On Training Neural Networks to Intuitively Classify Music Genre

Alexander Reid

Earlham College

TABLE OF CONTENTS

***Abstract*—This study is centered around the challenge of musical style classification, and in particular this study approaches this issue from the point of view of optimally training Artificial Neural Networks to accomplish this task. In the past, Artificial Neural Networks have been some of the most effective methods for approaching this issue, as the cited studies depict. This study sought to classify music into ten musical genres: Pop, Rock, Heavy Metal, Reggae, Country, Techno, Blues, Jazz, HipHop/Rap, & Dance using an Artificial Neural Network implementation with an underlying stratified K-fold Cross Validation paradigm, as this has proven to be effective against undesirable biases in this context. Ultimately the results of this study were 63% accuracy on the stock dataset of ten genres, while accuracy on the combination of the dataset & the Dummy Plug, which contained an additional twenty genres, was 45% accuracy on average. This approach & the framework created around it can be seen as a mechanism for audio platforms to quickly, intelligently, & effectively reply to the contemporary surge in the development of musical genres.**

***Index Terms*—Music, Genre Classification, Artificial Neural Networks**

I. INTRODUCTION

IN this paper the challenge of musical genre classification is explored—utilizing cited studies as both references and generalizable perspectives of this field of work as a whole. Specifically this paper views this problem from the perspective of training Neural Networks to accomplish this task optimally. Historically, Neural Networks have been one of the most optimal methods for approaching this issue, as the studies in [1],[2],[4] and [5] depict. The necessity of a solution for this problem is evident, in light of the fact that advances in digital storage and distribution capabilities have, in their wake, rendered past solutions ineffective and often impractical. This, in turn has lead to a rising number of instances in which businesses, corporations, organizations etc., are in dire need of intelligent and efficient methods to organize and disseminate their digital media. *Xue et al.* note that rapid development of affordable multimedia content, storage and throughput, due in part to the development of compression standards such as JPEG and MPEG "...have resulted in a rapid increase of the size of digital multimedia data collections and greatly increased the availability of multimedia content to the general user." [11] The researchers also observe that, music is distributed and transferred from various sources including "music shops, broadcasts and the internet" [11]

Burred et al. state that in recent years there have been many different approaches to musical genre classification ranging from "music/speech discriminators" to "systems based on elaborate music and non-musical taxonomies" and that "all of these systems rely on pattern recognition techniques."[2] *Tripp et al.* note that similar attempts had looked at the use of spectral guided classification, but that historically, none have been "consistently helpful in genre classification."[8] They also determine that many other studies have looked to classification schemes based on notational, i.e. MIDI data, or the recognition of tonal characteristics of various types of instruments in a song that may indicate genre.[8] *Xue et al.* comment that "actual" or raw sounds files (such as Wav and Mp3) are fundamentally different from MIDI, and while analysis and classification in the context of MIDI is much less complex and efficient, due to its lack of presence in the field "MIDI style classification is not practical in real application." [11] The importance of the general approach of recognizing abstract patterns in the characteristics of songs is evident in its inclusion in most, if not all studies of this type, especially the experiment done by *Xue et al.* in which they approach the problem of musical genre classification using a "multi-layer classifier" based on Support Vector Machines to obtain the optimal genre calls boundaries.[11]

In particular *Tzanetkais et al.* note that songs within the same genre generally exhibit the same or highly similar patterns of instrumentation, rhythmic structure, and pitch content. [7] *Xue et al.* note that it is "extremely much more difficult to discriminate musical genres than [it is to] discriminate music, speech and other sounds". *Xue et al.* also mention that in respect to speech, audio can be described in terms of phones (basic discrete units of speech related sounds) and that generally, "from an acoustic point of view, music can be described as a sequence of acoustic events. Therefore, we have to transform the acoustic signal into a sequence of abstract acoustic events". [11] *Soltau et al.* state that using a system that tries to do this type of discernment or pattern recognition by note distribution is likely to fail due to the complexity of polyphonic musical structures.[6] This is very evident in the cases of samples that have many sections of simultaneous note occurrence, as the lack of ability to distinguish notes that may relate to important classification factors adversely affect overall classification rate.

Xue et al. notice that as a general trend "...methods tried to use one classifier and several features to classify music into different genres at a time. However according to music knowledge, it is easy to discriminate some genres of music (i.e. pop and classic) using some features, but it may be difficult to use the same features to discriminate other genres of music (i.e. pop and rock)[11] *Burred et al.* made a point of noting that many important issues in the musical classification of genre such as "genre dependency of the features"(Some features are more suitable than others for the classification process of certain genres), "problems of dimensionality"(in some cases the addition of new features does not necessarily result in higher classification rates), "inappropriate taxonomies"(many suggested taxonomies are 'too simple or inconsistent' to be viable).[2] *Xue et al.* state that for humans, "it is not difficult to classify music into different genres, but that historically for computer based systems, is has been a significant challenge. They note that a possible reason for this is that, while genre classification systems use complex mathematical and statistical models to make these connections, there are still a wide range of "perceptual criteria" that are important to this task.

These criterion are related to the melody, tempo, texture, instrumentation, and rhythmic structure of music that can be used to characterize and discriminate different musical genres.[11] *Soltau et al.* also determine that a classification system must have the ability to cope with temporal structure of acoustic signals and that speech signals are often modeled using Hidden Markov Models, but that such an approach has poor discriminatory power, whereas Neural Networks have shown better performance in this context when they are provided specific "topologies" as training sets.[6] Both HMMs and NNs will be discussed in greater detail in the subsequent sections This appears to be not only accurate, but widely acknowledged as the overwhelming majority of studies in this field have turned to Neural Networks of varying types to implement their experiments.

The context of this study, as well as the justification of its judgments and assumptions, are based upon a comparative analysis of the studies cited here. These studies give noteworthy and intriguing perspectives on the matters that are arguably at the center of Musical Genre Classification. The perspectives and conclusions derived by these experiments are examined, juxtaposed, amalgamated and incorporated with that of this experiment.

II. Musical Genre Classification

Overview

This study found that there are three key components of any approach to Musical Genre Classification, which are Statistical Classification, Musical Analysis and Conceptions of Genre. These three components can be seen in any of the cited studies as the integral aspects and/or dependencies of the frameworks and approaches used. This holds true regardless of how radical these approaches are in relation to what has been done historically or contemporarily.

A. Statistical Classification

Musical Genre Classification deals with the general problem of classification, and thus categorization. Classification here is primarily concerned with the issue of identifying which of a certain set of categories a particular element belongs to, on the basis of exemplary sets containing data which has already been categorized. Statistical Classification is canonically implemented within the following methodologies: *Linear classifiers*[1], *Support Vector Machines*[2], *Quadratic Classifiers*[3], *Kernel Estimation*[4], *Boosting*[5], *Decision Trees*[6], *Neural Networks*[7], *Gene Expression*[8], *Bayesian Networks*[9], *Hidden Markov Models*[10], *Learning Vector Quantization*[11]. Statistical Classification has many varied application domains, as it is applicable to any problem in which the discovery of patterns is paramount to finding solutions. In terms

[1] A Linear Classifier uses an element's characteristics to identify which class it belongs to statistically by making decisions based on the value of a linear combination of the characteristics.

[2] SVMs are supervised learning models with associated learning algorithms that analyze data and recognize patterns, they are primarily used for classification and regression analysis.

[3] Quadratic classifiers are primarily used in machine learning and statistical classification to separate measurements of multiple classes of objects or events by a quadric surface. It is a more general version of the linear classifier.

[4] Kernel Estimation is a form of kernel density estimation, in which the kernels used for estimation have varied sizes depending on either the location of the samples or the location of the test point.

[5] Boosting is a machine learning meta-algorithm for reducing amount of bias that is gnereally inherent in supervised learning.

[6] The framework of machine learning using Decision Trees is a decision support tool that uses a tree-like graph or model of decisions and their possible consequences, including chance event outcomes, resource costs, and utility.

[7] Neural Networks are models inspired by anthropoidic nervous systems (in particular the brain) that are capable of machine learning and pattern recognition.

[8] Gene Expression is an evolutionary algorithm that creates computer programs or models.

[9] A Bayesian Network is a probabilistic graphical model (a type of statistical model) that represents a set of random variables and their conditional dependencies using graph theory, and in particular directed acyclic graphs.

[10] Hidden Markov Models are a type of statistical Markov model in which the system being modeled is assumed to be a Markov process with hidden states.

[11] Learning Vector Quantization is a special case of an Neural Network, that applies a winner-take-all Hebbian learning-based approach.

of fields and disciplines, Statistical Classification is found within the following domains: *Computer Vision, Drug discovery, Drug Development, Geostatistics, Speech Recognition, Handwriting Recognition, Biometric Identification, Biological Classification, Statistical Natural Language Processing, Document Classification, Internet Search engines, Credit Scoring and Pattern Recognition.*

B. Musical Analysis

In Muscial Analysis, individual elements of a piece are classified in terms of a set of quantifiable features that are globally applicable and that may be ordinal, categorical, integral or complex. Once these features have been defined they are applied to the analysis of the element in question, and the "class" of that element, in effect is the result of that analysis. Analysis granualizes the element into discrete sections, each of which may be individually examined to reach some conclusion—pertaining to that aspect of the element, or the element as a whole. In this context, the analysis of an element's features provide a definitive, or at least decisive answer to the genre of that particular piece of audio. Some of the canonical methodologies used in Musical Analysis are *BPM (Beats per Minute) analysis, ZCR analysis*[12]*, Spectral Characteristic analysis, Instrumentality analysis, Pitch analysis*[13]*, Beat Characteristic analysis*[14]*, Tonal analysis,*[15] *Tempo analysis*[16]*, LDA*[17]*, and Adaptive Boosting*[18].

C. Genre

Genre is ususally thought of as categories of artistic, musical, or literary composition characterized by a particular style, form, or content. The concept of genre itself is notable, as it is a collection of subjectively perceived sets of style criterions, that are combined in groups of traits per each respective genre. The inherent subjective component of genre is necessary because for any classification scheme to be useful as a tool, it must in a general sense, be able to adapt to changing meanings in its subject. Thus, it can be argued that genre always has the potential to evolve; although genre as a classification system still suffers from the same canonical limitations of any classification

[12]The Zero-Crossing Rate is the rate of sign-changes along a signal; the rate at which a signal changes from positive to negative or vice versa.

[13]Pitch is a perceptual property that allows the ordering of sounds on a frequency-related scale.

[14]In music and music theory, the Beat is the basic unit of time. Also in acoustics, the beat is an interference between two sounds of slightly different frequencies, perceived as periodic variations in volume whose rate is the difference between the two frequencies.

[15]Tonality is a system of music in which specific hierarchical pitch relationships are created based around a key "center"; which relates to the hierarchical relationships between the triads.

[16]Tempo is the speed or pace of a given piece.

[17]* Noted on page 1

[18]Adaptive Boosting (AdaBoost for short), is a machine learning meta-algorithm, and can be used in conjunction with many other learning algorithms to improve their performance. AdaBoost is adaptive in that generally subsequent classifiers that arise during the process are tweaked to account for those instances misclassified by previous classifiers.

system. Musical Genre Classification in any form then, is ultimately tasked with the definition or labeling of a piece—in terms of a defined set of features that are characteristic of music in general; and of a specific genre—to which that song should belong.

III. Historical Contributions

Overview

The historical approaches described and discussed below, have contributed vast amounts of knowledge and insight to solving the problem of Musical Genre Classification. Each of these paradigms gained more nuanced perspectives and arguably, formulated more comprehensive approaches, in light of their predecessors—particularly by considering the 3 aforementioned aspects of the problem (Statistical Classification, Musical Analysis, and conceptions of Genre itself). These 3 aspects are the core concepts of this problem itself, as the historical work done in this problem domain supports that notion.

*A. **AdaBoost***

In 2003 Yoav Freund and Robert Schapire won the Godel Prize for the creation of AdaBoost in 1995, a machine learning algorithm that can be used in conjunction with other machine learning algorithms to improve augment their performance. The moniker AdaBoost is short for ”Adaptive Boosting” which is due to the fact that when AdaBoost constructs a classifier, it inherently does so while simultaneously tweaking the new classifier in favor of any instances that were previously misclassified by previous classifiers. In [8] *Tripp et al.* utilized AdaBoost as one possible classifier for their feature set scheme. The basis of their study in particular was to combine spectral analysis with as many “relevant features” of a song as possible, that when applied to a classification algorithm would be more effective that previous classification attempts. In terms of “relevant features” the 102 musicological features that *Trip et al.* found were aspects of ZCR, BPM, Gross Spectral Analysis and Linear Coefficients. The four classification frameworks chosen in [8] are two instances of a SVM given different kernels, but all the same 102 features discussed before. In [8] the data used by this study consisted of a database of 2,600 songs each belonging to one of five distinct genres. Here AdaBoost achieved higher overall classification rates than all of the other classifiers used, (81% AdaBoost classification rate) but fell short of its successor LogitBoost, which is described in detail later. In some instances AdaBoost's performance augmentation can be weak, but as long as the modified algorithm's base performance is slightly better than random, AdaBoost will improve the final model.

B. *Gaussian Mixture Models*

A Gaussian Mixture Model (GMM) is a Parametric Probability Density function[19] represented as a weighted sum of Gaussian Component Densities. GMMs are commonly used as a Parametric Model[20] of the Probability Distribution[21] of continuous measurements, or even features in a biometric system (i.e. vocal-tract related spectral features in a speaker recognition system.) GMM parameters are estimated from training data using the iterative Expectation-Maximization algorithm[22] or Maximum A Posteriori (MAP) estimation[23] from a well-trained prior model.

For the experiment done by *Xu et al.* in [11] a GMM instance, augmented with data corresponding to musicological features is implemented. In this study researchers note that their approach was based off of other historical work done by *Jiang et al. (2002)* and ***Pye et al. (2000)***. The later of these two, extracted specifically the Mel-Frequency Cepstral Coefficients (MFCC) features of a given piece of music, and used a Gaussian Mixture Model as their classifier to distinguish six music types (Blues, Easy listening, Dance, Classic, Opera, and India Rock.) In [11] the dataset for *Xu et al.*'s experiment consisted of 850 samples from 17 distinct genres, the optimal classification agent derived by the study was a 3-component Gaussian Mixture Model(the components being MFCC, Energy and a composite feature set *Xu et al.* formulated (based solely upon instrumentation), which was able to achieve classification rates of 88%.

[19]A Probability Density function is one that expresses the relative likelihood of a random variable to take on a specific value.

[20]A Parametric Model or Parametric Family is a closely related set of distributions that can be expressed by a finite number of parameters.

[21]A Probability Distribution assigns a probability to every measurable subset of possible outcomes in random experiment of statistical inference.

[22]An iterative method for finding maximum likelihood or Maximum a Posteriori estimates of parameters in statistical models, where the model depends on unobserved latent variables.

[23]An estimate that represents a mode of the posterior distribution in a Bayesian statistical framework.

C. *Hidden Markov Models*

In Probability theory, a Markov model is a stochastic model that assumes the Markov property.[24] In particular, Hidden Markov Models (HMMs) are learnable finite stochastic automates. Contemporarily, HMMs are considered to be a specific form of Dynamic Bayesian Networks. These, as well as other Dynamic Bayesian Networks, are based on the work done by Thomas Bayes in the 1700s.[25] Moreover an HMM can be considered the simplest Dynamic Bayesian Network; and the mathematical framework at the center of the HMM was developed by *L. E. Baum et al.* in the early 1970s.

[24]The Markov property, named after the Russian mathematician Andrey Andreyevich Markov and developed in 1906, refers to the memoryless property of a stochastic process.

[25]*An Essay towards solving a Problem in the Doctrine of Chances* Bayes & Price, 1763

In essence the history of the HMMs consists of two parts. On the one hand there is the development of Markov process and Markov chains, and on the other hand there is the development of algorithms needed to formulate the Hidden Markov Model in order to solve problems in the modern applied sciences. In a Hidden Markov Model, the system being modeled is assumed to be a Markov process[26] with unobserved (hidden)[27] states. Therefore within any HMM, the states of a model are not directly visible, but output of that model, depending on the state of the HMM, is visible. Structurally a HMM consists of two stochastic processes[28]. The first stochastic process is a Markov chain[29] that is characterized by states and transition probabilities. The states of the chain are externally not visible, and are therefore “hidden”. The second stochastic process produces emissions observable at each moment, depending on a state-dependent probability distribution. Therefore the sequence of emitted tokens generated by an HMM gives some key information about the sequence of states itself.

Two illustrations of HMM implementations and evaluations are seen in the studies depicted in [6] & [10]. In [6] *Soltau et al.'s* experimental results show that their canonical HMM approach was able to produce classification rates of approximately 80% accuracy. Soltau et al.'s study consisted of a database of 360 samples evenly distributed between the four genres of Rock, Pop, Techno and Classical, each sample was approximately 30 seconds. In [10] *Shao et al.*'s HMM instance was able to achieve a 88% classification rate as well, and in their study the dataset was comprised of 100 music files that fit into one of four genres (Rock, Jazz, Classic & Pop). Hidden Markov Models are especially known for their application in temporal pattern recognition (*Speech*, *Handwriting* and *Gesture Recognition*[30], *Part-of-Speech Tagging*[31], *Musical Score following*[32], *Par-*

[26] A Stochastic process possesses the Markov property if the conditional probability distribution of its future states (conditional on both past and present values) depends only upon its present state, not on the sequence of events that preceded it.

[27] Here the adjective ‘hidden’ refers to the state sequence through which the model passes, not to the parameters of the model; even if the model parameters are known exactly, the model is still ‘hidden’.

[28] A Stochastic process, (also often dubbed as a random process) is a collection of random variables; the combination of which may be used to represent the evolution of some random value or system, over time.

[29] The Markov chains intoduced by Markov in 1906 when he produced the first theoretical results for stochastic processes by using the term “chain” for the first time.

[30] Gesture Recognition is a topic in computer science and language technology with the goal of interpreting human gestures via mathematical algorithms.

[31] In Corpus Linguistics, Part-of-Speech Tagging, also called grammatical tagging or word-category disambiguation, is the process of marking up a word in a text as corresponding to a particular part of speech, based on both its definition, as well as its context.

[32] Score Following is the process of automatically listening to a live music performance and tracking the position in the score.

tial Discharges[33] and *Bioinformatics*[34]).

D. Linear Discriminant Analysis

Discriminant Analysis was originally developed in 1936 by R.A. Fisher. Discriminant Analysis can be used only for classification (i.e., with a categorical target label), not for Regression[35]. For Discriminant Analysis to be effective the subject of the classification operation must have multiple possible categories to which it may belong. Specifically, Linear Discriminant Analysis (LDA) methods are used in statistics, pattern recognition and machine learning to deduce a linear combination of features which characterizes or differentiates two or more classes of objects or elements.[36] The resulting linear combination may be used as a linear classifier, or more effectively, as a form of dimensionality reduction before later classification. This process may be repeated several times where applicable. LDA is closely related to ANOVA (analysis of variance) and Regression Analysis, both of which also endeavor to express one dependent variable as a linear combination of other features. In these methods however, the dependent variable is a numerical quantity, while for LDA it is always a categorical variable (i.e. the class label). It is important to note that a fundamental assumption of the LDA method is that the independent variables in a given instance are normally distributed. In 2003 the study in [7] looked to SVMs to handle classification based on the feature sets of the concepts *Tzanetakis et al.* found important to the task of genre classifcation. The experiment utilized a dataset of 100 sound files, each of which belonged to one of 10 genres (Classical, Country, Disco, Hip-Hop, Jazz, Rock, Blues, Reggae, Pop, and Heavy Metal).*Tzanetaki et al.* note that their SVM implementation, given somewhat minimal feature sets of Beat[37], Full Beat[38], Pitch[39] and Full pitch[40] and LDA individually, showed moderate classification rates, (approx 25-30% on average). The researchers in [7] found

[33] In Electrical Engineering, partial discharge (PD) is a localized dielectric breakdown of a small portion of a solid or fluid electrical insulation system under high voltage stress, which does not bridge the space between two conductors.

[34] Bioinformatics is an interdisciplinary field that develops and improves on methods for storing, retrieving, organizing and analyzing biological data.

[35] In statistics, Regression Analysis is a statistical process for estimating the relationships among variables. It includes many techniques for modeling and analyzing several variables, when the focus is on the relationship between a dependent variable and one or more independent variables. More specifically, Regression Analysis helps one understand how the typical value of the dependent variable (or 'Criterion Variable') changes when any one of the independent variables is varied, while the other independent variables are held fixed.

[36] In Mathematics, a Linear Combination is an expression constructed from a set of terms by multiplying each term by a constant and adding the results

[37] 6 numerical features that attempt to summarize a Beat histogram.

[38] 300 numerical features that attempt to summarize a Beat histogram.

[39] 6 numerical features that attempt to summarize a Pitch histogram.

[40] 130 numerical features that attempt to summarize a Pitch histogram.

that those figures paled in comparison to the performance observed from the use of a more "full feature set" A, which combined thee four minimal feature sets, in combination with LDA. This dynamic allowed the SVM/LDA instance to achieve a significantly higher 71% overall classification rate.

E. *LogitBoost*

LogitBoost is a boosting algorithm formulated in 1998 by Jerome Friedman, Trevor Hastie, and Robert Tibshirani. LogistBoost views the AdaBoost algorithm as a component of a statistical framework. In some sense LogitBoost is primarily concerned with Convex Minimization[41]. Specifically, LogitBoost considers AdaBoost as a generalized additive model[42] and then applies the cost functional of logistic regression[43] to that additive model. In [8], LogitBoost is the third framework *Tripp et al.* used as classifier for their extracted feature sets. As previously mentioned in [8] the study's dataset consisted of 2,600 songs all belonging to one of five distinct genres. *Tripp et al.* determined that overall LogitBoost feed the 102 "relevant features of their classification scheme, outperformed even their two SVM instances, with LogitBoost attaining the highest classification rate (83% LogitBoost classification rate). *Tripp et al.* also note LogitBoost's predecessor Adaboost fell slightly short of that.

[41]Convex Minimization is a subfield of optimization, which studies the problem of minimizing convex functions over convex sets.

[42]In statistics, a generalized additive model is a generalized linear model in which the linear predictor depends linearly on unknown smooth functions of some predictor variables, and interest focuses on inference about these smooth functions.

[43]In statistics, Logistic Regression is a type of regression analysis used for predicting the outcome of a categorical dependent variable based on one or more predictor variables.

F. *Neural Networks*

Originally, the inspiration for Neural Networks arose from studies of the mechanisms for information processing in biological systems, and in particular the human brain. However, the basic concepts can also be understood from a purely abstract approach to information processing. In addition to high processing speed, Neural Networks have the paramount ability of learning a general solution to a problem from a set of specific examples.

The origins of Neural Networks, or Neural Computing can arguably be found in a innovative paper by McCulloch and Pitts in 1943, outlining a simple mathematical model of a single neuron. They showed that networks of model neurons are capable of universal computation,—they can in principle emulate any general purpose computing machine. During the late 1950s the first hardware Neural Network system was developed by Rosenblatt, dubbed the perceptron. Rosenblatt demonstrated theoretically the remarkable result that, if a given problem was solvable in principle by a perceptron, then the perceptron learning algorithm was guaranteed to find the solution in a finite number of steps. A dramatic resurgence of interest in Neural Networks began in the early 1980s partly due to a major development by way of the discovery of learning algorithms, based on error backpropagaition, which overcame the principal limitations of earlier Neural Networks.[1]

*1) **Classes of Neural Networks**:* A Feed-Forward Neural Network can be viewed simply as a non-linear mathematical function Which transforms a set of input variables into a set of output variables in a FFNN (*Feed-Forward Neural Network*.) This transformation is determined by a set of numeric parameters called weights, whose values can be determined on the basis of a set of examples of that accurately reflect the test set. [1]

In light of the biological notion that the human brain functions not as a single massive network, but in fact as a collection of small networks led to the formulation of MNNs (*Modular Neural Networks*), in which several small networks cooperate or compete to solve problems. The series of independent Neural Networks are moderated by some intermediary process, which guides the formulation of their decision. The intermediary process combines outputs from every module to produce the output of the network as a whole. Moreover, the intermediary process simply deals with the modules outputs; it has no interaction with the modules themselves—in terms of communication or otherwise. It is important to note that under the MNN paradigm the modules also do not interact with each other. Every independent NN acts as a module, which operates on separate inputs to accomplish some subtask of the main task the MNN hopes to perform.

Two examples of Modular Neural Networks are Committees of Machines, and Associative Neural Networks. *A Committee of Machines* (CoM) is a group of different neural networks that together ”vote” on a given decision based on a specific data element in question. A CoM specifically tends to stabilize the results of processes which typically generate local minima. The CoM paradigm has an inherent necessary variety of machines in the committee which is obtained by training each network from varied random starting weights.

The *Associative Neural Network* (ASNN) is an adjunct of the CoM paradigm that is more comprehensive than the simple/weighted average of different models seen in the CoM approach. An ASNN is formed by an ensemble of FFNNs with the k-nearest neighbor technique (kNN) applied to them. The ASNNs use of correlation between ensemble outputs as a distance measure in the context of analyzed cases for the kNN, corrects the biases of the neural network ensembles. When an ASNN is presented with new data the network automatically improves its predictive ability and provides added data approximation to respond to the novel elements of the updated datasetteaching itself the data without the need to actually retrain the ensemble.

In a RBF (*Radial Basis Function*) Network is a type of NN in which the only parameters that are adjusted in the learning process are the linear mapping from hidden layer to output layer. Due to this, RBF networks do not suffer from the issues of developing local minima as Multi-Layer Perceptrons normally do—linearity ensures that the error surface is quadratic and therefore has a single easily found minimum. However RBF networks are typically at a disadvantage when it comes to acquiring good coverage of the input data range by radial basis functions. RBF centers are determined with reference to the distribution of the input data, unrelated to the prediction task. As a result, representational resources may be wasted on areas of the input space that are irrelevant to the learning task. RBF networks are typically trained in a Maximum Likelihood framework—minimizing the error of the data under the model—similar to Gaussian Processes. In particular RBF networks show great performance in regression applications when the dimensionality of the input space is relatively small; even in this context though, RBF networks are usually outperformed by SVMs.

A RNN (*Recurrent Neural Network*) is a type of NN where connections between units form a directed cycle—a sequence of vertices that originate from and progress to the same vertex, therefore every pair of consecutive vertices has an edge directed from the earlier vertex to the later one. This structure forms an internal condition of the network that in particular allows it have dynamic temporal behavior. RNNs are especially applicable to tasks such as handwriting recognition due to their ability to use internal memory in processing arbitrary sequences of inputs. In this context RNNs are reported to surpass even Feed-Forward Neural Networks.

*2) **Neural Network Topologies**:*

a) Multilayer Perceptron: The first is called the multilayer perceptron (MLP) and is currently the most widely used Neural Network model for practical applications. [1]

b) Radial Basis Function: The second model is known as the Radial Basis Function (RBF) network, which has also been used successfully in a variety of applications, and which has a number of advantages, as well as limitations, compared with the MLP. [1]

IV. Contemporary Analogues

In light of the historical work in this problem domain of Musical Genre Classification, the section below discusses a contemporary analogue of the work previously covered. This analogue approaches a nearly identical problem, from a corresponding perspective—leveraging notions of musical, and other forms of analysis, to find statistically relevant, generalizable, and reproducible characteristics for matching homogeneous pieces of music.

It is important to note that this specific example and other frameworks like it, have had to address more contemporary issues and constraints in relation to this task. These include large-scale Database Management, Distributed Computing, and Data Integrity in Network Traversal. These, and other related problem domains, at the time were not relevant to, or even in the scope of the work done by their predecessors.

*A. **Shazam Entertainment Ltd.***

The Shazam service was developed with the intent to increase a user's exposure and knowledge of music by being able to recognize music spontaneously playing in the environment. This is ability is desired even in the presence of heavy ambient noise, compression, network interference and with a high rate of accuracy and at a high speeds. Shazams algorithm has been shown to identify samples hidden by loud vocoders and even reverberation, which is usually the case in radio samples—quickly over a database of over 2 million tracks with a low number of false positives. Therefore Shazam's algorithm can operate normally under high levels of noise and non-linear distortion—to such an extent that is can reliably perform recognition in the presence of voices, traffic noise, dropout[44], and even other music.[9] In fact, Wang notes that a statistically significant match can be derived from a sample in which only approximately 1-2% of candidate matches actually contribute to the recognition process. A rather impressive result of this is Shazam's ability to identify several songs that are mixed together—even multiple versions of the same piece. This particular property of the Shazam framework is called "Transparency". [9]

During its recognition process, Shazam fingerprints all samples in question—both unknown samples in queries, and that of tracks stored in their database for recognition. The fingerprints from unknown samples are compared to that of stored samples. After which candidate matches are generated, and then evaluated for accuracy. Wang notes that these fingerprints are required to be "temporally localized", "translation-invariant", "robust", and "sufficiently entropic." In particular, the translation invariant property signifies that regardless of where in the file matching content is found, recognition is always possible—a notable advantage, due to the fact that an unknown sample could be located at an arbitrary position in the original song.[9] Requiring a specific degree of entropy in a fingerprint inherently diminishes the possibility of spurious or erroneous, and excessive matches, due to lower correspondence rates in the presence of higher entropy. By implementing combinatorial hashing of fingerprints, Shazam gains a reported 1000% speed-up, with the trade-off of 10 times the amount of storage space normally

[44]The loss of sections of samples during playback and/or recording.

needed, and a minimal loss in probability of signal detection.

Ultimately, Shazams algorithm solves this problem of real-time recognition in approximately N*log(N) time, where ‘N’ denotes the number of matching fingerprint hash locations over the recognition time duration. Wang observes that search time in Shazams framework—when applied to a database of approximately 20 thousand tracks for example, is on the order of 5-500 milliseconds depending on the parameter settings and application.[9] With a heavily corrupted sample, recognition is reported to be possible within “a few hundred milliseconds”, whereas with respect to radio quality samples, recognition on average takes “less than 10 milliseconds.”[9]

Wang notes that analogous approaches have been implemented by competing services such as Musicwave, Relatable, Neuros, Audible Magic and Muscle Fish. The Shazam service is similar to the approach taken in this study in that it deconstructs a song; distilling it into a collection of descriptive features, used to find identical matches. Alternatively, the approach of this study similarly deconstructs a song in terms of features, but in order to juxtapose songs as collections of component features and cluster their similarities into sets of characteristics that denote a particular genreeffective in training neural networks to differentiate genres.

V. Contributions of this Study

A. *Approach*

The approach of this study is focused on training Neural Networks to intuitively classify musical genre. This training is comprised of several important aspects relating to, the actual formation of the dataset employed—decisions made in that process(range of variance; perception of elements), how the data is presented to the network in light of both pre and postprocessing techniques, and the parameters of the actual training method. Juxtaposition of data elements ultimately allows the network to learn from generalizable patterns within the dataset.

This study will classify music into one of 10 genres: *Pop, Rock, Heavy Metal, Reggae, Country, Techno, Blues, Jazz, HipHop/Rap,* & *Dance* using an Artificial Neural Network implementation with an underlying stratified K-fold Cross Validation paradigm as this has proven to be effective for this reducing unwanted biases in this context. Also in order to test the robustness of this studyś approach annother much larger set of genres called the “Dummy Plug” will be used in conjunction with the 10 genres in the aforementioned stock dataset to train and test the ANN implementation. The “Dummy Plug” is comprised of 20 genres: *Breakbeat, Chill, Chiptune, Deathcore, Djent, Downtempo, Drum & Bass, Easy Listening, Electronicore, Experimental, Folk, Hardcore, Psychedelic, Industrial, Metalcore*, and *Samba*.

The musical features this study will be considering as the framework is configured and trained are, Mel Frequency Cepstral Coefficients (MFCC), Beat, Pitch, Beat Spectrum, LPC-derived Cepstrum, Spectrum Power, Loudness, Zero Crossing Rates, Spectral Centroid, Spectral Centroid, Spectral Flux, Spectral Flatness, Spectral Crest Factor, Line Spectral Pairings and Chromatic components.

Integration of all of the considerations and aspects of the components detailed above, into the implementation of this study are possible with the M.A.R.S.Y.A.S.[45] suite of tools. The main priority of this experiment is to leverage the power of MARSYAS as efficiently and effectively from our perspective—to derive classification rates that are higher, or on par with leading experiments.

B. ***Frameworks Employed***

1) MARSYAS:

a) Available Tools: ***Bextract*** is one of the central and most comprehensive executables provided in Marsyas toolchain. It is used for complete feature extraction for multiple files. Bextract is a conglomeration of many different audio analysis algorithms leveraged by the framework. Bextract has various options for the analysis process including the size of the analysis windows, the degree to which those windows overlap, the amount of samples to be extracted from each window before it is analyzed, and whether or not features are extracted in stereo or in mono, to name a few.

Within the MARSYAS toolchain ***Kea*** is the product of Tzanetakis et al. establishing rigorous programming conventions for the implementation of machine learning in the MARSYAS suite. Kea (*named after a rare bird from New Zealand*) is the MARSYAS counterpart of Weka in that it has similar capabilities—although much more limited; as well as a command-line interface to Weka itself. Kea is primarily useful for developing first order approximations in NN testing and analysis.

Sfplugin is the real-time audio classifier provided with the MARSYAS suite. Sfplugin takes a Artificial Neural Network instance created by Bextract and uses it to analyze a single or a series of audio files. These can be either compressed in the case of the MP3 format, or uncompressed in the case of the WAV format. This combination of tools is capable of classification on the order of milliseconds per file. The audio files used do not need to be formatted or modified in for the process to work, and the process itself also does not modify the files in any way. The accuracy of the classifications made using Sfplugin depends on the underlying artificial neural network and how effectively it has been trained

2) WEKA:

a) Waikato Environment for Knowledge Analysis: ***WEKA*** is a suite of machine learning

[45] *Music Analysis Retrieval and SYnthesis for Audio Signals* created by Graham Percival and George Tzanetakis. Tzanetakis is also a coauthor of studies [5] & [7]

software written in Java, under the GNU General Public License. Weka was developed at the University of Waikato, New Zealand. The Weka suite is a collection of tools and algorithms designed for data analysis, predictive modeling, data pre-processing, and a Neural Network framework for machine learning experiments written in C. All components of the Weka suite are predicated on the assumption that the data presented as a single uniformly formatted collection or relation of elementswhere each piece of data is described by a fixed and finite number of attributes.Its underlying Java architecture makes Weka highly portable to any modern computing platform.

C. Languages Employed

1) Perl: Perl[46] is a family of high-level, general-purpose, interpreted, dynamic programming languages. Perl has the power and flexibility of a high-level programming language such as C and in fact, many of the features of the language are borrowed from C. However like shell script languages, Perl does not require a special compiler and linker to turn written code into working programs. Instead, the Perl interpreter runs whatever script it is pointed to. For this reason Perl is ideal for producing quick solutions, and/or creating prototypes of potential solutions. Perl also provides all the features of the script languages sed and awk, as well as a other features foreign to either of these languages. Therfore, Perl is as powerful as C, but as lightweight and convenient as shell scripts. Thus In particular, it is this cobination of accessibility, versatility and potential for high performance execution, that ultimately make Perl the ideal language for the code base created in this study. Perl's malleability is what primarily allows it to function as an add-on to Marsyas. Specifically, Perl 5 is used for the purposes of this study.

[46]Developed by Larry Wall in 1987 as a general-purpose Unix scripting language to make report processing easier, Perl has since undergone numerous modifications, utimately causing the language to segment into two separate derivatives in 2000: Perl 5 & Perl 6, which are aimed at accomplishing different things.

D. Dependencies

Dependencies of this project include *CMake, Marsyas, WEKA, Alsa, Perl5, FreeType 2, FFTW3, LAPACK, Argtable 2, HDF5, Libsndfile, Libmpg123-0 & MAD.*

VI. Discussion of Results

The results of this study were 63% accuracy on the stock dataset of 10 genres, while accuracy on the combination of the dataset and the "Dummy Plug was 45% accuracy on average. The genres that were accurately classified the most often were *Rock, Pop, Metal,* and *Dance*. The genres that were inaccurately classified the most often were *Reggae, Techno,* and *Jazz*. The genres in between this range were *Blues, Country,* and *Hip-Hop/Rap*. In the section below, *Figure 1.* displays the confusion matrix and overall performance of the network on the main dataset, while *Figure 2.* displays that of the combination of the dataset and "Dummy Plug."

Figure 1. Confusion Matrix —ANN Analysis on Dataset

=== Confusion Matrix == DATASET

a	b	c	d	e	f	g	h	i	j	<-- classified as
4	1	0	0	1	0	0	0	0	0	a = Blues
1	4	1	0	0	0	0	0	0	0	b = Country
0	1	6	1	0	0	0	0	0	0	c = Dance
0	0	1	4	1	0	0	0	0	0	d = HipHop-R
1	0	0	1	4	1	0	0	0	0	e = Jazz
0	0	0	0	1	8	1	0	0	0	f = Metal
0	0	0	0	0	1	5	0	0	0	g = Pop
0	0	0	2	0	1	1	0	0	0	h = Reggae
0	0	0	0	0	1	0	0	7	0	i = Rock
0	0	0	2	1	1	0	0	1	0	j = Techno

63% classified correctly (42/66)

Figure 2. Confusion Matrix —ANN Analysis on Dataset + Dummy Plug

=== Confusion Matrix === DATASET + DUMMY PLUG

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	{	<-- classified as
2	0	0	0	1	0	0	0	0	0	2	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	a = Blues
0	4	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	b = Breakbeat
0	1	2	1	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	c = Chill
0	0	2	2	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	d = Chiptune
1	0	0	0	2	1	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	e = Classical
0	0	0	0	1	0	1	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	2	f = Country
0	0	0	0	0	1	6	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	g = Dance
0	0	0	1	0	0	0	2	2	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	h = Deathcore
0	0	0	0	0	0	0	1	1	0	0	0	0	0	2	0	0	0	1	0	0	0	0	0	0	0	0	i = Djent
0	0	0	0	1	2	0	0	0	1	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	j = Downtempc
0	0	0	0	0	0	0	0	0	1	6	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	k = Dubstep
0	0	0	0	0	0	0	0	0	0	2	2	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	l = Electronicor
0	0	0	0	0	0	0	0	0	0	0	1	4	1	0	0	0	0	0	0	0	0	0	0	0	0	0	m = Experimei
0	0	1	0	0	0	1	0	0	0	0	0	0	1	1	0	0	2	0	0	0	0	0	0	0	0	0	n = Folk
0	0	0	0	0	0	0	0	0	0	0	0	0	0	7	0	0	0	0	1	0	0	0	0	0	0	0	o = Hardcore
2	0	0	0	0	0	1	0	0	0	0	0	0	0	0	4	0	0	0	0	0	0	0	0	0	0	0	p = HipHop-Ra
0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	1	0	0	0	1	0	0	0	q = Industrial
0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	1	0	0	0	0	3	0	0	0	r = Jazz
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	8	0	1	0	0	0	0	0	0	s = Metal
0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	8	0	1	0	0	0	0	0	t = Metalcore
0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	4	0	0	0	0	0	0	u = Pop
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	5	0	0	0	0	0	v = Psychedeli
0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	1	0	0	0	w = Reggae
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	5	0	2	0	x = Rock
0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	y = Samba
0	1	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	2	0	0	0	0	1	0	0	0	z = Techno
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	1	0	0	1	0	1	{ = World

45% classified correctly (78/171)

VII. Conclusion

The fact that the classification accuracy of this framework was dropped dramatically with the addition of the "Dummy Plug" to the dataset, indicates that this comibination goes past the threshold of effective diversity. That is to say that, with a grand total of 165 tracks of 30 genres, and 8-11 musical features extracted per song, at certain points the ANN is very likely to be developing local minimas or more simply unwanted biases towards certain types of data.

One reason for this problem arising is the possibility that within the dataset there are sets of genres that may be too similar to each other, therefore creating a bias in that direction. It should also be noted that the overall effectiveness of this entire approach, especially in real-time use, depends on the accuracy of the initial genre collections. For example, if a *Pop* song was mixed in with the *Dance* genre it could seriously skew the ANNs perception of the *Dance* genre itself. The significance of these findings at first glance is primarily of interest to media databases, in that this approach populates the genres by the shared and different characteristics of the audio files themselves.

Therefore the approach of this study, if applied to existing media databases and audio platforms, is very likely to lead to more accurate database organization, and in conjuction to that, better performance on queries to those databases. Furthermore this approach and the framework created around it, can be seen as a mechanism for these audio platforms to quickly, intelligently, and effectively respond to rapid growth of genres. There are over 1600 genres currently—a number that is steadily increasing with each passing year, and thus the necessity for a response to this phenomena is evident.

References

[1] Bishop, Chris M. "Neural Networks and Their Applications." Ev. Sci. Instrum. 65.1803(1994)

[2] Burred, Juan and Alexander Lerch. "A Hierarchical Approach To Automatic Musical Genre Classification." *In Proc. Of the 6 Th Int. Conf. on Digital Audio Effects* (2003)

[3] Friedmani, Jerome, Trevor Hastie, and Robert Tibshirani. "Additive Logistic Regression: A Statistical View of Boosting." Annals of Statistics (1998)

[4] Freund, Yoav, and Robert E. Schapire. "A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting." (1995)

[5] Li, Tao, and George Tzanetakis. "Factors in Automatic Musical Genre Classification of Audio Signals." *Applications of Signal Processing to Audio and Acoustics, 2003 IEEE Workshop On.* (2003): 143-46.

[6] Soltau, Hagen, Tanja Schultz, Martin Westphal, and Alex Waibel. "Recognition of Music Types." *In Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing* (1998)

[7] Tzanetakis, George, and Perry Cook. "MARSYAS: A Framework for Audio Analysis."(2000)

[8] Tripp, Charles, Hochak Hung, and Manos Pontikakis. "Waveform-Based Musical Genre Classification."

[9] Wang, Avery Li-Chun. "An Industrial-Strength Audio Search Algorithm." *Proceedings of the 4 Th International Conference on Music Information Retrieval* (2003)

[10] Xu, Changsheng, Xi Shao, and Fang Cao. "Musical Genre Classification Using Support Vector Machines." Acoustics,

Speech, and Signal Processing, 2003. Proceedings. (ICASSP '03). 2003 IEEE International Conference on (Volume:5) (2003): 429-32.

[11] Zhu, Jiajun, Xiangyang Xue, and Hong Lu. "Musical Genre Classification by Instrumental Features." *In Int. Computer Music Conference, ICMC 2004* (2004)

Related Works

[1] "An introduction to neural networks" IBM. Web. Jul. 01. 2001.

[2] Anssi Klapuri and Manuel Davy Signal processing methods for music transcription. *Springer* 2006.

[3] B. P. Bogert, M. J. R. Healy, and J. W. Tukey: "The Quefrency Alanysis of Time Series for Echoes: Cepstrum, Pseudo Autocovariance, Cross-Cepstrum and Saphe Cracking". Proceedings of the Symposium on Time Series Analysis (M. Rosenblatt, Ed) Chapter 15, 209-243. New York: Wiley, 1963.

[4] Berry, Wallace. Structural Functions in Music. 1986.

[5] Brynjolfsson, Erik. "The Productivity Paradox of Information Technology" *Communications of the ACM*. 1993. 36 (12): 6677.

[6] Charles Dodge, Thomas A. Jerse (1997). Computer Music. New York: Schirmer Books. p. 82.

[7] Chris C. Bissell and David A. Chapman. "Digital signal transmission (2nd ed.)" *Cambridge University Press*. 1992. p. 64.

[8] Christopher M. Bishop (2006). Pattern Recognition and Machine Learning. Springer. p. 205.

[9] Cortes, Corinna; and Vapnik, Vladimir N.; "Support-Vector Networks", *Machine Learning* 1995. 20.

[10] D. G. Terrell; D. W. Scott. "Variable kernel density estimation". *Annals of Statistics* 1992.

[11] D. Johnston "Transform coding of audio signals using perceptual noise criteria". *IEEE Journal on Selected Areas in Communications* 1988.

[12] Davidson, A.C. "Statistical Models" *Cambridge University Press.* 2003.

[13] Dempster, A.P.; Laird, N.M.; Rubin, D.B. (1977). "Maximum Likelihood from Incomplete Data via the EM Algorithm". Journal of the Royal Statistical Society, Series B 39 (1): 138.

[14] DeRose, Steven J. "Grammatical Category Disambiguation by Statistical Optimization." *Computational Linguistics* 1988.

[15] Fang Zheng, Guoliang Zhang and Zhanjiang Song, "Comparison of Different Implementations of MFCC" *Computer Science & Technology*. 2001.

[16] Ferreira, C. "Gene Expression Programming: A New Adaptive Algorithm for Solving Problems". *Complex Systems* 2001.

[17] Foote, Jonathan, and Shingo Uchihashi. *THE BEAT SPECTRUM: A NEW APPROACH TO RHYTHM ANALYSIS* 2001.

[18] G. Holmes; A. Donkin and I.H. Witten (1994). "Weka: A machine learning workbench". Proc Second Australia and New Zealand Conference on Intelligent Information Systems, Brisbane, Australia. Retrieved 2007-06-25.

[19] Giovanni Bianchi. Roberto Sorrentino. "Electronic Filter Simulation & Design" 2007.

[20] Gray R. IEEE Signal Processing Society, *Distinguished Lecturer Program*

[21] Grey, M. Gordon, W. "Perceptual effects of spectral modifications on musical timbres". *Journal of the Acoustical Society of America* 1978.

[22] H E.Kuffel, W.S. Zaengl, "High Voltage Engineering Fundamentals" *Pergamon Press First edition.* 1992.

[23] Harold W. Sorenson, Marcel Dekker. "Parameter Estimation: Principles and Problems" 1980.

[24] "Interference beats and Tartini tones." unsw. n.p. n.d. Web.

[25] Johnson, Roger A., Advanced Euclidean Geometry. *Dover*. 2007. (orig. 1929): p. 173

[26] Lawrence R. Rabiner "A tutorial on Hidden Markov Models and selected applications in speech recognition". *Proceedings of the IEEE 77* 1989.

[27] Markov, A. A. (1954). Theory of Algorithms. [Translated by Jacques J. Schorr-Kon and PST staff] Imprint Moscow, Academy of Sciences of the USSR, 1954 [Jerusalem, Israel Program for Scientific Translations, 1961; available from Office of Technical Services, United States Department of Commerce] Added t.p. in Russian Translation of Works of the Mathematical Institute, Academy of Sciences of the USSR, v. 42. Original title: Teoriya algorifmov. [QA248.M2943 Dartmouth College library. U.S. Dept. of Commerce, Office of Technical Services, number OTS 60-51085.]

[28] Michael Kearns. Thoughts on Hypothesis Boosting, *Unpublished manuscript* 1988.

[29] ”perl(1): Practical Extraction/Report Language - Linux man page”. Linux.die.net. Jul. 23. 2013.

[30] Pierre Simon de Laplace. ”Analytical Theory of Probability” 1812.

[31] Quinlan, J. R. Induction of Decision Trees. *Machine Learning 1:Kluwer Academic Publishers* 1986.

[32] Reynolds, Douglas. *Gaussian Mixture Models*

[33] Shlomo Dubnov ”Generalization of Spectral Flatness Measure for Non-Gaussian Linear Processes”. *Signal Processing Letters* 2004.

[34] T. Kohonen. Learning vector quantization. *In: M.A. Arbib, editor, The Handbook of Brain Theory and Neural Networks, MIT Press* 1995. 537540.

[35] Waterman, Michael S. ”Introduction to Computational Biology: Sequences, Maps and Genomes” *CRC Press.* 1995.

[36] Wio, S. Horacio, Deza, R. Roberto & Lopez, M. Juan. ”An Introduction to Stochastic Processes and Nonequilibrium Statistical Physics” *World Scientific Publishing*. 2012.

[37] www.acs.gantep.edu.tr/docs/perl-ebook/ch1.htm#WhatIsPerl

[38] www.en.wikipedia.org/wiki/Score_following

[39] www.niedermayer.ca/papers/bayesianlimitations

[40] www.stats.org.uk/hmm/hmm.pdf