

A Data-driven Exploration of Rhythmic Attributes and Style in Music

A Thesis

Submitted to the Faculty

of

Drexel University

by

Matthew K. Prockup

in partial fulfillment of the
requirements for the degree

of

Doctor of Philosophy

May 2016

© Copyright 2016
Matthew K. Prockup.

This work is licensed under the terms of the Creative Commons Attribution-ShareAlike
license Version 3.0. The license is available at
<http://creativecommons.org/licenses/by-sa/3.0/>.

Acknowledgments

Thank you to all those who helped support me through this ridiculous process as well as those who shaped my love for science, engineering, music, and percussion!

I would first like to thank my middle school and high school teachers who supported my scientific career through regional, state-wide and nation-wide science fair and computer programming competitions, and helped shape my general interest in engineering, science, and technology, specifically: Dr. Patricia Waller, Carlen Blackstone, and Brent Ohl.

I would also like to thank all of my music teachers and friends who supported my growth as a percussionist, musician, composer, and teacher while simultaneously being patient with me as I pursued a high level of music performance/composition along with my scientific endeavors, specifically: Rich Hammond, Chris Feist, Chip Bachman, fellow members of the Reading Buccaneers Drum and Bugle Corps, Jeremy Miller, Darrell “Bubba” Goslin, Bruce Denmead, Mark Beecher, Mike Moss, David Ruczhak, and Mike Robinson. I’d also like to thank Matthew Urquhart for his patience and understanding in my attempts to balance my Ph.D. work with our numerous music composition, teaching, and performance endeavors. Long live M^2 !

I offer many thanks to my advisor Youngmoo Kim for instilling in me not only a stellar work ethic, but a genuine appreciation for high quality research and the art of sharing and communicating that work in an extremely clear and understandable manner. This has taken shape in not only research publications and presentations, but in real-life applications and partnerships which have been invaluable experiences (Philadelphia Orchestra LiveNote, Science of Music, Remix Interactive). This achievement has been a long time coming, and I will remember fondly the 4 years of undergrad and the 5 years of graduate school advisorship. Furthermore, I’d like to thank DAn Ellis, Teresa Nakra, Matt Stamm, Yon Visell, and Steve Weber for their additional advice, support, and service on my thesis committee.

me, going above and beyond to help me see the bigger picture in my work and more importantly in life. I look forward to our new shared adventure to the West Coast and the many more adventures to come!

Finally I would like to thank my family, who was was my first and strongest influence in my love for all things music and technology. Growing up, it was not uncommon for us to be wrapped up in conversations about my mother's days as a heavy metal, rock and roll fan (she still is) or my father's days as an engineer and electronic musician. Nor was it uncommon for me to be allowed explore a collection of my fathers analog synthesizers, tinker with his electrical engineering lab equipment, and play my mother's organ so loud the house shook. They were also crazy^{loving} enough to buy me a set of drums (I'm sure the rest of the neighborhood "loved" us). All of this shaped my appreciation for math, science, and music as a single fused entity. I'd like to thank my brother James for the many games of backyard whiffle ball and for the times we could could sit and play or discuss music together. There was always a "sweet spot" before we both stormed out of the room because we each wanted too much creative control. Furthermore, the support and patience of my family and extended family through all of my endeavors is quite an achievement in and of itself, and for that I am forever grateful.

Anyway, I guess I'll end it here because "I need to put my *feet on*, 'make out the light', and go to the store because the milk is *all*."



Jawn

(noun, verb)

1. word used by Philly cats to describe anything and everything.
2. word used in Philadelphia to describe any noun or verb whose appropriate word could not be recalled by the brain in the necessary time.

examples:

- “Can I please have a SEPTA token to ride the trolley?” → “Let me get a *jawn* for the *jawn*.”
- “Write your thesis!” → “*Jawn* that *jawn*!”

urbandictionary.com

Table of Contents

LIST OF TABLES	xii
LIST OF FIGURES	xv
ABSTRACT	xix
1. INTRODUCTION	1
1.1 Contribution 1: A Large-scale Evaluation of Rhythmic Attributes in Audio Signals . .	2
1.2 Contribution 2: Rhythmic Attributes Are Necessary When Defining Genre	3
1.3 Contribution 3: Interpretable Rhythm Feature Spaces	3
2. BACKGROUND	4
2.1 Constructs of Music	4
2.2 Music Signal Processing	5
2.3 Capturing Rhythmic Elements in Audio Signals	8
2.3.1 Detecting Onsets	9
2.3.2 Beats	12
2.3.3 Detecting Metrical and Sub-metrical Structure	17
2.3.4 Rhythmic Pattern Analysis	19
2.3.5 Rhythm and Drum Transcription	22
2.4 Predicting Human-Labeled Attributes	24
2.4.1 Musical Style and Genre	24
2.4.2 Human-Tagged Attributes	27
2.5 Designing Visually intuitive Feature Spaces	31
2.5.1 Spaces: Emotion	32
2.5.2 Spaces: Performance Expression	32
2.5.3 Spaces: Non-Linear	35
3. DATA: LABELS OF RHYTHMIC COMPONENTS AND STYLE	36

3.1	The GTZAN Rhythm and Genre Dataset	36
3.2	The Ballroom Dataset	37
3.3	The Music Genome Project	37
3.3.1	Rhythm Attribute Labels	38
3.3.2	Orchestration Attribute Labels	38
3.3.3	Genre and Culture Labels	40
3.3.4	Testing/Training Sets and Evaluation	40
4.	MACHINE LEARNING	42
4.1	Linear Models	42
4.1.1	Linear Regression	42
4.1.2	Logistic Regression	45
4.1.3	Using Large Datasets	47
4.2	Decision Tree Ensembles	48
4.2.1	Binary Decision Trees	48
4.2.2	Random Forests	51
4.2.3	Gradient Boosted Trees	53
4.2.4	Hybrid Tree Ensemble Models	55
4.3	Supervised Model Evaluation	56
4.3.1	Evaluating Classification Models	56
4.3.2	Evaluating Regression Models	58
4.4	Visualizing High Dimensional Data	59
4.4.1	Basis Decompositions	59
4.4.2	t-Distributed Stochastic Neighbor Embedding	63
4.4.3	Exploring Non-parametric Spaces	65
5.	PRELIMINARY STUDY OF RHYTHM FEATURES AND RHYTHMIC STYLE	71
5.1	Overview	71
5.2	Rhythmic Feature Design	71

5.2.1	Deriving the RSHF	71
5.3	Feature Saliency Experiments	73
5.3.1	Supervised Experiments	74
5.3.2	Unsupervised Experiments	74
5.4	Conclusions	76
6.	RHYTHM ACOUSTIC FEATURES	78
6.1	Overview	78
6.2	Designing Features for Rhythm	79
6.2.1	Rhythm Signal Analysis	79
6.2.2	Rhythm Examples	81
6.2.3	Beat Profile	82
6.2.4	Tempogram Ratio	83
6.2.5	The Mellin Scale Transform	84
6.2.6	Multi-band Representations	86
6.2.7	Rhythmic Feature Evaluation	86
6.3	Analysis of Rhythmic Attributes and Tempo Estimation	88
7.	LEARNING RHYTHMIC COMPONENTS	91
7.1	Approach	91
7.1.1	Rhythmic Attributes of the Music Genome Project	92
7.1.2	Machine Learning Models	93
7.2	Predicting Rhythmic Attributes: Linear Models	93
7.2.1	Experiments	93
7.2.2	Results	94
7.3	Predicting Rhythmic Attributes: Tree Ensembles	95
7.3.1	Experiments	95
7.3.2	Results	96
7.4	Conclusion	97

8. LEARNING GENRE FROM RHYTHMIC ATTRIBUTES	98
8.1 Approach	98
8.2 Data: The Music Genome Project	99
8.2.1 Musical Attributes	100
8.2.2 Genre and Subgenre	100
8.3 Musical Attribute Models of Genre	101
8.3.1 Evaluating the Role of Musical Attributes	101
8.3.2 The Influence of Rhythm and Orchestration in Jazz	103
8.4 Predicting Genre from Audio	104
8.4.1 Timbre Related Features	105
8.4.2 Rhythm Related Features	105
8.4.3 Genre Recognition Experiments	106
8.4.4 Results	107
8.5 Conclusion	113
9. EXPLORING INTUITIVE FEATURE SPACE REDUCTIONS FOR RHYTHM	114
9.1 Motivation	114
9.2 Rhythm Space Reductions	115
9.2.1 Rhythm Attributes and Acoustic Features	115
9.2.2 Learning a 2D space.	116
9.3 Rhythm Space Evaluation	118
9.3.1 Evaluating Rhythmic Saliency	119
9.3.2 New Example Prediction	119
9.3.3 Learning in Other Domains using Rhythm	119
9.4 Results / Discussion	120
9.4.1 Exploring the Rhythm Spaces	120
9.4.2 Evaluating the Rhythm Spaces	121
9.4.3 Evaluating Space Regression	123

9.4.4	Evaluating the Space in Other Domains	124
9.5	Conclusions	127
10.	CONCLUSIONS AND FUTURE DIRECTIONS	128
	APPENDIX A: THE MELLIN SCALE TRANSFORM	131
	APPENDIX B: COLLECTION OF RHYTHM SPACE REDUCTIONS	134
B.1	Human-tagged Attributes and NMF: HA-NMF	135
B.2	Human-tagged Attributes and ICA: HA-ICA	137
B.3	Rhythm Audio Features and NMF: AF-NMF	139
B.4	Rhythm Audio Features and ICA: AF-ICA	141
B.5	Rhythm Audio Features and t-SNE: AF-tSNE	143
	APPENDIX C: VISUALIZING RHYTHM ATTRIBUTES IN MUSIC USING STACKED DENOISING AUTOENCODERS	147
C.1	Introduction	147
C.2	Stacked Denoising Autoencoders	148
C.3	Spatial Organization	149
C.4	Evaluation	150
C.5	Conclusions	151
	APPENDIX D: ATTRIBUTE PREDICTION AND TEMPO ESTIMATION	153
	BIBLIOGRAPHY	157

List of Tables

2.1	Beat tracking evaluation metrics	17
2.2	List of drum transcription literature.	23
2.3	Strengths and weaknesses of tag-based music annotation approaches. (sourced from [1])	31
2.4	List of MIREX Music Emotion Terms	32
3.1	Classes of the The Ballroom Dataset.	37
3.2	Definitions of the rhythmic attributes explored.	39
3.3	Abridged outline of the orchestration attributes explored.	39
3.4	Some of the musical genres and subgenres used.	40
4.1	Should the music group rehearse? The probability of rehearsal can be predicted from past experience based on the imminence of a performance, the time of week, and the time of day.	48
5.1	Accuracies in the style task for the raw RSHF and the best performing reductions. . . .	74
5.2	Accuracies in the duple vs. triple task for the raw RSHF and the best performing reductions	74
6.1	Small-scale tempo estimator evaluation.	80
6.2	Beat Tracking Evaluation	81
6.3	Ballroom dance style classification tasks results.	87
6.4	Tempo estimation results on the Ballroom and GTZAN Rhythm (Genre) Datasets. . . .	88
6.5	Meter and style classification on the Ballroom Dataset.	89
6.6	Mean accuracy of meter and genre multi-class classification on the GTZAN Rhythm Dataset.	90
6.7	Mean AUC of meter and feel attribute prediction on the GTZAN Rhythm Dataset. . . .	90
7.1	The results for rhythm construct learning are shown. Both the AUC and R^2 metrics have a maximum value of 1.0 and lower bounds of 0.5 when predicting a random class (AUC) and 0.0 when predicting the mean of the test labels (R^2).	94
7.2	The rhythmic attribute learning is evaluated with area under the ROC curve (AUC) for classification and R^2 for regression.	97
8.1	Explanations of rhythm and orchestration attributes	100

8.2	Some of the musical genres and subgenres used.	101
8.3	An overview of all models using musical attributes.	102
8.4	Experimental results for ‘Basic’ genre and Jazz subgenre models using musical attributes.	103
8.5	Attributes important to the Jazz subgenres are shown. AUC values greater than 0.70 are bold. The highest performing attribute for each genre is denoted with a *.	103
8.6	An overview of experimental results using audio-based models that utilize both timbre and rhythm features.	108
8.7	Experimental results for the ‘Basic’ genres and Jazz subgenres using audio-based models.	109
8.8	An overview of all experimental results for timbrel (T) and rhythm (R) attributes as well as their combination (T+R). Shown in each row is the mean of all genre classification task within a given group.	111
8.9	Classifying ‘Basic’ genre using timbrel (T) and rhythm (R) attributes as well as their combination (T+R).	111
8.10	Classifying Jazz sub-genre using timbrel (T) and rhythm (R) attributes as well as their combination (T+R).	112
8.11	The results for learning binary (AUC) and continuous (R^2) attributes important to Jazz are shown.	112
8.12	Overall summary of learned attributes.	113
9.1	The human-annotated rhythmic attributes defined by the MGP (top) and the rhythm audio features (bottom).	116
9.2	Mean R^2 and MAE of projecting into the new rhythm spaces ($k = 10$).	124
C.1	The rhythmic attribute predictions in the reduced spaces.	150
D.1	Meter classification of 4 genre meter classes from the GTZAN Rhythm Dataset	153
D.2	Genre classification of 10 genre classes from the GTZAN Rhythm Dataset	153
D.3	Triple Meter classification on the GTZAN Rhythm Dataset	154
D.4	Compound-Duple Meter classification on the GTZAN Rhythm Dataset	154
D.5	Mixed Meter classification on the GTZAN Rhythm Dataset	154
D.6	Duple Meter classification on the GTZAN Rhythm Dataset	155
D.7	Triplet Feel classification on the GTZAN Rhythm Dataset	155
D.8	Swing classification on the GTZAN Rhythm Dataset	155
D.9	Style classification on the Ballroom Dataset	156

D.10 Meter classification on the Ballroom Dataset 156

List of Figures

2.1	A representation of audio feature calculation.	6
2.2	An example of HPSS performed on a short audio example. The power spectrum (left), harmonic component (center), and percussive component(right) are shown.	7
2.3	Computation of Mel-Frequency Cepstral Coefficients	7
2.4	An example of an accent signal or ODF.	10
2.5	An overview of beat tracking by dynamic programming. An example ODF (a) is filtered by convolving it with a Gaussian kernel (b) to create the smoothed ODF (c). The beat consistency weighting function (d) is then slid across the local score function (c) to recursively create the cumulative score function $C^*(t)$ (e). Beat frames (e) are then found by a local maximum $P^*(t)$ of the cumulative score function.	15
2.6	The first table (left) shows an evaluation of all state of the art beat trackers on a comprehensive beat tracking dataset. The references are labeled relative to the publication. The second table (right) is an evaluation of all state of the art beat trackers on a selected subset that has been hand chosen to be difficult to track. (Sourced from [2])	16
2.7	Example beat spectra of <i>Bach, Prelude No. 1</i> (a) generated from Equation 2.7 and <i>Take Five</i> generated from Equation 2.8 (b). A windowed form of the beat spectrum over time can generate the beat spectrogram. A beat spectrogram of Pink Floyd's <i>Money</i> is shown in (c). (Sourced from [3])	20
2.8	Dynamic Bayesian network; circles denote continuous variables and rectangles discrete variables. The gray nodes are observed, and the white nodes represent the hidden variables. This network models the tempo $n = \frac{\text{bars}}{\text{audio frame}}$, position inside bar m , and rhythmic patterns r . (Sourced from [4])	23
2.9	A hierarchical layout of musical genre classification as presented by Tzanetakis. (Sourced from [5])	25
2.10	Semantic-Audio Retrieval of audio from a text query (a) and text descriptors from an audio query (b). (Sourced from [6])	28
2.11	A representation of the V-A space. (Sourced from [7]).	33
2.12	A representation of the Kinematics-Energy space. (Sourced from [8]).	34
2.13	An example of the Tempo-Loudness Space. (Sourced from [9])	34
2.14	t-SNE projections of acoustic features in 2 dimensions for (a) duple/triple designation and (b) individual style classifications.	35
4.1	An example of linear regression.	44
4.2	A sigmoid function	45

4.3	An example of logistic regression	47
4.4	An example of standard gradient descent (left) and stochastic gradient descent (right) for a quadratic bowl-shaped gradient surface. Arrows depict the model error trajectories.	48
4.5	A binary decision tree is generated. The leaves terminate with the probability of the positive class label (a rehearsal) present after traversal.	49
4.6	Random forest tree ensemble examples for classification (a) and regression (b).	53
4.7	A Gradient Boosted Tree example.	54
4.8	Leaf activations of each tree in the ensemble become features for another model. This learns interactions of branches.	56
4.9	t-SNE maintains small distances, but can expand further ones.	65
4.10	2 dimensional t-SNE projection of a toy example.	66
4.11	K-means clustering (k=4) of the toy example in the t-SNE space	67
4.12	Nearest neighbors of the cluster means in t-SNE space can be used to approximate cluster means in the original feature space.	67
4.13	2D Projection of MNIST using t-SNE. Ground truth labels are shown in different colors.	68
4.14	2D Projection of MNIST using t-SNE. k-means clusters and cluster centers (letter labels) are shown.	69
4.15	The means of the cluster center nearest neighbors in the original feature space are shown.	69
4.16	Rhythm features are reduced using t-SNE and candidate points are selected (left). From those candidate points, nearest neighbor means are computed in the rhythm feature space (right). A subset of 5 (from 15) means is shown.	70
5.1	RSHF design and calculation.	72
5.2	Rhythmic Style Histogram Feature.	73
5.3	The projections of the raw RSHF feature into 2 dimensions for (a) duple/triple designation and (b) individual style classifications.	75
5.4	The percentage of each style label in each k-means cluster for (a) $k = 2$ and (b) $k = 4$	76
6.1	These patterns define the Samba, Tango, and Jive rhythmic styles for drum set.	82
6.2	Examples of the BPMEAN Feature are shown. On the X axis, 0.0 denotes the beat, 0.5 denotes the 8th note, and 1.0 is the lead-in to the next beat.	82
6.3	Note values and multiples of each dimension of the Tempogram Ratio feature.	83
6.4	Examples of the TGR feature.	84
6.5	Examples of the MST and MST_DCT Feature.	85

6.6	Ballroom Dataset confusion matrices of the Mellin Transform and Tempogram features	88
8.1	An overview of the feature types used experiments performed.	99
9.1	Audio features and human annotations are reduced to a set of n components and activations. The 2 most salient components are selected and their activations are normalized to create a 2D space.	117
9.2	Audio features (AF) and human annotations (HA) are reduced to a set of components and activations. Shown here are the two selected components (<i>supervised component selection</i>) for the ICA and NMF reductions.	121
9.3	To visualize the audio feature information in the t-SNE space, a set of query points (A,B,C, etc.) is selected. Local structures can be explored by looking at audio features means of query point neighbors in the t-SNE space.	122
9.4	Mean AUC of predicting binary rhythm attributes across all trials.	123
9.5	Mean R^2 and MAE of predicting continuous rhythm attributes across all trials.	124
9.6	Experimental results for classifying “Basic” genre, Jazz subgenre, and geo-cultural factors using spaces designed to represent rhythmic attributes.	125
9.7	A selection of rhythm and genre labels for the HA-ICA, AF-ICA, and AF-tSNE spaces.	126
A.1	The accent signal, its autocorrelation, and exponential sampling.	132
A.2	The exponential weighting.	132
A.3	The Mellin Scale Transform, its DCT, and the normalized median-removed DCT	133
B.1	Color mappings for attribute and genre plots.	134
B.2	Components for NMF reductions derived from human annotations of rhythm.	135
B.3	Rhythm attribute colorings for NMF reductions derived from human annotations of rhythm.	135
B.4	Genre colorings for NMF reductions derived from human annotations of rhythm.	136
B.5	Components for ICA reductions derived from human annotations of rhythm.	137
B.6	Rhythm attribute colorings for ICA reductions derived from human annotations of rhythm.	137
B.7	Genre colorings for ICA reductions derived from human annotations of rhythm.	138
B.8	Components for NMF reductions derived from rhythm acoustic features.	139
B.9	Rhythm attribute colorings for NMF reductions derived from rhythm acoustic features.	139
B.10	Genre colorings for NMF reductions derived from rhythm acoustic features.	140
B.11	Components for ICA reductions derived from rhythm acoustic features.	141
B.12	Rhythm attribute colorings for ICA reductions derived from rhythm acoustic features. .	141

B.13 Genre colorings for ICA reductions derived from rhythm acoustic features.	142
B.14 Local component locations for t-SNE reductions derived from rhythm acoustic features.	143
B.15 Components for t-SNE reductions derived from rhythm acoustic features.	144
B.16 Rhythm attribute colorings for t-SNE reductions derived from rhythm acoustic features.	145
B.17 Genre colorings for t-SNE reductions derived from rhythm acoustic features.	146
C.1 Overview of stacked autoencoder models. N is the dimensionality of feature input (attributes: $N=10$, audio: $N=372$).	148
C.2 Spaces with selected label colorings from each model learned from human-annotated attributes (left) and audio features (right)	149
C.3 Selected label colorings for embeddings learned from human-annotated attributes (top) and audio features (bottom)	151

Abstract

A Data-driven Exploration of Rhythmic Attributes and Style in Music

Matthew K. Prockup

Youngmoo E. Kim, Ph.D.

Humans identify with three basic components of music: melody, harmony, and rhythm, in order to describe and differentiate songs. With these simple components, one can recognize higher level concepts such as the style and other expressive elements of a piece of music. In this thesis, I explore rhythmic components and their relationships to each other, to genre, and other geo-cultural factors (i.e., language) through data driven approaches using audio signals. Working in conjunction with Pandora[®], I employ a corpus of over 1 million expertly-labeled audio examples across many rhythmic styles and genres from their flagship *Music Genome Project*[®]. Each song is labeled with more than 500 attributes of rhythm, instrumentation, timbre, and genre.

In order to model the rhythmically related information from audio signals, I implement a set of novel and compact rhythm-specific acoustic features. They represent beat-level and meter-level information as well as elements of rhythmic variation and pulse stability. First, the acoustic features are used to predict the presence of human-annotated attributes of the meter and rhythmic feel (i.e., swing). Previous work has studied the general recognition of rhythmic styles in music audio signals, but few efforts have focused on the deconstruction and quantification of the foundational components of global rhythmic structures. Second, I focus on rhythm and its relationship to genre. Genre provides one of the most convenient categorizations of music, but it is often regarded as a poorly defined or largely subjective musical construct. I provide evidence that musical genres can to a large extent be objectively modeled via a combination of musical attributes, with rhythm playing a significant role. Finally, through a set of unsupervised machine learning experiments that employ both the human-labeled attributes and acoustic features, a set of low-dimensional, perceptually-motivated rhythm spaces is designed. These spaces provide grounded and visual insight into the relationships between rhythmic attributes and rhythmic styles.

Most previous work strives to automatically predict a specific phenomena (i.e., genre) without a contextual understanding of why a label is applied. This work is motivated by largely the same idea, however, I aim to not only predict the phenomena but also understand the components used to construct it. This opens up the door to a more grounded and intuitive understanding of these components and how they interact to create the different styles of music we enjoy.

Chapter 1: Introduction

When we listen to music, we are able to understand complex interactions and relationships of musical attributes that have very little quantifiable justification. It is easy for us to hear similarities and differences between songs or genres, but it is sometimes difficult to articulate and define those differences and apply them to an understanding of our individual preferences. Many of the attributes that lead to our enjoyment of music revolve around the rhythm (i.e., meter, rhythmic feel). In this thesis I outline methods to both capture rhythmic information and define its importance in broader musical contexts such as genre. Furthermore, I employ scalable, data-driven approaches, using information derived directly from the music audio signal, and develop models that leverage a corpus of more than 1 million expert-labeled examples from *Pandora's Music Genome Project* (MGP).

A grounded representation of rhythm can be influential to many areas of research and practice. For example, we can uncover rhythmic organizations previously only speculated, and discover attributes important to rhythmic style. This information can be used to develop a set of tools for musicologists to answer a wide range of research questions by exploring co-occurring factors (i.e., rhythm vs. genre and language). One can ask questions such as, “Does all jazz contain swing?”, and discover that some sub-genres of jazz do not (i.e., Afro-Cuban), and explore that sub-genre more deeply to uncover its global influences (i.e., Latin-American, Spanish language, African rhythm). An intuitive organization of rhythmic similarity can also be employed for automated playlist generation and music discovery. A playlisting algorithm could use rhythmic similarity (along with beat matching) when transitioning between music tracks, attempting to keep a consistent rhythmic pulse throughout. In another vein, by incorporating user feedback labels, we can develop a model of users' rhythm preferences within the styles of music they enjoy. In this section I will introduce the contributions presented in this thesis that lay some of the groundwork to transform each of these theoretical situations into real-life possibilities.

1.1 Contribution 1: A Large-scale Evaluation of Rhythmic Attributes in Audio Signals

Rhythm is one of the fundamental building blocks of music, and perhaps the simplest aspect for humans to identify with. But constructing compact, data-driven models of rhythm presents considerable complexity even when operating on symbolic data (i.e., musical scores). This complexity is compounded when developing algorithms to model rhythm in acoustic signals for organizing a large-scale library of recorded music. Previous work has studied the general recognition of rhythmic styles in music audio signals, but few efforts have focused on the deconstruction and quantification of the foundational components of global rhythmic structures. Through the design and implementation of targeted acoustic features, I first try to capture low-level rhythm descriptors from music audio signals. Each of the descriptors is computed from an *accent signal*, a generic measure of change over time in the audio signal where high points of change denote the presence of a new musical event. From this signal we can capture rhythmic attributes by exploring information related to the timing of these events. The descriptors presented in this thesis capture information at two levels. The *Tatum*-level explores information that occurs at the lowest perceivable pulse. Information at this level can uncover constancies or deviations in micro-timings (i.e., swing). Secondly, at a greater time-scale, *meter*-level features can capture information relating to higher-level rhythmic structures. This information can be used to capture the musical meter (time-signature) and other broader rhythmic patterns.

In Chapter 6, I outline and evaluate a set of novel rhythm descriptors. Using the new rhythm descriptors I look deeper into the compositional constructs of meter and rhythmic feel. Leveraging the rhythm-related labels from *Pandora's MGP*, a set of models are developed to predict meter, swing, shuffle, syncopation, danceability, and back-beat strength across more than 1 million examples. Each of the developed models are designed to be both simple and scalable due to the large amount of data. The evaluated models are both linear (*Linear Regression, Logistic Regression*) and non-linear (*Gradient Boosted Trees, Random Forests*). These experiments are outlined further in Chapter 7.

1.2 Contribution 2: Rhythmic Attributes Are Necessary When Defining Genre

With a more informed understanding of rhythmic attributes, I then explore their relationship to genre. Genre provides one of the most convenient categorizations of music, but it is often regarded as a poorly defined or largely subjective musical construct. In this area of work, I provide evidence that musical genres can to a large extent be objectively modeled via a combination of musical attributes. A data-driven approach is employed utilizing a subset of 48 hand-labeled musical attributes comprising instrumentation, timbre, and rhythm, again leveraging the scope of the *Pandora MGP*. Furthermore, using the acoustic features previously developed (Chapter 6), genre will be modeled directly and through audio-driven models of the hand-labeled musical attributes. This work shows that musical attributes are necessary to the definitions of genres and that rhythm plays a significant role in those definitions. This body of work is outlined further in Chapter 8.

1.3 Contribution 3: Interpretable Rhythm Feature Spaces

Different attributes of rhythmic meter and feel combine in complex and creative ways to create cohesive, distinct, and easily recognizable styles. In the final portion of the thesis I attempt to not only learn compositional and genre attributes, but understand their overarching relationships. I once again leverage the scope of *Pandora's MGP* to create a set of grounded and intuitive low-dimensional visual projections from human-annotated attributes and acoustic features. Each of the projection candidates is first evaluated on how well it represents the rhythmic attributes through a set of attribute prediction tasks. Second, each space is evaluated on the generalizability of its projection, showing its ability to generalize to new examples through acoustic feature similarity. Finally, I explore the efficacy of each space at representing other potentially rhythm-related attributes such as genre and sub-genre, as well as geo-cultural aspects such as language. An evaluation of these new visual rhythm spaces is found in Chapter 9.

Chapter 2: Background

There is a large body of work in Music Information Retrieval (Music-IR, MIR) that explores music’s symbolic representation, its audio signal, and related human-tagged attributes. In this chapter, I provide an overview of some of the work I build upon in this thesis. First, I will give a brief summary of musical constructs in Section 2.1, followed by an introduction to signal processing methods used for Music-IR in Section 2.2. In Section 2.3, I will dive deeper into methods used to capture rhythmic elements from audio signals. Section 2.4 will describe methods used to predict human-labeled music attributes. Finally, in Section 2.5, I will conclude with relevant work regarding the construction of visually intuitive feature spaces.

2.1 Constructs of Music

Music is comprised of an organization of pitches (melody and harmony) in temporal patterns (rhythm). This section outlines some of the constructs of music necessary for a domain-specific understanding of the work presented.

The rhythmic component of music refers to the organization of musical events in time. In music, the most basic aspect of rhythm is the *beat*. The beat is the continuous repetitive pulse felt throughout the music. Beats are also organized into *measures*, which are repetitive groupings of a defined number of beats. The first beat in a measure is known as the *downbeat*. The division between measures (or bars) is the *barline*, which refers to the visual lines used to separate these groupings in a musical score. The number of beats in a measure versus their relative duration is the *meter*. Groupings of two or four is *duple meter*. Duple meter can also sometimes have a *cut-time* feel, where the pulse is felt half as fast. Groupings of three is *triple meter*. A *compound-duple meter* refers to an even number of groupings (usually two or four) of three notes. The pulse is felt at the start of each grouping. An *odd meter* refers to the consistent grouping of an odd number of beats (other than 3). The pulse felt among these notes can vary within a measure, but usually remains

consistent measure to measure. In addition to meter groupings at the beat level, there are also *ticks* or *Tatum* (named for the jazz musician Art Tatum) at the sub-beat level. The tick or Tatum refers to the smallest rhythmic interval or distance in a musical phrase. In some rhythms, onsets at the anchor points of beat and meter are not explicitly present, however the concept of beat and meter still exist. The complexity of beat and meter in relation to the absence of these anchor points is known as *syncopation*. It can also be described as confusion created by early anticipation of the beat or obscuring meter with emphasis against strong beats [10].

When analyzing music, it is also important to consider the melody and harmony, which are comprised of an ordered structuring of pitches. In an *octave* there are 12 logarithmically spaced pitches, where each successive note frequency f is related by $f_{n+1} = f_n 2^{\frac{1}{12}}$. More generally $f_{n+x} = f_n 2^{\frac{x}{12}}$, where x denotes the number of successive note steps (semitones). If each note is played in succession, it forms the *chromatic* scale; usually, notes are not played in succession and not all notes are used in a piece. The *mode* denotes which notes to play and which to skip relative to the given *key*. Two of the most important concepts in melody and harmony are the key and mode. The key of a piece of music defines the pitch around which the music is centered. It defines the tonal center around which the mode is constructed. The mode denotes the ordering of pitches in a scale around the key.

2.2 Music Signal Processing

Audio signal analysis can occur in both the time and frequency domains. For rhythm representation, both are employed. One method widely used to learn temporal repetition is *autocorrelation*. The general form of autocorrelation is shown in Equation 2.1. It is the sum of a given discrete signal $x[n]$ multiplied by the complex conjugate of a shifted form of itself based on a lag l . The resulting calculation will have peaks where the original and shifted signals align. This emphasizes periodicities in the signal, which may correlate with periodic events in rhythm such as the tempo and meter.

$$R_{xx}[l] = \sum_n x[n] \bar{x}[n-l] \quad (2.1)$$

Analysis in the frequency domain stems first from the *Discrete Fourier Transform (DFT)* and the *Short-Time-Fourier-Transform (STFT)*. Given a window of audio samples, the DFT will describe the frequencies contained in the signal relative to the window size and the audio sample rate. The STFT is the DFT shown over time. The STFT is also called the *spectrogram*. There is an important trade off when considering window size and analysis size. The window size must be at least as many samples as the frequency analysis size. Smaller windows allow for more fine grained time resolution. Alternatively, larger windows allow for higher frequency resolution, but lower time resolution. To accommodate for decreased time resolution, successive windows usually overlap one another, and are multiplied by a weighting function that allow the overlapped windows to still sum to the original signal. This allows for a finer time resolution with a greater frequency resolution. However, some smearing effects over time still occur [11].

In order to describe audio, many features related to the time and frequency domains are used instead of the raw waveform or spectrogram. These are calculated over time, usually in a windowed fashion. The audio can then be described by sequence of these feature representations or simple statistics of that sequence. This process is shown in Figure 2.1

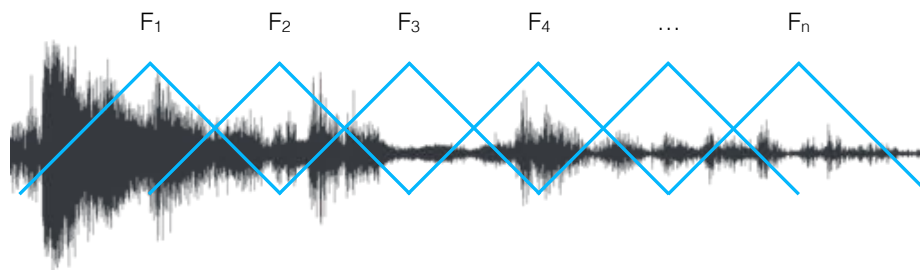


Figure 2.1: A representation of audio feature calculation.

It is sometimes necessary to separate harmonic and percussive sounds when analyzing the audio spectrum. In order to accomplish this there is a process known as *Harmonic Percussive Source Separation (HPSS)*. Harmonic components in a spectrogram tend to be continuous horizontally, at discrete points in frequency. Percussive components are the opposite; they are usually discrete in time and contain wide band noise that creates continuous vertical lines in frequency. This separation can be accomplished by methods of probabilistic modeling as in [12] as well as simpler methods such

as median filtering in [13]. An example of HPSS is shown in Figure 2.2. Notice the separation of the horizontal and vertical lines.

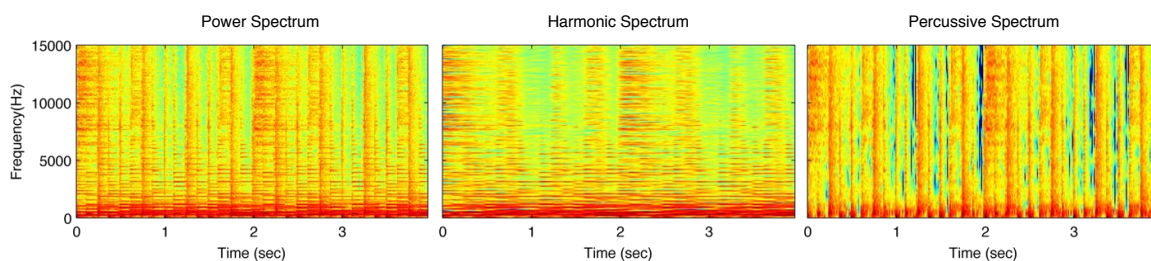


Figure 2.2: An example of HPSS performed on a short audio example. The power spectrum (left), harmonic component (center), and percussive component(right) are shown.

One of the most commonly used features in Music-IR are *Mel-Frequency Cepstral Coefficients* (*MFCC*). This feature is a measure of the frequency domain envelope of an audio signal. It was originally designed for speech recognition and has been widely adopted by the Music-IR community to describe timbre of music and audio. One of the most important components of the MFCC is the Mel-Spectrum, which is derived from a Mel-Spaced filter-bank. This is a representation of the frequency domain that, through empirical tests, was shown to more closely represent the perception of pitch in the human auditory system. The Mel-Spectrogram is the resulting spectrogram obtained from this filter-bank. This spectrogram is then scaled by squaring it and taking the log (log power mel spectra). From there, the Discrete Cosine Transform is taken (DCT). The coefficients from this transform are the MFCC's. In practice the first 13-20 coefficients are used. This process is shown in Figure 2.3.

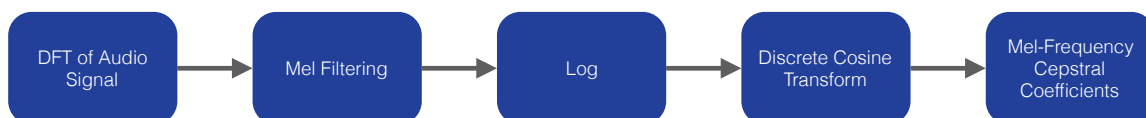


Figure 2.3: Computation of Mel-Frequency Cepstral Coefficients

A filterbank can also be designed to represent the frequencies of musical notes, with filter bin centers defined to correspond to specific note frequencies. A *Constant Q Filter Bank Transform* (*CQT*) arranges the frequency range of successive filter bins relative to constant multiple of the previous. This allows for a logarithmic spacing of filter bins. Because musical notes are also spaced

logarithmically ($2^{1/12}$), this type of filter bank allows for the capture of frequencies that have musically relevant weights. The application of the Constant Q Filter Bank over time is called the *Constant Q Filter Spectrogram*. Another, more compact capture of musical pitches is the *chroma* feature. The *chroma* captures the weighting of each pitch class (A, A#, B ... G, G#), independent of octave. With this feature, it does not matter in which octave a pitch exists, only that it is present. The application of the *chroma* feature over time is called the *chromagram*.

Both chroma and the CQT are not direct transcriptions, however. When played by a physical musical instrument, all notes contain *harmonics*. In the spectrum, there is weight at the fundamental frequency (the note being played), as well as weights at integer multiples of that frequency. These harmonics end up in bins of other notes as well because they are integer multiples (linearly spaced) and successive pitches are logarithmic (log spaced). While the exact transcriptions of notes in CQT and chroma can be clouded by the presence of these harmonics, they are still very robust representations of musical melody and harmony.

2.3 Capturing Rhythmic Elements in Audio Signals

Work by Longuet-Higgins in 1982 made some of the first attempts to quantify how humans interpret rhythm. It was stated that the assignment of rhythmic interpretation to a metrical piece of music calls for the knowledge of the underlying meter and the parsing of note values according to this meter [14]. In a later article, the following propositions are presented:

1. Any given sequence of note values is in principle rhythmically ambiguous, although this ambiguity is seldom apparent to the listener.
2. In choosing a rhythmic interpretation for a given note sequence, the listener seems to be guided by a strong assumption: if the sequence can be interpreted as the realization of an non-syncopated passage, then that is how they will interpret it.
3. Phrasing can make an important difference to the rhythmic interpretation that the listener assigns to a given sequence. Phrasing can therefore serve as a structural function as well as a purely ornamental one.

Much of the work on rhythm in MIR attempts to define components of these fundamentals as well as their use in combination in order to intuit more complex rhythmic structures. In this section I will outline some of the building blocks of rhythmic analysis. First I will focus on the detection of musical events using onset detection and beat tracking. With these building blocks, another body of work is presented that tries to both find and transcribe rhythmic patterns within the music. Certain aspects of musical rhythm define the structure and style of a piece as well as the culture from which it originates, so components of rhythm and style are important aspects of music to quantify.

2.3.1 Detecting Onsets

A task known as *onset detection* is an area of work that focuses on finding the positions, or *onsets*, where musical notes begin. It is an important step when trying to detect musical *beats*, which are the most basic reference to a song's pulse. Much work in onset detection has come at the service of beat tracking and is usually a preprocessing step. In this section, I will describe onset detection algorithms that stand on their own. Methods that work to serve specific beat tracking methods directly will be described in Section 2.3.2.

Onset detection data often exists in two forms, one of which is usually calculated from the other. The first form is the *onset detection function (ODF)* or *accent signal*. This is found by trying to measure how much a given feature of an audio signal is changing. The goal is to find a signal that is sparse with peaks or spikes that occur at musical events. From this signal, onset positions can be calculated by some form of peak-picking. An example of an onset detection function is shown in Figure 2.4.

While work in onset detection usually serves as input to a beat tracker, there is a body of work that focuses on this task specifically. In work by Klapuri an onset detection system that takes psychoacoustic knowledge into account is presented [15]. This is done by splitting the signal into multiple perceptually motivated frequency bands, and trying to find onsets over multiple bands and combining their results. Most previous systems utilized a peak picking method on the amplitude envelope. The Klapuri method and the Scheirer method (part of a beat tracker) [16] are the first to take advantage of this psychoacoustic information. The Klapuri method also goes a step further in

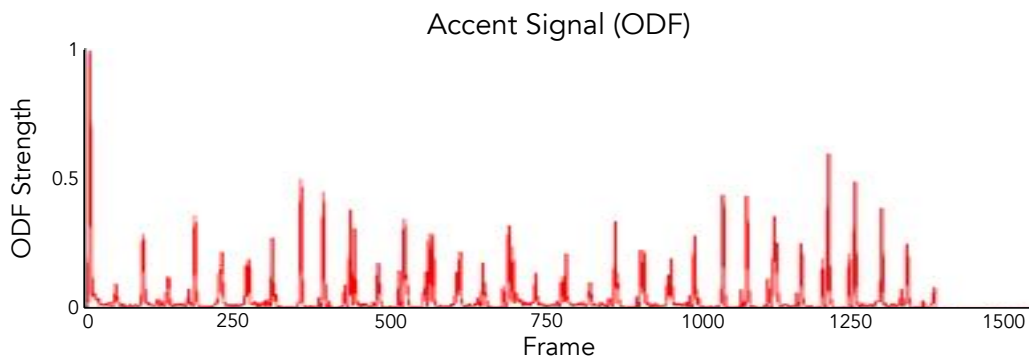


Figure 2.4: An example of an accent signal or ODF.

the selection of onsets. It processes the output of the psychoacoustic filterbank by taking the first order difference of each band. Instead of showing a representation of how much energy is contained in the present frame, it shows how that frame has changed from the previous one. This is shown to be a better predictor of when musical events occur, especially in music with instruments that lack strong percussive attacks. Dixon later expanded upon this. In his work, a much larger set of input features is used. His detection system included an amplitude based ODF as well as a binary mask of the local peaks and a multi-band spectrum with each channel containing an ODF and local peak mask. The calculated onset times are then compared to ground truth onsets recorded from a Bösendorfer piano, and the delay of detection to actual onset time is learned and incorporated into the model [17].

Early methods in onset detection were energy based, meaning they relied on the amplitude envelope or the magnitude of the complex spectrum and its various perceptually warped representations. Work by Bello takes both the magnitude and phase of a signal into account. While magnitude based onset detection is energy or energy-difference based, phase based onset detection relies on the phase of estimate of a signal to be somewhat constant. Large deviations in phase (*Phase Deviation*, *PD*) denote the position of new musical events. The authors claim that by combining both energy and phase based methods, they can produce a function that is sharp at the positions of onsets and smooth everywhere else [18]. Dixon improves on this by further refining the onset detection function

used. Because phase deviation can be noisy due to uninformative frequency bands, it is weighted by the magnitude of the respective frequency. This results in a similar ODF to Bello’s, however with a different fusion method. Another form of an ODF is computed via *Spectral Flux* (SF) [19]. This is shown in Equation 2.2 where X is the magnitude spectrum and $H(x)$ is the half-wave rectifier function $H(x) = \frac{x+|x|}{2}$.

$$SF(n) = \sum_{k=-\frac{N}{2}}^{\frac{N}{2}-1} H(|X(n, k)| - |X(n-1, k)|) \quad (2.2)$$

It has been shown that many of these onset detection methods perform well, and the best ones are not significantly different from one another. However, due to SF’s simplicity, ease of computation, and effectiveness, the results show that it is the best overall choice for a system-wide ODF.

In later work, Gouyon shows that not all onset detection methods are optimal for all types of music [20]. He shows that when beat-tracking certain types of music, the algorithms respond differently to certain ODFs. This motivates future systems to look at ODF functions optimized for their specific task, or for the necessity of a hybrid system that uses different features for different types of music.

Later work by Böck presents a method of onset detection known as *SuperFlux* [21]. It is an enhanced version of the traditional Spectral Flux algorithms. Rather than taking the difference of consecutive frames, a lag was introduced, making it possible to take the difference of the previous n^{th} frame. It is also designed to be causal, not taking into account any future information, which will allow it to work in real time. The spectrogram is log filtered via a CQT and smoothed over frequency to reduce micro-pitch deviations (i.e., vibrato). This change to Spectral Flux is shown in Equation 2.3.

$$SF'(n) = \sum_{k=1}^{\frac{N}{2}} H(|X_{log, filt}(n, k)| - |X_{log, filt}(n-\mu, k)|) \quad (2.3)$$

The resulting $SF'(n, k)$ can be improved by a moving maximum as shown in Equation 2.4.

$$X_{log, filt}^{max}(n, m) = \max(X_{log, filt}(n, m-1 : m+1)) \quad (2.4)$$

This resulting spectrogram is then summed over frequency in Equation 2.5. This becomes the *SuperFlux* onset detection function. As before, H is the half-wave rectifier function $H(x) = \frac{x+|x|}{2}$.

$$SF^*(n) = \sum_{m=1}^{m=M} H(X_{log, filt}(n, m) - X_{log, filt}^{max}(n - \mu, m)) \quad (2.5)$$

The ODF shown in Figure 2.4 was obtained using this SuperFlux algorithm.

While many methods of onset detection rely on advanced processing of a signal, there has been little work in using algorithms to learn locations of onsets. One such study by Degara uses knowledge about the structure of music to generate a model for detecting onsets. A Hidden-Markov-Model is used to learn the temporal regularity in successive onsets. This information is then used to judge whether an estimated onset makes sense musically and magnifies or down-weights its importance accordingly [22]. Similarly, work by Böck uses advanced machine learning in order to detect onsets. He introduced a neural-network based approach for peak picking onsets that can be employed with any of the previous ODF functions [21].

2.3.2 Beats

Some of the earliest work in beat tracking was used as a means for performance tracking and real time music accompaniment. Many also used symbolic representations of the music with methods designed for specific tasks as well as all music in the general sense. Some of this early work is presented in [14, 23, 24]

One of the first systems that tries to beat-track an audio signal is presented by Goto [25]. It finds local maxima in time and frequency from the audio spectrum and learns if those peaks are bass-like onsets or snare-like onsets. In order to estimate beat locations, it uses a combination of heuristics that were designed to keep IOI consistent and ensure the snare-drum and bass-drum beats alternated. A few years following this work, Dixon introduces a beat tracking method that was also used for rhythmic analysis and transcription. This method similarly found onsets in the audio spectrum [26]. Beats are found by a clustering method, automatically grouping IOIs between onsets. This naturally provides groupings that are multiples and sub-multiples of the beat. These

groupings can be used to learn an estimate of the musical tempo, and with an estimate of the tempo, the locations of the beats can be derived. Beat tracking algorithms were further advanced in early work by Scheirer in [16]. In this work each individual frequency band is analyzed as an accent signal. Each is summed with a resonator, or *comb filter*, that emphasizes impulses at a given tempo. When this resonator signal corresponds well with the accent signal of a given frequency band, the musically relevant positions in the accent signal are emphasized. Each band resonance is treated separately in order to estimate tempo and the beat.

As work in beat tracking progressed, it evolved into its own subfield, and broke away from purely serving performance tracking systems. Work by Gouyon in [27] and [28] uses a rhythmically and tempo dependent method for determining the time quantization level of an accent signal. This time quantization level is based on the *tick*, more commonly known as the *Tatum*. The Tatum is the smallest rhythmic separation between successive notes in a piece of music. Using the autocorrelation of low-level energy features quantized at these Tatum levels, a tempo estimate for the beat is found. A beat-phase is then calculated with a series of comb filters similar to [16]. This method also does not require detecting individual onsets; beats are found directly from the accent signals.

As the research progressed, a set of systems emerged that are still used as the basis for beat tracking systems today. These systems were the parallel work of Davies [29, 30], Dixon [31, 32], Ellis [33], and Oliveira [34, 35]. The Davies system fuses a general model and context-dependent model of beat tracking. The general model is based on comb filters similar to previous approaches. The context dependent model relies on statistics of previous beat estimates and attempts to maintain continuity. The algorithm switches between these two models in order to both find beat phase and remain consistent, giving greater confidence to estimates that occur at regular intervals. The Dixon system, known as *Beatroot* is similar to his previous system, however the ODF has been replaced with spectral flux, and is shown to perform similarly to the other state of the art methods of its time [31, 32]. Oliveira expanded the work of Dixon by extending the Beatroot system to make it run in real-time, forcing it to be causal and work with continuous input. This system, known as *IBT*, does a small pre-tracking step to get a general estimate of beat period. It then employs a multi-agent

system to estimate new beats. Each agent, starting the the pre-tracking estimates, outputs a set of hypotheses regarding possible beat periods and phases. These hypotheses are then ranked and accepted or rejected. The beats are then estimated in real-time [34].

Ellis in [33] has a different approach. His beat tracking approach uses *dynamic programming* in order to estimate consecutive beats. Similar to previous methods, the accent function (ODF) is the first order difference of a mel-filtered spectrogram. This difference is then half-wave rectified and the remaining positive values are summed over frequency bands. The resulting signal is then high-pass filtered to make it locally zero-mean. It is then smoothed by convolving it with a 10ms gaussian window, which becomes the accent signal. Tempo is then estimated through autocorrelation that is weighted with an exponential window with a long upper tail and a maximum weighting value at 120bpm. With an estimate of tempo, beats can be found through dynamic programming. In the dynamic program, a recursive cost function is formulated such that spikes in the accent signal within a widow of the tempo period are weighted relative to where the next beat should occur. This function recursively iterates to create a cumulative score function $C^*(t)$, which is a step-like version of the ODF with local maxima at probable beat locations. This dynamic programming update is shown in Equation 2.6.

$$C^*(t) = ODF(t) + \max_{\tau=0\dots t} \{ \alpha F(t - \tau, \tau_p) + C^*(\tau) \}$$

where, (2.6)

$$F(\Delta t, \tau) = -\ln \left(\frac{\Delta t}{\tau} \right)^2$$

A local maxima filter is then passed over this cumulative score function to assign a discrete beat frame time to all frames $P^*(t)$. The visual representation of this dynamic programming method is shown in Figure 2.5.

Among the current state of the art in beat tracking are a few methods formed around probabilistic models rather than systems built purely on signal processing, filtering, and peak picking. Peeters [36], Böck [37, 38], and Krebs [39] have developed models using HMMs and neural networks. In the Peeters method, a set of beat templates is learned through linear discriminate analysis (LDA).

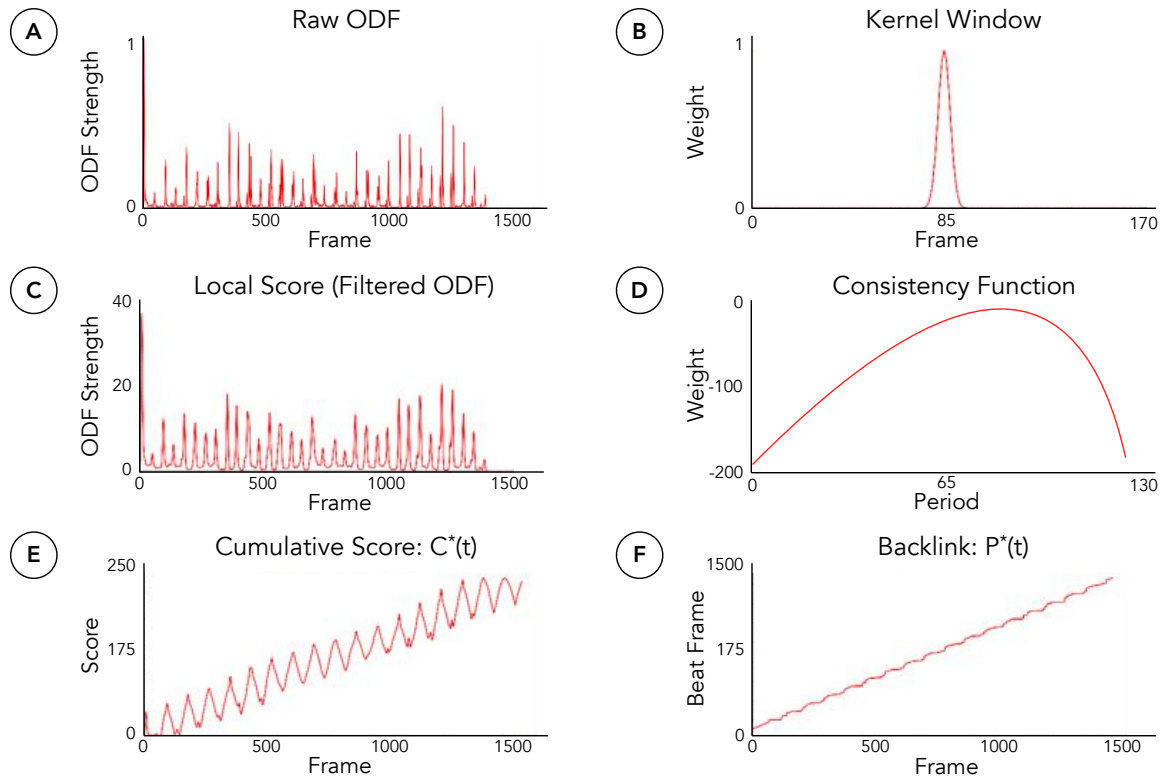


Figure 2.5: An overview of beat tracking by dynamic programming. An example ODF (a) is filtered by convolving it with a Gaussian kernel (b) to create the smoothed ODF (c). The beat consistency weighting function (d) is then slid across the local score function (c) to recursively create the cumulative score function $C^*(t)$ (e). Beat frames (e) are then found by a local maximum $P^*(t)$ of the cumulative score function.

This learns the most unique beat templates. These beat templates are derived from the derivatives of chroma and spectral balance features over time. Occurrences of the beat templates are used to derive the observation probabilities of an HMM. The hidden states are the beat times and their associated positions within a bar. A reverse Viterbi method is used to decode these hidden states [36]. Work by Böck introduces an approach that employs a recurrent neural network (RNN). A recurrent neural network is a network that contains both long-term and short-term memory, allowing it to obtain temporal context. Spectral difference audio features act as inputs to the network. The output is a learned beat activation function. With the activation function, Tempo is found through autocorrelation, and phase is found through its highest peak. This estimation of tempo and phase

allows it to delete spurious beats in the activation function. The remaining activations become the beat estimates [37].

In addition to research surrounding the design of beat tracking methods, there has been recent work in trying to define evaluation metrics for fair comparison. One such body of work is presented by Holzapfel in [2]. This paper uses mutual agreement of many state of the art systems in order to judge the difficulty of certain pieces of music, with or without previous ground truth beat annotations. This helps the field better understand where it is succeeding as well as where it is failing in order to better inform new approaches. In this work, Holzapfel also introduces a method that allows for the fusion of multiple trackers in order to better track more difficult pieces. The results of this evaluation are shown in Figure 2.6.

TABLE II
GROUND TRUTH PERFORMANCE OF EACH INDIVIDUAL BT
ON THE 217 ANNOTATED FILES IN DATASET2. BOLD
NUMBERS INDICATE BEST PERFORMANCES.

TABLE I
GROUND TRUTH PERFORMANCE OF EACH INDIVIDUAL BT ON DATASET1.
BOLD NUMBERS INDICATE BEST PERFORMANCES.

BT	AMLI (%)	F-measure (%)	Inf. Gain (bits)
Aubio (AUB) [25]	50.6	49.4	1.58
Beatit (BIT) [26]	61.0	52.7	1.62
Beatroot (DIX) [6]	70.8	61.7	1.98
BeatUJaén (BUJ) [27]	41.6	33.9	1.18
Boeck (BOE) [8]	58.7	66.6	1.98
Davies (DAV) [5]	75.9	62.8	2.25
Degara (DEG) [9]	77.7	65.3	2.26
Ellis (ELL) [4]	60.0	55.1	1.76
Essentia (ESS) [28]	57.3	51.7	1.43
Hainsworth (HAI) [11]	59.6	51.1	1.84
IBT causal (IB1) [29]	58.0	55.2	1.67
IBT non-causal (IB2) [29]	73.8	60.5	1.92
Klapuri (KLA) [10]	77.7	65.5	2.32
Lee (LEE) [30]	26.4	48.8	1.09
Scheirer (SCH) [31]	49.0	56.2	1.69
Stark (STA) [21]	71.0	59.5	2.03
Mean	60.6	56.0	1.79

BT	AMLI (%)	F-measure (%)	Inf. Gain (bits)
Aubio (AUB)	18.5	24.7	0.68
Beatit (BIT)	20.6	28.7	0.53
Beatroot (DIX)	27.6	32.2	0.66
BeatUJaén (BUJ)	23.9	27.7	0.60
Böck (BOE)	26.1	40.1	0.91
Davies (DAV)	33.4	32.2	0.90
Degara (DEG)	33.4	34.6	0.89
Ellis (ELL)	20.8	35.2	0.62
Essentia (ESS)	23.3	26.6	0.64
Hainsworth (HAI)	26.0	24.8	0.83
IBT causal (IB1)	21.1	26.8	0.70
IBT non-causal (IB2)	28.6	31.1	0.78
Klapuri (KLA)	33.9	36.2	0.92
Lee (LEE)	12.9	34.6	0.50
Scheirer (SCH)	18.5	30.2	0.70
Stark (STA)	26.0	27.3	0.74
Mean	22.7	30.8	0.73
Deterministic	16.1	21.2	0.46

Figure 2.6: The first table (left) shows an evaluation of all state of the art beat trackers on a comprehensive beat tracking dataset. The references are labeled relative to the publication. The second table (right) is an evaluation of all state of the art beat trackers on a selected subset that has been hand chosen to be difficult to track. (Sourced from [2])

Another study that informs the field was one performed by Davies in [40]. In this work, Davies studies and evaluates the metrics that the field uses to evaluate beat tracking approaches. These evaluation approaches can be found in Table 2.1. The study links the human perception of beat

understanding to the ratings given by the various evaluation methods. The work recommends that researchers use the continuity based metrics or the information gain because they are most resilient to various evaluation conditions and maintain a high subjective (humans ratings) vs. objective (evaluation scores) correlation.

Metric	Explanation
<i>F-Measure</i>	harmonic mean of precision p and recall r . $F_1 = 2 \left(\frac{p \cdot r}{p+r} \right)$
<i>P-Score</i>	normalized sum of the cross-correlation between the estimated beat locations and the ground truth.
<i>Cemgil</i>	a Gaussian error function is placed around each ground truth annotation and accuracy is measured as the sum of the “errors” of the closest beat to each annotation
<i>Goto</i>	the annotation interval-normalized timing error is measured between annotations and beat estimates
<i>Continuity Based</i>	a given beat is considered accurate if it falls within a tolerance window placed around an annotation and that the previous beat also falls within the preceding tolerance window.
<i>Information Gain</i>	this method performs a two-way comparison of estimated beat times to annotations and vice-versa. Information Gain is calculated as the Kullback-Leibler divergence between a histogram of timing errors and a uniform histogram.

Table 2.1: Beat tracking evaluation metrics

2.3.3 Detecting Metrical and Sub-metrical Structure

Expanding upon successful work in beat tracking, meter and downbeat detection algorithms attempt to determine where musical measure boundaries exist (*downbeat*, *barline*), as well as how many beats are contained between them (*meter*). These two tasks are strongly linked and can be explained in reference to one another, so much of the canon treats these concepts interchangeably. The main difference is that for detection of downbeats, there needs to be an estimate of rhythm phase. In meter detection, only the concept of music groupings is necessary. Additionally, there is a body of work that looks at musical events at the micro-scale between beats (*ticks* or *Tatums*). Work in detecting metrical structure is an important processing step in order to define rhythmic style and expression, as well as to aid systems performing automatic accompaniment and symbolic music transcription. While some previously explained work in beat tracking (Section 2.3.2) also contained meter and downbeat detection, this section will focus on work specifically about meter and downbeats.

In work by Seppanen, the Tatum grid was employed to estimate sound onsets [41]. Histograms of IOIs are calculated, and the shortest onset intervals are found. This allows music with different time bases to be directly compared, independent of tempo, as well as provide a meaningful structural segmentation. In the area of meter recognition, work by Gouyon seeks to determine whether a piece of music has a *duple* (groupings of two), or *triple* (groupings of three) feel. Meter was estimated by the periodic recurrence of low-level acoustic features [42]. A similar approach to Seppanen was taken. The signal becomes *Tatum-aligned* meaning that features correspond to musical Tatums rather than arbitrary frame widths (such as 20ms 40ms, 1024 samples, 4096 samples, etc.). An autocorrelation of these feature signals are then taken. These autocorrelations will show spikes at multiples of 2× or 3× the lag that corresponds to a tempo estimate. With this information, the meter of the piece can be determined [41].

In work by Jehan [43] and Klapuri [44] it was shown that different time scales are important when studying musical structure. The Tatum level, beat level (*tactus*), and meter level are all important. The Jehan method uses a semi-supervised approach through which beat tracking is performed along with an estimate of meter. There is a small supervised step in which the phase of downbeats is learned through human annotations as ground truth and modeled through a support vector machine (SVM). The Klapuri system performs the analysis of meter at three different time scales simultaneously. A time-frequency representation is passed through a set of specific comb filters. These outputs are passed into a Hidden Markov Model (HMM) with the goal of estimating the pulse periods. With the pulse periods and the output of the filters, a model of phase and periods are combined in order to estimate meter. Later work by Schuller uses a stripped down version of tempo and meter detection in order to discriminate ballroom dance styles. Features are Tatum-aligned and passed through a set of comb filters in order to obtain an estimate of tempo and meter [45]. Additionally, because these styles are tempo and meter dependent, the acoustic features as well as the meter and tempo estimates are good discriminators when attempting to classify the dance styles.

Work by Papadopoulos approaches the task of finding downbeats a little differently. This integrates the knowledge of mutual dependencies of chords and metrical structure in order to gain insight

into both. HMMs are trained to model chord progressions with the knowledge of downbeats and vice versa. This shows that the information is mutually informative. Experiments were performed on a variety of Beatles' songs, without restrictions on metrical changes or tempo changes [46].

2.3.4 Rhythmic Pattern Analysis

A large body of work exists in onset detection and beat tracking. These methods are employed to model the presence of musical events, but rarely explore musical meaning and context. In this section, I outline a body of work that attempts to assign meaning and context in patterns of rhythmic events.

The work by Dixon is a good example of expanding upon beat and onset detection. It uses the beat positions and positions of onsets in order to infer note durations of each onset. This allows for later use in music transcription systems and systems that discriminate musical styles [26]. Similarly, work by Seppanen performs a rhythmic analysis by employing inter-onset-intervals in order to develop a Tatum grid, allowing for the direct comparison of rhythm [47]. In work by Gouyon, the similar process of Tatum extraction is performed. In addition to Tatum extraction, Gouyon uses the found Tatums to perform instrument and pattern recognition within polyphonic drum tracks (rough transcription) as well as beat tracking [27].

In contrast, Alghoniemy presented an early rhythm analysis study that did not rely solely on beat tracking [48]. In this work, a clip of audio is low-pass filtered in order to preserve the parts of the music that contain most of the rhythm and beats. Using binary trees created from the thresholded signal, simple periodicity patterns are learned. The authors state that these patterns can be used to discriminate musical style. Paulus in [49] had a similar idea of using a similarity matrix, however, this time it is used to measure the similarity between two different pieces of music. The meter is estimated, and downbeats, beats, and Tatums are calculated. Two audio signals can be found as rhythmically similar by a low-cost alignment of their musical features at these various metrical levels.

Similar work by Foote also investigates periodicity patterns in music. In his work the concept of the *beat spectrum* is introduced [3, 50]. The beat spectrum is a measure of self-similarity as

a function of lag l . A simple estimate (Eq. 2.7) can be found by diagonally summing along the similarity matrix S .

$$B(l) \approx \sum_{k \in R} S(k, k+l) \quad (2.7)$$

A more robust estimate (Eq. 2.8) is found using the autocorrelation of S .

$$B(k, l) = \sum_{i, j} S(i, j) S(i+k, j+l) \quad (2.8)$$

Similar to an audio spectrogram (STFT), the beat spectrum can be shown in sliding windows over time. This is known as the *beat spectrogram*. Some examples of the beat spectrum and a beat spectrogram are shown in Figure 2.7.

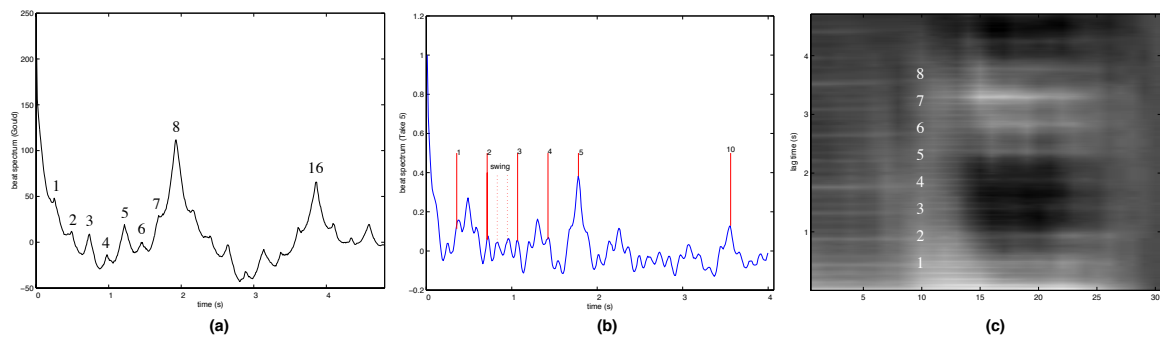


Figure 2.7: Example beat spectra of *Bach, Prelude No. 1* (a) generated from Equation 2.7 and *Take Five* generated from Equation 2.8 (b). A windowed form of the beat spectrum over time can generate the beat spectrogram. A beat spectrogram of Pink Floyd’s *Money* is shown in (c). (Sourced from [3])

Much of the work on rhythm centers around the tasks of classifying defined rhythmic styles. Work by Dixon uses some previous methods of IOI cluster histograms and autocorrelation as a means to find periodicities in audio. Using these periodicities, the system is able to estimate the meter, the tempo, and the beats. These attributes are then used as features in order to discriminate different styles of Ballroom dance music [51]. Work by Gouyon presented a set of rhythmic descriptors that are useful when describing audio. Using these features, a classification of ballroom dance styles is performed using the *Ballroom Dataset* [52]. These features are:

- tempo estimate
- PH percussiveness
- IOIH flatness
- periodicity histogram (PH)
- IOI Histogram mean (IOIH)
- IOIH distribution kurtosis
- distinctiveness of PH
- IOIH geometric mean
- IOIH high frequency
- PH power
- IOIH total energy
- DCT of IOIH (MFCC-like)
- PH centroid
- IOIH centroid

In work by Peeters, another more compact feature is presented that achieves similar results to state of the art on the *Ballroom Dataset* [53]. Similar to previous methods, autocorrelation, Fourier analysis and IOI histograms are computed for an ODF. Through interpolation and element-wise multiplication, various periodicity forms can be fused. The final form can be made compact by only keeping the values at musically relevant ratios of the tempo estimate.

Another rhythmic descriptor known as *Fluctuation Patterns (FP)* was introduced by Pampalk [54]. The objective of this descriptor is to measure the periodicity of the loudness in various frequency bands. This feature has high dimensionality, so others such as Pohle attempt to clean it up [55]. Pohle introduces a few changes that include reducing the signal to only parts of increasing amplitude (onsets), using semitone bands to detect onsets, and using improved windowing techniques when detecting periodicities.

Much of the work explained thus far has relied on a passive analysis of audio signals in order to classify rhythm. In order to better inform rhythmic representations, it may be important to seed the system with examples as a model. One such representation was introduced in a body of work by Tsunoo in [56, 57, 58]. This work starts with a set of basic symbolic rhythmic patterns that best represent a given style. These patterns are then synthesized. Each pattern is aligned to a given section of audio using dynamic time warping. The pattern that aligns best is chosen for that section. This is done over the entire signal, creating a list of best aligned clips over time. Within a specific style, the synthesized patterns are combined with the original patterns by taking the mean. This is performed iteratively until convergence. A compact rhythmic feature can be found by how well sections of a song align with learned patterns of a known style.

Work by Volkel performs a similar process of seeding a system with symbolic data [59]. A set of reference synthesized audio examples are created with symbolic patterns and percussion audio

samples. A set of autocorrelation based features are calculated both on the unknown examples and the reference pattern audio. These representations are then made tempo invariant with a log scaling of the lag axis. Tempo becomes a linear scaling factor of the resulting representation [60]. This invariance allows for direct distance calculation of two examples. Classification was done by a simple Nearest Neighbor classification, based on a chosen distance metric.

Similar to work in onset detection and beat tracking, later work has focused on learning rhythmic patterns through neural networks. Battenberg uses conditional deep belief networks to learn different styles of rhythmic patterns [61]. This differs from other general neural net approaches because time evolution is taken into account. Each node in the current time step can be conditioned on another previous node. Each network is trained using snare drum, bass drum, and hi-hat activations. The network is used to learn the likely Tatum position of each activation given previous activations. This can be altered to form a generative model of rhythmic style with the motivation of creating an automatic drum machine pattern generator.

Work by Krebs models the basics of rhythmic patterns in order to estimate the positions of beats and downbeats [4, 39]. The work presents an HMM approach, however it is formulated as a Dynamic Bayesian Network. It models Tatum, beats, downbeats, and rhythmic patterns simultaneously throughout its multiple layers. This allows the modeling and prediction of rhythm at all metrical levels. An example of this network is shown in Figure 2.8.

2.3.5 Rhythm and Drum Transcription

Other work in rhythmic analysis focuses solely on *drum transcription*, where the goal is to create an exact symbolic representation of a performed sequence. While understanding rhythmic patterns is important, it is only an intermediate step in this task. In my work, I am less interested in creating a transcription system. My goal is to capture the function of rhythm in music from a grounded and more general perspective. For further information on drum transcription, the reader is referred to the following texts shown in Table 2.2.

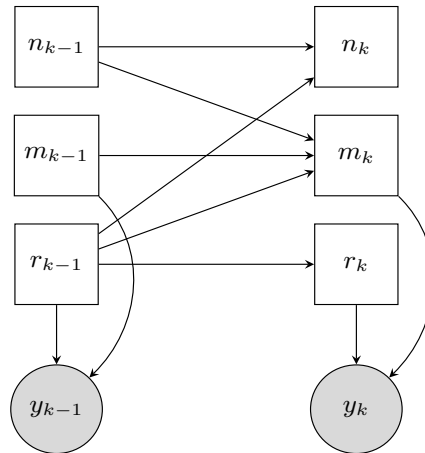


Figure 2.8: Dynamic Bayesian network; circles denote continuous variables and rectangles discrete variables. The gray nodes are observed, and the white nodes represent the hidden variables. This network models the tempo $n = \frac{\text{bars}}{\text{audio frame}}$, position inside bar m , and rhythmic patterns r . (Sourced from [4])

Author	Title	Source
FitzGerald	Sub-band independent subspace analysis for drum transcription	[62]
FitzGerald	Drum transcription in the presence of pitched instruments using prior subspace analysis	[63]
FitzGerald	Drum transcription using automatic grouping of events and prior subspace analysis	[64]
FitzGerald	Prior subspace analysis for drum transcription	[65]
FitzGerald	Automatic drum transcription and source separation	[66]
FitzGerald	Unpitched percussion transcription	[67]
Gillet	Automatic transcription of drum loops	[68]
Gillet	Drum Track Transcription of Polyphonic Music Using Noise Subspace Projection.	[69]
Gillet	ENST-Drums: an extensive audio-visual database for drum signals processing.	[70]
Gillet	Supervised and Unsupervised Sequence Modelling for Drum Transcription.	[71]
Gillet	Transcription and separation of drum signals from polyphonic music	[72]
Tzanetakis	Subband-based drum transcription for audio signals	[73]
Yoshii	Automatic Drum Sound Description for Real-World Music Using Template Adaptation and Matching Methods.	[74]
Yoshii	Adamast: A drum sound recognizer based on adaptation and matching of spectrogram templates	[75]
Yoshii	An error correction framework based on drum pattern periodicity for improving drum sound detection	[76]
Paulus	Drum transcription with non-negative spectrogram factorization	[77]
Paulus	Combining Temporal and Spectral Features in HMM-Based Drum Transcription.	[78]
Thompson	Drum transcription via classification of bar level rhythmic patterns	[79]

Table 2.2: List of drum transcription literature.

2.4 Predicting Human-Labeled Attributes

2.4.1 Musical Style and Genre

Musical *genre* is a high-level label given to a piece of music (e.g., Rock, Jazz) to both associate it with similar music pieces and distinguish it from others. Genre is a very popular way to organize music as it is being used by virtually all actors in the music industry, from record labels and music retailers, to music consumers and musicians via radio and music streaming services on the Internet. Genre labels are largely debated, however it is important to both sort and classify music in some differentiable way, and genre labels are how we do that.

With the explosion of music available to a listener at any given time, it is important to be able to automate this process of genre labeling. One such study to support these varied opinions and compare some machine based methods was explored by Sordo in [80]. In this work, expert genre taxonomies and general public *folksonomies* are compared. Expert taxonomies are important in order to characterize and sort data. A folksonomy is a direct human sourced version of organization, such as tags in a radio service. Results from this study show that experts, the crowd wisdom, and various computational methods agree on some genres (hip hop, blues), but not on others (rock). It is important to leverage experts, the general public, and machines in order to characterize genre. The rest of this section will give an overview of methods for the automatic detection of musical genre. Some methods use tag-based approaches while others use multi-class discriminative methods. Approaches for tagging will be explained later in Section 2.4.2.

Some of the earliest work in recognizing musical genre was presented by Soltau in [81]. In this work a system was presented that automatically identified music as one of four categories: rock, pop, techno, classical. The presented method employed a neural network to learn the dynamics of acoustic cepstral features over time. The hidden states of that network were then used in a more traditional network approach to discriminate music types. This method was similar to the *feature-learning* approaches of today. It was shown that this approach was superior to HMMs, which were popularly used for speech recognition at the time.

A few years later, Tzanetakis more formally presented the problem of genre recognition and introduced a few methods by which to approach it. The first contribution was a larger dataset of musical genre. It contained 10 genres with a few (classical and jazz), containing sub-genres. Each representative class contained 100 audio examples [5]. This genre hierarchy is shown in Figure 2.9. This dataset became the basis of genre classification research (and criticism) for many years.

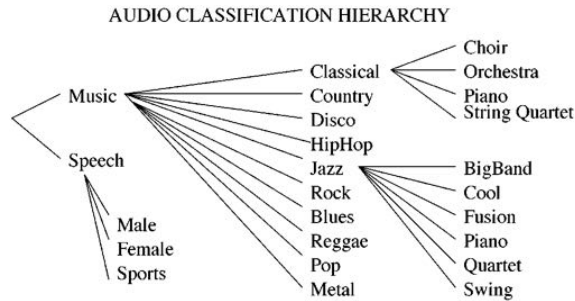


Figure 2.9: A hierarchical layout of musical genre classification as presented by Tzanetakis. (Sourced from [5])

In order to automatically classify musical genre, three types of features were proposed to represent the audio. Timbre was represented with simple characteristics of spectral shape and energy as well as MFCCs. Rhythm features were represented with characteristics of estimated beats found with a beat tracking algorithm such as relative beat strength over time and beat stability. The third type of features were pitch content features. These features include the prominent pitch class of the song (key information), the most common intervals (mode and harmony information), and the general octave range of the most prominent pitches. Using features from these three areas, Gaussian Mixture Model (GMM) and Nearest Neighbors (NN) classifiers were trained. The results were very promising, a three Gaussian GMM showed 61% classification accuracy for the general 10 class problem and 88% and 68% for the classical (4 class) and jazz (6 class) subgenres respectively.

Motivated by Tzanetakis, later work in genre recognition starts to improve the selection of features used and the methods of classification. In work by West [82, 83], the *spectral contrast* feature is used in conjunction with MFCCs in order to describe the audio signal. Spectral contrast, looks at the shape of spectral peaks and valleys in octave based frequency bands. A tree-based classification method is also introduced. The method by which features are represented and songs

are classified is also altered. Instead of a song containing mean and variance based features based on many windows calculated over time, each of the windows is treated independently when classifying. The full songs genre is based on majority vote of each of it's pieces. Work was also done to examine the size and number of bocks to be used per song. Subsequent work by [84] further improves feature representations with the introduction of Fluctuation Patterns and various features derived from them.

A body of work evaluated by Gouyon in [85] revolving around tempo detection provided something quite important for genre detection as well with the introduction of the *Ballroom Dataset*. This dataset provided 698 audio examples split across 8 ballroom dance styles. This was important for both the genre and rhythm subfields because these musical style labels were more grounded in musicology and more straight-forward. Labels had specific compositional attributes related to the associated dance style rather than vague opinions of cultural popularity contained in the discrimination of previous musical genre datasets. Schuller in [45] used proposed meter and tempo recognition algorithms in order to discriminate ballroom dance styles. It was one of the first genre related tasks that attempted to link concrete musicological attributes to musical genre. This study also uncovered some flaws in the Ballroom dataset. It showed tempo alone was a good discriminator of genres. Because of this, it is important to take tempo into account, and be sure that a proposed system is learning components of genre rather than being a decent tempo detector and performing well on this dataset (but not on others) because of that correlation.

More advanced work in feature design was presented by Tsunoo in [57, 58]. This work uses the alignment and agreement of genre representative rhythmic pattern and bass-line templates in order to describe and discriminate audio signals. This study tries to provide a direct link of theoretical constructs and genre. It was shown to greatly improve classification methods that use timbre alone.

Much work in genre focuses on smaller datasets. It is important that proper training instances are chosen in order to accurately describe genre but not overfit the small datasets. Work by Lopes in [86] introduces some methods for selective sampling of training sets. The work tries to maximize separability in a small sample that generalizes well to a larger sample. Work by Marques in [87] also

tries to better formulate the genre problem by attempting to better understand the feature space used for classification. It was found that quantizations and clusterings of the larger feature space showed little degradation in classification results, suggesting that the features we are using, while effective could be simpler.

There are also studies that suggest, similar to notions alluded to before, that we might not actually be recognizing genre with this work, but something that correlates well to small datasets but does not scale. Work by Sturm brings up some of these issues. He analyzes and criticizes the current work in the field through finding or sometimes introducing statistical anomalies in data [88, 89].

However, there are a few take-aways in the criticism of the work. It is important to examine what information features are capturing and if their representations are confounded with or causal to the labels. Additionally, if “everyday people” are the sole beneficiaries of these systems, and they use general folksonomies rather than academic taxonomies to describe music, the systems serve them directly, even if their definitions are not as concrete. If a system for describing a culture alienates those who it is supposed to benefit, it is of little use [80].

2.4.2 Human-Tagged Attributes

As the internet became a main stream medium for music listening, so did the ability to store and search vast amounts of music and meta-data. A large subfield in MIR studies semantic descriptors of audio. These descriptors can be used alone or in conjunction with audio features for song similarity, music exploration, recommendation, and musicology. These descriptors, known as *tags*, can be applied by musical experts (expert tags) as well as the general public (social tags). A couple of music industry examples of this in practice are *Pandora’s Music Genome Project* (expert tags) and *Last.fm* (social tags). This section will provide an overview of systems that recommend music based on tags, learn the tags based on the music, and collect tags. Many tag-based approaches are rooted in Natural Language Processing (NLP), Term-Document Indexing, and Multimedia Image Retrieval.

One of the cornerstone works in semantic-audio retrieval (SAR) presented by Slaney describes a system for connecting sounds and words in a linked multidimensional vector space [6]. The semantic

space uses multinomial models to represent and cluster semantic documents. The acoustic space is formed by performing linear discriminant analysis and learning anchor Gaussian Mixture Models (GMM) on acoustic features grouped into a set of classes that clearly represent them. Linkage is formed by agreement of terms in the semantic space and the acoustic class hierarchy. This allows semantic terms to retrieve sounds (text search query), as well gives sounds the ability to retrieve related terms (*auto-tagging*). Examples of this are shown in Figure 2.10.

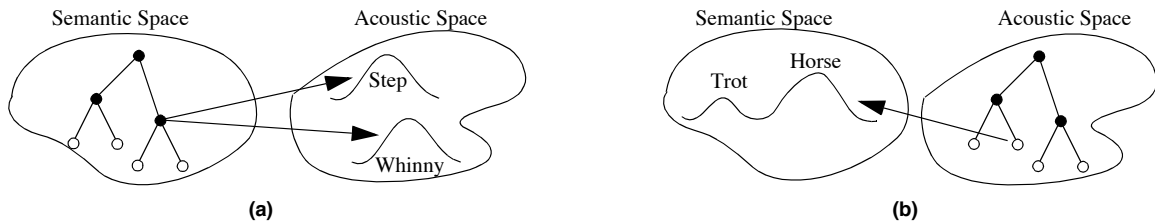


Figure 2.10: Semantic-Audio Retrieval of audio from a text query (a) and text descriptors from an audio query (b). (Sourced from [6])

Because musical meaning is not contained solely within the audio signal, but also in cultural and semantic descriptions of a song, a study by Whitman used web-crawled data in order to classify different artists [90]. Using the web-crawled data, a set of basis vectors was created that best represents the audio. The power spectral density (PSD) is mapped to the web mined descriptors of the audio. The PSD of unknown audio can then be reduced to a semantic, and hopefully more meaningful space.

In work by Knees, a method is proposed that uses word occurrences on artist websites in order to classify them [91]. Later work by Knees proposes the use of both content based methods and artist similarity by websites for playlist generation [92]. Artist similarity via the web is similar to the previous approach. Audio content similarity is based on MFCC's and GMM sampling. Both are combined and playlists are generated with self organizing maps (SOM). This work is novel in its ability to combine both semantic data and audio data to solve a related, but separate, task. Celma presents similar work in music recommendation. In this work however, much more data is scraped from the web than just artist profiles. Users themselves have generated profiles interests and listening habits. These are used in conjunction with artist profiles, new music releases, upcoming concerts (by location), podcast sessions, music blogs, and album reviews. The paper outlines that two important

types of recommendations can be considered. The first type, *static recommendations*, can be made by considering a users favorite artists and their similarity to other artists. The second type, *dynamic recommendations*, can be made by analyzing currently trending information, such as upcoming concerts and new song and album releases, in conjunction with the evolving listening habits of a user [93]. Another expansion of the tag space was presented by Bischoff [94]. This work uses tag-based information to infer *theme recommendations*. In this system, users can search based on the music’s context of use. In this case, “Theme” music refers to queries such as “Halloween”, “Christmas”, “workout”, “beach”, “dinner”, “driving”, etc.

In work by Levy, it is shown that while social tags are extremely varied and informal in their representation, a low-dimensional semantic space can still be derived that works well on the track level [95]. These social tags were obtained from *Last.fm*, an internet radio site where users can tag the music they are listening to with any term they want. By using Correspondence Analysis to put both tags and songs in the same space, other more specific queries, such as by mood, can be performed and songs can accurately be retrieved. Keeping with the theme of social tags, others have tried to create additional tools for tag collection. One such example is *TagATune* [96]. Through this online game, players tag songs and must decide if they are listening to the same song, or a different one. This adds some additional motivation for better and more descriptive tags. The game also has an auto-player bot for when a user is not paired up with a live partner. This can be automated with previous annotation tags. It can also be driven by another auto-tagging algorithm, and a live players agreement can be used to evaluate it [97]. A similar game named *MajorMiner* was presented by Mandel in [98]. Rather than the objective of deciding whether or not you are listening to the same clip (*TagATune*), *MajorMiner* scores points simply on tag agreement with other participants.

The previously discussed work up until this point has been purely relational in trying to directly map human responses to audio content. In some of the following methods, focus is placed on the sole goal of attempting to automatically tag music that is poorly curated or when tags are non-existent. In work by Eck, a supervised machine learning approach is proposed to automatically generate a tags from audio content. The set of predicted tags is also restricted to a standard set of common

tags. Tags are also treated as semi-continuous, allowing for the model to represent how much of a certain attribute there is. Finally, any song can contain any number of tags simultaneously, each with varying degrees of expression. For each tag, a separate AdaBoost based classifier is trained [99, 100]. Similar work was presented by Tingle. A data set was collected called *Swat10k* (also *Cal10k*), which consists of 10,870 songs annotated using a vocabulary of 475 acoustic tags and 153 genre tags from Pandora's *Music Genome Project (MGP)*. The Music Genome Project is an initiative by Pandora for the expert labeling of songs on a variety of musical attributes in order to recommend the music based on song similarity and listener preference. The acoustic tags were consistently applied to songs by experts, making the data much more reliable than social tags. This data is not publicly available, however each recommended song is presented with a few attributes that motivate Pandora's recommendation choice. Tingle used these public attributes as the tags. Using this data, an auto tagging system was created in conjunction with timbre and song features from the popular Echonest API (ENT and ENS) [101].

In work by Turnbull, it was explained that tagging must be more comprehensive in the methods used [1]. Not one is best for all situations. It is important to weight the strengths and weaknesses of each type of system. A comparison of tagging methods is shown in Table 2.3. It was shown that in combining approaches, results were better than the best performing single method. Subsequent methods by Turnbull take this into account. Work presented in [102] uses social tags, web documents and acoustic timbre features in order to create a query by text music retrieval system. The work further proves that it is important to use multiple information sources.

Later work by McFee also attempts to improve an area that tagging systems usually fail, *music's long tail*. This refers to a great deal of available music listened to by only a few people. Collaborative filtering methods work well at recommending popular music, but not as well at users more sparse preferences. This work introduces a method to learn an optimal similarity function that allows music in the long tail to still benefit from collaborative filtering based methods [103].

Tags generally refer to the song in its entirety. There have been a few methods however that take temporal context into account. One such method is presented by Coviello [104]. This work models

Approach	Strengths	Weaknesses
Survey	custom-tailored vocabulary high-quality annotations strong labeling	small, predetermined vocabulary human-labor intensive time consuming approach lacks scalability
Social Tags	collective wisdom of crowds unlimited vocabulary provides social context	create and maintain popular social website ad-hoc annotation behavior, weak labeling sparse/missing in long-tail
Game	collective wisdom of crowds incentives produce high-quality annotations fast paced for rapid data collection	“gaming” the system difficult to create viral gaming experience listening to short-clips
Web Documents	large public corpus of relevant documents no direct human involvement provides social context	noisy annotations due to text-mining sparse/missing in long-tail weak labeling
Autotags	not affected by cold-start problem no direct human involvement strong labeling	computationally intensive limited by training data based solely on audio content

Table 2.3: Strengths and weaknesses of tag-based music annotation approaches. (sourced from [1])

time-series acoustic features with dynamic texture mixtures for each tag. Other work by Mandel models the contextual relationships between tags and between tagged clips. Users agree more on tags applied to clips temporally near one another, so using a Conditional Restricted Boltzmann Machine (CRBM) can more accurately predict tags by taking context into account [105]. A CRBM is similar to a regular RBM, but nodes in the current time step can be conditioned on another previous set of nodes. This context smooths training data and allows an SVM to better rank clips on the smoothed data than the original tags.

2.5 Designing Visually intuitive Feature Spaces

Creating visually intuitive feature spaces is a very active subfield within MIR. Using these representations, researchers employ both semantic content and audio content based methods in conjunction with basic machine learning techniques to combine human annotations and audio feature representations of music in a shared visual analysis space. Spaces derived from human-tagged attributes have the potential to follow a uniquely human organization, which may be helpful when designing a human-interpretable space. However, they can only capture information humans have already deemed important. Conversely, in designing a space from audio features, we may be able to capture nebulous interactions that humans cannot easily deconstruct (i.e. rhythmic syncopation). An-

other thing to consider is parametric vs. non-parametric reduction methods. In this section I will overview a few methods used to create low-dimensional visual representations of high-dimensional music-related data. More information about the parametric and non-parametric methods employed in this thesis can be found in Section 4.4 and Chapter 9.

2.5.1 Spaces: Emotion

Some of the first work in space reduction for music-IR surrounded the representation of emotion in music. It was represented by a set of 66 adjectives encompassed in 8 subgroups [106]. Later work has added many more terms while more recent studies have been able to reduce these large dictionaries. These reductions are optimized for the maximal description and discrimination of emotion [107]. The Music Information Retrieval Exchange (MIREX) contest, a large-scale competitive research comparison and analysis contest, uses the widely accepted set of terms in Table 2.4 [108].

Clusters	Mood Adjectives
Cluster 1	passionate, rousing, confident, boisterous, rowdy
Cluster 2	rollicking, cheerful, fun, sweet, amiable / good natured
Cluster 3	literate, poignant, wistful, bittersweet, autumnal, brooding
Cluster 4	humorous, silly, campy, quirky, whimsical, witty, wry
Cluster 5	aggressive, fiery, tense / anxious, intense, volatile, visceral

Table 2.4: List of MIREX Music Emotion Terms

However, a low-dimensional continuous space was also developed to describe emotion through multidimensional scaling (MDS) of sets of emotion semantic terms. This idea of a continuous emotion space was first introduced by Russel and Thayer. It is known as the *Valence-Arousal (V-A) Space* and is shown in Figure 2.11. When discussing emotion, happy versus sad temperament is referred to as *valence* and higher versus lower intensity of that temperament is referred to as *arousal* [109]. There is sometimes a third dimension that relates to tension or kinetics as outlined in [110] and [111] respectively.

2.5.2 Spaces: Performance Expression

Musicians creatively vary timing, dynamics, and timbre of the musical performance, independent from the score, in order to communicate something of deeper meaning to the listener. For example,

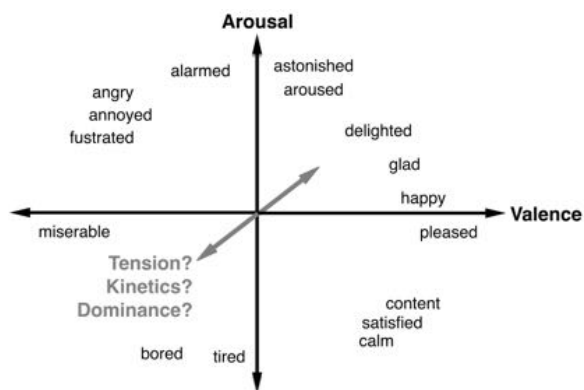


Figure 2.11: A representation of the V-A space. (Sourced from [7]).

a musician can alter tempo or change dynamics slightly to impart tension or comfort. Similarly, they can alter the timbre of their instrument to create different tonal colors. All of these parameters add an additional level of intrigue to the written pitches, rhythms, and dynamics being performed. With a musician’s mastery of these various nuances in technique, they can communicate more abstract concepts such as emotion and mood [111, 112], There is a large body of work that looks to quantify these parameters, similar to emotion, in both semantic and continuous spaces.

Some of the most influential work in this area is the work by Mion, De Poli, and Canazza. They performed a set of studies with the overall goal of quantifying expressive parameters in performance and capturing them with low-level audio features. In [8], Canazza defines a space through which expressive intent can be projected into two dimensions. This space is known as the *Kinematics-Energy (K-E) Space*, and is similar to the Valence-Arousal space in design. The *kinematics* refer to heavy and dark versus light and bright expressive timbre. The *energy* refers to soft versus hard intensity. This space is shown in Figure 2.12. Using the extremes of this K-E space and the A-V space, they prompt musicians to play representative examples of each. Then through *sequential feature selection (SFS)*, they find an optimal set of low-level expression-motivated audio features that best classify the intended emotion [111]. They then further explore expressive communication through a perceptual study and find that people are able to classify the different expressive intentions of the musicians. They then try to link the affective (A-V space) and sensorial (K-E space) domains

in order to try and derive which sensorial attributes result in specific induced emotions [113]. This links the expressive techniques used by musicians to the emotions their listeners perceive.

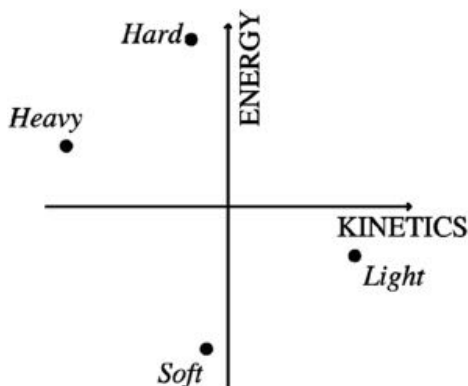


Figure 2.12: A representation of the Kinematics-Energy space. (Sourced from [8])

In work by both Repp and Windsor, it was stated that the combination of both timing and dynamics play a large role in the aesthetic impression of performance [114, 115]. One way to quantify this is the *Tempo-Loudness (T-L) Space* [9]. Its dimensions are relatively simple, tempo in *beats per minute (bpm)* and perceptual loudness (relating to dynamics) in *loudness sensation (sone)*. This work by Langner presents multiple performances of the same piano piece by different musicians and shows that this T-L space can capture expressive differences as well as create a simple compact snapshot of the pieces' expressive evolution over time.

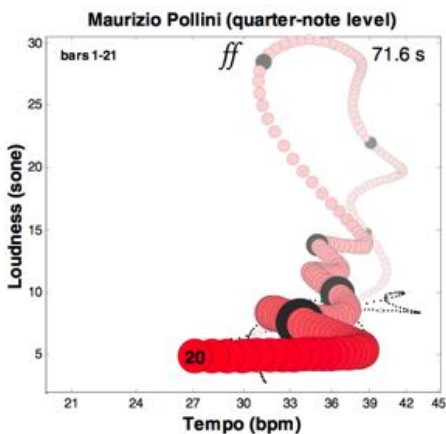


Figure 2.13: An example of the Tempo-Loudness Space. (Sourced from [9])

2.5.3 Spaces: Non-Linear

Many classic techniques of dimensionality reduction are linear and parametric. Linear methods are used to learn a set of components and corresponding activations with the objective of reconstructing the original feature space through linear combinations. Non-linear, and sometimes non-parametric, reductions such as *Self-Organizing Maps* (SOM) or *t-Distributed Stochastic Neighbor Embedding* (*t-SNE*) do not have this constraint and have been gaining traction in recent years for organizing and visualizing high-dimensional data [116, 117].

Early work employing SOMs has shown functionality in creating music similarity spaces [54, 118]. More recently, t-SNE has been employed to learn feature space reductions in a stochastic, and non-parametric manner [119, 120]. Because these spaces are non-parametric, it makes it difficult to define the meaning of each dimension. They are designed strictly to be similarity spaces, meaning that similarity in the high-dimensional space is maintained in the low-dimensional space. They have proven to be powerful for data organization and context informed retrieval due to their ability to capture non-linear organizations (manifolds) of the data. A few examples shown in Figure 2.14 show an organization of Ballroom dance styles and musical meter using a set of rhythm features and the t-SNE reduction method.

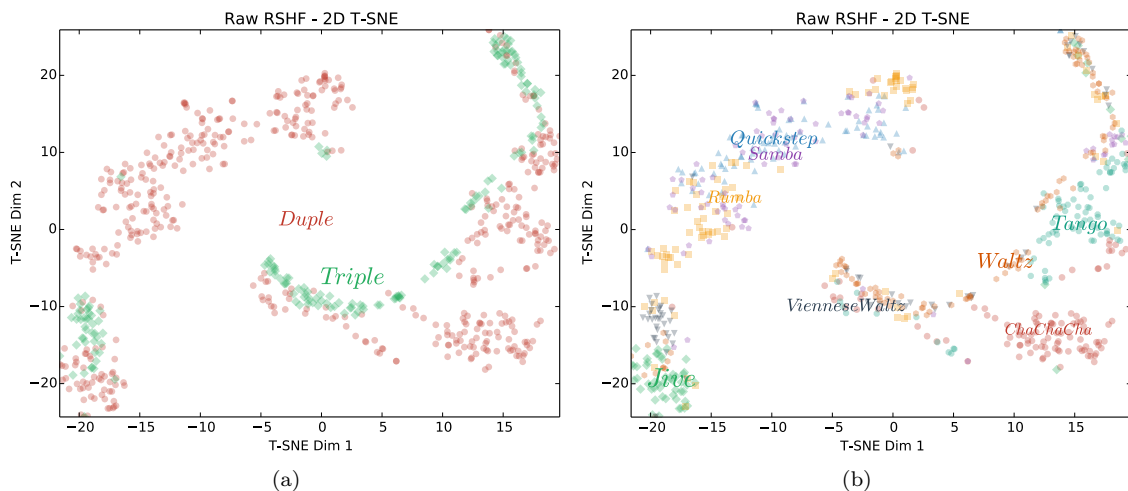


Figure 2.14: t-SNE projections of acoustic features in 2 dimensions for (a) duple/triple designation and (b) individual style classifications.

Chapter 3: Data: Labels of Rhythmic Components and Style

For the majority of this work, the rhythmic component and style labels are defined and collected by musical experts on a corpus of over one million audio examples from the *Pandora[®] Music Genome Project[®](MGP)*¹. In addition to the *MGP*, some evaluation is performed on more commonly used datasets, such as the *GTZAN Rhythm Dataset* and the *Ballroom Dataset*.

3.1 The GTZAN Rhythm and Genre Dataset

The *GTZAN Genre Dataset* was originally introduced by Tzanetakis [5] for the classification of music genre. The dataset includes 1000 songs and 10 genre classes each with 100 songs. The genres included are:

- Blues
- Classical
- Country
- Disco
- Hip hop
- Jazz
- Metal
- Pop
- Reggae
- Rock

More recently an updated version of this dataset was curated that includes rhythm annotations as well [121]. This new dataset is known as the *GTZAN Rhythm Dataset*. This new set includes the following additional labels:

- Tempo
- Meter
- Beat
- Downbeat
- Swing
- 2 vs. 3 Tatum-feel

In order to create a dataset that more closely approximates the *Music Genome Project* for comparison, each of the meter attributes can be transformed into their logical meter classes (Duple,

¹“Pandora” and “Music Genome Project” are registered trademarks of Pandora Media, Inc. <http://www.pandora.com/about/mgp>

Compound-Duple, Triple, Mixed) from the annotated time-signature fractions. These can then be formulated as a multi-class discrimination problems or binary labels of the expression of each meter type. The later more closely approximates the *MGP* labels outlined in Section 3.3.

3.2 The Ballroom Dataset

A set of classification tasks using the popular *Ballroom Dataset* [85] is performed in Chapters 5 and 6. The dataset contains audio examples that are each 30 seconds in length and labeled with a specific ballroom dance style. This dataset was chosen because its labels apply directly to terms that reference quantifiable attributes of the music rather than more nebulous attributes that relate to cultural popularity (e.g., the pop genre). The styles included in the dataset are shown in Table 3.1.

Dance Style	Count	Tempo	Meter	Origin	Characteristics
ChaChaCha	111	Moderate	4/4	Cuban	syncopated
Jive	60	Fast	4/4	USA	swing, rock & roll
Quick Step	82	Fast	2/4	USA	syncopated
Rumba	98	Slow	4/4	Cuban	ballad, syncopated
Samba	86	Moderate	2/4	Brazilian	dense, syncopated
Tango	86	Moderate	4/4	Argentine	march-like.
Waltz	110	Slow	3/4	Austrian	groupings of 3
Viennese Waltz	65	Fast	3/4, 6/8	Austrian	groupings of 3

Table 3.1: Classes of the The Ballroom Dataset.

3.3 The Music Genome Project

In order to obtain the best representations of music attributes available, I am working in conjunction with *Pandora Media Inc.* The labels were defined and collected by musical experts on a corpus of over one million audio examples from the *Music Genome Project*[®] (*MGP*) as part of the *Pandora*[®] Internet radio recommendation service. The labels were collected over a period of nearly 15 years and great care was placed in defining them and analyzing each song with a consistent set of criteria. Each track is labeled by musical experts on more than 500 compositional and cultural attributes. These labels are held as part of the streaming media service’s trade secrets, and unfortunately can not be made public. However, each analyst is heavily vetted and constantly monitored for quality control of the labels. I will therefore treat each label as accurate ground truth.

In this thesis, three types of *MGP* labels are explored: rhythm attributes, orchestration attributes, and genre. In this chapter, a brief description of each attribute label is given for context, but is by no means exhaustive. All labels are rated initially on a continuous scale. Due to the nature of some ratings denoting only absence or presence of an attribute, a simpler binary version is sometimes used for evaluation. The *continuous labels* rate attributes on a sliding scale according to their relative dominance in the music. The *binary labels* (binarized versions of continuous labels) discretely state an attribute absence or presence.

It is important to note that all labels represent only a single attribute. An absence or presence of one attribute does not necessarily imply that another attribute is absent or present. Any number of attributes can be present or absent simultaneously. Finally, there are some labels that do not apply to certain pieces of music. When evaluating this type of label, examples that are not relevant to the label context are ignored. However, when employing stacked approaches, such as the creation of a shared music attributes layer in Chapter 8), their estimates will be used for all examples regardless of context.

3.3.1 Rhythm Attribute Labels

The first type of labels describe compositional attributes of rhythm in music. These will be referred to as the *rhythm attribute labels*. The targeted attributes are compositional constructs, such as the meter, or well-defined components of the rhythmic feel, such as the presence of swing. Focus is placed on the 10 rhythmic attributes in Table 3.2. Each attribute is initially rated on a continuous scale. For model evaluation purposes the meter as well as the presence of swing, shuffle and high syncopation are binarized. Backbeat, danceability and relative tempo remain continuous ratings. Relative tempo is not a strict BPM, but a general relative rating of the tempo from slow (largo, adagio) to fast (allegro, presto).

3.3.2 Orchestration Attribute Labels

In addition to the rhythm attributes, another set of compositionally motivated attributes from the *MGP* is explored in order to produce more robust models in certain contexts. These *orchestration*

Cut-Time Meter contains 4 quarter notes per measure with emphasis on the 1st and 3rd note creating 2 felt beats. The tempo feels half as fast.

Triple Meter contains groupings of 3 with consistent emphasis on the first note of each grouping. ($\frac{3}{4}$, $\frac{3}{2}$, $\frac{3}{8}$, $\frac{9}{8}$)

Compound-Duple Meter contains 2 or 4 sub-groupings of 3 with emphasis on the 2nd and 4th grouping. ($\frac{6}{8}$, $\frac{12}{8}$)

Odd Meter identifies songs which contain odd groupings or non-constant sub-groupings. ($\frac{5}{8}$, $\frac{7}{8}$, $\frac{5}{4}$, $\frac{7}{4}$, $\frac{6}{4}$, $\frac{9}{4}$)

Swing denotes a longer-than-written duration on the beat followed by a shorter duration. The effect is usually perceived on the 2nd and 4th beats of a measure. (1 . . 2 . a 3 . . 4 . a)

Shuffle is similar to swing, but the warping is felt on all beats equally. (1 . a 2 . a 3 . a 4 . a)

Syncopation is confusion created by early anticipation of the beat or obscuring meter with emphasis against strong beats.

Back-Beat Strength is the amount of emphasis placed on the 2nd and 4th beat or grouping in a measure or set of measures.

Danceability is the utility of a song for dancing. This relates to consistent rhythmic groupings with emphasis on the beats.

Tempo is speed of the music pulse. In this work, it is scored on a relative scale similar to the other attributes rather than representing a direct beats per minute (bpm) rating.

Table 3.2: Definitions of the rhythmic attributes explored.

attributes are comprised of elements of the vocals, instrumentation, and sonority. This chosen subset of 38 additional *MGP* attributes is designed to have a generalized meaning across all genres (in western music) and map to specific and deterministic musical qualities. An overview of the attributes is shown in Table 3.3. Once again, some are binarized (instrumentation-related) while others remain continuous (timbre-related). Due to the proprietary nature of some of these labels, a more in-depth discussion about each can be made upon request.

Vocal attributes denote the presence of vocals and timbral characteristics of voice (e.g., male, female, vocal grittiness).

Instrumentation attributes denote the presence of instruments (e.g., piano) and their timbre (e.g., guitar distortion)

Sonority attributes describe production techniques (e.g., studio, live) and the overall sound (e.g., acoustic, synthesized)

Table 3.3: Abridged outline of the orchestration attributes explored.

3.3.3 Genre and Culture Labels

The last set of labels are the *genre labels*. Each of the labels represents a distinct musical style. While many are culturally motivated, they are not defined strictly by cultural popularity, such as *pop music* in the traditional genre labeling task. In this work I explore a selected subset of 12 ‘basic’ genres and 47 additional sub-genres. ‘Basic’ genre is assembled as a mix of very expansive genres (e.g., Rock, Jazz) as well as some more focused ones (e.g., Disco and Bluegrass), serving as an analog to many previous genre experiments in MIR. A selection of genre labels and a simplistic high-level organization for discussion purposes is shown in Table 3.4. ‘Basic’ genre and Jazz sub-genre lists are outlined completely, while Rock, Rap, Dance, and World genres as well as Geo-cultural attribute lists are abridged for proprietary sensitivity. Each of these genres is labeled on a continuous scale. For evaluation of prediction models, the labels are converted to a set of binary attributes.

Basic Genre:	Rock, Soul, Funk, Folk, Rap, Latin, Reggae, Country, Blues, Disco, Jazz, Bluegrass
Jazz Subgenre:	Free, Cool, Fusion, Bebop, HardBop, Boogie, Swing, Afro-Cuban, New Orleans, Acid, Brazilian, Smooth
Rock Subgenre:	Light, Hard, Punk, etc.
Rap Subgenre:	Party, OldSchool, Hardcore, etc.
Dance Subgenre:	Trance, House, etc.
World Subgenre:	Cajun, North African, Indian, Celtic, etc.
Geo-cultural (language, location):	Spanish, Eastern European, Central Asian, etc.

Table 3.4: Some of the musical genres and subgenres used.

3.3.4 Testing/Training Sets and Evaluation

For most tasks in Chapters 7, 8, and 9, as well as Appendix B and C, model training and evaluation was performed on a 70%:30% (train:test) split. The full *MGP* dataset used included more than 1.2 million examples. This yields a training set of approximately 840,000 examples and a testing set of approximately 360,000 examples. The same training and testing set was used across all experiments. The training/testing split was also chosen such that there were no artists represented simultaneously in both sets. Many audio-driven tasks in Music-IR can be susceptible to Album or Artist effects due

to shared mastering techniques. This is mitigated by not sharing any artists between training and testing sets.

In addition to the full dataset, a smaller subset of 50k examples was used for certain tasks involving space visualization in Chapter 9. While the dataset was smaller than the full dataset, it was sampled such that it contained similar label representation and distributions when compared to the full set. Once again both training and testing folds were selected such that there were no shared artists between the two.

Chapter 4: Machine Learning

In this chapter I outline methods that can be used to predict the expert-labeled data from the Pandora[®] *Music Genome Project*[®]. When choosing models, it is important to weigh both the cost of computation and effectiveness of the classifiers or regressors. The corpus used in this work is very large, containing more than one million examples. Furthermore, because many of the models will employ acoustic features, the model input dimensionality is high. Taking this into consideration, the methods employed must have tractability in computation time and resources, and must be generalizable to large amounts of data. Because an evaluation of musical attributes at this scale has yet to be performed, it is necessary to start with simpler linear methods. After evaluating simpler methods, a set of non-linear methods is introduced. However, when selecting more complex methods, the training and example prediction of models must be able to take advantage of parallel computing architectures.

Finally, because human centered music data is used, it is important to use models that are trained with human-interpretable parameters and produce human-interpretable results. While deep learning and neural network approaches are popular for big datasets, the learned mappings to a feature space are not always musically intuitive. Because a facet of this thesis is to quantify and capture components of musical rhythm from a grounded perspective, it is important that the methods have intuitive interpretations that can be explained in the domain of music. This gives insight into “how and why” rather than just solving a discrimination and regression task.

4.1 Linear Models

4.1.1 Linear Regression

Some of the attribute prediction tasks in later chapters require regression of *continuous labels*. These continuous attributes are first predicted with least squares *Linear Regression*. The goal of linear

regression is to fit a line to a set of data points and labels. A best-fit line is used to model the data; A new unlabeled data point can be predicted by calculating its location on that line.

Given a dataset with features X and labels Y , a linear mapping is learned with intercept β_0 and slope β_1 .

$$Y = \beta_0 + \beta_1 X \quad (4.1)$$

Because each feature x_i will not fit exactly on this line, each example can be represented by Equation 4.2. In this case ϵ refers to the error distance of the label y_i to the estimated line.

$$y_i = \beta_0 + \beta_1 x_i + \epsilon_i. \quad (4.2)$$

For each data point x_i and label y_i , the goal is to minimize the error ϵ_i . By using the mean of the features \bar{x} , the mean of the labels \bar{y} , the variance of the features σ_x^2 , and variance of the labels σ_y^2 , the intercept β_0 and slope β_1 can be found such that the error ϵ (both positive and negative values) over all examples sums to zero.

$$0 = \sum_{i=1}^N \epsilon_i \quad (4.3)$$

The value for β_1 (slope) is estimated using the covariance of X and Y (rise) and the variance of X (range). The variances σ_x^2 and σ_y^2 of X and Y are shown in Equations 4.4 and 4.5. The name least squares regression comes from the squared error $(x_i - \bar{x})^2$ present when finding these variances.

$$\sigma_x^2 = \frac{1}{N} \sum_{i=1}^N (x_i - \bar{x})^2 \quad (4.4)$$

$$\sigma_y^2 = \frac{1}{N} \sum_{i=1}^N (y_i - \bar{y})^2 \quad (4.5)$$

The covariance Σ of X and Y is given in Equation 4.6.

$$\Sigma(X, Y) = \frac{1}{N} \sum_{i=1}^N (x_i - \bar{x})(y_i - \bar{y}) \quad (4.6)$$

With the variances and covariance defined, the estimated slope $\hat{\beta}_1$ is given by

$$\hat{\beta}_1 = \frac{\Sigma(X, Y)}{\sigma_x^2} \quad (4.7)$$

In order to find the estimated intercept $\hat{\beta}_0$, the recently found slope $\hat{\beta}_1$ can be used in conjunction with the mean values of the data \bar{x} and \bar{y} . This is shown in Equation 4.9. An intuitive toy example of linear regression is shown in Figure 4.1.

$$\bar{y} = \hat{\beta}_0 + \hat{\beta}_1 \bar{x} \quad (4.8)$$

$$\hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x} \quad (4.9)$$

With estimates of $\hat{\beta}_0$ and $\hat{\beta}_1$ the model becomes:

$$Y = \hat{\beta}_0 + \hat{\beta}_1 X \quad (4.10)$$

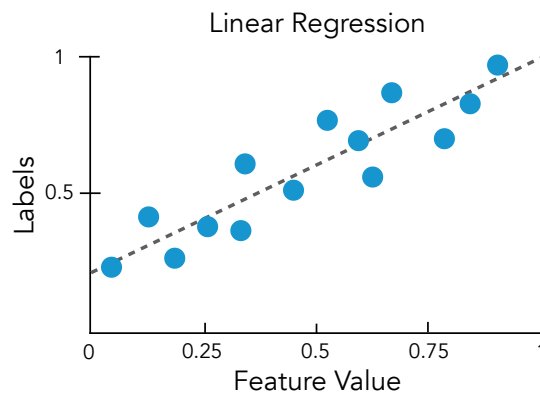


Figure 4.1: An example of linear regression.

4.1.2 Logistic Regression

The *Logistic Regression* classification method is similar in motivation to linear regression, but with the goal of classification of *binary labels* and not regression (the name is deceptive). A binary prediction (or probability) of present (1) or not present (0) is desired.

The the predictor for linear regression with parameters β_0 and β_1 is shown in Equation 4.11.

$$Y = \beta_0 + \beta_1 X = \beta^T X \quad (4.11)$$

This predictor is similar in motivation to *Linear Regression*, however Y is now constrained by $Y \in \{0, 1\}$. In order to accommodate this a sigmoid function is used. This is an S-shaped function $\sigma(\alpha)$ with asymptotes at 0 and 1 and a value of 0.5 at the center. The sigmoid function is shown in Equation 4.12. A plot of the sigmoid function is shown in Figure 4.2.

$$\sigma(\alpha) = \frac{1}{1 + e^{-\alpha}} \quad (4.12)$$

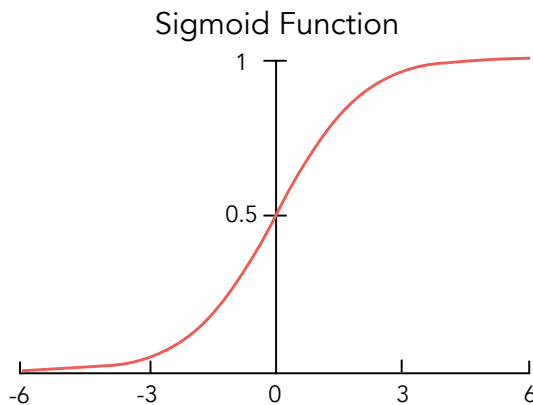


Figure 4.2: A sigmoid function

In order to force the predicted label values Y to be $Y \in \{0, 1\}$, the sigmoid function is applied to the weights β and data X . The new formulation is shown in Equation 4.13.

$$\hat{Y} = \sigma(\beta^T X) \quad (4.13)$$

This forces \hat{Y} such that $\hat{Y} \in [0, 1]$. The output \hat{Y} is still continuous though, and because of that, this is interpreted as the probability of a desired positive class $\hat{Y} = P(Y|X; \beta)$. The decision boundary for evenly weighted classes becomes the point in the sigmoid where $\hat{Y} = 0.5$. Values $\hat{Y} \geq 0.5$ are assigned $Y = 1$ and values $\hat{Y} < 0.5$ are assigned $Y = 0$. The updated expression is shown in Equation 4.14, with parameters β and features X . This boundary can also be shifted to accommodate a weighted representation or over-representation of each class.

$$P(Y|X; \beta) = \sigma(\beta^T X) \quad (4.14)$$

Similar to linear regression, the model parameters β need to be learned in order to fit the sigmoid to the logistic data. In order to estimate β , the maximum likelihood estimate (MLE) of Equation 4.14 must be found. Instead of fitting a line to the data, this can now be seen as fitting a line or plane to a decision boundary, or the point where $P(Y|X; \beta) = 0.5$. Each feature dimension is also fit separately, making each β_i independent from one another. An intuitive toy example of logistic regression is shown in Figure 4.3.

For high-dimensional data, an intuitive interpretation of the learned β is the weighting and correlation of a feature to the probability of selecting a specific class. For example, if β is positive and relatively large, it means that an increase in a specific feature leads the higher probability of the positive class occurring, and it is really strong relationship. Relative weightings can inform the classifier of the the importance of a feature. The sign denotes the features relation to a positive or negative decision. Because uninformative features will have lower relative weights, they will less effect on the decision made. This allows logistic regression to be less affected by the *curse of dimensionality*. It also allows the use of logistic regression as an identifier of important factors as

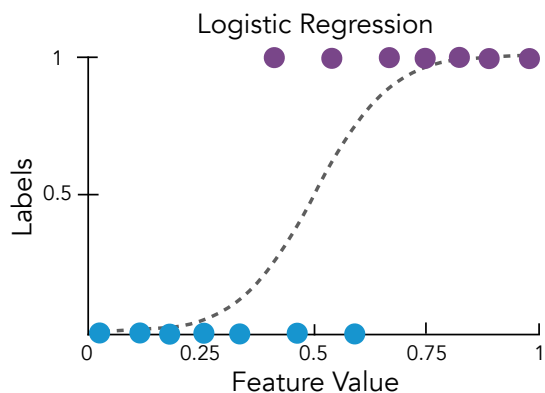


Figure 4.3: An example of logistic regression

they relate to a certain outcome. This can lead to an intuitive understanding of which features are important when analyzing rhythm as well as which feature components best quantify rhythm as interpreted by humans.

4.1.3 Using Large Datasets

In practice, on large datasets, models are trained using *Stochastic Gradient Descent* (SGD). *Gradient Descent* is a generic way to train a model by iteratively updating model parameters relative to a set *learning rate* and recomputing an objective function with the goal of reducing model error with each update. Unlike standard *Gradient Descent*, which computes the error sum across all the examples before making an update, SGD computes the error and updates the parameters relative to the fit of an individual example. This makes it possible to start tuning parameters without having to compute large error sums across all examples. Because each example is seen independently, the error does not always monotonically decrease from example to example. However, it will decrease over time, and converge much more quickly than standard gradient descent. This can be used to train a variety of algorithms, including linear and logistic regression, by simply adjusting the learning rate and objective function. Examples of gradient descent and stochastic gradient descent for a quadratic ‘bowl-shaped’ gradient is shown in Figure 4.4.

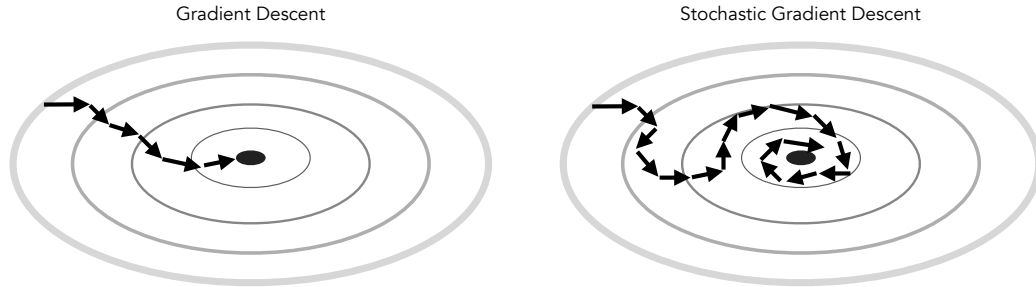


Figure 4.4: An example of standard gradient descent (left) and stochastic gradient descent (right) for a quadratic bowl-shaped gradient surface. Arrows depict the model error trajectories.

For *Linear Regression*, an initial slope and intercept is estimated. Based on the learning rate the slope and intercept values are updated in a manner to reduce the sum of the squared distance of a training example to the estimated line (the model). This represents a *least-squares loss*. For *Logistic Regression* a similar process is employed using *logistic (log) loss*.

4.2 Decision Tree Ensembles

4.2.1 Binary Decision Trees

A *Binary Decision Tree* is a structure that models features X and labels Y with a set of sequential, binary decisions. An example binary tree is shown in Figure 4.5. This toy example displays whether a music group should rehearse based the set of previous observations and motivating features of those observations. These features and observations shown in Table 4.1.

	Perform Soon?	Time of Week	Time of Day	Rehearse?
$X_1 =$	Yes	Sat&Sun	Morning	$Y_1 =$ No
$X_2 =$	No	M-F	Night	$Y_2 =$ No
$X_3 =$	Yes	M-F	Morning	$Y_3 =$ Yes
$X_4 =$	No	M-F	Morning	$Y_4 =$ Yes
...
$X_N =$	Yes	Sat&Sun	Night	$Y_N =$ Yes

Table 4.1: Should the music group rehearse? The probability of rehearsal can be predicted from past experience based on the imminence of a performance, the time of week, and the time of day.

In order to select which features to split on, the *information gain* metric is employed. The feature with the maximum information gain is the one chosen for the parent node split. Equations 4.15 and

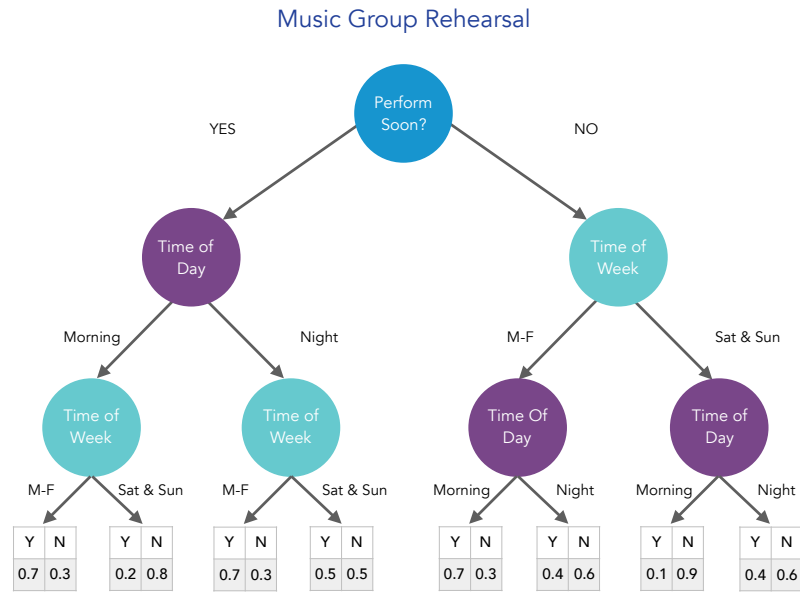


Figure 4.5: A binary decision tree is generated. The leaves terminate with the probability of the positive class label (a rehearsal) present after traversal.

4.16 show the information gain in the context of trees. Successive splits are found recursively in order to grow parent and child node branches of successive decisions.

Information gain is found through a difference in entropy E of the labels Y with and without a feature X . Where p_c is the probability of a class $c \in C$ in labels Y , the entropy $E(Y)$ of the labels is shown in Equation 4.15.

$$E(Y) = \sum_C -p_c \log_2 p_c \quad (4.15)$$

The information gain IG for a feature X with values X_v of all possible values $v \in V$ is

$$IG(Y, X) = E(Y) - \sum_V \frac{|X_v|}{|X|} E(Y|X_v) \quad (4.16)$$

Creation of trees requires the use of all data in the training set and the knowledge of all features. This can make for cumbersome training for very large datasets. Furthermore, because each successive parental node lowers the variance of the data contained in its children, trees have the tendency to over-fit the training data, resulting in poor test results. This over-fitting can be fixed with the use

of *ensemble methods*, or the use of many trees simultaneously to make a joint decision. A couple examples are *Random Forests* and *Gradient Boosted Trees*, which will be explained in Sections 4.2.2 and 4.2.3 respectively [122, 123]. These methods have been shown to be effective in Music Information Retrieval tasks in the past in both instrument and expression recognition [124, 125] as well as genre recognition [126].

One big advantage of using trees is their ability to handle real and semantic valued data simultaneously. They can handle these types of data in both the feature space and the label space. Also, the range of each feature does not affect the features in other dimensions because feature salience is evaluated using *information gain*. Another advantage is the nonlinearity in each tree. Because the tree greedily learns many small decisions boundaries, they can learn complex, non-linear boundaries without the need to use tuned feature kernels. For datasets that have features that are very high-dimensional, trees can be used as a form of feature selection and dimensionality reduction by deciding which subset of features are best for discrimination based on the use of information gain and the definition of feature splits.

Trees for Classification

The general formulation for the tree presented previously assumes a classification problem. A chosen class of features with an unknown label is found by traversing all decisions in the tree and assigning the class of containing the highest probability of occurrence in the lowest leaf. It is possible to build a tree to perfectly classify data without the need for a probability decision. However, in practice this leads to over-fitting the training data, leading to poorer test results.

Trees for Regression

Trees for regression are treated similarly to trees for classification. Given a set of continuous labels, a split is formed in each feature dimension at the location that minimizes an squared error. The feature split with the lowest mean squared error (instead of Information Gain) is chosen for each

successive parent and child node. Given the label mean \bar{y}_- and \bar{y}_+ for each set of labels $y \in Y_+$ and $y \in Y_-$ around the left (-) and right (+) of split s respectively,

$$\text{MSE}(s) = \sum_{y \in Y_+} (\bar{y}_+ - y_i)^2 + \sum_{y \in Y_-} (\bar{y}_- - y_i)^2. \quad (4.17)$$

When doing regression, the mean of all values present below each terminating leaf node is the value given to an query example that traverses to that point.

Continuous Valued Features

Sometimes it is necessary for features to have continuous values, as is the case in most models involving audio signal analysis. In order to deal with this, a few approaches are presented. Rather than exhaustively choose split values, they can be chosen relative to the training points, with each value denoting a split and a binary projection of features around that split. Each example and respective split is treated independently. The choice of the candidate splits are not affected by label values, so this can be used for both classification and regression.

Another method relies on sorting the features in each dimension. A split candidate is chosen at the midpoint between two consecutive sorted feature values that contain different discrete label values. Basically, split candidates are placed at label transitions. A final split is then chosen based on the maximum information gain of each of the candidate splits. Because this needs the understanding of class labels, it can only be used for classification.

4.2.2 Random Forests

As stated before, it can be very slow to train large trees. Similarly, large trees can over-fit the data because each parent minimizes the variance of the data around which each child can make a decision. This is great for modeling the training data exactly, with virtually no training error, but it can be disastrous at test time. A *Random Forest* can alleviate this problem. Instead of just one tree, many are trained in parallel. Each tree is small, and contains only a subset of the features and data (*bagging*). Because of this, each tree is different from one another, allowing for a greater capture of data variance. Each of the decisions from each tree are combined in order to make an ensemble

decision. This allows for better fitting of large datasets and allows for a distributed implementation. Sub-sampling of data and features with smaller trees can reduce training time through distributed computation as well as result in better models. This is one of the few times where the *no free lunch* concept in machine learning is violated. Usually there is a trade-off in performance versus cost. In this example, both are improved [123].

Bagging

The concept of *bagging* refers to the use of only a subset of features and examples. Each tree in the random forest is trained on only a subset of features across a subset of examples. This allows for the creation of many general trees that can be fused later. It also alleviates the problem of over-fitting on outliers.

Parameters

One downside to using trees is the large number of hyper parameters that need to be tuned. This is where most of the training cost comes from. The list below gives a brief explanation of the parameters used for random forests:

- number of features per tree
- number of examples used to train each tree
- number of trees in the “forest”
- tree depth, deep trees can lead to over-fitting.

Tree Combination: The Ensemble

The combination of many trees in a *Random Forest* is straightforward. Because each tree was trained on a different set of data and different sets of features, and because trees are grown greedily, each tree will be very different. A group decision can be found through a linear combination of each of the decisions in the ensemble. An example of an ensemble decision is for classification and regression is shown in Figure 4.6.

For classification, each terminating leaf of each tree has a multinomial probability distribution of class labels. In order to combine trees, a query example is traversed through all trees. Each traversal results in the terminal multinomial distribution of class labels. Each of these distributions

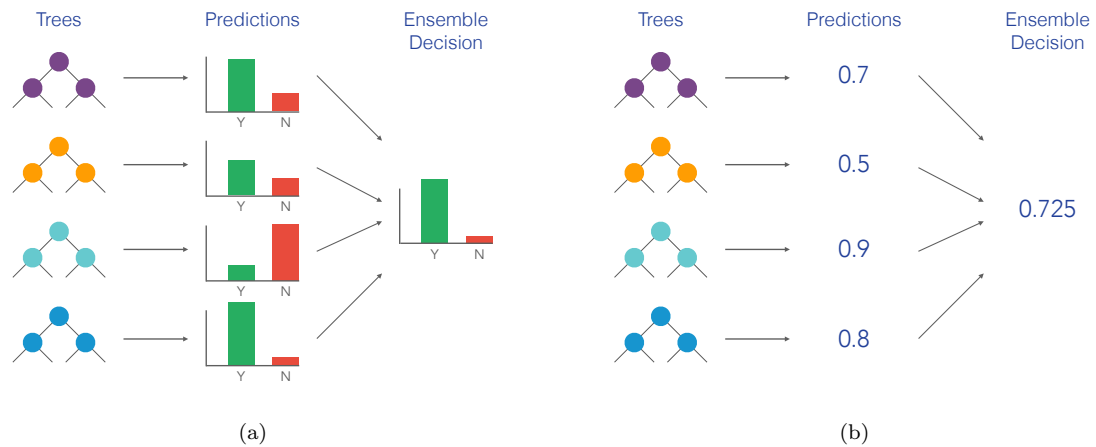


Figure 4.6: Random forest tree ensemble examples for classification (a) and regression (b).

can be summed (emphasize disagreement) or multiplied (emphasize agreement). The class with the highest probability is chosen.

For regression, the process is similar. Each terminating leaf of each tree has a value assigned to it. In order to combine trees, a query example is traversed through all trees. Statistics of the value at each of each of the terminating leaves can be used in order to estimate the continuous value label of the input features.

4.2.3 Gradient Boosted Trees

A *Gradient Boosted Tree* is another tree ensemble model. In this method many trees are successively learned hierarchically relative to the previous trees errors. This process is known as *boosting* with each tree as a *weak-learner*. Each successive tree is learned based on the failure of others [122]. Intuitively, trees in a *Random Forest* are diverse and randomly crated with the motivation that enough of them the will completely “shade” the feature space. In a *Gradient Boosted Tree*, one encompassing tree shades as much of the feature space as possible. New trees greedily sprout in the sun where previous trees do not shade.

Boosting

The concept of *boosting* refers to the usage of many simple, inexpensive *weak-learner* classifiers in combination to achieve a more powerful and complex classifier. In a *Gradient Boosted Tree*, the *weak-learner* is a small tree. In boosting, each member of the ensemble of weak-learners is forced to be an expert on the errors of its predecessor. Training examples are iteratively re-weighted based on these errors. Classification errors of each predecessor are weighted more than the examples it got right. The weighted decisions of each successive tree are aggregated in order to make the final decision.

In the regression formulation, gradient boosted trees are learned through *residual fitting*. This means that each successive regression split is found on the residual in the data by normalizing out the mean of each previous split and fitting. The final model becomes the summed regression of all trees. This sum re-adds the learned means into the data in order to model the original function. This formulation is known as gradient boosting because as each tree is created in succession, the error decreases. The goal is to choose parameters and trees that travel downward along this error gradient. A visual representation of a gradient boosted tree is shown in Figure 4.7.

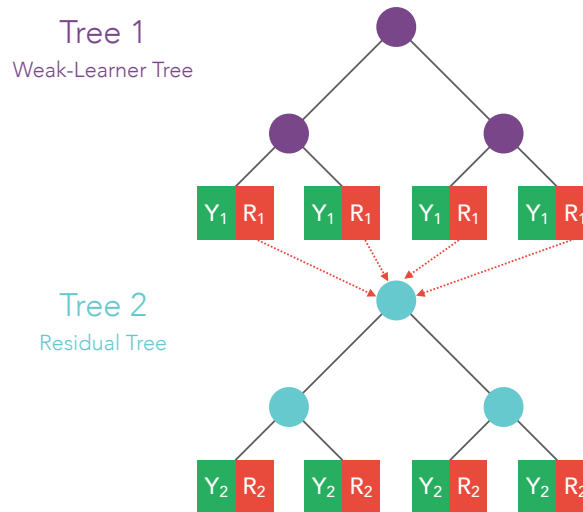


Figure 4.7: A Gradient Boosted Tree example.

Shrinkage

In order to combat the increase in the gradient function, the *learning rate* can be changed. This alters the weights given to positive and negative examples, making the tree learn faster or slower. Using a smaller learning rate will take more steps to reach its optimal point in the error gradient, however, the error it settles on will be lower than one at a larger rate. Each step is more incremental. This process is known as *shrinkage*.

Stochastic Gradient Boosting

Using too many boosted trees in combination can lead to over fitting. In order to combat this, a small sampling of features and training examples is used at each step. This is similar to *bagging* in random forests and *Stochastic Gradient Descent* in Section 4.1.3. The gradient becomes a bit noisier, but similar to the benefits of SGD, *Stochastic Gradient Boosting* reduces run time and creates a more robust model.

Parameters

One downside to using trees is the large number of hyper parameters that need to be tuned. This is where most of the training cost accumulates. The list below gives a brief explanation of the parameters used for gradient boosted trees optimized through grid search and cross validation.

- number of trees
- learning rate (shrinkage)
- tree depth, deep trees can lead to over-fitting.
- minimum allowable samples in a leaf, prevents over-fitting
- number of features per tree (stochastic gradient boosting)
- number of examples used to train each tree (stochastic gradient boosting)

4.2.4 Hybrid Tree Ensemble Models

In addition to using tree ensembles directly for classification and regression, they can be used as a feature transformation capable of learning complex feature interactions [127]. In a tree ensemble, the activation of each terminating leaf can be transformed into a feature vector. This new feature

vector can then be used in a simpler classification and regression model. An overview of this process is shown in Figure 4.8.

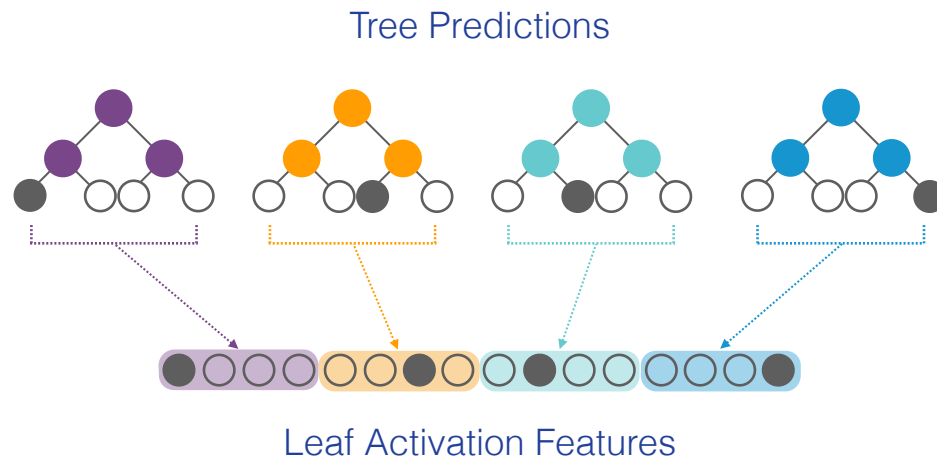


Figure 4.8: Leaf activations of each tree in the ensemble become features for another model. This learns interactions of branches.

4.3 Supervised Model Evaluation

4.3.1 Evaluating Classification Models

When evaluating and comparing classification models, it is important to measure how well your system correctly predicts only the desired examples and avoids others. In order to evaluate this, a few different measures of correctness must be computed, namely:

- *True Positive* (TP): Correctly classify a positive example (hit)
- *True Negative* (TN): Correctly classify a negative example (rejection)
- *False Positive* (FP): Incorrectly classify a negative example as positive (false alarm)
- *False Negative* (FN): Incorrectly classify a positive example as negative (miss)

Subsets of these correctness rates are used in combination to compute metrics to evaluate the efficacy of a classification system. These metrics will be explained in the following subsections. Precision vs. Recall Metrics and Receiver Operator Characteristic Curves, along with inversely weighting training by label occurrence, are ways to evaluate systems that have labels that are not

evenly distributed. For example, if one class is very unlikely, predicting that class is absent 100% of the time would provide high raw accuracy, but tell you nothing about what the system is learning.

Precision vs. Recall Metrics

One way to see if a recommendation is accurate is through the *F-Score* or *F-Measure*. This is a metric of recommendation precision vs. recommendation recall. A good recommendation system has a high F-Score, meaning that all of the right items are recommended without recommending those that are not relevant. F-Score is the harmonic mean of precision p , the number of correct positive results retrieved over all positive ($p = \frac{TP}{TP+FP}$) results estimated, and the recall r , the number of positive results that should have been retrieved ($r = \frac{TP}{TP+FN}$). This is shown in Equation 4.18

$$F_1 = 2 \frac{pr}{p+r} \quad (4.18)$$

The score can also be weighted by the amount of desired precision or recall. The general form of the F_n score is shown in Equation 4.19. Values of $1 > n > 0$ weights precision more than recall, and values of $n > 1$ weights recall higher than precision.

$$F_n = (1 + n^2) \frac{pr}{n^2p + r} \quad (4.19)$$

In addition to a single F-Score, a Precision-Recall Curve (pr-curve) can be computed. This curve is generated by sweeping the probability of classification threshold and evaluating the precision and recall at each of these probability threshold points. The area under the pr-curve (AU-PR) can be used to evaluate a classification system. By sweeping the threshold, this metric is less sensitive to class imbalance. An area of 1.0 represents perfect classification. An area of 0.0 represents a random classification. AU-PR is a beneficial metric because it can be used to evaluate the system without having to set a static classification threshold. When reporting a single F-Score, a threshold sweep is performed and the maximum score relating to that sweep is reported. In that case, the decision boundary tends to shift to correct for the class imbalance.

Receiver Operator Characteristic Curves

Another metric for evaluation of classification models that deals with class imbalance is the Receiver Operator Characteristic Curve (ROC curve). Similar to the pr-curve, a threshold sweep is performed. The true positive rate ($\frac{TP}{TP+FN}$, identical to recall) is plotted vs. the false positive rate ($\frac{FP}{FP+TN}$). The area under the curve (AUC) becomes another measure of accuracy. An AUC of 1.0 means perfect classification. An area of 0.5 means random classification. This is another metric that can be used to evaluate the system without having to set a static classification threshold.

4.3.2 Evaluating Regression Models

In order to evaluate the model, one can compare the mean absolute error and mean squared error of the data contained in the model vs a set of supervised regression experiments. If these numbers are similar, it will show that the data is not over-fitting. If these numbers are vastly different, it could represent an over-fitting of the data. Another way to test the model is the r^2 metric. This is a measure of the explained variance versus the total variance. The explained variance is the variance the model captures, and the unexplained variance is what is left over. For each value x_i a point \hat{y}_i is calculated based on the estimates of $\hat{\beta}_0$ and $\hat{\beta}_1$.

$$\hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_i. \quad (4.20)$$

The squared error ϵ_{res}^2 of the residual is given by the squared difference of the calculated value \hat{y}_i and true value y_i .

$$\epsilon_{\text{res}}^2 = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2 \quad (4.21)$$

This is compared to the variance σ_y^2 of all data in Y .

$$\sigma_y^2 = \frac{1}{N} \sum_{i=1}^N (y_i - \bar{y})^2 \quad (4.22)$$

The measure of explained variance r^2 is the percentage of total variance contained within the model. If the error of the model is low compared to the variance of the data, the model is explaining

a high percentage of the variation in the data, making it a good model. If the residual error is equal to the variance of the raw labels, the model is representing a low percentage of the variation in the data, making it a poor model.

$$r^2 = 1 - \frac{\epsilon_{\text{res}}^2}{\sigma_y^2} \quad (4.23)$$

4.4 Visualizing High Dimensional Data

The previous sections outlined methods to capture features of rhythm in music and evaluate their effectiveness on expertly-labeled, specifically-targeted, attributes. However, most peoples' perception of similarities and differences in rhythm does not rely on individually quantifying each of the components that construct it. In this section, I will outline a few methods that try to capture rhythmic similarity and dissimilarity perception on a set of low dimensional criteria. These sets of criteria will be defined by a resulting subspace created by reducing the dimensionality of the rhythmic attributes, the audio features that saliently define these attributes, and the styles defined by both the labeled attributes and the salient audio features. In order to achieve this, a few widely used dimensionality reduction techniques are explored. The goal of dimensionality reduction is to maintain the high dimensional differences or similarities in a lower dimensional space. In the later chapters, I explore two classes of reduction techniques: parametric basis decompositions (PCA, ICA, NMF, etc.) and non-parametric similarity mappings (t-SNE). In the non-parametric method of *t-Distributed Stochastic Neighbor Embedding (t-SNE)*, distributions are used to model local distances in the high dimensional space. A low dimensional space is generated that maintains these distance distributions and reproduce local similarity and global dissimilarity.

4.4.1 Basis Decompositions

One class of dimensionality reduction techniques, basis decomposition, involves the formulation of basis vectors and activation vectors. The basis vectors are template-like components that can be summed to re-create an original example based on their relative activations for that example. *Multidimensional Scaling (MDS)* is a technique used to model dissimilarities in data and *Principal Components Analysis (PCA)* is MDS using euclidean distance as a measure of dissimilarity. In PCA

a set of orthogonal bases and corresponding activations are found such that the variance along the activation of these bases is maximized. *Independent Components Analysis* (ICA) creates a set of independent signal components optimized across higher order statistics (i.e., kurtosis) as opposed to the mean and variance (PCA).

Non-Negative Matrix Factorization (NMF) is similar in interpretation, with a learned set of non-negative bases being summed in order to recreate an example. NMF is more constrained in that the bases and activations are strictly non-negative, resulting in an additive summation of bases, and thus, a more interpretable set of basis components.

MultiDimensional Scaling and PCA

In *Multidimensional Scaling* (MDS), the goal is to model high-dimensional data in a low-dimensional space. In order to do this, it tries to preserve the differences of points in the high-dimensional and low-dimensional spaces. If the distance metric used is *euclidean distance*, MDS will be equivalent to PCA. However, MDS has the ability to use any method of dissimilarity. Alternatively, one can also invert a measure of similarity and transform it to a dissimilarity metric and use it for MDS [128].

Classical MDS starts with a data dissimilarity matrix \mathbf{D} obtained from the chosen difference metric. This matrix contains individual dissimilarities d_{ij} from each of the examples to all other examples, where N is the number of examples, $i \in \{1, \dots, N\}$, and $j \in \{1, \dots, N\}$. A centering matrix \mathbf{C}_N is applied to the matrix \mathbf{D} in order to assign a relative anchor location of our dissimilarities to the origin of the new space. This centering matrix \mathbf{C}_N is shown in Equation 4.24, where $\mathbf{1}$ a matrix with all elements having the value 1.

$$\mathbf{C}_N = \mathbf{I} - \frac{1}{N} \mathbf{1} \mathbf{1}^T \quad (4.24)$$

This centering matrix \mathbf{C}_N is applied to the dissimilarity matrix \mathbf{D} to create a new matrix \mathbf{B} to represent the centered data. This is shown in Equation 4.25.

$$\mathbf{B} = -\frac{1}{2} \mathbf{C}_N \mathbf{D} \mathbf{C}_N \quad (4.25)$$

With this centered data, a matrix \mathbf{X}_m of new dimensionality $m = \{1, \dots, N\}$ can be estimated. The matrix \mathbf{B} can be formulated as $\mathbf{B} = \mathbf{X}\mathbf{X}^\top$, and can be factorized through eigenvalue decomposition. By extracting the m largest eigenvalues $\lambda_n, n \in \{1, \dots, m\}$ in $\mathbf{\Lambda}_m$ and the corresponding eigenvectors $\mathbf{v}_n, n \in \{1, \dots, m\}$ in \mathbf{V}_m , a lower-dimensional approximation for \mathbf{X}_m can be estimated. This process is shown in Equations 4.26, 4.27, 4.28, and 4.29.

$$\mathbf{B} = \mathbf{X}\mathbf{X}^\top \quad (4.26)$$

$$\mathbf{X}\mathbf{X}^\top = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^\top \quad (4.27)$$

$$\mathbf{X}_m\mathbf{X}_m^\top = \mathbf{V}_m\mathbf{\Lambda}_m\mathbf{V}_m^\top \quad (4.28)$$

$$\mathbf{X}_m = \mathbf{V}_m\mathbf{\Lambda}_m^{1/2} \quad (4.29)$$

Independent Component Analysis

Previously, the goal of PCA was to minimize the covariance of the data and create a set of orthogonal vectors ordered by high intra-dimensional variance. This means that PCA is useful when the data is Gaussian and linear. However, this is not always the case, so higher-order statistics more than just the 1st (expectation, mean) and 2nd moments (variance) must also be considered. Rather than minimizing the covariance as in PCA, *Independent Components Analysis* (ICA) uses higher order statistics to minimize mutual information of the output. ICA creates a set of independent components of non-Gaussian signals or features. However, due to the reliance on higher order statistics, variance of each component cannot be determined, and the order of dominant components can not be ranked. Furthermore, each of the resulting dimensions do not need to be orthogonal [129].

The goal of ICA is to find a set of components s and a square mixing matrix A such that a signal or feature space x can be captured through the combination of s and A . This follows the standard

component/activation formulation shown in Equation 4.30

$$\mathbf{x} = \mathbf{A}\mathbf{s} \quad (4.30)$$

In order to determine each of the components from n example observations $\{x^{(n)}; n = 1, \dots, N\}$ an un-mixing matrix $\mathbf{W} = \mathbf{A}^{-1}$ is estimated. The sources of an example $s^{(n)}$ can be recovered through the expression shown in Equation 4.31.

$$x^{(n)} = \mathbf{W}s^{(n)} \quad (4.31)$$

Given that there are M sources we can denote the contribution of each source in each example as $\{s_m^{(n)}; m = 1, \dots, M\}$. Where w_n^T denotes the n -th row of \mathbf{W} , the m -th source can be recovered by the expression in Equation 4.32.

$$s_m^{(n)} = w_n^T x^{(n)} \quad (4.32)$$

Non-Negative Matrix Factorization

The goal of *Non-Negative Matrix Factorization* is to factorize a matrix \mathbf{V} into two matrices \mathbf{W} and \mathbf{H} . This factorization is shown in Equation 4.33, where n is the number of examples, m is the number feature dimensions, and r is the number of basis components to be learned.

$$\mathbf{V}^{[n \times m]} \approx \mathbf{W}^{[n \times r]} \times \mathbf{H}^{[r \times m]} \quad (4.33)$$

NMF is performed by iteratively updating \mathbf{W} and \mathbf{H} and minimizing a cost function that relates $\mathbf{W} \times \mathbf{H}$ to \mathbf{V} . This cost function is usually defined in terms of the euclidean distance or the *K-L divergence* between $\mathbf{W} \times \mathbf{H}$ and \mathbf{V} . In practice rows of \mathbf{H} are a defined number non-negative basis components, or a set of “feature parts”, learned across all examples. The columns of \mathbf{W} are non-negative activations of those components, or “feature part” emphasis vectors. The dimensionality of the original feature space can be reduced by selecting the number of bases to be learned, and treating the activations as the new feature space. This effectively reduces the dimensionality in a

manner that is still representative of the original structure, with components representing a set of intuitive pieces that can recreate an estimation of an example through a sum that is scaled by the activations [130, 131]

NMF can be preferred over PCA because PCA only employs a weak orthogonality constraint. The PCA representation is allowed to use cancellations as well as additions of components in order to represent the original space, which is not always intuitive. Because NMF enforces non-negativity, each of the components can be more intuitively seen as additive parts, or building blocks, that can be summed to estimate the original feature space. This can give more understandable mappings of the original feature space, and allow for more concrete judgments of the original features and their behavior across a variety of tasks.

4.4.2 t-Distributed Stochastic Neighbor Embedding

Similar to the dimensionality methods suggested so far *t-Distributed Stochastic Neighbor Embedding (t-SNE)* attempts to build a map in which high-dimensional relationships are maintained in a lower-dimensional space. It aims to preserve local pairwise relationships, and focus less on large global relationships. However, instead of focusing on *differences* as in MDS, the t-SNE algorithm is motivated by maintaining local *similarities* between points.

Definition of t-SNE

The process of t-SNE is as follows. In a high-dimensional space, it is important to preserve local similarities of those high-dimensional objects. Given a point in space x_i , a Gaussian distribution is defined and centered at that point x_i . The similarity is then the normalized measure of density $p_{i,j}$ of all other points under this Gaussian. The density for $p_{i,j}$ is shown in Equation 4.34.

$$p_{i,j} = \frac{\exp\left(\frac{-\|x_i - x_j\|^2}{2\sigma^2}\right)}{\sum_k \sum_{l \neq k} \exp\left(\frac{-\|x_k - x_l\|^2}{2\sigma^2}\right)} \quad (4.34)$$

This provides a set of probabilities $p_{i,j}$ that measures the similarity of a point (x_i, x_j) . It represents the probability distribution over pairs where the probability of picking a pair is the *similarity* metric. Close points in the high-dimensional space will have a large joint probability $p_{i,j}$;

far points will have a small joint probability $p_{i,j}$. However, in practice, the joint distribution $p_{i,j}$ is not tractable to compute directly, so instead, the conditional probability $p_{i|j}$ is used to find it. This is shown in Equation 4.35.

$$p_{i|j} = \frac{\exp\left(\frac{-\|x_i - x_j\|^2}{2\sigma_i^2}\right)}{\sum_{i \neq j} \exp\left(\frac{-\|x_i - x_j\|^2}{2\sigma_i^2}\right)} \quad (4.35)$$

This reduces the normalization from being performed over all points and only focuses on point x_i . Additionally the bandwidth of the distribution can be scaled by σ_i . This allows for a fixed conditional perplexity and can ensure a set number of points fall within the mode of the Gaussian. This allows the model to adapt to different local densities throughout the space. This also becomes a clustering parameter that also allows the model to more tightly or loosely “squeeze” similar points in the eventual lower-dimensional subspace.

Even though the joint probability $p_{i,j}$ is hard to compute directly, it is still necessary as the measure of similarity. Assuming that the conditionals $p_{i|j}$ and $p_{j|i}$ are symmetric, they can be averaged in order to find the joint distribution. This is shown in Equation 4.36.

$$p_{i,j} = \frac{p_{i|j} + p_{j|i}}{2N} \quad (4.36)$$

The goal of t-SNE is to maintain a similarity obtained with a joint probability $p_{i,j}$ in a high-dimensional space in the low-dimensional space. The distribution in the low-dimensional space $q_{i,j}$ is learned with the goal of keeping the similarities the same. The distribution in the low-dimensional space $q_{i,j}$ is shown in Equation 4.37.

$$q_{i,j} = \frac{\exp(-\|y_i - y_j\|^2)}{\sum_k \sum_{l \neq k} \exp(-\|y_k - y_l\|^2)} \quad (4.37)$$

In order to optimize similarity of the low-dimensional space to the higher one, the *K-L divergence* is used. If both spaces allow for similar probability distributions for each data point, while little can be said about global structure, local structure is strongly preserved. Because the local structure

is preserved, it can cause the distance of further points to expand. An intuitive example of this is shown in Figure 4.9.

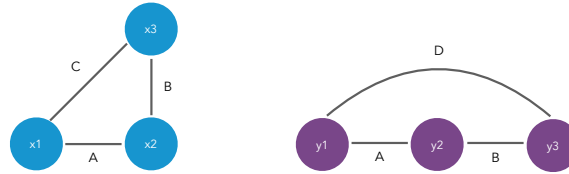


Figure 4.9: t-SNE maintains small distances, but can expand further ones.

Because of this increase in far distances, it is necessary to model the distribution of points in the low-dimensionality space $q_{i,j}$ such that it will still contain the probability mass in the center but have longer tails. To accommodate this warping of far points, the students-t distribution is chosen as the low-dimension model $q'_{i,j}$. The new expression for the students-t motivated distribution $q'_{i,j}$ is shown in Equation 4.38.

$$q'_{i,j} = \frac{(1 + \|y_i - y_j\|^2)^{-1}}{\sum_k \sum_{l \neq k} (1 + \|y_k - y_l\|^2)^{-1}} \quad (4.38)$$

4.4.3 Exploring Non-parametric Spaces

Unlike PCA or NMF, the resulting space from methods such as t-SNE are non-parametric, meaning it is difficult to interpret what the dimensions mean in terms of the original feature space. The only assumptions that can be made are that points close to each other in the t-SNE space are mapped as such because they were close in the original feature space. In PCA or NMF, it was easy to see how each of the basis components relate to the original features, and through their activations, it is possible to intuit an understanding of the activation space based on how much each of each component was present in an example. This is not true of t-SNE, so a method to explore and understand the new space in terms of the original feature space is desired. In order to accomplish this, a clustering inspired method can be employed. Based on the clusters and their statistics in the t-SNE space, representative cluster information in the original feature space can be generated. I will motivate this proposed process through a simple toy example. Later, in Sections 4.4.3 and 4.4.3, a few examples using real data are presented.

Given a dataset of high dimensionality, t-SNE allows for the creation of a non-parametric low dimensional space. In this space, it is important to remember that only local similarity is preserved in a meaningful way. An example of this projection in 2 dimensions is shown in Figure 4.10.

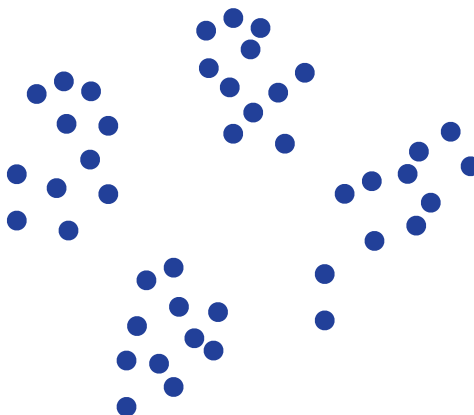


Figure 4.10: 2 dimensional t-SNE projection of a toy example.

Because t-SNE maintains local similarities between points, clustering in the t-SNE space can provide a way to create collections of similar points and separate them from globally non-similar ones. Doing k-means clustering for $k = 4$ clusters for the toy data is shown in Figure 4.11. When using k-means clustering, each point can be assigned to a cluster based on its distance to a cluster mean. Cluster assignment and computation of the mean is an iterative process. The process finishes when the objective of minimizing intra-cluster (inside cluster) distance and maximizing inter-cluster (between cluster) distances is met. The data in each cluster can be summarized by the value at its center of mass (cluster mean).

Because t-SNE is a non-parametric projection, the location of the means in t-SNE space can not be projected back to the original feature space. However, in order to understand the features' placement in the t-SNE space, this projection, or an estimate thereof, is desired. Once again, because t-SNE is designed to preserve local relationships, local similarity of examples in the original space should be preserved in the t-SNE space. Therefore, because of those local similarities, taking the mean of similar (close) points in the t-SNE space should be analogous to taking the mean of similar points in the original feature space. In order to generate a cluster mean the original feature space

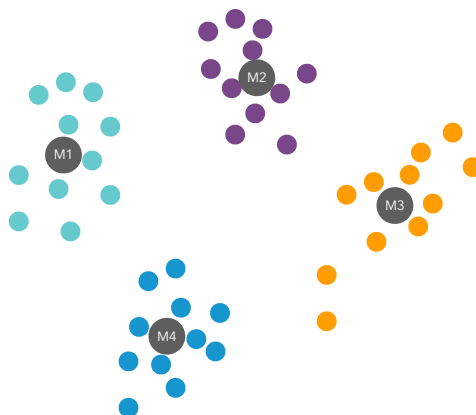


Figure 4.11: K-means clustering ($k=4$) of the toy example in the t-SNE space

analogous to the mean in the t-SNE space, a set of $n = 3$ *Nearest-Neighbors* relative to each cluster mean can be selected. An visual representation of this selection process is shown in Figure 4.12.

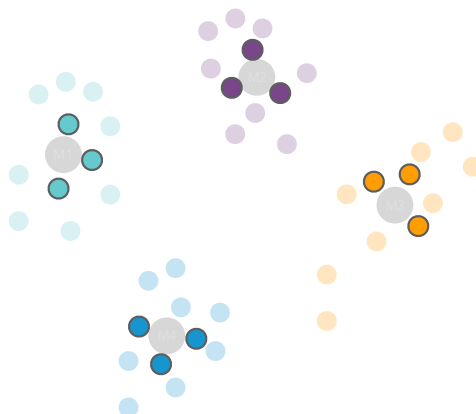


Figure 4.12: Nearest neighbors of the cluster means in t-SNE space can be used to approximate cluster means in the original feature space.

It is possible to find the mean in the original feature space using the mean of all data points in a t-SNE cluster. However, this will provide a less accurate snapshot of the data point due to the long-tailed nature of similarity versus difference in the t-SNE space. As the clusters expand through the t-SNE space, the notion of local similarity is no longer preserved, making it less fundamentally sound to describe the cluster mean in terms of the original space. If the cluster mean point existed in the original space, it would only be similar to the local points surrounding it. Therefore, when

estimating that projection, we should only use a small number of locally similar points, hence the *Nearest-Neighbors* approach.

An Intuitive Example

In this section, I provide a more intuitive example using a subset of 10k examples from the MNIST hand written digit dataset. Each example is a vectorized version 28x28 grayscale image of a hand written digit from 0-9. The steps to create the t-SNE reduction are as follows [117]:

1. select 10k subset of examples at random
2. vectorize each 28x28 gray-scale image to a single 784 dimensional feature vector
3. perform PCA to obtain 30 components to reduce dimensionality of data
4. perform t-SNE on the new 30 dimensional feature space to obtain a 2-D projection (Figure 4.13)

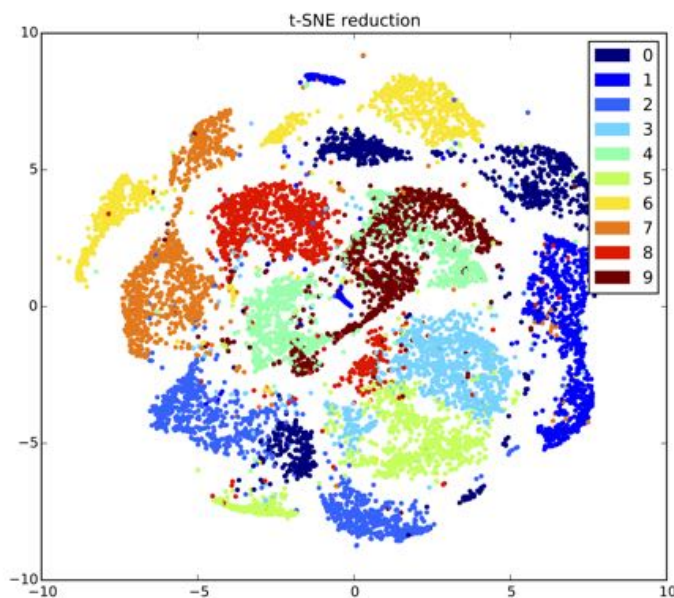


Figure 4.13: 2D Projection of MNIST using t-SNE. Ground truth labels are shown in different colors.

Because t-SNE is a non-parametric projection, relying only on similarity, it is hard to define what different locations in the space refer to. In order explore the space, I implemented the methods described previously in Section 4.4.3. The steps are as follows:

1. perform k-means clustering in t-SNE space (Figure 4.14).

2. find a set number of the nearest neighbors closest to each cluster center.
3. take the mean of the cluster center neighbors in the original feature space(Figure 4.15).

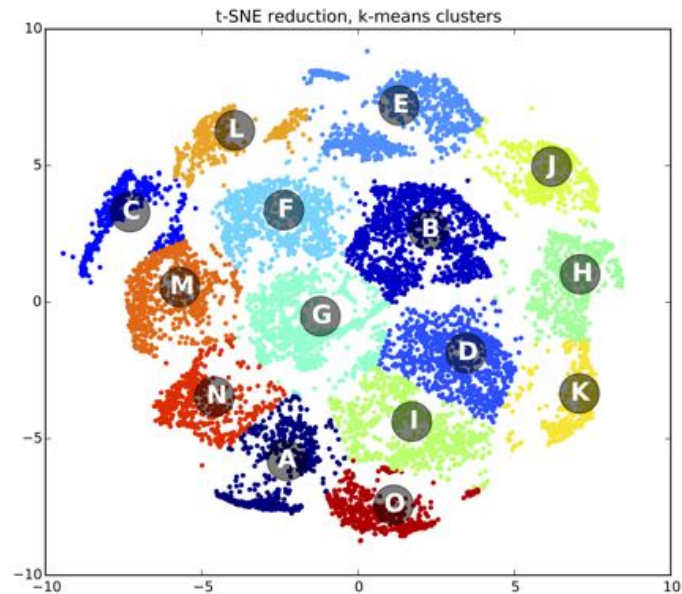


Figure 4.14: 2D Projection of MNIST using t-SNE. k-means clusters and cluster centers (letter labels) are shown.

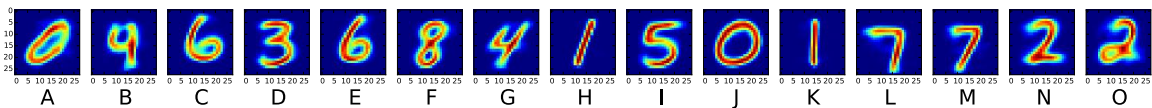


Figure 4.15: The means of the cluster center nearest neighbors in the original feature space are shown.

In this example, it is seen that the estimated cluster centers strongly resemble the different digits. While some clusters (i.e., cluster B) combine sometimes difficult to distinguish numbers (4 vs. 9), it also is good at distinguishing different types (slanted, looped) of the same number (0:A&J, 1:H&K, 2:N&O, 4:B&G, 6:C&E, 7:L&M). This provides some evidence that similarity is focused on the geometric shape of each digit. Similarly shaped digits are placed close together.

A Rhythm Example

In this example, a t-SNE space is learned from rhythm acoustic features. Each of the rhythm features is described in more detail in Chapter 6. The t-SNE space and a selection of 5 of the nearest neighbor means is shown in Figure 4.16.

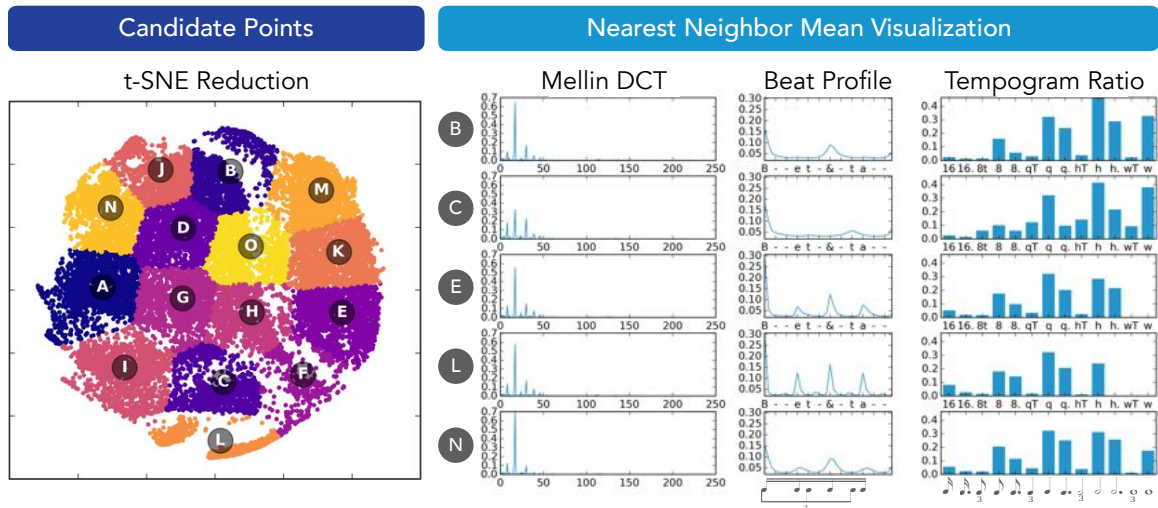


Figure 4.16: Rhythm features are reduced using t-SNE and candidate points are selected (left). From those candidate points, nearest neighbor means are computed in the rhythm feature space (right). A subset of 5 (from 15) means is shown.

This rhythm feature reduction example shares similarities with the handwritten digit example (Section 4.4.3). In the digit example, each cluster represents a different digit or handwriting geometry. This is seen in the rhythm example as well. Clusters show 8th note pulse (B), 16th note pulse (E,L,N), triplet pulse (C) and even the presence of a slight 32nd note pulse (L). Similar to some digit clusters having different handwriting geometry (i.e., slanted) in the written digit example, different types of the same pulse division show up in the rhythm example. Clusters E, L, and N all show a 16th note pulse. However in the Beat Profile, E and L have tighter spikes than N, suggesting that they have a more consistent (metronomic) interpretation of the pulse. The broader hills suggest a more varied, wide pulse interpretation. There are also differences in the Mellin Transform and The Tempogram Ratios across all three, suggesting that the repetition of notes across beats within a bar vary differently (i.e., different interpretations of meter, backbeat, syncopation)

Chapter 5: Preliminary Study of Rhythm Features and Rhythmic Style

5.1 Overview

Humans identify with basic components of melody and rhythm in order to describe and differentiate songs. With these simple components, one can usually recognize higher level concepts such as the style and other expressive components of a piece of music. Previous work in the MIR field has studied both style and expression of a song as a whole, but few efforts focus on deconstructing and quantifying these individual components to discover the specific roles that each of them play. In this set of preliminary experiments, I explore a new feature representation designed to capture rhythmic elements in acoustic music signals and provide evidence surrounding its ability to distinguish rhythmic style. This proposed Rhythmic Style Histogram Feature (RSHF) is a probabilistic model of Inter-Onset-Intervals (IOI) quantized to Tatum positions between estimated beat locations across multiple frequency bands.

5.2 Rhythmic Feature Design

In order to represent rhythmic style, the RSHF is designed to probabilistically model inter-onset-intervals across multiple frequency bands. The signal flow of the RSHF construction is shown in Figure 5.1.

5.2.1 Deriving the RSHF

The percussive component of Harmonic Percussive Source Separation (Figure 5.1a) is first computed [13], and beats are estimated¹ from the percussive signal [33]. The percussive power spectrum is then quantized into twelve (least common multiple of duple 16ths and triple 8ths) equally spaced bins between successive beat locations (Figure 5.1b).

The Tatum aligned spectrum is processed using a coarse Mel-spaced filter bank. The output of each Mel-filter becomes an independent onset signal $X_f[t]$, where f is the filter channel and t is

¹<https://github.com/bmcfee/librosa>

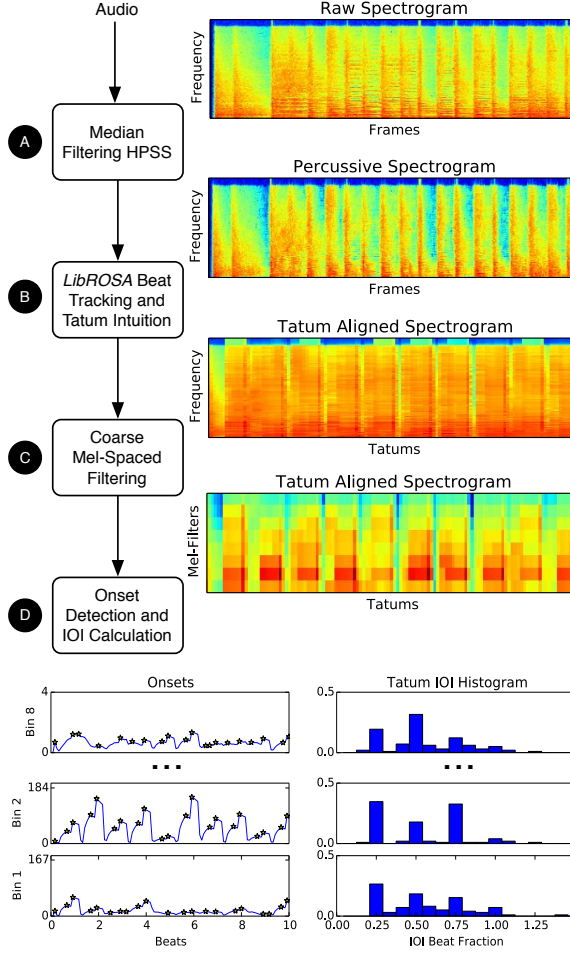


Figure 5.1: RSHF design and calculation.

the Tatum position (Figure 5.1c). The onset detection signals are then modified by subtracting the output of a moving average filter with a window length w and a tail multiplier m , yielding

$$Y_f[t] = X_f[t] - \frac{\sum_{k=t-mw}^{t+w} X_f[k]}{mw + w}. \quad (5.1)$$

Onsets are defined as the local maxima of the filtered signals $Y_f[t]$ within a window of length w . The calculation of the onset positions is similar to the method described in [22].

Using the Tatum-aligned onset positions, the IOIs for each Mel-frequency band are calculated (Figure 5.1d). The raw RSHF feature becomes a stacked set of histograms denoting the empirical probability of the IOI values for each Mel-frequency band. An example of this feature is shown in

Figure 5.2. In the lower frequency range, there is more probability mass in Tatums that are at $1/4$ and $3/4$ of the beat. This shows that both 16th notes and dotted 8th notes are common. In this example, it is representative of repetitive swing-like pattern in the bass drum composed of a dotted 8th note followed by a 16th note. In the higher frequency components there is high probability for IOI equal to $1/2$ of a beat. This means that there is likely a steady 8th note pulse, played possibly by a hi-hat or ride cymbal.

It is important to note that the RSHF is sensitive to the quality of the onset detection function and the beat tracker employed as part of the front-end system. Errors in the beat tracking may lead to improper binning in the metrical divisions defining the Tatum-aligned spectrogram in Figure 5.1C.

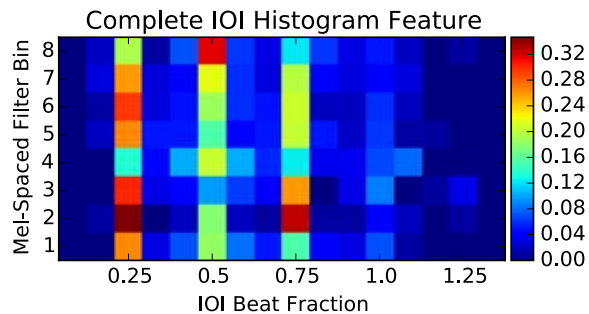


Figure 5.2: Rhythmic Style Histogram Feature.

5.3 Feature Salience Experiments

In order to test the RSHF, a set of both supervised and unsupervised machine learning experiments are performed. These experiments show that the feature has the potential to predict previously known intuitions about rhythm and style as well as reveal meaningful correlations learned directly from data that are sometimes difficult to quantify. It is important to note that while the following results are not state of the art in this stand-alone context, the RSHF is definitely salient, and has the potential to improve and inform other systems and tasks [58].

A set of classification tasks using the popular *Ballroom Dataset* [85] is performed. The dataset’s audio examples are 30 seconds in length, and are labeled with a specific ballroom dance style. This dataset is chosen because its labels apply directly to terms that reference quantifiable attributes of

the music, and not to cultural popularity (e.g. the pop genre). More information about this dataset can be found in Chapter 3.

5.3.1 Supervised Experiments

The goal of the first supervised experiment is to classify the *Ballroom Dataset* with respect to the given ballroom style label. This is performed using an *Support Vector Machine* (SVM) classifier with a *Radial Basis Function* (RBF) kernel. The dataset is split into 30% for test and 70% for training and the parameters of the model are fit using 10-fold cross validation of the training set. Results for style classification are presented in Table 5.1. This task has been performed by many others previously in [132, 45, 60] with classification results surpassing 90%. The goal of this work is not to solve this task specifically, but to show the RSHF’s salience in this domain. It was also shown that tempo alone is a good descriptor of styles on the Ballroom Dataset [45]. The beat-tracking algorithm inherently includes an estimate of tempo. By including that estimate along with the RSHF, classification is improved.

Feature	Accuracy Alone	Accuracy with Tempo
RSHF	0.562 ± 0.035	0.755 ± 0.017

Table 5.1: Accuracies in the style task for the raw RSHF and the best performing reductions.

In the second experiment a duple feel vs. triple feel discrimination is performed using the same experimental setup. By adding tempo, classification accuracy is improved once again. These results are shown in Table 5.2.

Feature	Accuracy Alone	Accuracy with Tempo
RSHF	0.831 ± 0.017	0.937 ± 0.033

Table 5.2: Accuracies in the duple vs. triple task for the raw RSHF and the best performing reductions

5.3.2 Unsupervised Experiments

In certain analysis tasks of expression and style, hard labels are not sufficient to describe a certain musical phenomena. It may be necessary for expression and style components to sit in a continuous space and employ unsupervised methods to find meaningful correlations and relationships in the

data. In order to test the RSHF’s effectiveness in this unsupervised domain, a set of simple k-means clustering experiments is performed ($k = 2$ and $k = 4$).

In order to explore the cluster space, the RSHF feature is shown along with the annotated ground truth labels for style and feel in Figure 5.3. The 96-dimensional feature space is visualized using t-distributed Stochastic Neighbor Embedding. t-SNE is a tool for visualizing high-dimensional data by preserving the distance between points in a lower-dimensional space. More on t-SNE can be found in [117] and in Chapter 4.4.2. Notice that while the different classes of style-labeled and feel-labeled data overlap, each occupies a unique area throughout the space.

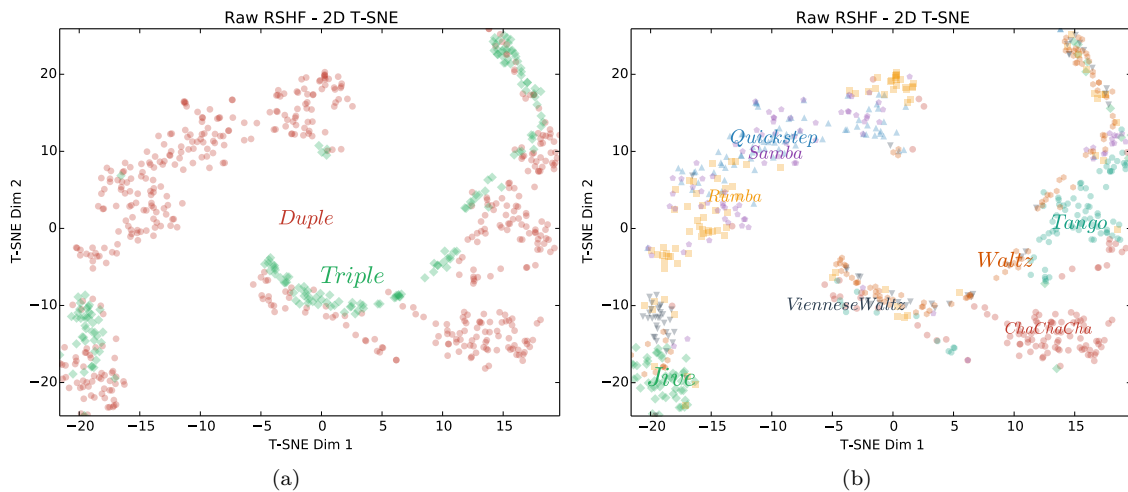


Figure 5.3: The projections of the raw RSHF feature into 2 dimensions for (a) duple/triple designation and (b) individual style classifications.

In order to capture these empirical observations organically, k-means clustering is performed on the RSHF. The results of the unsupervised clustering is shown in Figure 5.4. The percentages of the original style labels contained within a specific cluster are shown for all clusters. In the $k = 2$ clustering, the somewhat obvious separation of the convex and concave arc structures in the data occurs. The complex and more dense styles cluster apart from the simpler straight-forward styles. As the number of clusters increases to $k = 4$, each of the styles start to separate. Jive and Viennese Waltz separate from the other dense rhythms, which can be attributed to the triple and compound meters of Waltz and Jive versus the simple meters of Quickstep, Rumba, and Samba. The more straight-forward rhythmic styles also start to split. Tango and ChaChaCha still group tightly due

to the fact that their rhythms are composed of very similar structures with a heavy emphasis on the beat. Both Waltz styles also start to separate into multiple groups. Some of the Viennese Waltz and Waltz rhythms group together in their own cluster, while others tend to group with other more similarly dense and less dense styles.

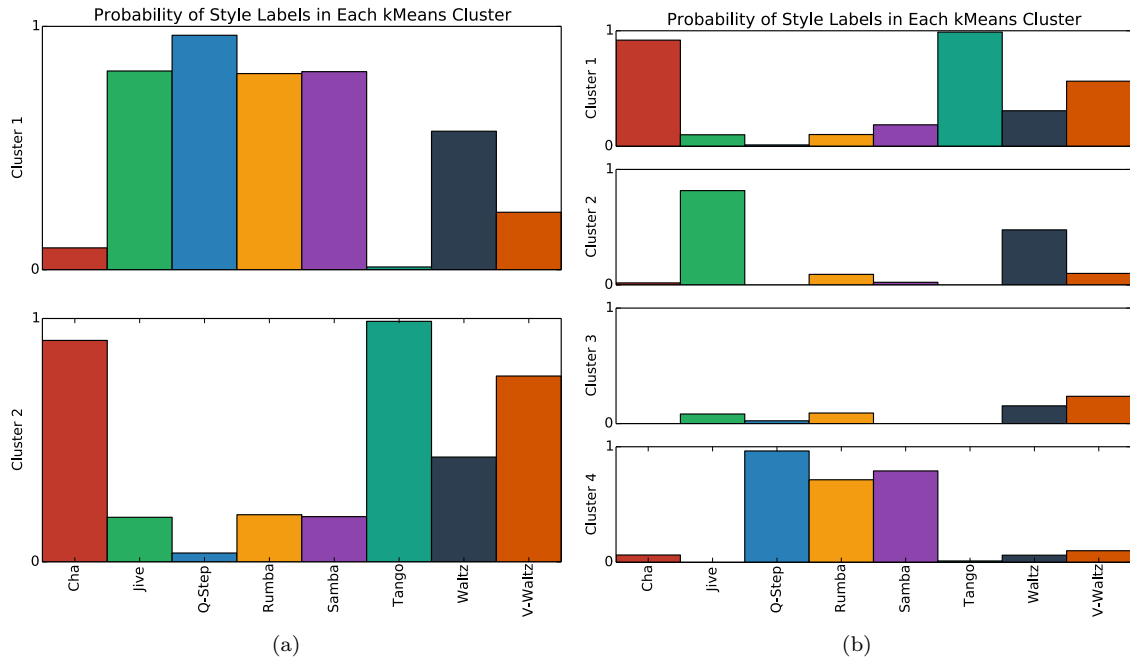


Figure 5.4: The percentage of each style label in each k-means cluster for (a) $k = 2$ and (b) $k = 4$.

5.4 Conclusions

The Rhythmic Style Histogram Feature captures musically informative patterns in a compact form, leveraging Tatum-aligned IOI interval histograms over multiple Mel-spaced frequency bands. Through a set of supervised and unsupervised machine learning experiments, I showed that the RSHF is informative for the task of rhythmic style classification and analysis. The RSHF has many future implications in the domain of rhythmic style analysis as a whole as well. Because the RSHF feature is a probabilistic model and intuitively maintains rhythmic components, it can be informative in the generation of rhythmic styles. This generation is also tunable among different frequency bands, allowing it to capture and synthesize rhythms that have contrasting low pitched and high pitched components, such as a kick drum vs. a snare drum.

Later chapters in this thesis are modeled on work presented in each section of this chapter. While this chapter's work is no-longer novel in and of itself, it serves as preliminary overview that motivates and touches on much of the work in the chapters to come.

Chapter 6: Rhythm Acoustic Features

6.1 Overview

In this chapter, I outline a set of newly developed acoustic features that aim capture rhythmic attributes automatically in music audio signals. These features are designed to be tempo-invariant, deterministic rhythmic descriptors that represent specific elements of the meter and rhythmic feel (i.e., swing). The descriptors are first evaluated and benchmarked to previous work with the widely-used meter and style classification tasks using the *Ballroom Dataset* and the *GTZAN Rhythm Dataset* rhythm annotations. Many of the features involve estimates of tempo for normalization, so the effects of these tempo estimates will be outlined as well.

The fundamental components of rhythm are metrical structure, tempo, and event timing [133]. There is a large body of prior work that attempts to estimate these components [33, 34, 85, 44], but in extracting only beats, tempo, and meter much of the rhythmic subtlety and feel is discarded. A mid-level representation known as the *accent signal* [21], which measures the general presence of musical events, is better suited to represent this rhythmic subtlety. However, the tempo, beat, and meter estimates are still beneficial, as they can provide important temporal context to rhythmic patterns derived from the accent signal. For example, the frequencies of periodicity in an accent signal can be used to infer beats per minute, and when normalized by an estimate of tempo, directly relate to musical note durations [53]. The accent signal can also be quantized and viewed in the context of beats or measures in order to capture discrete instances of rhythm patterns [132, 119]. In other work, Holzapfel introduced the *Mellin Scale Transform* as both a tempo-invariant and tempo-independent method for describing rhythmic similarity. Unlike previous methods, the transform achieves tempo-invariance by design rather than normalizing by a tempo estimate [60].

Most of the previous work in capturing rhythm has relied on evaluation through the classification of a generalized musical style or genre, while simultaneously focusing on specific aspects of rhythm in the feature design. Evaluation is usually performed with the *Ballroom Dataset* of dance styles [134],

which more precisely represents rhythm than a dataset that is labeled with basic genre. However, this remains a high-level approach with little regard to the meaning of the specific aspects of rhythm inherent in the music. As a result, researchers have started to overfit and exploit phenomena of the dataset rather than capture the attributes that relate more generally to music [135, 134]. Furthermore, work by Flexer demonstrates that general music similarity requires the context of many different factors outside of just rhythm [136]. While it is possible to argue that certain features may be capturing components of rhythm, the contextual complexities in the style labels make it difficult to infer meaning. This motivates the need for a more strict and concrete evaluation of rhythm features and their contributions to specific rhythmic components. This component-level analysis will be explained further in Chapter 7.

6.2 Designing Features for Rhythm

Each of the rhythm features described in Sections 6.2.3, 6.2.4, and 6.2.5 are derived using a combination of the *accent signal*, the *tempogram*, estimates of the tempo, and estimates of beat locations. Section 6.2.1 will provide a brief overview each of these processes used.

6.2.1 Rhythm Signal Analysis

Accent Signal (ODF)

In order to capture aspects of each rhythm label, a set of rhythm-specific features was implemented. The features are based on an *accent signal*, which measures the change of a music audio signal over time. High points of change denote the presence of a new musical event. The *accent signal* used is a variant of the *SuperFlux* algorithm [21] and is the half-wave rectified ($H(X) = \frac{X+|X|}{2}$) sum of frequency bands of a frequency smoothed (Eq. 6.1) *Constant Q Filter Bank Transform* (CQT) X_{cqt} of an audio signal (Eq. 6.2).

$$X_{cqt}^{max}[n, m] = \max(X_{cqt}[n, m - 1 : m + 1]) \quad (6.1)$$

$$SF[n] = \sum_{m=1}^{m=M} H(X_{cqt}[n, m] - X_{cqt}^{max}[n - \mu, m]) \quad (6.2)$$

Tempo Estimation

From the accent signal, an estimate of tempo is found. This was achieved through a hybrid of the standard inter-onset-interval (IOI) and autocorrelation function (ACF) methods that are widely used. The IOI method employs SuperFlux onset detection to create a histogram of inter-onset-distances. The ACF method is the autocorrelation of the accent signal. *Periodicity salience* is then found by summing across k harmonics and sub-harmonics of the ACF lag or the IOI Histogram distance (Eq. 6.3). A tempo is estimated using the maximum peak in the fusion tempogram.

$$S_{\text{ACF}}[l] = \sum_{k=1}^K \text{ACF}[kl] + \sum_{k=2}^K \text{ACF}\left[\frac{1}{k}l\right] \quad (6.3)$$

Periodicity salience is then converted to a *tempogram* by transforming the onset distance or lag l in time to a tempo $\tau = \frac{60}{l}$ bpm. A fusion tempogram $F_{TG}(\tau)$ can be found by multiplying the individual tempograms (Eq. 6.4) [53].

$$F_{TG}(\tau) = S_{\text{ACF}}(\tau) \odot S_{\text{IOI}}(\tau) \quad (6.4)$$

An evaluation using a small test set is shown in Table 6.1. I compare my method with two other widely available methods from *libRosa* and the *Echo Nest* API. The first metric determines the accuracy of the initial tempo estimate ($\pm 4\%$ of the annotated tempo). The second metric allowed for the algorithms to supply two estimates. The first of the two estimates is the initial peak from F_{TG} . The second is the maximum peak at relevant multiples of the first ($\frac{1}{3}\times$, $\frac{1}{2}\times$, $2\times$, $3\times$). The third metric chooses 5 candidate estimates of $\frac{1}{3}\times$, $\frac{1}{2}\times$, $1\times$, $2\times$, $3\times$ the initial tempo estimate to determine accuracy. It is labeled as correct if any of these 5 estimates is within $\pm 4\%$ of the annotated tempo.

Metric	F_{TG} (Eq. 6.4)	Echo Nest API (Spotify)	libROSA
1. Initial Guess	0.710	0.760	0.649
2. Two-Guesses	0.868	N/A	N/A
3. Within Multiple	0.876	0.880	0.730

Table 6.1: Small-scale tempo estimator evaluation.

Many of the rhythm features and processes outlined in this chapter rely on an estimate of tempo. These include beat tracking, the beat profile, and the tempogram ratio. For each of the following

methods, I will treat the estimated tempo as ground truth. In Section 6.3 I will perform a deeper evaluation of the effects of errors in tempo estimates.

Beat Tracking

Using this accent signal and the tempo estimate, beat tracking is performed using the dynamic programming method [33]. This method was chosen for its ease of implementation, scalability, deterministic nature, and consistency of beat position estimation. To compare this method with others, an evaluation of the beat tracking methods is performed using the SMC dataset [2]. The top 5 methods, including those presented in work by Holzapfel [2], libRosa, the *Echo Nest* API, and one presented in this thesis are shown in Table 6.2. The scores appear low but this dataset was designed specifically to be difficult to track [2] Across both metrics (AMLt, F-Measure), the presented method performs just under the best performing one. It is also important to note that some of the methods, such as the Böck method, are much more complex and use techniques such as Neural Networks that take a lot of time to train. This limits their scalability when applied to large datasets. The last column contains the average of both AMLt and F-measure, showing the best performing algorithm overall.

Algorithm	AMLt	F-Measure	Combined	Rank (out of 19)
Klapuri	0.339	0.362	0.351	1
My Method	0.330	0.370	0.350	2
Degara	0.334	0.346	0.340	3
Böck	0.261	0.401	0.331	4
libROSA	0.299	0.361	0.330	5
...
Echo Nest (Spotify)	0.295	0.261	0.278	10

Table 6.2: Beat Tracking Evaluation

6.2.2 Rhythm Examples

In order to visualize each feature, a set of consistent style examples of Samba, Tango, and Jive from the ballroom dataset will be used. A canonical representation of these rhythms for drum set, obtained from Tommy Igoe’s *Groove Essentials*, is shown in Figure 6.1 [137]. Samba is 16th note based with clave patterns present in the mid voices (snare drum). Tango is quarter-note based with a tied 8th note pickup leading into each successive measure. Jive is a fast Rock and Roll pattern

with a driving swing. Triplets are notated to more accurately represent the swing expression. Each feature explanation in the following sections will refer to these rhythmic examples.

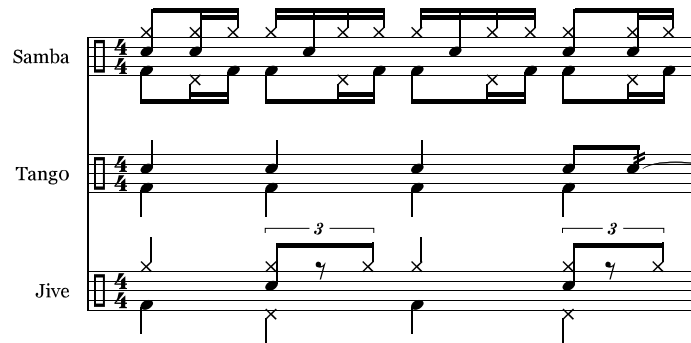


Figure 6.1: These patterns define the Samba, Tango, and Jive rhythmic styles for drum set.

6.2.3 Beat Profile

The *Beat Profile* is *Tatum*-level feature that captures a compact snapshot of the accent signal within beats. This is similar to the feature by Dixon [132], but it is simpler, deterministic, and free of human intervention. The accent signal between consecutive beat estimates is quantized in time to 36 beat subdivisions. The *Beat Profile* features are statistics of each of those 36 bins over all beats. The *Beat Profile Distribution* feature (BPDIST) is comprised of the mean of each beat profile bin (BPMEAN) and constrained such that that the collection of bins must sum to one. A set of BPMEAN features is shown in Figure 6.2.

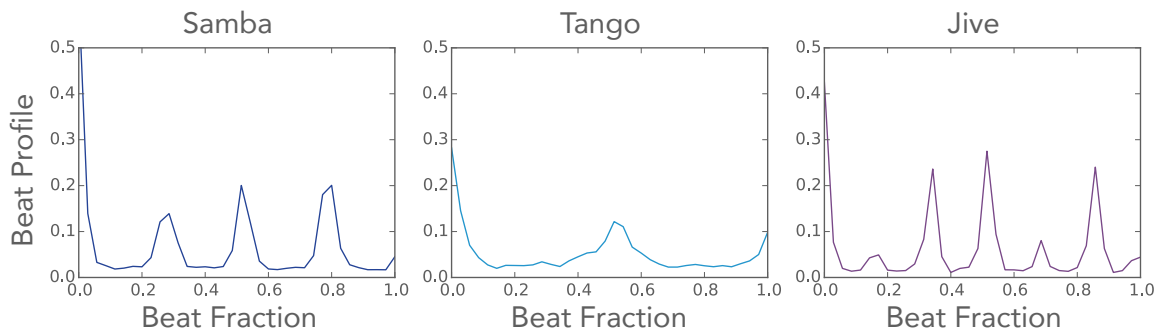


Figure 6.2: Examples of the BPMEAN Feature are shown. On the X axis, 0.0 denotes the beat, 0.5 denotes the 8th note, and 1.0 is the lead-in to the next beat.

The *Beat Profile* clearly shows the Samba rhythm’s 16th note pulse. It also shows heavier emphasis at 0.0, 0.5 and 0.75, corresponding to the beat and the 3rd/4th 16th note partials (counted as 1 . & a). This emphasis appears because these partials occur more regularly, and are shared on every beat of the Samba rhythm. The second 16th note partial at 0.25 (the ‘e’) only occurs some of the time, resulting from clave-like rhythms, and therefore has less weight. The Tango profile contains a strong down beat emphasis and a wide, low intensity 8th note. This is because Tango has heavy, staccato beats and a slurred (elongated) up-beat (8th note) that only occurs once per measure. Jive has a sharp, double-time triple feel with emphasis on the beats (0.0) and up-beats (0.5). It also contains emphasis on the 3rd triplet partial of the 16th note triplet (0.33 and 0.83). This is due to Jive’s fast swing and dance rhythms.

6.2.4 Tempogram Ratio

The *Tempogram Ratio* feature (TGR) uses the tempo estimate, similar to work by Peeters [53], to remove the tempo dependence in the tempogram. By normalizing the tempo axis of the tempogram by the tempo estimate, a fractional relationship to the tempo is gained. These fractional relationships are shown in Figure 6.3.














Note Name	16th Note	16th Note Dotted	8th Note Triplet	8th Note	8th Note Dotted	Quarter Note Triplet	Quarter Note	Quarter Note Dotted	Half Note Triplet	Half Note	Half Note Dotted	Whole Note Triplet	Whole Note
Note Symbol													
Tempo Multiple	4	$\frac{8}{3}$	3	2	$\frac{4}{3}$	$\frac{3}{2}$	1	$\frac{2}{3}$	$\frac{3}{4}$	$\frac{1}{2}$	$\frac{1}{3}$	$\frac{3}{8}$	$\frac{1}{4}$

Figure 6.3: Note values and multiples of each dimension of the Tempogram Ratio feature.

A compact, tempo-invariant feature is created by capturing the weights of the tempogram at musically related ratios relative to the tempo estimate. Examples of the tempogram and tempogram ratio features are shown in Figure 6.4.

time T_{up} (Eq. 6.7).

$$r'(l) = \frac{r(l) - \min\{r\}}{\max\{r\} - \min\{r\}} \text{ where, } r(l) = \sum_n x[n]\bar{x}[n-l] \quad (6.5)$$

$$R(c) = \frac{\sum_{k=1}^{\infty} [r'(kT_s - T_s) - r'(kT_s)](kT_s)^{1/2-jc}}{(1/2-jc)\sqrt{2\pi}} \quad (6.6)$$

$$\Delta c = \frac{\pi}{\ln \frac{T_{up} + T_s}{T_s}} \quad (6.7)$$

The transform is calculated on autocorrelations of 8s widows with a 4s overlap. The song is summarized by the mean over time. An example of the scale transform feature (MST) is shown in Figure 6.5. In order to remove correlations of harmonics in the transform, the discrete cosine transform (DCT) is computed. This is similar in motivation to MFCCs. Median removal (by subtracting the local median) and half-wave rectifying the DCT creates a new feature that emphasizes periodicities by performing a rough peak-picking and filtering. The energy component of the DCT is also removed (0th component, similar to MFCCs). This new feature (MST_DCT) is then normalized to sum to one. More about the Mellin scale transform can be found in [60].

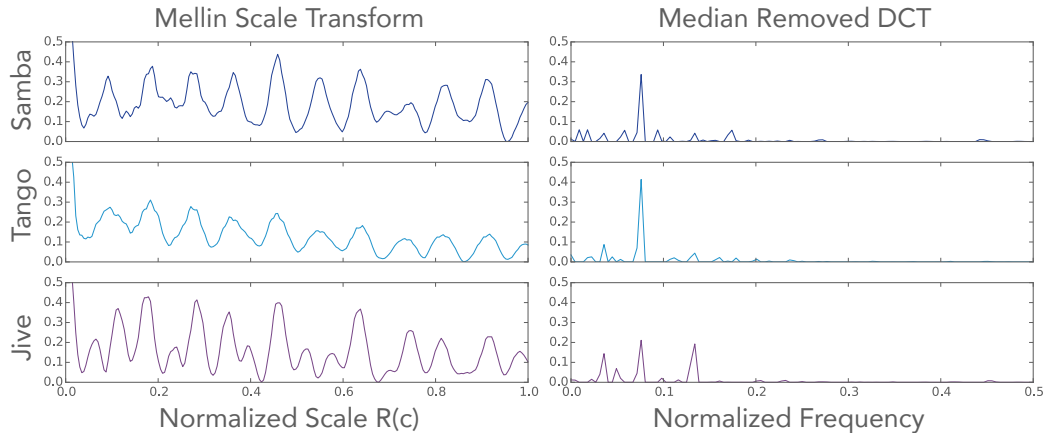


Figure 6.5: Examples of the MST and MST_DCT Feature.

The *Mellin Scale Transform* feature is slightly more complex to interpret. The goal of the transform is to measure consistency of signals among multiple time scales. Another interpretation of

the MST is the *Fourier Transform* of a time-domain signal that is sampled exponentially [60]. This is explained further in Appendix A. Because it is computed on the autocorrelation of an accent signal for this rhythm feature, it captures rhythmic repetition at different exponentially related time scales (Tatum, beat, bar, phrase) in a more linearly harmonic manner. This can be seen as a time-domain rhythm analog to the CQT creating linear mappings of logarithmic pitch relationships.

In Figure 6.5, Samba and Tango show similar transforms because they are similarly rhythmically stable on multiple time scales. They each have a consistent Tatum pulse, beat pulse, and meter pulse. There are consistent 16th notes for Samba, consistent quarter notes for Tango, and consistent bar repetitions for both. The transform for Jive is slightly different. This is due to inconsistent repetitions on the Tatum-level and beat-level from the swing pattern, creating a more complex harmonic structure in the transform. More information and examples of this transform can be found in Appendix A.

6.2.6 Multi-band Representations

Each of the rhythm features described in sections 6.2.3 and 6.2.4 rely on a global estimates of beats, tempo and an accent signal. These features can be extended to multiple-band versions by using accent signals that are constrained to be within a set of specific sub-bands of the CQT from which it is computed. Using separate accent signals, the rhythmic features can relate to the different compositional functions of instruments that occupy different frequency ranges. In this work, the following ranges were used:

1. Bass Frequency Band: $(A_0, A_3] \rightarrow (27.5\text{Hz}, 220\text{Hz}]$
2. Treble Frequency Band: $(A_3, A_6] \rightarrow (220\text{Hz}, 1.76\text{kHz}]$
3. High Frequency Band: $(A_6, A_9] \rightarrow (1.76\text{kHz}, 14.08\text{kHz}]$

6.2.7 Rhythmic Feature Evaluation

In order to evaluate and compare the new features, a set of general Music-IR classification tasks was performed on the *Ballroom Dataset* (from Chapter 3: 8 ballroom dance styles, 698 instances, 523 instances with duple meter and 175 instances with triple meter). The rhythm features were

used individually and in various aggregations with each feature dimension normalized from 0 to 1. Block-based Mel-Frequency Cepstral Coefficients (MFCC) are also used for comparison. Means and covariances of MFCCs are calculated across overlapping 6-second blocks. These block-covariances are further summarized over the piece by calculating their means and variances [138]. A simple logistic regression classifier was fit for 10 trials with a randomly shuffled 70:30 train:test split for each trial. A subset of these results is shown in Table 6.3. The tempogram (TG) feature shows state

Tempo-Invariant Feature	Dim.	Duple vs. Triple	Ballroom Style
BPDIST	36	0.849 ± 0.031	0.776 ± 0.035
BPDIST_M (multiband)	108	0.873 ± 0.016	0.794 ± 0.019
TGR	13	0.883 ± 0.024	0.747 ± 0.030
TGR_M (multiband)	39	0.952 ± 0.007	0.817 ± 0.022
MST	230	0.956 ± 0.011	0.868 ± 0.010
MST_DCT	230	0.936 ± 0.014	0.829 ± 0.018
MST BPDIST_M TGR_M	377	0.974 ± 0.010	0.917 ± 0.018
MST_DCT BPDIST_M TGR_M	377	0.959 ± 0.015	0.884 ± 0.019
MFCC	460	0.877 ± 0.016	0.511 ± 0.027
MFCC MST BPDIST_M TGR_M	837	0.942 ± 0.018	0.743 ± 0.020
MFCC MST_DCT BPDIST_M TGR_M	837	0.925 ± 0.017	0.707 ± 0.035
TG (tempo-variant)	500	0.962 ± 0.010	0.843 ± 0.011

Table 6.3: Ballroom dance style classification tasks results.

of the art performance on the Ballroom Dataset (as of 2014), which is evidence for the well-known class tempo-dependence [134]. Other features that are tempo-invariant perform similarly without exploiting the known class tempo-dependence of this dataset. Evidence of tempo-invariance vs. tempo-variance in classification is shown by the confusion matrices in Figure 6.6.

The tempogram (TG) confuses Jive (160-180bpm) with Waltz (78-98bpm), even though they are very different stylistically. However, it cannot easily differentiate the exact 2:1 tempo ratio because both styles have energy at similar tempo multiples. Rumba (90-110bpm) and Jive show a similar error relationship. Conversely, MST confuses Samba (96-104bpm) with Tango (120-140bpm) and ChaChaCha (116-128bpm), which do not overlap with Samba’s tempo range. However, these three styles contain similarity in their rhythmic self-repetition, which is something the MST feature is designed to capture. Furthermore, this lack of overlap makes Samba much easier to distinguish for the tempogram feature. This suggests that the rhythm features are representing something about the rhythmic characteristics, and not relying on tempo for discrimination.

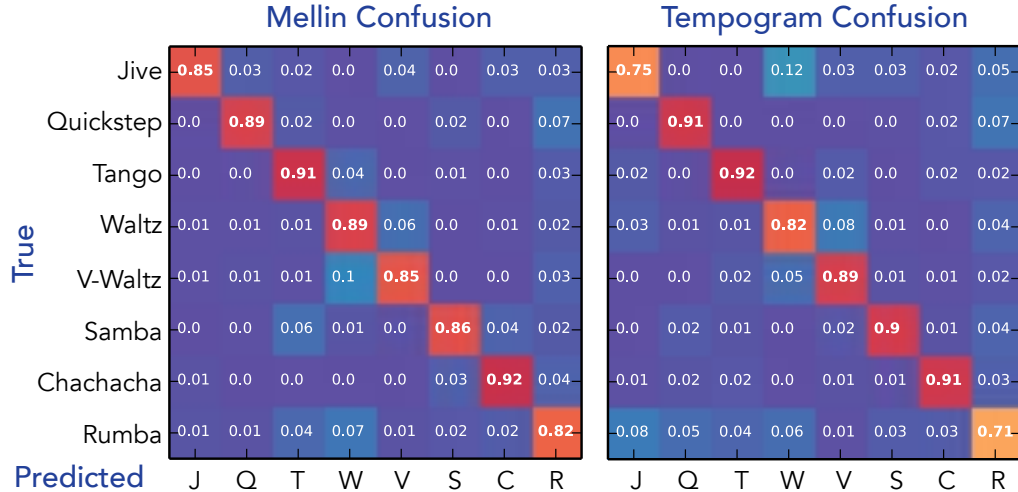


Figure 6.6: Ballroom Dataset confusion matrices of the Mellin Transform and Tempogram features

6.3 Analysis of Rhythmic Attributes and Tempo Estimation

In this section, I perform a preliminary analysis of musical attribute prediction using the *Ballroom Dataset* and the *GTZAN Rhythm Dataset*. Because many of the presented rhythm features rely on estimates of tempo, I also outline the effects of errors in that estimation on rhythmic attribute prediction models.

The effects of tempo are evaluated by computing the rhythm features using the ground truth tempo (τ_{gt}), a uniformly randomized octave error ($\frac{1}{2}\times$, $1\times$, $2\times$) of the ground truth tempo (τ_ϵ), and the estimated tempo (τ_{est}) from Section 6.2.1. Table 6.4 shows the results of evaluating each of these tempo parameterizations compared to the ground truth tempos on the *Ballroom Dataset* and the *GTZAN Rhythm Dataset* (Genre). Accuracy ‘A’ refers to $\pm 4\%$ of GT. Accuracy ‘B’ refers to $\pm 4\%$ of $\frac{1}{2}\times$, $1\times$, or $2\times$ of GT. These are similar to Accuracies 1 and 3 from Section 6.2.1 respectively.

Metric	Ballroom			Genre		
	GT Tempo	GT + Error	Estimated	GT Tempo	GT + Error	Estimated
Accuracy A	1.000	0.307	0.623	1.000	0.319	0.647
Accuracy B	1.000	1.000	0.924	1.000	1.000	0.913

Table 6.4: Tempo estimation results on the Ballroom and GTZAN Rhythm (Genre) Datasets.

With an understanding of how well tempo is being described within the rhythm feature computation, a set of attribute prediction experiments is performed using style, genre, meter, and feel labels from the *Ballroom Dataset* and *GTZAN Rhythm Dataset*. The first set of experiments predicts the class of an attribute (style, genre, meter) in a multi-class discrimination problem. This acts as an analog to prior work using this data. A second set of experiments breaks up each of rhythm attribute classes and performs an independent binary discrimination, which is similar to work in Chapter 7. Each model employs *Logistic Regression* trained on a randomly shuffled 70%:30% split across 10 trials with no artists shared between training and testing. The results in Tables 6.5, 6.6, and 6.7 show the mean accuracies across the 10 trials. The features used in these experiments is the aggregation of the *Mellin Scale Transform DCT*, the multi-band *Beat Profile*, and the multi-band *Tempogram Ratio* features. These were chosen to evaluate the same aggregation used for the attribute and genre models in Chapters 7 and 8.

The results in Table 6.5 show the effects of tempo estimation on meter and style classification on the *Ballroom Dataset*. Overall, tempo estimation errors have little effect on classifying the meter and style. This relatively low drop in error is partially due to the use of tempo-agnostic features (*Mellin Scale Transform*) in combination with the estimation-based tempo-invariant ones. In the 8-class discrimination problem, the model using the estimated tempo drops 0.023 compared to the one that uses the ground-truth tempo in feature computation. When classifying the meter, the drop of 0.002 is negligible. When forcing an octave tempo error, the decrease in performance compared to the ground-truth model is slightly more. However, it is a proportionally small decrease when considering that this model only estimates the correct tempo ~30% of the time.

Classification Attributes	GT Tempo (τ_{gt})	GT + Error (τ_{ϵ})	Estimated (τ_{est})	Estimated Diff. ($\tau_{est} - \tau_{gt}$)	Error Diff. ($\tau_{\epsilon} - \tau_{gt}$)
Dance Style (Mean Acc.)	0.939	0.885	0.915	-0.023	-0.053
Triple Meter (Mean AUC)	0.993	0.976	0.991	-0.002	-0.018

Table 6.5: Meter and style classification on the Ballroom Dataset.

In order to further evaluate the effects of tempo, meter and genre classification was performed on the *GTZAN Rhythm Dataset*. The mean accuracy across all trials of multi-class classification of meter and genre is shown in Table 6.6. Once again, incorrect estimation of tempo produces negligible

decreases. The largest drop occurs when forcing errors into the model. Genre classification drops $\sim 10\%$. This may be due to the context of the errors made. In some genres such as country, cut-time is common, meaning there is a different feeling of beat or pulse relative to the written tempo. If errors are made consistently in certain contexts, they are less detrimental then when applied artificially without context.

Multi-class Attributes (Mean Acc.)	GT Tempo (τ_{gt})	GT + Error (τ_{ϵ})	Estimated (τ_{est})	Estimated Diff. ($\tau_{est} - \tau_{gt}$)	Error Diff. ($\tau_{\epsilon} - \tau_{gt}$)
Meter (4-class)	0.890	0.872	0.882	-0.008	-0.017
Genre (10 class)	0.515	0.409	0.532	-0.017	-0.106

Table 6.6: Mean accuracy of meter and genre multi-class classification on the GTZAN Rhythm Dataset.

Later work in Chapters 7 and 8 rely on the binary prediction of attributes independently. In order to perform a similar evaluation, each of the meter and feel attributes from the *GTZAN Rhythm Dataset* are separated into binary attribute prediction tasks. The results (AUC) of predicting the presence of these attributes and the effects of tempo error on the models is shown in Table 6.7. In 4 out of 6 tasks, the estimated tempo produced less error than forcing octave errors. In these four tasks, there was a drop of only $\sim 5\%$. While there is a greater drop due to the effects of tempo estimation for Triplet Feel and Compound Duple Meter, they are still performing well above random, so information about each of the attributes is still captured in the features. This also provides some evidence suggesting from where the error comes in the multi-class meter discrimination in Table 6.6.

Binary Attributes (Mean AUC)	GT Tempo (τ_{gt})	GT + Error (τ_{ϵ})	Estimated (τ_{est})	Estimated Diff. ($\tau_{est} - \tau_{gt}$)	Error Diff. ($\tau_{\epsilon} - \tau_{gt}$)
Triple Meter	0.796	0.791	0.839	0.043	-0.005
Comp.-Duple Meter	0.917	0.925	0.729	-0.188	0.008
Mixed Meter	0.467	0.531	0.519	0.053	0.064
Duple Meter	0.824	0.725	0.760	-0.063	-0.099
Triplet Feel	0.921	0.848	0.808	-0.113	-0.073
Swing Feel	0.964	0.913	0.918	-0.046	-0.050

Table 6.7: Mean AUC of meter and feel attribute prediction on the GTZAN Rhythm Dataset.

A more in-depth analysis of attribute predictions using the *Ballroom Dataset* and *GTZAN Rhythm Dataset* is found in Appendix D. The appendix also evaluates attribute prediction with each rhythm feature individually and in various aggregations.

Chapter 7: Learning Rhythmic Components

Previous work has studied the general recognition of rhythmic styles in music audio signals, but few efforts have focused on the deconstruction and quantification of the foundational components of global rhythmic structures. The work in this chapter focuses on modeling rhythm-related attributes of meter and “feel” (e.g., “swing”) in music using the targeted acoustic features from Chapter 6. Each of the models is evaluated using more than one million expertly-labeled audio examples from the *Pandora*[®] *Music Genome Project*[®] (*MGP*).

Most of the previous work in capturing rhythm has relied on evaluation through the classification of a generalized musical style or genre, while simultaneously focusing on specific aspects of rhythm in the feature design. Rhythm tasks, as in Chapter 6, are sometimes evaluated on the *Ballroom Dataset* [134], which more precisely represents rhythm than a dataset that is labeled with basic genre. However, this remains a high-level approach void of targeted learning of specific rhythmic constructs. As a result, researchers have started to overfit and exploit phenomena of the dataset rather than capture the attributes that relate more generally to music [135, 134]. Furthermore, work by Flexer demonstrates that general music similarity requires the context of many different factors outside of just rhythm [136]. While it is possible to argue that certain features may be capturing components of rhythm, the contextual complexities in the style labels make it difficult to infer meaning. This motivates the need for a more strict and concrete evaluation of rhythm features and their contributions to specific rhythmic components.

7.1 Approach

In this section, I outline the set of approaches used to model rhythmic attributes from the *Music Genome Project*.

7.1.1 Rhythmic Attributes of the Music Genome Project

In this work, I seek to capture rhythmic attributes individually and automatically in music audio signals. Using the rhythm descriptors outlined in Chapter 6, a set of machine-learning models is trained to learn the presence of the meter and rhythmic feel components individually across more than one million audio examples.

The targeted attributes are compositional constructs, such as the meter, or well-defined components of the musical feel, such as the presence of swing. Namely I focus on the following 9 rhythmic attributes:

- **Meter:** Cut-Time, Triple, Compound-Duple, Odd
- **Tatum (micro) Feel:** Swing, Shuffle
- **Meter (macro) Feel:** Syncopation, Back-Beat Strength, Danceability

Previous work has looked at identifying musical meter. However, emphasis was placed on distinguishing duple versus triple in a more general sense rather than identifying the true meter, which has an important function in the context of rhythmic style. Because focus is placed on meter differentiation, *cut-time*, *triple*, *compound-duple*, and *odd* meters are targeted. The widely shared meter of *simple-duple* ($\frac{2}{4}$, $\frac{4}{4}$) is ignored. Rhythmic feel has also been studied, but mostly in the context of similarity. Individual components of the rhythmic feel are important in defining style. They are easily recognizable to a listener, but are sometimes difficult to quantify. In this work I seek to define and capture the the qualities of *swing*, *shuffle*, *syncopation*, *back-beat strength*, and *danceability*. The rhythmic component labels were defined and collected by musical experts on a corpus of over one million audio examples from the *Pandora[®] Music Genome Project[®](MGP)*. The labels were collected over a period of nearly 15 years and great care was placed in defining them and analyzing each song with a consistent set of criteria. More information can be found in Chapter 3.

7.1.2 Machine Learning Models

In order to learn the rhythmic attribute labels from audio features, a set of scalable models was employed. More information about each can be found in Chapter 4. They include linear models and tree ensembles, namely:

Linear

- *Logistic Regression* (binary attributes)
- *Linear Regression* (continuous attributes)

Trees

- *Gradient Boosted Trees* (GBT)
- *Random Forests* (RF)
- *Gradient Boosted Tree* hybrid models (GBT-H)
- *Random Forest* hybrid models (RF-H)

7.2 Predicting Rhythmic Attributes: Linear Models

7.2.1 Experiments

In order to predict the rhythmic attributes from Section 7.1.1, stochastic gradient descent (SGD) was formulated for classification of the binary labels (log loss, logistic regression) and regression of continuous labels (least-squares loss, linear regression). The learning rate was tuned adaptively. The training data was separated on a randomly shuffled 70:30 train:test split with no shared artists between training and testing. Due to the size of the dataset, a single trial for each attribute is both tractable and sufficient. More on SGD can be found in [139]. Cut-time, triple, compound-duple, and odd meters along with the presence of swing, shuffle, and heavy syncopation are all binary attributes and are therefore formulated as classification tasks. Danceability and back-beat strength are continuous ratings and are formulated as regression tasks.

7.2.2 Results

The classification and regression results for each of the rhythm attributes are shown in Table 7.1. The binary classification tasks are evaluated using the area under the receiver operating characteristic curve (AUC). The regression results are evaluated with the R^2 metric.

Features	AUC		Comp.					R^2	
	Cut	Triple	Duple	Odd	Swing	Shuf.	Sync.	Dance	Back-Beat
BPDIST	0.792	0.753	0.733	0.698	0.845	0.875	0.724	0.317	0.136
BPDIST_M (<i>B</i>)	0.864	0.807	0.772	0.756	0.871	0.886	0.745	0.412	0.301
TGR	0.645	0.759	0.804	0.728	0.795	0.840	0.658	0.317	0.136
TGR_M (<i>T</i>)	0.801	0.808	0.859	0.754	0.811	0.842	0.666	0.350	0.199
MELLIN (<i>S</i>)	0.810	0.916	0.945	0.840	0.868	0.914	0.743	0.452	0.269
MELLIN_D (<i>D</i>)	0.862	0.910	0.933	0.848	0.876	0.915	0.761	0.513	0.425
(<i>S</i>) (<i>B</i>) (<i>T</i>)	0.890	0.926	0.949	0.849	0.897	0.921	0.769	0.506	0.396
(<i>D</i>) (<i>B</i>) (<i>T</i>)	0.899	0.924	0.946	0.862	0.902	0.920	0.770	0.515	0.393
MFCC (<i>M</i>)	0.802	0.795	0.667	0.741	0.784	0.723	0.707	0.450	0.38
(<i>M</i>) (<i>S</i>) (<i>B</i>) (<i>T</i>)	0.899	0.920	0.942	0.843	0.897	0.922	0.780	0.537	0.464
(<i>M</i>) (<i>D</i>) (<i>B</i>) (<i>T</i>)	0.904	0.920	0.942	0.861	0.903	0.920	0.779	0.532	0.468

Table 7.1: The results for rhythm construct learning are shown. Both the AUC and R^2 metrics have a maximum value of 1.0 and lower bounds of 0.5 when predicting a random class (AUC) and 0.0 when predicting the mean of the test labels (R^2).

The results show that the rhythm-motivated features are best able to capture the rhythm attributes when compared to the timbre-motivated features. When both are used in combination, little improvement is gained. Timbre features alone can differentiate certain rhythmic attributes fairly well in some cases. For example, the cut-time meter is very common in the “country” and “bob jazz” genres and MFCC’s are possibly picking up on the genre’s similarly specific instrumentation rather than the rhythmic components. In all cases, the rhythm features are better than timbre alone, offering further proof that the rhythm features are learning something about the attributes they are targeting rather than their generalized correlation to a musical style.

Furthermore, it is seen among rhythm features that each have selected strengths. They tend to represent Tatum-level versus Meter-level information and single-band (global) versus multiple-band (range-specific) information. When considering Tatum-level versus Meter-level patterns, swing, shuffle, and syncopation are better represented by the beat profile features than the tempogram ratio features. This is because these rhythm attributes are defined on a local beat level, and the patterns within the beats (Tatums) have a specifically associated feel. Compound-duple and odd meter are

better defined by tempogram ratios, which suggests that they have patterns that cannot be captured within a single beat. It is also seen that the Mellin representations are effective across beat-level and measure-level attributes, suggesting that they are able to capture both.

When looking at single-band versus multi-band features, the rhythm components and associated features that capture interplay between multiple instrument ranges are highlighted. Meter, syncopation, danceability and back-beats all rely on the emphasis of specific points in a measure. In the context of a performance, the use of multiple instruments may be used to highlight these differences in emphasis, which is captured in multi-band representations. Attributes that rely on global feel and timing, such as a swing or shuffle, are not aided by the multi-band representations.

7.3 Predicting Rhythmic Attributes: Tree Ensembles

7.3.1 Experiments

In this section, tree ensembles are used for attribute prediction. Each model is trained with a rhythm feature vector, a timbre feature vector, and their combination. The rhythm feature vector is a combination of the Median removed *Mellin Scale Transform* DCT, multi-band *Beat Profiles*, and multi-band *Tempogram Ratios*. The timbre feature vector is a block based implementation of *MFCCs* [140].

Tree ensembles were employed because they can more powerfully represent non-linear relationships that may be present between the rhythm attributes and the acoustic features. Both *Random Forests* (RF) [123] and *Gradient Boosted Trees* (GBT) [122] formulated for both classification of binary attributes (e.g., meter) and regression of continuous attributes (e.g., danceability) are used. Additionally, similar to work by He [127], tree ensembles can function as a feature transformation and the output of each leaf can be used input features to a simpler classification or regression model (RF-H, GBT-H). The output decisions of each tree in the ensemble can be used as a new feature set that exploits the relationships of ensemble predictions. In this work, I use the leaf outputs of each tree as inputs into linear classifiers (*Logistic Regression*) and regressors (*Linear Regression*) trained using stochastic gradient descent (SGD). These models are discussed further in Chapter 4.

For each method, a 70%:30% (train:test) split of the data was used, and no artists were shared between the training and testing sets. For comparison, I will evaluate these methods against the linear models presented in previous work (Section 7.2) [10]. When training each model, the following tree parameters were tuned across generally accepted ranges: tree depth (3-8), number of estimators (50-250), percentage of features used per estimator (12.5%-50%). Only the best models will be discussed.

7.3.2 Results

The results for each of the experiments are shown in Table 7.2. Each model was trained using only rhythm features, only timbre features, or their combination. It is seen across the board, and similar to previous work [10], that the rhythm features perform better than timbre features when modeling rhythmic attributes. The combination of rhythm and timbre performs only slightly better than when using rhythm features alone. Each of the tree models outperform each of the linear models, suggesting that the relationship of rhythm features to rhythm attributes are more complex than those captured by linear models. When considering the tree ensemble models, the GBTs and GBT-Hs generally outperform RFs and RF-Hs respectively. For GBTs, the hybrid approach (GBT-H) in this context is not very helpful. However, for RFs, the RF-H approaches are helpful, especially for regression of the continuous attributes (danceability, back-beat), with model performance approaching that of the GBT and GBT-H models.

Features	Model	AUC		Comp.					R^2	
		Cut	Triple	Duple	Odd	Swing	Shuf.	Sync.	Dance	Back-Beat
Timbre	Linear	0.797	0.794	0.663	0.745	0.781	0.719	0.705	0.400	0.309
	RF	0.842	0.811	0.702	0.791	0.830	0.758	0.738	0.515	0.340
	GBT	0.877	0.808	0.689	0.769	0.828	0.761	0.737	0.570	0.401
	RF-H	0.853	0.817	0.707	0.793	0.835	0.762	0.750	0.560	0.373
	GBT-H	0.868	0.820	0.713	0.793	0.839	0.764	0.759	0.553	0.372
Rhythm	Linear	0.901	0.924	0.946	0.859	0.903	0.919	0.768	0.505	0.316
	RF	0.926	0.938	0.960	0.870	0.916	0.926	0.779	0.554	0.364
	GBT	0.944	0.956	0.962	0.862	0.932	0.928	0.779	0.615	0.463
	RF-H	0.930	0.943	0.960	0.875	0.922	0.927	0.787	0.602	0.444
	GBT-H	0.939	0.951	0.961	0.886	0.925	0.928	0.786	0.603	0.449
Timbre + Rhythm	Linear	0.905	0.920	0.943	0.865	0.903	0.919	0.777	0.486	0.441
	RF	0.930	0.936	0.960	0.881	0.921	0.930	0.791	0.594	0.417
	GBT	0.949	0.956	0.960	0.877	0.935	0.930	0.793	0.645	0.505
	RF-H	0.934	0.943	0.960	0.883	0.926	0.931	0.802	0.634	0.477
	GBT-H	0.946	0.953	0.962	0.899	0.931	0.932	0.805	0.631	0.487

Table 7.2: The rhythmic attribute learning is evaluated with area under the ROC curve (AUC) for classification and R^2 for regression.

7.4 Conclusion

A set of large-scale experiments was performed to quantify and label a set of rhythmic meter and feel attributes using the *Pandora*[®] *Music Genome Project*[®]. From a musicology perspective, these rhythmic attributes are important in the makeup of a musical style. From this work, we gain insight into the meanings of rhythmic features as they relate to meter and feel when applying them to style recognition tasks in the future. In later work, more complex, scalable models employing Random Forests, Gradient Boosted Trees and stacked tree ensembles [127] were evaluated. Similar to neural-network models, tree ensembles benefit from the ability to learn complex, non-linear mappings of the data. It was found that the tree ensemble methods are better than linear methods when modeling the complexities of rhythmic attributes. In most cases, Gradient Boosted Trees (GBT) perform best. For GBTs, the addition of the hybrid approach (GBT-H) provides little gain. However, the hybrid approach for Random Forests (RF-H), was helpful when modeling continuous attributes, which is an intriguing result.

Chapter 8: Learning Genre from Rhythmic Attributes

Just because genres are widely used does not necessarily mean that they are easy to categorize, or easy to recognize. In fact, previous research shows that the music industry uses inconsistent genre taxonomies [141], and there is debate over whether genre is the product of objective or subjective categorizations [135]. Furthermore, it is debated whether individual musical properties (e.g. tempo, rhythm, instrumentation), which are not always exclusive to a single genre, represent defining components [142, 143]. For example, an Afro-Latin clave pattern occurs many places, both in Antonio Carlos Jobim’s *The Girl from Ipanema* (Jazz) and in The Beatles’ *And I Love Her* (Rock). It can even be heard in the popular song, *All About that Bass*, by Meghan Trainor. However, when discriminating the more specific subgenres of ‘Bebop’ Jazz (fast swing) and ‘Brazilian’ Jazz (Afro-Latin rhythms), this clave property becomes much more salient. Despite these intriguing relationships, a large-scale analysis of the association of musical properties to genre has yet to be performed.

If it were possible to define a categorization of music genres that is useful, meaningful, consensual and consistent *at some level*, then an automated categorization of music pieces into genres would be both achievable and highly desirable. Since early research in Music Information Retrieval (MIR), and still to date, the automatic genre recognition from music pieces has precisely been an important topic [5, 143, 135]. In this chapter, the intriguing relationship of genre and musical attributes is explored.

8.1 Approach

In this chapter I outline four approaches to modeling musical genre, investigating both expert human annotations as well as audio representations (Figure 8.1). Attribute and genre relationships are evaluated using subset of 12 ‘Basic’ musical genres (e.g. Jazz) as well as a selected subset of 47 subgenres (e.g. Bebop). In the first approach (1), I address via data-driven experiments whether objective musical attributes of music pieces carry sufficient information to categorize their genre.

The next set of approaches (2a, 2b, 2c) uses audio features to model genre automatically. In the

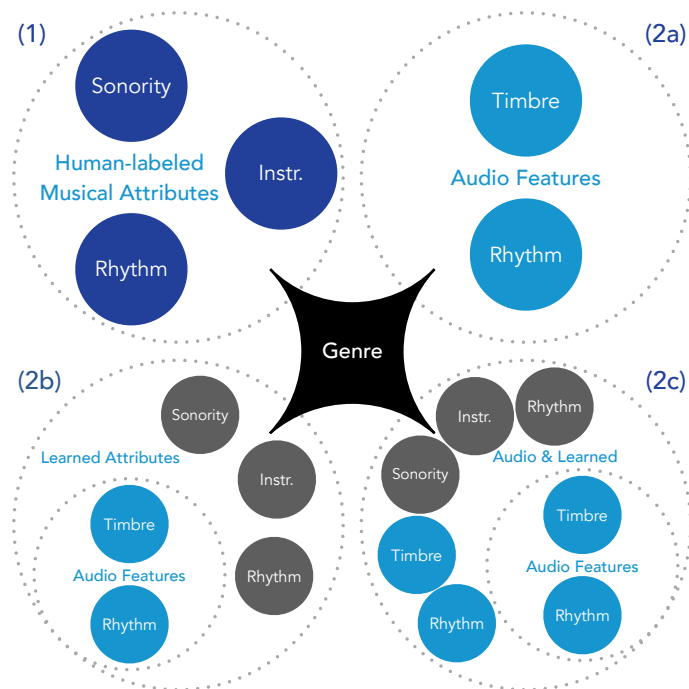


Figure 8.1: An overview of the feature types used in experiments performed.

second approach, audio features are used to categorize genre directly (2a). The third approach (2b) uses audio features to model each of the musical attributes individually, similar to Chapter 7. Estimated activations of each of those attribute models are then used as features to model and categorize genre. In the fourth approach (2c), the estimated attributes are used in conjunction with raw audio features. By injecting human-inspired context, I hope to automatically capture elements of genre in a manner similar to that of models derived from attributes labeled by music experts.

8.2 Data: The Music Genome Project

Both the musical attribute and genre labels used were defined and collected by musical experts on a corpus of over one million music pieces from the *Pandora[®] Music Genome Project[®] (MGP)*. The labels were collected over a period of nearly 15 years and great care was placed in defining them and analyzing each song with that consistent set of criteria.

8.2.1 Musical Attributes

The musical attributes refer to specific musical components comprising elements of the vocals, instrumentation, sonority, and rhythm. They are designed to have a generalized meaning across all genres (in western music) and map to specific and deterministic musical qualities. In this work, I choose subset of 48 attributes (10 rhythm, 38 timbre). An overview of the attributes is shown in Table 8.1.

Meter attributes denote musical meters separate from simple duple (e.g, cut-time, compound-duple, odd)
Rhythmic Feel attributes denote rhythmic interpretation (e.g., swing, shuffle, back-beat strength) and elements of rhythmic perception (e.g., syncopation, danceability)
Vocal attributes denote the presence of vocals and timbral characteristics of voice (e.g., male, female, vocal grittiness).
Instrumentation attributes denote the presence of instruments (e.g., piano) and their timbre (e.g., guitar distortion)
Sonority attributes describe production techniques (e.g., studio, live) and the overall sound (e.g., acoustic, synthesized)

Table 8.1: Explanations of rhythm and orchestration attributes .

Each of the attributes is rated on continuous scale from 0-1. In some contexts, when developing models for attribute prediction, it is helpful to convert them to binary labels if they show only low (absence) or high (presence) ratings with little in between. More information on each of the attributes can be found in Chapter 3.

8.2.2 Genre and Subgenre

In this work, evaluation is performed using a selected subset of 12 ‘Basic’ genres and 47 additional sub-genres. ‘Basic’ genre is assembled as a mix of very expansive genres (e.g., Rock, Jazz) as well as some more focused ones (e.g., Disco and Bluegrass), serving as an analog to many previous genre experiments in MIR. The presence of a genre is notated independently for each song by a binary label. A selection of genre labels and a simplistic high-level organization for discussion purposes is shown in Table 8.2. More information on each of the genres can be found in Chapter 3.

Basic Genre:	Rock, Jazz, Rap, Latin, Disco, Bluegrass, etc.
Jazz Subgenre:	Cool, Fusion, Hard Bop, Afro-Cuban, etc.
Rock Subgenre:	Light, Hard, Punk, etc.
Rap Subgenre:	Party, Old School, Hardcore, etc.
Dance Subgenre:	Trance, House, etc.
World Subgenre:	Cajun, North African, Indian, Celtic, etc.

Table 8.2: Some of the musical genres and subgenres used.

8.3 Musical Attribute Models of Genre

In order to see the extent to which genre can be modeled by musical attributes, a set of applied musicology experiments is first performed. The set of expertly-labeled attributes from Section 8.2.1 is used to classify genre. A model for each individual genre is trained on each of the musical attributes alone and in rhythm- and timbre-based aggregations. This will show the role that each attribute or collection of attributes plays and how they interact with one another in order to create joint representations of genre. Each model employs *Logistic Regression* trained using *stochastic gradient descent* (SGD). The training data was separated on a randomly shuffled 70%:30% (train:test) split with no shared artists between training and testing. Due to the size of the dataset, a single trial for each attribute is both tractable and sufficient. The learning rate for each genre model is tuned adaptively.

8.3.1 Evaluating the Role of Musical Attributes

In order to evaluate each of the models, the area under the receiver operating characteristic curve (AUC) will be used. Each genre has large and varying class imbalance, so this is first corrected for by weighting training examples by their inverse class count in the cost function. However, accuracy alone still does not tell the whole story. High accuracy can be achieved by predicting only the negative class (genre absence). Area under the ROC curve allows for a more comparable difference between each of the models than raw accuracy alone. It gives insight into the trade-off between true positive and false positive rates. Alternatively I could have used precision and recall (PR) curves for

evaluation, but it is shown that if one model dominates in the ROC domain, it will also dominate in the PR domain and vice-versa [144].

The results for each of the attribute-based genre models are shown in Tables 8.3 and 8.4. The tables outline the AUC values for classifying genre using orchestration attributes, rhythm attributes, and their combination. Table 8.3 summarizes all results, showing the mean of all AUC values for each genre model contained in the subgroups defined in Section 8.2.2. Using attributes of rhythm and orchestration together show better performance than using each alone. Secondly, orchestration tends to perform better than rhythm. This suggests that the orchestration attributes in this context are better descriptors. However in some cases, the rhythm attributes, even though there are less of them (10 rhythm, 38 orchestration), are not that far behind. They are especially important in defining Jazz and Rap, where rhythms such as swing in Jazz or syncopated vocal cadences over back-beat heavy drums in Rap play defining roles.

Genre Group	Orch.	Rhythm	Both
Basic	0.905	0.841	0.918
Rock Sub	0.910	0.819	0.919
Jazz Sub	0.925	0.856	0.945
Rap Sub	0.901	0.891	0.940
Dance Sub	0.961	0.881	0.965
World Sub	0.885	0.833	0.904
Mean	0.913	0.848	0.931

Table 8.3: An overview of all models using musical attributes.

In Table 8.4 I show the individual AUC results for the set of ‘Basic’ genres and subgenres of Jazz. Within these individual groups, rhythm and orchestration attributes together are once again able to better represent genre than when used individually. Each of the ‘Basic’ genres can be represented reasonably well with just orchestration, as each has slightly differing instrumentation. However, we again see the importance of rhythm, describing what instrumentation and timbre cannot capture alone. Genres heavily reliant on specific rhythms (e.g., Funk, Rap, Latin, Disco, Jazz) are all able to be represented rather well with only rhythm attributes. In the Jazz subgenre this emphasis on rhythm in certain cases is even more clear. In the next subsection, I will dive deeper into the attributes that best describe the Jazz subgenres.

Basic				Jazz			
Genre	Orch.	Rhythm	Both	Subgenre	Orch.	Rhythm	Both
Rock	0.843	0.759	0.856	New Orleans	0.970	0.957	0.989
Blues	0.913	0.783	0.915	Boogie	0.943	0.893	0.978
Gospel	0.810	0.664	0.843	Swing	0.970	0.933	0.984
Soul	0.869	0.793	0.887	Bebop	0.976	0.965	0.988
Funk	0.937	0.862	0.937	Cool	0.964	0.928	0.975
Rap	0.926	0.890	0.951	Hard Bop	0.944	0.905	0.967
Folk	0.943	0.760	0.952	Fusion	0.843	0.750	0.886
Country	0.952	0.794	0.955	Free	0.906	0.855	0.936
Reggae	0.893	0.819	0.905	Afro-Cuban	0.961	0.910	0.972
Latin	0.940	0.904	0.945	Brazilian	0.871	0.847	0.905
Disco	0.899	0.891	0.902	Acid	0.886	0.660	0.891
Jazz	0.937	0.850	0.963	Smooth	0.862	0.667	0.871
Mean	0.905	0.814	0.918	Mean	0.925	0.856	0.945

Table 8.4: Experimental results for ‘Basic’ genre and Jazz subgenre models using musical attributes.

8.3.2 The Influence of Rhythm and Orchestration in Jazz

In order to more deeply explore the defining relationships of rhythm and instrumentation within a subgenre, we will look further into Jazz. Table 8.5 shows a subset of the important musical attributes for the Jazz subgenres. The AUC accuracy of classifying each subgenre based on individual musical attributes is shown.

Jazz Subgenre	Orch.				Aux. Perc.	Rhythm				
	Solo Brass	Piano	Reeds			BackBeat	Dance	Swing	Shuffle	Syncop.
New Orleans	0.808	0.786	0.790	0.680	0.652	0.564	0.936*	0.513	0.515	
Boogie	0.510	0.924*	0.544	0.714	0.592	0.712	0.737	0.505	0.676	
Swing	0.721	0.784	0.748	0.679	0.624	0.578	0.923*	0.511	0.508	
Bebop	0.725	0.850	0.862	0.703	0.662	0.525	0.946*	0.509	0.602	
Cool	0.639	0.750	0.836	0.701	0.697	0.424	0.890*	0.504	0.568	
HardBop	0.606	0.774	0.737	0.669	0.726	0.555	0.808*	0.684	0.606	
Fusion	0.604	0.497	0.669	0.507	0.574	0.577	0.507	0.500	0.693*	
Free	0.606	0.538	0.784	0.615	0.809*	0.765	0.577	0.515	0.558	
Afro-Cuban	0.696	0.822	0.706	0.832*	0.782	0.648	0.512	0.501	0.790	
Brazilian	0.560	0.736	0.568	0.572	0.761*	0.555	0.532	0.504	0.635	
Acid	0.591	0.513	0.658*	0.507	0.585	0.622	0.509	0.515	0.635	
Smooth	0.530	0.577	0.748*	0.590	0.559	0.614	0.513	0.509	0.573	

Table 8.5: Attributes important to the Jazz subgenres are shown. AUC values greater than 0.70 are bold. The highest performing attribute for each genre is denoted with a *.

The presence of solo brass (e.g, trumpet), piano, reeds (e.g., saxophone) and auxiliary percussion (e.g., congas) are important defining characteristics of instrumentation. Boogie and Afro-Cuban styles, even though different, place heavy emphasis on the piano, which is shown here as well.

Bebop, Hard-bop, and Afro-Cuban Jazz show emphasis placed on solo brass, piano, and reeds, as they rely heavily on solo artists of these instruments (e.g., “Dizzy” Gillespie, Miles Davis, Thelonious Monk, John Coltrane). The presence of auxiliary percussion is also a good descriptor of Afro-Cuban Jazz, where the use of hand drums (e.g., bongos, congas) is very prevalent.

Rhythm is also important in Jazz subgenres. The danceability, back-beat, and presence of swing and syncopation are defining characteristics of certain Jazz rhythms. It is important to note that a high AUC does not necessarily denote the presence of that attribute, only its consistent relationship. For example, back-beat is a good predictor of Free Jazz possibly due to its absolute absence. Alternatively, one may think that the presence of swing is important in all Jazz. Bebop, Hard Bop, New Orleans, and Swing Jazz do have a heavy dependence on swing being present. However, Afro-Cuban Jazz relies on straight time, clave-based rhythms, so syncopation is a better predictor. It is also important to note that while the attributes of swing and shuffle are musically related, there is a clear distinction in their application. In this case, swing is very important, while shuffle is only slightly useful (e.g., Boogie). However, outside of the Jazz genre, the opposite case may be true, where shuffle is the more important attribute (e.g. Blues, Country). This suggests that it is important to make a clear distinction between swing and shuffle.

8.4 Predicting Genre from Audio

There is a large body of work on musical genre recognition from audio signals [5, 135]. However, most known prior work in this area focuses on discriminating a discrete set of basic genre labels with little emphasis on what defines genre. In response, researchers have tried to develop datasets that focus on style or subgenre labels (e.g., ballroom dance [145, 51, 53], latin [86], electronic dance [146], Indian [147]) that have clear relations to the presence of specific musical attributes. However, because models are designed for these specific sets, the methods used do not adapt to larger more generalized music collections. For example, tempo alone is a good descriptor for the ballroom dance style dataset, which is not true for more general collections [145].

Other work in genre recognition avoids the problem of strict genre class separations. Audio feature similarity, self organizing maps, and nearest-neighbor approaches can be used estimate genre

of an unknown example [84]. Similarly, auto-tagging approaches use audio features to learn the presence of both musical attributes and genre tags curated by the public [99, 148] or by experts [101].

In this section, I compare modeling genre both with audio features directly and with stacked approaches that exploit the relationships of audio features and musical attributes.

8.4.1 Timbre Related Features

In order to capture timbral components and model vocal, instrumentation, and sonority attributes, block-based Mel-Frequency Cepstral Coefficients (MFCC) are implemented. Means and covariances of 20 MFCCs are calculated across non-overlapping 3-second blocks. These block-covariances are further summarized over the piece by calculating their means and variances [138]. This yields a 460 dimensional timbre based feature set.

8.4.2 Rhythm Related Features

In order to capture aspects of each rhythm attribute, a set of rhythm-specific features was employed. All rhythm features described in this section rely on global estimates of an accent signal [21]. The *beat profile* quantizes the accent signal between consecutive beats to 36 subdivisions. The beat profile features are statistics of those 36 bins over all beats. The feature relies on estimates of both beats [33] and tempo. The *tempogram ratio* feature (TGR) uses the tempo estimate to remove the tempo dependence in a tempogram. By normalizing the tempo axis of the tempogram by the tempo estimate, a fractional relationship to the tempo is gained. A compact, tempo-invariant feature is created by capturing the weights of the tempogram at musically related ratios relative to the tempo estimate. The *Mellin scale transform* is a scale invariant transform of a time domain signal. Similar musical patterns at different tempos are scaled relative to the tempo. The Mellin scale transform is invariant to that tempo scaling. It was first introduced in the context of rhythmic similarity by Holzapfel [60], around which our implementation is based. In order to exploit the natural periodicity in the transform, the discrete cosine transform (DCT) is computed. Median removal (by subtracting the local median) and half-wave rectifying the DCT creates a new feature that emphasizes transform periodicities.

The rhythm features are also extended to multiple-band versions by using accent signals that are constrained to be within a set of specific sub-bands. This affords the ability to capture the rhythmic function of instruments in different frequency ranges. The rhythm feature set used in this work is an aggregation of the median removed Mellin Transform DCT and multi-band representations of the beat profile and the tempogram ratio features. This yields a 372 dimensional rhythm based feature set that was shown in previous work to be relatively effective at capturing musical attributes related to rhythm (see Chapter 6 for more details).

8.4.3 Genre Recognition Experiments

In addition to the first experiment from Section 8.3, three additional methods for modeling genre are presented, each based on audio signal analysis. The second method (2a in Figure 8.1) performs the task of genre recognition with rhythm and timbre inspired audio features directly. The third method (2b in Figure 8.1) is motivated similar to the first experiment, which employs the expertly-labeled musical attributes. However, inspired by work in transfer learning [149], audio features are used to develop models for the humanly-defined attributes which in turn are used to model genre. Through this supervised pre-training of musical attributes, models of genre can be learned from attributes' estimated presence. In the fourth approach (2c in Figure 8.1), inspired by Deng [150] and Knees [92], the learned attributes are combined with the audio features directly in a shared middle layer to train models of genre.

Similar to Section 8.3, genre is modeled with *Logistic Regression* fit using *stochastic gradient descent* (SGD). The data was separated on the same 70%:30% (train:test) split. Once again, there were no shared artists between training and testing. Due to the size of the dataset, a single trial for each genre, as well as for each learned musical attribute, is both tractable and sufficient. The learning rate for each model is tuned adaptively.

Using Audio Features Directly

Of the four presented approaches, the second uses audio features directly to model genre. The features from Sections 8.4.1 and 8.4.2 are used in aggregation and a model is trained and tested for

each individual genre. This provides a baseline for what audio features are able to capture without any added context. However, this lack of context makes it hard to interpret what about genre they are capturing.

Stacked Methods

The third and fourth approaches are also driven by audio features. However instead of targeting genre directly, models are learned for each of the vocal, instrumentation, sonority, and rhythm attributes. Inspired by approaches in transfer learning [149], and similar in structure to previous methods in the MIR community [151], the learned attributes are then used to predict genre. This approach is formulated similar to a basic neural network with a supervised pre-trained (and no longer hidden) musical attributes layer.

The rhythm-based attributes are modeled with a feature aggregation of the *Mellin Scale Transform DCT*, multi-band *Beat Profile*, and multi-band *Tempogram Ratio* features. The vocals, instrumentation, and sonority attributes are modeled with the block-based MFCC features. Each attribute is modeled using logistic regression for binary labels (categorical) and linear regression for continuous labels (scale-based). If an individual attribute is formulated as a binary classification task (see Section 8.2.1), the probability of the positive class (its presence) is used as the feature value.

The first version of the stacked methods (third approach) uses audio features to estimate musical attributes and employs only those estimated attributes to model genre. The second version (fourth approach) concatenates the audio features and the learned attributes in a shared middle layer to model genre [150, 92].

8.4.4 Results

In this section, I will present an overview of all of the results from the audio-based methods, and compare them to the models learned from the expertly-labeled attributes. In order to show the overall performance of each method in a compact way, only combined rhythm and timbre approaches will be compared initially. Once again each genre model will be evaluated using area under the ROC

curve (AUC). In order to better evaluate the stacked models, I will finish with a brief evaluation of the learned attributes.

Learning Genre

A summary of the results for the audio experiments using rhythm and timbre features is shown in Table 8.6. The human attribute model results are also included for comparison. Similar to Table 8.3, the mean AUC of each genre grouping is shown.

Genre Group	Human Attrib.	Audio Feat.	Learned Attrib.	Audio + Learned
Basic	0.918	0.892	0.878	0.899
Rock Sub	0.919	0.902	0.903	0.911
Jazz Sub	0.945	0.910	0.893	0.923
Rap Sub	0.940	0.916	0.914	0.927
Dance Sub	0.965	0.963	0.955	0.966
World Sub	0.904	0.850	0.846	0.865
Mean	0.931	0.905	0.897	0.915

Table 8.6: An overview of experimental results using audio-based models that utilize both timbre and rhythm features.

Compared to the human attributes approach, using audio features alone to model genre performs relatively well. This is especially true for the ‘Basic’, Rock, and Dance groups, where the audio feature AUC results are very close to human attribute performance. Across the other groups, the differences between the audio feature models and the musical attribute models suggest that the audio features lose some important, genre-defining information. Furthermore, performance that was close to musical attributes when using only audio features alone is also close when musical attributes learned from audio features. This suggests that, in these cases, the audio features may be capturing similarly salient components related to the musical attributes that describe these genre groups.

Overall, the learned attributes perform just as good as or worse than the audio features alone. This suggests that they are at most as powerful as the audio features used to train them. However, combining audio features and learned attributes shows significant improvement (paired t-test $p < 0.01$ across all genres) over using audio features or learned attributes alone. This evidence suggests that audio features and learned attribute models each contain slightly different information. The added human context of the learned attributes is helpful to achieve results that approach those of

the expertly-labeled attributes. This also suggests that the decisions made from learned labels are possibly more similar to the decisions made from human attribute labels, and the errors in estimating the musical attributes are possibly to blame for the performance decrease when used alone.

Basic Genre	Human Attrib.	Audio Feat.	Learned Attrib.	Audio + Learned	Jazz Subgenre	Human Attrib.	Audio Feat.	Learned Attrib.	Audio + Learned
Rock	0.856	0.831	0.835	0.839	New Orleans	0.989	0.947	0.951	0.956
Blues	0.915	0.892	0.883	0.899	Boogie	0.978	0.962	0.939	0.962
Gospel	0.843	0.798	0.794	0.805	Swing	0.984	0.929	0.929	0.940
Soul	0.887	0.833	0.818	0.842	Bebop	0.988	0.951	0.943	0.957
Funk	0.937	0.911	0.886	0.918	Cool	0.975	0.900	0.901	0.916
Rap	0.951	0.963	0.951	0.969	HardBop	0.967	0.946	0.930	0.952
Folk	0.952	0.905	0.903	0.916	Fusion	0.886	0.844	0.812	0.867
Country	0.955	0.885	0.880	0.897	Free	0.936	0.920	0.923	0.931
Reggae	0.905	0.926	0.885	0.929	AfroCuban	0.972	0.934	0.912	0.946
Latin	0.945	0.921	0.905	0.923	Brazilian	0.905	0.879	0.858	0.904
Disco	0.902	0.936	0.893	0.938	Acid	0.891	0.841	0.763	0.846
Jazz	0.963	0.907	0.906	0.916	Smooth	0.871	0.868	0.853	0.894
Mean	0.918	0.892	0.878	0.899	Mean	0.945	0.910	0.893	0.923

Table 8.7: Experimental results for the ‘Basic’ genres and Jazz subgenres using audio-based models.

The left half of Table 8.7 shows the results for predicting the ‘Basic’ genre labels. Within this set, we see some interesting patterns start to emerge. In the case of Rap, Reggae, and Disco, audio features are actually able to out-perform the musical attributes. This suggests that our small selected subset of 48 human attribute labels do not always tell the complete story, and that the audio features, which are much larger in dimensionality, possibly contain additional and/or different information. As in previous results, the learned attribute models perform similarly to methods that use audio features directly, but with a few exceptions. In the cases that the audio feature models do better than the human-labeled musical attribute models, the learned attribute models are able to perform *at most* as good as the human-labeled musical attribute models. This once again suggests that the learned attribute approach may be better approximating the decisions the human-labeled attribute approach is making. When adding audio and learned attributes together, the added context is once again beneficial, with combined methods outperforming models that use audio or learned attributes alone. Audio feature models that perform better than the human attributes models are additionally improved, showing again that the audio features and human attribute labels contain complementary information.

The right half of Table 8.7 shows the results for predicting the Jazz subgenre labels. The Jazz genre shows more expected relationships between the human attribute, audio feature, and learned attribute methods. The combined method outperforms each of the audio feature and learned attribute methods. The human attribute method performs better than almost all audio-based methods (except Smooth).

Extended Genre Results

In this section, I present a series of extended results for genre classification that separates timbral and rhythm attributes for each model. The mean AUC for each of the genre and sub-genre groups are shown in Table 8.8. The ‘human’ column of this table is identical to Table 8.3. In using the hand-labeled musical attributes, the orchestration relationships to genre are more informative than the rhythm relationships. When looking at the learned-attribute models, some rhythm relationships become more important than orchestration ones, specifically for the Jazz and the World sub-genres. This is possibly due to the ability to better represent rhythm attribute models than timbre attribute models. This is further shown in using audio features directly. The rhythm features across the board are better able to represent genre, showing that they are more powerful than the timbre audio features. The rhythm audio features are also more powerful than the rhythm musical attributes across the board. This also provides some evidence that the rhythm features are capturing some information missed in vector of rhythmic attributes. These relationships are preserved when looking deeper among ‘Basic’ genre and Jazz sub-genre. In Table 8.9 I show the results for classifying ‘Basic’ genre with features separated by timbral and rhythm contexts. Table 8.10 shows results for classifying Jazz sub-genre with features separated by timbral and rhythm contexts.

Learning Attributes of Rhythm and Instrumentation

In order to further explore the stacked audio-based models, I performed a small evaluation of how well the audio features are able to learn each of the expertly-labeled musical attributes. Sticking with a common theme, I will explore the results of modeling attributes that are important to Jazz (from Table 8.5). Table 8.11 shows the ability to directly predict these attributes from audio features. AUC

Genre Group Avg. AUC	Human			Audio			Learned		
	T	R	T+R	T	R	T+R	T	R	T+R
Basic	0.905	0.814	0.918	0.830	0.863	0.892	0.866	0.796	0.878
Rock Sub	0.910	0.819	0.919	0.863	0.875	0.902	0.900	0.833	0.903
Jazz Sub	0.925	0.856	0.945	0.850	0.899	0.910	0.829	0.846	0.893
Rap Sub	0.901	0.891	0.940	0.872	0.907	0.916	0.904	0.816	0.914
Dance Sub	0.961	0.881	0.965	0.933	0.947	0.963	0.950	0.898	0.955
World Sub	0.885	0.833	0.904	0.792	0.833	0.850	0.797	0.801	0.846
Mean	0.913	0.848	0.931	0.857	0.887	0.905	0.874	0.832	0.897

Table 8.8: An overview of all experimental results for timbre (T) and rhythm (R) attributes as well as their combination (T+R). Shown in each row is the mean of all genre classification task within a given group.

Basic Genre AUC	Human			Audio			Learned		
	T	R	T+R	T	R	T+R	T	R	T+R
Rock	0.843	0.759	0.856	0.809	0.793	0.831	0.834	0.763	0.835
Blues	0.913	0.783	0.915	0.816	0.876	0.892	0.880	0.828	0.883
Gospel	0.810	0.664	0.843	0.775	0.737	0.798	0.769	0.673	0.794
Soul	0.869	0.793	0.887	0.771	0.804	0.833	0.813	0.722	0.818
Funk	0.937	0.862	0.937	0.843	0.912	0.911	0.885	0.795	0.886
Rap	0.926	0.890	0.951	0.922	0.938	0.963	0.940	0.859	0.951
Folk	0.943	0.760	0.952	0.870	0.874	0.905	0.901	0.812	0.903
Country	0.952	0.794	0.955	0.798	0.869	0.885	0.879	0.838	0.880
Reggae	0.893	0.819	0.905	0.888	0.909	0.926	0.883	0.799	0.885
Latin	0.940	0.904	0.945	0.843	0.909	0.921	0.901	0.865	0.905
Disco	0.899	0.891	0.902	0.862	0.938	0.936	0.891	0.873	0.893
Jazz	0.937	0.850	0.963	0.851	0.888	0.907	0.898	0.842	0.906
Mean	0.905	0.814	0.918	0.837	0.871	0.892	0.873	0.806	0.878

Table 8.9: Classifying ‘Basic’ genre using timbre (T) and rhythm (R) attributes as well as their combination (T+R).

accuracies are reported for the binary attributes; R^2 values are reported for continuous attributes.

The results of evaluating the model for the training and testing sets is shown.

First of all, we see that testing and training AUC is almost identical. Because of this, a single trial (fold) is appropriate when learning attribute models. The learned models should generalize over all music without over fitting. This justifies using the the same 70%:30% (train:test) split for each layer in the stacked models. We see that MFCC’s do somewhat well for brass and reeds, but the lower AUC overall shows that these timbre features are not doing enough to capture these attributes, which may be a source of error in genre models that rely heavily on timbre. However, the rhythm results are much better, especially for swing and shuffle, which was argued in Section 8.3 and Table 8.5 as an important distinction to make when predicting Jazz subgenres.

Jazz Subgenre AUC	Human			Audio			Learned		
	T	R	T+R	T	R	T+R	T	R	T+R
NewOrleans	0.970	0.957	0.989	0.878	0.922	0.947	0.900	0.904	0.951
Boogie	0.943	0.893	0.978	0.912	0.908	0.962	0.876	0.889	0.939
Swing	0.970	0.933	0.984	0.849	0.926	0.929	0.875	0.903	0.929
Bebop	0.976	0.965	0.988	0.903	0.932	0.951	0.888	0.909	0.943
Cool	0.964	0.928	0.975	0.875	0.866	0.900	0.826	0.875	0.901
HardBop	0.944	0.905	0.967	0.900	0.930	0.946	0.888	0.884	0.930
Fusion	0.843	0.750	0.886	0.788	0.846	0.844	0.790	0.749	0.812
Free	0.906	0.855	0.936	0.846	0.922	0.920	0.850	0.910	0.923
AfroCuban	0.961	0.910	0.972	0.830	0.939	0.934	0.778	0.887	0.912
Brazilian	0.871	0.847	0.905	0.845	0.889	0.879	0.758	0.771	0.858
Acid	0.886	0.660	0.891	0.775	0.848	0.841	0.725	0.727	0.763
Smooth	0.862	0.667	0.871	0.804	0.864	0.868	0.795	0.746	0.853
Mean	0.925	0.856	0.945	0.850	0.899	0.910	0.829	0.846	0.893

Table 8.10: Classifying Jazz sub-genre using timbrel (T) and rhythm (R) attributes as well as their combination (T+R).

Musical Attributes	Audio Features	Training AUC/ R^2	Testing AUC/ R^2	Label Type
Solo Brass	Timbre	0.796	0.798	binary
Piano	Timbre	0.721	0.716	binary
Reeds	Timbre	0.790	0.789	binary
Aux Percussion	Timbre	0.750	0.750	binary
FeelSwing	Rhythm	0.907	0.902	binary
FeelShuffle	Rhythm	0.919	0.920	binary
FeelSyncopation	Rhythm	0.772	0.770	binary
FeelBackBeat	Rhythm	0.400	0.393	continuous
FeelDance	Rhythm	0.527	0.515	continuous

Table 8.11: The results for learning binary (AUC) and continuous (R^2) attributes important to Jazz are shown.

Table 8.12 shows a summary of learning the all of the selected 48 attributes from audio features. It shows similar trends to Table 8.11, with rhythmic attributes better described by audio features than timbral attributes. Furthermore, the continuous timbral attributes, which are sometimes complicated perceptually (e.g., vocal grittiness), are not modeled very well at all. This suggests that MFCC's, and possibly other spectral approximations, do not provide the full picture into what we perceive as the components of timbre. This is especially true in the context of instrument identification in mixtures, which is a main utility of the timbre features in this context. While these models as a whole can be improved, the problems of instrument identification and timbrel analysis are separate, large, and active research areas [152, 153, 154].

Attribute Type	Num	Mean	Median	Maximum
Continuous Rhythm (R^2)	3	0.432 ± 0.077	0.393	0.515
Continuous Timbre (R^2)	12	0.266 ± 0.192	0.194	0.514
All Continuous	15	0.299 ± 0.186	0.389	0.515
Binary Rhythm (AUC)	7	0.889 ± 0.059	0.902	0.946
Binary Timbre (AUC)	26	0.794 ± 0.074	0.794	0.925
All Binary	15	0.814 ± 0.080	0.806	0.946

Table 8.12: Overall summary of learned attributes.

8.5 Conclusion

In this work, it was demonstrated that there is potential to demystify the constructs of musical genre into distinct musicological components. The attributes selected from music experts are able to provide a great deal of genre distinguishing information, but this is only an initial investigation into these questions. I was also able to discover and outline the importance of certain attributes in specific contexts. This strongly suggests that the expression of musical attributes are necessary additions to definitions of genre.

It was also shown here (and in previous work in Section 7 [10]) that audio features motivated by timbre and rhythm are, with some success, able to model musical attributes. Audio features are also able to describe musical genre directly and through stacked approaches that exploit the learned models of musical attributes. This is strong evidence suggesting that audio-based approaches are learning the presence of the musical attributes, to some degree, when distinguishing genre. In some cases, the audio-based models were more powerful than the human musical attribute models. This suggests that there is more to genre than the chosen subset of rhythm and orchestration attributes. This prompts that there is more about the definition of genre yet to be discovered.

Chapter 9: Exploring Intuitive Feature Space Reductions for Rhythm

Rhythm is one of the most intuitive aspects of music with which humans can identify. People can easily perform tasks like tapping along with the beat, or recognizing similarities/differences in style. However, it is sometimes difficult to pinpoint which aspects of the music inform this intuition. Creative and complex combinations of rhythmic attributes combine to create cohesive, distinct, and easily recognizable styles. I attempt to bridge this gap and create representations that capture a variety of rhythmic attributes in a joint and intuitively organized manner.

9.1 Motivation

There is a large body of work that has examined the general recognition of rhythmic styles in music audio signals [145, 155], but few efforts have focused on the deconstruction and quantification of the foundational components of global rhythmic structures and how they interact to form a musical style. Previous work has shown that rhythmic components have very important relationships to definitions of style and musical genre (Section 8 [156]). It was also shown in Section 7 that models trained with compact features derived from the rhythmic accent signal are quite effective when representing rhythm-related attributes of meter and feel (e.g., swing) [10, 157]. However, prediction of these attributes in isolation does not tell the full story. In this work, I try to demystify a high-level measure of similarity by defining an organized and interpretable space that is able to jointly represent multiple rhythmic attributes. Furthermore, motivated by work in transfer learning [149, 158], I will show the salience of this joint representation in other domains as well (i.e., genre, language).

The approach is outlined as follows: In Section 9.2, a set of widely used dimensionality reduction techniques are employed to reduce high dimensional feature spaces to a set of fundamental components. In Section 9.3, I outline methods to test the rhythm spaces' ability to represent each of the rhythmic attributes, their ability to regress examples through audio feature similarity, and

their ability to generalize to other musical attributes (i.e, genre). Finally, in Section 9.4 I evaluate a subset of hand selected rhythm space candidates with intuitively interesting projections.

9.2 Rhythm Space Reductions

In order to create a set of rhythm spaces, a selection of low-dimensional projections inspired by work in other domains of Music-IR is performed. Used in music emotion recognition, the Arousal-Valence (A-V) Space was developed from a set of discrete emotion tags projected into a two-dimensional space [159, 160]. Other projections, such as the Tempo-Loudness space [9] and Kinetics-Energy space [111], are derived directly from audio music signal analysis in order to more intuitively capture the seemingly complex aspects of human performance expression. Spaces derived from human-tagged attributes have the potential to follow a uniquely human organization, which may be helpful when designing a human-interpretable space. However, they can only capture information humans have already deemed important. Conversely, in designing a space from audio features, we may be able to capture rhythmic interactions that humans cannot easily deconstruct (i.e. syncopation). Another thing to consider is parametric vs. non-parametric reduction methods. Many classic techniques of dimensionality reduction are parametric, and learn a set of components and corresponding activations with the objective of reconstructing the original feature space. Non-parametric reductions (i.e, t-SNE) do not have this constraint and have been gaining traction in recent years for organizing and visualizing high-dimensional data [119, 120].

In order to accommodate these trade-offs, I create candidate spaces derived from acoustic feature representations of rhythm (Chapter 6 [10]) and from a collection of human-annotated rhythmic attribute labels (Chapter 3). On each of these data source types, I perform both parametric and non-parametric dimensionality reductions techniques.

9.2.1 Rhythm Attributes and Acoustic Features

The human-annotated rhythm attributes are compositional constructs such as meter and tempo or elements of the rhythmic feel (e.g., swing). Namely I focus on the 10 rhythmic attributes shown in Table 9.1 (top), which are labeled by music experts from the *Pandora*[®] *Music Genome*

Project[®](MGP). All of the expert-tagged rhythmic attributes are rated with a continuous value from 0 – 1.

In order to capture aspects rhythm in audio signals, a set of rhythm-inspired features was implemented (Table 9.1, bottom). Each relies on global estimates of an accent signal [21]. The accent signal is also be separated into multiple versions that are each constrained to specific frequency sub-bands, allowing for rhythms with different compositional functions (e.g., bass, lead) to be captured separately (see Chapter 6 for more details).

These human-annotated attribute values along with audio features are used to create the candidate rhythm reductions. These same rhythm attributes will also be used to evaluate the spaces in the coming sections.

Meter attributes denote musical meter distinct from simple duple: *cut-time meter*, *compound-duple meter*, *triple meter*, *odd meter*.

Swing denotes longer durations on the beat followed by a shorter duration. It usually occurs on the 2nd and 4th beats.

Shuffle is similar to swing, but warping is on every beat.

Syncopation is confusion created by early anticipation of the beat or obscuring meter with emphasis against beats.

Back-Beat Strength is the emphasis placed on the 2nd and 4th beat or grouping in a measure or set of measures.

Danceability is the utility of a song for dancing. There are consistent rhythmic groupings with emphasis on beats.

Tempo is speed of the music pulse. In this work, it is scored on a relative scale similar to the other attributes rather than representing a direct beats per minute (bpm) rating.

Beat Profile features are statistics of a quantized version of the accent signal between consecutive beat estimates.

Tempogram Ratio features are tempo-invariant relationships of musical event timings to an estimated tempo.

Mellin Transform Periodicity emphasizes periodicities in the Mellin Scale Transform using the discrete cosine transform, median removal, and half-wave rectification [60].

Table 9.1: The human-annotated rhythmic attributes defined by the MGP (top) and the rhythm audio features (bottom).

9.2.2 Learning a 2D space.

In order to learn low-dimensional rhythm space embeddings, I employ the widely used reduction techniques of *Principal Components Analysis* (PCA), *Independent Components Analysis* (ICA), *Non-Negative Matrix Factorization* (NMF) [130], and *t-Distributed Stochastic Neighbor Embedding* (*t-SNE*) [117]. The method was used to create a 2D space directly. PCA, ICA, and NMF were used

to create a range of n -dimensional component/activation spaces ($n = 2, 3, 6, 9, 12, 15$). A *supervised component selection* using an Analysis of Variance (ANOVA) test with respect to the rhythm labels was then performed to select two candidate dimensions for each reduction. The f-statistic was computed for each of the component activations with respect to each rhythm label, and the two dimensions were chosen that maximized the f-statistic (conditioned on significance $p < .01$) across all rhythm labels. This component selection was done both for the expert-tagged attribute and acoustic feature reductions. In order to maintain the same scale in each space, the dimensions of each were normalized to be in the range $[-0.5, 0.5]$. This process is outlined in Figure 9.1.

From those resulting reductions, a set of 5 spaces with potentially interesting characteristics was qualitatively hand-selected for evaluation. Three are derived from acoustic features (AF) and two are derived from the human-curated rhythm annotations (HA). Four were parametric (ICA,NMF), and one was non-parametric (t-SNE). The selected reduction spaces were AF→ICA, AF→NMF, AF→t-SNE, HA→ICA, and HA→NMF. The analysis of these spaces will be explained in Section 9.4.

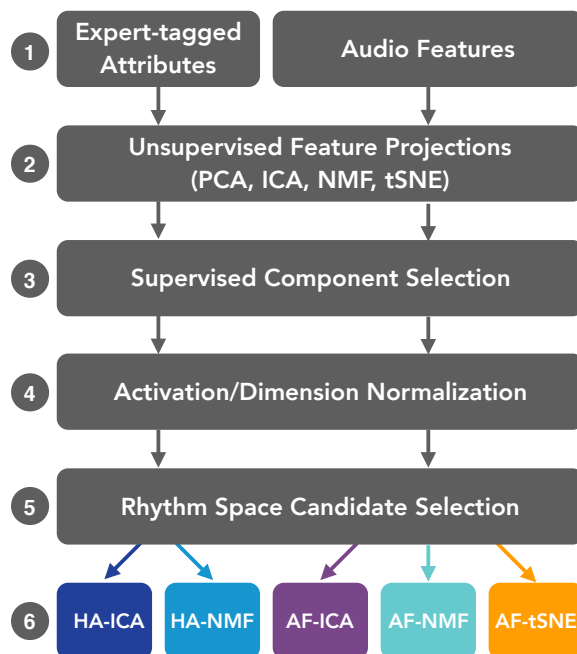


Figure 9.1: Audio features and human annotations are reduced to a set of n components and activations. The 2 most salient components are selected and their activations are normalized to create a 2D space.

9.3 Rhythm Space Evaluation

Based on previous work in visually informative feature spaces [159, 160, 9, 111], the following general conditions must be satisfied:

1. Represent the high dimensional data in an intuitive and organized manner.
2. Generalize to new unseen examples.
3. Offer useful information to other domains.

In order to evaluate each of these conditions, reductions will be trained using a subset of the *MGP* consisting of 50k examples. Each example has expert-annotated attributes for rhythm (Table 9.1) as well as genre, sub-genre, and other geo-cultural factors (language). Because the candidate spaces are both parametric (ICA, NMF) and non-parametric (t-SNE), a non-linear learning model is desired to predict the attributes within the new spaces. For ease of implementation, a *k-Nearest Neighbors* model will be used to both predict rhythm and genre attribute labels in the 2D candidate spaces (conditions 1 and 3) and as well as perform regression in the audio feature domain to place unseen examples into the 2D candidate space (condition 2). It was also shown in other work that k-NN is quite effective for rhythm related tasks, especially when using acoustic features that incorporate the Mellin Scale Transform [155]. For all experiments, the k value was swept in the range [5, 10, 50, 100, 500, 1000]. Through a set of tuning experiments, $k = 500$ was selected for experiments regarding conditions 1 and 3, and $k = 10$ was selected for condition 2.

Due to a large class imbalance among some of the labels in the binary classification tasks, each nearest neighbor was weighted relative to the inverse of its class size. Additionally, for all experiments each neighbor was weighted by the inverse distance to the query example (closer neighbors are weighted higher).

Each of the experiments is run for 5 trials using a randomized 70%:30% (train:test) split. In each of the splits, no artists were shared between training and testing. The same 5 randomized splits were constant across all experiments.

9.3.1 Evaluating Rhythmic Salience

In order to satisfy the first condition, a set of labeling tasks was performed in the candidate projection spaces to evaluate their salience with regards to each of the rhythmic attributes. A *k-Nearest Neighbors* model was trained using the locations of each example in a projection space. An attribute label of an unknown example was then predicted based on the labels of surrounding neighbors. Because some of the rhythm attributes (meter, swing, shuffle, and high syncopation) reduce to representing the presence or absence of a label, they are binarized for simplicity of evaluation when predicting them. Back-beat strength, dancability, and tempo remain continuous ratings.

9.3.2 New Example Prediction

To satisfy the second condition, an unseen example must be projected into the new space. Once again I use *k-Nearest Neighbors*. This time, the goal is not to predict a rhythm label, but to evaluate the organization of the space and its ability to generalize to new examples. For this task, a k-NN model was trained in the the audio feature domain. A test example is then projected into each dimension of the 2D rhythm candidate space based on its proximity to neighbors in the audio feature space. Treating the candidate space as ground truth, evaluation is performed by computing the distances of the regressed test points to their original locations in the 2D candidate space (euclidean distance).

9.3.3 Learning in Other Domains using Rhythm

To satisfy the third condition, I expand beyond rhythmic attributes and explore a selected subset of 12 ‘Basic’ genres, 12 additional Jazz sub-genres, and 14 language labels within the rhythm spaces. ‘Basic’ genre is assembled as a mix of very expansive genres (e.g., Rock, Jazz) as well as some more focused ones (e.g., Disco and Bluegrass), serving as an analog to many previous genre experiments in MIR. The presence of a genre is notated independently for each song by a binary label.

These transfer learning inspired experiments are set up similar to those in Section 9.3.1. The *k-Nearest Neighbors* model was trained using the locations of each example in the projection space. The label of a query example was then predicted based on the labels of surrounding neighbors. The organization of *genre* in the rhythm space suggests that learning aspects about rhythm can transfer

and be applied to learning genre. The efficacy of each rhythm space at predicting genre further supports work in [157] which argues that musical attributes are important when defining genre, and rhythm plays an relevant role.

9.4 Results / Discussion

9.4.1 Exploring the Rhythm Spaces

In order to explore the meaning of these spaces, we can look deeper into the selected components for each of the parametric reductions (ICA, NMF). For t-SNE, we can explore the space’s relationship to the original feature domain through candidate points and their nearest neighbors. While t-SNE is a non-parametric projection, it is designed to maintain local feature similarity in both the high dimensional space and low dimensional projection. By visualizing the mean of *t-SNE* neighbors in the context of the original feature space, we can gain an understanding of the projection’s local structures which should be maintained in both projections [117]. In Figure 9.2 I show the selected components for the parametric projections. In Figure 9.3, I show a set of query points in the t-SNE space (notated with letters) and a two local neighbor means of those query points in the audio feature space.

By viewing the components in Figures 9.2 and 9.3, we can infer a set of observations regarding the meaning of the dimensions and spatial relationships in each candidate space. I will explore each dimension through its ability to capture micro Tatum-level (pulse felt within beats) and macro meter-level (broader) structures. *HA-ICA* captures differences in groupings of 3. Triple Tatum-level attributes (compound-duple/swing) are contained in activations of dimension 1 and triple meter-level attributes are contained in activations of dimension 2. *HA-NMF* highlights differences in danceable, duple time in dimension 1 vs. triple and odd-time in dimension 2. *AF-ICA* highlights the presence of additional Tatum information between the the beats and 8th notes in dimension 1 vs. the presence of clearer triplet figures in dimension 2. *AF-NMF* very clearly places the activation of 8th and 16th note figures against triplet figures. *AF-tSNE* is non-parametric, but similar to other reductions in that it is able organize to distinct rhythmic structures. A 16th note pattern can be seen in the neighbors of query point ‘H’ and triplet structures can be seen in the neighbors of query point ‘C’.

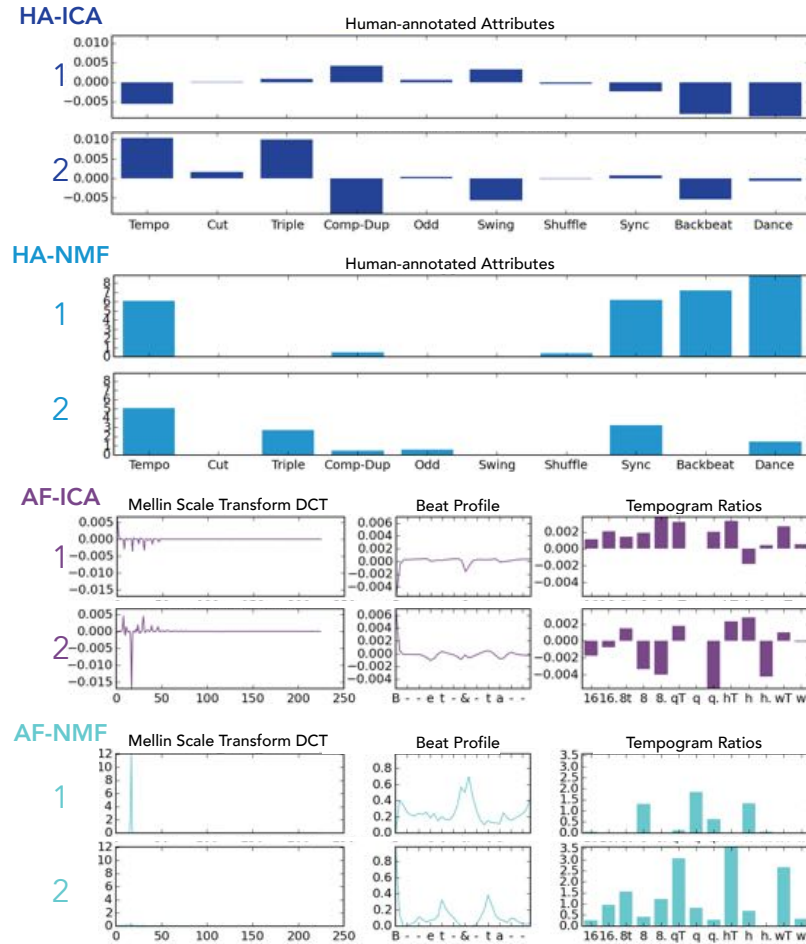


Figure 9.2: Audio features (AF) and human annotations (HA) are reduced to a set of components and activations. Shown here are the two selected components (*supervised component selection*) for the ICA and NMF reductions.

9.4.2 Evaluating the Rhythm Spaces

In the first experiment (Section 9.3.1), I attempt to predict the presence of rhythmic attribute labels through a *k-Nearest Neighbors* model trained using the candidate space projections. Binary attributes (*cut-time meter*, *triple meter*, *compound-duple meter*, *odd meter*, high *syncopation*, *swing*, and *shuffle*) are evaluated using the area under the receiver operator characteristic curve (AUC). Continuous attributes (*tempo*, *back-beat*, *danceability*) are evaluated with the R^2 metric and the mean absolute error (MAE). These results for each of the candidate spaces are shown in Figures 9.4 and 9.5.

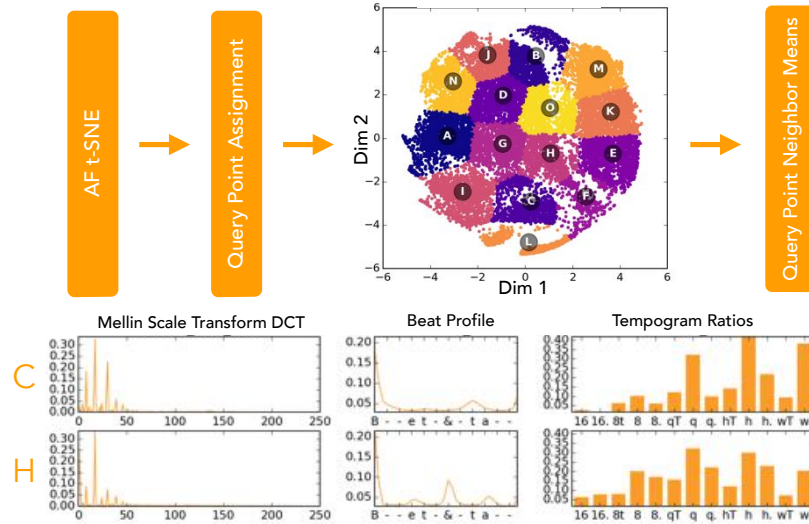


Figure 9.3: To visualize the audio feature information in the t-SNE space, a set of query points (A,B,C, etc.) is selected. Local structures can be explored by looking at audio features means of query point neighbors in the t-SNE space.

With respect to the the binary rhythm attributes, the HA-ICA space is best at representing meter in music. It has the highest AUC in three out of the four meter attributes (cut-time, triple, compound-duple, odd). Because each of the ICA components by design are assumed to be independent, it is likely that this reduction targeted the naturally occurring independence of meter labels (songs usually have a single meter). While the reductions learned from human annotations outperform the acoustic feature reductions in 6 out of the 7 binary classification tasks, the acoustic feature reductions do capture distinguishing information. They perform well in tasks where distinct Tatum-level information, which they have the power to represent, is necessary (compound-duple, swing, shuffle). Furthermore, in shuffle classification, all acoustic feature reductions outperform the human-tagged attribute reductions. For the continuous attributes, the reductions learned from human annotations once again outperform the acoustic feature reductions. Overall these differences are to be expected, as the human-derived attribute spaces are being evaluated on a slightly modified version (some were binarized) of the attributes they were designed to capture. However, the audio feature reductions, which are mostly unsupervised, are still informative representations.

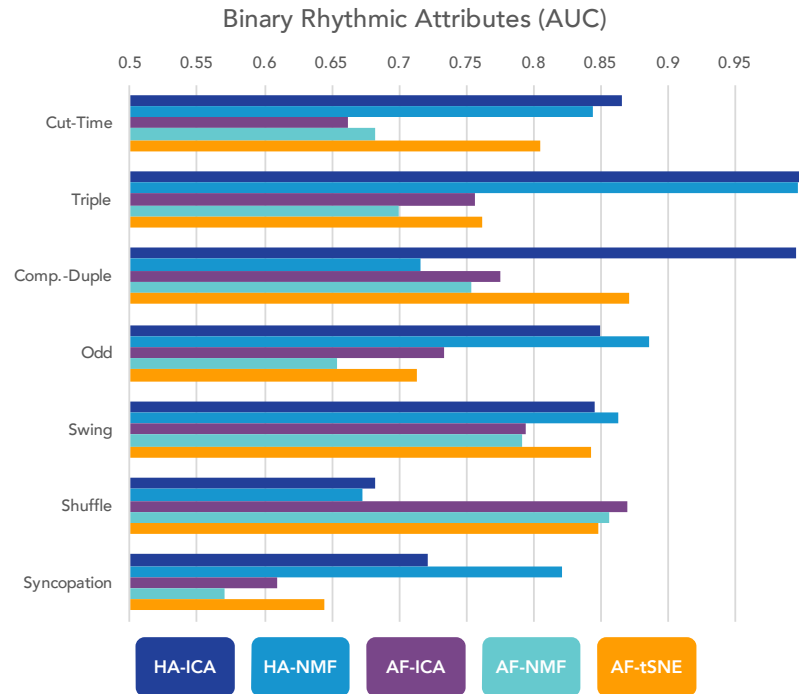


Figure 9.4: Mean AUC of predicting binary rhythm attributes across all trials.

9.4.3 Evaluating Space Regression

The second set of experiments is designed to test the ability of each rhythm space to generalize to unseen examples. I evaluate this by treating the candidate rhythm space projections as ground truth, performing regression on audio features using *k-Nearest Neighbors* to predict the spatial locations in the rhythm candidate projections, and computing the error. In these experiments, k-NN was used to project new, unseen examples into the rhythm space based on their proximity to examples in the original feature space. I evaluate each space on how well each dimension was captured and predicted through R^2 and MAE. I also evaluate the models in both dimensions simultaneously with euclidean distance MAE. Each space was normalized to be in the range $[-0.5, 0.5]$, so the error is equivalent to distance and is directly comparable between the different space candidates.

In satisfying the regression condition, the audio feature reductions perform better. Because the regression is performed in the audio feature domain, it is expected that the audio-feature reductions are more easily learned. AF-ICA performs the best out of the audio feature reductions with a total

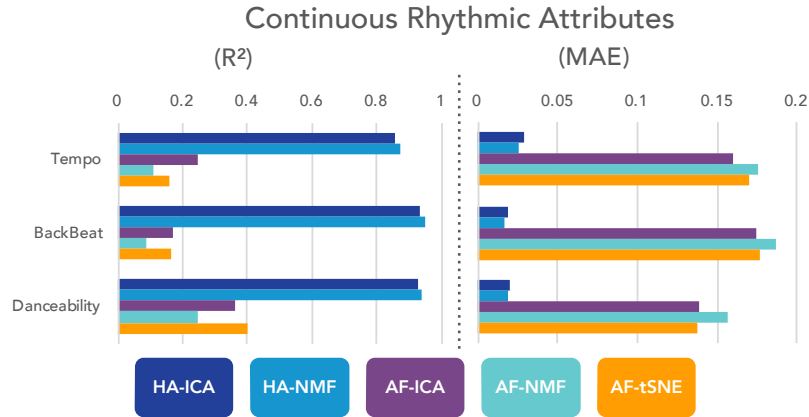


Figure 9.5: Mean R^2 and MAE of predicting continuous rhythm attributes across all trials.

Space	Dim 1		Dim 2		Total MAE
	R^2	MAE	R^2	MAE	
HA-ICA	0.538	0.079	0.251	0.063	0.113
HA-NMF	0.494	0.115	0.314	0.092	0.160
AF-ICA	0.932	0.024	0.848	0.039	0.050
AF-NMF	0.810	0.060	0.900	0.020	0.067
AF-tSNE	0.892	0.051	0.883	0.057	0.085

Table 9.2: Mean R^2 and MAE of projecting into the new rhythm spaces ($k = 10$).

MAE of 0.05. This means that the regressed locations are, on average, accurate to within $\pm 5\%$ of the space. The human-annotation derived candidate spaces, while having a higher error, are still able to be regressed from audio features, showing that audio features are able to capture differences in that new reduced human-inspired label space.

9.4.4 Evaluating the Space in Other Domains

In the third and final set of experiments, I evaluate each space similar to Section 9.4.2. To discover evidence of transfer learning, I evaluate the classification of “Basic” genre, “Jazz” sub-genre, and geo-cultural factors (“language”). Predicting each genre and language label is a binary classification task. These tasks are evaluated using AUC in Figure 9.6.

Once again, the reductions learned from human-annotated attributes performed the best when discriminating most genre labels. However in a few cases, the audio-feature reductions are effective as well, sometimes outperforming the reductions from the human annotations. For example, rhythm spaces derived from audio features are able to best discriminate both Blues and Reggae. While

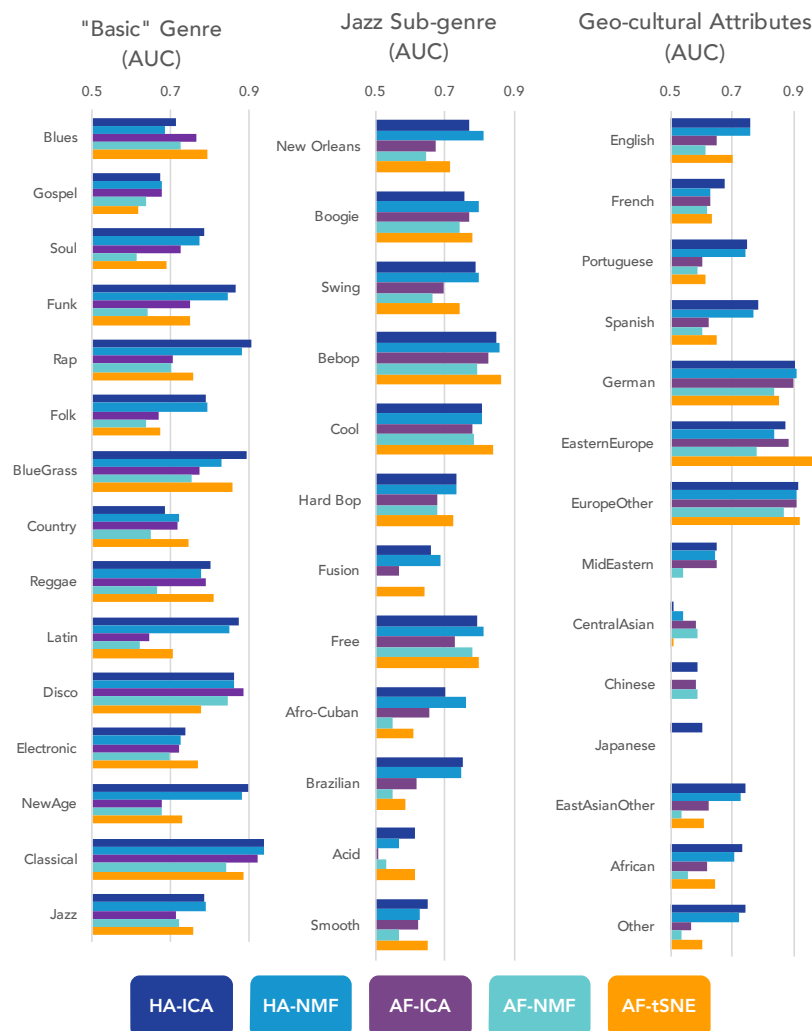


Figure 9.6: Experimental results for classifying “Basic” genre, Jazz subgenre, and geo-cultural factors using spaces designed to represent rhythmic attributes.

seemingly different, the both contain the defining Tatum-level characteristic of shuffle. But because they are two different styles, it is possible they express it differently. It was seen in Figure 9.2 that audio feature reductions are able capture Tatum-level characteristics and seen in Figure 9.4 that they are the best at discriminating shuffle. This suggests that using spaces that capture Tatum-level characteristics may be informative when representing genre. Conversely, a single absence/presence annotation for these characteristics is not designed to capture this difference, and may be blind to genre context in this case.

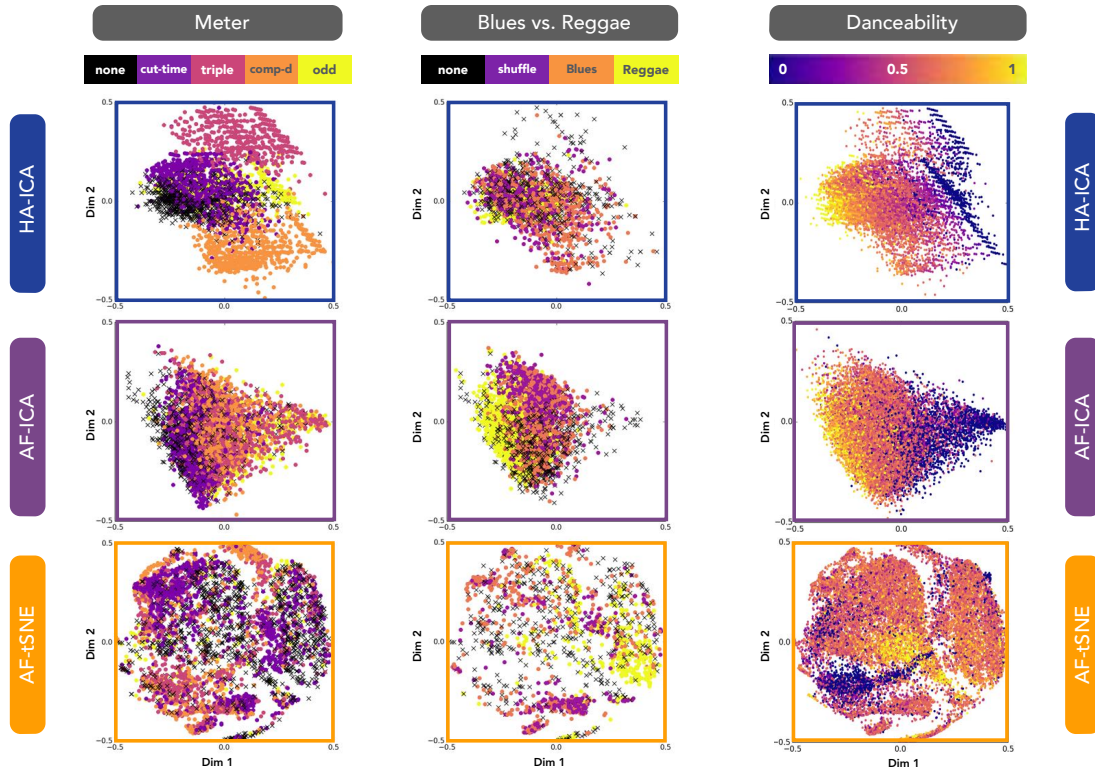


Figure 9.7: A selection of rhythm and genre labels for the HA-ICA, AF-ICA, and AF-tSNE spaces.

We can also perform a more qualitative evaluation of the rhythm spaces. In Figure 9.7, we take a closer look at the HA-ICA, AF-ICA and AF-tSNE reductions. The first column shows a collection of all meter labels organized jointly in each space along with a ‘none’ label, which can be interpreted as simple-duple (defined in Table 9.1). From these meter plots, it becomes clear why HA-ICA was the most successful at predicting meter attributes. While not as separable as HA-ICA, similar meters do cluster in the AF-ICA and AF-tSNE spaces as well. In the second column, we look at the examples where shuffle is present, along with examples of Blues, Reggae, and ‘none’. ‘None’ in this context means, not Blues, not Reggae, and not shuffle. In this set, HA-ICA is not able to capture differences in the styles, while AF-ICA and AF-tSNE are able to discriminate between them to some degree. Because the audio features can also capture meter, this suggests the audio features’ ability to capture similar attributes in different contexts. In the last column, we explore danceability ratings for each example in the rhythm spaces. Comparing columns 2 and 3 for the

audio feature reductions (AF-ICA and AF-tSNE), we can see that in areas where Reggae is present, there are moderately high danceability ratings. Blues areas have lower danceability ratings. More space reduction examples are found in Appendix B.

9.5 Conclusions

In this work, I showed the viability of a low-dimensional representation of rhythm in music. Across each of the three criteria outlined in Section 9.3, the HA-ICA space performed best. This suggests that a space generated from human-tagged attributes is best able to represent discriminative differences in not only rhythmic attributes but other attributes that humans find useful for music categorization. The audio feature reductions are also viable spaces. While less accurate at direct discrimination of attributes, they are easier to learn through regression, making them scalable and potentially better at representing an unknown example in a more consistent manner. There is also evidence that the audio reductions are better able to capture contextual differences between attributes (shuffle in Blues vs. Reggae). I argue that a space that can represent these additional contextual relationships is potentially more useful for musicology and music discovery/playlisting than those that can only discriminate.

In later work, I explored more powerful methods, such as autoencoder networks, to learn reductions with objectives targeted more acutely to the rhythmic attributes. These methods and results are shown in Appendix C.

Chapter 10: Conclusions and Future Directions

In this thesis, I explored rhythmic components and their relationships to each other, to genre, and other geo-cultural factors (i.e., language) through data driven approaches using audio signals. Working in conjunction with *Pandora*[®], I employed a corpus of over 1 million expertly-labeled audio examples across many rhythmic styles and genres from their flagship *Music Genome Project*[®]. Each song is labeled with more than 500 attributes of rhythm, instrumentation, timbre, and genre. This supports the work's scalability to very large datasets and its applicability to real-world problems.

First, a set of rhythm inspired features was developed. They were designed to capture elements of rhythm at both the Tatum-level (micro) and meter-level (macro) time scales. A large-scale set of experiments was then performed to quantify and label a set of rhythmic meter and feel attributes using the *Pandora*[®] *Music Genome Project*[®]. In later work, more complex, scalable models employing Random Forests, Gradient Boosted Trees, and hybrid tree ensembles were evaluated. Similar to neural-network models, tree ensembles benefit from the ability to learn complex, non-linear mappings of the data. It was found that the tree ensemble methods are better than linear methods when modeling the complexities of rhythmic attributes. From a musicology perspective, these rhythmic attributes are important in the makeup of a musical style. From this work, insight is gained into the meanings of rhythmic features as they relate to meter and feel when applying them to style recognition tasks.

Second, it was demonstrated that there is potential to demystify the constructs of musical genre into distinct musicological components. The attributes selected from music experts are able to provide a great deal of genre distinguishing information. I was also able to discover and outline the importance of certain attributes in specific contexts. This strongly suggests that the expression of musical attributes are necessary additions to definitions of genre. It was also shown that audio features motivated by timbre and rhythm are, with some success, able to model musical attributes. Audio features are also able to describe musical genre directly and through stacked approaches that

exploit the learned models of musical attributes. This is strong evidence suggesting that audio-based approaches are learning the presence of the musical attributes, to some degree, when distinguishing genre. In some cases, the audio-based models were more powerful than the human musical attribute models. This suggests that there is more to genre than the chosen subset of rhythm and orchestration attributes, and prompts that there is more about the definition of genre yet to be discovered.

In seeking to improve on this work, it may be necessary to investigate late fusion of context-dependent classifiers (e.g., rhythm, timbre), which has shown improved results for genre classification [136]. It may also be helpful to use a greater number of the available attributes than the chosen 48, as well as additional attribute types (e.g., melody, harmony). Furthermore, perhaps the most interesting direction is to treat each musical attribute model as a hidden layer in a neural network. In these cases, the models that are trained to predict musicological attributes will serve as a form of domain-specific pre-training. Using deep models allows for back-propagation across an additional layer which connects our attributes to genres. This will potentially help to learn better models of genre as well as adjust the models of musical attributes in order better capture their genre relationships.

Finally, I showed the viability of a low-dimensional representation of rhythm in music. This work suggested that spaces generated from human-tagged rhythm attributes are best able to represent discriminative differences in not only rhythmic attributes but other attributes that humans find useful for music categorization. The audio feature reductions are also viable spaces. While less accurate at direct discrimination of attributes, they are easier to learn through audio feature regression, making them potentially better at representing an unknown example in a more consistent manner. There is also evidence that the audio reductions are better able to capture contextual differences between attributes (i.e., shuffle in Blues vs. Reggae). I argue that a space that can represent these additional contextual relationships is potentially more useful for musicology and music discovery/playlisting than those that can only discriminate.

In seeking to improve the feature spaces, more powerful methods, such as autoencoder networks, can be used to learn reductions with objectives targeted more acutely to the rhythmic attributes.

This was investigated briefly in Appendix C, but more work is beyond the scope of this thesis. Furthermore, it may be interesting investigate creating a genre space using timbre, harmony, and rhythm features along with genre labels, and explore other musicological relationships. Furthermore, because the applicability of each of these spaces is human-focused, it may be necessary to create evaluations of each of the spaces through a set of listening tests and human feedback.

Appendix A: The Mellin Scale Transform

The *Mellin Scale Transform* is a scale invariant transform of a time domain signal. Similar musical patterns at different tempos are scaled relative to the tempo. The *Mellin Scale Transform* is a scale invariant (meaning tempo invariant) transform that captures periodicity at multiple metrical levels simultaneously. It was introduced in the context of rhythmic similarity by Holzapfel [60]. It was introduced in this thesis in Chapter 6.2.5. In that Chapter, the discrete form of its computation was introduced. In this appendix, I will give a brief overview of its relation to the Fourier Transform. The Fourier Transform is shown in Equation A.1. The Mellin Scale Transform is shown in Equation A.2. Notice that they are both time domain signals multiplied by a time/frequency exponential. The Mellin Scale Transform is the Fourier Transform of an exponentially sampled time-domain signal scaled by an exponential time weighting window. In the Mellin Scale Transform, c takes the place of ω .

$$X(\omega) = \mathcal{F}\{x(t)\} = \int_{-\infty}^{\infty} x(t)e^{-j\omega t} dt \quad (\text{A.1})$$

$$R(c) = \mathcal{F}\{r(e^\tau)e^{\frac{1}{2}\tau}\} = \int_0^{\infty} r(e^\tau)e^{\frac{1}{2}\tau}e^{-jc\tau} d\tau \quad (\text{A.2})$$

In order to dive deeper, I will provide an overview of its computation in relation to a few music examples. The computation first starts with the accent signal. As stated previously in Chapter 6.2.5, the Mellin Scale transform is not shift invariant so the autocorrelation $r(\tau)$ of the accent signal must be used for input. This autocorrelation is then exponentially sampled $r(e^\tau)$. This exponential sampling warps rhythmic structures at different time scales from exponential relationships to approximately linear relationships. This makes large lag periodicity more linearly related to short lag periodicity. This process is shown in Figure A.1.

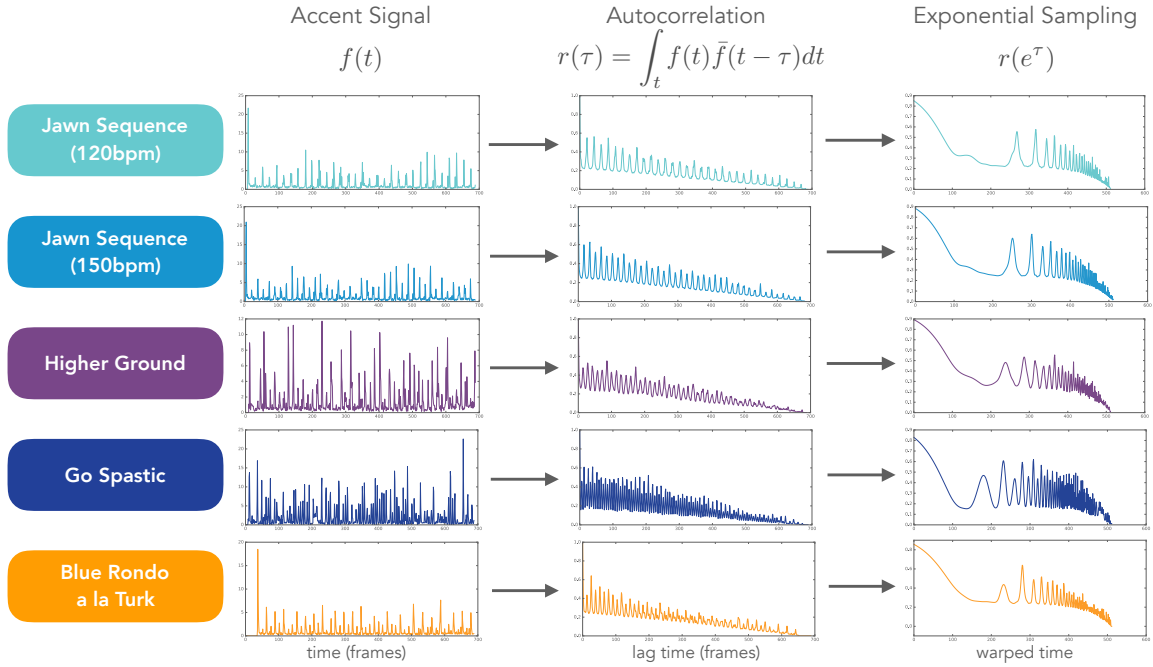


Figure A.1: The accent signal, its autocorrelation, and exponential sampling.

The autocorrelation now is exponentially decreasing so we can emphasize long-scale periodicity through an exponential weighting. This is shown in Figure A.2.

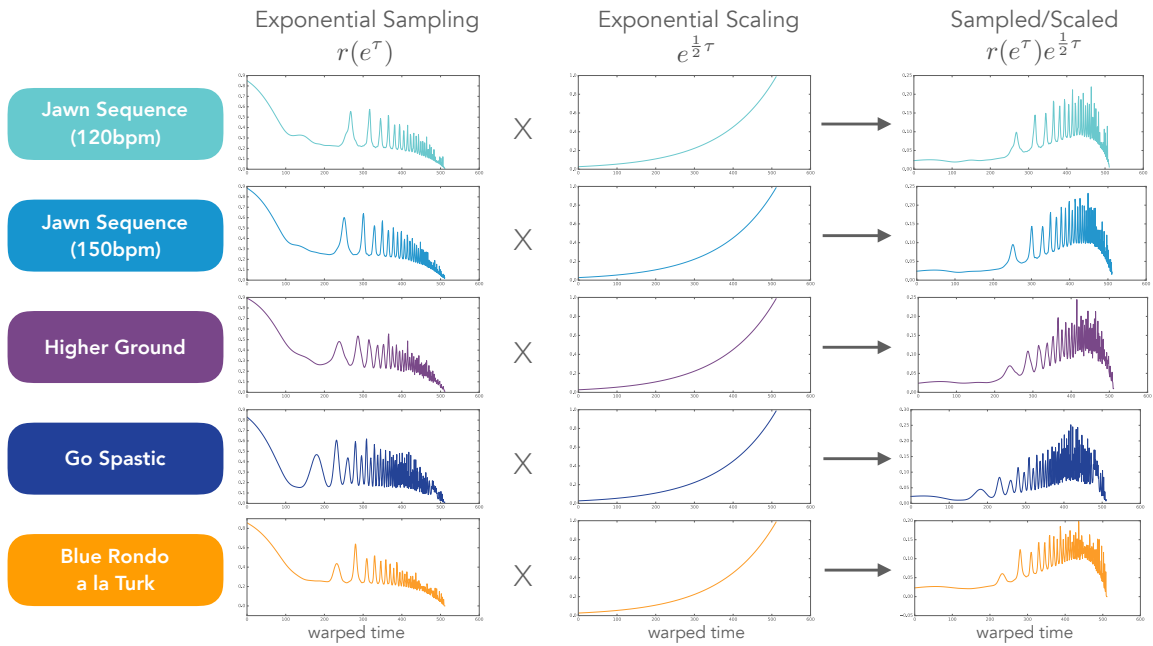


Figure A.2: The exponential weighting.

The Mellin Scale Transform is the Fourier Transform of this resulting signal from Figure A.2. The Mellin Scale frequencies and harmonics are related to how short-time tatum-level patterns relate to large-scale meter-level patterns. Harmonics in the transform can be interpreted as the periodic consistency of the “fractal-like” relationships of rhythmic patterns at different time scales (Tatums, Beats, and Meter). The Mellin Scale Transform is shown in the first column of Figure A.3.

There is a general periodicity in the Mellin Transform, similar to harmonics in a standard Fourier Transform. These periodic structures can be exploited to create a more sparse, cepstral-like version using the Discrete Cosine Transform (DCT). We now see a set of sparse peaks in the DCT. In order to remove the massive DC component in the Mellin Transform (0^{th} scale coefficient), we can also perform a simple peak picking by local median removal and half-wave rectification. This process is shown in the second and third columns of Figure A.3.

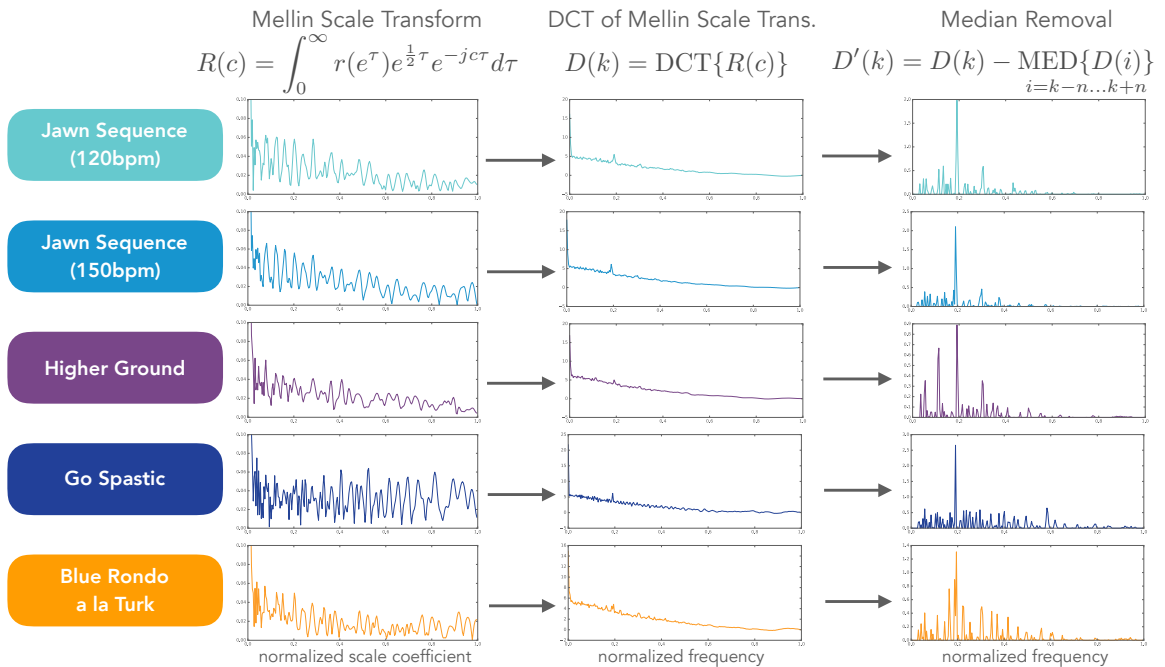


Figure A.3: The Mellin Scale Transform, its DCT, and the normalized median-removed DCT

Appendix B: Collection of Rhythm Space Reductions

This appendix contains a large number of visualizations of the rhythm spaces introduced in Chapter 9 similar to Figure 9.7. Each section outlines a reduction type, displays the associated space dimension components, and shows a selection of rhythm label and genre label colorings in the space. The genre and rhythm colorings are a selection of the ‘Basic’ Genres and all the rhythm labels from Chapter 3.3. Color mappings for the genre and rhythm labels are shown in Figure B.1.

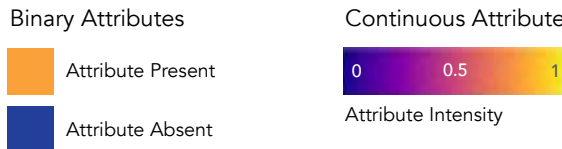


Figure B.1: Color mappings for attribute and genre plots.

In each of the plots, the space is normalized from -0.5 to 0.5 to have uniform area. Because each of the parametric spaces is additive, The point (-0.5, -0.5) in the bottom left corner means there is no emphasis on either component and the point (0.5, 0.5) in the upper right corner has the maximum relative emphasis on both components.

The first two spaces are derived from human annotated attributes of rhythm. The last three are derived from rhythm acoustic features. Four of the spaces (HA-ICA, HA-NMF, AF-ICA, AF-NMF) are derived from parametric reductions with the two visual components found through *supervised component selection* from Chapter 9.2.2. The last method (AF-tSNE) is derived from t-SNE performed on acoustic features.

B.1 Human-tagged Attributes and NMF: HA-NMF

First introduced in Chapter 4.4.1, *Non-Negative Matrix Factorization* (NMF) decomposes a matrix \mathbf{V} into two matrices \mathbf{W} and \mathbf{H} . $\mathbf{V}^{[n \times m]} \approx \mathbf{W}^{[n \times r]} \times \mathbf{H}^{[r \times m]}$, where n is the number of examples, m is the number feature dimensions, and r is the number of basis components to be learned.

Figure B.2 shows the component bases from NMF and *supervised component selection* on human rhythm attribute annotations. The spaces in Figures B.3 and B.4 show the rhythm and genre labels based on the activations of each of the selected component dimensions.

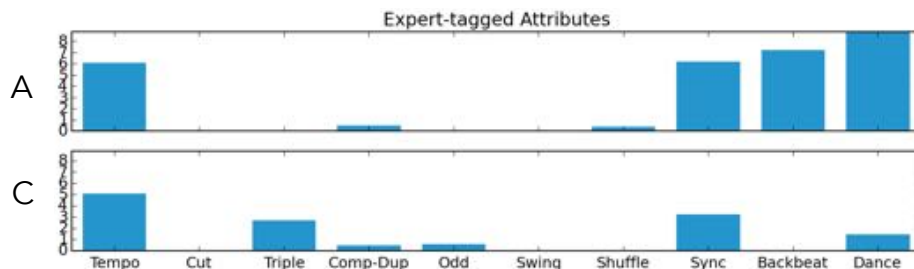


Figure B.2: Components for NMF reductions derived from human annotations of rhythm.

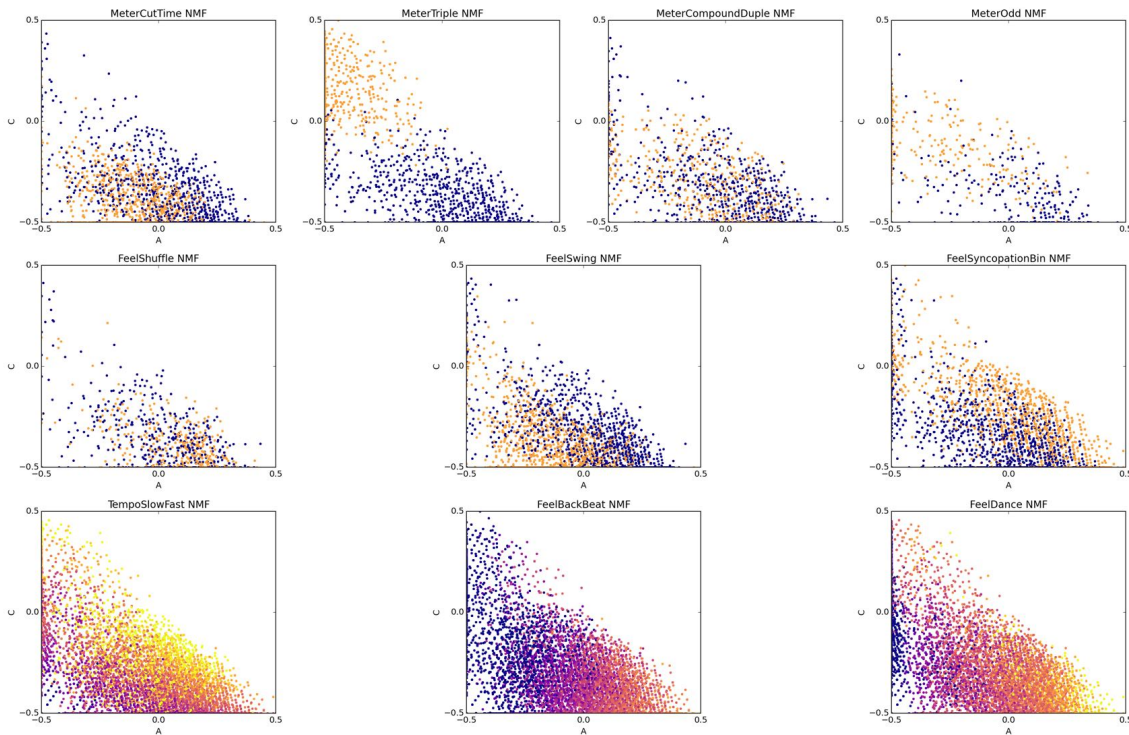


Figure B.3: Rhythm attribute colorings for NMF reductions derived from human annotations of rhythm.

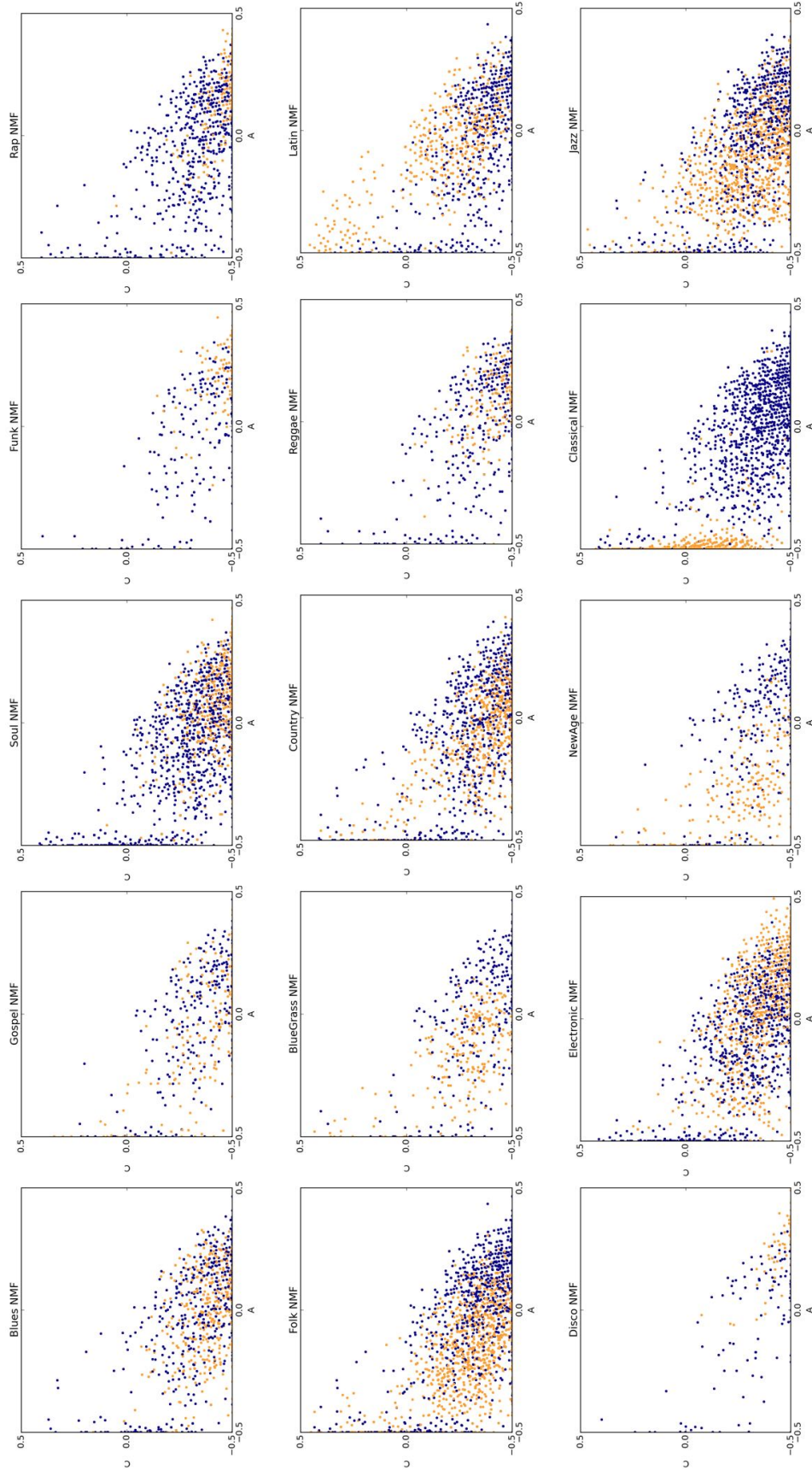


Figure B.4: Genre colorings for NMF reductions derived from human annotations of rhythm.

B.2 Human-tagged Attributes and ICA: HA-ICA

Independent Components Analysis (ICA) considers higher-order statistics more than the 1st and 2nd moments (expectation, variance) to minimize mutual information of the output. ICA creates a set of independent components of non-Gaussian signals or features. Each of the resulting dimensions do not need to be orthogonal [129]. Figure B.5 shows the component bases from ICA and *supervised component selection* on human rhythm attribute annotations. The spaces in Figures B.6 and B.7 show the rhythm and genre labels based on the activations of each of the selected component dimensions.

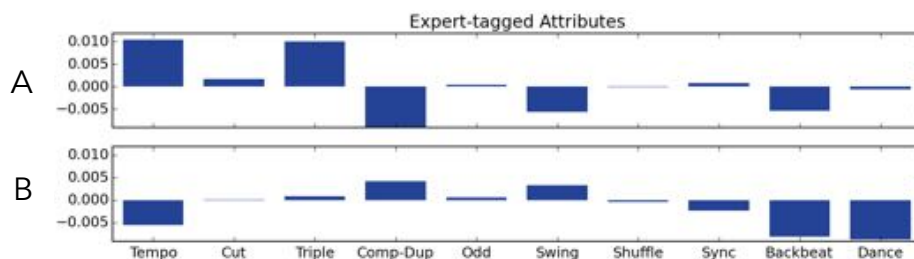


Figure B.5: Components for ICA reductions derived from human annotations of rhythm.

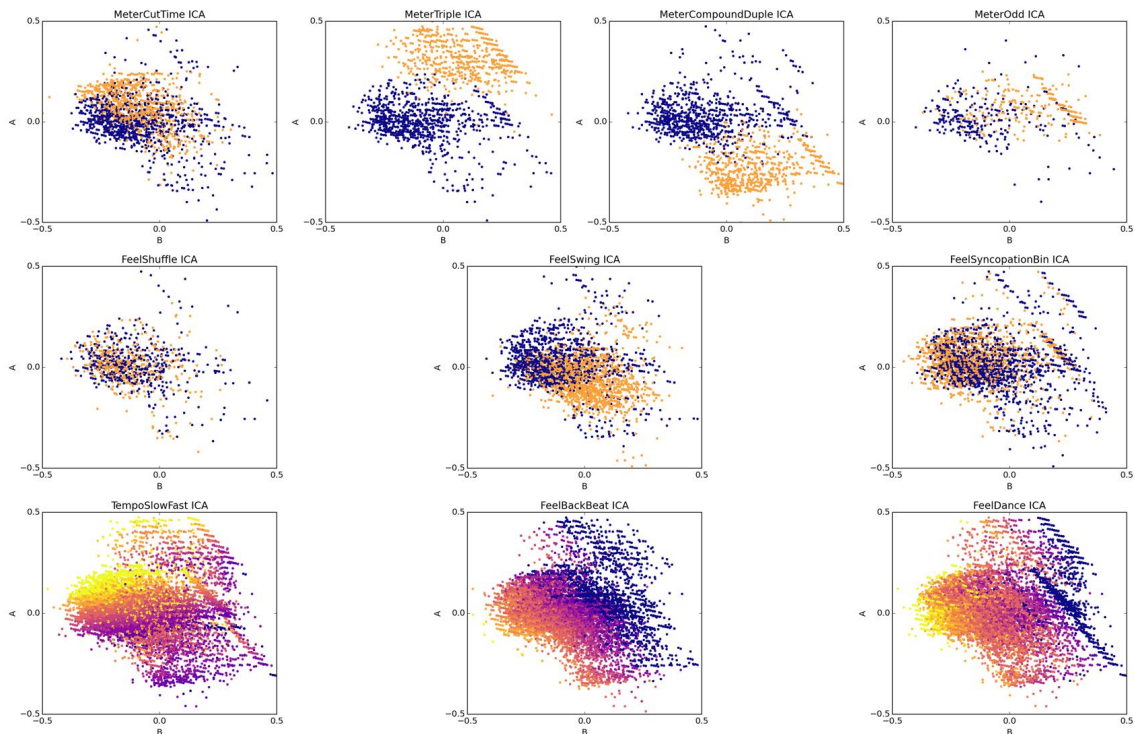


Figure B.6: Rhythm attribute colorings for ICA reductions derived from human annotations of rhythm.

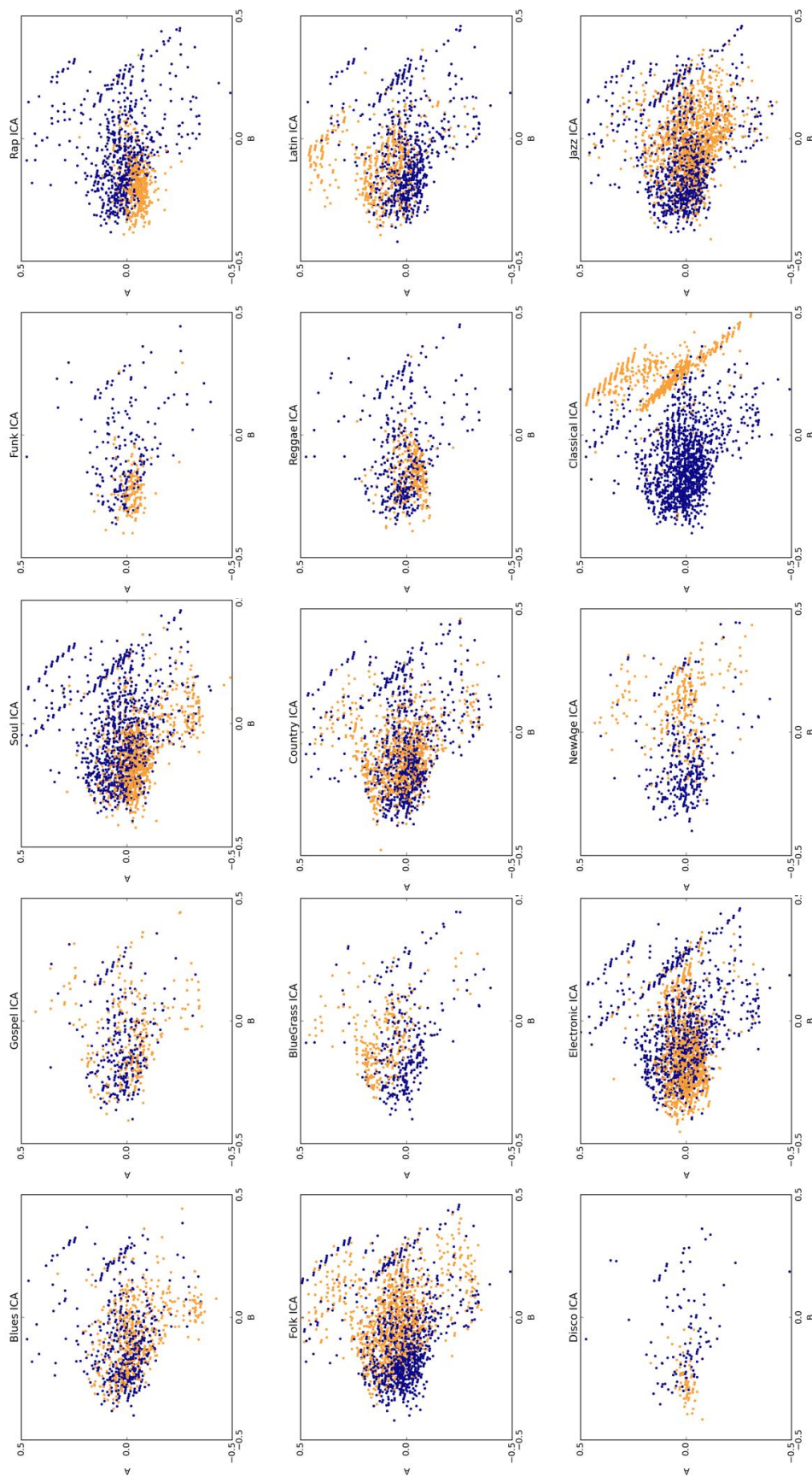


Figure B.7: Genre colorings for ICA reductions derived from human annotations of rhythm.

B.3 Rhythm Audio Features and NMF: AF-NMF

First introduced in Chapter 4.4.1, *Non-Negative Matrix Factorization* (NMF) decomposes a matrix \mathbf{V} into two matrices \mathbf{W} and \mathbf{H} . $\mathbf{V}^{[n \times m]} \approx \mathbf{W}^{[n \times r]} \times \mathbf{H}^{[r \times m]}$, where n is the number of examples, m is the number feature dimensions, and r is the number of basis components to be learned.

Figure B.8 shows the component bases from NMF and *supervised component selection* on acoustic features. The spaces in Figures B.9 and B.10 show the rhythm and genre labels based on the activations of each of the selected component dimensions.

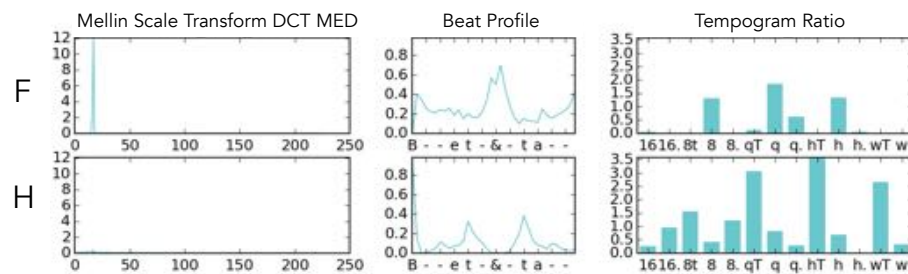


Figure B.8: Components for NMF reductions derived from rhythm acoustic features.

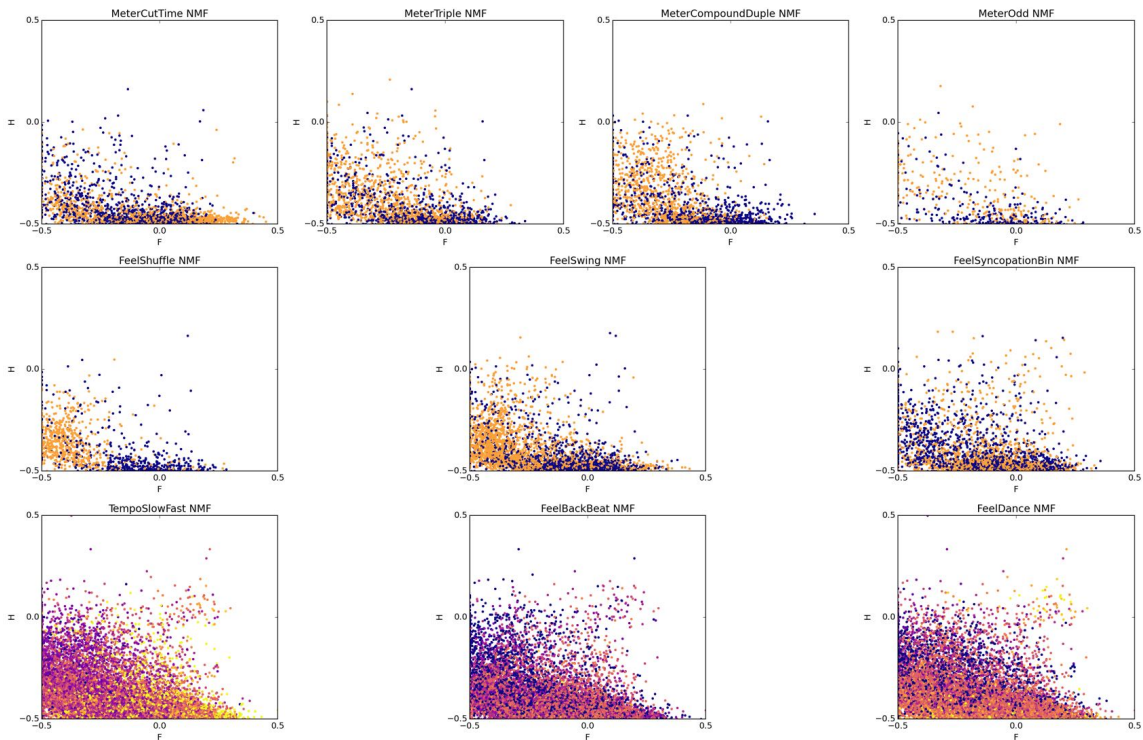


Figure B.9: Rhythm attribute colorings for NMF reductions derived from rhythm acoustic features.

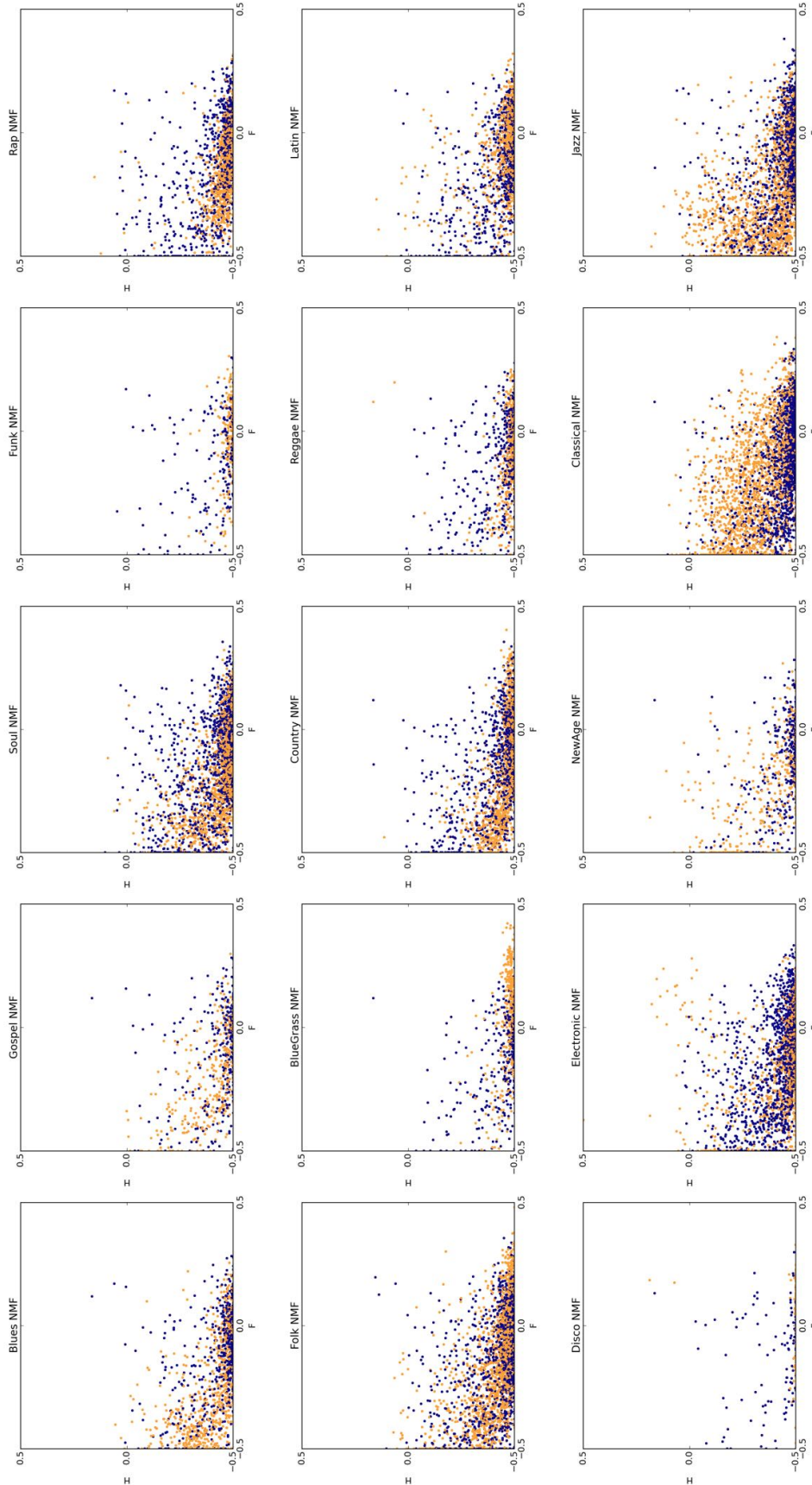


Figure B.10: Genre colorings for NMF reductions derived from rhythm acoustic features.

B.4 Rhythm Audio Features and ICA: AF-ICA

Independent Components Analysis (ICA) considers higher-order statistics more than the 1st and 2nd moments (expectation, variance) to minimize mutual information of the output. ICA creates a set of independent components of non-Gaussian signals or features. Each of the resulting dimensions do not need to be orthogonal [129]. Figure B.11 shows the component bases from ICA and *supervised component selection* on acoustic features. The spaces in Figures B.12 and B.13 show the rhythm and genre labels based on the activations of each of the selected component dimensions.

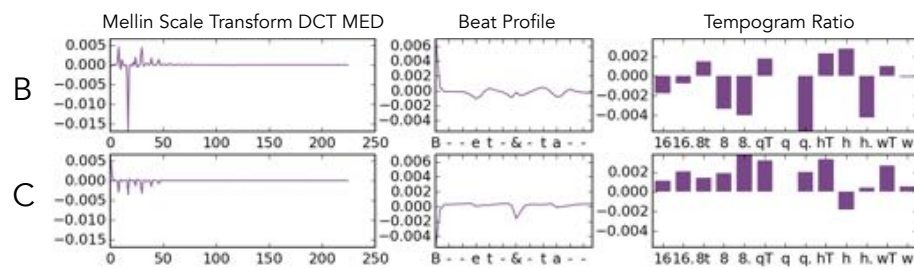


Figure B.11: Components for ICA reductions derived from rhythm acoustic features.

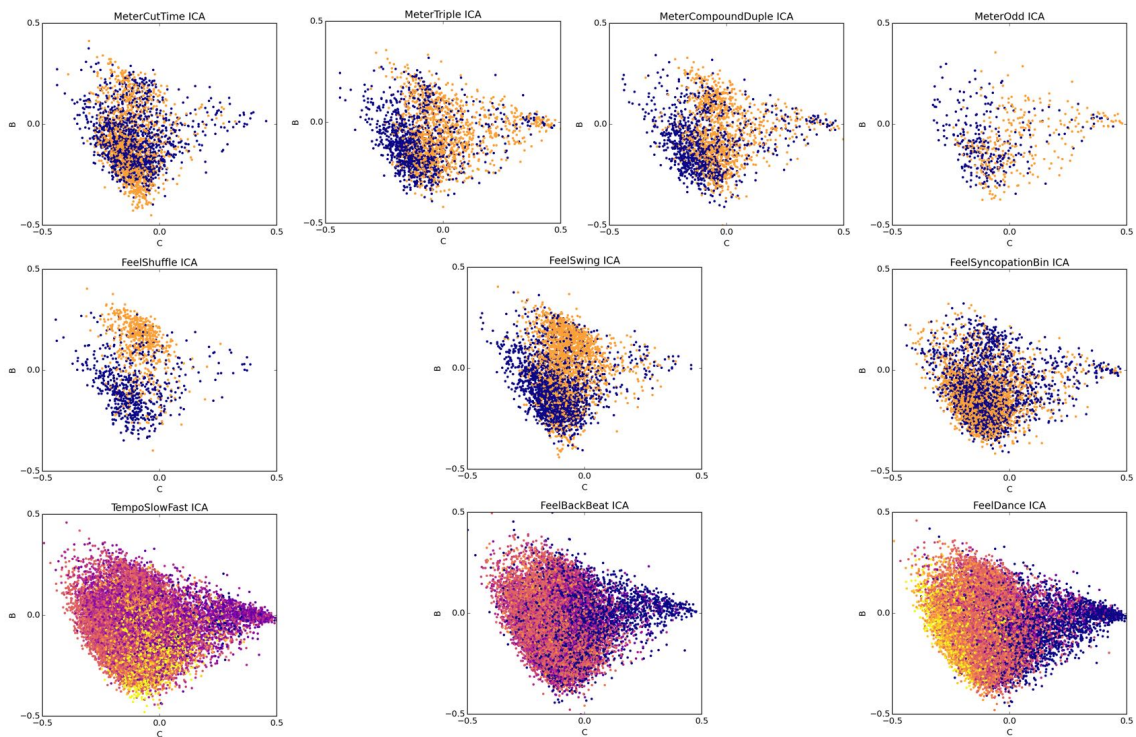


Figure B.12: Rhythm attribute colorings for ICA reductions derived from rhythm acoustic features.

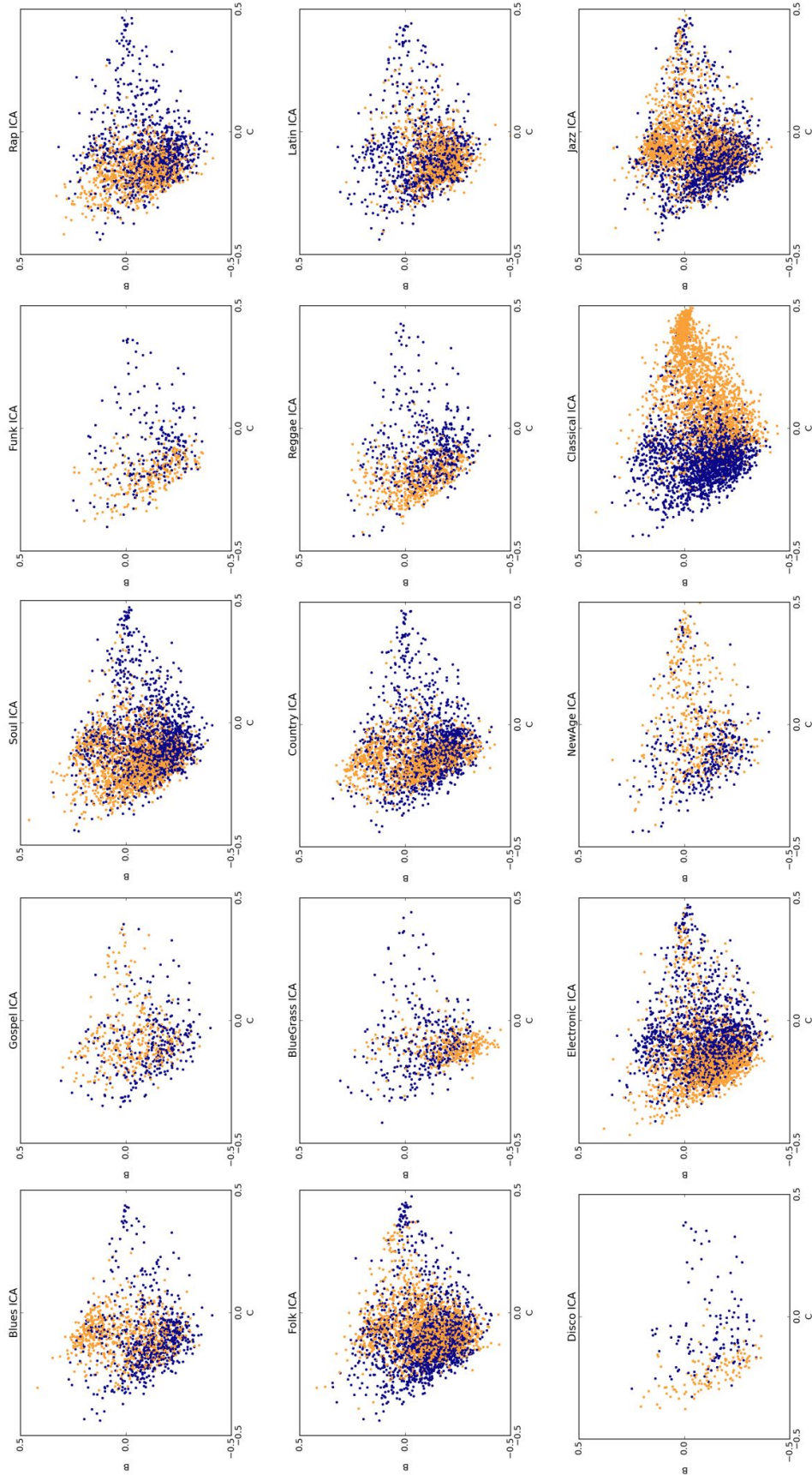


Figure B.13: Genre colorings for ICA reductions derived from rhythm acoustic features.

B.5 Rhythm Audio Features and t-SNE: AF-tSNE

Similar to methods suggested so far, *t-Distributed Stochastic Neighbor Embedding (t-SNE)* attempts to build a map in which high-dimensional relationships are maintained in a lower-dimensional space. It aims to preserve local pairwise relationships, and focus less on large global relationships. This space is explored in relation to the candidate point selection method described in Chapter 4.4.2.

Unlike ICA or NMF, the resulting space from *t-SNE* is non-parametric, meaning it is difficult to interpret what the dimensions mean in terms of the original feature space. The only assumptions that can be made are that points close to each other in the t-SNE space are mapped as such because they were close in the original feature space. In PCA or NMF, it was easy to see how each of the basis components relate to the original features. Through the activations, it is possible to intuit an understanding of the space as it relates to an expression of a component. This is not true of t-SNE. In order to explore the t-SNE space, a clustering inspired method is employed to find a set of candidate points. Figure B.14 shows the space and candidate points. Local means of those candidate points are shown in Figure B.15. The spaces in Figures B.16 and B.17 show the rhythm and genre labels based in the t-SNE space.

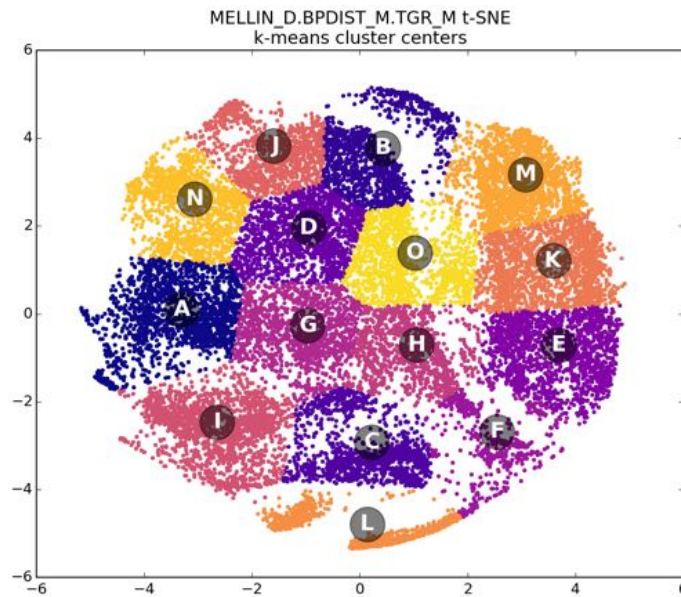


Figure B.14: Local component locations for t-SNE reductions derived from rhythm acoustic features.

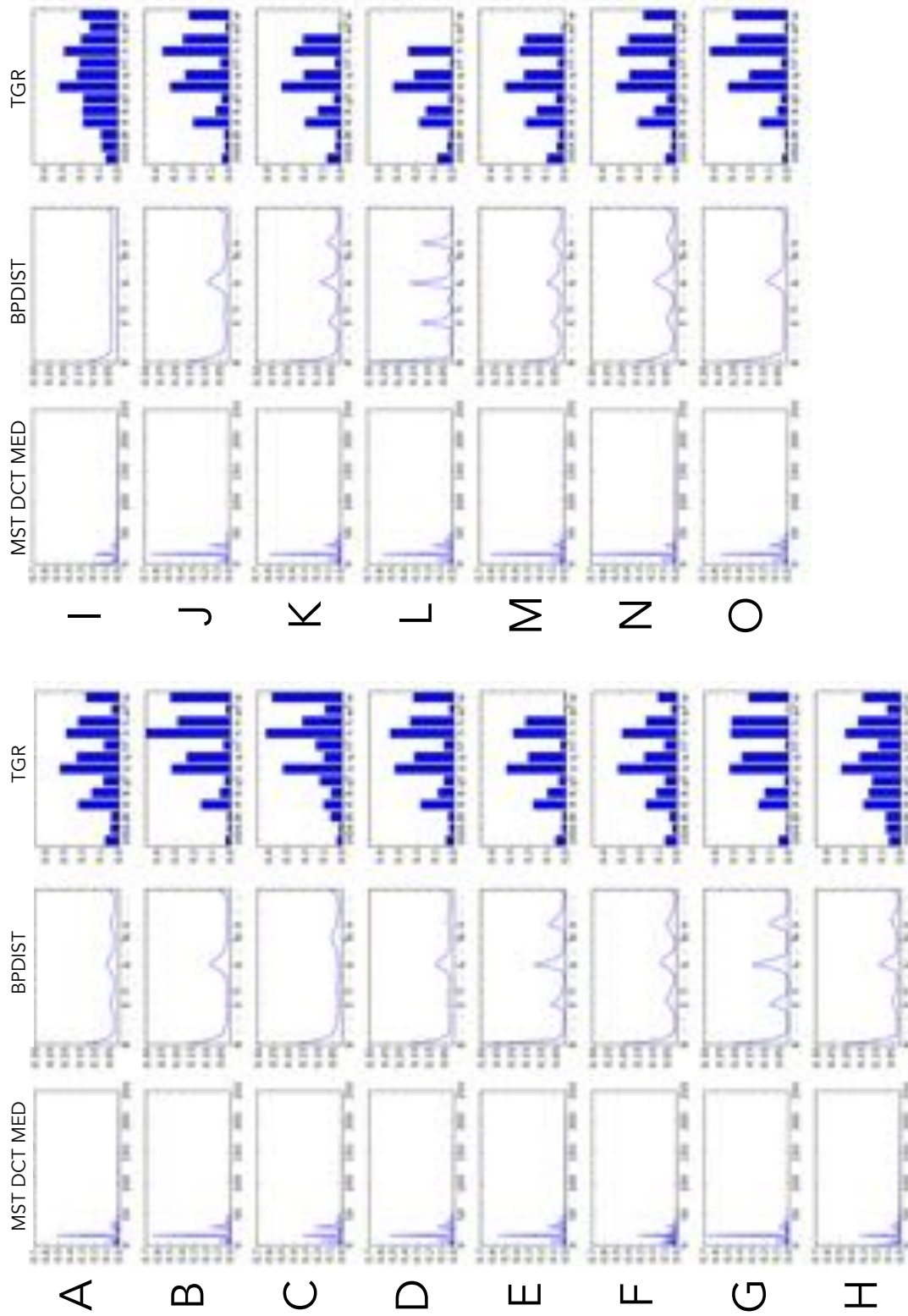


Figure B.15: Components for t-SNE reductions derived from rhythm acoustic features.

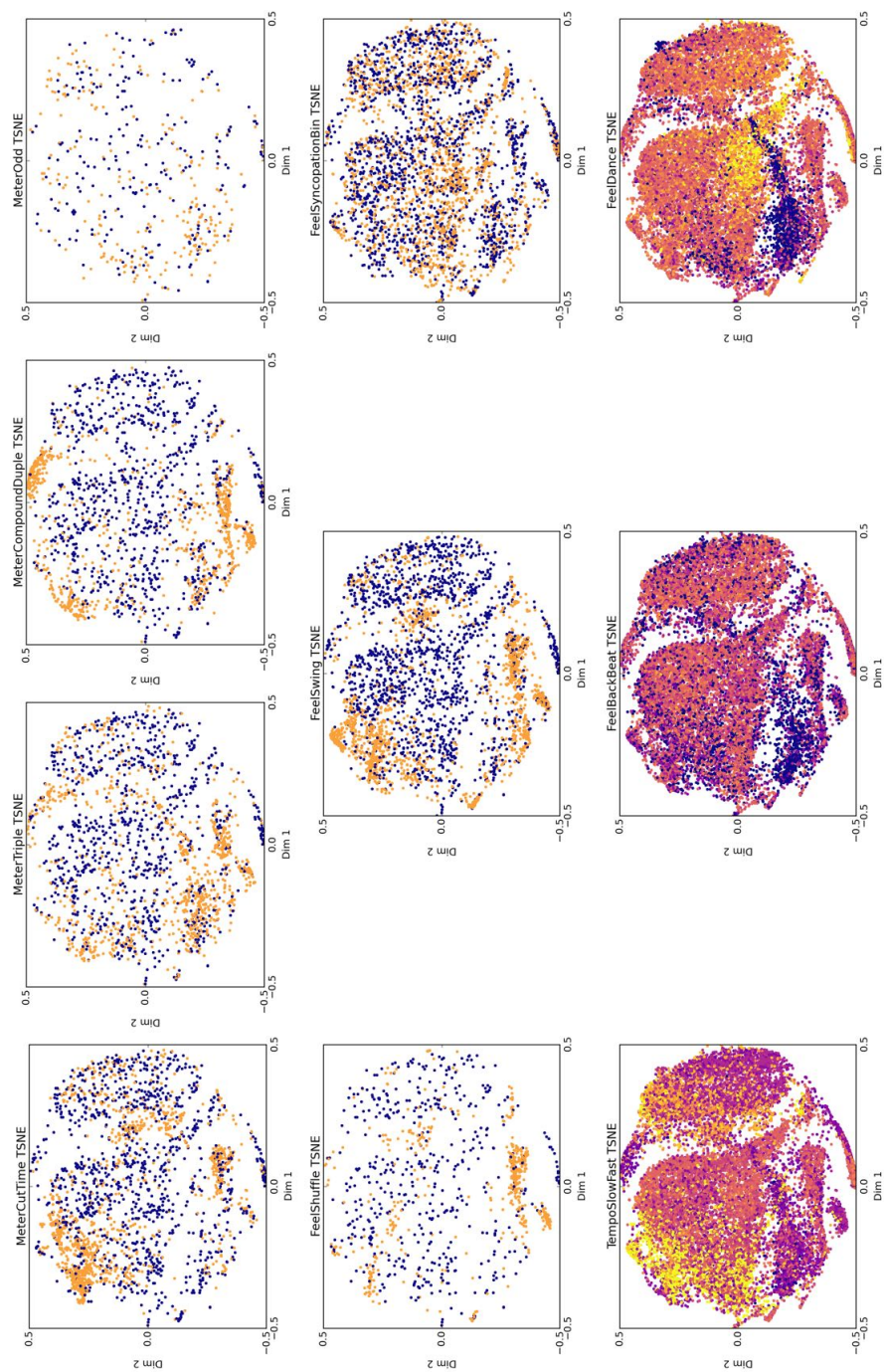


Figure B.16: Rhythm attribute colorings for t-SNE reductions derived from rhythm acoustic features.

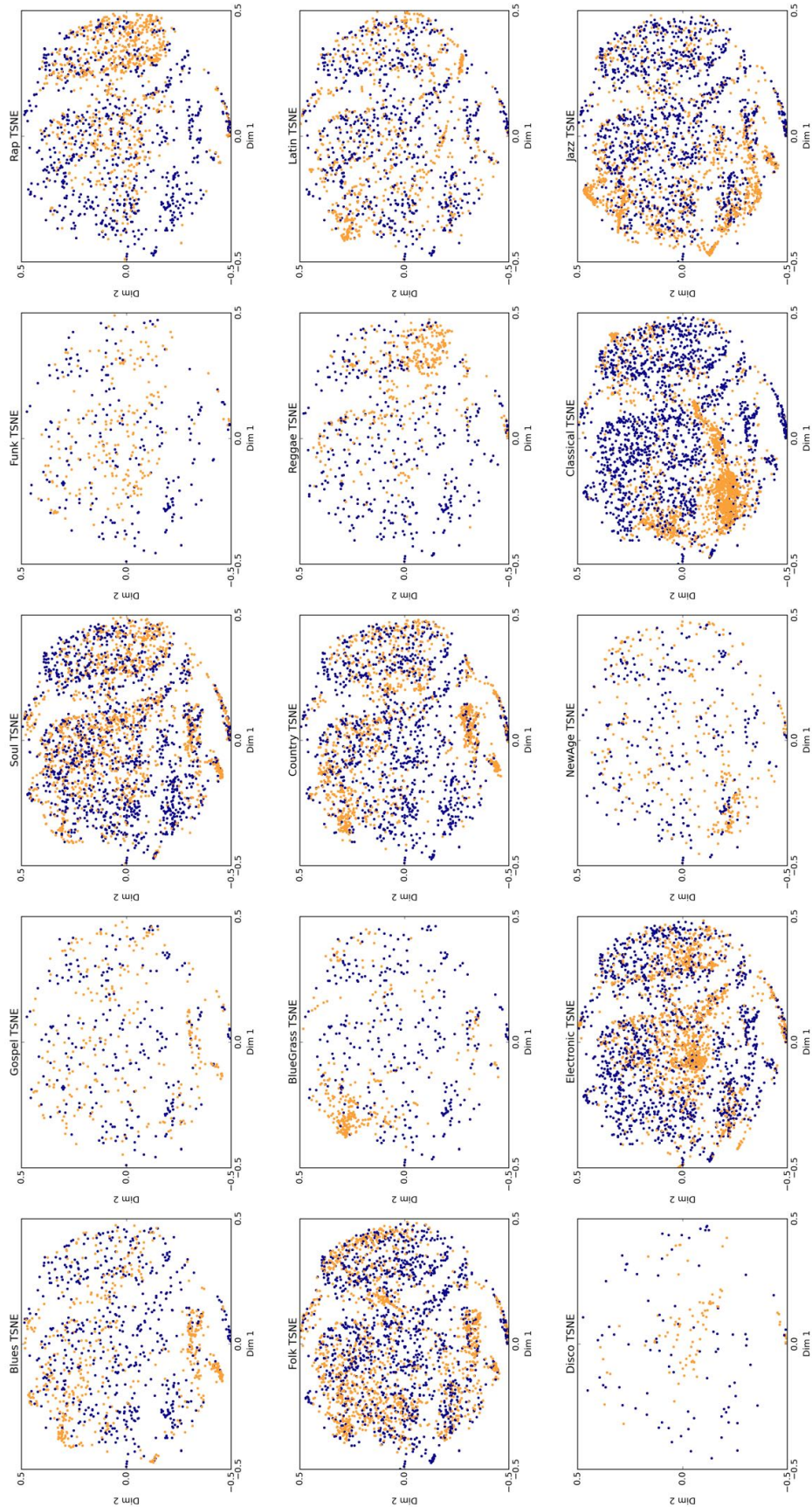


Figure B.17: Genre colorings for t-SNE reductions derived from rhythm acoustic features.

Appendix C: Visualizing Rhythm Attributes in Music Using Stacked Denoising Autoencoders

Different attributes of rhythmic meter and feel combine in complex and creative ways to create cohesive, distinct, and easily recognizable styles. In this work, I attempt to not only learn organizations of these compositional attributes, but understand their overarching relationships by creating a set of visualizations using *stacked denoising autoencoders*. The scope of *Pandora's Music Genome Project* is leveraged to create a set of visual projections from human-annotated attributes and rhythm-inspired audio features.

C.1 Introduction

In this work, I explore representations that capture a variety of rhythmic attributes in a joint and intuitively organized manner using stacked denoising autoencoders. This is the direct extension of previous work in Chapter 9 and Appendix B.

There is a large body of work that has examined the general recognition of rhythmic styles in music audio signals [52, 155], but few efforts have focused on the deconstruction and quantification of the foundational components of global rhythmic structures and how they interact to form a musical style. Previous work has shown that rhythmic components have very important relationships to definitions of style and musical genre [156] and models trained with compact features derived from the audio signal are quite effective when representing rhythm-related attributes of meter and feel (e.g., 'swing') [10, 157]. However, I believe that prediction of these attributes in isolation does not tell the full story. In this work I try to demystify a high-level measure of similarity by defining an organized and interpretable space that is able to jointly represent multiple rhythmic attributes. Furthermore, motivated by work in transfer learning [149, 158], the salience of this representation is explored in other domains as well (i.e., genre).

C.2 Stacked Denoising Autoencoders

An *autoencoder* is a neural network trained to learn a low-dimensional coding that captures distinguishing information about the original feature space [161]. They can also be used to learn a higher-dimensional coding (*contractive*) to de-tangle complex data relationships [162]. In both cases, they are trained to accurately reconstruct the model input using the learned code. A *denoising autoencoder* adds noise or corruption to the original feature input during training to capture a more coherent structure in the hidden layers. In order to learn more complex non-linear relationships, deep networks can be created by stacking autoencoders, creating a *stacked denoising autoencoder* [163].

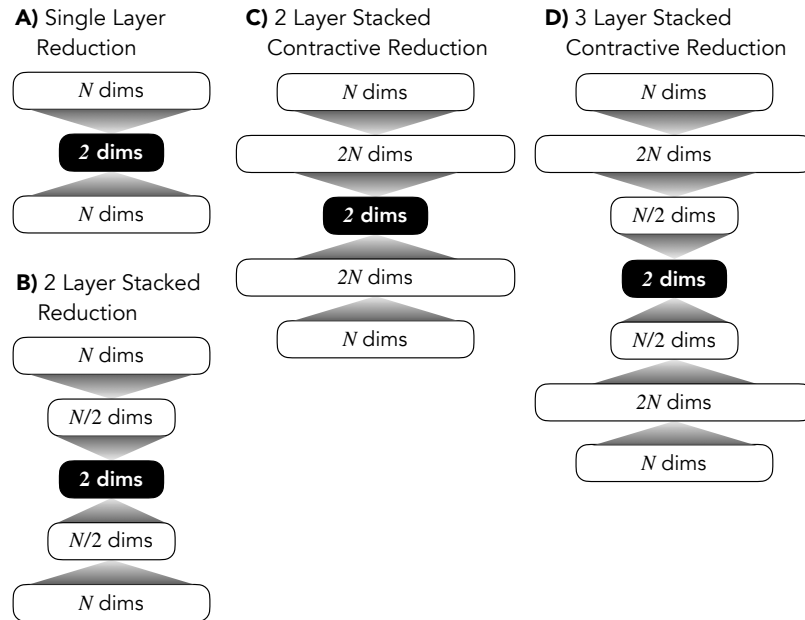


Figure C.1: Overview of stacked autoencoder models. N is the dimensionality of feature input (attributes: $N=10$, audio: $N=372$).

Autoencoders can be used as a visualization tool, as in this work (Figure C.1), by learning a 2-dimensional coding. Each structure is used to reduce both the hand-annotated attributes and the rhythm acoustic features on 50k song examples. The first set of reductions uses a feature vector of 10 hand-annotated rhythm attributes. These attributes are collected by music experts from the *Pandora[®] Music Genome Project[®](MGP)*. The second set of reductions use a rhythm-inspired feature vector (372 dims) computed directly from the audio signal: *Beat Profile*, *Tempogram Ratio*,

Mellin Scale Transform [10]. The organization of the reduced space is quantitatively evaluated through rhythm attribute prediction using k-Nearest Neighbors.

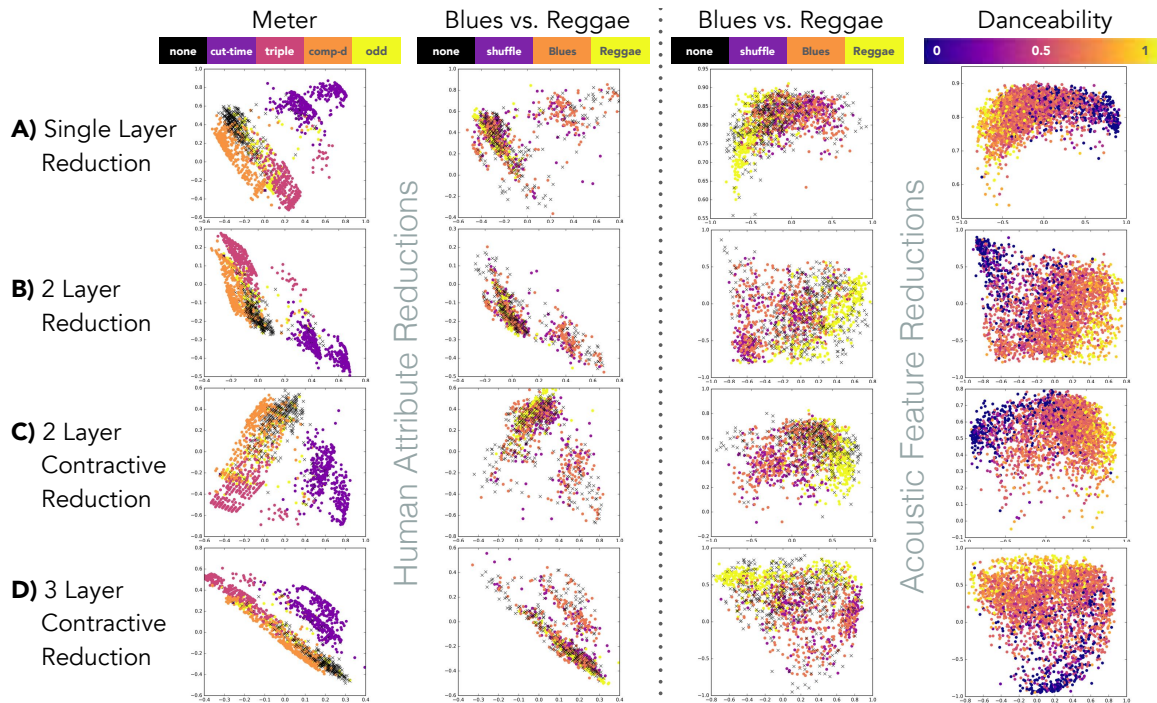


Figure C.2: Spaces with selected label colorings from each model learned from human-annotated attributes (left) and audio features (right)

C.3 Spatial Organization

In Figure C.3, we take a closer look at each of the generated spaces. Each space is evaluated on its ability to capture meaningful similarities and differences among rhythmic attributes. The 1st and 2nd columns show reductions from human-tagged attributes. The 1st column displays a collection of all meter labels organized jointly in each space along with a ‘none’ label (interpreted as simple-duple). From these meter plots, it is clear that human-attribute driven spaces are able provide a pretty substantial meter separation. In the 2nd and 3rd column, we explore examples where shuffle (a rhythmic feel attribute) is present along with examples of two genres that contain shuffle (Blues, Reggae) and a ‘none’ label (not Blues, Reggae, or shuffle). In this set, the human attribute reductions are not able to capture differences in the styles, while the audio feature reductions are, suggesting some understanding of genre context in the interpretation of shuffle. Comparing columns

3 and 4, areas where Reggae is present also have high danceability ratings while Blues areas have lower danceability ratings, suggesting the capture of broader contexts.

C.4 Evaluation

The efficacy of the reductions are evaluated quantitatively by predicting a set of rhythmic attribute labels in the reduced space using *k-Nearest Neighbors* (Similar to Chapter 9). Table C.1 (left) shows that the reductions learned from human-labeled attributes are powerful representations, capturing much of the rhythmic information in the lower-dimensional embedding. Each of the reduction types (A,B,C,D) perform similarly when reducing the human-annotated rhythm attributes, which, as stated previously, may result from the low initial dimensionality of the attribute space (10-D). The audio feature reductions, while not as effective, still capture distinguishing rhythm information in the reduced space (Table C.1, right). This shows the potential for use in scalable systems that require efficient rhythm representation.

Labels Bin. (AUC)	Human				Audio			
	1L (A)	2L (B)	2L (C)	3L (D)	1L (A)	2L (B)	2L (C)	3L (D)
Cut-Time	0.994	0.995	0.995	0.995	0.671	0.710	0.662	0.698
Triple	0.995	0.997	0.995	0.992	0.735	0.736	0.746	0.736
C-D	0.986	0.978	0.981	0.976	0.707	0.714	0.716	0.718
Odd	0.944	0.911	0.926	0.919	0.688	0.707	0.660	0.672
Swing	0.993	0.994	0.993	0.992	0.731	0.783	0.760	0.753
Shuffle	0.940	0.927	0.942	0.931	0.780	0.835	0.854	0.800
Syncop.	0.971	0.972	0.973	0.969	0.633	0.635	0.595	0.641
Cont. (R^2)								
	1L (A)	2L (B)	2L (C)	3L (D)	1L (A)	2L (B)	2L (C)	3L (D)
Tempo	0.862	0.840	0.869	0.841	0.081	0.107	0.105	0.098
Backbeat	0.918	0.913	0.925	0.914	0.331	0.350	0.382	0.332
Danceable	0.909	0.895	0.918	0.906	0.382	0.401	0.418	0.388

Table C.1: The rhythmic attribute predictions in the reduced spaces.

Along with the quantitative evaluation in Table C.1, it is necessary to qualitatively explore the intuition of these spaces as well. This was introduced previously in this section, but here we will explore the meaning of the spatial organization. For simplicity, we'll explore the single layer (A) human attribute space and the 2-layer (B) acoustic feature space (Figure C.3). A few obvious organizations appear when looking at the embeddings. In the human annotation trained visualizations, meter clusters very clearly, swing is a distinguished feel, and Danceability and Tempo have clear diagonal, perpendicular gradients. Similar types of organization appears in the audio

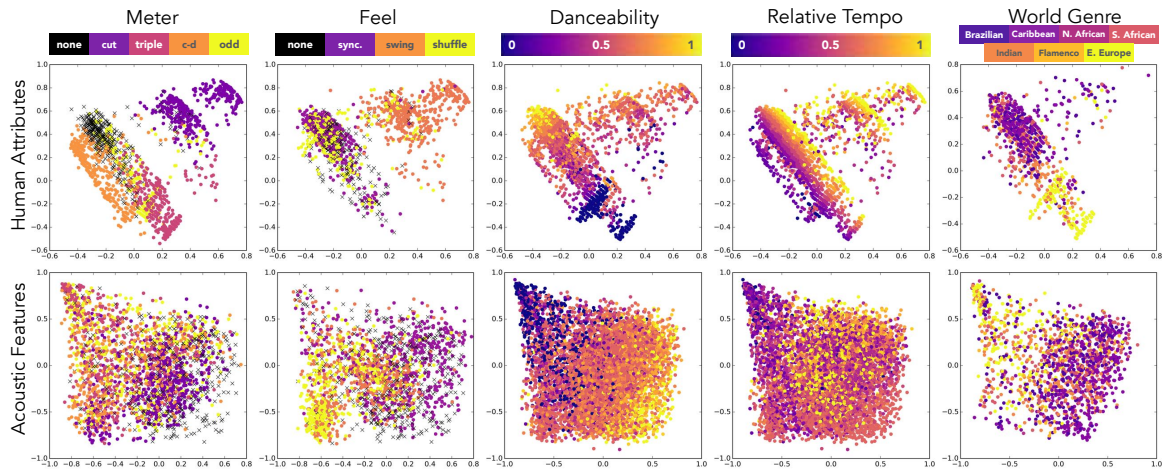


Figure C.3: Selected label colorings for embeddings learned from human-annotated attributes (top) and audio features (bottom)

feature reductions. Meter evolves from odd/triple to duple along the top-L to lower-R diagonal, swing and shuffle are separate from syncopation, and danceability shows another clear diagonal gradient. Interesting patterns emerge when genre is explored in these spaces as well. When looking at “World Music”, both spaces pick up on the similarity of music in Africa to music in the Caribbean and Brazil, suggesting the importance of African rhythmic influence on music in Latin America (*Afro-Latin* music).

C.5 Conclusions

A set of rhythm spaces was generated from expert-annotated attributes as well as acoustic features and consistent and intuitive embeddings of rhythmic attributes were present in both. These consistent representations learned from acoustic features can be used to develop scalable automated tools to explore co-occurring musical attributes and uncover relationships that are sometimes nebulous (i.e., culture, influence). In another vein, a low-dimensional organization of rhythmic similarity that embeds expert annotations can be employed for more efficient automated playlist generation. It can also be used for intuitive music organization, exploration, and discovery. In future work, through listener feedback, one can evaluate the embeddings’ ability to capture the rhythmic information

that listeners deem important. Furthermore, by incorporating genre, it may be possible to develop a model of listeners' rhythm preferences within the context of the styles they enjoy.

Appendix D: Attribute Prediction and Tempo Estimation

Meter (4-class, Genre) Feature	GT Tempo (τ_{gt})	GT + Error (τ_{ϵ})	Estimated (τ_{est})	Estimated Diff. ($\tau_{est} - \tau_{gt}$)	GT Diff. ($\tau_{\epsilon} - \tau_{gt}$)
BPDIST	0.818	0.808	0.785	-0.033	-0.009
BPDIST_M (B)	0.806	0.797	0.783	-0.023	-0.009
TGR	0.854	0.896	0.834	-0.020	0.042
TGR_M (T)	0.849	0.885	0.823	-0.026	0.035
MELLIN (S)	0.848	0.851	0.853	0.005	0.003
MELLIN_DCT_MED (D)	0.867	0.862	0.851	-0.016	-0.005
(S) (B) (T)	0.891	0.862	0.880	-0.011	-0.029
(D) (B) (T)	0.890	0.872	0.882	-0.008	-0.017
MFCC (M)	0.787	0.767	0.787	0.000	-0.019
(M) (S) (B) (T)	0.877	0.874	0.887	0.010	-0.003
(M) (D) (B) (T)	0.891	0.871	0.885	-0.006	-0.020

Table D.1: Meter classification of 4 genre meter classes from the GTZAN Rhythm Dataset

Genre (10-class, Genre) Feature	GT Tempo (τ_{gt})	GT + Error (τ_{ϵ})	Estimated (τ_{est})	Estimated Diff. ($\tau_{est} - \tau_{gt}$)	GT Diff. ($\tau_{\epsilon} - \tau_{gt}$)
BPDIST	0.458	0.347	0.407	-0.051	-0.112
BPDIST_M (B)	0.490	0.384	0.476	-0.013	-0.105
TGR	0.313	0.163	0.345	0.032	-0.150
TGR_M (T)	0.389	0.174	0.391	0.002	-0.216
MELLIN (S)	0.434	0.417	0.407	-0.028	-0.017
MELLIN_DCT_MED (D)	0.364	0.358	0.364	0.000	-0.006
(S) (B) (T)	0.548	0.475	0.546	-0.002	-0.073
(D) (B) (T)	0.515	0.409	0.532	0.017	-0.106
MFCC (M)	0.654	0.670	0.666	0.011	0.016
(M) (S) (B) (T)	0.727	0.667	0.694	-0.032	-0.060
(M) (D) (B) (T)	0.713	0.657	0.693	-0.020	-0.056

Table D.2: Genre classification of 10 genre classes from the GTZAN Rhythm Dataset

Triple Meter (Genre) Feature	GT Tempo (τ_{gt})	GT + Error (τ_{ϵ})	Estimated (τ_{est})	Estimated Diff. ($\tau_{est} - \tau_{gt}$)	GT Diff. ($\tau_{\epsilon} - \tau_{gt}$)
BPDIST	0.776	0.695	0.789	0.013	-0.081
BPDIST_M (B)	0.745	0.670	0.819	0.074	-0.075
TGR	0.745	0.550	0.685	-0.061	-0.195
TGR_M (T)	0.692	0.686	0.669	-0.023	-0.006
MELLIN (S)	0.851	0.849	0.861	0.010	-0.002
MELLIN_DCT_MED (D)	0.769	0.770	0.766	-0.003	0.001
(S) (B) (T)	0.887	0.873	0.835	-0.051	-0.014
(D) (B) (T)	0.796	0.791	0.839	0.043	-0.005
MFCC (M)	0.813	0.797	0.830	0.017	-0.016
(M) (S) (B) (T)	0.850	0.841	0.881	0.031	-0.009
(M) (D) (B) (T)	0.848	0.863	0.877	0.029	0.014

Table D.3: Triple Meter classification on the GTZAN Rhythm Dataset

C-D Meter (Genre) Feature	GT Tempo (τ_{gt})	GT + Error (τ_{ϵ})	Estimated (τ_{est})	Estimated Diff. ($\tau_{est} - \tau_{gt}$)	GT Diff. ($\tau_{\epsilon} - \tau_{gt}$)
BPDIST	0.953	0.912	0.707	-0.246	-0.041
BPDIST_M (B)	0.933	0.842	0.618	-0.315	-0.090
TGR	0.953	0.769	0.675	-0.278	-0.184
TGR_M (T)	0.936	0.719	0.551	-0.385	-0.218
MELLIN (S)	0.785	0.794	0.833	0.048	0.009
MELLIN_DCT_MED (D)	0.763	0.744	0.749	-0.014	-0.019
(S) (B) (T)	0.959	0.877	0.881	-0.077	-0.082
(D) (B) (T)	0.917	0.925	0.729	-0.188	0.008
MFCC (M)	0.612	0.608	0.567	-0.045	-0.004
(M) (S) (B) (T)	0.946	0.919	0.868	-0.079	-0.028
(M) (D) (B) (T)	0.912	0.919	0.772	-0.140	0.006

Table D.4: Compound-Duple Meter classification on the GTZAN Rhythm Dataset

Mixed Meter (Genre) Feature	GT Tempo (τ_{gt})	GT + Error (τ_{ϵ})	Estimated (τ_{est})	Estimated Diff. ($\tau_{est} - \tau_{gt}$)	GT Diff. ($\tau_{\epsilon} - \tau_{gt}$)
BPDIST	0.691	0.719	0.754	0.063	0.028
BPDIST_M (B)	0.571	0.699	0.643	0.072	0.127
TGR	0.703	0.561	0.615	-0.088	-0.142
TGR_M (T)	0.505	0.486	0.445	-0.060	-0.019
MELLIN (S)	0.760	0.758	0.744	-0.016	-0.002
MELLIN_DCT_MED (D)	0.504	0.537	0.449	-0.055	0.033
(S) (B) (T)	0.681	0.719	0.702	0.021	0.038
(D) (B) (T)	0.467	0.531	0.519	0.053	0.064
MFCC (M)	0.655	0.632	0.661	0.006	-0.023
(M) (S) (B) (T)	0.734	0.767	0.772	0.038	0.033
(M) (D) (B) (T)	0.527	0.568	0.588	0.061	0.041

Table D.5: Mixed Meter classification on the GTZAN Rhythm Dataset

Duple Meter (Genre) Feature	GT Tempo (τ_{gt})	GT + Error (τ_{ϵ})	Estimated (τ_{est})	Estimated Diff. ($\tau_{est} - \tau_{gt}$)	GT Diff. ($\tau_{\epsilon} - \tau_{gt}$)
BPDIST	0.812	0.772	0.732	-0.079	-0.040
BPDIST_M (B)	0.746	0.719	0.759	0.013	-0.027
TGR	0.765	0.669	0.724	-0.040	-0.095
TGR_M (T)	0.769	0.666	0.703	-0.066	-0.103
MELLIN (S)	0.843	0.845	0.825	-0.018	0.002
MELLIN_DCT_MED (D)	0.639	0.677	0.683	0.044	0.038
(S) (B) (T)	0.810	0.781	0.823	0.013	-0.029
(D) (B) (T)	0.824	0.725	0.760	-0.063	-0.099
MFCC (M)	0.712	0.711	0.717	0.005	-0.001
(M) (S) (B) (T)	0.827	0.812	0.824	-0.002	-0.014
(M) (D) (B) (T)	0.778	0.737	0.764	-0.013	-0.041

Table D.6: Duple Meter classification on the GTZAN Rhythm Dataset

Triplet Feel (Genre) Feature	GT Tempo (τ_{gt})	GT + Error (τ_{ϵ})	Estimated (τ_{est})	Estimated Diff. ($\tau_{est} - \tau_{gt}$)	GT Diff. ($\tau_{\epsilon} - \tau_{gt}$)
BPDIST	0.934	0.908	0.645	-0.289	-0.027
BPDIST_M (B)	0.958	0.880	0.686	-0.272	-0.078
TGR	0.955	0.747	0.655	-0.301	-0.208
TGR_M (T)	0.962	0.710	0.509	-0.452	-0.251
MELLIN (S)	0.845	0.836	0.851	0.006	-0.008
MELLIN_DCT_MED (D)	0.703	0.718	0.720	0.017	0.015
(S) (B) (T)	0.964	0.933	0.895	-0.069	-0.031
(D) (B) (T)	0.921	0.848	0.808	-0.113	-0.073
MFCC (M)	0.578	0.633	0.584	0.006	0.054
(M) (S) (B) (T)	0.935	0.910	0.863	-0.072	-0.025
(M) (D) (B) (T)	0.921	0.880	0.796	-0.125	-0.041

Table D.7: Triplet Feel classification on the GTZAN Rhythm Dataset

Swing Feel (Genre) Feature	GT Tempo (τ_{gt})	GT + Error (τ_{ϵ})	Estimated (τ_{est})	Estimated Diff. ($\tau_{est} - \tau_{gt}$)	GT Diff. ($\tau_{\epsilon} - \tau_{gt}$)
BPDIST	0.971	0.899	0.889	-0.082	-0.072
BPDIST_M (B)	0.950	0.884	0.850	-0.100	-0.066
TGR	0.971	0.725	0.853	-0.118	-0.246
TGR_M (T)	0.971	0.719	0.858	-0.114	-0.252
MELLIN (S)	0.928	0.922	0.931	0.003	-0.006
MELLIN_DCT_MED (D)	0.859	0.848	0.856	-0.003	-0.011
(S) (B) (T)	0.971	0.931	0.935	-0.036	-0.040
(D) (B) (T)	0.964	0.913	0.918	-0.046	-0.050
MFCC (M)	0.658	0.648	0.647	-0.011	-0.010
(M) (S) (B) (T)	0.970	0.943	0.936	-0.034	-0.027
(M) (D) (B) (T)	0.962	0.937	0.930	-0.032	-0.026

Table D.8: Swing classification on the GTZAN Rhythm Dataset

Style (Ballroom) Feature	GT Tempo (τ_{gt})	GT + Error (τ_{ϵ})	Estimated (τ_{est})	Estimated Diff. ($\tau_{est} - \tau_{gt}$)	GT Diff. ($\tau_{\epsilon} - \tau_{gt}$)
BPDIST	0.769	0.551	0.743	-0.026	-0.218
BPDIST_M (B)	0.739	0.518	0.723	-0.016	-0.221
TGR	0.838	0.557	0.748	-0.090	-0.281
TGR_M (T)	0.869	0.566	0.806	-0.063	-0.303
MELLIN (S)	0.832	0.851	0.837	0.004	0.018
MELLIN_DCT_MED (D)	0.866	0.854	0.863	-0.003	-0.011
(S) (B) (T)	0.906	0.852	0.898	-0.008	-0.054
(D) (B) (T)	0.939	0.885	0.915	-0.023	-0.053
MFCC (M)	0.462	0.447	0.460	-0.002	-0.016
(M) (S) (B) (T)	0.924	0.867	0.914	-0.010	-0.056
(M) (D) (B) (T)	0.947	0.909	0.923	-0.024	-0.039

Table D.9: Style classification on the Ballroom Dataset

Triple Meter (Ballroom) Feature	GT Tempo (τ_{gt})	GT + Error (τ_{ϵ})	Estimated (τ_{est})	Estimated Diff. ($\tau_{est} - \tau_{gt}$)	GT Diff. ($\tau_{\epsilon} - \tau_{gt}$)
BPDIST	0.946	0.878	0.931	-0.015	-0.068
BPDIST_M (B)	0.919	0.837	0.920	0.001	-0.082
TGR	0.954	0.838	0.968	0.014	-0.116
TGR_M (T)	0.982	0.868	0.965	-0.017	-0.114
MELLIN (S)	0.988	0.986	0.982	-0.006	-0.002
MELLIN_DCT_MED (D)	0.952	0.952	0.954	0.002	0.000
(S) (B) (T)	0.994	0.987	0.992	-0.002	-0.007
(D) (B) (T)	0.993	0.976	0.991	-0.002	-0.018
MFCC (M)	0.914	0.912	0.920	0.006	-0.001
(M) (S) (B) (T)	0.997	0.990	0.996	-0.001	-0.007
(M) (D) (B) (T)	0.993	0.986	0.994	0.001	-0.006

Table D.10: Meter classification on the Ballroom Dataset

Bibliography

- [1] D. Turnbull, L. Barrington, and G. Lanckriet, “Five Approaches to Collecting Tags for Music.” *Proc. of the International Society for Music Information Retrieval Conference*, 2008.
- [2] A. Holzapfel, Others, and M. Davies, “Selective Sampling for Beat Tracking Evaluation,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 9, pp. 2539–2548, 2012.
- [3] J. Foote and S. Uchihashi, “The beat spectrum: a new approach to rhythm analysis,” *IEEE International Conference on Multimedia and Expo, 2001. ICME 2001.*, pp. 881–884, 2001.
- [4] F. Krebs, S. Böck, and G. Widmer, “Rhythmic Pattern Modeling for Beat and Downbeat Tracking in Musical Audio.” *Proc. of the International Society for Music Information Retrieval Conference*, 2013.
- [5] G. Tzanetakis and P. Cook, “Musical genre classification of audio signals,” *IEEE Trans. on Audio, Speech and Language Processing*, vol. 10, no. 5, pp. 293–302, 2002.
- [6] M. Slaney, “Semantic-audio retrieval,” *Proc. of the International Conference on Acoustics, Speech and Signal Processing*, 2002.
- [7] Y. E. Y. Kim, E. M. E. Schmidt, R. Migneco, B. Morton, P. Richardson, J. Scott, J. A. Speck, and D. Turnbull, “Music emotion recognition: A state of the art review,” *Proc. ISMIR*, no. Proc. of the International Society for Music Information Retrieval Conference, pp. 255–266, 2010.
- [8] S. Canazza, G. Poli, A. Rodà, and A. Vidolin, “An abstract control space for communication of sensory expressive intentions in music performance,” *Journal of New Music Research*, 2003.
- [9] J. Langner and W. Goebel, “Visualizing expressive performance in tempo—loudness space,” *Computer Music Journal*, 2003.
- [10] M. Prockup, A. F. Ehmann, F. Gouyon, E. M. Schmidt, and Y. E. Kim, “Modeling musical rhythm at scale using the music genome project,” *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, 2015.
- [11] A. V. Oppenheim and R. W. Schaffer, *Discrete-time signal processing*. Pearson Higher Education, 2010.
- [12] N. Ono, K. Miyamoto, and J. L. Roux, “Separation of a monaural audio signal into harmonic/percussive components by complementary diffusion on spectrogram,” *Proc. of the European Signal Processing Conference*, pp. 1–4, 2008.
- [13] D. Fitzgerald, “Harmonic/Percussive Separation Using Median Filtering,” *Proc. of the International Conference on Digital Audio Effects*, no. 1, pp. 10–13, 2010.
- [14] H. C. Longuet-Higgins and C. S. Lee, “The perception of musical rhythms,” *Perception*, vol. 11, pp. 115–128, 1982.
- [15] A. Klapuri, “Sound onset detection by applying psychoacoustic knowledge,” *Proc. of the International Conference on Acoustics, Speech and Signal Processing*, 1999.
- [16] E. D. Scheirer, “Tempo and beat analysis of acoustic musical signals.” *The Journal of the Acoustical Society of America*, vol. 103, no. 1, pp. 588–601, Jan. 1998.

- [17] S. Dixon, “Learning to detect onsets of acoustic piano tones,” *Proceedings of the Workshop on Current Directions in Computer Music Research*, 2001.
- [18] J. Bello and M. Sandler, “Phase-based note onset detection for music signals,” *Proc. of the International Conference on Acoustics, Speech and Signal Processing*, 2003.
- [19] S. Dixon, “Onset detection revisited,” *Proc. of the International Conference on Digital Audio Effects*, 2006.
- [20] F. Gouyon, S. Dixon, G. Widmer, and Others, “Evaluating Low-Level Features for Beat Classification and Tracking,” in *Proc. of the International Conference on Acoustics, Speech and Signal Processing*, 2007.
- [21] S. Böck and G. Widmer, “Maximum filter vibrato suppression for onset detection,” in *Proc. of the International Conference on Digital Audio Effects (DAFx-13)*, 2013.
- [22] N. Degara, S. Member, M. E. P. M. Davies, A. Pena, and M. D. Plumbley, “Onset Event Decoding Exploiting the Rhythmic Structure of Polyphonic Music.” *J. Sel. Topics Signal Processing*, vol. 5, no. 6, pp. 1228–1239, 2011.
- [23] R. B. Dannenberg and B. Mont-Reynaud, “Following an Improvisation in Real Time,” in *Proceedings of the 1987 International Computer Music Conference*, 1987, pp. 241–248.
- [24] P. Allen and R. Dannenberg, “Tracking musical beats in real time,” *Proc. of the International Computer Music Conference*, 1990.
- [25] M. Goto and Y. Muraoka, “A beat tracking system for acoustic signals of music,” *ACM Multimedia*, 1994.
- [26] S. Dixon, “Beat induction and rhythm recognition,” *Advanced Topics in Artificial Intelligence*, 1997.
- [27] F. Gouyon, P. Herrera, and P. Cano, “Pulse-dependent analyses of percussive music,” *International Conference on Virtual, Synthetic and Entertainment Audio*, 2002.
- [28] F. Gouyon and P. Herrera, “A beat induction method for musical audio signals,” *Proceedings of the Fourth European Workshop on Image Analysis for Multimedia Interactive Services*, 2003.
- [29] M. Davies and M. Plumbley, “Beat tracking with a two state model,” *Proc. of the International Conference on Acoustics, Speech and Signal Processing*, 2005.
- [30] M. M. Davies and M. M. Plumbley, “Context-Dependent Beat Tracking of Musical Audio,” *Audio, Speech, and Language Processing, IEEE Trans.*, vol. 15, no. 3, pp. 1009–1020, Mar. 2007.
- [31] S. Dixon, “Automatic Extraction of Tempo and Beat From Expressive Performances,” *Journal of New Music Research*, vol. 30, no. 1, pp. 39–58, Mar. 2001.
- [32] —, “Evaluation of the audio beat tracking system beatroot,” *Journal of New Music Research*, 2007.
- [33] D. P. W. Ellis, “Beat Tracking by Dynamic Programming,” *Journal of New Music Research*, vol. 36, no. 1, pp. 51–60, Mar. 2007.
- [34] J. Oliveira, F. Gouyon, L. Martins, and L. Reis, “IBT: A real-time tempo and beat tracking system,” 2010.
- [35] J. Oliveira and M. Davies, “Beat tracking for multiple applications: A multi-agent system architecture with state recovery,” *Audio, Speech, and Language Processing, IEEE Trans.*, 2012.

- [36] G. Peeters and H. Papadopoulos, “Simultaneous Beat and Downbeat-Tracking Using a Probabilistic Framework: Theory and Large-Scale Evaluation,” *Audio, Speech, and Language Processing, IEEE Trans.*, vol. 19, no. 6, pp. 1–17, Aug. 2011.
- [37] S. Böck and M. Schedl, “Enhanced beat tracking with context-aware neural networks,” *Proc. of the International Conference on Digital Audio Effects*, 2011.
- [38] S. Böck, F. Krebs, and G. Widmer, “A multi-model approach to beat tracking considering heterogeneous music styles,” *Proc. of the International Society for Music Information Retrieval Conference*, 2014.
- [39] F. Krebs and F. Korzeniowski, “Unsupervised learning and refinement of rhythmic patterns for beat and downbeat tracking,” *Proc. of the European Signal Processing Conference*, 2014.
- [40] M. Davies, I. TEC, and S. Böck, “Evaluating the Evaluation Measures for Beat Tracking,” *Proc. of the International Society for Music Information Retrieval Conference*, 2014.
- [41] J. Seppänen, “Computational models of musical meter recognition,” Ph.D. dissertation, 2001.
- [42] F. Gouyon and P. Herrera, “Determination of the meter of musical audio signals: Seeking recurrences in beat segment descriptors,” *Audio Engineering Society Convention 114*, 2003.
- [43] T. Jehan, “Downbeat prediction by listening and learning,” *Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, 2005.
- [44] A. a. A. Klapuri, Others, a. A. Eronen, and J. J. Astola, “Analysis of the Meter of Acoustic Musical Signals,” *IEEE Transactions on Speech and Audio Processing*, vol. 14, no. 1, pp. 342–355, Jan. 2006.
- [45] B. Schuller, F. Eyben, and G. Rigoll, “Fast and Robust Meter and Tempo Recognition for the Automatic Discrimination of Ballroom Dance Styles,” *Proc. of the International Conference on Acoustics, Speech and Signal Processing*, Apr. 2007.
- [46] H. Papadopoulos and G. Peeters, “Joint estimation of chords and downbeats from an audio signal,” *Audio, Speech, and Language Processing, IEEE Trans.*, 2011.
- [47] J. Seppanen, “Tatum grid analysis of musical signals,” *Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, 2001.
- [48] M. Alghoniemy and A. H. Tewfik, “Rhythm and periodicity detection in polyphonic music,” *Multimedia Signal Processing, 1999 IEEE 3rd Workshop on.*, 1999.
- [49] J. Paulus and A. Klapuri, “Measuring the similarity of Rhythmic Patterns.” *Proc. of the International Society for Music Information Retrieval Conference*, vol. 1, 2002.
- [50] J. Foote and M. Cooper, “Visualizing musical structure and rhythm via self-similarity,” in *Proc. of the International Computer Music Conference*, 2001.
- [51] S. Dixon, E. Pampalk, and G. Widmer, “Classification of dance music by periodicity patterns.” *Proc. of the International Society for Music Information Retrieval Conference*, no. Proc. of the International Society for Music Information Retrieval Conference, pp. 1–7, 2003.
- [52] F. Gouyon, S. Dixon, E. Pampalk, and G. Widmer, “Evaluating rhythmic descriptors for musical genre classification,” in *Proc. of the AES 25th International Conference*, 2004, pp. 196–204.
- [53] G. Peeters, “Rhythm classification using spectral rhythm patterns.” in *Proc. of the International Society for Music Information Retrieval Conference*, 2005, pp. 644–647.
- [54] E. Pampalk, A. Rauber, and D. Merkl, “Content-based organization and visualization of music archives,” *ACM Multimedia*, 2002.

- [55] T. Pohle, D. Schnitzer, M. Schedl, P. Knees, and G. Widmer, “On Rhythm and General Music Similarity.” *Proc. of the International Society for Music Information Retrieval Conference*, no. Proc. of the International Society for Music Information Retrieval Conference, pp. 525–530, 2009.
- [56] E. Tsunoo, N. Ono, and S. Sagayama, “Rhythm map: Extraction of unit rhythmic patterns and analysis of rhythmic structure from music acoustic signals,” *Proc. of the International Conference on Acoustics, Speech and Signal Processing*, 2009.
- [57] E. Tsunoo and G. Tzanetakis, “Audio genre classification using percussive pattern clustering combined with timbral features,” *Multimedia and Expo, 2009. ICME 2009. IEEE International Conference on.*, 2009.
- [58] E. Tsunoo, G. Tzanetakis, N. Ono, and S. Sagayama, “Beyond timbral statistics: Improving music classification using percussive patterns and bass lines,” *Audio, Speech, and Language Processing, IEEE Trans.*, vol. 19, no. 4, pp. 1003–1014, May 2011.
- [59] T. Völkel, J. Abeßer, C. Dittmar, H. Großmann, and T. V. Et al., “Automatic genre classification of Latin American music using characteristic rhythmic patterns.” in *Audio Mostly Conference*, 2010, p. 16.
- [60] A. Holzapfel and Y. Stylianou, “Scale transform in rhythmic similarity of music,” *IEEE Trans. on Audio, Speech and Language Processing*, vol. 19, no. 1, pp. 176–185, 2011.
- [61] E. Battenberg and D. Wessel, “Analyzing Drum Patterns Using Conditional Deep Belief Networks,” in *Proc. of the International Society for Music Information Retrieval Conference*, no. Proc. of the International Society for Music Information Retrieval Conference, 2012, pp. 37–42.
- [62] D. FitzGerald, R. Lawlor, and E. Coyle, “Sub-band independent subspace analysis for drum transcription,” *Proc. of the International Conference on Digital Audio Effects*, 2002.
- [63] D. FitzGerald, B. Lawlor, and E. Coyle, “Drum transcription in the presence of pitched instruments using prior subspace analysis,” *Proc. of the Irish Signals and Systems Conference*, 2003.
- [64] —, “Drum transcription using automatic grouping of events and prior subspace analysis,” *Proceedings of the 4th European Workshop on Image Analysis for Multimedia Interactive Services*, 2003.
- [65] D. FitzGerald, R. Lawlor, and E. Coyle, “Prior subspace analysis for drum transcription,” *Audio Engineering Society Convention*, 2003.
- [66] D. Fitzgerald, “Automatic drum transcription and source separation,” 2004.
- [67] D. FitzGerald and J. Paulus, “Unpitched percussion transcription,” *Signal Processing Methods for Music Transcription*, 2006.
- [68] O. Gillet and G. Richard, “Automatic transcription of drum loops,” *Proc. of the International Conference on Acoustics, Speech and Signal Processing*, 2004.
- [69] —, “Drum Track Transcription of Polyphonic Music Using Noise Subspace Projection.” *Proc. of the International Society for Music Information Retrieval Conference*, 2005.
- [70] —, “ENST-Drums: an extensive audio-visual database for drum signals processing.” *Proc. of the International Society for Music Information Retrieval Conference*, 2006.
- [71] —, “Supervised and Unsupervised Sequence Modelling for Drum Transcription.” *Proc. of the International Society for Music Information Retrieval Conference*, 2007.

- [72] —, “Transcription and separation of drum signals from polyphonic music,” *Audio, Speech, and Language Processing, IEEE Trans.*, 2008.
- [73] G. Tzanetakis, “Subband-based drum transcription for audio signals,” *Multimedia Signal Processing, IEEE Workshop on.*, 2005.
- [74] K. Yoshii, M. Goto, and H. Okuno, “Automatic Drum Sound Description for Real-World Music Using Template Adaptation and Matching Methods.” *Proc. of the International Society for Music Information Retrieval Conference*, 2004.
- [75] —, “Adamast: A drum sound recognizer based on adaptation and matching of spectrogram templates,” *Proc. of the International Society for Music Information Retrieval Conference*, 2005.
- [76] K. Yoshii and K. Komatani, “An error correction framework based on drum pattern periodicity for improving drum sound detection,” *Proc. of the International Conference on Acoustics, Speech and Signal Processing*, 2006.
- [77] J. Paulus and T. Virtanen, “Drum transcription with non-negative spectrogram factorisation,” in *Proceedings of the 13th European Signal Processing Conference*, 2005, p. 4.
- [78] J. Paulus and A. Klapuri, “Combining Temporal and Spectral Features in HMM-Based Drum Transcription.” *Proc. of the International Society for Music Information Retrieval Conference*, 2007.
- [79] L. Thompson, M. Mauch, and S. Dixon, “Drum Transcription Via Classification of Bar-Level Rhythmic Patterns,” *Proc. of the International Society for Music Information Retrieval Conference*.
- [80] M. Sordo, O. Celma, M. Blech, and E. Guaus, “The quest for musical genres: Do the experts and the wisdom of crowds agree?” *Proc. of the International Society for Music Information Retrieval Conference*, 2008.
- [81] H. Soltau and T. Schultz, “Recognition of music types,” *Proc. of the International Conference on Acoustics, Speech and Signal Processing*, 1998.
- [82] K. West and S. Cox, “Features and classifiers for the automatic classification of musical audio signals.” *Proc. of the International Society for Music Information Retrieval Conference*, 2004.
- [83] —, “Finding An Optimal Segmentation for Audio Genre Classification.” *Proc. of the International Society for Music Information Retrieval Conference*, 2005.
- [84] E. Pampalk, A. Flexer, and G. Widmer, “Improvements of audio-based music similarity and genre classificaton.” in *Proc. of the International Society for Music Information Retrieval Conference*, vol. 5, 2005, pp. 634–637.
- [85] F. Gouyon, A. Klapuri, S. Dixon, M. Alonso, G. Tzanetakis, C. Uhle, and P. Cano, “An experimental comparison of audio tempo induction algorithms,” *Audio, Speech, and Language Processing, IEEE Trans.*, vol. 14, no. 5, pp. 1832–1844, Sep. 2006.
- [86] M. Lopes, F. Gouyon, A. L. Koerich, and L. E. Oliveira, “Selection of training instances for music genre classification,” in *Proc. of the International Conference on Pattern Recognition*. IEEE, 2010, pp. 4569–4572.
- [87] G. Marques and T. Langlois, “Short-term feature space and music genre classification,” *Journal of New Music Research*, 2011.
- [88] B. Sturm, “An analysis of the GTZAN music genre dataset,” *ACM Workshop on Music Information Retrieval with User-Centered and Multimodal Strategies*, 2012.

- [89] —, “Two systems for automatic music genre recognition: What are they really recognizing?” *ACM Workshop on Music Information Retrieval with User-Centered and Multimodal Strategies*, 2012.
- [90] B. Whitman, “Semantic rank reduction of music audio,” *Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, 2003.
- [91] P. Knees, E. Pampalk, and G. Widmer, “Artist Classification with Web-Based Data.” *Proc. of the International Society for Music Information Retrieval Conference*, 2004.
- [92] P. Knees, T. Pohle, M. Schedl, and G. Widmer, “Combining audio-based similarity with web-based data to accelerate automatic music playlist generation,” in *Proc. of the 8th ACM international workshop on Multimedia information retrieval*. ACM, 2006, pp. 147–154.
- [93] O. Celma and X. Serra, “FOAFing the music: Bridging the semantic gap in music recommendation,” *The Semantic Web-ISWC 2006*, 2008.
- [94] K. Bischoff, C. Firan, R. Paiu, and W. Nejdl, “Music Mood and Theme Classification—a Hybrid Approach.” in *Proc. of the International Society for Music Information Retrieval Conference*, Kobe, Japan, 2009.
- [95] M. Levy and M. Sandler, “A semantic space for music derived from social tags,” *Austrian Computer Society*, 2007.
- [96] E. L. M. Law, L. von Ahn, R. B. Dannenberg, and M. Crawford, “{TagATune:} A Game for Music and Sound Annotation,” in *Proc. of the International Society for Music Information Retrieval Conference*, Vienna, Austria, 2007.
- [97] E. Law, K. West, M. Mandel, M. Bay, and J. Downie, “Evaluation of Algorithms Using Games: The Case of Music Tagging.” *Proc. of the International Society for Music Information Retrieval Conference*, 2009.
- [98] M. Mandel and D. Ellis, “A web-based game for collecting music metadata,” *Journal of New Music Research*, 2008.
- [99] D. Eck, T. Bertin-Mahieux, and P. Lamere, “Autotagging Music Using Supervised Machine Learning.” *Proc. of the International Society for Music Information Retrieval Conference*, 2007.
- [100] D. Eck, P. Lamere, T. Bertin-Mahieux, and S. Green, “Automatic generation of social tags for music recommendation,” in *Advances in neural information processing systems*, 2008, pp. 385–392.
- [101] D. Tingle, Y. E. Kim, and D. Turnbull, “Exploring automatic music annotation with acoustically-objective tags,” in *Proc. of the international conference on Multimedia information retrieval*. ACM, 2010, pp. 55–62.
- [102] D. Turnbull and L. Barrington, “Combining audio content and social context for semantic music discovery,” *ACM Special Interest Group on Information Retrieval*, 2009.
- [103] B. McFee, L. Barrington, and G. Lanckriet, “Learning Similarity from Collaborative Filters.” *Proc. of the International Society for Music Information Retrieval Conference*, 2010.
- [104] E. Coviello, L. Barrington, A. B. Chan, and G. R. Lanckriet, “Automatic music tagging with time series models.” in *Proc. of the International Society for Music Information Retrieval Conference*, 2010, pp. 81–86.
- [105] M. Mandel, R. Pascanu, and D. Eck, “Contextual tag inference,” *Multimedia Computing, Communications, and Applications, ACM Trans.*, 2011.

- [106] K. Hevner, “The Affective Value of Pitch and Tempo in Music,” *The American Journal of Psychology*, vol. 49, no. 4, pp. pp. 621–630, 1937.
- [107] M. Zentner, D. Grandjean, and K. Scherer, “Emotions evoked by the sound of music: characterization, classification, and measurement.” *Emotion*, 2008.
- [108] X. Hu, J. Downie, C. Laurier, M. Bay, and A. Ehmann, “The 2007 MIREX Audio Mood Classification Task: Lessons Learned,” *Proc. of the International Society for Music Information Retrieval Conference*.
- [109] R. E. Thayer, *The Biopsychology of Mood and Arousal*. Oxford, U.K.: Oxford Univ. Press, 1989.
- [110] T. Eerola, O. Lartillot, and P. Toivainen, “Prediction of Multidimensional Emotional Ratings in Music from Audio Using Multivariate Regression Models.” in *Proc. of the International Society for Music Information Retrieval Conference*, Kobe, Japan, 2009.
- [111] L. Mion, G. De Poli, and G. D. Poli, “Score-Independent Audio Features for Description of Music Expression,” *Audio, Speech, and Language Processing, IEEE Trans.*, vol. 16, no. 2, pp. 458–466, Feb. 2008.
- [112] S. Kamenetsky, D. Hill, and S. Trehub, “Effect of tempo and dynamics on the perception of emotion in music,” *Psychology of Music*, 1997.
- [113] L. Mion, G. Poli, and E. Rapana, “Perceptual organization of affective and sensorial expressive intentions in music performance,” *ACM Transactions on Applied Perception (. . . ,* vol. 7, no. 2, 2010.
- [114] B. Repp, “musical expression. III. Contributions of timing and dynamics to the aesthetic impression of pianists’ performances of the initial measures of Chopin’s Etude in E Major,” *The Journal of the Acoustical Society of America*, 1999.
- [115] W. Windsor and E. Clarke, “Expressive timing and dynamics in real and artificial musical performances: Using an algorithm as an analytical tool,” *Music Perception*, 1997.
- [116] T. Kohonen, “The self-organizing map,” *Proceedings of the IEEE*, vol. 78, no. 9, pp. 1464–1480, 1990.
- [117] L. Van der Maaten and G. Hinton, “Visualizing data using t-sne,” *Journal of Machine Learning Research*, vol. 9, no. 2579-2605, p. 85, 2008.
- [118] K. Dickerson, “Musical Query-by-Content Using Self-Organizing Maps,” 2009.
- [119] M. Prockup, J. Scott, and Y. E. Kim, “Representing Musical Patterns via the Rhythmic Style Histogram Feature,” in *Proceedings of the ACM International Conference on Multimedia - MM ’14*. New York, New York, USA: ACM Press, Nov. 2014, pp. 1057–1060.
- [120] A. Flexer, “Improving visualization of high-dimensional music similarity spaces.” in *Proc. of the International Society for Music Information Retrieval Conference*, 2015.
- [121] U. Marchand, Q. Fresnel, and G. Peeters, “Gtzan-rhythm: Extending the gtzan test-set with beat, downbeat and swing annotations,” 2015.
- [122] J. Ye, J. Chow, J. Chen, and Z. Zheng, “Stochastic gradient boosted distributed decision trees,” *ACM Information and knowledge management*, 2009.
- [123] L. Breiman, “Random forests,” *Machine learning*, 2001.
- [124] M. Kurasa and W. Rudnicki, “Musical instruments in random forest,” *Foundations of Intelligent Systems*, 2009.

- [125] A. Wierzchowska and M. Kurasa, “A comparison of random forests and ferns on recognition of instruments in jazz recordings,” *Foundations of Intelligent Systems*, 2012.
- [126] X. Jin and R. Bie, “Random forest and PCA for self-organizing maps based automatic music genre discrimination,” *Conference on Data Mining—DMIN*, 2006.
- [127] X. He, J. Pan, O. Jin, T. Xu, and B. Liu, “Practical Lessons from Predicting Clicks on Ads at Facebook,” *ACM SIGKDD*, 2014.
- [128] J. Kruskal and M. Wish, *Multidimensional scaling*, 1978.
- [129] P. Comon, “Independent component analysis, a new concept?” *Signal processing*, vol. 36, no. 3, pp. 287–314, 1994.
- [130] D. D. Lee and H. S. Seung, “Learning the parts of objects by non-negative matrix factorization,” *Nature*, vol. 401, no. 6755, pp. 788–791, 1999.
- [131] —, “Algorithms for non-negative matrix factorization,” in *Advances in neural information processing systems*. Cambridge, MA: MIT Press, 2001, no. 1.
- [132] S. Dixon, F. Gouyon, and G. Widmer, “Towards Characterisation of Music Via Rhythmic Patterns,” *Proc. of the International Society for Music Information Retrieval Conference*, 2004.
- [133] F. Gouyon and S. Dixon, “A review of automatic rhythm description systems,” *Computer music journal*, 2005.
- [134] F. Gouyon, N. Wack, and S. Dixon, “An open source tool for semi-automatic rhythmic annotation,” *International Conference on Digital Audio Effects*, 2004.
- [135] B. L. Sturm, “The state of the art ten years after a state of the art: Future research in music information retrieval,” *Journal of New Music Research*, vol. 43, no. 2, pp. 147–172, 2014.
- [136] A. Flexer, F. Gouyon, S. Dixon, and G. Widmer, “Probabilistic combination of features for music classification.” in *Proc. of the International Society for Music Information Retrieval Conference*, 2006, pp. 111–114.
- [137] T. Igoe, *Groove Essentials*. Hudson Music, 2006.
- [138] K. Seyerlehner, M. Schedl, P. Knees, and R. Sonnleitner, “A refined block-level feature set for classification, similarity and tag prediction,” *Extended Abstract to MIREX*, 2011.
- [139] C. M. Bishop, *Pattern recognition and machine learning*. Springer, 2006.
- [140] K. Seyerlehner and M. Schedl, “Using block-level features for genre classification, tag classification and music similarity estimation,” *Submission to Audio Music Similarity and Retrieval Task of MIREX 2010*, 2010.
- [141] F. Pachet and D. Cazaly, “A taxonomy of musical genres.” in *Content-Based Multimedia Information Access Conference*, 2000, pp. 1238–1245.
- [142] F. Fabbri, “A theory of musical genres: Two applications,” *Popular music perspectives*, vol. 1, pp. 52–81, 1982.
- [143] J.-J. Aucouturier and F. Pachet, “Representing musical genre: A state of the art,” *Journal of New Music Research*, vol. 32, no. 1, pp. 83–93, 2003.
- [144] J. Davis and M. Goadrich, “The relationship between precision-recall and roc curves,” in *Proc. of the 23rd international conference on Machine learning*. ACM, 2006, pp. 233–240.
- [145] F. Gouyon and S. Dixon, “Dance music classification: A tempo-based approach,” in *Proc. of the International Society for Music Information Retrieval Conference*, 2004.

- [146] M. Panteli, N. Bogaards, and A. Honingh., “Modeling rhythm similarity for electronic dance music,” *Proc. of the International Society for Music Information Retrieval Conference*, 2014.
- [147] S. Jothilakshmi and N. Kathiresan, “Automatic music genre classification for indian music,” *Proc. Int. Conf. Software Computer App*, 2012.
- [148] T. Bertin-Mahieux, D. Eck, and M. Mandel, “Automatic tagging of audio: The state-of-the-art,” *Machine audition: Principles, algorithms and systems*, pp. 334–352, 2010.
- [149] R. Caruana, “Multitask learning,” *Machine learning*, vol. 28, no. 1, pp. 41–75, 1997.
- [150] L. Deng and D. Yu, “Deep convex net: A scalable architecture for speech pattern classification,” in *Proc. of Interspeech*, 2011.
- [151] F. Pachet and P. Roy, “Improving multilabel analysis of music titles: A large-scale validation of the correction approach,” *IEEE Trans. on Audio, Speech and Language Processing*, vol. 17, no. 2, pp. 335–343, 2009.
- [152] P. Herrera-Boyer, G. Peeters, and S. Dubnov, “Automatic classification of musical instrument sounds,” *Journal of New Music Research*, vol. 32, no. 1, pp. 3–21, 2003.
- [153] P. Hamel, S. Wood, and D. Eck, “Automatic identification of instrument classes in polyphonic and poly-instrument audio.” in *Proc. of the International Society for Music Information Retrieval Conference*, 2009, pp. 399–404.
- [154] J. Scott and Y. E. Kim, “Instrument identification informed multi-track mixing.” in *Proc. of the International Society for Music Information Retrieval Conference*, 2013, pp. 305–310.
- [155] U. Marchand and G. Peeters, “The modulation scale spectrum and its application to rhythm-content description.” in *Proc. of the International Conference on Digital Audio Effects*, 2014, pp. 167–172.
- [156] M. Prockup, A. F. Ehmann, F. Gouyon, E. M. Schmidt, O. Celma, and Y. E. Kim, “Modeling genre with the music genome project: Comparing human-labeled attributes and audio features,” in *Proc. of the International Society for Music Information Retrieval Conference*, 2015.
- [157] M. Prockup, A. J. Asman, A. F. Ehmann, F. Gouyon, E. M. Schmidt, and Y. E. Kim, “Modeling rhythm using tree ensembles and the music genome project,” in *Machine Learning for Music Discovery Workshop at the 32nd International Conference on Machine Learning*, 2015.
- [158] J. L. Moore, T. Joachims, and D. Turnbull, “Taste space versus the world: an embedding analysis of listening habits and geography.” in *Proc. of the International Society for Music Information Retrieval Conference*, 2014, pp. 439–444.
- [159] J. A. Russell, “A circumplex model of affect.” *Journal of personality and social psychology*, 1980.
- [160] E. M. Schmidt and Y. E. Kim, “Projection of acoustic features to continuous valence-arousal mood labels via regression.” *Proc. of the International Society for Music Information Retrieval Conference*, vol. 14, no. 1, p. 2009, 2009.
- [161] Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle, “Greedy Layer-Wise Training of Deep Networks,” in *NIPS*. MIT Press, 2007.
- [162] S. Rifai, P. Vincent, X. Muller, X. Glorot, and Y. Bengio, “Contractive auto-encoders: Explicit invariance during feature extraction,” in *Proceedings of the 28th international conference on machine learning (ICML-11)*, 2011, pp. 833–840.
- [163] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol, “Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion,” *The Journal of Machine Learning Research*, vol. 11, pp. 3371–3408, 2010.

Index

- accent signal, 2, 9, 78, 79
- autocorrelation, 5

- back-beat, 38, 92, 121
- Ballroom Dataset, 20, 21, 26, 36, 37, 73, 74, 78, 86, 88–91
- barline, 4, 17
- beat, 4, 9
- Beat Profile, 82, 83, 89, 95, 107, 116, 148
- binary labels, 38, 45

- Cal10k, 30
- compound-duple meter, 4, 38, 92, 116, 121
- Constant Q Filter Bank Transform, 7, 79
- continuous labels, 38, 42
- cut-time, 4
- cut-time meter, 38, 92, 116, 121

- danceability, 38, 92, 121
- Discrete Fourier Transform (DFT), 6
- downbeat, 4, 17
- duple, 18
- duple meter, 4

- genre, 24, 119, 124
- Gradient Boosted Tree, 2, 50, 53, 54, 93, 95
- GTZAN Genre Dataset, 36
- GTZAN Rhythm Dataset, 36, 78, 88–90

- Independent Components Analysis, 60, 61, 116, 137, 141

- Linear Regression, 2, 42, 45, 48, 93, 95
- Logistic Regression, 2, 45, 48, 89, 93, 95, 101, 106

- Mel-Frequency Cepstral Coefficients, 7
- Mellin Scale Transform, 78, 85, 89, 95, 116, 131, 149
- Mellin Scale Transform DCT, 89, 107
- meter, 4, 17
- MFCC, 7, 95
- MGP, 2, 3, 30, 36–40, 91, 92, 99, 116, 118, 148
- Music Genome Project, 1, 27, 36, 37, 42, 91, 92, 97, 99, 115, 128, 147, 148

- Nearest Neighbors, 118, 119, 121, 123, 150
- Nearest-Neighbors, 67, 68
- Non-Negative Matrix Factorization, 60, 62, 116, 135, 139

- odd meter, 4, 38, 92, 116, 121
- onset detection, 9
- onset detection function, 9
- onsets, 9

- Pandora, 1–3, 27, 36, 37, 91, 92, 97, 99, 115, 128, 147, 148
- Principal Components Analysis, 59, 116

- Random Forest, 2, 51–53, 93, 95

- Short-Time-Fourier-Transform (STFT), 6
- shuffle, 38, 92, 116, 121
- Spectral Flux, 11
- spectrogram, 6
- stacked denoising autoencoder, 148
- stacked denoising autoencoders, 147
- stochastic gradient descent, 101, 106
- SuperFlux, 11, 12, 79
- supervised component selection, 117, 121, 134, 135, 137, 139, 141
- swing, 38, 92, 116, 121
- syncopation, 5, 38, 92, 121

- t-Distributed Stochastic Neighbor Embedding, 35, 63, 116, 143
- t-SNE, 35, 63, 116, 120, 143
- tactus, 18
- Tatum, 5, 13, 82
- Tatums, 17
- tempo, 38, 121
- tempogram, 79
- Tempogram Ratio, 83, 89, 95, 107, 116, 148
- tick, 13
- ticks, 5, 17
- triple, 18
- triple meter, 4, 38, 92, 116, 121

Vita

Matthew K. Prockup

CONTACT INFORMATION

E-mail: mprockup@gmail.com *Phone:* (484) 664-8693 *Homepage:* mattprockup.com

RESEARCH INTERESTS

Modeling musical attributes: When searching, sorting, and recommending music, humans apply a variety of attributes for similarity and discrimination. By designing audio features that capture these attributes, we can develop models that allow us to automatically generate descriptions of music that are grounded and intuitive.

Interactive live performance systems: A large subset of musical performance requires a relationship between the performer and their audience. By creating interactive media technologies for musical performance, musicians can better communicate contextually relevant information and create more intimate relationships with their audiences.

EDUCATION

Drexel University, Philadelphia, Pennsylvania USA
Ph.D. Electrical Engineering, May 2016
Combined B.S./M.S., Electrical Engineering, June 2011
Minor in Music Theory/Composition, June 2011

Advisor: Youngmoo E. Kim

EXPERIENCE

Pandora Media Inc., Oakland, California USA
Scientist

Fall 2014 - present

Drexel University, Philadelphia, Pennsylvania USA
Research Assistant
Drexel AppLab Manager
Teaching Assistant
Multi-touch API Research Intern

Fall 2009 - Spring 2016

Fall 2013 - Spring 2016

Fall 2011 - Spring 2013

Spring - Summer 2008,2009

PUBLICATIONS

Prockup, M., Ehmann, A., Gouyon, F., Schmidt, E., Celma, O., Kim, Y. (2015), "Modeling Genre with the Music Genome Project: Comparing Human-Labeled Attributes and Audio Features." International Society for Music Information Retrieval Conference, Malaga, Spain, 2015.

Prockup, M., Asman, A., Ehmann, A., Gouyon, F., Schmidt, E., Kim, Y. (2015), "Modeling Rhythm Using Tree Ensembles and the Music Genome Project." Machine Learning for Music Discovery Workshop at the 32nd International Conference on Machine Learning, Lille, France, 2015.

Prockup, M., Ehmann, A., Gouyon, F., Schmidt, E., Kim, Y. (2015), "Modeling Rhythm at Scale with the Music Genome Project". IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, New Paltz, New York, 2015.

Prockup, M., Scott, J., and Kim, Y. (2014). "Representing Musical Patterns via the Rhythmic Style Histogram Feature." Proceedings of the ACM International Conference on Multimedia, Orlando, Florida, 2014.

Prockup, M., Schmidt, E. M., Scott, J. and Kim, Y. E. (2013). "Toward Understanding Expressive Percussion Through Content Based Analysis." Proceedings of the 14th International Society for Music Information Retrieval Conference. Curitiba, Brazil.

Schmidt, E. M., Prockup, M., Scott, J., Dolhansky, B., Morton, B. G., and Kim, Y. E. (2013). "Analyzing the Perceptual Saliency of Audio Features for Musical Emotion Recognition." Computer Music Modeling and Retrieval. Music and Emotions.

Prockup, M., Grunberg, D., Hrybyk, A., Kim, Y.E., (2013) "Orchestral Performance Companion: Using Real-Time Audio to Score Alignment." IEEE MultiMedia , vol.20, no.2, pp.52,60, April-June 2013

Schmidt, E. M., Prockup, M., Scott, J., Dolhansky, B., Morton, B. G. and Kim, Y. E. (2012). "Relating Perceptual and Feature Space Invariances in Music Emotion Recognition." Proceedings of the International Symposium on Computer Music Modeling and Retrieval, London, U.K.: CMMR. (*Best Student Paper Award*)

Scott, J., Schmidt, E. M., Prockup, M., Morton, B. G. and Kim, Y. E. (2012). "Predicting Time-Varying Musical Emotion Distributions from Multi-track Audio." Proceedings of the International Symposium on Computer Music Modeling and Retrieval, London, U.K.: CMMR.

Scott, J., Dolhansky, B., Prockup, M., McPherson, A., Kim, Y. E. (2012). "New Physical and Digital Interfaces for Music Creation and Expression." Proceedings of the 2012 Music, Mind and Invention Workshop, Ewing, NJ

Scott, J., Prockup, M., Schmidt, E. M., Kim, Y. E. (2011). "Automatic Multi-Track Mixing Using Linear Dynamical Systems." Proceedings of the 8th Sound and Music Computing Conference, Padova, Italy: SMC.

