the predicted location (the mode value of estimated density) and the ground-truth location. In a test with 4,633,478 Japanese tweets, the CMDN model achieved the second lowest mean error of 159.4 km and the lowest median value of 10.7 km.

While some works were already dedicated to the task of predicting the geographical origin and dating of songs, they use small datasets or simple machine learning methods. As reviewed in this section, recent works with multimedia objects rely on neural networks to model the audio content. The present work builds on ideas surveyed in this section, namely recurrent neural networks for processing sequences, the attention mechanism and the adaptive discretization of the globe.

3 The Proposed Approach

Taking inspiration on previous works with multimedia objects (Weyand et al., 2016; Vo et al., 2017; Iso et al., 2017), and particularly music (Weninger and Eyben, 2014; Choi et al., 2016), I propose a recurrent neural network for processing sequential data, complemented with an attention mechanism. Figure 3.1 illustrates the proposed deep learning model composed by Gated Recurrent Units, a recurrent neural network detailed in the following Section.

3.1 Deep Neural Networks for Processing Sequential Data

The main computational unit of *Recurrent Neural Networks* (RNNs) is the recurrent neuron. The recurrent neuron is similar to the regular perceptron (Rosenblatt, 1958), complemented with a feed-back loop which causes the output of the neuron at time step $t+1$ to depend on the output at time step t. Recurrent Neural Networks are context sensitive and can be used to model input sequences of arbitrary length. Since the classic RNNs suffer from a problem of vanishing gradient, the *Long-Short Term Memory networks* (Hochreiter and Schmidhuber, 1997) and *Gated Recurrent Units* (Cho et al., 2014) were proposed to model long sequences.

The main concept behind *Gated Recurrent Units* (GRU) is the gate. Gates are sets of trainable weights that can be combined with the input of time step t and the output of the previous time step $(t-1)$. A GRU has two gates, an update gate r, and a reset gate z. The update gate (z_t) defines which parts of the previous output will carry to the current state. The reset gate (r_t) can clear the context, so that the state is reset with the current input only. The

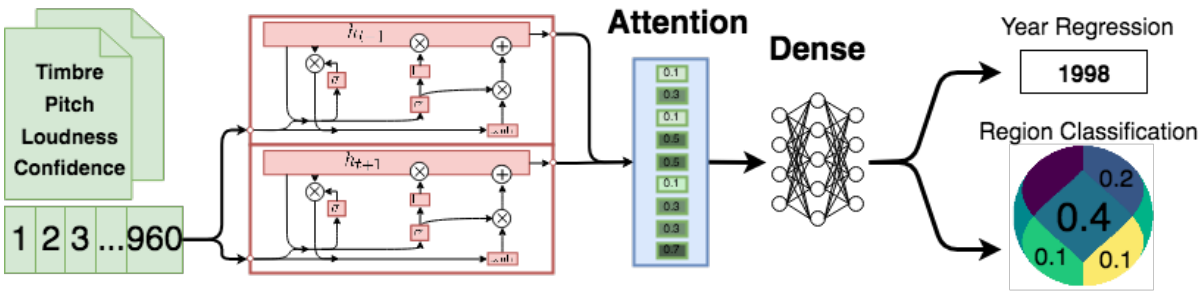
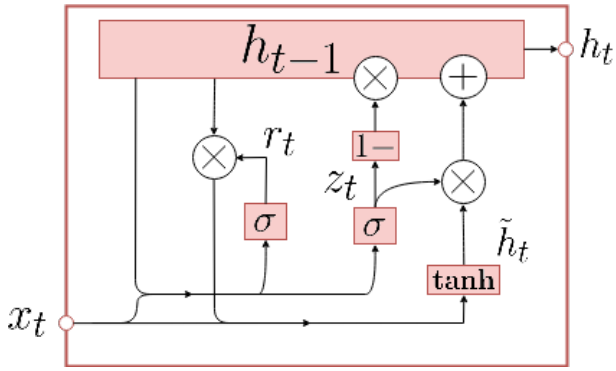


Figure 3.1: Representation of the proposed architecture.

output (h_t) is a combination on the mentioned gates.



$$z_t = \sigma\left(W_z \cdot [h_{t-1}, x_t]\right) \tag{3.1}$$

$$r_t = \sigma\left(W_r \cdot [h_{t-1}, x_t]\right) \tag{3.2}$$

$$\tilde{h}_t = \tanh\left(W \cdot [r_t * h_{t-1}, x_t]\right) \tag{3.3}$$

$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t \tag{3.4}$$

Figure 3.2: Representation of the Gated Recurrent Unit.

3.2 A Deep Neural Network for Geocoding or Dating Music Pieces

The proposed model uses two Gated Recurrent Units to generate an intermediate hidden representation of the sequential audio features. This unit allows the network to leverage the temporal structure, being able to model rhythmic patterns or particular chord progressions.

RNNs are capable of leveraging information from previous time steps, however, information from forward steps may also be important to model a context, this can be achieved with a bidirectional (Schuster et al., 1997) mechanism, a stack of RNNs, each processing the same input sequence in opposite directions.

The GRUs are connected to an Attention (Xu et al., 2015) layer. This mechanism allows a network to attend parts of an input sequence with more or less attention, effectively scoring what is most important for the prediction. A Recurrent Neural Network processes input vectors of length L, $x = (x_1, x_2, ..., x_{L-1}, x_L)$, into hidden representations $h = (h_1, h_2, ..., h_{L-1}, h_L)$. At each time step t, the output y of the attention layer, shown in equation 3.5, is generated by weighting the segments of the hidden state. The attention weights are trained similarly to a Multilayer Perceptron and then normalized to sum to 1. See equations 3.6 and 3.7 where h is the input, w are the trainable weights and b is the bias parameter. At last, the result of the Attention layer is connected to dense layers, responsible for outputting the estimated target.

$$y_t = \sum_{j=1}^{L} h_{t,j} a_{t,j} \tag{3.5}$$

$$a_{t,j} = \frac{\exp(e_{t,j})}{\sum_{j=1}^{L} \exp(e_{t,j})} \tag{3.6}$$

$$e_{t,j} = \tanh\left(h \cdot w + b\right) \tag{3.7}$$

The proposed network is trained from end-to-end. In the case of estimating the geographical origin, the very last dense layer outputs two regression values, latitude and longitude, or the most likely cell, for region classification. In the first case (coordinate estimation), the network is trained with Vincenty's formulae (equation 2.1), a differentiable function for calculating the distance between two points on the Earth, following the shortest path along the surface. In the case of region classification, the network is trained with categorical-crossentropy. In order to estimate the year of release, the only adaptation needed is to change the last dense layer to predict a single regression value and train the network with the loss function mean-squared-error. The model is implemented with Keras[1] and Theano[2], using ADAM for learning rate control.
I compare the proposed network to a shallow algorithm used in previous works (Zhou et al., 2014; Schedl and Zhou, 2016), the K-Nearest Neighbor (KNN), a method that I will use as baseline for comparison. The KNN can infer when and where a song was made based on the overall track similarity. In order to perform regression with the KNN method, we calculate the distance between the sample to predict and every other sample in the dataset. Then, considering the k closest, we aggregate their labels: in the case of predicting the year we simply calculate the mean year; in the case of coordinate regression, we can use the Geographical Midpoint Formula (see equations 2.2 to 2.4).

3.3 Overview

This chapter presented recurrent neural networks, a robust machine learning method that is capable of processing sequential data. The proposed architecture starts with the features extracted from the audio signal, generating an intermediate representation that takes into account what comes before and after each moment. The hidden representation is fed to the Attention layer that highlights the most important parts of the sequence. The result is then passed to dense layers that output the target values, e.g., a region or a year. The proposed approach does not include novel techniques, it is instead an aggregation of units that worked well in previous works on similar tasks.

[1] `https://github.com/fchollet/keras`
[2] `https://github.com/Theano/Theano`

4 Experimental Evaluation

This section reports the experimental evaluation of the aforementioned machine learning algorithms in the tasks of predicting the release year and geographical origin of songs. The experiments performed are as follows:

- Predicting the release year of a song with:
  - The K-nearest neighbor method;
  - The proposed neural network model.
- Predicting the geographical origin of a song with:
  - The K-nearest neighbor method;
  - The proposed neural network model for latitude and longitude regression.
  - The proposed neural network model for region classification.

Section 4.1 presents a characterization of the datasets used in the experiments. Section 4.2 presents the metrics considered for evaluation and Section 4.3 elaborates on the obtained results.

4.1 The Million Song Dataset

The Million Song Dataset[1] (MSD) is an open-source collection of features and meta-data for a million contemporary popular songs. This dataset contains the following groups of descriptors:

- Time marks of beats, bars and tatums, which mark the rhythm of a song and are multiple of each other.
- Segment descriptors. A segment is defined as set of sound entities (typically under a second), each relatively uniform in timbre and harmony. Each segment is described by timbre, pitch, loudness and confidence.
- Estimated key, tempo, mode and time signature for each song.

[1] `https://labrosa.ee.columbia.edu/millionsong/`

- Miscellaneous information about each song such as title, duration and year of release.
- Miscellaneous information about each artist such as latitude, longitude and MusicBrainz[2] tags created by human users.

For additional information refer to the Echonest Analyzer Documentation.

4.1.1 Features

From the set of audio descriptors available, features extracted at segment level are particularly interesting given their sequential nature. For the rest of this document, the features considered in all experiments are the following:

- Segments timbre: 12 audio texture features (similar to MFCC), useful to distinguish between instruments.
- Segments pitches: 12 chromatic features, one value per semitone.
- Segments loudness maximum: maximum dB value, can be used to analyze how calm or busy a song is.
- Segments confidence: is a measure of how reliable the features extracted are.

For the experiments with neural networks, each track is trimmed to the mean segment length and padded with zeros, resulting in a sequence of 960 time steps, with 26 features each. In order to use sequential features with the KNN method, these features are aggregated in time, obtaining the mean and variance of each feature, resulting in a vector of 52 features per track.

4.1.2 Targets and Subsets

For the task of predicting the geographical origin of a song, I use the latitude and longitude of the location where the artist is associated. We should take into consideration that this target does not take into account if an artist lived in multiple countries. For the task of predicting the date of a song I simply use the release year.
Not every song in the Million Song Dataset contains information about the release year, and not every artist is tagged with latitude and longitude, therefore, I created two different subsets, the Geographical Subset with 396,070 examples (Table 4.1 and Figure 4.1) and the Year Subset (Figure 4.2) with 208,503 examples.

[2] `https://musicbrainz.org/`

Table 4.1: Distribution of songs per continents in the Geographical Subset.

	Africa	Asia	Europe	North America	Oceania	South America
Track Count	2669	5565	105588	270114	6076	6058

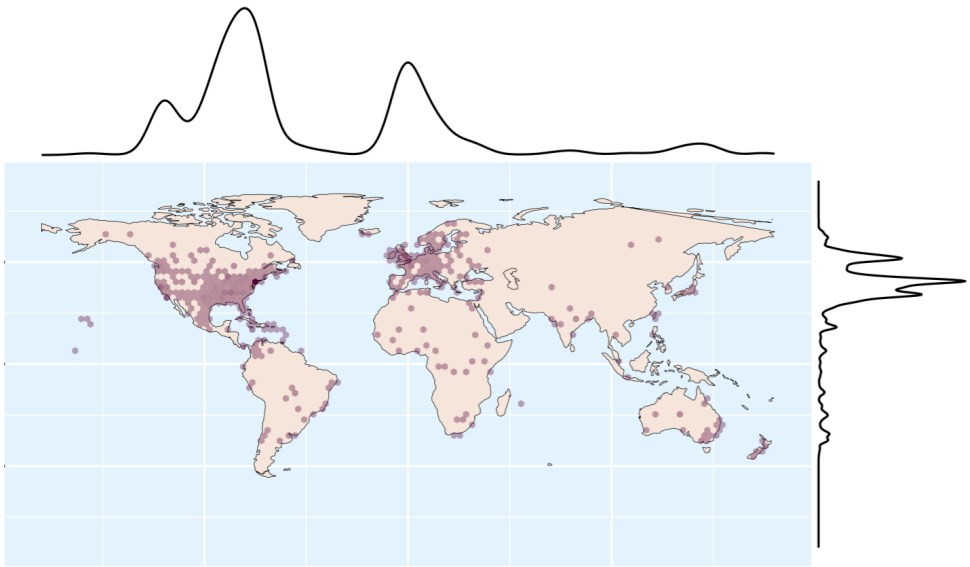


Figure 4.1: Distribution of tracks in the Geographical Subset.

For the tasks at hand, an ideal collection of music would contain many examples from multiple regions and eras. By observing the distribution of songs in the mentioned subsets it is noticeable that they are unbalanced, nevertheless, there are at least a thousand songs per continent. In this manner, we must keep in mind that we are dealing with a dataset that contains mostly Western contemporary popular music.

Since countries and continents vary in size and may have a very different sample count, and drawing inspiration from previous works (Weyand et al., 2016; Hays and Efros, 2008; Vo et al., 2017), I use an adaptive discretization mechanism. Recurring to the Healpy[3] library, I convert latitude and longitude to HEALPix areas. Every song is initially attributed to the respective lowest resolution bin, then, if a bin has more than 5,000 songs, it is split into four lower resolution areas, and the process is repeated three times. Figure 4.3 shows the partitioning of the Geographical Subset where each point represents a song and each color represents a bin.

The Geographical and Year subsets are divided into training and test splits with a ratio of 75%/25%, using stratified sampling based on the continent.

[3] `https://github.com/healpy/healpy`

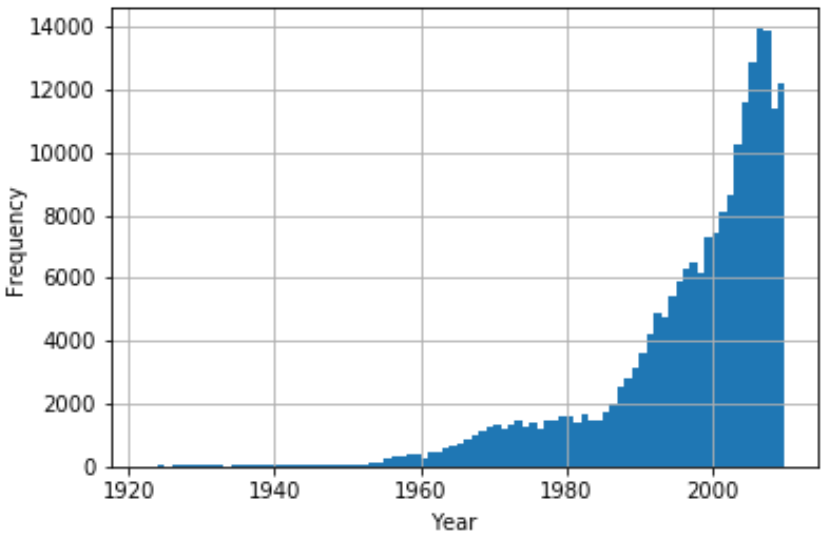


Figure 4.2: Frequency of songs per year in the Year Subset.

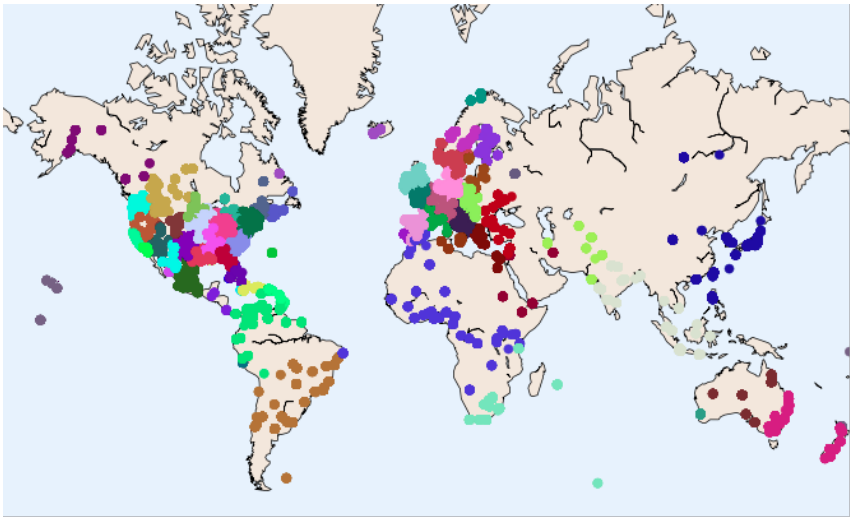


Figure 4.3: Adaptive areas based on the number of samples.

4.2 Evaluation Metrics

For assessing the overall quality of the stated models I measure the mean (see equation 4.1) and median absolute errors, in years and kilometers, according to the respective task.

$$MAE = \frac{\sum_{i=1}^{n} |y_i - x_i|}{n} \tag{4.1}$$

Furthermore, for the model that achieves the best performance, I detail the proportion of samples that is predicted within various intervals of error, enabling the study of the performance per area or eras.

Table 4.2: Performance metrics in the task of predicting the Release Year.

	Mean	Median	Median Confidence Interval at 95%
KNN	8.84 ± 0.6 years	7.11 years	[5.18, 9.14] years
Neural Network	7.02 ± 0.6 years	4.69 years	[3.22, 6.71] years

Table 4.3: Intervals of error per decade for the neural network model.

	Absolute Year Difference per Interval				
Decade	<5 years	[5, 10] years	[10, 20] years	>20 years	Support
1920-1960	0%	0%	0%	100%	309
1960s	0%	0%	0.33%	0.67%	860
1970s	0.05%	0.17%	0.57%	0.21%	1671
1980s	0.53%	0.25%	0.21%	0.01%	2388
1990s	0.59%	0.33%	0.07%	0.01%	6602
2000s	0.67%	0.17%	0.15%	0.01%	12819
2010s	0.33%	0.44%	0.19%	0.04%	345

4.3 Obtained Results

In this section I first present the results of the task of predicting the release year, followed by the results of the task of predicting the geographical origin.

4.3.1 Predicting the Release Year

Table 4.2 shows the evaluation measures for both the K-Nearest Neighbors method and the proposed neural network architecture. While the KNN achieves a mean absolute error of 8.84 years for $n = 50$, the recurrent neural network augmented with an Attention mechanism achieves a mean absolute error of 7.02 years, significantly outperforming the baseline.

Looking at Table 4.3, we can observe with more detail the prediction error per decade. The results are presented in the form of intervals of error. We can observe that songs from 1920 until 1960 are inaccurately predicted, while the model performs best for the music originated in the 80s, 90s and the first decade of 2000, predicting more than 50% of songs in a 10 year range (5 years before and after).

4.3.2 Predicting the Geographical Origin

The evaluation is performed by calculating the mean error distance between the ground truth coordinates and (i) the predicted coordinates, for regression, and (ii) the centroid of the estimated HealPix bin, for classification. Table 4.4 shows the results obtained.

The lowest mean error distance obtained with the KNN method is 3317 km for $n = 80$ neighbors.

Table 4.4: Performance metrics in the task of predicting the geographical origin.

Method	Mean	Median	Confidence Interval at 95 %
KNN	3317± 28 km	2653 km	[1785, 3810] km
Neural Network Regression	3810 ± 32 km	2377 km	[1683, 4123] km
Neural Network Classification	3203 ± 19 km	2106 km	[1097, 3537] km

By comparing the original distribution of songs and the distribution of predicted coordinates, I find that the KNN method does not effectively learn to geo-locate music, the reason being that predictions are spread around a center point located at the highest frequency region (i.e., the east coast of the U.S.A.).

The mean error distance obtained with the proposed neural network architecture in the task of coordinate regression is 3810 km. With this setup, I also observe that the model tends to predict most songs around the highest frequency region, similarly to the results obtained with the KNN model.

Since most songs are originated from the East coast of the U.S.A and there is a large distance from that point to Europe or Asia, I use the same architecture in the task of region classification, circumventing the effect of predicting songs in the ocean.

The mean error distance obtained with the neural network architecture in the task of region classification is 3203 km. Table 4.5 allows a view of the classifier performance per continent. We can observe that the model is very inaccurate for regions outside North America, for example only 5% of songs from Europe are predicted within a 5,000 km radius. Table 4.5 also presents the error intervals for some representative countries, including some states from the United States of America. Southeastern American states like Louisiana and Mississippi have a considerable amount of correctly predicted songs (<500 km), the rest being confused with other American regions (estimated within a 2000 km radius). On the other hand, for Western states like California, most songs are predicted in a 5000 km radius, an indicator that the model is not able to separate music from this region.

Regarding other countries, Canadian songs are confused as being from the U.S.A. given the geographical proximity; European countries like Finland, Great Britain and Poland have around 10% of songs predicted in a reasonably close distance (<2,000 km); for countries other than the U.S.A, Mexico, Cuba, and particularly Jamaica present the best results, with a sizable portion of songs being placed within a 1000 km radius.

4.4 Overview

In this chapter I detailed the set of features used to represent the audio. The chosen features allow the model to leverage which instruments are present and which notes are being played. The recurrent network enables modeling the progression of the song, highlighting the relevant patterns related with geography and date. Both targets are unbalanced, however, the model is able to perform better in the task of predicting the release year rather than predicting the geographical origin. Considering the obtained results, we can argue that the place associated to the artist is not a good indicator of geographical influence since the model is not able to separate music effectively.

Table 4.5: Intervals of error distance per country

	Error Distance in km						
Region	>500	[500, 1000]	[1000, 2000]	[2000, 5000]	[5000, 10000]	>10000	Support
Africa	0%	0.01%	0%	0.01%	45.6%	52.2%	662
Asia	0%	1.7%	0%	1.2%	19.3%	77.6%	1390
Europe	0.4%	1%	3%	0.6%	94.6%	0.1%	26405
North America	30.5%	20.6%	18.7%	28%	18%	0.1%	67510
Oceania	0%	0%	0%	0%	0.6%	99.3%	1519
South America	0%	0.1%	2.6%	22%	72.6%	1.9%	1529
Canada	29%	30%	12%	27%	2%	0%	3049
Jamaica	43%	0%	5%	51%	1%	0%	978
Mexico	18%	3%	3%	72%	2%	3%	616
Cuba	8%	2%	10%	77%	3%	0%	330
Finland	4%	6%	0%	1%	88%	0%	473
Great Britain	0%	0%	4%	0%	96%	1%	13165
Poland	1%	4%	7%	0%	87%	1%	204
U.S.A	31%	21%	20%	27%	2%	0%	62040
Alabama	13%	12%	74%	0%	1%	0%	649
Arkansas	5%	31%	62%	0%	1%	0%	766
California	0%	0%	0%	98%	1%	0%	10631
Louisiana	34%	0%	66%	0%	1%	0%	1513
Mississippi	45%	4%	51%	0%	0%	0%	1818

5 Conclusions and Future Work

In this document, I proposed a deep learning method for automatically estimating the year and geographical origin of songs. In the case of estimating the release year of a song, the results can be compared to the previous work by Bertin-Mahieux et al. (2012), even though I use a different version of the MSD, with fewer songs. I obtained a mean absolute error of 7.02 years in a test with 208,503 samples while they achieved a mean absolute error of 6.14 years in a a test with 515,576 samples. In this manner I conclude that there is no particular advantage in using recurrent neural networks to leverage audio descriptors like timbre pitch and loudness.
Regarding the problem of predicting the geographical origin of songs we can not directly compare the results to the previous works by Zhou et al. (2014) and Schedl and Zhou (2016) since these works use much smaller collections. The performance of the proposed classifier is very limited in terms of accuracy, highlighting that Jamaican and Mexican music are easier to separate from the rest. The lacking performance can be due to one of two reasons: either the features do not serve for the task at hand or there is no significant relation between the audio content and the place where the artist is associated.

5.1 Contributions

The most important contributions of my M.Sc. thesis are as follows:

- The proposal of a neural network architecture with recurrent units and neural Attention for processing audio descriptors in order to predict when and where a song was made.

- The comparison of the KNN algorithm against the proposed neural network model in the task of predicting the geographical origin, a task that has not been studied with the Million Song Dataset.

- The comparison of the KNN algorithm against the proposed neural network model in the task of predicting the date of music in the Million Song Dataset.

- A study of the performance of the proposed neural network, detailed per region for the task of geo-location. The results reported in this dissertation suggest that there is no

strong relationship between the audio content and coordinates in the Million Song Dataset. Nevertheless, the reported metrics serve as a baseline for this particular task and dataset.

5.2 Future Work

For future work, following inspiration from recent works like Choi et al. (2016), I envision using convolutional neural networks to derive an intermediate representation from raw audio, instead of relying on feature extractors. Recently, a tool was published that allows to define a Keras layer to derive the log-amplitude Mel spectrograms from raw audio. Kapre [1] is a very convenient way to feed a collection of songs to a convolutional neural network. Additionally, I propose formulating the problem as a density estimation, in the same manner as the work by Iso et al. (2017). Finally, instead of using only one place where the artist lived, we could automatically gather places associated to the artist, coming closer to model a geographical influence rather than a place where the artist was born.

[1] `https://github.com/keunwoochoi/kapre`

Bibliography

Abdallah, S., Benetos, E., Gold, N., Hargreaves, S., Weyde, T., and Wolff, D. (2016). The digital music lab: A big data infrastructure for digital musicology. *Journal on Computing and Cultural Heritage*.

Benetos, E., Kotti, M., and Kotropoulos, C. (2007). Large scale musical instrument identification. *Proceedings of the Sound and Music Computing Conferece*.

Bertin-Mahieux, T., Ellis, D. P., Whitman, B., and Lamere, P. (2012). The million song dataset. *Proceedings of the Conference of the International Society for Music Information Retrieval*.

Breiman, L. (1996). Bagging Predictors. *Machine Learning*, 24.

Breiman, L. (2001). Random Forests. *Machine Learning*, 45.

Cho, K., van Merrienboer, B., Gülçehre, Ç., Bougares, F., Schwenk, H., and Bengio, Y. (2014). Learning phrase representations using rnn encoder-decoder for statistical machine translation. *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

Choi, K., Fazekas, G., Sandler, M., and Cho, K. (2016). Convolutional Recurrent Neural Networks for Music Classification. *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*.

Cortes, C. and Vapnik, V. (1995). Support-vector networks. *Machine Learning*, 20.

Cover, T. and Hart, P. (1967). Nearest neighbor pattern classification. *Transactions on Information Theory*, 13.

Duval, E., van Berchum, M., Jentzsch, A., Chico, G. A. P., and Drakos, A. (2015). Musicology of early music with europeana tools and services. *Proceedings of the Conference of the International Society for Music Information Retrieval*.

French, K. (2017). Geography of American rap: rap diffusion and rap centers. *GeoJournal*, 82.

Gorski, K. M., Hivon, E., Banday, A. J., Wandelt, B. D., Hansen, F. K., Reinecke, M., and Bartelman, M. (2004). HEALPix – a Framework for High Resolution Discretization, and Fast Analysis of Data Distributed on the Sphere. *The Astrophysical Journal*, 622.

Gouyon, F., Pachet, F., and Delerue, O. (2000). On the Use of Zero-Crossing Rate for an Application of Classification of Percussive Sounds. *Proceedings of the COST G-6 Conference on Digital Audio Effects.*

Haykin, S. (1998). *Neural Networks: A Comprehensive Foundation.* Prentice Hall PTR, 2nd edition.

Hays, J. and Efros, A. A. (2008). im2gps: estimating geographic information from a single image. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.*

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9.

Iso, H., Wakamiya, S., and Aramaki, E. (2017). Density estimation for geolocation via convolutional mixture density network. *arXiv:1705.02750.*

Jenness, J. (2008). Calculating areas and centroids on the sphere. *Proceedings of the International User Conference.*

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *In Proceedings of the conference on Neural Information Processing Systems.*

Li, S., Harner, E. J., and Adjeroh, D. a. (2011). Random KNN feature selection - a fast and stable alternative to Random Forests. *BMC Bioinformatics*, 12.

Logan, B. (2000). Mel Frequency Cepstral Coefficients for Music Modeling. *Proceedings of the International Symposium on Music Information Retrieval.*

Minsky, M. and Papert, S. (1969). *Perceptrons: An Introduction to Computational Geometry.* MIT Press.

Mitrovic, D., Zeppelzauer, M., and Breiteneder, C. (2010). *Advances in Computers.* Elsevier, 78 edition.

Pugin, L. (2015). The challenge of data in digital musicology. *Frontiers in Digital Humanities*, 2.

Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). Learning internal representations by error propagation. In *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Volume 1: Foundations.* MIT Press.

Schedl, M. and Zhou, F. (2016). Fusing web and audio predictors to localize the origin of music pieces for geospatial retrieval. *Proceedings of the European Conference on Information Retrieval.*

Schuster, M., Paliwal, K. K., and General, A. (1997). Bidirectional recurrent neural networks. *Proceedings of the IEEE Transactions on Signal Processing*, 45.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. (2015). Going deeper with convolutions. *Computer Vision and Pattern Recognition.*

van den Oord, A., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A., and Kavukcuoglu, K. (2016). WaveNet: A Generative Model for Raw Audio. *Proceedings from the International Speech Communication Association Speech Synthesis Workshop.*

Vincenty, T. (2008). Direct and inverse solutions of geodesics on the ellipsoid with application of nested equations. *Survey Review*, 23(176).

Vo, N., Jacobs, N., and Hays, J. (2017). Revisiting im2gps in the deep learning era. *Proceedings of the IEEE International Conference on Computer Vision.*

Weninger, F. and Eyben, F. (2014). On-line continuous-time music mood regression with deep recurrent neural networks machine intelligence. *Proceedings of the IEEE International Conference on Acoustic.*

Weyand, T., Kostrikov, I., and Philbin, J. (2016). PlaNet - Photo Geolocation with Convolutional Neural Networks. *Proceedings of the European Conference on Computer Vision.*

Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A., Salakhudinov, R., Zemel, R., and Bengio, Y. (2015). Show, attend and tell: Neural image caption generation with visual attention. *Proceedings of the International Conference on Machine Learning.*

Yang, Y.-H. and Chen, H. H. (2011). *Music Emotion Recognition.* CRC Press, Inc., 1st edition.

Zhou, F., Claire, Q., and King, R. D. (2014). Predicting the Geographical Origin of Music. *Proceedings of the IEEE International Conference on Data Mining.*