Technische Universität Berlin

Fakultät I - Geisteswissenschaften
Fachgebiet Audiokommunikation
Audiokommunikation und -technologie M.Sc.

Self-Organizing Maps for Sound Corpus Organization

Master's Thesis

Vorgelegt von: Jonas Margraf
Matrikelnummer: 372625
E-Mail: jonasmargraf@me.com

Erstgutachter: Prof. Dr. Stefan Weinzierl
Zweitgutachter: Dr. Diemo Schwarz
Datum: February 19, 2019

Eidesstattliche Erklärung

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbstständig und eigenhändig sowie ohne unerlaubte fremde Hilfe und ausschließlich unter Verwendung der aufgeführten Quellen und Hilfsmittel angefertigt habe.
Berlin, den February 19, 2019

. . . .
Jonas Margraf

Abstract An english abstract.

Zusammenfassung Die Zusammenfassung auch auf Deutsch.

Acknowledgements

This is where the thank yous go.

Contents

1 Introduction

This is the Introduction. Here's a citation about Self-Organizing Maps (SOMs)(Kohonen, 1990).

1.1 Motivation and Problem Description

1.2 Aims and Objectives

1.3 Previous Work

2 Background

The following section intends to provide a theoretical background for two key concepts underlying the work presented in this thesis, namely Audio Feature Extraction and the Self-Organizing Map.

2.1 Audio Feature Extraction

Audio Feature Extraction is the process of deriving *features* from a digital audio signal. A feature represents some sort of descriptive information about the audio data. According to Lerch (2012), this extraction process serves a dual purpose; that of dimensionality reduction as well as a more meaningful representation. A large variety of features for different purposes have been developed (refer to Peeters (2004) for an extensive list as well as Lerch (2012) for an in-depth look at the topic). The following subsections introduce the features used in this work, starting with the pre-processing required to prepare an audio signal for feature extraction and then moving into individual definitions for each feature. The equations presented here are based on the formal definitions given by Lerch (2012) along with the computational implementations of the features in the *Meyda* library for feature extraction in JavaScript (Rawlinson et al., 2015), which in turn adapted the *yaafe* library for Python (Mathieu et al., 2010).

2.1.1 Audio Pre-Processing

Consider a digital audio signal of the form $x[n]$, where n denotes the sample index and $x[n]$ the value of the individual sample at that index.

2.1.1.1 Normalization

In order to have a standardized maximum amplitude of 1 across all audio signals, they are normalized such that

$$x_{norm}[n] = \frac{x[n]}{\max x[n]}. \tag{1}$$

2.1.1.2 Mono Conversion

Spatial information, as contained in an audio file with more than one channel, is not deemed necessary in the presented work. For this reason, all audio signals are converted to mono by taking the average of all channels.

2.1.1.3 Frame-Based Feature Extraction

Rather than performing feature extraction on the entirety of the audio signal, it is common practice to divide the signal into smaller chunks or *frames*, typically consisting of some 2^n samples (512, 1024, 2048 are often found values). The resulting feature values for each frame form a trajectory of the feature's evolution over time, which can either be used as such or can be averaged. In this work, audio signals are divided into frames with a length of 512 samples. In order to avoid computational errors (such as Not a Number (NaN) in JavaScript) during potentially silent portions of the audio signal, frames with $v_{RMS} < -60\,\mathrm{dBFS}$ (see Root Mean Square (RMS) definition below) are omitted from the feature extraction. The following equations define each feature for a single frame.

2.1.2 Time Domain Features

Time domain features are features derived directly from the discrete-time signal $x[n]$.

2.1.2.1 Duration The overall duration of the signal $x[n]$ in seconds:

$$v_{DUR} = \frac{n}{fs} s, \tag{2}$$

where n is the number of samples and fs is the sampling rate.

2.1.2.2 Root Mean Square (RMS) measures the power of a signal (Lerch, 2012, p.73f). It describes sound intensity and is sometimes used as a simple measure for loudness (Rawlinson et al., 2019a) that does not take the nonlinearity of human hearing into account (Fletcher and Munson, 1933). It is calculated for an audio frame $x[n]$ consisting of n samples such that

$$v_{RMS} = \sqrt{\frac{\sum_{i=1}^{n} x(i)^2}{n}}. \tag{3}$$

2.1.2.3 Zero-Crossing Rate (ZCR) represents the rate of the number of sign changes in a signal. It can be used as a measure of the tonalness of a sound (Lykartsis, 2014) and as a simple pitch detection method for monophonic signals (de la Cuadra, 2019). It is defined as

$$v_{ZCR} = \frac{1}{2 \cdot n} \sum_{i=1}^{n} |sgn[x(i)] - sgn[x(i-1)]|. \tag{4}$$

2.1.3 Frequency Domain Features

Frequency domain features or *spectral* features are derived from the discrete complex spectrum $X(k)$, where k refers to the frequency bin number. $X(k)$ is calculated from $x[n]$ by performing an Fast Fourier Transform (FFT) (for information on the Fourier transform, refer to any signal processing textbook, such as Oppenheim and Schafer (2014)).

2.1.3.1 Spectral Centroid is a measure of the center of gravity of a spectrum. A higher value indicates a brighter, sharper sound (Lerch, 2012). The spectral centroid is defined as

$$v_{SC} = \frac{\sum\limits_{k=0}^{N_{FFT}/2-1} k \cdot |X(k)]^2}{\sum\limits_{k=0}^{N_{FFT}/2-1} |X(k)]^2}. \tag{5}$$

2.1.3.2 Spectral Flatness is a measure for the tonality or noisiness of a signal, defined as the ratio of the geometric and arithmetic means of its magnitude spectrum. Higher values indicate a flatter (and therefore noisier) spectrum, whereas lower values point towards more tonal spectral content. It is defined as

$$v_{SFL} = \frac{\sqrt[N_{FFT}/2]{\prod\limits_{k=0}^{N_{FFT}/2-1} |X(k)|}}{(2/N_{FFT}) \cdot \sum\limits_{k=0}^{N_{FFT}/2-1} |X(k)|}. \tag{6}$$

2.1.3.3 Spectral Kurtosis indicates whether a given magnitude spectrum's distribution is similar to a Gaussian distribution. Negative values result from a flatter distribution, whereas positive values indicate a peakier distribution. A Gaussian distribution would result in a value of 0. Spectral Kurtosis is defined as

$$v_{SKU} = \frac{2 \sum\limits_{k=0}^{N_{FFT}/2-1} (|X(k)| - \mu_{|X|})^4}{N_{FFT} \cdot \sigma_{|X|}^4} - 3, \tag{7}$$

where $\mu_{|X|}$ represents the mean and $\sigma_{|X|}$ the standard deviation of the magnitude spectrum $|X|$.

2.1.3.4 Spectral Skewness assesses the symmetry of a magnitude spectrum distribution. It is defined as

$$v_{SSK} = \frac{2 \sum\limits_{k=0}^{N_{FFT}/2-1} (|X(k)| - \mu_{|X|})^3}{N_{FFT} \cdot \sigma_{|X|}^3}. \tag{8}$$

2.1.3.5 Spectral Slope represents a measure of how sloped or inclined a given spectral distribution is. The spectral slope is calculated using a linear regression of the magnitude spectrum such that

$$v_{SSL} = \frac{\sum\limits_{k=0}^{N_{FFT}/2-1} (k - \mu_k)(|X(k)| - \mu_{|X|})}{\sum\limits_{k=0}^{N_{FFT}/2-1} (k - \mu_k)^2}. \tag{9}$$

2.1.3.6 Spectral Spread is a descriptor of the concentration of a magnitude spectrum around the Spectral Centroid and assesses the corresponding signal's bandwidth. It is defined as

$$v_{SSP} = \frac{\sum\limits_{k=0}^{N_{FFT}/2-1} (k - v_{SC})^2 \cdot |X(k)|^2}{\sum\limits_{k=0}^{N_{FFT}/2-1} |X(k)|^2}. \tag{10}$$

2.1.3.7 Spectral Rolloff measures the bandwidth of a given signal by calculating that frequency bin below which lie κ percent of the sum of magnitudes of $X(k)$. Common values for κ are 0.85, 0.95 (Lerch, 2012) or 0.99 (Rawlinson et al., 2019a). It is defined as

$$v_{SR} = i \Big|_{\sum\limits_{k=0}^{i} |X(k)| = \kappa \cdot \sum\limits_{k=0}^{N_{FFT}/2-1} |X(k)|}. \tag{11}$$

2.1.4 Perceptual Features

Both the time and frequency domain features introduced above are derived from raw audio samples without taking into account any concept of human sound perception. Perceptual features incorporate some sort of model that approximates this perception. While only a single perceptual feature is used in this work, more do exist (see Peeters (2004) for a list of some of them).

2.1.4.1 Total Loudness

represents an algorithmic approximation of the human perception of a signal's loudness based on Moore et al. (1997), which uses the Bark scale as introduced by Zwicker (1961). The Total Loudness is the sum of all 24 bands' specific loudness coefficients, defined by Peeters (2004) as

$$v_{TL} = \sum_{i=1}^{24} v_{SL}(i), \tag{12}$$

where

$$v_{SL}(i) = E(i)^{0.23} \tag{13}$$

is the specific loudness of each Bark band (see Moore et al. (1997) for further details).

2.2 Self-Organizing Map

The *self-organizing map* (SOM) is a machine learning algorithm for dimensionality reduction, visualization and analysis of higher-dimensional data. Sometimes also referred to as *Kohonen map* or *network*, it was introduced in 1981 by Teuvo Kohonen (Kohonen, 1990).
The SOM is a variant of an *artificial neural network* that uses an unsupervised, competitive learning process to map a set of higher-dimensional observations (the *input vectors*) onto a regular, often two-dimensional grid or *map* of *neurons* or *nodes* that is easy to visualize. The SOM can be regarded as a nonlinear generalization of a principal component analysis (PCA) (Yin, 2007) or as a quantization of the input data, with the nodes along the map functioning as pointers into that higher-dimensional space. Each node has a position on the lower-dimensional grid as well as an associated position in the input space, which takes the form of a n-dimensional weight vector $m = [m_1, ..., m_n]$, where n is the number of dimensions of the input vectors. Nodes that are in close proximity to each other on the SOM will also have similar weight vectors (Vesanto et al., 2000), although the inverse (neighboring positions in the input space also mapping to neighboring nodes) is not necessarily true (Bauer et al., 1996).
For an in-depth look at the algorithm, its variants and applications, as well as an extensive survey of research on SOMs, the avid reader is referred to Kohonen (2001).

2.2.1 Algorithm Definition

The following definition is based on Kohonen (1990), Kohonen (2005), Kohonen and Honkela (2007) and Bauer et al. (1996).

Consider a space of input data in the form of n-dimensional vectors $x \in \mathbb{R}^n$ and an ordered set of nodes or model vectors $m_i \in \mathbb{R}^n$. A vector $x(t)$ is mapped to that node m_c with the shortest Euclidean distance from it:

$$||x(t) - m_c|| \leq ||x(t) - m_i|| \; \forall i. \tag{14}$$

This "winning" node m_c is referred to as the Best Matching Unit (BMU) for $x(t)$.

During the learning or adaptation phase of the algorithm, all nodes m_i are adjusted by a recursive regression process

$$m_i(t+1) = m_i(t) + h_{c(x),i}(x(t) - m_i(t)), \tag{15}$$

where t is the index of the current regression step, $x(t)$ is an input vector chosen randomly from the input data at this step, c is the index of the BMU for the current input vector $x(t)$ according to equation 14 and $h_{c(x),i}$ represents a so-called *neighborhood function.* The name-giving *neighborhood* is a subset N_c of nodes centered on m_c. At each learning step t, those nodes that are within N_c will be adjusted, whereas those outside of it will not. The reason for employing such a neighborhood function is so that the nodes "doing the learning are not affected independently of each other" (Kohonen, 1990, p.1467) and "the topography of the map is ensured" (Bauer et al., 1996, p.5). At its most basic, the neighborhood function is a decreasing distance function between neurons m_i and m_c. Its most common form, which is also employed in this work, is that of a Gaussian function with its peak at m_c such that

$$h_{c(x),i} = \alpha(t) \exp\left(- \frac{||r_i - r_c||^2}{2\sigma^2(t)} \right). \tag{16}$$

Here, α denotes a learning rate factor or adaptation "gain control" $0 < \alpha(t) < 1$, which decreases over the course of the regression, $r_i \in \mathbb{R}^2$ and $r_c \in \mathbb{R}^2$ are the locations of m_i and m_c on the SOM grid (the lower-dimensional output map, not the input space!), and $\sigma(t)$ is the width of the neighborhood function, which again decreases as the regression step index increases.

2.2.2 Node Initialization

Because of the iterative nature of the SOM algorithm, its outcome depends on the initial positions chosen for the nodes. The method implemented in this work uses random initialization, meaning the starting positions of the nodes are chosen randomly from within the bounds of the input space. An often employed alternative approach is to first perform a Principal Component Analysis (PCA) on the input data, select the largest d components, where d is the number of desired output dimensions for the SOM, and then distribute the nodes at equidistant intervals along those component vectors.

2.2.3 Input Data Scaling

Some consideration should be given to the dynamic range of the input data across its different dimensions. Are the dimensional ranges comparable in their limits? What about their variance? There does not appear to exist a clear consensus across the literature on whether or not normalization of input data is strictly necessary (Vesanto et al. (2000, p.34), Kohonen (1990, p.1470), Kohonen and Honkela (2007)).

Because the range of the data derived from the audio feature analysis used in this work varies considerably between features, the data for feature n is rescaled to have unit variance by dividing by the features' standard deviation σ_n:

$$x_n = \frac{x_n}{\sigma_n}. \tag{17}$$

2.2.4 Alternative Learning Rate Factors

2.2.4.1 Linear The traditional SOM algorithm uses a learning rate factor α that decreases linearly as a function of the regression step t:

$$\alpha(t) = \alpha_0\left(\frac{1-t}{T}\right), \tag{18}$$

where α_0 is the initially chosen learning rate and T is the total training length or number of regression steps.

Two other approaches to the decreasing learning rate factor were implemented in this work:

2.2.4.2 Inverse The first is a reciprocally decreasing function where

$$\alpha(t) = \frac{\alpha_0}{\left(\frac{1+100t}{T}\right)}. \tag{19}$$

2.2.4.3 BDH The second alternative approach is that of an adaptive local learning rate as developed by Bauer et al. (1996) (BDH algorithm, also see Merenyi et al. (2007)):

$$\alpha(t) = \alpha_0\left(\frac{1}{\Delta t_c}\left(\frac{1}{|x(t) - m_c|^n}\right)\right)^m, \tag{20}$$

where Δt_c represents the time since the current BMU for the current input vector was last selected as a BMU for any vector and m is a newly introduced, free control parameter. For a more complete review of the uses of this algorithm, the reader is referred to the original paper (Bauer et al., 1996) as well as Merenyi et al. (2007).

2.2.5 Forced Node Population

One of the aspects that make the SOM immediately interesting for music technology applications is that it is fundamentally based on a regular grid structure, which opens a direct connection to the grid as a musical structure.

In order to avoid "empty" nodes on the SOM - meaning nodes to which no input vectors are mapped - a post-processing extension was added to the original algorithm that inverts the mapping process, explicitly iterates over empty nodes and assigns each one that input vector which is closest. This algorithm extension, which we call Forced Node Population, executes the following sequence after the regular SOM has been calculated:

1. Select a random empty node.
2. Find closest vector for that node and assign it to this node.
3. Remove this vector from the possible choices.
4. Repeat.

This process is only performed once for all nodes that are empty immediately after the initial SOM calculation. It is possible that the forced node population creates new empty nodes, but in order to minimize distortion introduced by this procedure (see Results in section 5), it is not repeated. For a look at the results of this algorithm extension, please refer to Results (section 5).

3 Implementation

This is the Implementation.

3.1 Groundwork: CataRT Extension

3.2 SOM Browser

4 Evaluation

This is the Evaluation.

4.1 Sound Corpus Selection

- which sample library to choose?
    - no established reference data set
    - DrumEssentials: why? what is it?

4.2 Measuring SOM-Induced Quantization

4.3 Online Sound Similarity Survey

4.4 Semistructured User Interviews

- why qualitative evaluation?
    - interview subject selection
    - informed consent form
    - test subject code design
    - three part structure:
    current workflow, sample library interaction, SOM Browser interaction
    - selection of main questions
    - selection of appropriate ratings scale
    - process for categorization of responses

5 Results

This is the Results section.

6 Discussion

This is the Discussion.

6.1 Outlook

7 References

Bauer, H.-U.; Ralf Der; and Michael Herrmann (1996): “Controlling the magnification factor of self-organizing feature maps.” In: *Neural computation*, **8**(4), pp. 757–771.

de la Cuadra, Patricio (2019): “Pitch Detection Methods Review.” URL `https://ccrma.stanford.edu/~pdelac/154/m154paper.htm`.

DeSieno, Duane (1988): “Adding a conscience to competitive learning.” In: *IEEE international conference on neural networks*, vol. 1. Institute of Electrical and Electronics Engineers New York, pp. 117–124.

Fasciani, Stefano (2016): “TSAM: a tool for analyzing, modeling, and mapping the timbre of sound synthesizers.” In: .

Fiebrink, Rebecca and Baptiste Caramiaux (2016): “The machine learning algorithm as creative musical tool.” In: *Handbook of Algorithmic Music*.

Fletcher, Harvey and Wilden A Munson (1933): “Loudness, its definition, measurement and calculation.” In: *Bell System Technical Journal*, **12**(4), pp. 377–430.

Fried, Ohad; Zeyu Jin; Reid Oda; and Adam Finkelstein (2014): “AudioQuilt: 2D Arrangements of Audio Samples using Metric Learning and Kernelized Sorting.” In: *NIME*. pp. 281–286.

Gillet, Olivier and Gaël Richard (2006): “ENST-Drums: an extensive audiovisual database for drum signals processing.” In: *ISMIR*. pp. 156–159.

Goto, Masataka; Hiroki Hashiguchi; Takuichi Nishimura; and Ryuichi Oka (2002): “RWC Music Database: Popular, Classical and Jazz Music Databases.” In: *ISMIR*, vol. 2. pp. 287–288.

Kohonen, Teuvo (1990): “The Self-Organizing Map.” In: *Proceedings of the IEEE*, **78**(9), pp. 1464–1480.

Kohonen, Teuvo (1997): “Exploration of very large databases by self-organizing maps.” In: *Proceedings of International Conference on Neural Networks (ICNN’97)*, vol. 1. IEEE, pp. PL1–PL6.

Kohonen, Teuvo (2001): *Self-Organizing Maps*, vol. 30 of *Springer Series in Information Sciences*. Heidelberg: Springer.

Kohonen, Teuvo (2005): “The Self-Organizing Map (SOM).” URL `http://www.cis.hut.fi/somtoolbox/theory/somalgorithm.shtml`.

Kohonen, Teuvo and Timo Honkela (2007): “Kohonen network.” URL `http://www.scholarpedia.org/article/Kohonen_network`.

Lazar, Jonathan; Jinjuan Heidi Feng; and Harry Hochheiser (2017): *Research methods in human-computer interaction*. Morgan Kaufmann.

Lerch, Alexander (2012): *An introduction to audio content analysis: Applications in signal processing and music informatics*. Wiley-IEEE Press.

Lykartsis, Athanasios (2014): *Evaluation of accent-based rhythmic descriptors for genre classification of musical signals*. Master’s thesis, Master’s thesis, Audio Communication Group, Technische Universität Berlin ....

Mathieu, Benoit; Slim Essid; Thomas Fillon; Jacques Prado; and Gaël Richard (2010): “YAAFE, an Easy to Use and Efficient Audio Feature Extraction Software.” In: *ISMIR*. pp. 441–446.

Mathieu, Benoit; Slim Essid; Thomas Fillon; Jacques Prado; and Gaël Richard (2019): “Yaafe - audio features extraction.” URL `http://yaafe.sourceforge.net/`.

Mayring, Philipp (2010): “Qualitative Inhaltsanalyse.” In: *Handbuch qualitative Forschung in der Psychologie*. Springer, pp. 601–613.

Merenyi, Erzsbet; Abha Jain; and Thomas Villmann (2007): “Explicit magnification control of self-organizing maps for “forbidden” data.” In: *IEEE Transactions on Neural Networks*, **18**(3), pp. 786–797.

Moffat, David; David Ronan; Joshua D Reiss; et al. (2015): “An evaluation of audio feature extraction toolboxes.” In: .

Moore, Brian CJ; Brian R Glasberg; and Thomas Baer (1997): “A model for the prediction of thresholds, loudness, and partial loudness.” In: *Journal of the Audio Engineering Society*, **45**(4), pp. 224–240.

Oppenheim, Alan V and Ronald W Schafer (2014): *Discrete-time signal processing*. Pearson Education.

Peeters, Geoffroy (2004): *A large set of audio features for sound description (similarity and classification) in the CUIDADO project*. Tech. rep., IRCAM.

Rawlinson, Hugh; Nevo Segal; and Jakub Fiala (2015): “Meyda: an audio feature extraction library for the web audio api.” In: *The 1st Web Audio Conference (WAC). Paris, Fr.*

Rawlinson, Hugh; Nevo Segal; and Jakub Fiala (2019a): “Meyda: Audio feature extraction for JavaScript.” URL `https://meyda.js.org/audio-features`.

Rawlinson, Hugh; Nevo Segal; and Jakub Fiala (2019b): “Meyda: Audio feature extraction for JavaScript.” URL `https://github.com/meyda/meyda`.

Scholler, Simon and Hendrik Purwins (2010): “Sparse coding for drum sound classification and its use as a similarity measure.” In: *Proceedings of 3rd international workshop on Machine learning and music*. ACM, pp. 9–12.

Shier, Jordie; Kirk McNally; and George Tzanetakis (2017): “Analysis of Drum Machine Kick and Snare Sounds.” In: *Audio Engineering Society Convention 143*. Audio Engineering Society.

Vagias, Wade M (2006): “Likert-type Scale Response Anchors. Clemson International Institute for Tourism.” In: *& Research Development, Department of Parks, Recreation and Tourism Management, Clemson University*.

Vesanto, Juha; Johan Himberg; Esa Alhoniemi; and Juha Parhankangas (2000): “SOM toolbox for Matlab 5.” In: *Helsinki University of Technology, Finland*, p. 109.

Villmann, Thomas and Jens Christian Claussen (2006): “Magnification control in self-organizing maps and neural gas.” In: *Neural Computation*, **18**(2), pp. 446–469.

Yin, Hujun (2007): “Nonlinear dimensionality reduction and data visualization: a review.” In: *International Journal of Automation and Computing*, **4**(3), pp. 294–303.

Zwicker, Eberhard (1961): “Subdivision of the audible frequency range into critical bands (Frequenzgruppen).” In: *The Journal of the Acoustical Society of America*, **33**(2), pp. 248–248.

Appendices

A LaTeX Sources

The LaTeX sources for this work can be found in XXX.

B Thesis Bibliography

The references used in this work can be found in XXX.

Acronyms

BMU	Best Matching Unit.
FFT	Fast Fourier Transform.
NaN	Not a Number.
PCA	Principal Component Analysis.
SOM	Self-Organizing Map.

List of Figures

List of Listings

List of Tables

Digital Resource

This page holds a data disk.