



City Research Online

City St George's, University of London

Citation: Wolff, D. (2014). Spot the Odd Song Out: Similarity Model Adaptation and Analysis using Relative Human Ratings. (Unpublished Doctoral thesis, City University London)

This is the accepted version of the paper.

This version of the publication may differ from the final published version. To cite this item please consult the publisher's version.

Permanent repository link: <https://openaccess.city.ac.uk/id/eprint/5916/>

Copyright and Reuse: Copyright and Moral Rights remain with the author(s) and/or copyright holders. Copies of full items can be used for personal research or study, educational, or not-for-profit purposes without prior permission or charge, unless otherwise indicated, provided that the authors, title and full bibliographic details are credited, a hyperlink and/or URL is given for the original metadata page and the content is not changed in any way. For full details of reuse please refer to [City Research Online policy](#).

Spot the Odd Song Out: Similarity Model Adaptation and Analysis using Relative Human Ratings

Daniel Wolff

A thesis submitted for the degree of

Doctor of Philosophy (PhD)



**CITY UNIVERSITY
LONDON**

Music Informatics Research Group
Department of Computer Science

August 2014

Table of Contents

List of Figures	9
List of Tables	15
Acknowledgements	19
Abstract	21
Style and Notation Conventions	23
List of Abbreviations	25
List of Publications	27
1 Introduction	31
1.1 Aims of this Thesis	33
1.1.1 Research Questions	34
1.1.2 Adapting Models to Relative Music Similarity Ratings	34
1.1.3 Analysis of Relative Similarity Data	36
1.1.4 A Game Framework for Data Collection	37
1.1.5 A New Relative Similarity Dataset	38
1.1.6 Music Features For Large Datasets	38
1.2 Contributions	39
1.3 Thesis Overview	40
2 Fundamentals & Related Work	43
2.1 A Perceptual Perspective on Music Similarity	43
2.1.1 The Symmetry Assumption	45
2.1.2 Similarity in Music Research	45
2.2 Music Information Retrieval	46
2.3 Ground Truth User Data from Games With A Purpose	48
2.4 MIR on Similarity Data from Surveys	50
2.4.1 Absolute Similarity Data	51
2.4.2 Class-based Similarity Data	51

2.4.3	Relative Similarity Data	53
2.4.3.1	Similarity Research on MagnaTagATune	54
2.4.4	User Preference Data	55
2.5	Adapting Computational Models to Music Data	56
2.5.1	Support Vector Machines	57
2.5.2	Neural Networks	57
2.5.3	Other Classifiers	58
2.5.4	Feature Selection and Processing	59
2.5.4.1	Feature Learning	61
2.6	Conclusion	62
3	Relative Similarity Data	65
3.1	The Similarity Graph	66
3.1.1	Determining Constraint Weights	66
3.1.2	Cycles and Inconsistent Data	67
3.1.3	Connectedness	69
3.2	The MagnaTagATune Dataset	70
3.2.1	Similarity Data	70
3.2.2	Genre Distribution over Triplets	71
3.3	Conclusion	72
4	Collecting Culture-Aware Data via Games With A Purpose	73
4.1	A Generic Framework for Game With a Purpose (GWAP)	74
4.1.1	CASimIR API and back-end Implementation	77
4.1.1.1	Participant Numbers and Song Subsets	77
4.1.1.2	Example Selection	78
4.1.1.3	Management of Player Input	79
4.1.1.4	Automatic Answers	80
4.1.2	Game Framework	80
4.1.2.1	Social Networks and Participant Login	81
4.1.2.2	Game State Model and Multi-Player Synchronisation	82
4.1.2.3	Reward Mechanisms	83
4.2	Case study: Spot the Odd Song Out	83
4.2.1	Game Modules and Interfaces	84

Table of Contents

4.2.2	Song Library	86
4.2.3	Lifecycle	86
4.2.4	User Participation Over Time	87
4.2.5	The CASimIR dataset - Data Collected with Spot The Odd Song Out	89
4.2.6	Music Similarity Data	90
4.2.7	Tempo and Rhythm Data	91
4.3	Towards Culture-Aware Similarity Modelling	92
4.3.1	A Comparative Approach for Cultural Modelling	94
4.3.1.1	Data Preparation	94
4.3.1.2	Geographic Data Subsets of CASimIR	95
4.4	Conclusions	96
5	Features for Large Online Datasets	99
5.1	Processing Features from The Echo Nest API	101
5.1.1	Chroma and Timbre Audio Features	102
5.1.1.1	Aggregation via Averaging	103
5.1.1.2	Clustered Aggregation	103
5.1.1.3	Variance	104
5.1.1.4	Normalisation and Clipping	104
5.1.2	Further Energy-based and Higher-Level Audio Features	105
5.2	Tags from Catalogue Annotations and Folksonomies	107
5.2.1	Genre Tags	107
5.3	Feature Transformations	110
5.3.1	Principal Component Analysis	111
5.3.2	Transforming Features with RBM	111
5.4	Conclusions	114
6	Computational Models for Learning Music Similarity from Relative Data	117
6.1	Mapping Features to Model Input	118
6.1.1	Sub-space Transformations	119
6.1.2	The Delta Function	120

6.2	Metric models	120
6.2.1	Support Vector Machines (SVM)	121
6.2.1.1	Weighted Learning with SVM	123
6.2.2	Metric Learning to Rank (MLR)	123
6.2.2.1	Diagonal MLR (DMLR)	125
6.2.2.2	Robust MLR	125
6.2.3	Weighted Learning with W(D)MLR	125
6.3	Adapting Methods to Relative Data	126
6.3.1	General Distance Functions	127
6.3.2	Generating Absolute Training Targets	128
6.3.3	Updating the Model	130
6.3.4	Regression	131
6.3.5	Regression Trees	133
6.3.6	Information Theoretic Metric Learning	134
6.3.6.1	Relative Learning with RITML	135
6.3.7	Relative Learning with Neural Networks (RDNNs)	137
6.4	Conclusions	138
7	A Framework for Reproducible Training and Evaluation of Music Similarity Models	141
7.1	Music Dataset Exploration	143
7.2	Similarity Data Processing and Analysis	144
7.2.1	Connectedness of Single Clips	145
7.2.2	Similarity Graphs on Clip Pairs	145
7.2.2.1	Alternative Edge Weightings	145
7.2.3	General Graph Functions and Connectedness	146
7.3	Feature Extraction	146
7.3.1	Feature Extraction Flow	147
7.3.2	Audio Features	148
7.3.3	Tag Data	148
7.3.4	Dataset Specifics	149
7.3.5	Similarity Models and Training Algorithms	149
7.4	Experiment Scripts and Result Management	151

Table of Contents

7.5	Conclusions	152
8	Evaluation	155
8.1	Strategies for Cross-Validation	157
8.1.1	Sampling for Transductive Training	157
8.1.2	Sampling for Inductive Training	158
8.2	Growing Subsets	159
8.3	General Performance	160
8.3.1	Training Set Size	162
8.3.2	Relation of Training and Test Set Performance	163
8.3.3	Training speed and efficiency	163
8.4	Feature Influence	164
8.4.1	Types of Information Contained in Features	164
8.4.2	Principal Component Analysis (PCA) processed Features and Dimensionality Reduction	166
8.4.3	Restricted Boltzmann Machine (RBM) Features	169
8.5	Sampling: Effects of Transductive Learning	174
8.6	Learning with Weighted Constraints	175
8.6.1	Weighted Performance	175
8.6.2	Effects on Unweighted Performance	177
8.7	Geographically Specific Similarity Models	178
8.7.1	Transfer Learning	179
8.7.2	Analysis of Learnt Similarity Models	181
8.8	Conclusions	186
9	Summary and Outlook	191
9.1	Thesis Background	191
9.2	Data Collection, Analysis and Preparation	192
9.3	Similarity Models: Structural Framework and Training Methods	195
9.4	Implementation and Experiments	196
9.5	Main Contributions	198
9.6	Perspectives for Future Work	200
10	Appendix	215

List of Figures

2.1	Schematic architecture of an adaptive music information retrieval system.	47
2.2	The HerdIt [5] game rewards blind input-agreement after players have input their data. Here, emotion annotation data was collected. The player blindly agreed to 70 percent with other players (the “herd”). . .	49
3.1	Graph induced by a single “odd-one-out” statement, C_k is the odd-one-out as in Equation 3.2. Vertices represent pairs of clips and edges represent the relation <i>more-similar-than</i>	66
3.2	Graph containing a length-2 cycle. Cycle highlighted in light red. . . .	67
3.3	Graph containing a length-3 cycle. Cycle highlighted in light red. Edge weights have been hidden.	67
4.1	The CASimIR framework consists of a back-end and possibly several front-ends. The API can be used by different clients simultaneously to provide annotations to the same dataset.	74
4.2	Communication of a game with the CASimIR API. Game related data such as player scores are transferred in communication between the game client and the game server. The game server requests or sends only the annotation-relevant data to the API.	76
4.3	Clip and Question Working Sets are dynamically expanded as subsets of the Song Library.	79
4.4	Entry and synchronous state progression of two game clients connected to the same match.	82
4.5	Spot the Odd Song Out (Spot the Odd Song Out) modules: Similarity. .	85
4.6	User interfaces of the TapTempo (left) and TapRhythm (right) modules.	86

4.7	Similarity data collected in a pre-phase, during the main run and the further life of Spot the Odd Song Out.	88
4.8	Logged-in users in Spot the Odd Song Out per month from February 2013 until May 2014.	89
5.1	Common signal flow diagram for music features and similarity data.	100
5.2	Tag cloud representation of the genre data added to MagnaTagATune from the Magnatune catalogue. The printed size of each genre corresponds to its frequency of occurrence in the dataset, while the spatial arrangement is not related to the genre data.	107
5.3	Graph displaying the genre hierarchies for the MagnaTagATune dataset. Arrows correspond to hierarchical subordination as deduced from the Magnatune annotations. Red nodes correspond to top-level genres.	109
5.4	Restricted Boltzmann Machine with 4 nodes in the visible (V) and $hidNum = 3$ nodes in the hidden layer (H) as well as connection weights W.	113
6.1	Comparison of three target generators. The y-axis (Δ) refers to the difference in calculated distance targets for a and b, while the x-axis denotes the current difference of a and b. Only positive values for (a-b) are denoted as Δ is only evaluated for violated constraints ($a - b > 0$).	130
6.2	Scheme for Relative Data Neural Net (RDNN) neural network learning from relative data as suggested by Braun.	138
7.1	The database components of the Culture-Aware Information Retrieval (CAMIR) framework.	142
7.2	Experiment workflow in the CAMIR framework.	152
8.1	Overall test set performance for combined features with averaged low-level information: Support Vector Machine (SVM) (SVM-Light (SVM-Light)), Metric Learning To Rank (MLR), Diagonal-restricted Metric Learning To Rank (DMLR) and RDNN performance for full features, with increasing training set size. The baseline (unweighted Euclidean distance) is plotted as dots.	160

List of Figures

8.2 Overall training set performance: SVM, MLR, DMLR and RDNN performance for full features, with increasing training set size.	161
8.3 SVM Feature performance at 12 dimensions: chroma (mean), timbre (mean), Slaney08, genre, combined features. X-axis shows increasing training set size.	169
8.4 SVM Feature performance at 52 dimensions: chroma (4 clusters), timbre (4 clusters), combined audio, genre, combined features. Increasing training set size.	170
8.5 SVM feature training performance at 12(l) and 52 (r) dimensions: Increasing training set size.	171
8.6 Transductive sampling: SVM, MLR, DMLR and RDNN test set performance for full features. The training set size increases from left to right.	174
8.7 Overall training set performance, weighted evaluation for training with: SVM, weighted SVM (WSVM), MLR, DMLR, Weighted Metric Learning To Rank (WMLR) and Weighted Diagonal-restricted Metric Learning To Rank (WDMLR). The bottom dashed curve displays the weighted baseline performance.	176
8.8 Overall weighted (E:W, -) / unweighted (E:UW, · - ·) generalisation performance for weighted training: WSVM, WMLR, WDMLR	177
8.9 Flow diagram for the fine tuning process, exemplified for the Q^{De} dataset.	180
8.10 Difference of normalised Mahalanobis matrices before (W_0 template) and after (W) fine tuning. Axes show associated features. Dark colours indicate feature weights were changed when compared to the template W_0 . Red corresponds to weight increase, blue indicates a decrease. Some blocks appear larger due to print scaling.	184
8.11 Mahalanobis matrix W after fine tuning by W_0 -Relative Information-Theoretic Metric Learning (RITML). Dark red colours indicate large weight of the feature coefficient. Some blocks appear larger due to print scaling.	185

10.1 ClipComparedGraph (Section 7.2.1) of the MagnaTagATune dataset. Vertices represent clips, undirected edges reflect co-occurrence of the clips in at least one triplet. Question triplets are clearly distinguishable. Only few clips are linked/compared to more than two other clips. Colors indicate the number of permutations a triplet has been presented in (blue=1, yellow=2, red=3). The spatial arrangement minimises edge collisions. 216

10.2 ClipComparedGraph (Section 7.2.1) of the current (01/05/2014) CASimIR dataset. Vertices represent clips, undirected edges reflect co-occurrence of the clips in at least one triplet. Less clips exist, but clips are strongly interlinked through questions. The spatial arrangement algorithm fails to unknot the many edges. 217

10.3 MagnaTagATune similarity graph (Sections 3.1 and 7.2.2) before removal of cycles. Vertices represent clip pairs, directed edges represent the relation “more similar than”. The question triplets are already completely separated. The triplets arranged along lines in the centre correspond to triplets with no inconsistent data. Lighter colours correspond to greater edge weights α . The spatial arrangement minimises edge collisions, but is not further related to data. 218

10.4 MagnaTagATune similarity graph (Sections 3.1 and 7.2.2) **after** removal of cycles. Vertices represent clip pairs, directed edges represent the relation “more similar than”. No inconsistent data is left and even more triplets have only two of three possible connections through edges. Lighter colours correspond to greater edge weights α . . . 219

10.5 CASimIR similarity graph (Sections 3.1 and 7.2.2) before removal of cycles. Vertices represent clip pairs, directed edges represent similarity relations. Several large groups of similarity are linked through transitive relations. Lighter colours correspond to greater edge weights α . The spatial arrangement minimises edge collisions. 220

10.6 CASimIR similarity graph (Sections 3.1 and 7.2.2) **after** removal of cycles. Vertices represent clip pairs, directed edges represent similarity relations. Several large groups of similarity remain linked through transitive relations even after cycle removal. Lighter colours correspond to greater edge weights α 221

10.7 Excerpt of the biggest connected component of the CaSimIR similarity dataset before cycle removal. Inconsistent similarity data are represented by two-sided arrows. Vertices represent clip pairs and are tagged with the clips' (artist A vs. artist B) each, directed edges represent the relation "more similar than". Lighter colours correspond to greater edge weights α . The spatial arrangement minimises edge collisions. 222

10.8 Graph with the 11th biggest connected component of the CaSimIR similarity dataset **after** removal of cycles. Vertices represent clip pairs and are tagged with the clips' (artist A vs. artist B) each, directed edges represent the relation "more similar than". Lighter colours correspond to greater edge weights α . The spatial arrangement minimises edge collisions. 223

List of Tables

3.1	Number of triplets with n clips sharing the same genre tag.	71
4.1	Number of annotations collected per module (8th May 2013).	89
4.2	Number of connections to other clip pairs for MagnaTagATune (MTT) and CASimIR dataset after filtering inconsistent data.	91
4.3	Number of songs being annotated at least #Annots times for TapTempo and TapRhythm.	92
4.4	Number of unique constraints and clips contained in each of the four country datasets.	96
5.1	Features from Slaney, Weinberger and White [84] used in our experiments.	106
6.1	Representation of clips, clip pairs and similarity data in terms of feature data.	119
7.1	Selection of core functionalities of the Clip, MTTclip, MSDClip and CASIMIRClip classes.	143
7.2	Similarity as stored in the comparison data variables, clip ids are relative to comparison_ids. Votes count the frequency of clip x being the “Odd Song Out”. Rows can contain further vote information to the right.	144
7.3	Front-end graph subclasses in CAMIR.	144
8.1	Average training time per dataset in minutes, accumulated over all 20 subset sizes	164

8.2	SVM Single features test set performance. Values for single average audio features and 4-cluster audio features are separated by slashes (average / 4-cluster).	165
8.3	Significance of performance differences between feature types (Wilcoxon signed rank p values). Significant values at the 5% level are set in bold type. Values for $p < 10^{-3}$ are shown as 0.000.	166
8.4	Summary of SVM single features test and training performance. The Slaney08 features are not available to 52-dimensional PCA features. .	168
8.5	Values used for the RBM grid search.	171
8.6	Parameters chosen for gradient ascent (GRAD) and SVM in the final experiments.	172
8.7	Comparison of original features and those with PCA and RBM pre-processing.	172
8.8	Test set performance per country of different learning strategies. The average performance over all countries is denoted in the rightmost column. The highest performance is highlighted per column.	180

Acknowledgements

First of all, I want to thank my supervisor Tillman Weyde for enabling me to pursue this research: The lively discussions, to which he was available beyond even PhD working hours, his constructive criticism, which was highly motivating, and his hard work on creating the working environment of the Music Informatics Research Group (MIRG), provided a carefree substrate for this work. Thanks to Andy MacFarlane for being my second supervisor, whose supportive feedback and complementing advice has added greatly to the quality of my research and enabled me to deal with academic and ethical requirements effectively. It was particularly the excellent support of Mark Firman, our research administrator with almost supernatural powers and the great help from our technical support team, that enabled my use of specialist server hardware, great travels and publications in times of structural change. Like these, Guillaume Bellec went above and beyond his “duties” as a summer intern at City to help with my research, in this case the Spot the Odd Song Out game discussed in Chapter 4. In our PhD club, Mark and my fellow PhD students Rafael and Milena made me feel as part of a familial group.

The MIRG was such a group to me. I thank Reinier De Valk for our daily musical and sociocultural discussions, his puns have changed me for life. As the group grew, Emmanouil Benetos, Srikanth Cherla and Andy Lambert extended my perspective in music research. With Son Tran I dived into new depths along our award-winning publication on feature learning. I was introduced into the world of Music Information retrieval under the aegis of Michael Clausen at the Bonn University Multimedia Signal Processing Group, by Frank Kurth and Meinard Müller during my (under)graduate studies. These friends were with me “durch dick und dünn” (through thick and thin), including Sebastian Ewert. It was also them who enabled a long collaboration with Tini, Karl-Heinz and Klaus at the Animal Sound Archive in Berlin and my research experience related to this.

I was fortunately able to pursue many collaborations, and I would like to thank Guillaume Bellec, Anders Friberg, Sebastian Stober, Antoine Winckels, Simone Stumpf

and Marilyn Niven as well as all the others who helped me at City and beyond! Special kudos go out to James Hampton, Emmanuel Pothos and the quantum psychology group for sharing their knowledge on similarity. Thanks to my fellow researchers including Brian McFee, Schultz and Joachims and Davis et al. who shared their code and knowledge free and open source, enabling a great part of the research and tools presented and shared again in this thesis. Of course, without my studentship and further funding by City University, the School of Informatics, our Graduate School, the AMR workshop and finally the AHRC Digital Media Lab Project (AH/L01016X/1), this thesis would not have been possible.

The fundamental motivation of my research, at the border of informatics to musicology and about adaptive systems lies in its interdisciplinarity. The seminars and support of Prof. Fisher and Prof. Schlüter and the Institute for Sound Studies/Musicology at Bonn University are amongst my most valuable academic experiences. Our discussions with Julia and Alexander continue to challenge and put into perspective what otherwise could remain technocentric content. The quotes accompanying this thesis try to provide this external perspective to some core ideas relevant to this thesis.

However excellent the support at University, my friends were those who helped me in times of trouble and made up the best moments throughout. I have many of those thanks to the student's union and particularly Giulio, Alan and Max Grieve. They and others encouraged me to get involved into student politics and the LGBT society, which provided a new facet and source of learning to my experience as PhD student. Through the FANG magazine for experimental literature and City University Experimental Music Ensemble I met many inspiring people and salvaged some artistic output. I particularly would like to thank Chris Cullen for great evenings, recordings and shelter when there was no other, as well as Matthias Waters to help me move there. I was glad to have Patrick Wharton for British film nights and train excursions and Benjamin Szepan for countless endless evenings, graphical game design and the love of pixelated animals, and – of course – Kevin, Fabian and a family who are constantly supporting me at home and abroad.

Abstract

Understanding how listeners relate and compare pieces of music is a fundamental challenge in music research as well as for commercial applications: Today’s large-scale applications for music recommendation and exploration utilise various models for similarity prediction to satisfy users’ expectations. Perceived similarity is specific to the individual and influenced by a number of factors such as cultural background and age. Thus, adapting a generic model to human similarity data is useful for personalisation and can help to better understand such differences.

This thesis presents new and state-of-the-art machine learning techniques for modelling music similarity and their first evaluation on relative music similarity data. We expand the scope for future research with methods for similarity data collection and a new dataset. In particular, our models are evaluated on their ability to “spot the odd song out” of three given songs. While a few methods are readily available, others had to be adapted for their first application to such data. We explore the potential for learning generalisable similarity measures, presenting algorithms for metrics and neural networks. A generic modelling workflow is presented and implemented.

We report the first evaluation of the methods on the MagnaTagATune dataset showing learning is possible and pointing out particularities of algorithms and feature types. The best results with up to 74% performance on test sets were achieved with a combination of acoustic and cultural features, but model training proved most powerful when only acoustic information is available. To assess the generalisability of the findings, we provide a first systematic analysis of the dataset itself. We also identify a bias in standard sampling methods for cross-validation with similarity data and present a new method for unbiased evaluation, providing use cases for the different validation strategies.

Furthermore, we present an online game that collects a new similarity dataset, including participant attributes such as age, location, language and music background. It is based on our extensible framework which manages storage of participant input, context information as well as selection of presented samples. The collected data enables a more specific adaptation of music similarity by including user attributes into similarity models. Distinct similarity models are learnt from geographically defined user groups in a first experiment towards the more complex task of culture-aware similarity modelling. In order to improve training of the specific models on small datasets, we implement the concept of transfer learning for music similarity models.

Style and Notation Conventions

This thesis uses the authorial “we” when referring to the contributions of its author. Parts where the we includes other persons will be explicitly identified as such in the text.

Self-citations of the author are indicated by prefixed publication numbers, e.g. [pub:10], which refer to the list of publications on page 28. Chapters and sections which contain parts of previously published material are marked accordingly by means of margin notes.

Notation

Music clips are referred to as $C_i, i \in \mathbb{N}$. Vectors $x \in \mathbb{R}^N$, for $n \in \mathbb{N}$ are column vectors by default, and vectors $x_{(i)} \in \mathbb{R}^N$ with a brace-enclosed index correspond to an instance belonging to C_i . Indexing into vectors is denoted by the second subscript level, e.g. $x_{(i)_k} \in \mathbb{R}$ refers to the scalar component of vector $x_{(i)}$ at position k . Transposed matrices and vectors are denoted by x^\top .

Where software and programming frameworks are discussed, uniformly spaced typewriter font will be used to refer to entities in programming language that may be found in the discussed software or framework.

List of Abbreviations

7digital Seven Digital Online Music Service.

CAMIR Culture-Aware Information Retrieval Framework, Chapter 7.

DBN Deep Belief Network.

DMLR Diagonal-restricted Metric Learning To Rank, Section 6.2.2.1.

FRI Dataset from Friberg and Hedblad [25].

GBT Gradient Boosting Tree.

GWAP Game With a Purpose, Section 2.3.

ID-sampling Inductive Sampling, Section 8.1.

ITML Information-Theoretic Metric Learning, Section 6.3.6.

MFCC Mel Frequency Cepstral Coefficient.

Million Song Dataset Million Song Dataset.

MIR Music Information Retrieval, Section 2.2.

MLR Metric Learning To Rank, Section 6.2.2.

MySQL MySQL.

PCA Principal Component Analysis.

RBM Restricted Boltzmann Machine, Section 5.3.2.

RDNN Relative Data Neural Net, Section 6.3.7.

RITML Relative Information-Theoretic Metric Learning, Section 6.3.6.1.

Spot the Odd Song Out The Spot the Odd Song Out game, Section 4.2.

SVM Support Vector Machine, Section 6.2.1.

SVM-Light SVM-Light implementation, Section 6.2.1.

TD-sampling Transductive Sampling, Section 8.1.

WDMLR Weighted Diagonal-restricted Metric Learning To Rank, Section 6.2.3.

WMLR Weighted Metric Learning To Rank, Section 6.2.3.

List of Publications

- [pub:1] Bellec, Guillaume, Friberg, Anders, Wolff, Daniel, Elowsson, Andreas and Weyde, Tillman. "A Social Network Integrated Game Experiment to Relate Tapping to Speed Perception and Explore Rhythm Reproduction". In: *Proceedings of the Sound and Music Computing Conference*. [In collaboration with the KtH Stockholm. The author's contribution consists of the CASimIR framework and similarity module]. 2013, pp. 19–26.
- [pub:2] Tran, Son, Wolff, Daniel, Weyde, Tillman and Garcez, Artur d'Avila. "Feature Preprocessing with Restricted Boltzmann Machines for Music Similarity Learning". In: *Audio Engineering Society Conference: 53rd International Conference: Semantic Audio*. [In collaboration with the Machine Learning Group at City University. The first two authors share equal contribution to the work]. 2014.
- [pub:3] Wolff, Daniel, Bellec, Guillaume and Weyde, Tillman. "A Music Similarity Game Prototype Using the CASimIR API". In: *Proceedings of ISMIR 2012, Demo*. Porto, Portugal, 2012.
- [pub:4] Wolff, Daniel and Weyde, Tillman. "Adapting Metrics for Music Similarity Using Comparative Judgements". In: *Proceedings of ISMIR 2011*. 2011, pp. 73–78.
- [pub:5] Wolff, Daniel and Weyde, Tillman. "Adapting Similarity on the MagnaTagATune Database: Effects of Model and Feature Choices". In: *Proceedings of the 21st international conference companion on World Wide Web. WWW '12 Companion*. Lyon, France: ACM, 2012, pp. 931–936.
- [pub:6] Wolff, Daniel and Weyde, Tillman. "Combining Sources of Description for Approximating Music Similarity Ratings". In: *Proceedings of AMR 2011*. Barcelona, Spain, 2011, pp. 114–124.
- [pub:7] Wolff, Daniel and Weyde, Tillman. "Learning Music Similarity from Relative User Ratings". In: *Information Retrieval (2013)*, pp. 1–28.

- [pub:8] Wolff, Daniel and Weyde, Tillman. "On Culture-dependent Modelling of Music Similarity". In: *Proceedings of Fourth International Conference of students of Systematic Musicology Sysmus*. Cologne, Germany, 2011.
- [pub:9] Wolff, Daniel, Stober, Sebastian, Nürnberger, Andreas and Weyde, Tillman. "A Systematic Comparison of Music Similarity Adaptation Approaches". In: *Proceedings of ISMIR 2012*. 2012, pp. 103–108.
- [pub:10] Wolff, Daniel, Bellec, Guillaume, Friberg, Anders, MacFarlane, Andrew and Weyde, Tillman. "Creating Audio Based Experiments as Social Web Games with the CASimIR Framework". In: *Audio Engineering Society Conference: 53rd International Conference: Semantic Audio*. 2014.

- to my family and friends
in memory of Ziggy Tomasz Zygmunt -

The development of the internet has more to do with human beings becoming a reflection of their technologies. [...] After all, it is we who adapt to the machine. The machine does not adapt to us.

(Friedrich Kittler, 2006)

1 Introduction

This thesis is devoted to models of music similarity that can be adapted to user data. In a time of globally networked communication, facing floods of continually produced data, we, where we operate within the digital realm, rely on automatic processes for the retrieval of information relevant for our endeavours. With the massive user gain of online services and the advent of the “Web 2.0”, online music providers and integrated social networks for music became successful ventures with millions of users¹ and large music turnarounds². Such networks and platforms presented a need for automatic means of music organisation and retrieval, similar to the sophisticated tools of web search engines. For example in music recommendation, given a piece of music in a query, a user wants to find relevant music in the database. Music similarity is a concept necessary for most structured organisations of music, although it might be referring to different facets and associations of items compared. The facets to be modelled depend on the application of the similarity measure. They include meta-data such as musical genre for cataloguing, musical preference and demographics for recommendation and human statements from surveys for analysis and modelling of the similarity assessment by listeners.

Meanwhile, the strong involvement of the users of online music networks presents the opportunity of substantial user feedback data to be analysed and integrated

¹<http://press.spotify.com/se/information/>

²<http://www.apple.com/pr/library/2010/02/25iTunes-Store-Tops-10-Billion-Songs-Sold.html>

into the relevance estimation. Capitalising on this, commercial approaches to music recommendation primarily use collaborative filtering, the quasi-standard approach for online recommendation: “Other customers who bought this also purchased the following . . .”. The drawback of this approach is that it relies on user annotations. As has been pointed out by Celma [17], among other drawbacks such as the tendency to foster artist hubs, this approach fails when there is no or only little annotation available, for example for newly published music.

Research in Music Information Retrieval (MIR) has resulted in strong alternatives for methods of organising digital music collections, and continues to do so. Classic techniques for accessing music similarity compare music using fixed mathematical measures based on acoustic features or tag data. The practical disadvantage of such approaches is that their performance tends to hit a *glass ceiling*, where it becomes difficult to raise performance although the task is solvable with better performance by humans. Referring to the quote by Kittler heading this chapter, it is also the rigidity of the underlying techniques and cultural assumptions in the system’s design that are perpetuated and imprinted into its users and therefore such systems might sustain cultural assimilation.

To address these issues, relevant information of context and user of the application can be included in the evaluation of e.g. a music recommendation task. This thesis presents a step towards more flexible and adaptive music retrieval systems, whilst provisioning for further musicological analysis of models trained with our approaches.

Using novel and state-of-the-art machine learning techniques, our models are enhanced by using self-reported music similarity perception as the basis for similarity modelling: We use relative similarity data provided by participants of a game who “spot the odd song out” of three songs. We show that this method allows for efficient data collection and modelling of similarity. Apart from predicting users’ choices, the models can be used to provide estimates for the similarity of music clips which is necessary for recommendation.

Furthermore, the training process allows for adaptation to user groups and resulting models can be analysed to reveal statistical correlations of musical features

1.1 Aims of this Thesis

and reported similarity. Personalised recommendation of music based on previously associated user groups – determined for instance by cultural context – is made possible by integrating user attributes into the training data. A similarity model should learn well from annotated music data and generalise to a determined, possibly broad set of music genres.

Efforts have been increased in recent years to adapt retrieval to specific cultures, contexts and individual users, as in the CompMusic project [82], the work of Kaminskas, Ricci and Schedl [41] or Park, Yoo and Cho [70]. Context-based and user-adapted retrieval have become popular research goals, following and fostering developments in machine learning to provide algorithms applicable to accumulated user data. A key incentive for this development is the growing amount of data collected on user preferences and behaviour while browsing web pages. Namely, click-through data for ranked search results, playlists and social network based crowd wisdom is now integrated into general classification and distance measure learning tasks. Especially within social networks, new opportunities are being explored using Games With a Purpose (GWAPs), where data is collected while the participants are playing a game.

In this thesis we will use a GWAP to collect similarity and other data, in conjunction with attributes of the players. We will present a framework for collecting the data and strategies necessary for efficient design and operation of a GWAP. Especially social networks today allow for the provision of rich information by the participant of a game. We will use such information to create a country-annotated similarity dataset as well as related comparative experiments on similarity models of geographical regions. In the following, the precise aims of the thesis are laid out, framing the content of the research underlying this work.

1.1 Aims of this Thesis

The overarching goal of this thesis is to contribute and evaluate methods for learning models from relative music similarity data (see Section 1.1.3), with a user-

and culture-centred perspective. This poses the following research questions concerning the three tasks of data gathering, similarity model adaptation and model evaluation.

1.1.1 Research Questions

rq:1 What are the best similarity models, training methods and features for predicting music similarity from relative data? Can we improve similarity modelling by integrating additional user attributes?

rq:2 How can we improve or develop new methods for learning from relative data?

rq:3 How can we reproducibly evaluate similarity models and what are relevant qualitative and quantitative differences? How to proceed where music audio content is not freely available?

rq:4 How can we analyse and describe relative similarity data and what are relevant data properties? Is there a bias in the MagnaTagATune dataset (the largest available dataset so far)?

rq:5 How can we efficiently and sustainably acquire further similarity data or other music annotations? Can we introduce more control into web-based data collection?

The above research questions motivate the aims of this thesis which are now presented and ordered according to the significance of the associated contributions.

1.1.2 Adapting Models to Relative Music Similarity Ratings

We present several new and state-of-the-art methods for learning a music similarity measure from user similarity ratings. General methods for similarity modelling and adaptation form a central prerequisite to adaptation to specific contexts. Some general purpose algorithms are available for accomplishing the task of adapting a distance measure to training data. The options narrow down considerably when relative similarity input data is concerned. Still, for reasons discussed in the fol-

1.1 Aims of this Thesis

lowing, this data type presents an opportunity for new and efficient methods for the adaptation of similarity models to human ratings and perception.

To this end, we examine and evaluate existing machine learning methods for training distance metric models to relative similarity data (*rq:1*). We furthermore introduce several new algorithms, derived from general regression methods as well as based on neural networks, and compare their performance to the existing methods. The new methods allow for further learning from weighted similarity data and a novel transfer learning method for music similarity (*rq:2*). The methods are integrated into a general methodical framework, which is also implemented as open source programming framework.

This framework enables us to analyse the effectiveness (*rq:3*) of different training methods in our evaluation, including new and state-of-the-art algorithms. A special focus is put on the generalisation of learnt models to unseen data. We explore the influence of musical information helpful for similarity learning by comparing the performance of different isolated and combined features. Furthermore, other aspects of the similarity data such as weights and sampling considerations are evaluated. This allows for a better understanding of different methods, features and the relative similarity data itself which can inform future research on the topic.

Based on the above general adaptation strategies for models of music similarity, cultural information can be introduced into similarity modelling methods via user attributes. The assumption is that such an adaptation will lead to better models for the various groups, enhancing the similarity estimation performance. This can be done by for instance combining several models previously trained for cultural subgroups, or including cultural attributes directly into the similarity models. In this thesis we will lay the foundations for such models by providing user-attribute annotated data as well as a first dataset and experiment on geographical similarity models, using the similarity and user data collected by the approach described below.

The interdisciplinary research outputs in this thesis potentially appeal to a wide range of researchers, commercial applications and future users. As we motivate

above, new similarity models have applications in systems for music recommendation and indexing of media catalogues. They assist users of such systems with more personalised and potentially more intuitive search results. Our new algorithms for adaptation of similarity models add to the corpus of general distance learning methods in machine learning, and enable further research on the modelling of similarity data for Music Information Retrieval (MIR) researchers. Information contained in learnt models motivates research in musicology and music psychology regarding influence of certain music features on similarity perception. This can be extended to comparative research between user groups or cultures.

1.1.3 Analysis of Relative Similarity Data

Although relative similarity data is not as readily accessible as customer preference or social network data, it provides a valuable change of focus from general classification and recommendation success towards modelling musical similarity and the users' perception of it when engaged in a comparison task. Thus, instead of targeting a general relevance criterion e.g. for music recommendation, the optimisation tasks tackled in the following address reported perceived similarity, which only constitutes one of the many variable aspects of relevance.

The modelling of similarity perception itself and the appropriate pre-processing of collected similarity data has only recently been discussed in Music Information Retrieval (MIR). Users provide information by spotting the "odd song out" of three songs. We derive relative similarity data of the form "Song A is more similar to Song B than to Song C", represented as relative constraints. Using models trained with this data, we can predict the odd song out, but also evaluate the similarity of arbitrary music clips.

Relative similarity data has rarely been used in MIR (see Section 2.4.3). To assess the quality of the MagnaTagATune similarity dataset, we recollect the methods available for analysis of relative similarity data. We report methods for analysing and preparing relative similarity data (*rq:4*) using graph theory. The methods are adapted and a first analysis of the MagnaTagATune relative music similarity dataset, the only one available at the time of this study, is presented. In order to enable

further research and reproduction of the experiments reported here, the code for preparing similarity data, model training and evaluation is published online as open source.

1.1.4 A Game Framework for Data Collection

More open similarity data is needed in order to evaluate different strategies of adapting similarity models, particularly for machine learning methods. To our knowledge there currently is no large music annotation dataset freely available containing information about the providing participants.

To address this issue, we here have developed an open source framework for collecting music annotations via games with a purpose (*rq:5*). High numbers of collected data entries are encouraged by different gamifications of a standard odd-one-out music similarity survey, also known as triad survey in psychology and anthropology (e.g. Kelly [43]). The core of this framework, originally named the Culture-Aware Similarity Information Retriever (CASimIR), is given by a web service providing selected sets of music clips to a front-end application and collecting the user votes in a growing dataset. We thereby enable an easy exchange or parallel usage of several music similarity games and user interfaces, requiring no recoding on the similarity collection. A series of several mini-games has been developed in cooperation with KTH Stockholm. Here, amongst other strategies, an approach rewarding user-agreement between team members is used. As the game is deployed on the online social network Facebook, built-in mechanisms enable users to recommend the game to their friends.

Our framework further establishes the GWAP data collection paradigm for data-driven research and makes implementation – and thus large scale dynamic data collection – possible to potential researchers and users who are not specialised in web development.

1.1.5 A New Relative Similarity Dataset

Our new CASimIR dataset satisfies requirements (*rq:5*) discovered as necessary when using the MagnaTagATune database (see Section 3.2): There is a minimum required number of users which are answering the same survey - for example an odd-one-out instance of three clips. Furthermore, the triplets presented in the surveys should overlap in the clips they contain. The genres of the clips presented in the surveys are also controlled in the CASimIR API, to guarantee a better quality of the resulting dataset. This is with regard to expected training performance on the data and to allow for an interpretation of the learnt models.

The dataset enables further research in similarity modelling for MIR and music perception researchers, including comparison with the only similar dataset (MagnaTagATune, *rq:4*) and evaluation of similarity models including user attributes: For experiments with cultural attributes, a first country-annotated similarity dataset is presented and used for testing new methods of transfer learning.

1.1.6 Music Features For Large Datasets

When using commercial pop music for research, evaluation of similarity models for music whose audio content is not freely available becomes difficult: For modelling music similarity, the music itself has to be represented to the model and training algorithm. Feature design is a classic task in Music Information Retrieval, and various open source toolboxes exist for extraction of standard features¹. We here show how features for large datasets can be extracted based on pre-computed features from The Echo Nest API. Given that today digital commercial music is still subject to highly restrictive copyright, this allows researchers without direct access to the audio to perform large-scale experiments (*rq:1*). The effectiveness of standardised content-based features and genre tags for similarity learning is evaluated. The evaluation and analysis of similarity models regarding the importance of features provides insights into the correlation of extracted features with the similarity data used for training the model.

¹<https://www.jyu.fi/hum/laitokset/musiikki/en/research/coe/materials/mirtoolbox>

Our analysis includes feature transformations using PCA and RBM. We use the PCA transformation for fixing feature dimensionality whilst changing the contained information. With the RBM transformation we examine a novel approach of feature transformations that helps even simple learning methods to model complex relationships in the data.

1.2 Contributions

The following is a list of the methods and frameworks contributed by the author of this thesis.

- Methods for deriving audio- and tag-based features from free online API's for large datasets
- Methods for similarity graph analysis including building and pruning of similarity relation graphs from odd-one-out experiments
- A thorough analysis of the MagnaTagATune dataset using these methods
- A framework and game for collecting music annotations with user attributes via the web and social networks
- The CASimIR similarity dataset, and a country-annotated dataset derived from it
- A general and extensible framework for training and evaluation of music similarity models on large-scale databases
- A number of new and adapted methods for learning from relative data :
 - A methodical framework for relative similarity learning, allowing for integration of absolute similarity learners
 - The WMLR/WDMLR method for learning from weighted relative similarity data
 - A new approach of using RDNN for similarity learning
 - The RITML method for learning weighted relative similarity data, and W_0 -RITML for transfer learning with similarity models
- The *inductive sampling* method for unbiased sampling of relative similarity data for cross-validation

- A PCA-based evaluation of influence of feature information with constant dimensionality
- An approach for comparative analysis of culture-based similarity models, using transfer learning with RITML

Some of the methods and figures presented here have been previously published by the author of this thesis. Such own publications are listed on page 28. Relevant publications furthermore appear at the start of respective sections. This includes parts of text which, if appropriate, have been transferred into the document at hand.

1.3 Thesis Overview

We now summarise the structure of this thesis. The following Chapter 2 introduces the reader to the background of this thesis, covering music similarity and its embedding in Musicology, Music Information Retrieval (MIR) and psychology. We highlight the importance of similarity models within the paradigms of Music Information Retrieval, list computational training methods and refer to their wider usage in MIR applications. Also, different types of similarity data are distinguished with their means of acquisition such as Games With A Purpose.

Chapter 3 presents analysis and processing techniques for relative similarity data, as derived from odd-one-out statements and a representation through clip-pair graphs. The usage of methods for graph analysis enables the identification and filtering of inconsistencies between participant data entries. We then apply the analysis on the MagnaTagATune dataset and analyse its properties for similarity training including data point connection and musical genre.

The shortcomings of MagnaTagATune and the absence of alternative datasets encouraged us to start a new dataset collection and development of the CASimIR framework for collecting media annotations with Games With A Purpose described in Chapter 4. This enables the development and usage of several user interfaces on the basis of an independent central data back-end, allowing for efficient and

reusable design of data collection interfaces. We present an early overview of first data collected, including a country-specific relative similarity dataset.

Chapter 5 introduces processing methods on acoustic, cultural and metadata features for learning music similarity on large databases. We extend the MagnaTagATune dataset using genres annotated by the Magnatune label. Also, feature post-processing with PCA is discussed and a new method using RBMs is introduced.

Based on these features, Chapter 6 discusses similarity models with new and state-of-the-art adaptation methods for relative similarity data. An abstraction of facet difference vectors to model differences in clip pairs is described before we discuss metric and neural-net-based model architectures for learning distance measures as dual representations the similarity models. We also integrate a framework for enabling learning from relative data for methods designed for absolute data to our framework of similarity learning, resulting in new and more flexible model training methods.

Chapter 7 describes our CAMIR framework including code for all methods presented in this thesis. The experiment part of the framework manages a typical similarity learning workflow, and includes third-party implementations of state-of-the-art metric learning methods besides the new implementations of RDNN and RITML.

We present a first comprehensive evaluation of these similarity models and training methods in Chapter 8. This includes a new evaluation strategy, based on cross-validation. We evaluate generalisation performance of algorithms, feature influence and transformations as well as learning from weighted similarity data. Our experiments conclude with an application of our new W_0 -RITML method to culture-aware similarity modelling using four different similarity datasets divided by country.

We allow for a final summary of contributions and results in Chapter 9 only to point out the possibilities the methods presented here may yield for future music research and music information retrieval.

It is no accident that Gutenberg's moving letters have been called history's first assembly line. For it was the compiling of drawings and lettering, and of [...] instruction manuals, which first made it possible for engineers to build further and further on the shoulders – or rather on the books – of their predecessors, without being in any way dependent on oral tradition.

(Friedrich Kittler, 1999)

2 Fundamentals & Related Work

This section covers the interdisciplinary background of the thesis, discussing the concept of similarity in general. We discuss how it can be and has been applied to music, how it is related to perception, and how it can be modelled using machine learning techniques. This wide context of similarity and the limitations of scope for this work require a certain focus on methodical precursors and limitations of our approach, whereas we provide a more detailed report on the embedding into Music Information Retrieval (MIR). Apart from the introduction above and in the following chapters we assume the reader has a general knowledge of basic techniques of MIR such as feature extraction and classification techniques and refer to the references provided for further introduction.

2.1 A Perceptual Perspective on Music Similarity

We here consider similarity as a relation between entities of somehow comparable nature. As a general term, similarity can refer to different relations depending on the context of its usage and the entities it is applied on. In science, similarity is often defined as the numerical closeness of two values or vectors in a vector space where a distance measure exists. This mathematical definition of similarity is used in Chapter 6 to define computable models of similarity and their training and

evaluation algorithms. Still, in the practical application of these models, it is the entities which are represented by the numbers, or features, which hold a meaning to the human user and assure the relevance of the resulting similarity models. Thus, in psychology similarity refers to the perceptual closeness of physical or imagined entities. Music similarity, in the fields of Psychoacoustics and Cognitive Musicology, then refers to the perceptual closeness of music clips or songs. Within this frame, similarity models can be thought of as modelling perceptual features of music as well as cognitive processes involved in the assessment the common and similar features as well as differences between two clips.

Apart from the actual acoustic content of the music there exists a multitude of factors influencing similarity perception. These include external factors such as context of listening and surrounding environment, factors related to cultural en-trainment such as music education and familiarity with particular music examples and finally personal factors including mood, attention span as well as physical capabilities. For a comprehensive overview on such factors for the context of music preference we refer to Leblanc [48] as well as more recently [11, 55].

Most computable similarity models are based on features, as proposed by Tversky [94]. In Tversky's approach, perception of similarity depends on the accumulated similarity of various single (in his case binary) features in the compared objects. In this thesis we combine acoustic measurements with cultural genre descriptions of music and train models against human similarity ratings. In Section 4.3.1 we even add participant attributes to the similarity data. A common mathematical approach is nowadays to view the features as dimensions of a vector space and model dissimilarity as a distance measure, e.g. using the Euclidean or other metrics.

Distance measures normally treat the dimensions uniformly, which ignores the different natures of features and their relations, e.g. the aspect of systematicity as pointed out by Gentner and Markman [29]. This can be addressed to some degree by using a Mahalanobis distance [54] (see Section 6.2), which models correlations between features.

2.1.1 The Symmetry Assumption

Distances in vector spaces are normally symmetric, and metrics are symmetric by definition. However, Tversky [94] already pointed out that similarity perception may be asymmetric. In music perception, asymmetry can be expected, because two comparable clips are presented sequentially and order may play a role. Gentner and Markman [29] relate asymmetry to prototype-instance relationship of objects to compare.

For example, the Beatles may function more like a prototype in popular music than Oasis, as they are more popular and profit from historic precedence. Yet both bands share similar music in their repertoire. Thus, a recording by Oasis might be expected to sound more like The Beatles than The Beatles sound like Oasis.

Yet, most mathematical and computational similarity models so far are symmetric. This is due to the simplification that symmetry brings to practical and theoretical aspects of the model. Considerations of the mode of data collection and the information available in the data also make a symmetric model a reasonable choice.

2.1.2 Similarity in Music Research

In musicology, similarity of music is explicitly or implicitly encompassed in many different applications, including the distinction of repetition and variation or the authenticity of a work or recording. Here, judgements are often made based on scores, lyrics, or other representations of music which are different from the acoustic perception of a musical performance. This is reflected in the features, the description of music given to similarity models, which are already used in automating musicological classification of folksongs [4]. Many methods require a close reading by the musicology expert, which follows established paradigms of classification and judgement, but also apply heuristics. Furthermore, music theories as presented by Lerdahl and Jackendoff [50] and musicological classifications of works are related to our perception, be it through a basis in psychoacoustic principles or learnt behaviour. The work of Lomax [52] followed an ethnographic approach by associating song structures and singing styles to distinct cultures. Their research towards a

“Global Jukebox”, included the annotation of features to songs and dances from many world cultures, allowing those cultures to be compared by their musical repertoire and practices.

The methods from Music Information Retrieval reported in the following often use a combination of music representations, rules and automatically trained classifiers. Furthermore, especially in the context of music recommendation, the term similarity is often used to refer to the relevance a piece of music has to a user. In this case, some approaches via collaborative filtering techniques use only commonalities in listeners for a certain piece of music to infer the similarity of music.

In this thesis, contrasting most existing works in MIR, we explicitly and solely define and evaluate similarity based on similarity data of human listeners. The general computational framework provided can be used to statistically model and predict *reported similarity* given learnt correlations with the knowledge and music representations fed into the model. The evaluation on purely reported listener perception data encourages the use of diverse representations of music, including cultural context of both music and the recipient, to predict reported music similarity. The quote from Kittler heading this chapter now points us the practical implications of trainable models: Being able to externalise information about (or perception of) music similarity by storing it into similarity models allows for the automatisisation of social acts such as the recommendation of similar music.

2.2 Music Information Retrieval

The context of this study is Music Information Retrieval (MIR), where a standard architecture for adaptive systems as sketched in Figure 2.1 has become prevalent for information retrieval involving audio data [13, 16, 69]. In this architecture, an audio clip is analysed with regards to a number of features (see Chapter 5) using a diverse range of signal processing methods. The features are presented as a single vector per audio clip, representing a range from low-level features like loudness to higher level properties, for instance key or tempo. The audio features can be complemented with professionally produced metadata and user

2.2 Music Information Retrieval

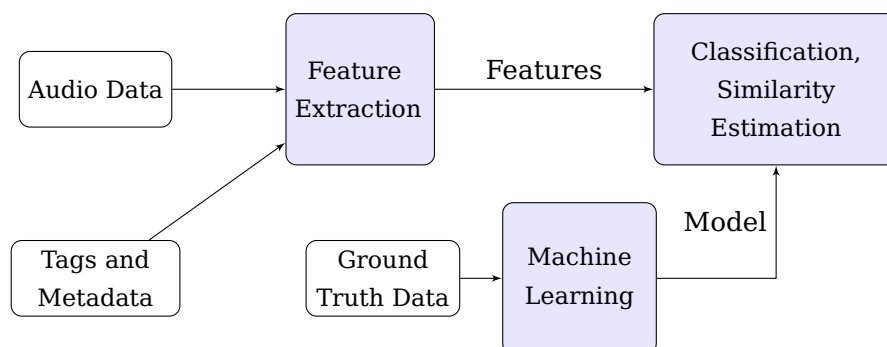


Figure 2.1: Schematic architecture of an adaptive music information retrieval system.

annotations. When a query is processed, a matching process takes place, that typically involves classification or similarity.

Traditional content-based approaches model similarity with regards to information from audio data. They have been shown to work well in some scenarios, and are now being used on a wider scale in web services like The Echo Nest [39] or The Freesound Project [2]. Content based music similarity models need to incorporate the extraction of acoustic and psychoacoustic attributes derived from audio and music theoretic information. The applicability of such extraction methods and the models themselves is highly dependent on the context of the music, the application, and the user. Learning models including information outside the content-domain can help adapt the system to the users' needs and the designers' intentions for music where user data are not available. Such models should generalise from limited amounts of user data to a larger set of music.

In adaptive systems, the classification or similarity model is optimised, typically using supervised machine learning techniques. Ground truth that is used for training and evaluation of models consists of information on actual class membership or similarity data, against which the the adapted system is evaluated, typically with cross-validation. From this perspective we now discuss general and music specific work on similarity models, methods for collecting similarity data, and computational methods to learn from the data.

2.3 Ground Truth User Data from Games With A Purpose

The type and provenance of ground truth data needed depends highly on the task to be solved by the algorithms. For learning from and understanding human behaviour, data collected from humans themselves is very promising. The classical method of obtaining such data is via surveys, as the following sections on previous similarity experiments will show. While providing high control over the set and setting of data capture and participants, the cost of traditional surveys is quite high when comparing the cost to data points ratio. For many practical applications, today, different trade-offs can be achieved which allow for more data to be collected, and more advanced methods of remote supervision and control of data input have been discovered.

Potentially accessible by a larger amount of participants, web-based approaches have become increasingly popular for collecting new music-related human input. This includes surveys for sound quality as by Foster, Mauch and Dixon [24]¹, but also relating music to psychological values such as emotions² ³, associations of nonmusical entities like locations⁴ or stories⁵. In this context, new interfaces are also being explored, allowing for user interaction with and manipulation of content such as in DarwinTunes [53], where users influence the evolution of automatically generated music. Another example is the Songle web music interface [30], which provides means for direct content annotation and interaction between users. The project CURIO plans to launch a crowdsourcing service offering interfaces for different areas of research (including chemistry, history, biology) [46].

When such data collection interfaces become gamified – i.e. they utilise enjoyment to motivate data entry, employ game rules or facilitate playful or competitive user interaction – they are Game With a Purpose (GWAP). GWAP can be used to collect many types of ground truth data, but their key difference to traditional collection strategies lies in the participants' motivation: In a GWAP, data are provided as

¹<http://webprojects.eecs.qmul.ac.uk/matthiasm/audioquality-pre/check.php>

²<http://www.isophonics.net/content/music-and-emotion-listening-test>

³<https://www.bbcarp.org.uk/m4/UserTrial/>

⁴http://lamj.inf.unibz.it:8180/music_for_places/

⁵<http://marg.snu.ac.kr/radio/recommender.php>

2.3 Ground Truth User Data from Games With A Purpose

a side effect of interactive game play, thus motivation is based on the player's enjoyment of the game. Successful GWAP facilitate the annotation of large data quantities, carried out by the self-motivated participants. Examples of well known GWAPs are the Google Image labeller¹ and the "GWAP website"² which includes the ESP game, Verbosity and TagATune amongst others.

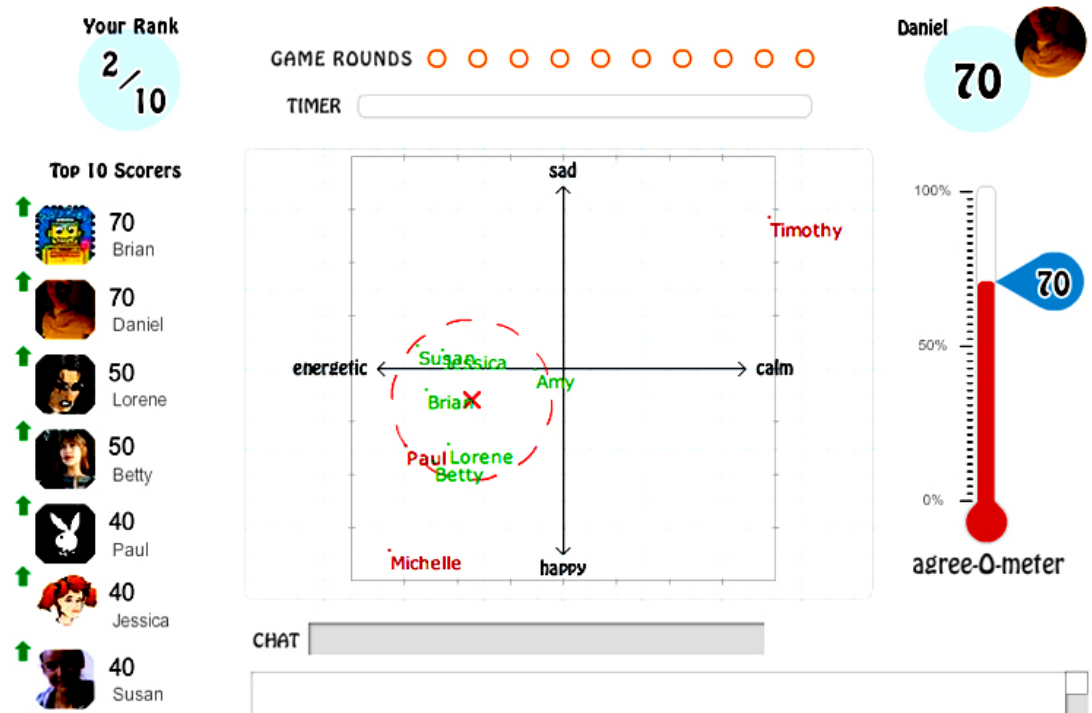


Figure 2.2: The HerdIt [5] game rewards blind input-agreement after players have input their data. Here, emotion annotation data was collected. The player blindly agreed to 70 percent with other players (the "herd").

Given the indirect motivation of participants, different strategies have been employed to achieve the collection of sound data. Three forms of GWAP are distinguished by Von Ahn and Dabbish [98]: output agreement games, inversion-problem games and input-agreement games. In output agreement games, participants are given the same input media and cannot communicate with other participants. Success is defined by the participants deciding on the same output. A popular example of this form is the ESP game where participants are given an image and try to

¹<http://www.seroundtable.com/google-image-labeler-dead-14663.html>

²<http://www.gwap.com/gwap/>

provide a descriptive tag which agrees with the other participant. In the Spot the Odd Song Out game, (see Section 4.2) we use this strategy to collect similarity information for triplets of music clips. In Figure 2.2, the HerdIt [5] game uses this strategy to collect tags and mood annotations for popular music. In input agreement games, players need to guess whether or not they have been given the same inputs. An example of this is TagATune [45] where success is determined by two participants guessing whether they have been given the same song based on the tags they provide. Inversion-problem games, such as Verbosity [99], allow a subtle form of communication where a ‘describer’ provides some information which is given to a ‘guesser’, who tries to guess the described concept.

When designing a GWAP for a specific application, as with traditional controlled paper surveys, particular attention has to be given to the presentation of questions and the general interface. In addition, the motivation of the player to provide the data, be it more or less explicit, has to be carefully chosen in order to minimise the bias for the intended interpretation of the collected data. Note that the issue of motivation is not limited to GWAP, but future research is needed to allow for analysis of the impact different standard GWAP practices have on data quality and bias. This promises to improve the comparison between traditional surveys and GWAP approaches at the design stage of a data collection.

Similarity data is just one of many data types that can be collected using GWAP, but will be the central focus of this thesis. The following section will discuss existing types of similarity data and how they can be learnt with machine learning methods.

2.4 MIR on Similarity Data from Surveys

Although there are many publications in MIR related to similarity, most deal with related topics such as relevance as there is only few similarity data available. Before we discuss such potentially applicable methods, we introduce the similarity data types we consider survey data most closely related to *perceived similarity*. The following discussion of related research is organised by data type, namely absolute similarity data, class-based similarity data, and relative similarity data.

2.4 MIR on Similarity Data from Surveys

We then go on to discuss music preference data leading to general methods for similarity modelling.

2.4.1 Absolute Similarity Data

Many surveys collect *absolute similarity data* by asking for similarity ratings of two clips on a fixed scale, as in e.g. Ferrer and Eerola [23]. The MIREX Audio Music Similarity and Retrieval task uses absolute similarity data collected from humans for evaluation of submitted similarity estimation algorithms. Competing algorithms return a ranked list of similar results to a query song, and are judged based on the human similarity ratings given the five most similar results according to the specific method, omitting music of the query's artist. The data is collected via the Evalutron6000¹ web interface, which lets the participant enter both a rough-scale (Not Similar / Somewhat Similar / Very Similar) and a fine-scale (0-100) similarity value. The human similarity judgements can be downloaded while the audio is restricted by copyright. The usage of absolute similarity data is a common approach in psychological and music research. It allows subjects to make statements on their perceived similarity on potentially very fine scales. A critical aspect of experiments with similarity data is the order in which clips are presented. Especially the gradual changes of the participants' statements can be affected by the history of clips already listened. Such bias can be accounted for, e.g. by randomisation of the order of presentation. An important factor in similarity data collection is the number of total annotations as well as their distribution over the clip pairs: Collecting a large number of annotations for the same pairs allows for more stable results, but a large set of potential clip pairs needs to be covered as well.

2.4.2 Class-based Similarity Data

An alternative approach is to gather *class-based similarity data* by asking subjects to classify clips by assigning them to one of a fixed number of unlabelled classes (e.g. Musil, El-Nusairi and Müllensiefen [65] in their musicality test). This type of

¹http://www.music-ir.org/mirex/wiki/2011:Evalutron6000_Walkthrough

experiment typically requires choosing an appropriate number of classes beforehand, and does not solve the problem of inter and intra class similarities. The assumption is here that distances within classes should generally be smaller than distances between classes. This leaves class information as a provider of usually very rough similarity data. Also, depending on the number of classes, class-based data often contains relatively little information. However, consistency is less problematic in class information, which is a standard part of many music datasets, e.g. genre labels. Therefore, it is interesting to use class information to adapt similarity models.

A considerable range of distance learning methods has been used for learning from class information, including Linear Discriminant Analysis, nearest-neighbour-based optimisation, and kernelised approaches such as Support Vector Machines (SVM) [20, 56, 101, 105].

E.g. Novello et al. [68] apply this in their perceptual evaluation of music similarity. They collect relative similarity judgements from 36 participants on triplets of songs, and find a positive correlation of users' similarity ratings with class data using musical genres. However, this is not by design, as class data are normally not designed to model similarity, but to represent other, often cultural, criteria.

In [4], Bade et al. use expert classifications of folk song melodies for training localised similarity measures on folk songs. The melody type classifications can be interpreted as relative similarity constraints: Assuming similarity to follow the categorisation as above, songs of the same category can be expected to be more similar than songs being assigned different categories. This information is then applied to learn a linear weighting of similarity measures for a folk song database containing symbolic music data and metadata.

Another approach is to require the subjects to cluster a small set of clips. Depending on the setup of the survey, several phases of re-clustering the data to more or less narrow groups allow for the collection of absolute as well as relative similarity data. The latter kind of data only describes the relation of different comparison pairs to each other, instead of directly assigning a similarity score to the individual pairs. This paradigm is known from psychology [62], and has recently been used

in the BBC musicality test for short snippets of music [65]. Clips belonging to the same class would then be regarded as more similar to each other than clips from different classes.

2.4.3 Relative Similarity Data

With relative similarity data, only relative similarity information such as “Song A is more similar to Song B than to Song C” is required. A common way to request such information is via triad questions which are better known in anthropology and psychology [10, 43]. Such questions do only require a qualitative decision – one out of three songs is chosen as an “Odd Song Out” – instead of a quantitative similarity value. All similarity data types discussed here are affected by listening order. In our approach using relative data, the survey participant is presented with a simple interface which has the potential to reduce skew in participants’ responses over multiple questions. On the other hand, the effects of the context of another clip – the third song – have to be explored. The similarity data obtained relates a greater number of clips than absolute data, which allows for a faster coverage of all clips but also increases the set of possible clip combinations. We find that relative similarity data with its related methods presents an effective tool for modelling music similarity, with the potential to collect large amounts of data in fast-paced online games and surveys. As during the last decade, relative similarity data has only occasionally been addressed in MIR, the thesis presented aims to present a comprehensive set of methods for the application and evaluation of relative data in music similarity modelling.

Ellis and Whitman [22] use relative artist similarity data from a comparative survey to evaluate similarity metrics based on similar artist lists from the All Music Guide¹ to define their ERDÖS distance. Their artist similarity data covers 412 popular musicians, for whom they gathered 16385 relative comparisons. Moreover, they compare crowd-sourced similarity measures based on listening patterns and text analysis of web pages. The distance measures are regularised using Multidimensional Scaling (MDS) to fit metric requirements of symmetry and transitivity.

¹<http://www.allmusic.com/>

They find that the unregularised ERDÖS distance outperforms the cultural crowd-sourced similarity measures. Berenzweig et al. [9] continued work on this data in a large scale comparison of different similarity data sources and models. Notably they use the method of transferring features between institutions – instead of the copyrighted audio data – to facilitate the large-scale experiments. Mcfee and Lanckriet [57] later use semidefinite programming with multiple kernels to learn a multi-modal distance metric from the artist similarity data discussed above. Their results show strong improvement of the learnt metric over the baseline. They firstly present the *partial order embedding* which is later used in the Metric Learning To Rank (MLR) algorithm discussed in Section 6.2.2.

Relative similarity, as the use of MLR in our experiments suggests, is closely related to *ranking data*: Similarity data can be inferred e.g. by assuming that items with higher rank are more similar to the query than those with lower ranking. Rankings have been collected for melody sequences by experts in a work by Typke et al. [95]. Their strategy of collecting relevance rankings which is used in the MIREX melodic similarity challenge¹.

Allan et al. [3] discuss the challenges of gathering consistent relative similarity data via surveys. Besides introducing an interface for the interactive collection of song similarity data, they tackle the problem of subjects' coverage of survey examples. As already pointed out by Novello, Mckinney and Kohlrausch [68], it is usually not feasible to present all triplet permutations for even a medium-sized dataset to a single subject. Their approach of a *balanced complete block design* guarantees a balanced number of occurrences for individual clips and also accomplishes a balancing of the positioning of the clips within the triplets presented to a particular subject.

2.4.3.1 Similarity Research on MagnaTagATune

In our experiments we use data collected by the game with a purpose MagnaTagATune (see Section 3.2.1 for a detailed analysis). Similar to a survey, the game collected relative similarity via triad questions. Motivation of the participants was

¹http://www.music-ir.org/mirex/wiki/2014:Symbolic_Melodic_Similarity

achieved by rewarding input-agreement. Concurrently to our first experiments, Stober and Nürnberger [91] also explored the MagnaTagATune dataset, comparing algorithms for linear and quadratic optimisation of a similarity measure based on feature weighting. They apply early fusion of the feature data via musically motivated facet differences followed by adapting a linear model. They analyse the training methods on two different subsets of the similarity constraints in the dataset (see Section 3). The smaller of these subsets is designed to be solvable by all of the optimisation approaches, showing the learnability of a large subset of the data. For the larger set, where not all constraints can be learned, an Support Vector Machine (SVM)-based approach by Cheng and Hüllermeier [18] achieves the best results. They find that training similarity measures from this data was possible but the resulting measure slightly violated the constraints of being a metric. While we concurrently used the SVM-Light library after Schultz and Joachims [81] for similar experiments, both approaches showed similar results and behaviour when using a larger similarity dataset from MagnaTagATune as published in [pub:5].

The early fusion approach can support better user understanding and interaction, and the results are similar to a late fusion approach as we show in a systematic comparison of our work to the one of Stober et al. (see [pub:5]). Using multiple learning cycles with model selection, manually-designed tag features and facet difference measures the quadratic optimisation method outperforms the linear regression approach.

2.4.4 User Preference Data

User preference data is closely related to music similarity, in that user preference is influenced by perceived similarity of music pieces in the listening context [55]. Although such preference data can be easily mined through playlists and playback behaviour of online services, it is still different from essential similarity data in that it also involves factors such as relevance. For a very recent comprehensive book on preference and Music Recommendation we refer the reader to the thesis of Bogdanov [11]. In particular, different information sources used for the comparison of music, namely metadata, audio-based or combined information, are compared

for their performance in music recommendation. Bodganov concludes that audio-based approaches currently are inferior to those utilising metadata, but that the difference can be alleviated by adding very little and easily accessible metadata to the audio-based approach. The reduced amount of human annotations needed renders the latter strategy particularly attractive. Motivated through the finding that current methods for music recommendation would not cogently address the task, semantic features are introduced that “reduce the gap between low-level features and human-level judgments”, also known as the *semantic gap*, and furthermore allow for analysis of music preference in humans based on the resulting models. Stober [89] also utilised user preference data in optimising a similarity space for neighbourhood-preserving projections of multimedia collections. Although their evaluation focusses on image retrieval, the system can also be applied to music.

In the general field of machine learning, more approaches exist particularly for the learning from class-based and absolute similarity data. In most cases, computational similarity models optimise the dual problem of a distance measure. Yang [105] has summarised a considerable range of distance learning methods, including Linear Discriminant Analysis, nearest-neighbour-based optimisation, and kernelised approaches such as SVM.

2.5 Adapting Computational Models to Music Data

Instead of directly learning from similarity ratings, other data can be used to learn music similarity measures. The learnt models can be used as an approximation for similarity if the data type allows for it, or for other music classification tasks. Crowd-sourcing, as such a data source, makes use of the large numbers of people that can be reached through the Internet. Based on users’ playlists, ‘like’ data, music purchase history and tag annotations, substantial datasets can be collected [12, 59]. Many models learnt from such data have been introduced in the recent years, but their applicability depends on the relationship of the data source to the application scenario. Unfortunately, there are very few open ground truth data sources for such tasks and experiments are often performed on closed, individual datasets.

The approaches discussed below use music similarity data as well as structurally related data from crowd-sourcing to derive music similarity or relevance models which show potential for music similarity modelling.

2.5.1 Support Vector Machines

Support Vector Machines (SVMs)-based and related techniques, constitute a promising technique for similarity and metric learning in this thesis. Apart from the relevant work cited in Section 2.4.3.1 above, McFee, Barrington and Lanckriet [56] parametrise a music similarity metric using collaborative filtering data. They use Mahalanobis metrics to describe a parametrised linear combination of content-based features, using Metric Learning To Rank (MLR) for training. Post-training analysis of feature weights revealed that tags relating to genre or radio stations were assigned greater weights than those related to music theoretical terms. In our experiments in Chapter 8, we use MLR to adapt a metric to user-provided music similarity data.

2.5.2 Neural Networks

When compared to distance metric learning, artificial neural networks allow for a larger function space to be searched for an optimal modelling of the similarity data. On the downside of the flexibility gained, precautions such as regularisation have to be met in order not to overfit to small datasets. Weyde [104] uses a network architecture able to learn from relative data for matching of symbolic note sequence. We will use a modified approach when developing the RDNN method for relative similarity learning in Section 6.3.7.

Sotiropoulos, Lampropoulos and Tsihrintzis [88] use Radial Basis Functions Networks to model music similarity perception. Their system for modelling music similarity perception (MUSIPER) initialises multiple such neural networks using acoustic feature information. Afterwards, a relevance feedback cycle is performed with users to gradually adapt the neural networks to the participants perception:

In each cycle, given a target piece, participants rank the suggestions of each network according to perceived similarity. The ranking is then used for fine-tuning the networks. Their evaluation involves 100 participants in 6 feedback cycles each, and demonstrates successful increase of the systems predictive power. Analysis of different feature types showed that networks fed with MFCC-related features performed better than those with beat- or pitch-related features.

2.5.3 Other Classifiers

A feedback cycle approach similar to the one above is used in an early similarity adaptation approach by Rolland [75] who adapts a feature weighting for a query-by-humming system with user feedback.

Slaney et al. [85] also present a general method for learning a Mahalanobis distance metric. They adapt similarity on user preference data. Their experiments evaluate the similarity metrics based on artist name identity of k nearest neighbours (kNN). They find that the collaborative-filtering based measure outperforms a content-based metric. The unknown variety of style given an artist is an instance of a general problem associated to using vaguely defined labels as classes. Secondly the imbalanced distribution of collaborative-filtering information in their data is discussed, as the pre-selection of the users' playlists influences the items they can "like". The variety of similarity models is later extended by Slaney, Weinberger and White [84], comparing six approaches of adapting content-based similarity on the same ground truth (unmodified, whitening, LDA, NCA, LMNN and RCA), showing significant improvement through training for all models.

Davis et al. [20] present the Information-Theoretic Metric Learning (ITML) algorithm optimising a fully parametrised Mahalanobis metric, which allows for a regularisation towards another predefined Mahalanobis metric. An online version of the algorithm is available as well. The results of their experiments with several standard classification datasets show a similar or slightly superior performance of ITML when compared to other state-of-the-art approaches. In Section 6.3.6.1, we introduce a new method for learning distance from relative constraints which is based on ITML.

2.5 Adapting Computational Models to Music Data

Schedl, Hauger and Urbano [78] use methods from text retrieval to define an artist similarity measure based on term co-occurrence in microblogs. Their experiments are evaluated against crowd-sourced artist similarity from Last.fm. The number of data and results encourage the integration of this data-source with current models in MIR, as it also contains cultural information which is not represented in standard feature sets. To this end, Hauger et al. [34] very recently released the open Million Musical Tweets Dataset¹.

2.5.4 Feature Selection and Processing

The information used to represent music in similarity and other computational models is usually extracted from different data sources such as audio recordings, scores or further data. Such more (computationally) descriptive music attributes are called features. Features, as detailed in Chapter 5 are traditionally extracted and finalised before being input into similarity models for training or prediction (see Figure 2.1). There is a wide variety of features available for extraction from audio, and general overviews are available [63, 64, 72]. For initial description and overview, features are usually assigned a loose position on a scale concerning the closeness to being a semantic descriptor rather than a physical measurement. Another indicator for assigning the features term such as “high-level, medium-level or low-level” is the complexity of their extraction.

Typical low-level features include time-domain signal analysis such as energy and zero-crossing rates, or direct frequency domain results such as spectral centroid. Mid-level features then comprise derivatives from low-level features. For example, chroma features allow for representation of some harmonic context. The notion of high-level features then points to information that corresponds to musical or generally semantic descriptors which are often also used in human descriptions of music such as identified chords.

Feature selection is used in information retrieval for optimising efficiency by means of only considering relevant data streams. In this general case, Dash and Liu [19] produced a systematic method for feature selection in generic classification

¹<http://www.cp.jku.at/datasets/MMTD/>

tasks. The final feature selection highly depends on the actual application's context. Factors to consider comprise the dataset size, number of classes, and robustness against noise. Based on such background knowledge, their method allows selecting an appropriate set of features. For music information retrieval, Pickens [73] categorised features for retrieval and similarity modelling with symbolic score data, separating "shallow-structure" and "deep-structure" features. In their paper, most of the features suitable for automatic extraction belong to the shallow-structure group.

Especially for music exploration tasks, simple similarity models can allow users to manually weight features, adapting the similarity parameters directly while they browse the resulting music library. Baumann and Halloran [6] present such an approach where users can choose recommendations by sound, style and lyrics. Vignoli and Pauws [96] provide a system using a larger set of features including timbre, genre, mood, tempo, and year. Their user-evaluation shows that the selection of features increases complexity of the music listening interface. This motivates automatic means for parametrising similarity models as discussed above and in this thesis.

Bello [7] compares the structural similarity of music pieces using a normalised compression distance based on their self-similarity matrices. For creating the self-similarity matrices, describing the repetitive structure of a recording, chroma and Mel Frequency Cepstral Coefficient (MFCC) are used. A comparison of two pieces' matrices using compression methods, *bzip2* being the most effective, delivers the similarity measure based on their structure. This approach of comparison is interesting in that similarity matrices can be calculated from a wide variety of features including those we use and describe in Chapter 5. The evaluation analyses the effectiveness of the similarity measure when used for clustering a collection of classical music audio recordings into groups of performed works. Bello extends his approach to recurrence plots in [8], testing more feature variants including more timbre-invariant chroma features as well as long-time chroma statistics.

2.5.4.1 Feature Learning

Recently, several algorithms for feature learning from datasets have been developed in different domains [35, 36, 47, 49, 76, 97]. For example, in computer vision, state-of-the-art feature learning shows similar or better performance compared to algorithms using only conventional feature extraction methods [47, 49, 76].

Feature learning strategies are now successfully employed in some MIR tasks. The feature extraction is mostly performed on the basis of low-level features such as the spectrogram or MFCCs. A methodical overview for using learnt features for MIR tasks is presented by Nam et al. [67]. They further show the effectiveness of their approach in tag classification with linear kernel SVM on the CAL500 dataset. Nam et al. [66] also use Deep Belief Networks (DBNs) for automatic transcription of piano music using a similar SVM classifier. They test both a shallow structure only using the first hidden layer of an RBM as well as a DBN which was fine-tuned using backpropagation. Both methods for transforming spectrogram features improved performance over baseline approaches.

Schlüter and Osendorfer [79] model similarity regarding musical genre with RBMs. They apply a Mean-Covariance RBM on MFCCs to learn local high-level features, which are then aggregated for whole songs via feature histograms. The similarity of songs is then quantified as distance between the songs' feature histograms using five measurement methods: cosine distance, the Euclidean metric, Manhattan distance, and symmetrized Kullback-Leibler and Jensen-Shannon divergence.

Hamel and Eck [33] use Deep Belief Networks (DBNs) for genre classification with a Gaussian kernel Support Vector Machine (SVM) and show improvements on their baseline approach. Dieleman, Brakel and Schrauwen [21] apply Convolutional Deep Belief Networks to learn from audio features and metadata in the Million Song Dataset (Million Song Dataset) for artist recognition, genre recognition, and key detection. In all three tasks, they first train the DBN and subsequently use it for initialising a multilayer perceptron. As reported, their approach achieves better performance than naive Bayesian and windowed logistic regression models.

Schmidt, Scott and Kim [80] apply DBNs to learn three types of emotion-based acoustic features using different approaches of time-frame representation. Their short-time features learnt by DBN outperform individual results from MFCC, chroma, spectral shape, timbre, and spectral contrast features. The performance is further improved by the outputs from hidden layers of DBN trained on a multi-frame features. Best results are achieved for an universal background model trained on an unlabelled music data set before fine tuning to the music emotion data.

2.6 Conclusion

In the above summary of this thesis' background, we introduced different concepts of music similarity such as its embedding in research paradigms of Musicology and Music Information Retrieval. Earlier research and the multiplicity of usages of the term similarity show the intricate position it plays in research and applications of these two fields. We discussed psychological research including the feature-based similarity model of Tversky, and how the assumption of symmetry in metric models potentially limits their predictive performance for human-based similarity data.

For the discipline of MIR, we explained how music similarity models fit into the typical signal flow. Here, the music clips, represented as audio signal and external information, undergo feature extraction before the similarity model can be applied. Some similarity models can be adapted at this stage by feature selection or weighting of different features. Furthermore, feature transformations using PCA or RBMs have been shown to improve performance in MIR applications. Traditionally, the models are fixed after initial experiments and expert judgements, but more recent research has established models that can be trained given the clips' features and additional similarity data as ground truth.

The majority of similarity model adaptation experiments have not been performed on human statements of perceived music similarity data but on related information such as genre or artist similarity. The systematic description of types of similarity data that has been used as ground truth in MIR experiments allows for distinguishing different types of similarity data. This thesis focusses on relative human simil-

2.6 Conclusion

arity data, but other forms include absolute similarity data, class-based similarity data and user preference data.

Only one large dataset of relative similarity data has been collected in MagnaTagATune and we currently collect more data through a Game With A Purpose (GWAP) in the CASimIR dataset (see Section 4.2.5). GWAP achieve control over and motivation of participants through different strategies of user interaction. When looking at the wider usage of similarity models in MIR, research mostly uses proximate information such as preference data or genre data for experiments, as this data can be modelled easier with mainstream machine learning methods and is more widely available. Unfortunately this leads to a very heterogeneous landscape of experiment-specific datasets which are often not public. The MagnaTagATune dataset, being available for download online, allows for some of the first reproducible similarity learning experiments on a relatively large collection of music and human music similarity ratings. In general, relative similarity data, in combination with GWAP promises to facilitate large-scale MIR research on similarity and user data.

The early stage of research with relative music similarity data, leaves us at a very limited selection of models and training algorithms, mostly designed for ranking data of websites or other non-musical applications. Still, the results for learning distance metrics from collaborative filtering and the availability of data from GWAPs motivate the first systematic evaluation (Chapters 6 and 8) of such methods for similarity learning in music. In order to widen the scope of comparison and increase performance of the models predictions we will here furthermore develop and evaluate new algorithms for similarity learning. To this end, we will adapt methods successful with absolute or class-based similarity data, such as ITML and neural networks mentioned above, to relative data. We thereby try different model architectures and music representations. Addressing the cultural and perceptual contexts of similarity, we provide means for the development of culture-specific similarity models and transfer learning, as well as to further research on asymmetry in perception.

Like for the similarity models, there is little published on relative music similarity data itself and its analysis. The following chapter will discuss methods for ana-

lysing relative similarity data as well as a first analysis of the MagnaTagATune similarity dataset with such methods.

In language there are only differences.

(Ferdinand de Saussure, 1909)

3 Relative Similarity Data

In order to use relative similarity data for modelling, we require methods to analyse [pub:7] the data itself prior to model training, as our research question *rq:4* states. In [pub:9] this chapter we therefore discuss properties of such data, its representation in a graph model and methods for analysis and processing of whole similarity datasets. Relative similarity data can be collected through an odd-one-out game, as in the MagnaTagATune and CASimIR datasets which we use in this study. We furthermore use the MagnaTagATune dataset as an example application case for the methods, providing the first analysis of the contained similarity data. The CASimIR dataset, still growing, will be discussed in detail in the following Chapter 4.

We gather relative similarity data in the form of relations between two pairs of clips. In general, given the clips C_i , C_j , C_k and C_l , we can express a similarity relation y using the following:

$$(C_i, C_j) \overset{y}{>} (C_k, C_l), \quad (3.1)$$

where the relation $\overset{\text{sim}}{>}$ denotes "more similar than".

In an odd-one-out game, three clips C_i , C_j and C_k are presented to the players, who are asked to choose the one which least fits with the others. This rating indicates a relatively higher similarity between the two remaining clips than to the selected one: A vote for C_k as the odd-one-out can thus be interpreted modelled using the following two relations $y_1 = (i, j, k)$, $y_2 = (j, i, k)$:

$$\begin{aligned} & (C_i, C_j) \overset{y_1}{>} (C_i, C_k) \\ \wedge & (C_i, C_j) \overset{y_2}{>} (C_j, C_k). \end{aligned} \quad (3.2)$$

3.1 The Similarity Graph

Relative similarity relations can be represented as edges in a directed weighted graph of pairs of clips (McFee et al. [58], Stober et al. [90]): Given the clip index I for all clips $C_i, i \in I$ and similarity information \hat{Q} containing constraints in form (3.1), our Graph $G = (V, E)$ consists of vertices representing clip pairs

$$V = \{(C_i, C_j) \mid i, j \in I\}$$

and edges

$$E = \left\{ ((C_i, C_j), (C_i, C_k), \alpha_{(i,j,k)}) \mid (i, j, k) \in \hat{Q}, \alpha_{(i,j,k)} \in \mathbb{N} \setminus 0 \right\}$$

representing the pairs' similarity relations.

3.1.1 Determining Constraint Weights

The weights $\alpha_{(i,j,k)}$ assigned to the edges represent the number of occurrences of a particular constraint (i, j, k) . We here follow the common representation of weights as integer numbers as presented by Stober and Nürnbergger [90]. They refer to this graph as a multigraph where the weights count the number of identical edges repeating a constraint. A weighted graph corresponding to Equation 3.2 is shown in Figure 3.1, for clips C_i, C_j and C_k and weights $\alpha_{(i,j,k)}$ and $\alpha_{(j,i,k)}$.

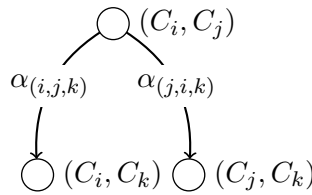


Figure 3.1: Graph induced by a single “odd-one-out” statement, C_k is the odd-one-out as in Equation 3.2. Vertices represent pairs of clips and edges represent the relation *more-similar-than*.

An alternative representation of the weights, especially considering accumulation across multiple user inputs, is implemented and described in Section 7.2.2.1.

3.1.2 Cycles and Inconsistent Data

The induced graph can include inconsistent similarity information, for instance from users directly disagreeing on the outlying clip in a triplet, or multiple votes leading to an inconsistency when considering the transitivity of the induced similarity metric. Inconsistencies appear as cycles in the graph as shown in Figures 3.2 and 3.3. A more complex example is given in the appendix, Figure 10.7 on page 222. Such cycles can be found and analysed using standard methods for extracting strongly connected components in directed graphs.

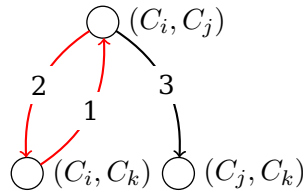


Figure 3.2: Graph containing a length-2 cycle. Cycle highlighted in light red.

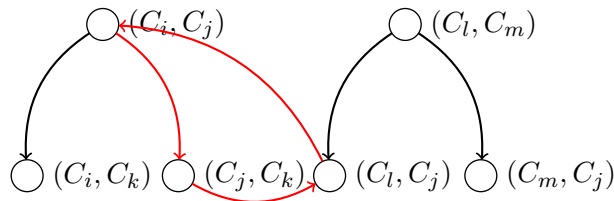


Figure 3.3: Graph containing a length-3 cycle. Cycle highlighted in light red. Edge weights have been hidden.

Some of the methods for similarity modelling as discussed in Chapter 6 require the similarity data to be consistent. For example, this is the case for the partial order feature used by MLR (see [61]). In order to apply these methods, we use an approach used by Stober and Nürnbergger [90] for filtering inconsistent data. In this thesis we only use filtered data for consistency, but Section 9.6 describes perspectives to test methods with unfiltered data. The information to be discarded is selected based on the minimal number of associated user votes: For removing direct inconsistencies we remove cycles of length 2 by removing the edge (i, j, k) with the smaller weight $\alpha_{(i,j,k)}$ and subtracting its weight from the weight $\alpha_{(i,k,j)}$ of the edge in the opposite direction. Thus, the edge with greater weight, backed

by the larger amount of votes, is retained. If two inconsistent edges have equal weight, both are deleted, possibly leaving a vertex disconnected from the graph. In that case, as similarity data is represented by edges, the corresponding clip combination would be removed from the dataset.

Removing cycles of greater length and finding the maximal acyclic subgraph of G is an NP-hard problem [42]. McFee et al. [58] use a randomised algorithm by Aho, Garey and Ullman [1] to extract an acyclic subgraph for this application. The graph is created by iteratively adding edges to a new graph and testing for cycles. Edges that complete a cycle are omitted. Depending on the similarity data, different means of finding an acyclic subgraph may give better or even optimal results. The MagnaTagATune and CASimIR datasets are already cycle-free after the first step of removing cycles of length 2. For MagnaTagATune this is due to the missing connectivity between triplets, but the new data collected with CASimIR suggests that longer cycles are unlikely to remain after elimination of short cycles, maybe because for this to happen a the majority of ratings has to back every one of the edges involved. See the Section 3.2 for a first analysis of the structure of the MagnaTagATune similarity data.

The resulting acyclic weighted graph Q provides the similarity constraints $(i, j, k) \in Q$ that we use to train the similarity measures. The analysis of the adjacent components in this graph gives information on transitive similarity relations expressed by the constraints, such as

$$(C_i, C_j) \stackrel{y_1}{>} (C_i, C_k) \stackrel{y_1}{>} (C_k, C_l) \quad (3.3)$$

The role of such relations for similarity learning have yet to be more closely researched, but transitivity relations and sharing of clips between clip pairs do impact the evaluation of model performance, as we show in our experiments Section 8.5. The length of cycles is limited by largest connected component in the similarity graph, and we now discuss the usage of such analysis for similarity data.

3.1 The Similarity Graph

3.1.3 Connectedness

The average number of clip pairs which can be reached from a specific clip pair (C_i, C_j) via edges of the graph defines the connectedness of the resulting similarity graph. This can be measured by counting the vertices in connected components of the undirected graph which is derived from the similarity graph by removing directionality of the edges. To this end the DiGraph class uses Tarjan's strongly connected components algorithm [92].

Algorithm 1 Connectedness Analysis of Similarity Graph

Require: Similarity Graph $G = (V, E)$

Get undirected graph $G' := (V, E')$, $E' = \{(C_i, C_j), (C_j, C_i) \mid (C_i, C_j) \in E\}$

Calculate connected components $\text{SCC}(G')$

return maximal cardinality $\max(|\text{SCC}(G')|)$

A completely connected similarity graph allows for a direct inference of similarity relations between all clip pairs contained, given the graph is not containing contradictory similarity information and is therefore acyclic. Given a query clip, absolute similarity values can be assigned e.g. using Dijkstra's shortest path algorithm. This is useful for converting relative similarity data to absolute similarity data for use with methods such as Multidimensional Scaling. Thus, connectedness in the similarity graph is desirable both for analysing contradictions (cycles) and experiments comparing similarity data types.

In MagnaTagATune, the connectedness of the graph is very low as explained in the following section. The CASimIR framework, (see Section 4.1.1.2) allows to control the connectedness of data during collection and thereby achieves a dataset with far larger connected components (see Table 4.2).

3.2 The MagnaTagATune Dataset

The MagnaTagATune dataset is to our knowledge the only similarity dataset that is freely available¹ with the corresponding music audio data. Therefore, the majority of our experiments in Chapter 8 are based on this set to make our results reproducible and comparable. The cycle-free similarity data used in Chapter 8 can be downloaded² from the web.

3.2.1 Similarity Data

In the bonus mode of the TagATune game, two participants are asked to agree on the odd-one-out of three audio clips. This is a typical instance of an output-agreement game with a purpose. Regardless of the agreement of the participants, the votes of both users are saved in the history for this triplet. The MagnaTagATune dataset contains 7650 such votes for a total 346 of triplets, referring to 1019 clips. No information about the providing participants is published with the dataset. Some of the triplets have been presented as permutations, and the order of display is in the dataset, as well, but not the order of listening. On average, each instance of a triplet permutation counts 14 votes. In our experiments, the information of each player's vote, e.g. C_k being the outlier in (C_i, C_j, C_k) is used to derive two relative similarity constraints as stated in Equation 3.2.

The induced weighted graph, derived from $2 \cdot 7650 = \sum_{(i,j,k) \in \hat{Q}} \alpha_{(i,j,k)}$ votes and depicted on Figure 10.3 on page 218, includes cycles of length 2, but no cycles of greater length. Thus, through removal of cycles of length 2 which resolves all cycles in the initial graph (see [pub:5]), the similarity graph loses 8402 weight points. The resulting directed acyclic weighted graph consists of 337 connected subgraphs G_{sub}^i , each containing 3 vertices, i.e. clip pairs. The 6898 weight points α for 860 unique connections contain the remaining similarity information Q . Equal vote counts for inconsistent statements lead to the isolation of 27 vertices. Thereby,

¹With the expiry of the tagatune web domain, the dataset has moved to <http://mi.soi.city.ac.uk/datasets/magnatagatune>

²<http://mi.soi.city.ac.uk/datasets/phdthesisdw>

3.2 The MagnaTagATune Dataset

26 songs are left without reference to any remaining similarity constraints, reducing the number of referenced clips to 993. The resulting graph is depicted in Figure 10.4 on page 219

When excluding the isolated vertices with no associated similarity information, the combination of clips in the remaining subgraphs corresponds to a subset of the triplets in the initial dataset, now associated with modified weights. This is due to the fact that combinations are only possible within similarity triplets presented to the users. Thus no information about interrelations in between the different clip triplets can be directly extracted from the similarity data.

3.2.2 Genre Distribution over Triplets

In Chapter 2 we discussed the usage of genre data for similarity learning. Genre annotations for this dataset are available from the catalogue of the Magnatune label. Unfortunately, with this dataset, genre-specific similarity measures cannot be studied, as the amount of similarity data per genre is too small for similarity learning. To give an impression of the dataset’s structure, we present below the frequencies of genre groups in the presented triplets for the most frequently annotated genres:

Table 3.1: Number of triplets with n clips sharing the same genre tag.

Genres	$n = 3$ of 3	2 of 3	1 of 3
Electronica, New Age, Ambient	43	159	447
Classical, Baroque	8	65	257
Rock, Alt Rock, Hard Rock, Metal	6	59	251

As becomes apparent in Table 3.1, the number of genre-consistent triplets in the MagnaTagATune dataset is very low. Only for 1 genre group (Electronica...) one can find more than 10 triplets that are genre-consistent. Thus, this dataset does not allow for the comparison of similarity between different genres due to the selection of triplets. Further details of the genre data used with MagnaTagATune are presented in Section 5.2.

3.3 Conclusion

Relative similarity datasets of the size found in MagnaTagATune or CASimIR have only recently become available in the field of Music Information Retrieval. In this chapter, we respond to *rq:4* by providing a basis of analysis methods for relative similarity data. After a formalisation of the contained information and representation through graphs, we presented general methods for analysis of similarity datasets. This includes the definition of constraint weights, with accumulation of data from several users or trials. Furthermore, we discuss the challenge of removing inconsistencies as they naturally occur in relative similarity data collected from human participants, and a method for generating a consistent subset of the similarity data. For determining the maximal cycle length as well as the maximal number of clips connected through a transitive chain in the similarity data, we describe an analysis of connected components on the direction-insensitive version of the similarity graph.

We have presented the first thorough analysis of the MagnaTagATune similarity data, extending on the cycle removal statistics determined by the author of this thesis and firstly reported in a joint publication with Stober et al. in [pub:9]. The knowledge from this analysis was used to inform a method for unbiased cross-validation sampling in our experiments in Section 8.1. Concerning cycles, we discovered that the individual triplets of clips presented to the users for MagnaTagATune are not interconnected through common clips. The absence of genre consistency within triplets motivated the development of the mechanisms we present for explicit control of genre-homogeneous triplets in the CASimIR framework.

When compared to absolute similarity data, the collection of relative data is less complicated and can be performed through short surveys and games enabled by the CASimIR game framework which we describe in the following section. There, we will also focus on the design and criteria for similarity datasets, with the general intention to attach cultural attributes to the similarity data.

Even the most perfect reproduction of a work of art is lacking in one element: its presence in time and space, its unique existence at the place where it happens to be.

(Walter Benjamin, 1936)

4 Collecting Culture-Aware Data via Games With A Purpose

There are many tasks which are difficult to solve without human input. For example, the meaning of an image or a piece of music are subjective information that only humans can provide. Perceived music similarity is one of many context- and user-dependent attributes that requires user input data both for training of models and evaluation. One way of obtaining such input is using Games With a Purpose (GWAPs) , a form of crowdsourcing which utilises enjoyment to engage users to provide valid information in large numbers. [pub:10] [pub:1] [pub:3]

Research question *rq:5* addresses the challenges and opportunities of collecting data on the web, related to development efforts and participant control. Integrating methodical solutions to these issues, we now present a generic framework that supports data collection in the social web via GWAP and surveys with multiple players. We will discuss an overall architecture for GWAP surveys that ensures extensibility and light development through modularity. This includes a back-end API and example implementation for management and selection of survey tasks. For displaying content in web and social network games for desktop and mobile platforms, a game front-end is presented.

The viability of the presented approach and framework is then exemplified through the Spot the Odd Song Out¹ game, which is currently used for expanding the CASimIR dataset. An analysis of the similarity dataset collected with Spot the

¹Spot the Odd Song Out was implemented with the help of Guillaume Bellec.

Odd Song Out to date will show the effectiveness of the presented means for real-time collection control in a comparison to the MagnaTagATune dataset examined in Section 3.2. As the presented framework allows for a wide range of data collection tasks, Spot the Odd Song Out furthermore collects tempo and rhythm data via two modules developed at KTH Stockholm (see Figure 4.6) which we provide first statistics on.

Finally, in Section 4.3 we present the first country-annotated music similarity dataset of its size and sketch a methodology for extending adaptive similarity models to culture-aware models. This is enabled through the particular feature of Spot the Odd Song Out, in that it explicitly gathers anonymous participant attribute data which is linked to the collected music annotations. This opens up the possibility of user- and group-based similarity models, taking into account the specifics, such as a cultural embedding of the participants providing the similarity data or other annotations.

4.1 A Generic Framework for GWAP

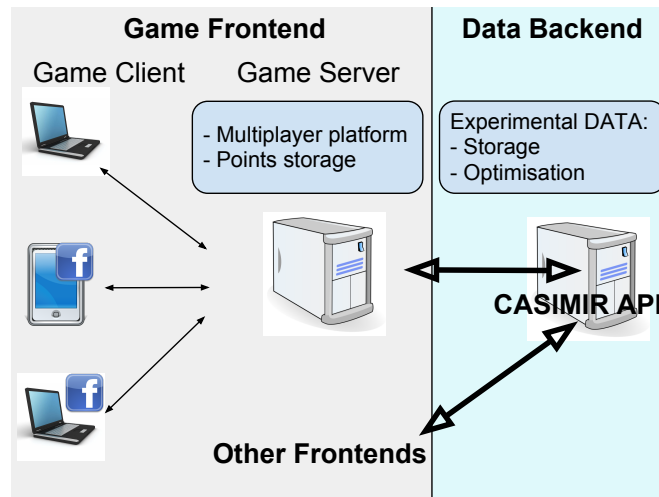


Figure 4.1: The CASimIR framework consists of a back-end and possibly several front-ends. The API can be used by different clients simultaneously to provide annotations to the same dataset.

So far, to our knowledge all existing GWAP systems have been coded from scratch.

4.1 A Generic Framework for GWAP

As we implemented the Spot the Odd Song Out game¹ we felt the need to provide a versatile and reusable framework, the Culture Aware Similarity Information Retriever (CASimIR), that makes the development and use of GWAP more effective and efficient. Therefore, Spot the Odd Song Out will serve as an example for a CASimIR configuration in the following text.

The CASimIR framework as well as the first data collected with Spot the Odd Song Out are made available online² under Open Source and Creative Commons licenses. The source code is provided via a version control system, and we hope that the framework will facilitate more research with and into GWAP, resulting to further collaborations and contributions to the current framework.

The CASimIR framework consists of a back-end that manages and controls the use of the media and annotation data and front-ends for managing the game interaction and players. The framework was specifically designed to separate the collection of annotations (*Game Framework*) from the management of data and survey questions (*CASimIR API*(API)). Figure 4.1 gives an overview of the architecture. This separation of the data back-end enables the provision of a stable environment for data collection, which can be accessed by different front-ends, thus encouraging collaboration and sharing of datasets. It also facilitates rapid development of alternative front-ends without the need to reimplement any back-end functionality.

The interaction with the back-end API is structured on the basis of *questions*. A question contains the bundle of information (media and metadata) which is necessary for presenting a question to the user and retrieve an answer. Note that in CASimIR, we address music content by songs rather than clips. As currently no two clips of the same song are included in the CASimIR song library, the terms can be used interchangeably. The integration of multiple clips from the same song in CASimIR is straightforward by using different internal identifiers. Figure 4.2 shows the interaction between the API and a game front-end. The API is implemented as a remote procedure service using the SOAP³ protocol in PHP5 based

¹<http://mi.soi.city.ac.uk/camir/game/>

²<http://mi.soi.city.ac.uk/datasets/aes2013casimir/>

³<http://www.w3.org/TR/soap/>

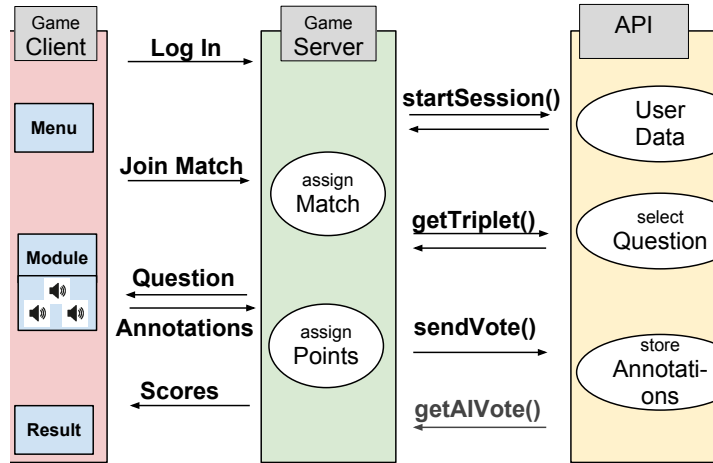


Figure 4.2: Communication of a game with the CASimIR API. Game related data such as player scores are transferred in communication between the game client and the game server. The game server requests or sends only the annotation-relevant data to the API.

on the following objects: Questions are transferred via the Song and Triplet objects, and annotations are represented by Votes. Firstly the game registers one or more participants using the `startSession` function. Then one or more questions can be retrieved by the front-end using `getSong`, `getTriplet`, or newly developed question types. `getAIVote` returns synthesized answers given a question.

The CASimIR game framework implements each question type as a module in the game server, which handles the presentation of the question to the participant. It also collects and evaluates the player's input, creates the data for back-end storage and the feedback for the player. The game server is providing a number of web technologies over a web server. The server interacts with the game client that runs in a browser on a computer or mobile device. The annotations are then sent to the back-end using the `sendVote` interface where they are stored. If the specific question type supports automatic answers, the function `getAIVote` can return an answer based on ground truth or collected annotations.

4.1.1 CASimIR API and back-end Implementation

The CASimIR API and back-end implementation provide three functionalities which are essential to almost any project annotating media content:

1. Organisation of the database with media examples to be shown to the participants
2. Selection of appropriate questions (examples)
3. Storage and referencing of received annotations

It is important to keep this functionality separate from the game playing logic and the user interface. This enables us to create a back-end that can be simultaneously used by different applications. The back-end centrally controls the experiment regarding the number of users and presented media. Also it is independent of the constraints of the interface, such as hardware, programming language or user interaction. Our back-end implementation provides a dynamic control of the experiment in terms of users and data, which is a specific challenge in GWAP compared to traditional surveys or lab experiments.

4.1.1.1 Participant Numbers and Song Subsets

The selection of the media and questions to be presented to participants is normally determined based on a fixed number of participants, a fixed number of questions presented to a single participant, and the intended coverage of the available questions (e.g. see Allan, Müllensiefen and Wiggins [3] for similarity triplets). Especially for paper-based studies, the layout and selection of the questions cannot be changed during the study, and thus the number of participants is fixed as well. In a web-based application it is possible and desirable to allow as many participants as possible to take part. The CASimIR framework accounts for this by using an extendable subset of media and questions, called working sets.

The *Song Library* in the MYSQL database contains all songs or clips available for inclusion in the system. The audio source is linked to the content via a URL e.g. using a preview link from the 7digital library, which is then played back by the front-end. Each media example is assigned an internal identifier as well as genre

description where available. Additional information can contain artist and album names, ISRC codes, MusicBrainz and The Echo Nest ids.

The back-end manages a *Song Working Set*, subset of all media identifiers, which are allowed to be included into questions. Songs are dynamically added to the Song Working Set when there is enough user input for the existing songs.

The Song Working Set is then used to generate the questions which can be a combination of media and metadata presented to the participant. This generates the *Question Working Set* which is dynamically updated like the Song Working Set. For example, for triplet questions (`getTriplet`), the Question Working Set contains an increasing set of triplets of elements from the Song Working Set. Figure 4.3 shows an example configuration of song and question working sets.

4.1.1.2 Example Selection

As in our approach the total number of participants is undefined a priori, it cannot be used to define the selection of questions. Still, statistical control has to be exercised over the presented questions. During runtime, the *Question Working Set* is used to pick the questions for each round in a game. CASimIR provides building blocks for rules to select questions based on statistics and metadata such as genre. For example, in Spot the Odd Song Out, the triplets for a game round are chosen randomly, but the order of presentation (permutation) is selected using a circular strategy for equal distribution and to minimise repeated presentation of the same permutation to a participant. Also, *control questions* for checking participants credibility can be issued at a regular rate. Such questions are automatically generated for triplets: Our red herring questions contain two identical clips and thus test whether the participant chooses the remaining clip as correct answer. Furthermore, additional parameters for question selection can be provided via the API interface and easily integrated into the real-time selection process.

For large working sets, the response time of the API can be reduced by only considering a random subset of questions when selecting them from the Question Working Set.

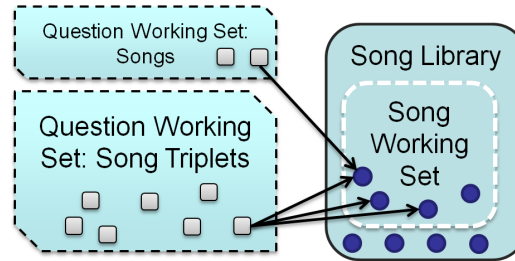


Figure 4.3: Clip and Question Working Sets are dynamically expanded as subsets of the Song Library.

4.1.1.3 Management of Player Input

A game front-end can communicate with the CASimIR API back-end via the `sendVote` function once it has received a question. The annotations are represented by `Vote` objects. When received by the back-end these are checked for integrity and saved into the appropriate database depending on their associated question type.

Designed to serve a variety of annotation tasks, the generic `Vote` objects are built on a flexible database-object mapping that can hold a wide range of data which is automatically stored. Flexibility is given on a per-annotation basis, as information may or may not be available for each participant depending on success in completing the task, software limitations and privacy restrictions. For retrieval, each annotation is linked to the respective participant, time and date, session and question.

CASimIR was built to allow for linking participants' cultural and other attributes to the collected annotations. Therefore the API maintains anonymous datasets of participant information. For the Spot the Odd Song Out game, this includes information about the participants approximate location and country, languages and general as well as musical education. This optional information is extended by information about favourite bands etc. from a social network, where such information is available and shared by the participant with the application.

4.1.1.4 Automatic Answers

GWAP based on agreement between participants need to provide automatic answers where participants cannot be teamed up. Therefore the CASimIR API provides the `getAIVote` function, which can be implemented for each question type. It returns a user vote, based on either recorded or synthesized data. Currently, automatic answers for Spot the Odd Song Out are based on previously recorded answers where possible: A generic strategy has been implemented that randomly chooses a recorded vote for the specified question, which is then returned by the API. This allows for the simulation of human players to the participant. We prefer the use of human answers as they provide answers that potentially are more consistent with new participants answers, and, in case of disagreement, with their expectations on alternative answers. For questions where no answers have yet been recorded, a random answer is provided by the system. The random selection or generation of answers allows for multiple AI players to cover a wider range of answers and thus increase the chance of agreement and reward of the human participants in the match. Furthermore, the ratio of data-based and random answers can be configured to provide for specific experiment requirements.

4.1.2 Game Framework

The CASimIR game framework has been developed to work with the the CASimIR API. The game framework supports the creation of new games and gamification of existing surveys. The game framework provides functionality to run a multi-player game logic and its player interface.

Our game framework allows for the implementation of both single- and multi-player games. In order to support a multi-player experience, the game logic is centralised in the *Game Server*, which provides the back-end to the game interface website in PHP/MYSQL. It provides the following functionalities:

- Central game management and logic
- Grouping and synchronisation of participants in multi-player matches
- Retrieving questions from the API and storing the participants answers

4.1 A Generic Framework for GWAP

- Publishing of game-related information streams (achievements and scores) through social networks

The *Game Client* provides the user interface (in HTML5 and JavaScript) in a web browser on the participant's machine such as mobile phone, tablet, desktop or notebook pc. The functionalities of the game client include:

- Support for various platforms including mobile browsers or a social network environment via an HTML iFrame embedding
- Dynamic display of the questions and results, providing the input graphical user interface for answers via a variety of standardised animated objects.
- Display of game menu, options and collection of additional participant information
- Social networks: Log-in of participants and retrieval of their attributes

4.1.2.1 Social Networks and Participant Login

The integration of a game into social networks opens the potential of collecting many participant attributes, including listening habits, while providing effective means of advertisement through social channels. The CASimIR game framework integrates core social network building blocks for Facebook such as "Like", "Invite Friends", and the posting of game-related stories including scores. Participants' unique identities are used to provide a personalisation of the game experience through avatars and genre selection.

The standard interface for the CASimIR game framework starts with an entry point such as a game menu. The game client locally processes information from the social network or provided by the participant, only transmitting the information necessary for the task to the game server. The game server then only stores information relevant for providing the game whilst transmitting any further participant data such as cultural attributes, but not the social network identifier, to the API using `startSession`.

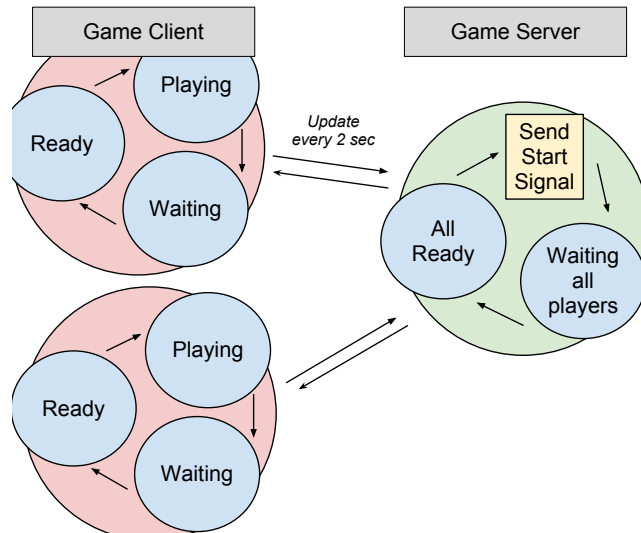


Figure 4.4: Entry and synchronous state progression of two game clients connected to the same match.

4.1.2.2 Game State Model and Multi-Player Synchronisation

Like the Music Information Retrieval GWAP “HerdIt”, games built with our framework can consist of multiple players. Spot the Odd Song Out for example joins four players in a game, which are filled up with AI players if necessary. The number of players was chosen as a tradeoff of multi-player experience and interface restrictions: More players allow for a larger number of alternative answers and thus increase the chance of motivating agreement. On the other hand, the envisaged devices for playing include smartphones with very limited screen size. Thus, in contrast to the 10 players in the desktop game HerdIt, four players were chosen for Spot the Odd Song Out, although the number of players can be increased given an adaptation of the user interface.

In contrast to HerdIt, CASimIR allows for real-time connection of players in addition to the playback of recorded player input via AIVote. Real-time synchronisation over the WWW between HTML front-ends is no trivial task, and we hope our implementation will help other researchers to use this game mode where needed.

As mentioned above, the gaming experience in CASimIR is structured into matches, which consist of a series of mini-games. From the entry point, the participant can

4.2 Case study: Spot the Odd Song Out

progress to the main game by joining a match. A match represents a predefined sequence of mini-games implemented as Modules, which can be run for a single participant or synchronously for all participants of a match. A single module in the game presents a question previously retrieved from the CASimIR API. In a multi-player module, all players are given a certain time to answer the question. The module collects the participants' answers and moves on to the result screen as soon as all participants have answered or when the timeout occurs. The result screen is shown for a predetermined amount of time, after which a new module is started synchronously for all participants. When reaching the end of the module sequence, the game returns to the menu or another exit screen, e.g. the feedback form.

4.1.2.3 Reward Mechanisms

Especially for synchronised multi-player matches, the calculation of scores for participants is nontrivial, as participants can leave at any point during the game. The CASimIR game module architecture provides the complete management of the task of aggregation of annotations or module results for determining rewards, as well as the submission of annotations to the back-end API. Modules can implement their specific logic of awarding points and achievements, including GWAP strategies such as input- or output-agreement. Results are calculated based on the complete available participant answers for a module before the result screen is displayed.

4.2 Case study: Spot the Odd Song Out

To exemplify the possibilities of the CASimIR framework, we present the three annotation collection modules of Spot the Odd Song Out and conducted data collection on the Internet and Facebook.

4.2.1 Game Modules and Interfaces

In Spot the Odd Song Out we combine several different modules and annotation tasks in the same game. In addition to the central similarity module, we add two further tempo-related modules developed by KTH which appear in alternating order. A first motivation for this strategy are the synergy effects in terms of acquiring participants, who now can contribute to several experiments in one match. Furthermore the varying tasks can provide for a more diverse game, and thus may keep players motivated for longer and appeal to their curiosity. Here, even more advanced strategies of creating bonus rounds¹ or specific games as rewards are possible. The alternating games also might reduce the effect of memorisation of previously answered questions on the next instance within a specific task. On the counter-side, less answers are collected for a specific task, and users might be demotivated by specific tasks they do not enjoy.

The first similarity module (Figure 4.5) collects relative similarity data using an odd-one-out question. This namesake for Spot the Odd Song Out presents a question featuring three songs and a request to "Choose the clip which seems the most different to the others". Points are awarded by output-agreement, i.e. by the number of players agreeing on the outlying clip.

For tempo and rhythm experiments, Bellec and Friberg at KtH Stockholm joined us and implemented modules collecting real-time sensor information: The TapTempo and TapRhythm modules (Figure 4.6) require the participants to tap along to a song indicating the tempo, or a freely selected rhythm. Reward is based on comparison with expert tempo annotations and tapping regularity, which adds a competitive element to encourage user engagement.

This, to our knowledge, is the first GWAP including real time sensor data via a web interface for music annotation. By using the highly cross-platform google closure library the usage of the same code on almost all platforms is enabled. Still, the capturing of key and touch timing is subject to some jitter, up to 30ms depending on the platform (see Bellec et al. [pub:1] for details).

¹The TagATune game collected the similarity data in a bonus round, see Section 3.2.1.

4.2 Case study: Spot the Odd Song Out

Allowing for better accessibility, the central user interface elements are shared amongst all modules: The playback of media can be controlled using speaker-symbols in the centre of the screen, which are animated to visually feed back the playing status of the represented song.

CASimIR also saves timing data and the sequence in playback for each media item. Close to the display of current players in the match at the bottom (see e.g. Figure 4.5), an (orange) time bar on the left shows the remaining time (from 60 seconds) for the particular module. The application starts with login and main menu. A match then consists of a synchronised (see Section 4.1.2.2) succession of 6 modules (currently: STOSO, TapTempo, STOSO, TapTempo, STOSO, TapRhythm) followed by reward displays (Section 4.1.2.3).

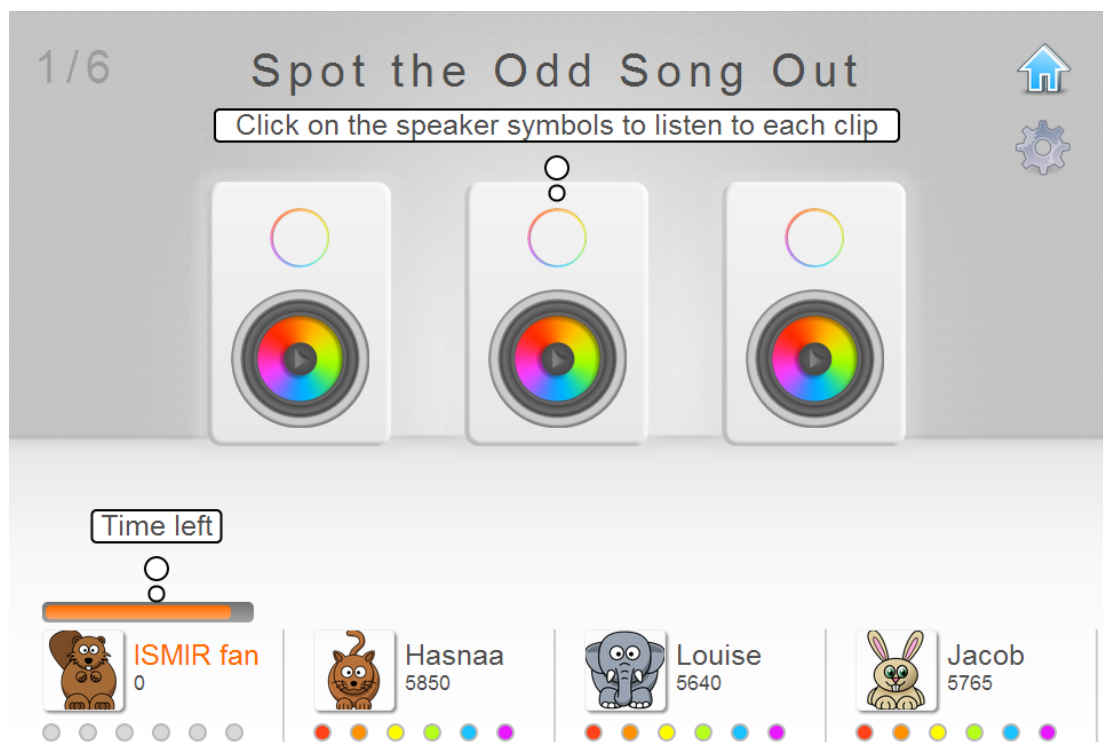


Figure 4.5: Spot the Odd Song Out modules: Similarity.

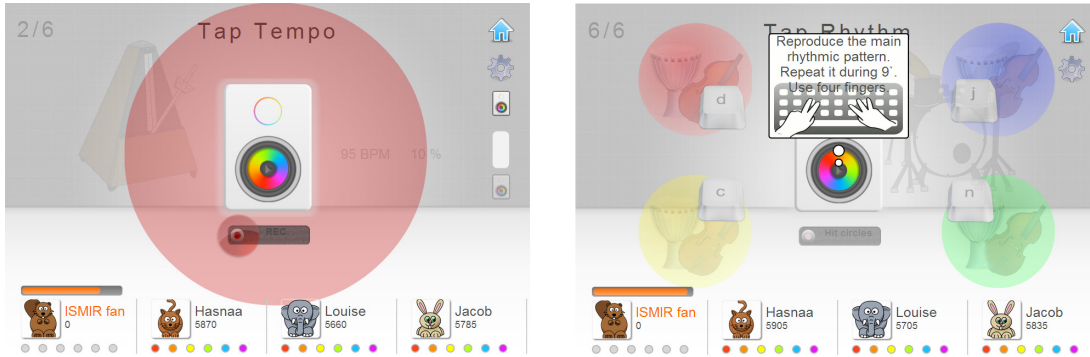


Figure 4.6: User interfaces of the TapTempo (left) and TapRhythm (right) modules.

4.2.2 Song Library

The current Song Library of the back-end in Spot the Odd Song Out contains the 1019 song excerpts from the MagnaTagATune dataset which also are referenced by similarity data in that dataset. These excerpts have a length of 30 seconds. Furthermore the 10,000 songs from the Million Song Dataset subset are included. For the latter songs the participants are presented preview excerpts varying in length from 30 seconds to the whole song¹. The data has been enriched with ISRC codes and preview URLs obtained from 7digital² as well as genres provided by ROVI through the Allmusic site³. Finally, 100 audio clips synthesised from MIDI for a related perceptual experiment (FRI) [25] were included for use with the TapTempo and TapRhythm modules. Those were fully annotated with tempo and further information by experts, and compared to the collected data in their later evaluation [pub:1].

4.2.3 Lifecycle

The Spot the Odd Song Out game, initially only containing the similarity module, was set up in three phases: After an internal evaluation of the user interface with

¹Because of the 1-minute time restriction for a single similarity question, participants rarely listen to more than 10 seconds of a song. The mean listening time per song in CASimIR is 3.5 seconds.

²<http://developer.7digital.net/>

³<http://developer.rovicorp.com/docs>

4.2 Case study: Spot the Odd Song Out

10 participants at City University, a first field experiment was started in August 2012 by integrating the game into a study on music tags and similarity pursued by Marily Niven at City University. Out of the 33 participants reaching the game in the survey, only 20 were able to provide data, as this field test revealed some incompatibilities with different browsers which could now be addressed and fixed. The collected similarity data also allowed to bootstrap the AI players for the game by providing answers to similarity questions.

A following phase contained a larger distribution of the game within City University and KTH Stockholm. In this phase, the multi-player functionality was tested and improved. Also, the TapTempo and TapRhythm modules were added to the game. We furthermore presented a prototype of Spot the Odd Song Out at ISMIR2012 in Porto [pub:3].

After final improvements including development and further testing of a more professional UI design¹, Spot the Odd Song Out was released on Facebook in late February 2013, and efforts were made to distribute the game widely. As an exemplary measure for the effectiveness of the data collection in the different phases, Figure 4.7 shows the amount of similarity data collected with Spot the Odd Song Out until the date this thesis was submitted.

4.2.4 User Participation Over Time

Figure 4.7 shows the number of participating users until May 2014. The graph shows a large participation during the first months where the game was actively promoted. Most participants were guided to the game via Facebook, although less than half of them used the option to link their Facebook profile to the game. As is visible from the steep rise of the graph during February and March 2013, it was possible to attract a large number of users within a short time using peer-to-peer advertisement on the social network and mailing lists such as reddit² or music-ir³. During this time, the majority of similarity data was collected, as visible in Figure 4.7 and denoted in Table 4.1.

¹Thanks to Benjamin Szepan for providing the draft of the current look.

²<http://www.reddit.com/>

³now ISMIR-community

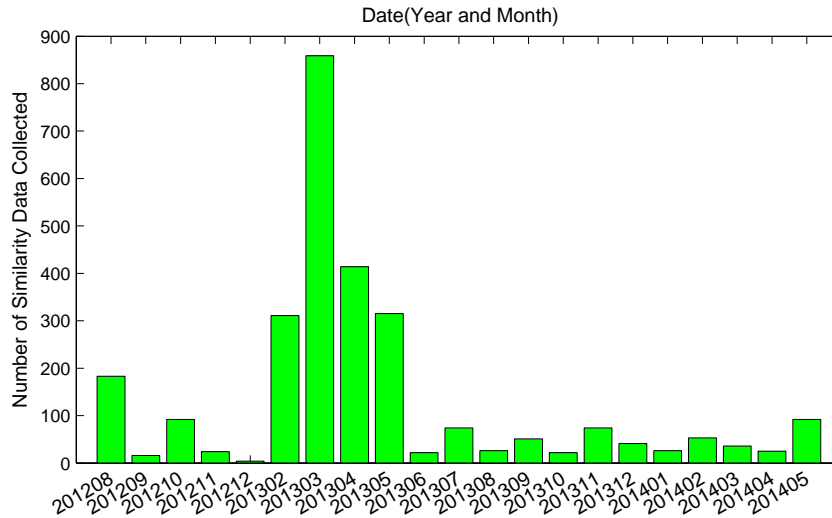


Figure 4.7: Similarity data collected in a pre-phase, during the main run and the further life of Spot the Odd Song Out.

The number of returning users, as plotted in Figure 4.8, is helpful in understanding the sudden drop in engagement after the very active months in the game. As users provided their data anonymously, and profiles were stored only on the basis of the provided user attributes such as age etcetera, only the data regarding users logged in via Facebook can be used to gain such insight. Note that those users are not identified directly but through anonymised hash values. Our analysis of the number of distinct days on which users would return to the game revealed that of the 94 users who logged in with their accounts, 26% returned at least once, and 10% returned on two or more days.

We allowed participants to provide feedback of their experiences with Spot the Odd Song Out at the end of each match. In total, 128 full-text responses were submitted until June 2014. The general feedback was very positive and participants reported they enjoyed playing the game. The Spot the Odd Song Out module was understood by all commenting players, although some asked for more guidance regarding the relevant features for similarity and possibly missed the multi-player agreement basis of the reward process. Most critique concerned the TapRhythm module for collecting rhythm. For many players this task was unclear and very complex to involve in within the short given time.

4.2 Case study: Spot the Odd Song Out

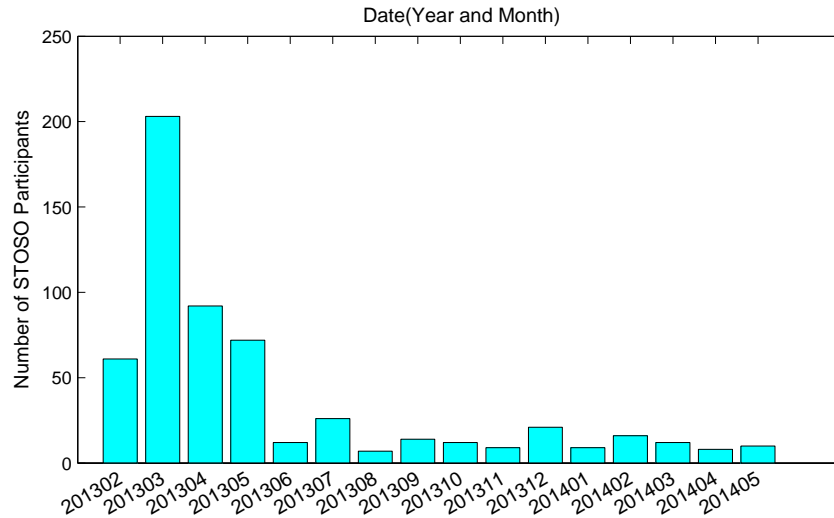


Figure 4.8: Logged-in users in Spot the Odd Song Out per month from February 2013 until May 2014.

4.2.5 The CASimIR dataset - Data Collected with Spot The Odd Song Out

The Spot the Odd Song Out game is available publicly on the World Wide Web and Facebook since March 2013. Note that the data analysed in the following text is limited to the amount collected until May 8th 2013, when first experiments were performed on a snapshot of the data.

As of 8th May 2013, 356 unique players played Spot the Odd Song Out, with 256 players gathering more than zero points. More than a third of those logged in using the Facebook social network, and more than 68% provided valid age and country information. Within this time, we have collected the number of annotations expressed in Table 4.1:

Similarity	TapTempo	TapRhythm
1928	1080	492

Table 4.1: Number of annotations collected per module (8th May 2013).

The number of similarity data is the largest amongst the data types collected through Spot the Odd Song Out due to the number of similarity modules contained

in each game. In general, although games can be joined and left at any position, the distribution of modules in a match (3,2,1) is very well reflected in the number of collected data. The data as counted in Table 4.1 is shared under a Creative Commons license and can be downloaded online¹. We furthermore plan to provide updates on the dataset.

4.2.6 Music Similarity Data

The Spot the Odd Song Out similarity module collects relative similarity data as described in Chapter 3. Choosing a clip C_k out of three clips C_i, C_j, C_k in the similarity module results in 2 similarity statements: Clip C_i and C_j are more similar to each other than C_i is to C_k , and Clip C_j and C_i are more similar to each other than C_j is to C_k (see Equation (3.2)). The Question Working Set of triplets was designed to create a densely interconnected set of similarity data, where transitivity of similarity relations can be capitalised: The Song Working Set for this task was initialised with only 20 random songs across several genres. Data from test studies was used to provide automatic answers (getAIVote) for the known triplets. For the full publication of the game, the Song Working Set and triplet Question Working Set were expanded to provide a controlled number of Questions containing songs of the same genre. The Song Working Set was further expanded when 70% of currently active songs were annotated more than 7 times, with half of the added triplets having consistent genre. This resulted in a total of 165 referenced clips on 8th May 2013 through the automatic Working Set expansion.

The controlled expansion of the Song Working set was effective: While the number of clips directly or transitively related to each other via the similarity relation $>^{\text{sim}}$ in the MagnaTagATune dataset was maximal 3 (see [pub:9]), Table 4.2 shows that the majority of the clips in the CASimIR similarity data are related to at least 5 other clips. This also is clearly visible in the visualisations of those datasets (see Figures 10.3 and 10.5 on pp. 218 of the appendix).

Interconnections between clip pairs come at the cost of fewer new clips: the current CASimIR similarity dataset contains only 165 clips, while MagnaTagATune

¹http://mi.soi.city.ac.uk/datasets/aes2013casimir/stosoaes2013_r1298.zip

4.2 Case study: Spot the Odd Song Out

# Pairs	≥ 40	≥ 20	≥ 5	3	Σ
MTT	0	0	0	1011	1011
CASimIR	155	211	421	751	751

Table 4.2: Number of connections to other clip pairs for MagnaTagATune (MTT) and CASimIR dataset after filtering inconsistent data.

references about 500 clips with the same amount of votes. Also, control questions with a pre-determined odd song out and two identical clips were inserted into the game to assess the overall quality: 7% of monitor triplets containing 2 identical clips (A,A,B) were failed by participants, still hinting at good data quality in this task.

A later snapshot of the collected similarity data, which makes use of the participant attributes collected with the game data, was presented at the DMRN+8 workshop and is described in sections 4.3.1.2 and 8.7. The graphs in Figures 10.5 and 10.6 on pages 220 and 221 show the current (01/05/2014) state of the CASimIR similarity dataset.

4.2.7 Tempo and Rhythm Data

Our publication [pub:1] with KTH Stockholm presents the first example of a collaboration being built on the CASimIR framework: For research in speed and tempo perception, two modules (see Figure 4.6) were added to the Spot the Odd Song Out game by Bellec et al. [pub:1]. The modules were the first to employ live recording of keyboard and mouse / finger tapping data via a HTML5-based music GWAP.

The Song Working Set for this question type was initialised with 10 songs from the Million Song Dataset and 10 synthesised clips from the Friberg Dataset (FRI). It was expanded if 70% of currently active songs were annotated more than 10 times. After filtering of duplicate taps due to client problems, we report 1080 TapTempo annotations. For the following Section 4.2.7, we only consider annotations with a standard deviation of the inter-tap-intervals below 25%, and single timing deviations below 35%. This leaves 668 annotations representing tempi between 30 and

300 beats per minute. Analysis by Bellec et al. [pub:1] shows a good correspondence of tapped tempo with the expert annotations for the FRI dataset.

The published CASimIR dataset also contains 492 rhythm tapping performances. The interpretative freedom of the task lead to different understandings reported by participants which also could be identified in the collected data: Some player tapped regularly, some repeated a particular pattern, other performances changed and added rhythmic layers during the recording. Bellec et al. [pub:1] showed that by aggregating over annotations for each song, time signatures can be extracted. However the rhythm data would not allow for further deductions.

#Annots	≥ 3	≥ 10	≥ 20	≥ 40
Tempo	47	37	27	5
Rhythm	44	29	5	0

Table 4.3: Number of songs being annotated at least #Annots times for TapTempo and TapRhythm.

4.3 Towards Culture-Aware Similarity Modelling

As described in Section 4.1.2.1, CASimIR enables the collection of participant attributes through social networks and additional login forms. Moreover, anonymised participant information is linked to any annotations provided by a participant during a game or survey.

For the task of similarity learning, in addition to the similarity data itself, we gain annotations of that data, enabling the verification, grouping and further analysis of distinct subsets of the collected data.

As Bogdanov [11] elaborates for music preference in music recommendation systems, the context in which listeners experience music has a strong influence in whether further song recommendations are seen as appropriate and pleasant or not. McDermott [55] summarises influences outside the acoustic material as follows:

4.3 Towards Culture-Aware Similarity Modelling

“Context matters, as does experience – we like things we have heard before. Music preferences additionally involve the interaction of personality traits and emotional content, aesthetic principles such as optimal complexity, and physiologically realized episodes of peak emotional arousal.”

Leblanc [48] provides a model for variation in music preference, which amongst other such factors, includes what we summarise as short-time feedback processes within the mind during the listening experience. It becomes clear that the matter of music preference and similarity is affected by many factors not observable through today’s web-based GWAP. Still, Leblanc [48] mentions a many static influences that can be gathered through CASimIR, such as self-reported musical ability, musical training, sex, ethnic group, socio-economic status, and age relating to personal maturation. Social network integration even allows for deriving general information such as nationality and location of peer group, family, attended media, music preference and authority figures. More advanced gamified strategies may allow to test players’ attention, memory and auditory sensitivity in the future. Of course, wherever large quantities of such data are combined, ethics of privacy and data protection should become of primary interest.

Because of various reasons including personal data protection, to our knowledge, few similarity data is publicly available containing information about the associated provider of the data and their context. We here focus on cultural impact on similarity data, thus CASimIR aims to link annotations to cultural profiles rather than participant data. Spot the Odd Song Out collects basic information including a 5-year-quantised age, gender, country of location, nationality, spoken languages and musical experience or training where participants enter this data or allow for it to be collected from the social network. Furthermore, information such as an estimated country-based location of data input, time of day for data entry, and debugging information is associated with the similarity data. The amount and type information available varies between participants, which lies in the nature of the deliberate data provision by them.

We consider cultural information as it is encoded in participant attributes collected via the Spot the Odd Song Out game. Considering the later inclusion in similarity

models, collected attributes can be grouped as follows: Quantifiable participant attributes such as age or length of music education allow for more or less direct representation in numeric values, whereas categorical information such as favourite genres, bands, and language identifiers require further mining.

4.3.1 A Comparative Approach for Cultural Modelling

The cultural information can be employed on a super-model level, relating multiple similarity models for cultural subgroups: Here, several models are trained as described in Chapter 6, but only on subsets of the similarity data which have been determined based on the cultural attribute data. The sets can either be selected to contain similarity votes from users sharing a certain combination of cultural attributes, or picked randomly for later analysis of algorithm performance. Either explicitly (as with cultural information based predefined data subsets) or implicitly (when testing random subsets for model performance), the combinations themselves constitute the definitions of potential cultural groups. Now, models are adapted specifically to the similarity data for the data subsets, and the combination of all of the similarity measures constitutes a super-model. When applying the model in recommendation, a (e.g. linear) combination of the sub-models' outputs can be used to determine the similarity of music clips given the cultural information about the user.

Alternatively to training separate models for subsets of the data, the cultural information could be directly included at the model level. Cultural attributes would become part of the feature space now describing both the song and the participant. Unfortunately, the multiplicity of possible cultural attributes and their values requires more data than has been collected through Spot the Odd Song Out at this point.

4.3.1.1 Data Preparation

The creation of similarity data subsets necessitates the usage of implicit similarity measures on the user attribute data: If we are not interested in creating as many

4.3 Towards Culture-Aware Similarity Modelling

subsets as we have unique profiles, clusters of user attribute data have to be determined which then allow for the selection of associated similarity data subsets. For quantifiable participant attributes, this process is quite straightforward, as it only demands for numerical intervals to be defined or detected using the underlying dataset. For categorical participants attributes, similarity has to be defined given external information about the specific annotations. The categories may be selected manually as well, as in the following paragraphs which deals with the comparison of similarity models over geographical regions. In any case, the selection and grouping of relevant attributes imposes a structure on the cultural attributes of the participants and the music similarity constraints, and thus predetermines the possible cultural relations to analyse in the data.

4.3.1.2 Geographic Data Subsets of CASimIR

As a first experiment towards culture-aware music similarity models, we chose to analyse location as a specific culture indicator for further investigation. Location of input is the most frequently annotated user attribute in the CASimIR similarity dataset, as it was gathered from users' Internet Protocol (IP) addresses by matching them to local GeoIP¹ database at the time of input. We did not expect a large influence of this attribute on the similarity votings, as pop music is a macroculture² existent and communicated on a scale exceeding our scope of single countries. Still, local subcultures and differences in appreciation of styles exist.

From the similarity data collected via Spot the Odd Song Out until 15th November 2013, four subsets were selected, each containing similarity votes from one specific European country. For the experiments below, we chose the countries of France (Fr), Germany (De), Sweden (Sw) and the United Kingdom (Uk) for reasons of available dataset size and to maximise comparability between the datasets. The similarity data provided from participants located in these countries was used to generate the *single country data sets* \hat{Q}^{De} , \hat{Q}^{Fr} , \hat{Q}^{Sw} and \hat{Q}^{Uk} .

¹http://www.maxmind.com/en/geolocation_landing

²See Slobin [86] for a terminology on cultural groups.

After collection and combining, inconsistencies were removed from these similarity datasets as described in Section 3.1.2 resulting in the cycle-free similarity datasets $Q^{\text{De}}, Q^{\text{Fr}}, Q^{\text{Sw}}, Q^{\text{Uk}}$.

	Q^{De}	Q^{Fr}	Q^{Se}	Q^{Uk}
constraints	459	463	309	411
clips	151	152	123	151

Table 4.4: Number of unique constraints and clips contained in each of the four country datasets.

As a first experiment towards hierarchical modelling of music similarity across cultural groups, the experiments reported in Section 8.7 analyse how well specific models can be trained on the single-country subsets. We then go on to analyse how the performance of the specific models compare Q^{De} to non-specific general models trained on the complete dataset. Furthermore, a comparative analysis is presented comparing the individual country model to a general model for all data. This reveals some specific musical facets important for the adaptation to the Q^{De} dataset.

4.4 Conclusions

The new presented framework provides support for the creation of multi-player games and surveys on the web using modern cross-platform technologies such as HTML5. A means for efficient collection of music annotations, CASimIR and the presented GWAP methods represent a solution to research question *rq:5*. In particular, the open source framework delivers the following prerequisites for a fast game development:

- An overall architecture for GWAPs
- A back-end API and example implementation for data management
- A game framework for web and social network games for desktop and mobile platforms
- A stable game implementation, the Spot the Odd Song Out game.

4.4 Conclusions

It is designed to reduce the development effort for creating a GWAP, especially for researchers whose core interest is not web development. Collecting human data is essential for research into the semantics and perceptual effects of music, such as music similarity.

An important novel aspect of our GWAP architecture is the separation of data management and game front-end: The CASimIR back-end organises the media and collected annotations, thereby providing methods for dynamically reacting to participant numbers, making experiments scalable from the start of their design. The clear definition of annotation collection tasks and formats through the API facilitates the independent development of different user interfaces. User interfaces can be built with the CASimIR game front-end, which also allows for multi-player synchronisation and social network integration. Adding new types of data collection questions to the game front-end is facilitated through a modular structure, enabling our partners at KTH Stockholm to collect 2000 annotations including tempo and rhythm data.

The analysis in Section 4.2.5 encourages that our framework is enabling the collection of music annotations via GWAP: We successfully acquired a considerable number of over 250 unpaid participants in a short time, particularly by reaching them via multiple platforms (different browsers, two game servers, media streaming and Facebook). The quantity of more than 2000 similarity votes collected is sufficient to enable first similarity experiments on cultural groups as described in Section 4.3. Results of these experiments will be presented in Section 8.7, including an analysis of regional specifics in importance of music descriptors for music similarity. Regarding the quality of collected data, a certain control of the experimental process can be exerted on the side of the participant interface, and quality can be measured using control questions. Still the tap rhythm module showed the difficulties in providing an intuitive user interface where no external control or explanation can be given. The combination of different tasks within one match proved to be well-received according to our collected feedback, and players very consistently played through the whole match. In future work, a selection of specific games might allow players to exclude less motivating modules, reducing their possibly negative effect on the motivation to play additional rounds.

The decline in usage after the first very active months corresponding to the our reduced marketing points out the importance of marketing and other measures to keep participants interested and motivate their return. Feedback collected shows that most users enjoyed playing the game, but few returned more than on two days. Although Spot the Odd Song Out allows for customisation of the game with avatars and individual genres, only few users used the provisions, pointing to a need of further research in motivation of users over larger periods of time. Provision of marketing that finally achieves a snowball effect in participant acquisition certainly has to be accounted for when comparing the costs of creating a GWAP to a traditional survey with paid participants. Most users found out about the game through friends and group posts on Facebook. It is indicative of the limitations of zero-funding marketing that our participant locations were strongly related to the collaborating researcher's social networks, with strongest participation in the four European countries mentioned in Section 4.3. Here, our framework provides new methods for dynamically adapting the test data (questions) to unknown participant numbers.

CASimIR supports and encourages the reuse and a sharing of data. By creating different annotations on audio data in a coordinated way, it helps provide more data research that integrates and relates the different annotations. Therefore, the CASimIR system and the collected data are available as open source/data and we hope for it to enable more data collection and more collaboration and sharing of datasets among the research community.

The basis of any culture-aware similarity modelling though lies in the general methodical ability to adapt computational models to the collected similarity data as such. Tackling this topic, the following sections will comprise the central modelling part of this thesis, with a general overview of adaptive models, new and existing methods for training them, and the representation of music clips within the models. To this end, we will now discuss acoustic and cultural descriptors for audio clips in large datasets such as MagnaTagATune and the Million Song Dataset.

Is it high?
Is it low?
Is it in the middle?
Is it soft?
Is it loud?
Are there two?
Are there more than two?
Is it a piano?
Why isn't it?
Was it an airplane?
Is it a noise?
Is it music?

(John Cage, 1961)

5 Features for Large Online Datasets

This chapter discusses computational representations of music clips, further referred to as *features*, for the application of similarity modelling. In the above text, we discuss the representation of similarity data via graphs and similarity constraints, and the users providing it by means of attributes provided by them. For measuring or predicting similarity, the computational representation of music determines the facets such as physical, musical, and cultural attributes that can contribute to the similarity assessment. It is therefore necessary to carefully choose features when considering the optimal models for music similarity searched for in *rq:1*. [pub:6] [pub:9] [pub:7]

As depicted in Figure 5.1 our signal flow for modelling music similarity is divided into two stages: Firstly, features are extracted from available representations of music such as audio recordings, tag annotations and further metadata. In the second stage, these features are then fed into models based on classifiers or regression for either training or prediction of similarity.

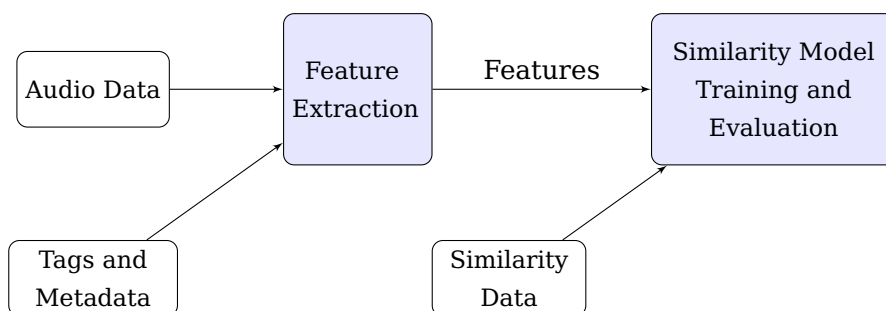


Figure 5.1: Common signal flow diagram for music features and similarity data.

Unfortunately, the audio recordings themselves are not always available to the researcher for analysis. Especially with popular and commercial music, this is because of copyright restrictions, but other limitations can be set by available means for storage and distribution of the material. Both the MagnaTagATune and Million Song Dataset datasets tackle this issue by providing precomputed features. This has enabled music research on a large scale which resulted in numerous publications¹. The mentioned datasets utilise features available via the The Echo Nest Analyse API. The availability of such pre-computed features allows for reproducible and comparable experiments across large datasets. Features are often freely available, and can be shared more easily – in terms of legal restrictions and data volume – than the audio itself. Fixed datasets with features furthermore define a standard for further processing and provide for better comparability, as often implementations of the same feature extractors differ in small but relevant details. Still, the types of features available via APIs or pre-computation are limited when no access to the acoustic information is given. Furthermore, in the case of The Echo Nest, the feature extraction process is not transparent. It would be beneficial if, instead of sharing the feature data itself, usage of gold-standard open implementations or even processing facilities can complement the sharing of fixed datasets of music features in the future. Further research on more configurable extraction APIs is pursued through the NEMA project [103], the SAWA² framework and the ongoing Digital Music Lab³ is aimed at tackling this with remote services for configurable

¹see <http://dc.ofai.at/browser/all?q=magnatagatune%20dataset> and <http://dc.ofai.at/browser/?y=all&q=million+song+dataset> for related publications.

²<http://www.isophonics.net/sawa/about>

³<http://dml.city.ac.uk>

5.1 Processing Features from The Echo Nest API

feature extraction.

To facilitate the reproduction of our evaluation (see research question *rq:3*) on the Million Song Dataset and MagnaTagATune datasets, we only use content-based features derived from the The Echo Nest data as described in the following Section 5.1. The usage of similar features for MagnaTagATune helped our collaborative work in Wolff et al. [pub:9]. For tag and metadata, we combine this information with information available from third parties via free APIs.

Currently, holistic approaches for classification on audio including both feature extraction and classification procedures within Deep Networks promise an alternative paradigm to the two-stage signal flow displayed in Figure 5.1. As audio data is difficult to acquire for the Million Song Dataset, for now, we provide and evaluate a promising novel method for using RBM networks with available, derived features in Section 5.3.2.

5.1 Processing Features from The Echo Nest API

The Echo Nest Analyse API allows for the remote extraction of many common audio features¹ from submitted audio as well as the retrieval of feature data and metadata for known tracks. Both the MagnaTagATune and Million Song Dataset datasets contain features from this API as part of the freely available data. Using this data for experiments allows for easy reproduction of results, as the features are available both via the fixed dataset and online API, and no access to copyrighted content is necessary. The features can also be freely shared with experiment results. In this way The Echo Nest provides a quasi-standard for features that can be used for music analysis where audio content is not accessible, although the range of applications is limited by the types of features available from The Echo Nest [44].

Features are provided at the cost of the exact extraction procedure of some of the features being held as a company secret. Largely though, the features are commonly known in the MIR community and as such can be largely approximated,

¹http://developer.echonest.com/docs/v4/_static/AnalyzeDocumentation_2.2.pdf

or replaced by improved versions as shown by Khadkevich and Omologo [44]. The features provided cover a large range from low-level such as energy and loudness measures to high-level features such as mode and tempo. Here, the quality of the higher-level features such as mode, key and particularly meter depends on the performance of the underlying classifiers. Our initial experiments suggested that especially the beat, meter and tempo features seem to work well only for clips with a very steady meter and no long intros. For the music contained in the MagnaTagATune dataset, with a large amount of world music and electronic drone sounds, the beat-based features do not perform well and were therefore not included into our experiments.

On the other hand, the chroma (called pitch features in the API) and timbre features have been widely used for experiments related to music similarity [57, 100] and alignment of different recordings [71] and were therefore chosen as the basic audio features in our music similarity models.

5.1.1 Chroma and Timbre Audio Features

For our initial experiments ([pub:4, pub:6]), we only used the *chroma* and *timbre* information, encompassed in the MagnaTagATune dataset. Chroma features summarise a range of the spectrum into 12 coefficients corresponding to the tones of the western chromatic scale. These features allow for representation of some harmonic context without providing explicit analysis. Similarly, the timbre features included in the dataset provide information that correlates to timbre in sounds. These mid-level features have been extracted using The Echo Nest API. As many toolboxes exist for chroma and timbre features, choosing this information as a basis for our features allows for an easy extraction of similar acoustic features independent of the web-based, and regularly updated API of The Echo Nest.

Both types of features are extracted on a segmentation of the clips. The segments correspond to temporal regions with relatively steady frequency distribution. Details of the segmentation algorithm are given by Jehan [39]. For each of these segments, the MagnaTagATune dataset contains a single chroma and timbre vector, each $\in \mathbb{R}^{12}$.

5.1.1.1 Aggregation via Averaging

The features noted above vary over time, and the number of feature values depends on the specific time windows they have been calculated on. Other, especially higher level features such as mode or key exist only on the time frame of a whole song. In order to use the time-variant features to address similarity at the song level, we aggregate their information over time using statistics such as mean and standard deviation. This allows features based on different time window sizes to be combined into one feature vector representing a whole music clip. We here present two simple strategies of feature aggregation: Below, chroma and timbre values are aggregated to average values per clip. We then present a method that uses clustering for defining multiple typical feature constellations (e.g. chords for chroma) per clip, thereby covering some of the variance. Although other types of aggregators, e.g. using code vectors or temporal models exist, we here restrict ourself to basic methods, that do not require the dataset to be fixed at the time of model training. Besides variance as discussed in Section 5.1.1.3, simple statistics such as median can be tested in place of the presented averaging strategy, but this is outside the scope of this thesis.

As the timbre and chroma features are calculated on multiple segments for each clip, they need to be aggregated to the 30 seconds time scale of a clip. As by Stober and Nürnberger [91], a straightforward approach is to take the mean and variance of the features over time and use these values for representing the clip. In our experiments, the variance of chroma and timbre features has not proven helpful for modelling clip similarity. Thus, in our experiments (see Section 8.4.1), we only evaluate features based on the means of chroma features and timbre features respectively. Earlier tests including variance statistics did not improve results. For each clip $c_{(i)}$, $i \in \{1, \dots, 1019\}$, a single timbre average $t_{(i)}^1$ and chroma average $c_{(i)}^1$, $t_{(i)}^1 \in \mathbb{R}^{12}$ and $c_{(i)}^1 \in \mathbb{R}_{\geq 0}^{12}$, are extracted.

5.1.1.2 Clustered Aggregation

In Section 8.4.1 we also test using the means of 4 cluster centroids $t_{(i)}^j \in \mathbb{R}^{12}$, $c_{(i)}^j \in \mathbb{R}_{\geq 0}^{12}$, $j \in \{1, \dots, 4\}$ being extracted for timbre and chroma features on each clip

$c_{(i)}$, $i \in \{1, \dots, 1019\}$. The incentive for this approach is to preserve some of the variety of harmony and timbre in the clips. The motivation is that multiple chords are expected throughout a 30 second MagnaTagATune excerpt. The centroids are extracted using a weighted k-means variant, which accounts for the differing temporal lengths of the individual segments: centroids are influenced more strongly by feature data from longer segments. The final relative temporal weights of the cluster centroids are saved in scalars $\lambda(c_{(i)}^j), \lambda(t_{(i)}^j) \in [0, 1]$. The centroids are then concatenated ordered by descending weight. This allows to the representation of multiple centroids in a single feature vector.

5.1.1.3 Variance

For the above aggregation strategies, the variance from the average or cluster centroid can be calculated and integrated into the feature vector using our implementation (see Section 7.3). As our experiments showed no or negative impact of this additional information on model training, the variance for chroma and timbre means is not included in the audio features used for experiments in Chapter 8 if not stated otherwise.

5.1.1.4 Normalisation and Clipping

For later use of the feature vector with classification and regression methods certain restrictions on range and distribution of the features have to be satisfied. Too small or large values may pose numerical problems during optimisation. Also, the relation of ranges in feature dimensions is of importance: When modelling similarity through a weighted sum of individual feature differences, a later analysis of weights requires all feature dimensions to occupy the same value range. The process of *normalisation* may refer to the equalisation of ranges, mean, and possibly variance across feature dimensions, taking into account a set (possibly the complete data set) of feature instances. We call this *global normalisation*. Alternatively normalisation is also applied on single-instance basis for some features, scaling and shifting feature dimensions by constant factors to fit the values in the single feature vectors to a defined range (*local normalisation*).

Local normalisation is used with centroids or averages of the chroma features, which are normalised on a per-clip basis to fit the interval $[0, 1]$ using

$$\tilde{c}_{(i)}^j = \frac{c_{(i)}^j}{\max_k(c_{(i)_k}^j)} \quad (5.1)$$

for clusters $j \in \{1, \dots, 4\}$, clips $c_{(i)}$, $i \in \{1, \dots, 1019\}$ and chroma dimensions k . This normalisation focusses the later comparison of features on the relative shape of the chroma distribution, as energy levels are now scaled to the same extrema across clips .

Global normalisation and clipping is applied to the timbre features: The timbre data is provided in an open numerical range $[-\infty, \infty]$ by The Echo Nest. This also applies to the extracted centroids and averages. In order to adapt the timbre feature data's range to those of the chroma and other features, the values are clipped to a maximum threshold on a global level. The clipping threshold was chosen such that 85% of the timbre data values for all clips relevant for the similarity dataset are preserved. Afterwards, the timbre data is shifted and scaled to fit $t_{(i)}^j \in [0, 1]$.

5.1.2 Further Energy-based and Higher-Level Audio Features

In early experiments ([pub:4, pub:6]), we restricted the set of features for similarity learning to the easily extractable features mentioned above. Slaney et al. [84] have shown a feature set complementary to those described above to also facilitate the adaptation of music similarity measures to ground truth based on annotations. In their experiments, the segment-based chroma and timbre features are not used. Instead, they use those features from the Last.fm API which are already given on the temporal scale of the clips, as well as statistics for segment and beat locations and their frequencies. These features are the result of different classification, structure analysis and optimisation algorithms for music, which have been described in detail in Tristan Jehan's PhD thesis [39].

In the experiments presented in this paper, we extend our low-level features by reproducing the features used by Slaney et al. [84], as far as the relevant information

is available in the MagnaTagATune and Million Song Dataset datasets. The remaining features have been omitted to ensure reproducibility of the experiments. See Table 5.1 for a list of the features incorporated in our study.

segmentDurationMean	tempo
segmentDurationVariance	tempoConfidence
timeLoudnessMaxMean	beatVariance
loudness	tatum
loudnessMaxMean	tatumConfidence
loudnessMaxVariance	numTatumsPerBeat
loudnessBeginMean	timeSignature
loudnessBeginVariance	timeSignatureStability

Table 5.1: Features from Slaney, Weinberger and White [84] used in our experiments.

Most of the features in Table 5.1 are directly taken from the features included in the dataset. The "-Mean" and "-Variance" features represent the respective statistical operation on the provided feature data, with no further processing besides a final normalisation, as explained in the following paragraph. The *beatVariance* feature represents the variance of the time between detected beats. If no beats are detected, the variance is set to zero. The *tatum* feature contains the median length of the inter-tatum intervals. Analogously, the *numTatumsPerBeat* feature results from the division of the median inter-beat interval by the tatum length as described above. As a fallback, if no tatum positions are detected, the *tatum* and *tatumConfidence* features are set to zero, while the *numTatumsPerBeat* feature is set to a default of 2.

Finally, each of these features is globally normalised over the values for the clips in the whole similarity dataset: The values are scaled and subtracted their minimal value, to result in a one-dimensional $s_i^j \in [0, 1]$, for clips $c_{(i)}$. The features are not whitened as described by Slaney et al [84], as we are interested in keeping the features' original associations to properties in music theory. Note that some of the features allocate only a limited number of actual values. For example, the *timeSignature* feature uses only values out of $\{\frac{0}{7}, \frac{1}{7}, \dots, \frac{7}{7}\}$.

5.2 Tags from Catalogue Annotations and Folksonomies

Although the above audio features capture information about the clips’ acoustic contents, further information is needed to more adequately represent associations and cultural context of the work. Such information includes user associations with the clip, in form of textual annotations, internet “like” data, and metadata. We introduce such contextual information on the clips via tag-based features. Such culture-related metadata has been shown to complement acoustical information present in the audio, e.g. in Novello et al. [68] and the experiments reported in Section 8.4.1.

5.2.1 Genre Tags



Figure 5.2: Tag cloud representation of the genre data added to MagnaTagATune from the Magnatune catalogue. The printed size of each genre corresponds to its frequency of occurrence in the dataset, while the spatial arrangement is not related to the genre data.

To this end we employ genre tags from the Magnatune label’s catalogue, which is available online¹.

¹<http://magnatune.com/info/api.html>

Genre information constitutes a special case of social tag data, as it is affected by the interplay of musicological standards and market- as well as consumer-based categorisations. Several theories have been proposed on the relationship of categorisation and similarity perception in psychology, underlining the strong relation of these cognitive processes [29, 32].

The Magnatune catalogue contains descriptions of the songs which have been associated to the MagnaTagATune dataset's clips: Each song is annotated with a sequence of 2-4 genre descriptions. Figure 5.3 on page 109 captures the almost hierarchical relationship of these genres in a graph: Iterating over all clips in MagnaTagATune, edges are drawn from genres on second or following positions to the genre listed first. The resulting graph shows that relation suggesting that genres are annotated starting from the most general to the most specific association. We assign these genres to the corresponding clips, using binary vectors $c_{(i)} \in \{0, 1\}^{44}$. Here, the positions corresponding to existing annotations of clip $c_{(i)}$ with genre $c_{(i)}^j$ are set to 1, whilst the other entries remain at 0.

The CAMIR framework generalises the representation described above to tag data from other sources such as online folksonomies created by users of social music networks. In particular, we have implemented and tested data gathered from Last.fm for replacing the genre features for the CASimIR dataset, but further filtering and processing of the tag data, similar to [90], is necessary for effective usage of that data.

This concludes the acoustic and tag-based features to be used for our experiments in similarity modelling. Although the above feature definitions include all the information available, the performance of models and model training is also affected by the way this information is represented in the feature vector and therefore in the model itself. The following section discusses two methods to achieve transformations of the feature representation, PCA commonly used for efficient data representation as well as Restricted Boltzmann Machines for adaptive feature transformation.

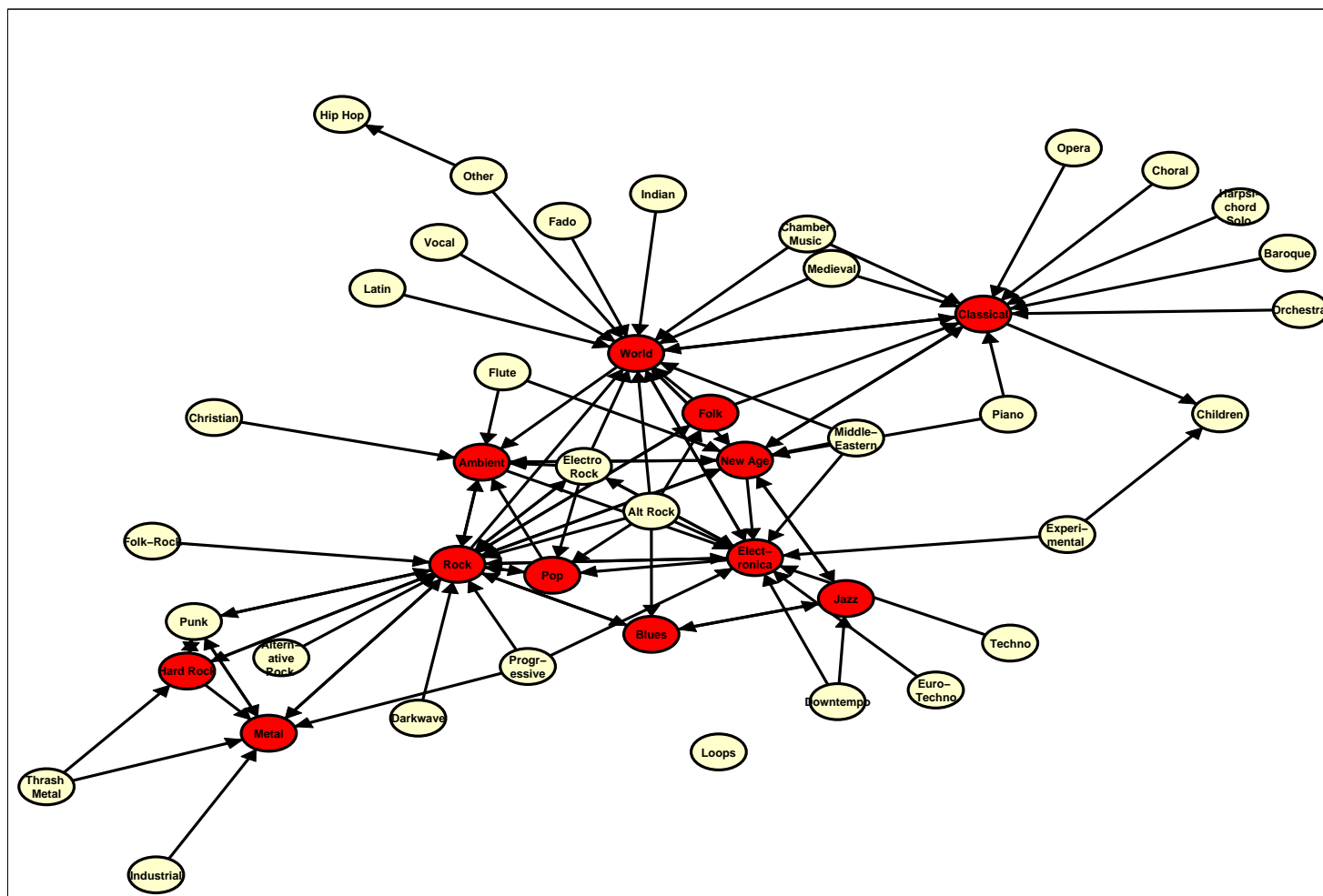


Figure 5.3: Graph displaying the genre hierarchies for the MagnaTagATune dataset. Arrows correspond to hierarchical subordination as deduced from the Magnatune annotations. Red nodes correspond to top-level genres.

5.3 Feature Transformations

After extraction of the feature information, general transforms can be applied to change the overall representation of the combined or single features. We now discuss PCA with the intention to reduce the dimensionality of large feature vectors and effectively equalise the resulting complexity of metric models based on different features. This allows us to compare the impact of the information inherent to those features, as well as the effect of feature dimensionality in Section 8.4.2.

Furthermore, the representation of the feature information as given to the model is crucial especially for rigid model architectures such as metric models. Although some feature pre-selection is included in the standard weighted Euclidean and Mahalanobis distance models, only existing, and in the case of the Mahalanobis distance, fixed interaction terms can be used to represent the clips and approximate the similarity data. Especially when using simple models, such as parameterised linear models trained with gradient ascent or SVM, limitations apply to what similarity relations can be modelled and how well complex data can be adapted to during the training process. Here, transformations of the features before training can be used to change the input feature space of similarity measures, thereby e.g. including interaction terms between individual features or maximising independence across the feature dimensions. This effect is relevant for both the PCA and RBM transformations discussed, and is primary focus in our RBM experiments in Section 8.4.3.

In all our experiments with feature transformations, the transformations are determined prior to the splitting of the data into test and training sets. This allows for a better adaptation to the data in total, and is particularly useful for the comparison of different representations on the largest available basis of data points. As discussed in Section 8.1.2 the knowledge of the whole datasets features in advance is only representative for a specific set of applications.

5.3.1 Principal Component Analysis

Principal Component Analysis (PCA) allows to transform the feature data from the original vector space onto a basis which allows for decorrelation of the resulting feature dimensions. The analysis involves the identification of the fraction of variance being explained by each component, which enables a reduction of feature dimensions while minimising the loss of information in the data. As the number of parameters of most similarity models depends on the feature dimension, Section 8.4.2 uses PCA for comparing model performance with reduced feature dimensionality. The PCA is computed directly on the feature data, based on the covariance of features across the dataset. Thus, the selection of clips and their features to compute the PCA influences the resulting transformation. For all experiments with PCA in this thesis, a fixed PCA transformation was determined upon the whole dataset prior to cross-validation.

For each of the single and combined feature types, a PCA can be performed. After sorting according to variance in the principal components, we reduce the dimensionality of the transformed features, keeping only a fixed number of components with greatest variance across the relevant dataset. In our experiments in Section 8.4.1, we use 12 and 52 dimensions based on the smallest dimensionality of the underlying single features.

After transformation into the principal component space, across the whole dataset, the individual feature components are shifted and scaled to fit the interval of $[0, 1]$. The impact of such normalising of the features was tested. We found that normalising the features after transformation generally improved the results of the resulting similarity model after training. In this way we gain sets of features containing various information types but sharing a constant dimensionality, which enables insight on the usefulness of different features independent of their dimensionality.

5.3.2 Transforming Features with RBM

Restricted Boltzmann Machines (RBMs) are artificial networks that can be used to [pub:2]

learn a non-linear (in contrast to the linear PCA) transformation of an input feature space. An RBM as depicted in Figure 5.4 consists of two layers: the visible layer V represents input features while the hidden layer H yields the transformed features after training of the RBM. The individual nodes are connected between, but not within the two layers through symmetric connections w_{ij} .

As described in Section 2.5.4.1, RBMs have recently been successfully applied for learning and extracting audio features, mostly based on spectrograms of audio recordings. For the datasets used in this thesis and the evaluation in Section 8.4.3, hardly any audio material can be analysed because of legal restrictions. We therefore provide a concept for using the available features from The Echo Nest for transformation with RBMs. We here report from a collaboration with Son Tran which has been published in [pub:2]. The RBM toolbox by Tran used for our experiments can be downloaded online¹.

RBM are probabilistic models. Thus, different training runs can yield different transformations even when used with the same training data. After being fed through the trained RBM, each feature in the transformed space captures the weighted non-linear combination of the original features. Therefore, the transformed features can represent relations between original features in the dataset, which are not available to a linear weighting of components in a distance metric but which can be useful for comparing the similarity between samples.

We now sketch the mathematical background of RBMs to allow for an understanding of their operation. Smolensky [87] describes RBMs as two-layer connectionist systems characterised by an energy function E

$$E(\mathbf{v}, \mathbf{h}) = - \sum_{ij} v_i w_{ij} h_j - \sum_i a_i v_i - \sum_j b_j h_j \quad (5.2)$$

to represent the joint distribution:

$$P(\mathbf{v}, \mathbf{h}) = \frac{1}{Z} \exp(-E(\mathbf{v}, \mathbf{h})) \quad (5.3)$$

with the partition function $Z = \sum_{\mathbf{v}, \mathbf{h}} \exp(-E(\mathbf{v}, \mathbf{h}))$.

¹<http://mi.soi.city.ac.uk/datasets/aes2013framework/>

5.3 Feature Transformations

In the above equation, v and h are notations of states of visible and hidden layers, the matrix W contains the connection weights $w_{i,j}$, and a, b represent the biases for the visible and hidden layers respectively.

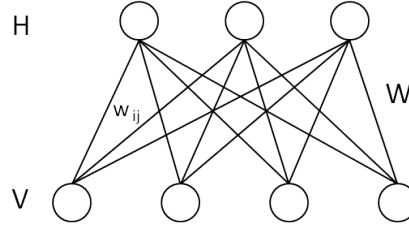


Figure 5.4: Restricted Boltzmann Machine with 4 nodes in the visible (V) and $hidNum = 3$ nodes in the hidden layer (H) as well as connection weights W .

Since all units in one layer are conditionally independent of each other, the state of a unit in the hidden layer depends only on the states of units from the visible layer and vice versa. The probability of a unit being activated is given by

$$P(h_j = 1|v) = \sigma\left(\sum_i v_i w_{ij} + b_j\right) \quad (5.4)$$

$$P(v_i = 1|h) = \sigma\left(\sum_j h_j w_{ij} + a_i\right), \quad (5.5)$$

where $\sigma(x) = \frac{1}{1+\exp(-x)}$ represents the logistic sigmoid function, which introduces the non-linearity into the transformation represented by an RBM.

The parameters $\theta = \{W, a, b\}$ of an RBM are determined using an unsupervised learning process: Training an RBM means to maximise its average log-likelihood $\hat{\ell}$ (or equivalently the product of probabilities). This can be done given a set of independent and identically distributed samples $\mathcal{V} = \{v^{(1)}, v^{(2)}, \dots, v^{(N)}\}$ and maximising

$$\hat{\ell} = \frac{1}{N} \ln(\mathcal{L}(\theta|\mathcal{V})) = \frac{1}{N} \sum_{k=1}^N \ln P(v^{(k)}|\theta) \quad (5.6)$$

using gradient ascent. Here, the number of units in the visible layer is determined by the dimensionality of the input features, while the number of hidden units $hidNum$ remains a hyper-parameter.

However, it is not easy to compute the exact gradient from the log-likelihood because of the need to compute the partition function Z . In particular, the gradient ascent requires an expectation of data sampled from the model as

$$w_{ij} = w_{ij} + \eta(\langle v_i p(\mathbf{h}_j | \mathbf{v}) \rangle_0 - \langle v_i \mathbf{h}_j \rangle_\infty). \quad (5.7)$$

Here, $\langle \cdot \rangle_0$ is the average with regards to the data distribution and $\langle \cdot \rangle_\infty$ is the average with respect to distribution from the model. An approximative approach to this problem is to sample the states from the model using Markov Chain Monte Carlo (MCMC). This method, however, is very slow and unstable since the model needs to perform a long and unspecified pre-sampling process before reaching an equilibrium state and generating valid samples. In [35], Hinton proposed an algorithm named *Contrastive Divergence* (CD) showing that even with small number of pre-sampling steps – in fact even with a single step – the learning can approximately minimize the divergence between the data distribution and the distribution of the model:

$$w_{ij} = w_{ij} + \eta(\langle v_i \mathbf{h}_j \rangle_0 - \langle v_i \mathbf{h}_j \rangle_n). \quad (5.8)$$

If we now input feature values in the visible layer of an RBM, the resulting hidden layer activations represent a non-linearly transformation of those features. Here, the number of units in the hidden layer (*hidNum*) determines the dimensionality of the new feature space.

5.4 Conclusions

In this chapter we discussed methods for feature extraction and processing for similarity learning on large datasets. After describing the role of feature extraction in the common signal flow of Music Information Retrieval, we presented feature extraction techniques based on pre-computed features by The Echo Nest. As these features are included in major open music datasets such as MagnaTagATune and the Million Song Dataset, the methods described above allow for reproducible experiments including a variety of features without requiring access to the possibly copyrighted audio recordings. This allows for reproducible evaluation of models on the mentioned large datasets respective research question *rq:3*.

5.4 Conclusions

The different features presented here present the basis of representing clips to similarity models. The variety in methods presented here as well as their later evaluation contribute to the wide spectrum of approaches considered to answer *rq:1*. Features extracted include a large range of commonly used audio descriptors as well as a new cluster-based representation of chroma and timbre information.

We extend the audio-based clip descriptions with binary-vector context features based on genre tags associated to the clips. The genre tags being mined from the Magnatune catalogue are a new extension to the MagnaTagATune dataset and available online through the CAMIR framework¹. They complement existing audio features with cultural information. Normalisation is performed to match the value range of [0,1] for all features, but other value ranges can be easily achieved via parametrisation. The acoustic and genre-based features can be combined which proves to be effective for learning similarity in our experiments (Section 8.4).

Transformation of features after extraction using the PCA method for decorrelation of feature input can allow for a more efficient representation of the information. We introduced a new method for transforming features via RBMs. In contrast to PCA, the projection allows for a non-linear combination of features, which is parametrised in an unsupervised machine-learning step. Our experiments in Section 8.4.3 show that similarity modelling can be improved with the novel RBM method and made more efficient with PCA. But transformation prohibits later musicological analysis of information captured by the models.

As this thesis focusses on the general modelling and learning of music similarity, some options to refine feature aggregation and representation, such as including temporal envelope information of chroma and timbre, are left untouched in favour of presenting a set of basic but easily reproducible features. Methods that can be explored are mentioned in Section 9.6. All methods for feature extraction are published as open source in the CAMIR framework as described in Section 7.3.

¹<http://mi.soi.city.ac.uk/datasets/camirframework/>

To understand, for example, how people organize social systems, we have to discover the principles that we create to make some societies intelligible

(Noam Chomsky, 1948)

6 Computational Models for Learning Music Similarity from Relative Data

Given the features as introduced in the previous chapter, it is now possible to define [pub:7] similarity measures on clips based on their feature representation. Simple similarity measures can be pre-defined, such as the Euclidean metric on the feature space. Our goal here is to model human similarity data as described in Chapter 3, and for this task parametrisable models for music similarity are needed. Analogously to the above quote, not only the parameters of the similarity models, but particularly the model types themselves and their training algorithms influence the performance of our learnt similarity models, determining what is salient and trainable information. As pointed out in the literature review, many methods for modelling similarity from absolute similarity statements exist, but comparably few have been developed to accept the relative similarity data this thesis deals with. We will here discuss state-of-the-art methods for learning from relative similarity data, with their first applications on human music similarity judgements. Following research question *rq:2*, we introduce new as well as extended methods for the task. A framework originally developed by Zheng et al. [106] for general regression will be generalised to enable methods based on absolute data to deal with relative similarity learning. The application of this new framework also allows us to introduce the concept of transfer learning – reusing similarity information from previously learnt music similarity models – during model training. In particular, we will present models based on generalised Euclidean metrics, Mahalanobis distance measures, Neural Nets, linear regression and regression trees. Our enhancement of modelling possibilities

will allow for a comparative evaluation of model performance regarding research question *rq:1*.

It is common to model similarity as the inverse of the distance of two clips, especially for metric models. Thus the similarity data of one clip pair (C_i, C_j) being more similar than (C_i, C_k) according to an implicit similarity relation y (see Equation (3.1)), corresponds to the constraint requiring the distance of clips C_i and C_j to be greater than for C_i and C_k :

$$\begin{aligned} (C_i, C_j) &\stackrel{y}{>} (C_i, C_k) \\ \Leftrightarrow \text{dist}(C_i, C_j) &< \text{dist}(C_i, C_k) \end{aligned} \tag{6.1}$$

Although perceptually the two relations may be different, the duality of the concepts works well for mathematical metric spaces: it allows for a straightforward adaptation of distance and metric learning approaches to similarity estimation, still leaving room to adapt to perceptual peculiarities during model adaptation. The more general regression methods discussed below would indeed allow for direct implementation of the concept of similarity, but for reasons of consistency and ease of integration we keep to modelling distance.

As summarised in Table 6.1, the representation of a clip’s feature information changes within the signal flow according to the context it is used in. Many of the methods discussed in this section will deal with the feature information in the combined form of clip pairs rather than involving separate feature data for single clips. To address this we extend on the idea of facet distances (see Stober [89]) as a mapping of two clips’ feature vectors towards a combined representation of their relationship.

6.1 Mapping Features to Model Input

Given the clip index I for all clips $C_i, i \in I$ and similarity information Q , Clip pairs (C_i, C_j) are represented by their *Facet Distance* vector $d_{(i,j)}$. Stober and Nürnberger [91] use the term *facet* to refer to a mapping of e.g. perceptual musical

6.1 Mapping Features to Model Input

features such as the “timbre” facet based on physical measurements like spectral centroid etc. In the following, the term *Facet Distance* will be used in a more general way, describing the mapping or combination of feature information into more complex sub-distances which then can be used to train the final distance measure.

A simple facet distance vector is given by the component-wise squared difference of the involved clip pairs’ features:

$$d_{(i,j)}^E = ((x_{(i)_1} - x_{(j)_1})^2, \dots, (x_{(i)_n} - x_{(j)_n})^2). \quad (6.2)$$

It allows for calculating the Euclidean distance (indicated by the E in d^E) of two feature vectors by

$$dist_1(x_{(i)}, x_{(j)}) = \sqrt{d_{(i,j)}^E \top d_{(i,j)}^E} \quad (6.3)$$

Further choices for d are the simple difference and the absolute value difference

$$d_{(i,j)}^L = x_{(i)} - x_{(j)} \text{ or} \quad (6.4)$$

$$d_{(i,j)}^{|L|} = (|x_{(i)_1} - x_{(j)_1}|, \dots, |x_{(i)_n} - x_{(j)_n}|). \quad (6.5)$$

Although the facet distance vectors used in this thesis have a dimensionality similar to the underlying features, this is not necessary. Stober and Nürnbergger [91] introduce facet distances that map large feature vectors to a single value, whereas the sub-space facet distances discussed below can greatly exceed the dimensionality of their input features.

	Clip	Clip Pair	Constraint
Clip notation	C_i	(C_i, C_j)	$(C_i, C_j) \succ (C_i, C_k)$
Feature rep.	Features $x_{(i)}$	Facet Distances $d_{(i,j)}$	Delta Function $\delta_{(i,j,k)}$

Table 6.1: Representation of clips, clip pairs and similarity data in terms of feature data.

6.1.1 Sub-space Transformations

Instead of using the direct differences of feature dimensions for the facet distance, we group consecutive features into subspace-features which are similar to patch-

based feature analysis in image processing [27]. By combining consecutive features, simpler models can be enabled to represent relationships between consecutive feature values. This method is relevant for similarity modelling where special relationships between consecutive features exist, as is the case with chroma features.

The components of the sub-space facet distance vector $d_{(i,j)}^\tau$ are given by

$$d_{(i,j)_o}^\tau = \sum_{p=o}^{o+\tau-1} (x_{(i)_p} - x_{(j)_p})^2 \quad (6.6)$$

where $i = 1, 2, \dots, N + \tau - 1$, and τ is the size of the subspace region, and o, p refer to positions in the input and output feature vectors.

6.1.2 The Delta Function

In many learning algorithms which are based on classification or regression, the training data of the relative similarity constraint $(i, j, k) \in Q$, referring to $(C_i, C_j) \stackrel{\text{sim}}{>} (C_i, C_k)$, is represented by a single delta vector

$$\delta_{(i,j,k)} := d_{(i,k)} - d_{(i,j)}, \quad (6.7)$$

containing the difference of the clip pairs (C_i, C_j) and (C_i, C_k) .

The function enables relative similarity constraints to be represented by a single delta vector (see Table 6.1), e.g. for the case of SVM in Section 6.2.1. More complex representations of multiple-clip relationships include the feature map ψ in Section 6.2.2 defining a ranking of clips.

6.2 Metric models

A generalisation the standard Euclidean metric was introduced by Mahalanobis in 1936 [54]. It is defined as

$$\text{dist}_W(x_{(i)}, x_{(j)}) = \sqrt{d_{(i,j)}^E \top W d_{(i,j)}^E}, \quad (6.8)$$

6.2 Metric models

Here, $x_{(i)}, x_{(j)} \in \mathbb{R}^N$ are feature vectors and $W \in \mathbb{R}^{N \times N}$ represents the *Mahalanobis matrix*, parametrising the similarity space. Davis et al. [20] show how each Mahalanobis matrix W induces a multivariate Gaussian distribution

$$P(x_{(i)}; W) = \frac{1}{\beta} \exp\left(-\frac{1}{2} \text{dist}_W(x_{(i)}, \mu)\right). \quad (6.9)$$

The standard definition [54] of the Mahalanobis distance defines W as the inverse covariance matrix of the underlying data, β as a normalising factor and μ as the mean of the feature data. With W depending on the feature covariance, the Mahalanobis distance can be used to calculate the distance of a point from the data average or any another point in the vector space in relation to the distribution of the data.

Instead of deriving W from the feature data itself, the distance learning methods below use additional similarity constraints to derive the optimal values for W . This allows for more freedom in the search for an optimal distance measure. The following properties of W can be useful in determining the general behaviour of a learnt dist_W . If W is the identity matrix, dist_W resolves to the Euclidean metric.

If W is diagonal, as in ($W_{i,j} = 0$ for $i \neq j$), the components of d are separately weighted within the distance function. This *Weighted Euclidean Metric* is learnt by the SVM, DMLR and regression algorithms introduced in Sections 6.2.1, 6.2.2.1 and 6.3.4 below. If W is positive definite, dist_W represents a metric, satisfying symmetry, non-negativity and the triangle inequality. Should W be positive semi-definite, allowing $\text{dist}_W(x_{(i)}, x_{(j)}) = 0$ for $x_{(i)} \neq x_{(j)}$, the resulting distance function is called a pseudometric [102]. The constraints on W regarding these properties differ amongst the algorithms below, as does the success in enforcing them, leaving dist_W an arbitrary function representing distance in some cases.

6.2.1 Support Vector Machines (SVM)

In [81], Schultz and Joachims present a metric learning strategy based on their SVM-Light framework¹. As for the standard Mahalanobis distance, Clip pairs (C_i, C_j) are represented by the clips' feature difference: for each constraint triplet

¹<http://svmlight.joachims.org/>

(i, j, k) , we consider the component-wise squared difference of the involved clip pairs' features. Here, the matrix W , as introduced in Equation 6.8 is decomposed into a kernel transformation A and a diagonal matrix W . We use the identity transform as kernel ($A = I$). Thus, $dist_W$ describes a weighted Euclidean metric.

The proposed algorithm optimises the distance measure by representing it as the length of the normal vector to the hyperplane $\text{diag}(W)$, separating triplets (i, j, k) from those representing the contrary information (i, k, j) . To this end, the $\delta_{(i,j,k)}$ with facet distance $d_{(i,k)} := d_{(i,k)}^E$ are used as constraints for the following optimisation problem:

$$\begin{aligned} \min_{w, \xi} \quad & \frac{1}{2} \|W\|_F^2 + c \cdot \sum_{(i,j,k) \in Q_{\text{train}}} \xi_{(i,j,k)} \\ \text{s.t.} \quad & \forall (i, j, k) \in Q_{\text{train}} : \langle \text{diag}(W), \delta_{(i,j,k)} \rangle \geq 1 - \xi_{(i,j,k)} \\ & w_{i,j} \geq 0, \xi_{(i,j,k)} \geq 0. \end{aligned} \quad (6.10)$$

This minimises the loss consisting of the sum of the per-constraint slack variables $\xi_{(i,j,k)}$ and the regularisation term $\|W\|_F^2 = \text{tr}(W^T \cdot W)$ using the squared Frobenius norm. Here, $\text{tr}(\cdot)$ denotes the trace of a matrix. The factor $c > 0$ determines the trade-off between regularisation and slack loss.

The particular implementation¹ by Schultz et al. calculates W in its dual form consisting of the support (basis) vectors $\delta_{(i,j,k)}$ and weights $a_i y_i$. W can be easily reconstructed using

$$\text{diag}(W) = \sum_{(i,j,k)} a_{(i,j,k)} y_{(i,j,k)} \delta_{(i,j,k)}. \quad (6.11)$$

The resulting Mahalanobis matrix W usually turns out to be positive semidefinite, but this is not guaranteed. Cases occur where some of the $w_{i,i}$ are slightly below zero. This behaviour has also been reported for the LIBLINEAR framework by Stober et al. [90], where they use the approach of Cheng and Hüllermeier [18]. In these cases, the measure does not qualify as a metric or pseudometric but may still perform well in terms of training error and generalisation.

¹<http://svmlight.joachims.org/>

6.2.1.1 Weighted Learning with SVM

The weight information for individual constraints as described in Section 3.1.1 can be used to prioritise certain constraints during training. In SVM-Light, this is implemented by weighting the individual slack variables $\xi_{(i,j,k)}$ in the penalty term of Equation (6.10).

6.2.2 Metric Learning to Rank (MLR)

Mcfee and Lanckriet [60] describe the Metric Learning To Rank (MLR) algorithm for learning a fully parametrised Mahalanobis distance based on the SVM^{struct} framework of Tsochantaridis et al. [93]. A implementation of the MLR algorithm in Matlab script language has been published by McFee¹. Specifically well-suited for use in retrieval environments, this method utilises rankings for the specification of training data as well as for the in-training evaluation of candidates for distance metrics. Such rankings assign a relevance position to each of the clips in our dataset given one of these as query item. Thereby they define a partial similarity relation y_q^* which is used as constraint in the MLR optimisation: For all constraints $y_q^* = (i, j, k) \in Q$, referring to the similarity relation $(C_i, C_j) \stackrel{y_q^*}{>} (C_i, C_k)$, the final metric should rank C_j before C_k , when the query is C_i .

For a set X of training query feature vectors $q \in X \subset \mathbb{R}^N$ and associated training rankings y_q^* , MLR minimizes

$$\begin{aligned} \min_{W, \xi} \quad & \text{tr}(W^\top W) + c \frac{1}{n} \sum_{q \in X} \xi_q, \\ \text{s.t.} \quad & \langle W, \psi(q, y_q^*) - \psi(q, y) \rangle_F \geq \Delta(y_q^*, y) - \xi_q \quad \forall q \in X, \forall y \in Y \setminus \{y_q^*\} \\ & W_{i,j} \geq 0, \quad \xi_q \geq 0 \end{aligned} \tag{6.12}$$

Optimization is subject to the constraints, creating a minimal slack penalty of ξ_q . A regularisation term based on the trace $\text{tr}(W)$ of the Mahalanobis matrix is used in the optimisation. c determines the trade-off between regularization and the slack penalty for the constraints. The Frobenius product $\langle W, \psi(q, y) \rangle_F$ in [60] assigns a

¹<http://cseweb.ucsd.edu/~bmcfee/code/mlr/>

score to the validity of a ranking y given the query q with regard to the Mahalanobis matrix W and feature map ψ which we discuss later. In the above equation, the difference of the fit of W to the training rankings $y_q^* \langle W, \psi(q, y_q^*) \rangle_F$ and arbitrary rankings $\langle W, \psi(q, y) \rangle_F$ is enforced to be greater than a margin: $\Delta(y_q^*, y)$, a measure of the difference of the rankings themselves. Several standard information retrieval performance measures can be used to compare the rankings y_q^* and y . We use the area under the ROC curve as measure for Δ .

In order to measure the fitness of W , analogously to $\delta_{(i,j,k)}$ in Equation (6.10), the feature map ψ maps the combination of query features and ranking into the same vector space as W . To this end, McFee's implementation of MLR uses the *partial order feature*

$$\psi(q, y) := \sum_{i \in X_q^+} \sum_{j \in X_q^-} \text{sig}_{i,j} \frac{\phi_{(q,i)} - \phi_{(q,j)}}{|X_q^+| \cdot |X_q^-|}, \quad (6.13)$$

$$\text{where } \text{sig}_{i,j} = \begin{cases} 1 & \text{if } (C_i, C_q) \stackrel{y}{>} (C_j, C_q), \\ -1 & \text{otherwise.} \end{cases} \quad (6.14)$$

Finally, $\phi_{(q,i)}$ captures the information of the facet distance vector $d_{(i,j)}$, as

$$\phi_{(q,i)} := -d_{(i,j)}^\top W d_{(i,j)}, \quad (6.15)$$

$$\text{with } d_{(i,j)}^{mlr} = (x_{(i)} - x_{(j)}), \quad (6.16)$$

$$\text{yielding } \phi_{(q,i)_{r,v}} = (x_{(i)_r} - x_{(j)_r}) * (x_{(i)_v} - x_{(j)_v}) \quad (6.17)$$

Thus $\phi_{(q,i)}$ captures the correlations of different entries of the facet distance vector as defined in Equation (6.2). The feature map ψ is added the difference matrix $\phi_{(q,i)} - \phi_{(q,j)}$ if C_i and C_j are ordered correctly by y . Otherwise the difference is subtracted. The squared Mahalanobis matrix directly operates on

$$\begin{aligned} \text{dist}_W(x_{(i)}, x_{(j)})^2 &= d_{(i,j)}^\top W d_{(i,j)} \\ &= \left\langle W, d_{(i,j)} d_{(i,j)}^\top \right\rangle_F \\ &= \left\langle W, -\phi_{(q,i)} \right\rangle_F, \end{aligned} \quad (6.18)$$

which MLR optimises. As there may be too many alternative rankings y for each training ranking y_q^* , only a few rankings $y \in Y$ are selected for comparison with the training rankings: A separation oracle is used for predicting the most violated constraints (see [40]).

6.2.2.1 Diagonal MLR (DMLR)

A variant of the MLR algorithm (Diagonal-restricted Metric Learning To Rank (DMLR)) restrains W to a diagonal matrix with $W_{i,j} = 0$ for $i \neq j$. Whilst still allowing for the weighting of different feature dimensions, rotations and translations in features space are ruled out by this restriction. For feature vectors $x_{(i)} \in \mathbb{R}^N$, using the facet distance $d_{(i,j)}^E$, this reduces the number of training parameters from N^2 to N .

6.2.2.2 Robust MLR

Recently, Lim, Mcfee and Lanckriet [51] published the RMLR (robust MLR) method that allows to learn a more sparse Mahalanobis matrix. In our experiments we were not able to improve the performance of music similarity models using this method, but it may be interesting for future research in feature importance for music similarity.

6.2.3 Weighted Learning with W(D)MLR

To our knowledge, no methods for weighted training with MLR have been published. MLR uses a 1-slack approach, prohibiting the direct weighting of individual constraints via their slack penalty like it can be done with SVM (Section 6.2.1).

Here we describe an adaptation of MLR for learning from weighted constraints: The weighting is implemented by repeating individual constraints by a factor proportional to their weight. Given the training data Q_{train} and weights $\alpha_{(i,j,k)}$ ¹, a

¹Weights of similarity data are related to input agreement, see Section 3.1.1.

maximal weight s is determined on the basis performance considerations. The $\alpha_{(i,j,k)}$ are then normalised and rounded to fit the interval $[1, s] \in \mathbb{N}$.

$$\alpha_{(i,j,k)}^* = \frac{\alpha_{(i,j,k)} * s}{\max_{(m,n,o) \in Q_{train}} (\alpha_{(m,n,o)})} \quad (6.19)$$

Now each of constraints in the training data is repeated $\alpha_{(i,j,k)}^*$ times, resulting in the pseudo-set.

$$Q_{train}^W = \{(i, j, k)_1, \dots, (i, j, k)_{\alpha_{(i,j,k)}^*} \mid (i, j, k) \in Q_{train}\}. \quad (6.20)$$

The repeated constraints gain their respective weights during slack aggregation, as the slack error is averaged along all (non-unique) training constraints. We call this method WMLR, and WDMLR in the case of DMLR, respectively.

Setting $s = \max_{(m,n,o) \in Q_{train}} (\alpha_{(m,n,o)})$ allows for an accurate representation of the weight data. This approach is obviously not efficient, but for the MagnaTagATune similarity dataset it is feasible. In our experiments, efficiency is improved with similar results by quantising the constraint weights. The performance of weighted learning with WMLR and WDMLR is presented in Section 8.6.1.

6.3 Adapting Methods to Relative Data

Many algorithms exist that can be used for learning similarity from absolute ground truth data. Such algorithms learn a distance function directly by mapping the feature vectors $x_{(i)}, x_{(j)}$ to a scalar output value $dist(x_{(i)}, x_{(j)})$. Considering the facet distance vectors $d_{(i,j)}$, this can be expressed as the choice of a function

$$\hat{dist}_P : \mathbb{R}^N \mapsto \mathbb{R}_{\geq 0}, \quad (6.21)$$

where P describes a parametrisation of the function family fulfilling the desired properties.

The far more conventional task of generalised regression or learning of functions using existing input and output values holds a rich variety of methods including

linear regression, regression trees, polynomial fitting, neural networks as well as specific metric learning methods such as ITML (see Section 6.3.6). The goal of this section is to render accessible these methods to scenarios where only relative similarity data is available.

Zheng et al. [106] describe a framework to learn ranking functions from relative relevance judgements, which they successfully apply to learning relevance data using gradient boosting trees. Their framework is applicable to any regression method which can approximate Equation (6.21), and the following paragraphs describe a slight adaptation of this iterative method allowing for a more flexible application and more precise parametrisation of the resulting learning algorithm.

For adapting a regression or similarity learning method to learning from relative constraints, our method requires the following components:

- The similarity model to be parametrised, e.g. a Mahalanobis distance
- The learning and evaluation methods for this similarity model using absolute similarity data
- A facet distance function to derive $d_{(i,j)}$ from two clips' features
- A target generator Υ determining absolute similarity constraints for learning
- An update rule Λ to accumulate the learnt information for each iteration into the model parameters

6.3.1 General Distance Functions

Firstly, the distance mapping function \hat{dist}_P (see Equation (6.21)) with parameter P has to be defined by using a specific similarity model. In the case of many learning methods, such as Neural Networks or Gradient Boosting Trees, the similarity model is implicitly determined by or contained in the training and evaluation methods themselves. Making generic regression methods such as linear regression or lasso regression applicable to relative similarity data allows for more flexibility in the selection of models. Zheng et al. [106] use Gradient Boosting Trees for modelling their ranking functions which is described in Section 6.3.5.

We present a simple new application of the framework using standard linear regression which allows to learn a weighted Euclidean distance measure. A main factor determining the nature of the resulting distance function or similarity model is the facet distance function $d_{(i,j)}$. When restricting the Mahalanobis matrix W in Equation (6.8) to be diagonal, the distance function reduces to

$$\hat{dist}_W(d_{(i,j)}) = \text{diag}(W)^\top (d_{i,j}^E)^2. \quad (6.22)$$

Thus, the distance function results in a linear combination (with factors $W_{i,i}$) of the squared feature distances. In order for our pairs of clips and their feature vectors $(x_{(i)}, x_{(j)})$ to be represented, arbitrary mappings $d_{(i,j)}$ including music-specific facet distance functions can be chosen to calculate the initial input to the similarity model.

6.3.2 Generating Absolute Training Targets

Given an initial training set $Q_{train} = \{(i, j, k) \mid (C_i, C_j) \stackrel{y}{>} (C_i, C_k)\}$, $i, j, k, \in I$, based on ground truth similarity data y , the following iteration operates on an active constraints set $Q^* \subset Q_{train}$, containing only those constraints which are violated by the current model determined by the parameters P (equals W in the example of linear regression).

$$Q^* = \{(i, j, k) \in Q_{train} \mid \hat{dist}_P(d_{(i,j)}) + \iota\tau \geq \hat{dist}_P(d_{(i,k)})\} \quad (6.23)$$

Here, $\iota\tau$ with $\iota \in \mathbb{R} \cap [0, 2]$, $\tau \in \mathbb{R}^+$ allow for enforcing a margin between the more and less similar clip pairs. Given the active constraints set Q^* , new training target distances are being determined on the basis of the current model output, using the target generator.

$$\Upsilon(a, b) : \mathbb{R}^N \times \mathbb{R}^N \mapsto \mathbb{R}^2, \quad (6.24)$$

where $a = \hat{dist}_P(d_{(i,j)})$,

and $b = \hat{dist}_P(d_{(i,k)})$.

6.3 Adapting Methods to Relative Data

The output of Υ represents the new training targets for the function $\hat{dist}_P(d_{(i,j)})$ and $\hat{dist}_P(d_{(i,k)})$. Different target generators are compared in Figure 6.1. A semi-linear margin target generator is used by Zheng et al. [106]:

$$\Upsilon_{semilin}(a, b) = (b - \tau, a + \tau). \quad (6.25)$$

From Equation (6.23) it follows that for any clip pair in Q^* we can assume that distance $a \geq b$. The training constraints created by this target generator then enforces at least the distance of the last iteration m

$$(\hat{dist}_{P_{m+1}}(d_{(i,k)}) - \hat{dist}_{P_{m+1}}(d_{(i,j)})) > (\hat{dist}_{P_m}(d_{(i,j)}) - \hat{dist}_{P_m}(d_{(i,k)})) + 2\tau \quad (6.26)$$

between the resulting absolute training targets after iteration $m + 1$. This target generator is called semilinear because the enforced minimum margin between the training examples is linearly growing with the difference of distance measures ($b - a$), but only where the difference, and thereby the error induced by the constraint, is positive.

Another target generator is the constant margin target generator whose resulting training targets have a minimum margin of 2τ . Here, both target values for the distances of clips (C_i, C_j) and (C_i, C_k) are centred around the current mean of their distance values.

$$\begin{aligned} \Upsilon_{const}(a, b) &= (\mu - \tau, \mu + \tau), \text{ with} \\ \mu &= \frac{a + b}{2}. \end{aligned} \quad (6.27)$$

The sigmoid margin target generator assigns relatively large margins to small violations of the distance constraint, at the same time assigning relatively small margins to constraints strongly violated and thus already difficult to enforce.

$$\begin{aligned} \Upsilon_{sigmoid}(a, b) &= (a - \tau, b + \tau), \text{ with} \\ \tau &= \gamma * \text{sigm}(\gamma * (b - a)) * \text{sigm}'(\gamma * (b - a)) \\ \text{sigm}(x) &= \frac{1}{1 + \exp^{-x}} \\ \text{sigm}'(x) &= \text{sigm}(x)(1 - \text{sigm}(x)). \end{aligned} \quad (6.28)$$

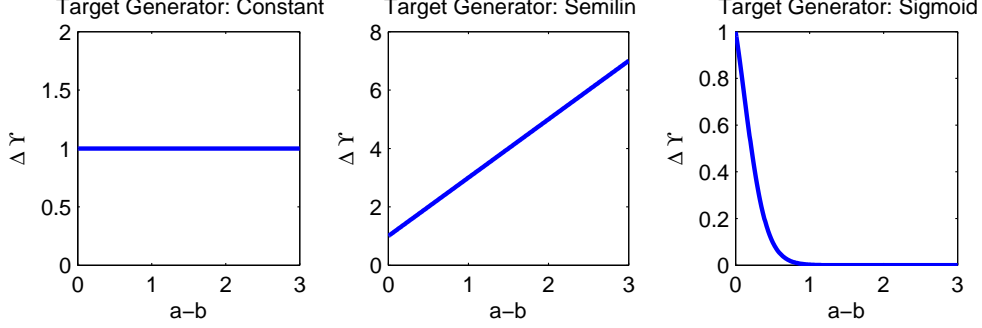


Figure 6.1: Comparison of three target generators. The y-axis ($\Delta\Upsilon$) refers to the difference in calculated distance targets for a and b , while the x-axis denotes the current difference of a and b . Only positive values for $(a-b)$ are denoted as Υ is only evaluated for violated constraints ($a - b > 0$).

6.3.3 Updating the Model

Given the current set of training constraints Q^* and the absolute training targets $\{(t_{(i,j)}, t_{(i,k)}) \mid (i, j, k) \in Q^*\}$, it is now possible to approximate \hat{dist}_P with constraints

$$\begin{aligned} \hat{dist}_P(d_{(i,j)}) &\leq t_{(i,j)} \quad \text{and} \\ \hat{dist}_P(d_{(i,k)}) &\geq t_{(i,k)}. \end{aligned} \quad (6.29)$$

Note that for standard regression, we use the strict equality relation $=$ for fitting a function to the constraints whilst for other optimisation methods \leq is more adequate. Given the facet distance vectors $D = \{(d_{(i,j)}, d_{(i,k)}) \mid \exists (i, j, k) \in Q^*\}$ and training targets $T = \{\Upsilon(\hat{dist}_P(d_{(i,j)}), \hat{dist}_P(d_{(i,k)})) \mid (d_{(i,j)}, d_{(i,k)}) \in D\}$, we will denote the learning of the parameters as a function

$$P^* = \text{TRAINING}(D_m, T_m). \quad (6.30)$$

Once the new model parameters $P^* \in \mathbb{P}$, are estimated using a regression or other metric learning function, they are integrated into the accumulated parameters $P_{m+1} \in \mathbb{P}$. Here, \mathbb{P} denotes a general parameter space. Therefore we define a parameter update function

$$\Lambda : \mathbb{P} \times \mathbb{P} \times \mathbb{N} \mapsto \mathbb{P}. \quad (6.31)$$

6.3 Adapting Methods to Relative Data

The parameters denoted above only contain the essential parameters necessary for an update function. Additional parameters may be added depending on the specifics of the function used. For the case of using standard regression as discussed above, we use the parameter update function

$$\Lambda_{lin}(W_m, W^*, m) = \frac{m * W_m + \eta * W^*}{m + 1}. \quad (6.32)$$

Here, $\eta \in \mathbb{R} \cap [0, 1]$ is a pre-set update factor determining the rate at which new learnt information is added to the existing parameters.

This idea is inspired by the definition of the recursion used by Zheng et al. [106] to approximate the desired function. Instead of updating the parameters directly, a recursion is used which includes all parametrised distance functions of the M training iterations. Let $(\hat{dist})^*$ be the newly learnt function for iteration m , then

$$(\hat{dist})_m(\cdot) := \frac{m * (\hat{dist})_{m-1}(\cdot) + \eta * (\hat{dist})^*(\cdot)}{m + 1}. \quad (6.33)$$

If we assume that the functions are linear regarding their parameters, thus if:

$$\begin{aligned} (\hat{dist})_a(\cdot) + (\hat{dist})_b(\cdot) &= \hat{dist}_{P_a}(\cdot) + \hat{dist}_{P_b}(\cdot) \\ &= \hat{dist}_{(P_a+P_b)}(\cdot) \end{aligned} \quad (6.34)$$

we find $(\hat{dist})_m(\cdot) = \hat{dist}_{\Lambda_{lin}(P_{m-1}, P^*)}(\cdot)$.

The general form of our modified algorithm for using absolute data training methods with relative similarity data is sketched in Algorithm 2.

6.3.4 Regression

We compared the approach used by Zheng et al. [106] to our linear regression approach for learning a weighted Euclidean metric $\hat{dist}_W(d_{(i,j)}) = \text{diag}(W)^\top (d_{(i,j)}^E)^2$ for diagonal $W \in \mathbb{R}^{N \times N}$. To this end we trained a function based on the standard facet distance vectors $d_{(i,j)}^E$ using as output targets such as gained the semilinear target generator $\Upsilon_{semilin}$ (see Equation (6.25)). In order to allow for a constant

Algorithm 2 Generic Relative Training via Absolute Constraints

Require: Constraints Q_{train} , features $x_{(i)} \forall i \in I$, initial P_0 , margin τ , enforcement factor ι , number of cycles k

$m = 0$

while $m \leq k \wedge Q^* \neq \emptyset$ **do**

$Q_m^* = \{(i, j, k) \in Q_{train} \mid dist_{P_m}(x_{(i)}, x_{(j)}) + \iota\tau > dist_{P_m}(x_{(i)}, x_{(k)})\}$ ▷ update violated constraints

$D_m = \{(d_{(i,j)}, d_{(i,k)}) \mid \exists(i, j, k) \in Q_m^*\}$ ▷ facet distance vectors

$T_m = \{\Upsilon(\hat{dist}_{P_m}(d_{(i,j)}), \hat{dist}_{P_m}(d_{(i,k)})) \mid \exists(d_{(i,j)}, d_{(i,k)}) \in D_m\}$ ▷ training targets

$P^* = \text{TRAINING}(D_m, T_m)$ ▷ get new parameter set

$P_{m+1} = \Lambda(P_m, P^*, m)$ ▷ update the model parameters

$m = m + 1$

end while

return P_m

term to be used within the similarity model, we extend $d_{(i,j)}^E$ by a constant entry of 1:

$$d_{(i,j)_k}^{EC} := \begin{cases} 1 & \text{for } k = 0 \\ d_{(i,j)_{k-1}}^E & \text{otherwise.} \end{cases} \quad (6.35)$$

The linear function

$$\hat{dist}_W(d_{(i,j)}^{EC}) = \sum \left(d_{(i,j)_k}^{EC} \right)^2 W_{k,k} \quad (6.36)$$

is then parametrised using the built-in Matlab regression function `regress`, and the resulting W^* is used to update the final model using the update function Λ_{lin} (see Equation (6.32)).

Similar to other methods for metric learning, it is possible to include the interaction of different features into the regression model. Instead of using $d_{(i,j)}^{EC}$ as input vector for the regression process, we define a matrix of all possible binary products of facet distances:

$$d_{(i,j)}^{EI} = d_{(i,j)}^E \left(d_{(i,j)}^E \right)^\top \in \mathbb{R}^{N+1 \times N+1} \quad (6.37)$$

Given corresponding indexes k, l into $d_{(i,j)}^{EI}$, and the resulting regression coefficients $W_{k,l}$ it is possible to adapt a squared Mahalanobis distance (see Equation (6.8)) using any regression method.

$$\begin{aligned}
 \hat{dist}_W(d_{(i,j)}^E) &= \sum_k \sum_l \left(d_{(i,j)k,l}^{EI} \right) W_{k,l} \\
 &= d_{(i,j)}^E \top W d_{(i,j)}^E \\
 &= dist_W(x_{(i)}, x_{(j)})^2
 \end{aligned} \tag{6.38}$$

Note that, unless explicitly enforced in the regression method, the $W_{k,l}$ may be negative and thus the resulting distance function may not be a metric.

6.3.5 Regression Trees

Originally, Zheng et al. [106] apply their framework via greedy function approximation using *Gradient Boosting Trees* as described by Friedman [26]. Applying their approach to the scenario of similarity data, we choose the standard facet distance vectors $d_{(i,j)}^E$ as input and use the Gradient Boosting Trees (GBTs) to approximate absolute similarity targets. In order to approximate $\hat{dist}_B(d_{(i,j)}^E)$, regression trees B_m are trained in an iterative process that assigns as target the negative gradient $r_{(i,j)_m}$ of iteration m (see Algorithm 3).

Regression trees $y = B(x)$ approximate functions by hierarchical partitioning of the input training data's space. The data ($x \in X_{train}$) is divided into regions R_j of minimal variance regarding the output data $y \in Y_{train}$. Given a new input, x , the corresponding leaf region R_j is determined and the representative data mean $B(x) = \frac{1}{|R_j|} \sum_{y_i \in R_j} y_i$ for that leaf is used as approximative output.

Using Gradient Boosting Trees for learning from relative ranking data, Zheng et al. [106] apply the target generator $\Upsilon_{semilin}$ (Equation (6.25)) and an update function similar to Λ_{lin} (Equation (6.32)). To this end they define a recursion of a series of functions $(\hat{dist}_B)_m$ over the iterations resulting in

$$(\hat{dist}_B)_m(d_{(i,j)}^E) = \frac{m * (\hat{dist}_B)_{m-1}(d_{(i,j)}^E) + \eta \cdot \hat{dist}_B^*(d_{(i,j)}^E)}{m + 1}, \tag{6.39}$$

for \hat{dist}_B^* being the new approximation of the current step's targets using GBT.

Algorithm 3 Gradient Boosting Trees (GBT)

Require: Facet distance vectors $D = d_{(i,j)}^E$, targets $T = \{t_{(i,j)} \text{ (e.g. } \Upsilon(\dots)) \mid \exists d_{(i,j)}^E \in D\}$, shrinkage factor γ , max. cycles M

Define $\hat{dist}_{B_0}(d.) = \frac{1}{|T|} \sum_{t_{(i,j)} \in T} t_{(i,j)}$ ▷ first tree outputs the average target

$m = 1$

while $m \leq M$ **do**

$r_{(i,j)_m} = t_{(i,j)} - B_{m-1}(d_{(i,j)}^E)$ ▷ negative gradient as new target

$B_m = \text{TreeFit}(D, \{r_{(i,j)_m} \mid \exists d_{(i,j)}^E \in D\})$ ▷ fit regression tree B_m to targets

$\hat{dist}_{B_m}(d_{(i,j)}^E) = \hat{dist}_{B_{m-1}}(d_{(i,j)}^E) + \gamma \cdot B_m(d_{(i,j)}^E)$

$m = m + 1$

end while

return $B = \{B_m \mid m \in \{0, \dots, M\}\}$

6.3.6 Information Theoretic Metric Learning

Davis et al. [20] describe Information-Theoretic Metric Learning (ITML) for learning a Mahalanobis distance from absolute constraints. Their approach uses Bregman optimisation for determining the Mahalanobis matrix.

A particularly interesting feature of ITML is that a template Mahalanobis matrix $W_0 \in \mathbb{R}^{n \times n}$ can be provided for regularisation towards a precomputed metric or distinct data distribution. If not specified otherwise the identity transform is used for W_0 . The regularisation then exploits the interpretation of Mahalanobis matrices as multivariate Gaussian distributions (see Equation (6.9) on page 121). The distance between two Mahalanobis distance functions parametrised by W and W_0 is measured by the relative entropy of the corresponding distributions:

$$\text{KL}(P(x_{(i)}; W_0) \parallel P(x_{(i)}; W)) = \int P(x_{(i)}; W_0) \log \frac{P(x_{(i)}; W_0)}{P(x_{(i)}; W)} dx_{(i)} \quad (6.40)$$

For feature vectors $x_{(i)} \in \mathbb{R}^n$. KL denotes the Kullback-Leibler divergence. In order to apply Bregman's method for optimisation, Davis et al. [20] describe this relative entropy using the LogDet divergence

$$\begin{aligned} D_{ld}(W, W_0) &= \text{tr}(WW_0^{-1}) - \log \det(WW_0^{-1}) - n \\ &= 2 * \text{KL}(P(x_{(i)}; W_0) \parallel P(x_{(i)}; W)). \end{aligned} \quad (6.41)$$

6.3 Adapting Methods to Relative Data

For details of the transformation see [20]. Given the sets R_s of similar and R_d of dissimilar clip indices, the optimisation problem is then posed as follows:

$$\begin{aligned}
\min_{W \succeq 0, \xi} \quad & \text{ITML}(W, \xi, \gamma, R_s, R_d) = D_{ld}(W, W_0) + \gamma D_{ld}(\text{diag}(\xi), \text{diag}(\xi_0)) \quad (6.42) \\
\text{s.t.} \quad & \text{tr}(W d_{(i,j)}^L (d_{(i,j)}^L)^\top) \leq \xi_{ij} \quad \forall (i, j) \in R_s \\
& \text{tr}(W d_{(i,j)}^L (d_{(i,j)}^L)^\top) \geq \xi_{ij} \quad \forall (i, j) \in R_d \\
\text{with} \quad & d_{(i,j)}^L = (x_{(i)} - x_{(j)}). \quad (6.43)
\end{aligned}$$

Here, ξ_{ij} correspond to slack variables allowing for and controlling the violation of individual constraints. The ξ_{ij} are initialised to given upper bounds u_{ij} , if $(i, j) \in R_s$ or lower bounds l_{ij} , if $(i, j) \in R_d$. During optimisation, they are regularised by comparison to the template slack ξ_0 using diagonal matrices $\text{diag}(\xi)$ and $\text{diag}(\xi_0)$.

6.3.6.1 Relative Learning with RITML

The method as published by Davis et al. [20] does not allow for training with relative similarity constraints. In the following we present an adaptation of the ITML algorithm using our modified relative learning strategy based on Zheng et al. [106]. To this end, we embed ITML into the relative learning framework as described in Algorithm 2. We call the resulting new algorithm Relative Information-Theoretic Metric Learning (RITML).

The standard ITML parameters such as γ , as well as the relative learning parameters including shrinkage factor η , margin τ , enforcement factor ι and number of cycles k are given at the beginning. It is also possible to provide an initial choice of W_0 . In most cases we use the identity matrix as default. We use the euclidean facet distance vectors $d_{(i,j)}^E$ for representing the clip pairs.

During iteration m , the active training set of violated constraints Q_m^* is calculated as in Equation (6.23). Afterwards, absolute training targets $\xi_m = T_m$ acquired via the semilinear target generator $\Upsilon_{\text{semilin}}$ are used to set the upper and lower bounds for ITML. The training data is divided into sets of similar and dissimilar constraints

S_m and D_m ,

$$R_{s_m} = \{(i, j) \mid (i, j, k) \in Q^*\} \quad (6.44)$$

$$R_{d_m} = \{(i, k) \mid (i, j, k) \in Q^*\} \quad (6.45)$$

Now, the current W^* can be calculated using

$$W^* = \text{ITML}(W_m, \xi_m, \gamma, S_m, D_m). \quad (6.46)$$

The final Mahalanobis matrix is then accumulated using the model update function Λ_{lin} as defined in Equation (6.32).

Algorithm 4 Relative Training with ITML

Require: Constraints Q_{train} , features $x_{(i)} \forall i \in I$, initial W_0 , regularisation factor γ , shrinkage factor η , margin τ , enforcement factor ι , number of cycles k

$W = W_0$ ▷ initialise variables

$m = 0$

while $m \leq k \wedge Q^* \neq \emptyset$ **do**

$Q_m^* = \{(i, j, k) \in Q_{train} \mid \text{dist}_{W_m}(x_{(i)}, x_{(j)}) + \iota\tau > \text{dist}_{W_m}(x_{(i)}, x_{(k)})\}$ ▷ update
violated constraints

for all $(i, j, k) \in Q_m^*$ **do**

$(\xi_{ij}, \xi_{ik}) = \Upsilon_{semilin}(\text{dist}_{W_m}(x_{(i)}, x_{(j)}), \text{dist}_{W_m}(x_{(i)}, x_{(k)}))$ ▷ upper / lower
bounds

end for

$R_{s_m} = \{(i, j) \mid (i, j, k) \in Q_m^*\}$ ▷ similar constraints

$R_{d_m} = \{(i, k) \mid (i, j, k) \in Q_m^*\}$ ▷ dissimilar constraints

$W^* = \underset{W}{\text{argmin}} \text{ITML}(W_m, \xi, \gamma, R_{s_m}, R_{d_m})$ ▷ update W via ITML

$W_{m+1} = \Lambda(W_m, W^*, m)$ ▷ update the model parameters

$m = m + 1$

end while

return Mahalanobis matrix W

A very nice property of RITML is that it enables *transfer learning*: If a specific starting value or template of W_0 other than the identity matrix is provided, the optimisation tends to produce results close to the provided W_0 . We call this “template-start and fine-tune” method W_0 -RITML. The performance of RITML and W_0 -RITML is evaluated in Section 8.7.

6.3.7 Relative Learning with Neural Networks (RDNNs)

This section shows an application of Neural Networks to learning similarity from relative data. Unlike models parametrising a Mahalanobis metric, Multi Layer Perceptrons (MLP) are capable of approximating arbitrary functions (see Hornik, Stinchcombe and White [38]). This means that more complex interactions of the features can be modelled.

As for general function approximators, training algorithms for MLP require absolute target data. One way of unlocking MLP for use with relative data would be to apply the general strategy as described above. Indeed, we have adapted a similar strategy presented by Hörnel [37] and based on earlier work by Braun, Feulner and Ullrich [15] which translates the idea of Section 6.3 into the language of Neural Networks.

As shown in Figure 6.2, our strategy is based on a combined network with two MLP networks, *net1* and *net2* that have the same structure and share their weights. This type of architecture is named *siamese* by Hadsell, Chopra and Lecun [31]. The input of each net is given by the facet distance vector $d_{(i,j)}^{|L|}$ representing a pair of clips. From a similarity constraint (i, j, k) , *net1* gets the vector $d_{(i,k)}^{|L|}$ of the less similar pair, and should thus output a larger distance value than *net2*, getting the more similar pair $d_{(i,j)}^{|L|}$. It is also possible to not use the absolute value facet distance vector $d_{(i,k)}^L$ which would allow for asymmetric similarity modelling. But in the experiments in Chapter 8 we chose $d_{(i,j)}^{|L|}$ for comparability, as results did not improve when using the asymmetric model. The outputs of *net1* and *net2* are connected to a comparator neuron c with negative fixed weight $-/ + v$ for *net1/net2* respectively. Thus c outputs a higher value if the correct input has not been achieved. The activation function of c is chosen to produce non-negative values, and the whole network can now be trained with target values of 0 for every training example.

Hörnel used a comparator neuron with sigmoid activation function similar to Equation (6.28), and a weight fixed with a negative sign for the 'left' network and a positive sign for the 'right' network. An alternative suggested by Braun [14] is the use of a semi-linear activation function f_c for the comparator neuron as indicated in

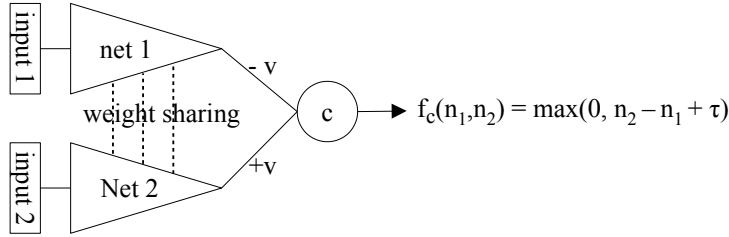


Figure 6.2: Scheme for RDNN neural network learning from relative data as suggested by Braun.

Figure 6.2. We also introduce a margin between the higher and the lower ratings with a variable τ .

In this work, we present our new implementation of this scheme using a single network: Relative Data Neural Net (RDNN). This is based on the observation that the derivatives of the sum-of-squares error ($SSE(P)$) on a set of inputs P with regards to the output $n_1^{(p)}$ and $n_2^{(p)}$ of *net1* and *net2* for input p are

$$\frac{\partial SSE(P)}{\partial n_1^{(p)}} = v \cdot (n_2^{(p)} - n_1^{(p)} + \tau) \text{ and } \frac{\partial SSE(P)}{\partial n_2^{(p)}} = v \cdot (n_1^{(p)} - n_2^{(p)} + \tau). \quad (6.47)$$

The relatedness of the output values allows us to integrate the above approach into our relative training framework as described in Section 6.3. For this, we use a single network with resilient backpropagation (cf. Riedmiller and Braun [74]) including a regularisation term. We also choose $\tau = 0.5$, $\iota = 2$ (see Equation (6.23)) and Υ_{const} as the most successful target generator in the final implementation. The resulting MLP calculates a distance measure between two clips C_i, C_j , given the vector $d_{(i,j)}^{|L|}$ of absolute differences of the two clips' features. The procedure is described in Algorithm 5.

6.4 Conclusions

In this chapter we presented new (RITML, RDNN, WMLR) and existing state-of-the-art methods for modelling music similarity from relative constraints. The methods

Algorithm 5 RDNN Training

Require: Constraints Q_{train} , features $x_{(i)} \forall i \in I$, # of cycles k

Define $D := \{ (d_{(i,j)}^{|L|}, d_{(i,k)}^{|L|}) \mid \exists (i, j, k) \in Q^* \}$ ▷ training data

Define $T := \{ (t_{(i,j)}, t_{(i,k)}) \mid \exists (i, j, k) \in Q^* \}$ ▷ training targets

MLP = initRandomMLP() ▷ initialise MLP with random weights

$m = 0$

while $m \leq k \wedge Q^* \neq \emptyset$ **do**

$Q_m^* = \{ (i, j, k) \in Q_{train} \mid d_{MLP}(x_{(i)}, x_{(j)}) + 2\tau > d_{MLP}(x_{(i)}, x_{(k)}) \}$ ▷ update train set

for all $(i, j, k) \in Q^*$ **do**

$(t_{(i,j)}, t_{(i,k)}) = \Upsilon_{const}(dist_{MLP}(x_{(i)}, x_{(j)}), dist_{MLP}(x_{(i)}, x_{(k)}))$

end for

MLP $_m$ = trainRp(MLP $_{m-1}$, Q_m^* , D , T) ▷ Train MLP with new targets

$m = m + 1$

end while

can be organised by the architecture of the underlying models. Categories include metric-based models such as the Mahalanobis distance learnt by MLR, SVM and RITML, neural networks (RDNN) and models based on general regression as described in Section 6.3. A model’s structure determines constraints that can limit or guide the learning process. For example, metric-based similarity models assume symmetry in the similarity relation. This limits the existing value ranges of parameters, and improves the robustness of the resulting model. Our new methods introduce novel properties into music similarity learning from relative data: W_0 -RITML enables transfer learning using pre-trained models, and RDNN can model asymmetric similarity relations.

We integrate and modify the general regression framework from [106] for use with relative music similarity data. This enables the adaptation of further existing metric learning algorithms – such that where previously only accessible for absolute similarity constraints – to relative data. This formulates similarity learning from relative data as a generic regression task. Finally, an effort is made to integrate the methods available and relevant for modelling relative similarity into a general conceptual framework, from facet distance vectors representing clips’ differences

with regard to specialised metrics, along the model definition defining the parameters of our new model to the model training itself.

Our structured discussion of methods for similarity modelling provides an overview of existing and newly developed solutions to research question *rq:1*. This structure is reflected by the CAMIR programming framework for reproducible experiments with models of relative similarity data, discussed in the following Chapter 7. Using this framework, an evaluation of the aforementioned methods will be provided in Chapter 8.

I was working with tape loops, sort of primitive technology. This was in the late 50's early 60's. [...] fact technology wasn't very good no matter how much money you had.

(Terry Riley, 1992)

7 A Framework for Reproducible Training and Evaluation of Music Similarity Models

This chapter discusses the CAMIR framework which was developed as a tool for the analysis of music similarity datasets and training of similarity models. One aim in developing the framework was reproducibility of research, and it won the SoundSoftware prize in 2014¹. The great part of code for CAMIR has been published online². Licensing code as open source should be the preferred method, particularly for public research institutions. It facilitates direct extension on and adaptation of existing research, as well as it encourages collaborations as it has during the research for this thesis. To this end, the Subversion source code management software for associating code state to experimental results is strongly integrated into CAMIR.

CAMIR is primarily written in the Matlab scripting language, but it includes wrappers and components written in python or included as compiled code. Functionalities include the playback and systematic exploration of music audio by genre or tags from Last.fm. For audio features, it enables the visualisation and extraction both from audio and The Echo Nest API as discussed in Chapter 5. The functions for relative similarity data include an extensive set of graph theory functions for

¹<http://soundsoftware.ac.uk/rr-prize-aes53-winners>

²<http://mi.soi.city.ac.uk/datasets/camirframework/>

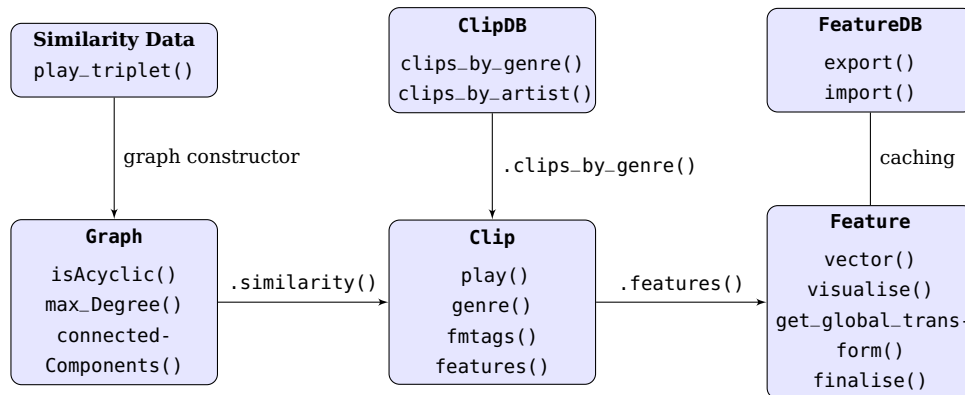


Figure 7.1: The database components of the CAMIR framework.

the analysis of coverage of clips, connectedness of similarity data and visualisation of the data structure as proposed in Chapter 3.

CAMIR currently can be used with the MagnaTagATune, Million Song Dataset Subset and CASimIR data sets. For the CASimIR framework, in addition to access to a fixed dataset snapshot, code is available to import, filter and analyse the similarity annotations collected from the MySQL (MySQL) database.

The framework provides a unified interface to the algorithms for similarity learning discussed in Chapter 6. This includes the wrapping of third-party implementations such as MLR, modifications of those such as WMLR, as well as the implementations of new algorithms such as the regression framework for similarity learning via regression from relative constraints (Section 6.3.4)

Finally, the framework allows for running similarity modelling experiments with the above methods using cross-validation. Results of such experiments are archived in a unified manner and strong emphasis is given to the reproducibility of experiments (see research question *rq:3*) by including parameter and context data such as revision identifiers for the producing code.

The typical workflow for an experiment in CAMIR is as follows:

1. Create test script with parameter combinations for grid search
2. Script runs `test_generic_features_parameters_crossval()`
3. Results are saved in version-annotated folder

7.1 Music Dataset Exploration

Clip Member Function	Purpose
<code>play()</code> , <code>skip()</code>	playback clip audio, either local or via stream
<code>title()</code> , <code>album()</code> , <code>artist()</code> , <code>artist_id()</code> , <code>album_id()</code> , <code>isrc()</code>	retrieve core metadata
<code>mbtags()</code> , <code>fmtags()</code> , <code>tags()</code> , <code>mbtag_ids()</code> , <code>fmtag_ids()</code> , <code>tag_ids()</code>	retrieve MusicBrainz, Last.fm or standard tags
<code>features(<i>type</i>)</code>	calculate features

Table 7.1: Selection of core functionalities of the `Clip`, `MTTclip`, `MSDclip` and `CASIMIRclip` classes.

4. Run `test_generic_display_results()` for results and graphical display

7.1 Music Dataset Exploration

With music as the primary media type for analysis, CAMIR provides specific facilities for exploring the datasets. Most of these functions can be accessed for individual clips:

In order to select a clip, a `Clip` object is instantiated either using a direct identifier with the `Clip` constructor or via the functions `clips_by_artist_name(artist)` or `clips_by_genre_name(genre)` of the corresponding `ClipDB` database.

The direct identifiers change across datasets: For the Million Song Dataset, clip identifiers have been copied from the original dataset, whereas the Million Song Dataset clips are referred to by their Seven Digital (7digital) reference. The CASimIR dataset, containing a subset of the above, identifies clips using the existing references. To be identified across datasets, all clips are associated with an ISRC code obtained via APIs such as the ROVI API¹. The data necessary for these operations is loaded into memory during startup by `startup_music_research.m` for fast access.

¹<http://developer.rovicorp.com/docs>

| clip id 1 | clip 2 | clip 3 | votes 1 | votes 2 | votes 3 | ... |

Table 7.2: Similarity as stored in the comparison data variables, clip ids are relative to `comparison_ids`. Votes count the frequency of clip x being the “Odd Song Out”. Rows can contain further vote information to the right.

Class Name	Nodes	Edges (Captured Relation)
<code>ClipComparedGraph</code>	single clips	clips appear in same triplet
<code>ClipSimGraphMulti</code>	clip pairs (C_i, C_j)	sim. constraints $(C_i, C_j) \stackrel{y}{>} (C_k, C_l)$, integer weights of constraints $\alpha_{(i,j,k)}$
<code>ClipSimGraphMD</code>	clip pairs (C_i, C_j)	similarity constraints with balanced relative weights per clip pair

Table 7.3: Front-end graph subclasses in CAMIR.

7.2 Similarity Data Processing and Analysis

This section explains how relative similarity data (see Chapter 3) is represented in CAMIR as well as methods for processing it. The similarity data contained in the MagnaTagATune dataset, and as published with a creative commons license from the CASimIR dataset (see [pub:7] and <http://mi.soi.city.ac.uk/datasets/aes2013casimir/> for fixed data) follows the schema depicted in Table 7.2.

The identifiers used in the table are remapped to a consecutive index via the mapping stored in `comparison_ids`. For MagnaTagATune, this information is loaded into global variables during startup, whereas a snapshot of the CASimIR data is stored in `casimirdb_15112013_public.mat`.

In order to analyse the characteristics of similarity data, it is converted into a graph structure, as this lends itself to represent similarity as a relationship of clips. Here, different types of analysis are provided by several graph structures. All classes are inherited from the basic `Graph` and `DiGraph` classes. This allows to use algorithms from classical graph theory on the similarity graphs, as well as a structured interaction and conversion between different representations of the same similarity data. All graph classes can be constructed by either providing raw similarity data as in Table 7.2 or graph instances derived from it.

7.2.1 Connectedness of Single Clips

The `ClipComparedGraph` represents clips C_i as vertices. An edge in this graph corresponds to the two connected clips occurring in the same similarity triplet (i, j, k) presented to users: $E(m, n) \Leftrightarrow \exists(i, j, k) \in \hat{Q} : |\{i, j, k\} \cap \{m, n\}| = 2$. Examples of `ClipComparedGraphs` are given in Figures 10.1 and 10.2.

7.2.2 Similarity Graphs on Clip Pairs

The methods for analysis of relative similarity data as described in Section 3.1 are implemented in `ClipSimGraphMulti`. This class implements a `MultiGraph` using a weighted directed graph with integer edge weights $\alpha_{(i,j,k)} \in \mathbb{N}$, counting the number of (remaining) votes supporting the specific constraint. As described in Section 3.1.2, `ClipSimGraphMulti` deals with inconsistent data by subtracting the weights of edges with contradictory similarity information, leaving the constraint with greater frequency in the dataset.

7.2.2.1 Alternative Edge Weightings

An alternative method for representing inconsistent similarity data is implemented in `ClipSimGraphMD`. Here, the weight of an edge between (C_i, C_j) and (C_i, C_k) , $\beta_{i,j,k}$ is calculated using

$$\beta_{i,j,k} = \frac{\max(0, \alpha_{(i,j,k)}) - \max(0, \alpha_{(i,k,j)})}{A} \in \mathbb{Q}, \text{ and} \quad (7.1)$$

$$A = \alpha_{(i,j,k)} + \alpha_{(i,k,j)} + \alpha_{(j,k,i)}$$

where $\alpha_{(i,j,k)}$ counts the number of votes behind the particular constraint without removal of contradictions. The scalar A provides a measure of the total unique votes present for the presented triplet. This alternative representation allows for the constraints to be weighted in relation to the total votes given for this triplet. The current implementation was only used with the `MagnaTagATune` dataset, but

when replacing A by the set of all similarity constraints containing the clip pair (C_i, C_j) .

$$A = \frac{1}{2} \sum_{\{(m,n,u) \mid |\{m,n,u\} \cap \{i,j\}| = 2\}} \alpha_{(m,n,u)} \quad (7.2)$$

7.2.3 General Graph Functions and Connectedness

The Graph and DiGraph classes then provide standard graph functionalities as well as analysis functions for finding strongly connected components (SCC) and visualisation via GraphViz¹. Conversion between Graph and DiGraph is possible by simply using the constructors of one class with the other class as input. This is for example used when analysing the connected components in similarity graphs as described in Section 3.1.3. Retrieving the connected components on the undirected graph mapping of ClipComparedGraph reveals the size of the longest transitive chain of clips compared to each other. Note that the availability of similarity data between the clips does not necessarily follow as inconsistent constraints might have cancelled out themselves in ClipSimGraphMulti.

As the ClipSimGraph classes operate on pairs of clips, an intermediate class ClipPairGraph mapping the basic graph functions to pairs of clips.

Given the clips and similarity data, we now need to represent the musical and cultural information about the clips in a machine-readable form via features.

7.3 Feature Extraction

The CAMIR system is currently built to extract audio features on the basis of existing features from The Echo Nest as included in the MagnaTagATune dataset and Million Song Dataset. Both datasets provide this “raw” feature data in their own file formats (XML / HDF5). Further tag features from third parties, such as from last.fm can be accessed through the extractors as well.

¹<http://www.graphviz.org/>

A variety of different features containing audio and / or tag information has been implemented into this framework. Both the signal flow and class structure for feature extraction follow a hierarchical structure: Especially for the audio features, the typical signal flow consists of a sequence of feature extractors. Where multiple features are extracted from the same source, such as audio features, it is common for more advanced features to share the same basic features and partial signal paths in their extraction. Therefore, each derived feature inherits the parameter options from the underlying feature types, calculates the underlying features and uses them for the extraction of potentially different feature types.

The in-memory caching of features (using several instances of `MTTAudioFeatureDBgen`) allows for an efficient reuse of shared basic feature types. Furthermore it is possible to save extracted features to disk, including their extraction parameters and Subversion code revision number for later reproducibility of the research.

7.3.1 Feature Extraction Flow

Although the basic feature information concerning a clip can be gained separately for each clip, processes such as normalisation or PCA transformations require the involvement of information from several clips. Addressing this, a three-stage process was implemented which enables a consistent structure for feature extraction:

1. `clip.extract()`: Calculate basic features separately on single clips.
2. `features.define_global_transform()`: Calculate transforms based on a given set of features representing certain clips. For example, define normalisation parameters or PCA transformation on a training set.
3. `features.finalise()`: Apply the global transformations on a given set of features. The set may be different from the above.

Transformations requiring knowledge from multiple clips might be involved on several levels of the hierarchical feature extraction process. As the `define_global_transform()` function is processed in reverse hierarchical order for all levels, but only once per extraction, some features are de facto only computed during the `features.define_global_transform()` and `finalise()` stages.

Once finalised extracted features can be saved to disk for later use including their parameters in XML format and code versioning information using `features.saveto()` for single-clip feature vectors or `MTTAudioFeatureDBgen.export()` for all currently extracted features.

7.3.2 Audio Features

All audio features extracted by CASimIR are based on the classes `MTTAudioFeaturesRaw` and `MSDAudioFeaturesRaw`, which load the existing feature information from the specific datasets' files. These raw features from The Echo Nest are aggregated in the mid-level feature `MSDAudioFeatureBasicSM`. Parameters allow to in- or exclude any component, as well as the option to include multiple clusters for aggregated features such as chroma or timbre features (see Section 5.1.1.2 or [pub:6]). Furthermore, other derived features from The Echo Nest as described in Table 5.1 (page 106) are included via `MTTAudioFeatureSlaney08`. The features mostly used for experiments with MagnaTagATune in Chapter 8 are `MTTAudioFeatureSlaney08`, `MTTMixedFeatureSlaney08GenreBasicSm` and `MTTMixedFeatureSlaney08GenreBasicSmPCA`.

The latter `MTTMixed` features combine audio features with genre tag features, which are extracted as below:

7.3.3 Tag Data

Currently, two features based on (textual) tag annotations are implemented: `MTTTagFeatureGenreBasic` with genre information from the Magnatune label and `MTTTagFeatureLastFMBasic` representing tags gathered from Last.fm. Parameters include a maximum percentile of the most common tags to include. This allows to reduce the feature dimension by abandoning very sparsely annotated tags. CAMIR includes basic scripts to acquire tags and metadata from Last.fm, The Echo Nest and similar online Web APIs.

7.3.4 Dataset Specifics

Most features with names starting with MTT can be also used with the Million Song Dataset as the features share a common data structure and signal processing flow. The most comparable features being available for both the MagnaTagATune and MagnaTagATune and therefore the CASimIR dataset, are the MTTMixedFeature-Slaney08LastFMBasicSm features, as for the Million Song Dataset, no genre tags but only Last.fm tags are available. Still, the number of Last.fm annotations varies significantly between these datasets, as the Million Song Dataset contains particularly popular music clips, which should be associated to more annotations than some clips from the MagnaTagATune.

7.3.5 Similarity Models and Training Algorithms

The models and training algorithms described in Chapter 6 have been integrated into the CAMIR framework.

All model training algorithms comply to a generic input / output interface for providing the training similarity constraints, feature data and parameter information. This is ensured by providing a `_wrapper` interface to each method, which, given constraints, features, and parameters, output the learnt similarity measure `A` and an information structure `diag`. Third party training frameworks have been encapsulated in these wrappers. The following training algorithms are currently available for training similarity models:

- `mlr_wrapper`: Metric Learning To Rank (Section 6.2.2) and Weighted Metric Learning To Rank (Section 6.2.3) (selection via parameters `weighted` and `diagonal`)
- `svmlight_wrapper`: Support Vector Machines (Section 6.2.1)
- `relnn_wrapper`: Relative Learning with Neural Networks (Section 6.3.7)
- `itml_relative_wrapper`: Relative Information-Theoretic Metric Learning (Section 6.3.6.1)
- `regression_wrapper`: Relative learning with standard regression (Section 6.3.4)

Further non-adaptive similarity measures have been implemented for use as baseline and for general testing:

- `euclidean_wrapper`: Unweighted Euclidean metric
- `mahalmat_wrapper`: Pre-defined Mahalanobis distance measure via template matrix
- `random_diag_wrapper`: Random diagonal Mahalanobis matrix

The resulting similarity measure A can be of two different types. The output type is set by the wrapper function via the output value `diag.interpreter` and determines how the similarity model data is to be interpreted during post-training evaluation.

The standard type is `DistMeasureMahal`. Here, A is provided as Mahalanobis matrix and can be evaluated following Equation (6.8). In the case of using Neural Nets or Regression Trees for modelling similarity, the type `DistMeasureGeneric` indicates that the output similarity model A is an object providing the function `A.evaluate(Ci, Cj)`, which allows for determining the similarity of two clips (C_i, C_j) with the provided model.

The wrapper functions for training the similarity models also choose the facet distance function $d_{(i,j)}$ (see Section 6.1) with respect to the parameter `deltafun`. The following facet distance functions have been implemented. Facet distance functions can be selected by the parameter `Note` that usually the training method can only accept certain facet distance measures, as discussed in Chapter 6.

- `squared_dist_delta`: $d_{(i,j)}^E$ (eq. (6.2))
- `simple_minus_delta`: $d_{(i,j)}^L$ (eq. (6.4))
- `abs_delta`: $d_{(i,j)}^{|L|}$ (eq. (6.4))
- `abs_delta_plusconst`: $\left| \sqrt{d_{(i,j)}^{EC}} \right|$ (eq. (6.35))
- `conv_subspace_delta`: Subspace facet distance values (eq. (6.6))

7.4 Experiment Scripts and Result Management

When creating code for research purposes, provisioning for reproducibility of experiments and their results is critical for comparison and validation. For research code this can be particularly challenging as there is a great amount of change in the code and parametrisations to be expected between experiments. Furthermore there is a meta-level such as found in grid search where a series of parameters is tested and results are compared across parameters with regard to the algorithm's performance.

In CAMIR, tracking of parameters and code is enabled through the use of experiment scripts defining the complete set-up and the integration of Subversion source code control. Results and extracted features are saved in a date and version annotated folder which contains all necessary information to rerun the experiment.

The experiment workflow (see Figure 7.2) is divided into three components: definition of the experiment, execution of the experiment and analysis of the results. The test scripts allow for the definition of parameter ranges for features, the model, training methods and evaluation via cross-validation. There is also an option of analysing series of experiments over growing subsets of training data (see Chapter 8). Furthermore, the similarity dataset to be used will be defined here.

The `test_generic_parameters_crossval()` routine for running experiments is designed for unsupervised execution over potentially long experiments runs. Even if errors occur with specific parameter configurations, the framework will continue with other remaining configurations. Once the feature data has been calculated and stored in the results folder, rerunning experiments and adding training and evaluation parameters is automatically performed upon the saved features unless feature parameters are changed.

Several generic visualisations of experiment results are provided by the method `test_generic_display_results()` which takes as argument the results folder location. This includes bar-diagram comparison of training and test set results of different configurations as well as an analysis of the influence of different parameter

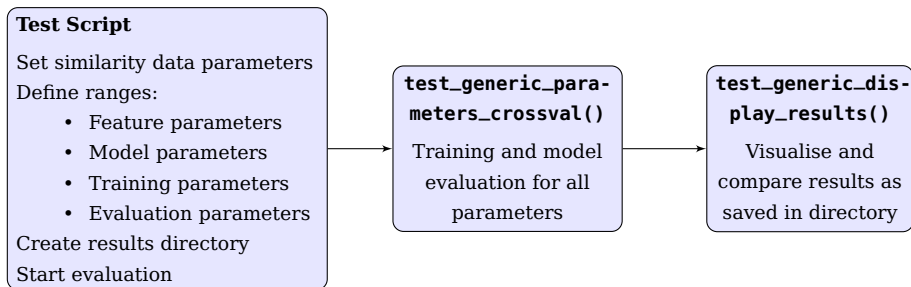


Figure 7.2: Experiment workflow in the CAMIR framework.

types on the models' performance. The best and worst performing parameter values are also reported for each parameter. The saved result data contains various other diagnostic information including the model training time per parameter set, information on training success and possible errors during training.

7.5 Conclusions

We have presented a software framework allowing for the analysis of relative similarity data, feature extraction and reproducible experiments on music similarity model training and evaluation. The included work flow for explicit experiment set-up and parametrisation, together with strong integration of source code control into both code and saved results, encourage reproducibility of any experiments performed with the toolbox, as desired by research question *rq:3* for the following evaluation.

The framework is available as open source on the web, and includes all similarity learning methods described in Chapter 6. It combines new methods for similarity modelling with several third party methods for metric learning, their adaptations and other external tools e.g. for graph visualisation. CAMIR is built around many components that have been published as open source by other researchers and developers. We encourage users of the CAMIR framework to make their contributions open source, and where possible integrate their additions into the existing framework.

7.5 Conclusions

The following section will discuss our similarity modelling experiments for evaluation of the methods described in Chapter 6. Except for special methods in Section 8.4.3, all results reported in this thesis have been produced with CAMIR.

Unfortunately, human subjects cannot be regimented as easily as cards of a deck, and the investigator of human behaviour faces sampling problems which are not sufficiently allowed for by pencil and paper statisticians.

(Alfred Kinsey, 1948)

8 Evaluation

With the similarity learning and evaluation framework introduced above, we now [pub:7] go on to evaluate the effectiveness of similarity modelling approaches given real user-generated similarity data. We evaluate training algorithms as introduced in Chapter 6 in their general ability to fit the similarity data in Section 8.3, allowing for insights and answers to our primary research question *rq:1*. For most of these algorithms, this is their first thorough evaluation for adapting to user-based similarity data. Here, especially the test-set or generalisation performance, describing how well previously learnt models generalise to unknown data will be in focus. In particular, we measure the percentage of test-set constraints that are fulfilled by the learnt similarity models. The trade-off between fitting training data and generalisation is observed for different algorithms, with their hyper-parameters tuned for the best generalisation performance. As we compare different metric-based models and our new neural net-based model, we are interested in which models will be most suitable for this task. Also, the time needed for training models will be compared between methods.

Second to the model structure itself, the representation of clips and their musical content via features is a central part in modelling music similarity. If not specified otherwise, our experiments use features derived from the The Echo Nest data contained in the MagnaTagATune dataset and Million Song Dataset, as introduced in Chapter 5. These combined features include acoustic and tag information. In Section 8.4, they will be compared with model training tests on single feature types

as well as on different combinations of those. We analyse the influence of both the representation as well as the contained information regarding the modelling success. One comparison method we introduce, using PCA for reduction to equal dimensionality, allows for the analysis the impact of different feature information types while keeping the feature vector dimensionality constant.

In Section 6.2.3, we introduced the WMLR method for learning from weighted similarity data, which promises to more accurately represent similarity data during training. A comparative analysis using weighted SVM training is performed and allows for assessment of the quality of weightings derived from the MagnaTagATune dataset and thereby respond to *rq:4*.

Our new, and currently growing CASimIR dataset is the first to contain participant attributes linked to the similarity statements. Section 8.7 introduced our novel country-annotated similarity dataset, which we will capitalise on for first culture-aware similarity experiments in Section 8.7. The new RITML method presented in Section 6.3.6.1 is now evaluated on the country-specific datasets, and we present a first application of its template-based learning facility for a first study on transfer-learning with similarity models in Section 8.7. This will give some interesting insights on the information that can be stored in similarity models, and how it can be exploited for research in comparative musicology.

With the exception of the latter experiments, all following analysis is performed on MagnaTagATune, as this was the only relative similarity dataset available throughout the period underlying this research. Furthermore, the CASimIR dataset is still growing and at this point not as large as MagnaTagATune.

In order to provide a fair comparison and analysis of the different model training methods, we take a closer look on the influence of sampling strategies for relative similarity data on our evaluation of model performance in Section 8.5. For now, Section 8.1 introduces our new generic graph-based Inductive Sampling (ID-sampling) method for dividing similarity data into disjunct training-and test-sets to account for unwanted effects of transfer learning in some algorithms.

8.1 Strategies for Cross-Validation

In our experiments, the performance of the learnt metrics regarding the similarity data is evaluated using cross-validation. Cross-validation uses on disjunct sets of training and test constraints. In k -fold cross-validation, the complete constraint set is divided into k disjunct bins of approximately equal size. Afterwards, one of the bins is left out during training and used for testing the performance. Repeating this procedure for each of the bins, k test results are calculated. In our experiments, we observe the average satisfaction of constraints over all k bins, as well as the corresponding standard deviation.

In our initial experiments we noted that the division of similarity constraints into test and training sets affected the outcome for certain algorithms. Considering that multiple constraints refer to the same clip or clip pair (max. 2 constraints per clip in the MagnaTagATune dataset), random sampling across the constraints leads to clips being both referred to by constraints in the test and training sets. As the features for the clips are part of the training data given to the model training methods, feature information for certain test set constraints may already occur in the training set. Algorithms may now take advantage of the knowledge of the feature space for the otherwise unknown similarity constraints in the test set. We refer to this situation as *Transductive Training* (see Gammerman, V. and Vapnik [28] for the terminology).

In order to avoid bias through transduction of information from training to test set, we here present a new method for inductive training with relative similarity data, intelligently splitting training and test set on the clip and constraint level, which will be used for the majority of our experiments.

8.1.1 Sampling for Transductive Training

As will be assessed in Section 8.5, only separating constraints, ignoring the multiple notion of clips, as in most of the experiments on the MagnaTagATune similarity data [pub:4, pub:6] and [91], leads to transductive training effects due to reappearing feature values: Clips may reappear in up to three constraints, and the

feature vectors used in the training of a metric are likely to also be included in the constraint-disjunct test sets leading to over-estimation of the results in cross-validation.

Especially MLR, but also SVM-based approaches can utilise such a priori knowledge of the test set’s feature space when determining the relevant constraints (MLR) or choosing support vectors (SVM). This transductive sampling will be referred to as Transductive Sampling (TD-sampling). We test the effects of TD-sampling in Section 8.5, where the similarity constraints Q are randomly sampled into $k = 10$ cross-validation bins. Each of these bins is used as the test set Q_{test}^k of 86 constraints, while the remaining 9 subsets are combined to the training set Q_{train}^k of 774 constraints. Using this configuration, the training sets reference 989 clips. Thus, on average, we find that 98% of the test set clips already appear in the training set. Depending on the application of the learnt similarity measure, e.g. for recommendation within a database with fixed known tracks, this approach can be realistic and helpful: Although only some feature information, and no similarity data is shared between training and test sets, the performance on unknown similarity data is improved.

8.1.2 Sampling for Inductive Training

On the other hand, for assessing the capabilities of a model to generalise over unknown test data, further means have to be applied. Wolff et al. in [pub:9], test a cross-validation sampling method which separates the MagnaTagATune similarity data on the basis of the connected similarity subgraphs. Here, the k bins used for cross-validation are not defined on the basis of single constraints $(i, j, j) \in Q$, but on the subgraphs G_{sub}^i . Choosing disjunct bins on the basis of these 337 connected components of the similarity graph guarantees the bins to be disjunct with regard to the set of clips as well. Such a sampling based on the G_{sub}^i comes with several advantages when compared to a sampling based on clips: Due to the structure of the underlying similarity triplets, all clips in a subgraph G_{sub}^i have to exist in a testing bin in order to represent the distance constraints involving one of the clips in this graph. Now, any other distance constraint existent in this subgraph

has either to be included in the same cross-validation bin or must be omitted, as its similarity information exclusively relates to the two remaining clips mentioned in its subgraph. The G_{sub}^i differ in their number of edges or distance constraints included, which results in the cross-validation bins slightly varying in their size in constraints. With the exception of Section 8.5, in all of the following experiments 337 subgraphs have been divided into $k = 10$ bins, each corresponding to 33 or 34 subgraphs. This results in bins containing 85 constraints on average. The maximal training set size varies from 771 to 779 constraints referencing on average 896 clips. This sampling strategy for inductive training will be referred to as ID-sampling.

8.2 Growing Subsets

Some of the figures in this chapter show the dynamics of model training over a growing number of training constraints. For this, similarity constraints are extracted as described in Section 8.1. Then, subsets $Q_{\text{train}(p)}^v$ of the training sets are defined, which are used individually for training: For each cross-validation fold, the training set size $|Q_{\text{train}(p)}^v|$ is increased in a manner of cascading subsets starting from only a small set of 13 constraints on average. The performance is then tested against a logarithmically increasing test set size p until the complete test set is reached. The successive $Q_{\text{train}(p)}^v$ for growing p are selected in a manner to assure that the smaller sets are subsets of larger training sets:

$$Q_{\text{train}(p_1)}^v \subset Q_{\text{train}(p_2)}^v \text{ for } p_1 < p_2 \quad (8.1)$$

Still, the selection of the constraints into the cross-validation bins and the early selection of only a small subset introduces a bias into the test results. To control this effect, we use four independently selected samplings (type: ID) for each data point plotted, across which the results are averaged.

8.3 General Performance

For the metric learning approaches, DMLR and SVM are compared in learning a weighted Euclidean distance, whilst MLR is adapting a Mahalanobis distance with a full matrix W . The regularisation trade-off factors c were set to $c = 10^{12}$ for MLR, $c = 10^2$ for the diagonally restricted DMLR (Section 6.2.2), and $c = 3$ for the SVM algorithm (Section 6.2.1). They have been determined using a grid-based search on training data for the optimal configuration, the algorithms' performance being evaluated via cross-validation. For RDNN the MLP is set up with two hidden layers, containing 20 and 5 neurons, respectively. Training of the MLP is performed in 5 RPROP epochs, growing to 14 epochs along 4 outer cycles.

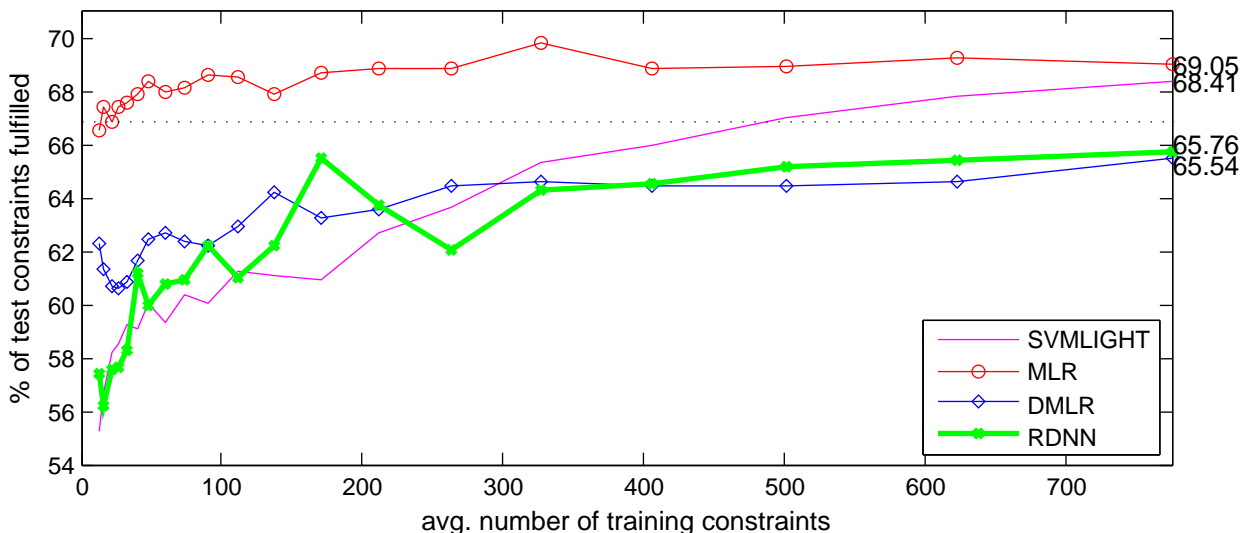


Figure 8.1: Overall test set performance for combined features with averaged low-level information: SVM (SVM-Light), MLR, DMLR and RDNN performance for full features, with increasing training set size. The baseline (un-weighted Euclidean distance) is plotted as dots.

In Figure 8.1, the different algorithms are compared using the combined features containing averages for audio and timbre features, Slaney08 features and genre features. This combination was chosen for showing relatively good results for all of the algorithms. Considering the training with the maximum size training sets, both MLR and SVM achieve similar performance on the unknown test set. DMLR

8.3 General Performance

is not able to generalise well from the training set (see Figure 8.2) onto the test set.

In this experiment the test for the largest subsets results by MLR and SVM are approximately 2% and 1.5% above the Euclidean baseline of 66.86%. At 5% significance level only the MLR results are significantly better than the Euclidean metric ($p = 0.0007$). Both DMLR and RDNN remain below the baseline performance by 1% on the test sets. The effect of these learning algorithms will become clearer when features whose Euclidean distance provides less competitive results are used, and will be explored in Section 8.4.1 below.

The test set results for few training data $Q_{\text{train}(p)}^v$ highly depend on the algorithm used, and for SVM and DMLR lie considerably below the baseline. For SVM, this is an effect of overfitting and regularisation parameters being optimised for the larger training sets. In [pub:9], an adaptive regularisation is suggested which could trade some of the training set results, being superior for small training sets (Figure 8.2) for better generalisation. This may be achieved by using stronger regularisation for smaller training sets.

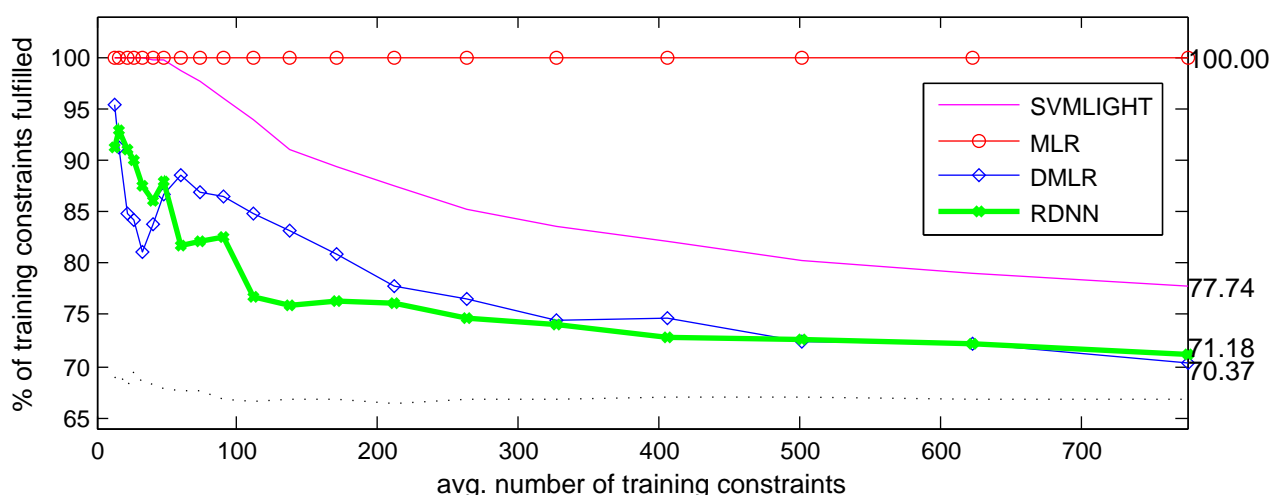


Figure 8.2: Overall training set performance: SVM, MLR, DMLR and RDNN performance for full features, with increasing training set size.

Considering the behaviour of the generalisation with regard to growing training

sets, Figure 8.1 already shows a substantial difference between MLR and SVM: While MLR does not significantly fall below the baseline, for the small training sets, SVM starts with very low generalisation. MLR then almost reaches maximum performance within the first 100 test examples, SVM only reaches the baseline performance at 500 training constraints. Both DMLR and the MLP network miss the baseline performance by 1% for the test sets, although their training performance is considerably better.

The training set performance curves in Figure 8.2 exhibit several particular types of learning behaviour. Note that the baseline plotted as a dotted line below 70% slightly varies while the training subsets grow to their full size. This is because only a subset of the data is used for training over all bins, growing with the training size. For each of the four samplings, the baseline can vary in the scale of 10% depending on the subsets used for training. In accordance with results published earlier ([pub:5, pub:6], MLR is able to fulfil all of the training constraints provided. The training performance of SVM shows a continuous regularisation trade-off allowing for additional constraints to be learned whilst preserving good generalisation at the final full training set size. The worst performance is shared by DMLR and the MLP network, overfitting to the training examples for small training sets with a consistently inferior performance when compared to SVM and MLR. For these algorithms, little gain is achieved with regard to the baseline and even the results of 65% on unknown test sets. Note that the MLP net results depicted here are early ones. Improvement would be expected with larger net sizes at least for the training performance of MLP.

8.3.1 Training Set Size

In Figure 8.1, the generalisation success of the algorithms is plotted over the number of constraints used for training the model. The training sets are a series of increasing subsets of the full trainign set (see Equation (8.1)).

For MLR, DMLR and RDNN, the final generalisation performance is almost reached with around 300 training constraints. Only few improvements if any are made after this point by these algorithms. This suggest that the maximal generalisation results

8.3 General Performance

are reached with comparably few training constraints, especially for MLR, which shows most gain within the first 100 constraints. A typical result for SVM is the very slow and more linear increase of generalisation performance over the growing training size. The final generalisation result is similar to the performance of MLR.

8.3.2 Relation of Training and Test Set Performance

Although the behaviour of SVM could feed the expectation of even higher generalisation for larger training sets, this is not likely given the combined features used for this experiment: Figure 8.2 shows the maximal training performance for the training resulting in Figure 8.1. It is a usual case, given the optimal values for the regularisation trade-off c , that training and generalisation performance converge towards a close final performance. For SVM, there is some room to reach over 70% performance, but it is unlikely that RDNN and DMLR will show great increases in performance when trained with more constraints. MLR shows a notable exception, as it does perfectly fit the training data, but not increase in generalisation with increasing training set sizes. This might be related to a constant overfitting, and different regularisation strategies as well as checks on validation sets could help further increase the performance of MLR.

8.3.3 Training speed and efficiency

Running within different environments, the comparison of the time needed by the algorithms for the task above does not allow for direct conclusions regarding the algorithmic efficiency. SVM is used as its compiled windows executable, while MLR, DMLR and the MLP net run within the Matlab interpreter. Especially for the large feature spaces used with MLR and SVM, the RDNN as described in Algorithm 5 is still by far the slowest of the approaches described in this paper, using large amounts of time even for the small training sets.

SVM	MLR	DMLR	RDNN
5	40	30	60

Table 8.1: Average training time per dataset in minutes, accumulated over all 20 subset sizes

8.4 Feature Influence

8.4.1 Types of Information Contained in Features

As has been shown in our early publications (e.g Wolff and Weyde [pub:6]), both feature type and feature dimensionality have an influence on the algorithms' performances of similarity adaptation. We now present an evaluation of these parameters on the complete similarity data as described above. To this end, we compare the performances of SVM using

- acoustic-only features
 - single chroma via average or 4 cluster centroids
 - single timbre via average or 4 cluster centroids
- genre-only features,
- slaney-only features,
- combined acoustic features and
- complete combined features.

The performance of SVM has been chosen as representative algorithm for the following experiments, providing the most stable results regarding variations over datasets and features. Although MLR outperforms SVM in Figure 8.1, it needs careful parametrisation depending on the input features. The results for the features selected below should be comparable without needing to change the training algorithms parametrisation, but for MLR the optimal regularisation trade-off parameter c_{mlr} can vary by several orders of magnitude, including local performance maxima along the scale. Until an efficient validation-set based approach for selecting c_{mlr} is developed (see Future Work, Section 9.6), SVM is selected as the most reliable candidate for the examination of feature influence. Providing the baseline in these experiments, the constraint satisfaction performance of an unweighted

8.4 Feature Influence

Euclidean distance metric has been evaluated as well for all of the feature configurations. The results are plotted at the left of Table 8.2.

Features	Chroma(1/4)	Timbre(1/4)	Slaney08	Genre
Test	56.44 / 52.08	64.70 / 65.80	65.80	63.32
Training	61.60 / 59.48	68.97 / 66.27	68.06	68.91
Baseline	56.86 / 56.87	60.84 / 59.33	60.52	47.79

Features	Combined Acoustic(1/4)	Combined All(1/4)
Test	66.03 / 61.50	68.41 / 66.26
Training	71.53 / 76.08	77.74 / 83.92
Baseline	61.07 / 59.44	66.86 / 64.68

Table 8.2: SVM Single features test set performance. Values for single average audio features and 4-cluster audio features are separated by slashes (average / 4-cluster).

Table 8.2 compares the test set performance of SVM compared to an unweighted Euclidean distance, using different portions of the complete feature information available. Here, the combined features achieve the greatest performance, followed by the Slaney08, timbre and genre features. The Slaney08 features, including relatively high-level summary information on the clips, are particularly successful on the unknown test set constraints (difference to training only 2.06%), which the Chroma features prove least effective on (training difference above 5%). Here, even the baseline performance is not reached by the learning algorithm, pointing to an overfitting to the training data. The good performance of the timbre features compared to chroma resonates well with results of Sotiropoulos, Lampropoulos and Tsihrintzis [88] as reported in Section 2.5.2.

Also notable is the low baseline of the genre features: this is partly due to the sparsely populated feature space. Each song is assigned 2-3 genres, and only a few different discrete distance values actually occur on the binary vectors. Therefore many constraints are not satisfied because of equal distance ($dist_W(C_i, C_j) = dist_W(C_i, C_k)$). Besides this effect on the Euclidean distance, songs being annotated with the same genres results in a zero distance which prohibits training of these constraints, and degrades performance significantly, as has been shown in [pub:5].

Features	Chroma(1/4)	Timbre(1/4)	Slaney08	Genre	Acoustic (1/4)
Comb. All(4)	0.000 / 0.000	0.001 / 0.000	0.000	0.000	0.000 / 0.002
Comb. All(1)	0.000 / 0.000	0.015 / 0.002	0.008	0.000	0.000 / 0.013
Acoustic(4)	0.000 / 0.008	0.002 / 0.006	0.000	0.145	0.000 / -
Acoustic(1)	0.000 / 0.000	0.753 / 0.179	0.823	0.116	- / 0.000
Genre	0.000 / 0.000	0.076 / 0.244	0.037	-	
Slaney08	0.000 / 0.000	0.751 / 0.505	0.000 / 0.000	-	
Timbre(4)	0.000 / 0.000	0.251 / -			
Timbre(1)	0.000 / 0.000				
Chroma(4)	0.000 / -				

Features	Comb. All (1/4)
Comb. All(4)	0.086 / -

Table 8.3: Significance of performance differences between feature types (Wilcoxon signed rank p values). Significant values at the 5% level are set in bold type. Values for $p < 10^{-3}$ are shown as 0.000.

Table 8.3 shows that the differences between the chroma features and the others are statistically significant at the 5% level. Most of the differences between the Slaney08, genre and timbre are not significant. However, the combined feature sets are significantly better than any individual feature set. Clustering vs. averaging makes a significant difference only for chroma but not for timbre or combined features.

8.4.2 PCA processed Features and Dimensionality Reduction

We now compare similarity models using different feature types whilst fixing the model complexity. For models based on the Mahalanobis distance measure (see Section 6.2), the latter is determined by the dimensionality of the feature space. We now fix the model complexity by equalising the different feature dimensionalities. For the experiments below, all features are transformed to the intended dimensionality using Principal Component Analysis (PCA). Prior to training, all features are reduced to the same dimensionality by omitting the PCA dimensions with lowest variance across the dataset. Note that the PCA transformation is performed on

8.4 Feature Influence

the full set of features containing clips from the training and test sets. This pre-processing has the potential to bias the results by boosting performance similar to transductive learning when compared to the previous and other results based on inductive learning, although our observations do not indicate this. Still, as our goal is to compare performance between features, any equally distributed bias will not affect the relative comparison.

Given the different dimensionality of the features, determining the maximal number of their principal components, we compare two sets of dimension-reduced features: PCA12 for single chroma mean features, timbre mean features, Slaney08 features, combined audio features and combined all features, reducing the respective PCA-transformed information to the 12 dimensions carrying most of the variance. In the same manner, PCA52 features are built from 4-cluster chroma and timbre features, genre features, combined audio features and combined all features. In order to achieve a high dimensionality, the 4-cluster chroma and timbre vectors were chosen for the respective tests. They are also included in the combined features plotted here. The Slaney08 features do not contain enough information to build a single high-dimensional PCA feature. Still, these are included in the combined audio and combined all features in Figure 8.4. As above, SVM is used for comparing the effectivity of the different feature information.

Table 8.4 shows that learning on the PCA12 chroma features did not improve generalisation results when compared to the raw features in Table 8.2. The Slaney08 and timbre features both provide significant performance increase over chroma data. The combined features further improve the performance, with PCA12 all-features-combined reaching better result than the original features (see Figure 8.1).

All pairwise differences in test performance between feature types are significant at $p < 5\%$, except timbre vs. Slaney08 and genre vs. Slaney08, indicating that the reduced dimensionality makes learning more effective, at least with SVM.

Figure 8.3 and Figure 8.4 show training performance for increasing training test sizes with PCA12 and PCA52 features. With increasing dimensionality, maximal performance is reached much later, therefore needing more training data as an effect of a less steep learning curve for all of the features. Generally, the increased

Features	Chroma	Timbre	Slaney08	Genre	Audio Comb.	Combined
Test12	55.54	64.22	62.00	60.20	66.65	69.73
Training12	59.43	66.74	63.03	62.77	69.324	71.18
Baseline12	55.81	61.40	59.42	60.12	58.37	66.86
Gain12	-0.27	2.82	2.58	0.08	8.28	2.87
Test52	51.71	57.41	/	61.46	63.73	69.50
Training52	64.41	68.03	/	65.43	71.50	75.78
Baseline52	50.70	51.28	/	58.26	53.02	55.93
Gain52	1.01	6.13	/	3.20	10.71	13.57

Table 8.4: Summary of SVM single features test and training performance. The Slaney08 features are not available to 52-dimensional PCA features.

number of parameters allows for more specific optimisation whilst delaying the generalisation resulting from larger training sets.

With PCA12 features, in Figure 8.3, the chroma features show only little potential for learning. The Slaney08 and timbre features both allow for a significant increase of the metric's performance through learning. Finally, the combined features almost reach the performance of SVM with unreduced features in Figure 8.1. Thus combining different feature sources, even when reducing the dimension afterwards, proves to represent a powerful tool in feature design for music similarity. As above, most of the training success is achieved with small training set sizes, up to 100 constraints. For the chroma features, test set performance even slightly decreases towards reaching the full training set.

52-dimensional PCA features are compared in Figure 8.4. The graph, though bearing results quite similar to the one above, shows generally lower performance for the single features. The timbre features' performance drops by 7% in comparison to the raw and PCA12 features, and the chroma features loose further 1.7% in performance, rendering them useless for this single-feature application. As the performance of 4-cluster and simple average timbre features is almost the same for the raw feature type, the number of PCA components kept remains as the determining factor for the performance.

The training performance, as depicted in Figure 8.5, allows for the interpretation

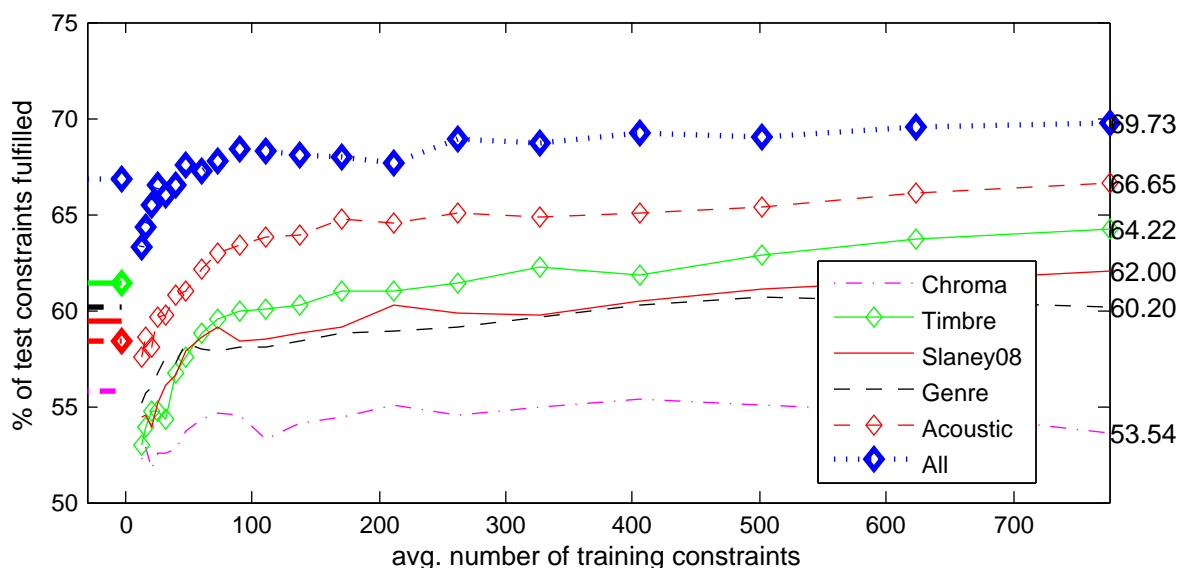


Figure 8.3: SVM Feature performance at 12 dimensions: chroma (mean), timbre (mean), Slaney08, genre, combined features. X-axis shows increasing training set size.

of the above results relating to bad generalisation of 52-dimensional features as a result of overfitting: The training performance of 52-dimensional PCA features is considerably (3-5%) higher than the performance of 12-dimensional PCA features. Furthermore, the baseline of the 25-dimensional features is much lower (-5% for all except genre features). Considering this, the performance gained by adapting a distance measure via SVM is far larger (factor 2-3) than for the 12-dimensional features. This good performance in adapting to the training constraints shows potential for adapting similarity models to similarity data even with inferior features, but this is not reflected in the test set performances.

8.4.3 RBM Features

In this experiment we evaluate how the performance of similarity models is affected by using the RBM feature transformation described in Section 5.3.2. We therefore train and apply RBM transformations on the combined features as a pre-processing step before similarity modelling. The PCA transform is also evaluated for comparison of the effects of the two pre-processing methods. As with the other [pub:2]

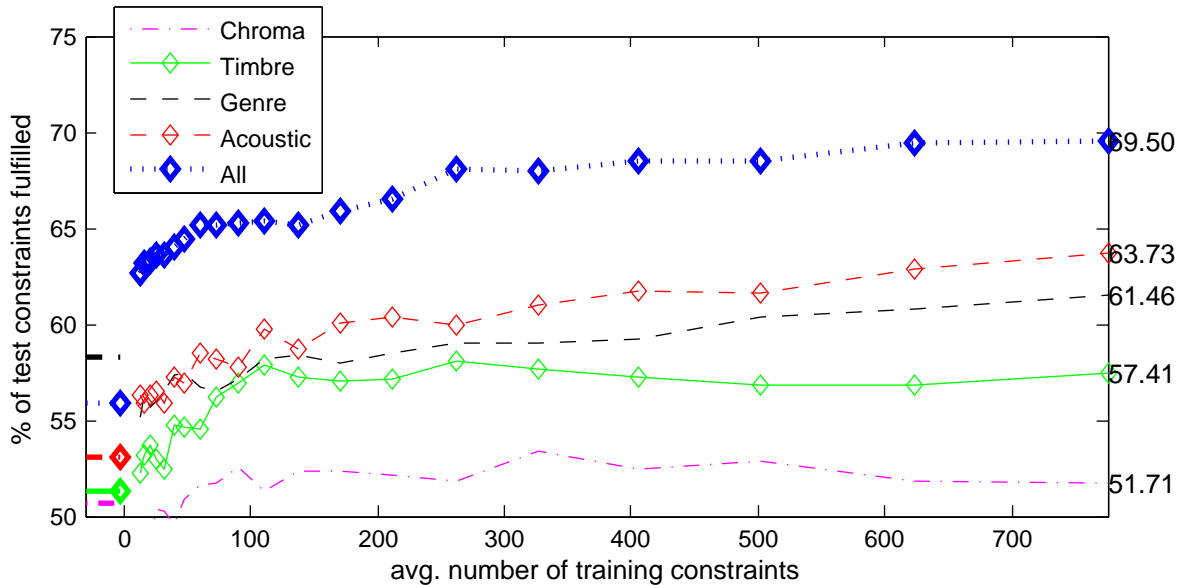


Figure 8.4: SVM Feature performance at 52 dimensions: chroma (4 clusters), timbre (4 clusters), combined audio, genre, combined features. Increasing training set size.

experiments in Section 8.3, we apply 10-fold crossvalidation and ID-sampling to make experiments comparable.

This experiment is a collaboration with Son Tran et al., and the resulting publication [pub:2] was awarded with the SoundSoftware Price for Reproducible Research¹. To allow for direct comparability, as described below, we use features already published as open data in [pub:9]. The complete data and code, including the CAMIR framework (see Chapter 7) and an RBM toolbox by Son Tran can be downloaded via a creative commons license online².

Different from the parameter-free PCA, the architecture and training method of the RBM transformation require several hyper-parameters, as listed in Table 8.5 to be set. Here, *hidNum* defines the number of hidden units in the RBM network, while the remaining parameters affect the learning rate and regularisation of the training itself. The parameters are selected using a grid search over a predefined range of values as displayed in Table 8.5.

¹<http://soundsoftware.ac.uk/rr-prize-aes53-winners>

²<http://mi.soi.city.ac.uk/datasets/aes2013framework/>

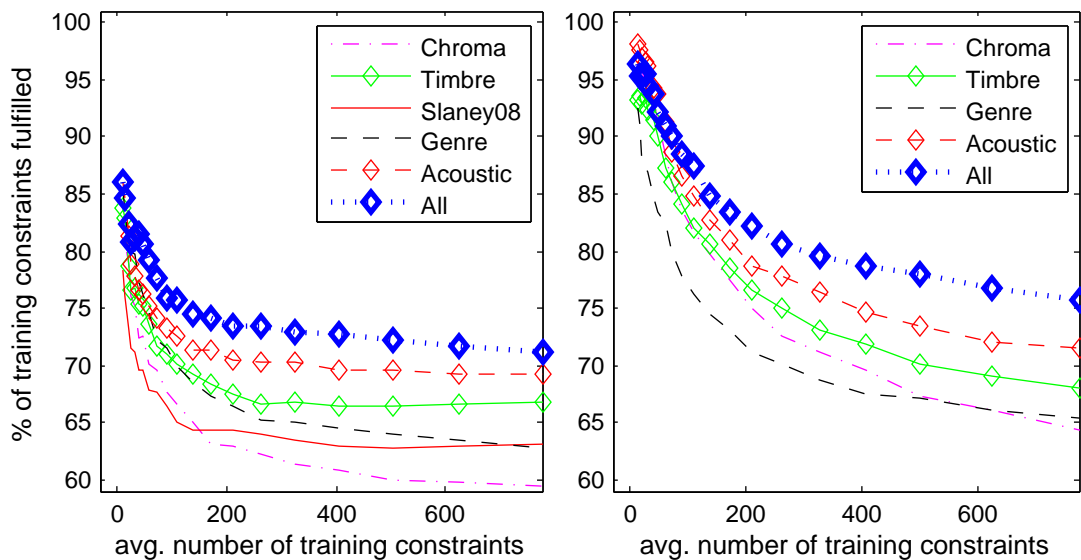


Figure 8.5: SVM feature training performance at 12(l) and 52 (r) dimensions: Increasing training set size.

Param.	Values Tested
<i>hidNum</i>	30, 50, 100, 500, 1000
<i>lrate1</i>	0.02, 0.05, 0.1, 0.5, 0.7
<i>lrate2</i>	0.1, 0.5, 0.7
<i>momentum</i>	0.05, 0.1
<i>cost</i>	0.00002, 0.01

Table 8.5: Values used for the RBM grid search.

For training the RBM, we use the complete set of features including songs from both the test and training sets of the cross-validation. This approach was used for a first evaluation due to time constraints, as it reduces the number of models to train by a factor of 10. Still, the similarity data is kept separate, and only the similarity data from the training set is used for model selection: We use the mean training set accuracy of each tested model to fix a model and its parameters to be used for the final evaluation. The results of those models on the test sets of similarity data, unknown to both the RBM training and similarity learning are then reported. Since using training accuracy for model selection is susceptible to overfitting, we apply strong regularisation during training of the models. The final

RBM parametrisations for each algorithm are depicted in Table 8.6.

Param.	Approach	
	GRAD	SVM
<i>hidNum</i>	500	1000
<i>lrate1</i>	0.70	0.05
<i>lrate2</i>	0.70	0.10
<i>momentum</i>	0.05	0.10
<i>cost</i>	$2.0e - 5$	$2.0e - 5$

Table 8.6: Parameters chosen for gradient ascent (GRAD) and SVM in the final experiments.

For further comparison with the results published in [pub:9], gradient ascent is added as a method of learning a simple weighted Euclidean model. Tran et al. [pub:2] explain some implementation details. Being very unstable over different RBM parametrisations, the result of the model with best training performance within 20 runs is reported for gradient ascent in each RBM parametrisation. Unfortunately, this was not possible with SVM because of time constraints, and the best results within 5 runs are displayed for this approach. We here update the values published in [pub:2] with those directly produced with the code published for reproducibility.

Appr.	Features		
	Original	PCA	RBM
GRAD	70.47 / 71.68	70.54 / 70.52	73.38 / 73.50
SVM	71.20 / 83.54	70.17 / 75.29	73.93 / 81.01

Table 8.7: Comparison of original features and those with PCA and RBM pre-processing. Test and training set results are listed as percentages of correctly predicted similarity constraints for the configurations with the best training success. The SVM Original values are taken from [pub:9].

Table 8.7 shows the performance of different feature pre-processing strategies. We compare original, unprocessed features with features transformed through PCA and RBMs. As we reuse features from experiments with Stober et al. [pub:9], they differ slightly from those presented in Section 8.3: In addition to the information contained in combined features, they also contain tag features derived from the

8.4 Feature Influence

tags in the MagnaTagATune dataset as described by Stober and Nürnberger [90] to allow for direct comparison. Furthermore, the standard deviations of chroma and timbre are added. When comparing the results for SVM with original features in Table 8.7 to our result in Figure 8.1, a small gain of 1.2% effected by the additional tag features can be seen, but the effect almost vanishes of PCA-transformed features. As the extraction of these features includes manual combination of tags [90], which has been specifically performed for the limited vocabulary in MagnaTagATune, we continue to use our more general features in other experiments to ensure transferability of our methods to datasets such as CASimIR.

We reran our gradient ascent (GRAD) implementation on the original features, and when compared to [pub:9] using the same features, the results are very similar. The SVM results are taken from this paper as the implementation and data are the same. When using PCA, the results for SVM are slightly worse than in the original features, while the gradient approach does improve very little.

The RBM features improve the results for all approaches, with gradient ascent showing the greatest generalisation performance boost by 2.91% over the original features. Still, SVM gains a significant ($p = 0.0684$) 2.73% through the transformation, and larger gains are to expected if more RBM models can be tested.

When finally compared with Section 8.3, SVM gains 4.88% of performance, rendering the RBM transformation an attractive method for improving similarity models with relatively simple model training methods such as gradient ascent and SVM. A drawback of the RBM technique is its stochastic nature, hindering exact reproduction of experiments. Also, the information contained in the RBM for transforming the data cannot be analysed to a level accessible for music research yet. Continuing the comparison with earlier experiments on MagnaTagATune, we now discuss the bias introduced though transductive learning as used in the first experiments with this similarity data.

8.5 Sampling: Effects of Transductive Learning

As detailed in Section 8.1, sampling for cross-validation can be realised as inductive (ID)-sampling, like in the experiments presented in this thesis and [pub:2, pub:7, pub:9], or as transductive (TD)-sampling ([pub:4, pub:6] and [91]), where pairs and individual clips (but not constraints) can appear in both training and test set. Figure 8.6 shows the results for the SVM, MLR and DMLR algorithms. The baseline shows the performance of an unweighted Euclidean distance measure for the test sets. During cross-validation, baseline results are averaged over all test sets and the average performance is calculated for the whole dataset. With TD-sampling, both MLR and SVM performance are significantly better than the baseline (both $p < 0.001$).

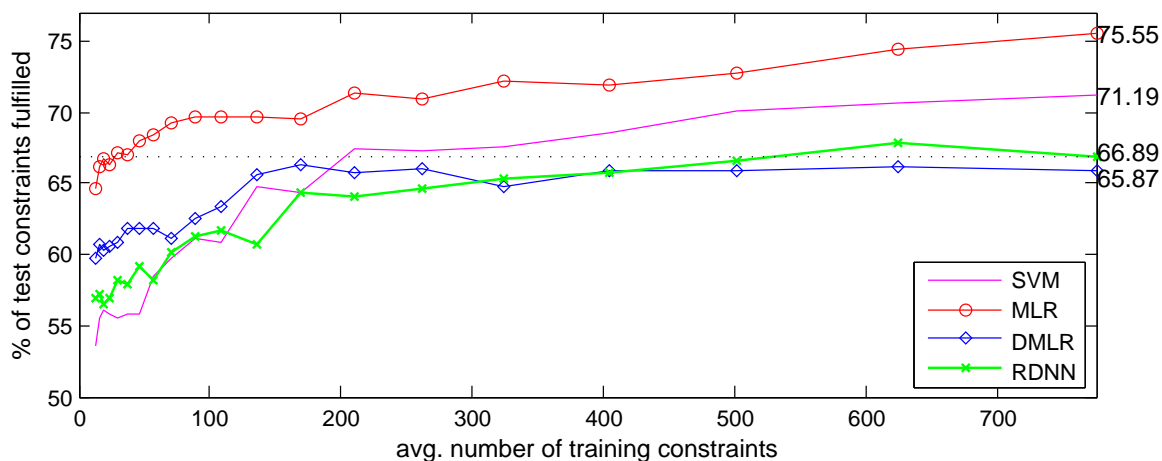


Figure 8.6: Transductive sampling: SVM, MLR, DMLR and RDNN test set performance for full features. The training set size increases from left to right.

The training performance of all algorithms displayed is similar to the performance with ID-sampling as plotted in Figure 8.2. In contrast, the performance on the test sets, as in Figure 8.6, shows a considerable increase of performance (6%) for MLR and a slight increase for SVM. This reproduces our findings in Wolff et al. [pub:9]. Involving almost all the feature vectors of the test set in training allows for MLR to make better decisions when the separation oracle selects the instances of the constraints to involve in the optimisation process (see Section 6.2.2). For the

Support Vector Machine SVM, the set of possible support vectors is increased with the number of feature vectors. This increase of data amounts to 10% additional (93 clips, see Section 8.1) feature vectors referenced during training via TD-sampling.

8.6 Learning with Weighted Constraints

We now focus on another aspect of the similarity data which has been neglected in experiments with relative music similarity data so far: The weights of individual constraints $\alpha_{(i,j,k)}$ can be used to prioritise constraints where many participants agreed with their input. One hypothesis we explore is whether prioritising constraints with large weights leads to a better generalisation performance of the learnt model. To this end we use our new WMLR and the SVM-based training method for weighted training of similarity models, and compare their effectiveness using weighted and unweighted performance measures. The following experiment is designed to evaluate their ability to train models according to the constraint weights.

8.6.1 Weighted Performance

So far, performance has been analysed on an unweighted basis, each constraint having the same importance at the start of the model training. As described in Section 3.2, the 860 unique similarity constraints in MagnaTagATune represent $\sum_{(i,j,k) \in Q} \alpha_{(i,j,k)} = 6898$ weights after removal of inconsistencies in the similarity graph. The vote difference for each edge can be used as an indicator for the reliability of the constraints. In the following experiment each constraint (i, j, k) is weighted in proportion to its weight $\alpha_{(i,j,k)} > 0$, using the new weighted MLR training introduced in Section 6.2.3 and weighted SVM (see Section 6.2.1).

To this end, instead of performing an unweighted evaluation considering the unique constraints satisfied, as used above, we measure the *weighted performance* of a metric as sum of the weights satisfied by the metric divided by the total sum of weights in the respective test or training set.

$$\frac{\sum_{(i,j,k) \in Q_{good}} \alpha_{(i,j,k)}}{\sum_{(i,j,k) \in Q_{base}} \alpha_{(i,j,k)}}, \quad \text{for} \quad (8.2)$$

$$Q_{good} = \{(i, j, k) \in Q_{base} \mid \text{dist}_W(C_i, C_j) < \text{dist}_W(C_i, C_k)\}.$$

The test set performance is then defined with $Q_{base} = Q_{test}$ and the training performance analogously with $Q_{base} = Q_{train}$.

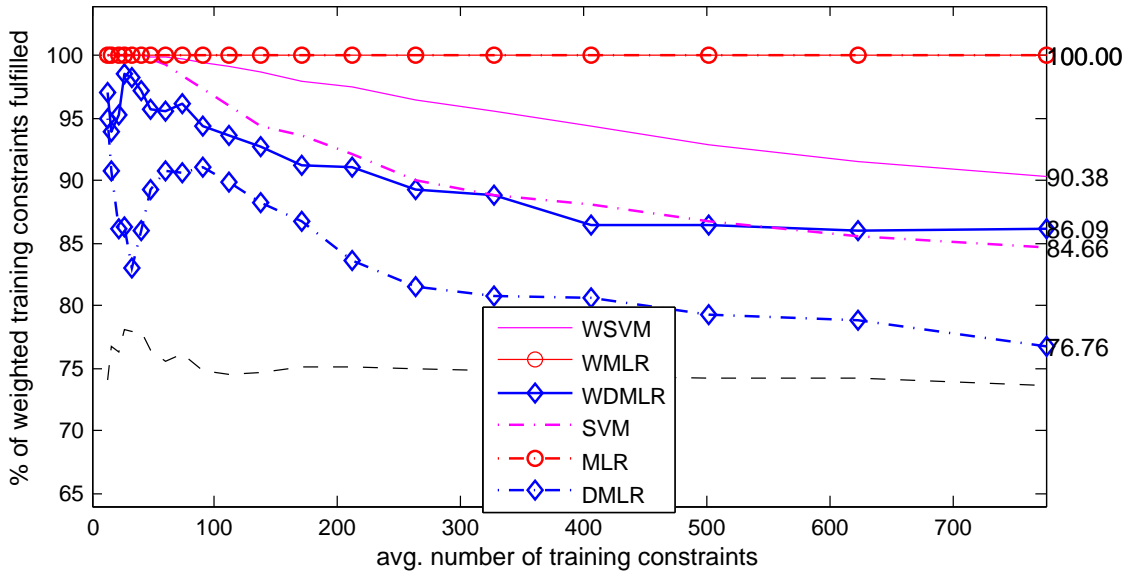


Figure 8.7: Overall training set performance, weighted evaluation for training with: SVM, weighted SVM (WSVM), MLR, DMLR, WMLR and WDMLR. The bottom dashed curve displays the weighted baseline performance.

Figure 8.7 shows the weighted performances on the training sets of weighted training with WMLR, WDMLR, and SVM. We compare these to weighted performance (E:W) of the unweighted training with MLR, DMLR, SVM and an Euclidean metric as baseline. For the Euclidean metric, the weighted evaluation yields about 6% better performance than using unweighted evaluation, indicating a correlation of the weighted constraints with the Euclidean distance in feature space. For WMLR and MLR, satisfying 100% of the unique training constraints, the weighting makes no performance difference. The results of the other algorithms improve by similar amounts as the baseline. This shows the weighted learning approach described in Section 6.2.3 succeeds in improving results towards the weighting of the constraints supplied during training.

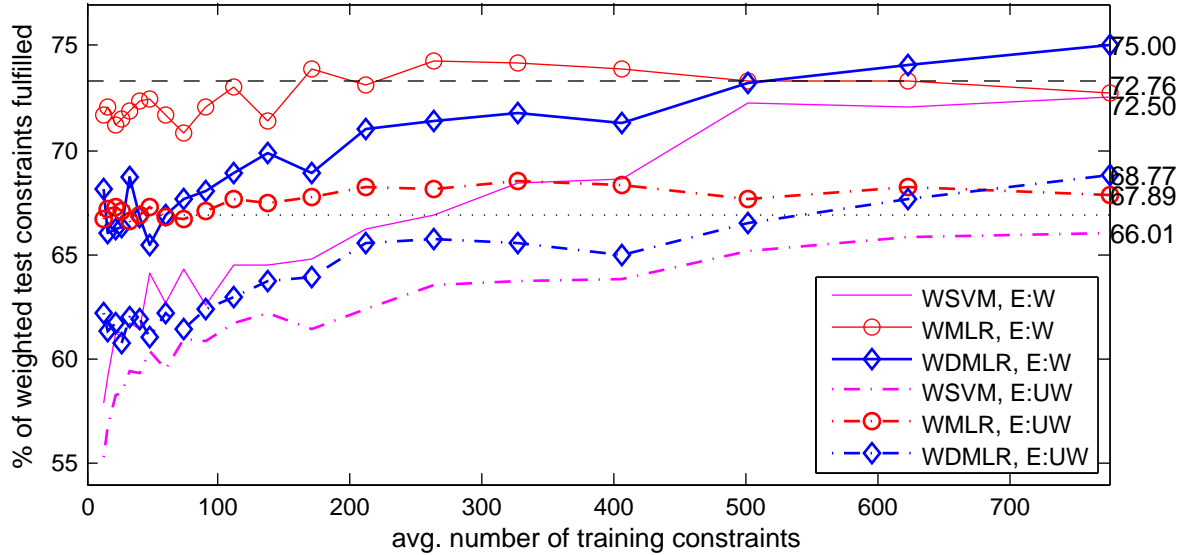


Figure 8.8: Overall weighted (E:W, -) / unweighted (E:UW, · - ·) generalisation performance for weighted training: WSVM, WMLR, WDMLR

The full lines in Figure 8.8 show the generalisation results of weighted training with WMLR, WDMLR, and SVM on the test sets. Here, only WDMLR exceeds the baseline performance for weighted evaluation, which is also the only significant result on test sets in this comparison. Thus, the other algorithms fail to improve generalisation by using the weights. Given that the DMLR training performance was lower than for the other algorithms, this seems to indicate that the lower model complexity of WDMLR allows more effective learning on the weighted dataset.

8.6.2 Effects on Unweighted Performance

Considering the unweighted performance of the models learnt from weighted constraints, drawn as dotted lines in Figure 8.8, the tested algorithms perform worse than in Figure 8.2, but still significantly better than the baseline. According to the Wilcoxon test, only WDMLR can reach significant improvement above the baseline, although weighted training is effective on the training data. On overall, the use for weighted data from MagnaTagATune seems not to improve the generalisation of learnt models.

However, as the distribution of weights we extract from MagnaTagATune depends on both the number of votes and the ratio of conflicting vote (see Section 3.1.1), there is no straightforward interpretation of these results. We now continue the evaluation of geographically annotated similarity data as collected from the Spot the Odd Song Out game. Note that the following experiments are again using unweighted training and evaluation methods.

8.7 Geographically Specific Similarity Models

In this section we report from our experiments towards creating culture-aware similarity models, using CASimIR as the first country-annotated relative similarity dataset. We analyse how specific similarity models for cultural subsets of the similarity data can be used for learning local specificities of music similarity. Therefore, we divide similarity data collected by Spot the Odd Song Out into four *single country data sets* as described and analysed in Section 4.3.1.2. The subsets were selected based on the location associated to the IP address of the users providing the similarity data: \hat{Q}^{De} (Germany), \hat{Q}^{Fr} (France), \hat{Q}^{Sw} (Sweden) and \hat{Q}^{Uk} (United Kingdom). Besides these datasets, we define complimentary datasets such as \hat{Q}^{FrSwUk} combining the similarity data from all countries but one (De in this case).

Note that these are the first experiments reported from the CASimIR similarity dataset. Unfortunately, for the CASimIR dataset, no detailed genre information is yet readily available. Although the Million Song Dataset dataset provides tags from Last.fm and MusicBrainz, using these tags as raw binary features did not increase performance in earlier experiments. Therefore, the following experiments are using the combined acoustic features as defined in Section 8.4.1.

To allow for comparative analysis of the different learnt similarity models and transfer learning, we use the RITML algorithm (see Section 6.3.6.1) for training the single country similarity models. For the single models, initial tests with MLR and SVM provided similar or lower model performance than the ones reported for RITML below. Firstly, we test RITML separately for adapting similarity models to each single country data set using 10-fold cross-validation. RITML is chosen as

the method for training as it performs similar to other algorithms on the single-country datasets, but also allows for the transfer learning strategy as applied in Section 8.7.1. The two bottom rows in Table 8.8 show the test set results of RITML and an Euclidean metric serving as baseline. As before, results are reported in percentage of fulfilled test set constraints. Compared to the baseline, training with RITML clearly improves results of those datasets. On average, the generalisation performance is increased by over 3%. Still, regarding statistical significance the results are not significantly better: A Wilcoxon signed rank test comparing the baseline and RITML over all four datasets results in a p -value of 14.6%. Also, we would expect better results from earlier experiments. In Table 8.2, the acoustic features achieve over 66% of generalisation performance. It seems like the datasets are too small to allow for good generalisation by themselves alone.

8.7.1 Transfer Learning

In order to improve the generalisation results, we now apply transfer learning. The overall process is depicted in Figure 8.9. First, an experiment with 10-fold cross-validation is performed on each of the 3-country datasets Q^{FrSwUk} , Q^{DeSwUk} , Q^{DeFrUk} and Q^{DeFrSw} using training and test data from these sets. Then, for each of the 3-country datasets we select the Mahalanobis matrix with best generalisation (on that dataset) as template W_0 for further usage.

Now, the template Mahalanobis matrix W_0^{FrSwUk} can be used as starting point for another RITML learning step on the one-country dataset Q^{De} and so forth. We call this learning step W_0 -RITML or “fine tuning”. Our hypothesis is that this can allow us to leverage commonalities in the datasets. Also, during the fine tuning, we can observe the deviation of W from the template matrix W_0 as indicator of specifics of the single country dataset (see Section 8.7.2).

The top row of Table 8.8 shows the results of this fine-tuning step in “ W_0 -RITML”. In order to evaluate our method of transfer learning, we also report the performance of the Mahalanobis distances based on W_0 template matrices without fine tuning, labelled as “ W_0 -Direct”. Note that none of the test set constraints whose results are reported here have been used for training of the template matrix W_0

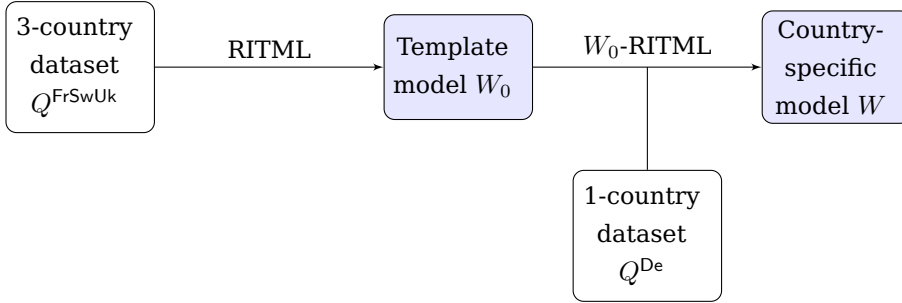


Figure 8.9: Flow diagram for the fine tuning process, exemplified for the Q^{De} dataset.

or during fine-tuning. As the results are based on the same test sets as the ones reported for RITML and Euclidean baseline, they can be directly compared.

As a last means of comparison we add the performance of RITML when trained on data from all countries. To this end we join the training data of the corresponding three country datasets with the training data from the previously excluded one country dataset (e.g. $Q_{\text{train}}^{\text{FrSwUk}} \cup Q_{\text{train}}^{\text{De}}$ for test on $Q_{\text{test}}^{\text{De}}$). This test corresponds to a test where a general model is trained with training data from all countries and tested using a specific one country test set. The results are reported as “JOINT” in Table 8.2.

	Q^{De}	Q^{Fr}	Q^{Se}	Q^{Uk}	Avg.
W_0 -RITML	69.28	64.34	64.40	70.36	67.09
W_0 -Direct	67.61	65.97	64.81	69.02	66.85
JOINT	67.80	67.39	64.05	70.46	67.43
RITML	64.35	62.71	61.75	63.78	63.15
Euclidean	60.79	62.09	58.11	62.65	60.91

Table 8.8: Test set performance per country of different learning strategies. The average performance over all countries is denoted in the rightmost column. The highest performance is highlighted per column.

The most notable result in Table 8.2 is that the three methods using additional data outperform RITML and the Euclidean baseline. According to the Wilcoxon test, these three top algorithms are significantly better on each single country dataset as well as on average ($p < 3.1\%$). This difference is highlighted by a divider in the

table.

The results of the three best performing algorithms are not significantly different. On average, the JOINT model shows the highest performance, showing that in this case the separation of data based on country of input was not helpful in general.

Still, when examining the success of fine tuning (W_0 -RITML), we find that it outperforms W_0 -Direct on average and in two out of four individual cases (De and Uk), with the result of W_0 -RITML outperforming all other approaches for Q^{De} . Thus, it was possible to specialise the similarity model towards Q^{De} , but not to the other datasets. The changes applied to the template matrix W_0^{FrSwUk} during fine tuning are discussed in Section 8.7.2 below. Also, both W_0 -RITML and W_0 -Direct show lower p -values ($p \leq 0.5\%$) than the JOINT approach ($p \leq 3.1\%$) and are thus more significant when compared to RITML and the Euclidean baseline.

The results of W_0 -Direct underline that similarity information can be stored and transferred via the Mahalanobis matrix, without the need of making the underlying clip and similarity data accessible to training. Furthermore, the success of W_0 -RITML on Q^{De} shows that further fine tuning can be effective given a suitable dataset.

8.7.2 Analysis of Learnt Similarity Models

In the above experiment, a similarity measure $dist_W$ was created by firstly calculating a template Mahalanobis matrix W_0 using the three country dataset Q^{FrSwUk} and then fine tuning it using W_0 -RITML with Q^{De} . As the performance of the resulting matrix significantly improved during fine tuning, we now analyse the changes made to W_0 :

As formulated in Equation (6.8), the Mahalanobis matrix transforms the facet difference vectors during the calculation of the Mahalanobis distance function. This effectively results in summation of combinations of different facet differences with weights determined by W .

Figure 8.10 on page 184 shows the relative difference of the Mahalanobis matrix before and after fine tuning. As the fine tuning process can rescale the similarity

measure and thereby W , the matrices have been normalised to the interval of $[0, 1]$ via

$$\hat{W} = \frac{W - \min_{i,j \in 1 \dots N} (W[i, j])}{\max_{i,j \in 1 \dots N} (W - \min_{i,j \in 1 \dots N} W[i, j]) [i, j]}. \quad (8.3)$$

The operations and subtraction and division are applied to W on a point-wise manner. Dark red colours indicate weights have been increased during fine tuning towards Q^{De} given the template W_0 . Blue colours show a decrease of weights towards the original matrix. In the very centre of the matrix a dark red corresponds to a specific importance of a timbre component for the Q^{De} dataset. Also, the heightened correlation of *tempo* and *tatums confidence* to timbre (see mid-bottom) is specific to this dataset. When examining where weight has been subtracted during fine tuning, we firstly recognise that generally weights are taken from the diagonal to more specific off-diagonal entries. *Segment duration*, *numTatumsPerBeat* and loudness factors have lost most weight on the diagonal.

Figure 8.11 on page 185 shows the final Mahalanobis matrix after fine tuning on Q^{De} . Red / dark colours correspond to high values while yellow / light colours indicate to low values. The diagonal of the Mahalanobis matrix shows most of the features being assigned an equal weight. The centre of the matrix contains some *timbre* components with particularly high weights even for close off-diagonal combined features. Also, combinations of *tempo* and *tatum* with *timbre* features show high weights in the lower centre of the matrix.

The homogeneous weight distribution along the diagonal, shows that the Q^{De} similarity data corresponds well to a Mahalanobis distance close to the euclidean metric. This is interesting as Figure 8.10 shows this is less the case for the template matrix W_0 derived from the Q^{FrSwUk} dataset. Moreover, the final matrix W (Figure 8.11) contains some negative weights which, as discussed in Section 6.2, disqualifies the resulting distance measure as a metric. Still, the performance of the resulting distance measure encourages the allowance of small violations of W 's positive definiteness.

8.7 Geographically Specific Similarity Models

These results should provide a methodical incentive for further research into recommendation for (cultural) user groups. In the above experiments, the Q^{De} dataset is too small to make more generic or ethnomusicological assumptions from the peculiarities of the generated distance measures. Still, we have shown that analysis of Mahalanobis distance measures provides relevant information when applied to music similarity datasets from selected users with specific attributes.

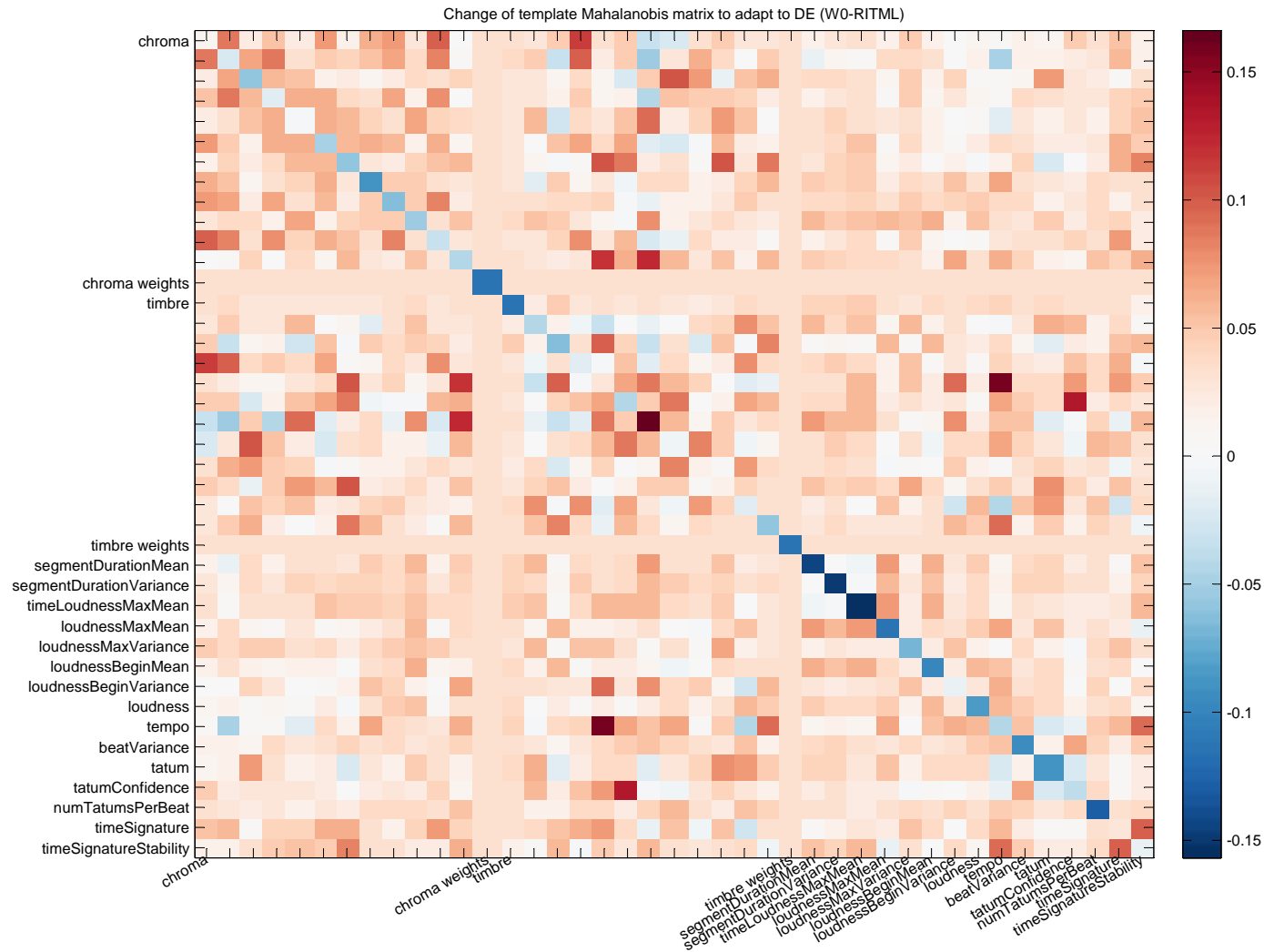


Figure 8.10: Difference of normalised Mahalanobis matrices before (W_0 template) and after (W) fine tuning. Axes show associated features. Dark colours indicate feature weights were changed when compared to the template W_0 . Red corresponds to weight increase, blue indicates a decrease. Some blocks appear larger due to print scaling.

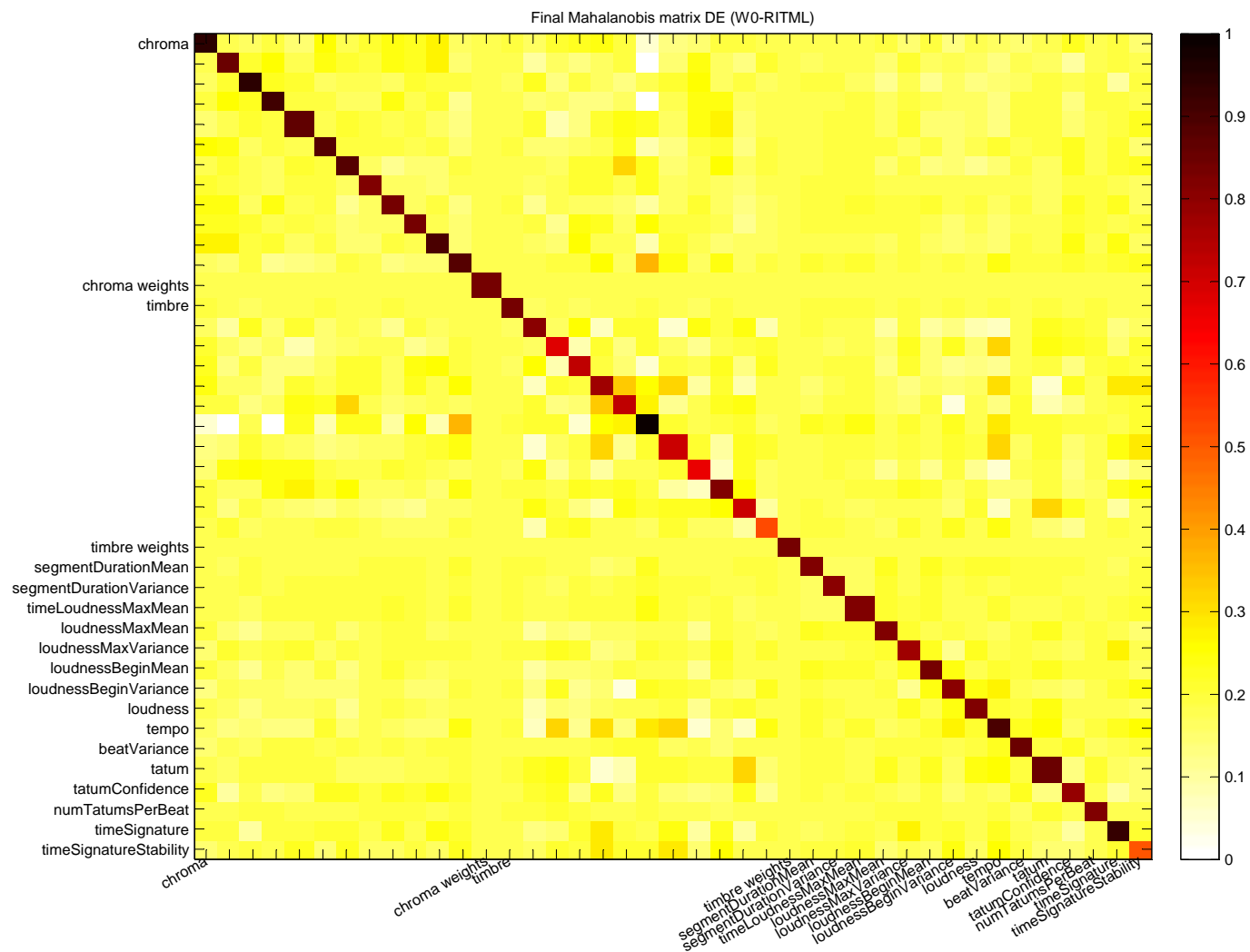


Figure 8.11: Mahalanobis matrix W after fine tuning by W_0 -RITML. Dark red colours indicate large weight of the feature coefficient. Some blocks appear larger due to print scaling.

8.8 Conclusions

In this chapter, to solve research question *rq:1*, we presented a comprehensive evaluation of the various facets of modelling music similarity from relative similarity data.

Our comparison of metric learning methods showed that using various supervised learning methods we are able to improve model performance over a Euclidean metric baseline. Still, our answer to *rq:1* is not singular, as the choice of algorithm clearly depends on the scenario: Both MLR and SVM achieve statistically significant improvements over the Euclidean metric, reaching up to almost 70% of training performance with standard combined features. Quantitatively, the approximate 2% of improvements that we achieve point to a difficult learning problem. Our experiments show performance to depend on dataset sizes, feature information, and consistency of data. Apart from consistency, which might be inherent to the subjective notion of similarity, the other factors can be addressed by further data collection and more elaborate features. MLR results are better than the ones presented for other methods, especially in training, but SVM, reaching similar generalisation results, is more efficient in the implementation we used. DMLR shares the drawbacks of the MLR approach whilst typically performing worse than SVM. Generally, parametrisation is key to using either MLR algorithm successfully.

The experiments with RDNN show low performance in all tasks despite the potentially higher flexibility of the model. However, the near perfect training performance of the MLR shows that the flexibility of the Mahalanobis matrix is already sufficient. With the exception of gradient ascent, all algorithms showed high differences in performance between training and test sets, even with optimised regularisation. This indicates that improving the amount of data may lead to improved results, given the level of inconsistencies in the data does not increase.

For our experiments we used the new graph-based ID-sampling method for unbiased selection samples for cross-validation of model performance. Our experiments in Section 8.5 confirm that for the MLR and SVM methods, significant performance boosts of up to 5% and therefore a bias in evaluation result from the

8.8 Conclusions

naïve TD-sampling used in earlier experiments, as it allows for feature information to be shared in test-and training sets. The experiments presented here were therefore performed using our new ID-sampling method. Still, in specific applications with known and fixed music data, transductive learning can be helpful. We thus distinguish the two evaluation methods in respect to research question *rq:3* depending on the evaluation context.

When considering the weights derived from the similarity data, we showed that our new weighted version WDMLR as well as weighted SVM are able to learn from the additional information. Moreover, we observe that the standard Euclidean metric corresponds well to the vote weights, increasing performance when using weighted evaluation. But for generalisation, apart from a weak gain by WDMLR, the weighted data turned out to be not easily generalisable. Unfortunately, the DMLR method WDMLR is derived from does not show high performance in any other task.

The choice of input features has significant effects in almost all experiments in Section 8.4.1, even if the input dimensionality is normalised as in the PCA12 and PCA52 datasets. Chroma features generally perform poorly, timbre features show best single performance, while genre and the music-structural features defined by Slaney, Weinberger and White [84] add useful information. The calculation of clusters for chroma and timbre features provides additional information to the system, but the simpler averaging features show more stable results especially with 52-dimensional PCA features. The single most effective way to improve the performance is to combine different types of features, which yields significant improvements over all individual features, regardless of whether clustering or dimension reduction is applied or not. Combining the available feature information results in more complex models, but already the Euclidean model with combined features outperforms any single feature performance after training. When using only acoustic features, the model training is particularly effective (5 percentage point improvement) and most useful, as the baseline is much lower for only acoustic vs. all features (61% vs 67%). Thus, using trained models can be particularly helpful when additional non-acoustic information, such as in genre features, is not available.

The reduction of the input dimensionality with PCA (Section 8.4.2) has no significant effect on the generalisation with either the 12- or the 52-dimensional feature sets, although the training results improve considerably with larger feature dimensionality, at the cost of generalisation for the acoustic feature types. This is likely the result of slight overfitting due to the number of parameters increasing with the feature dimensionality.

The generalisation results show that the SVM algorithm is robust and extracts relevant information from input data in high and low dimensions. Higher dimensional features might improve in generalisation given a greater amount of training data. We cannot find any improvement of the combined PCA features over the raw feature equivalent, and thus no evidence of possible transductive learning effects in feature pre-processing is detectable.

Our experiments in Section 8.4.3 show that transforming features using RBMs does improve both training and test set results of similarity learning with gradient ascent and support vector machines, with generalisation results improving more than 2.7%. In fact, the SVM result of 73.9% represents the best model performance reported in this thesis, although using additional features specific for MagnaTagATune. Comparing to the features extracted using PCA, the features learned from RBM show better performance and more consistent improvement in all three approaches. The results for gradient ascent show that given RBM pre-processing, even simple training methods can achieve competitive model performance. A likely benefit of RBMs is that they allow to use linear regression with non-linear combinations of feature types. Like the PCA features, RBM are trained on the full feature data and gains might be related to transductive learning from the features also included in the test sets, although this was not observed for SVM in Section 8.5. A drawback of the RBM transformation is its black-box character which, at the state-of-the-art, prohibits further deduction of musicological knowledge from the models.

Section 8.7 provides a novel example of such analysis with unprocessed features: The analysis of influence of specific features for model fine-tuned to CASimIR similarity data collected in Germany is a first step towards culture-aware similarity

8.8 Conclusions

modelling and analysis, and opens future perspectives for research in user attributes for *rq:1*. Although the amount of data in CASimIR did not suffice to learn competitive individual models from data of single countries, our new method for transfer learning with W_0 -RITML allowed for country-specific models with competitive performance of 69.28%. The models learnt from combined-country data achieve 70.46% performance which are not significantly improving results. Already without genre tag features, which are not yet available for CASimIR, the models outperform acoustic-only results (66.03%) from MagnaTagATune in Table 8.2. This proves the our new RITML algorithm to be effective and competitive with methods tested before, while enabling transfer-learning with template similarity models. The culture-specific modelling of similarity furthermore aims at generating models for culturally similar participants, ideally resulting in more homogeneous similarity datasets. With enough similarity data available, we aim at higher performance for the specific models for data from their target group of participants.

9 Summary and Outlook

In this thesis, we have developed methods for analysis, computational modelling and evaluation of music similarity, based on relative data collected from humans. Relative similarity data are easy to collect as they only require qualitative statements from participants. However, this data type has not been given much attention in the Music Information Retrieval (MIR) research field so far. We presented the application and evaluation of new and existing methods for similarity learning from relative data on existing and new datasets. Our evaluation showed that learning similarity models with the presented methods is effective, but limited to small improvements. Specific properties of training algorithms, music features and evaluation strategies were pointed out. We also introduced a method and framework for collection of music annotations, including relative similarity data, on the web and in social networks. The data collected with a game based on this framework allowed us to explore culture-specific similarity modelling. Our analysis of data quality in MagnaTagATune provides suggestions for improvements in future collections, and the feature analysis pointed out the importance and possible potential in feature development. Generally, the results presented here and in related research indicate that predicting music similarity is a difficult task: We observe a limitation of generalisation performance similar to the *glass ceiling* effect in MIR tasks which also substantially relate to (cultural) information external to the audio, e.g. genre recognition.

9.1 Thesis Background

We motivated our research in Chapter 2 with the embedding of music similarity in research paradigms of musicology and MIR. Existing research and the multiplicity

of usages of the term similarity show the strong role it plays in research and various applications of these and other fields.

This thesis focuses on the analysis of similarity data from human ratings, with the overarching aim to relate similarity to cultural features. Most work on music similarity has been done in psychological studies with relatively small quantities of data. We have developed methods for training models on larger datasets and evaluated them on the MagnaTagATune dataset, which is the largest available dataset with user-based similarity data.

So far, there was no dataset available with both similarity and user information. Games With a Purpose (GWAPs), which motivate participants through enjoyment and user interaction, have recently become popular for collecting user data. In order to facilitate further research, we have developed a GWAP framework and collected the new CASimIR dataset which is the only available dataset with music similarity and user information.

Most research on similarity models in MIR mostly uses proximate information, such as preference data, as ground truth for learning similarity, as larger datasets and mainstream machine learning methods are available for such data. This thesis presented methods enabling the collection of large sets of relative music similarity data as well as modelling strategies for this data type, as these are currently scarce. The central parts of this thesis are now summarised in three steps: collection and preparation, similarity models and methods for training leading to implementation and evaluation.

9.2 Data Collection, Analysis and Preparation

In Chapter 3, answering research question *rq:4*, we provided an overview of analysis and processing techniques for relative similarity data which serves as ground truth for the training and evaluation of all models in this thesis. We showed how similarity constraints can be extracted from odd-one-out statements, and how this information is represented through edges in a clip-pair graph. As we represent

multiple votes via weights in the similarity graph, a strategy for dealing with inconsistencies between participant data entries, which appear as cycles in the graph, is described. We thereby found that analysing the connected components in the similarity graph allows for prediction of the largest cycle to be expected and thus to analyse the data quality.

Our analysis of MagnaTagATune presented in Section 3.2 is the first comprehensive analysis of the dataset and presents a pioneering methodological example for future large relative similarity datasets. The MagnaTagATune similarity dataset is the first available dataset of its kind, containing relative similarity data for 1019 clips. With the audio, feature and similarity data freely available, it presented a useful dataset for our experiments on similarity modelling. However, our analysis revealed several several issues that potentially impede effective learning and interpretation of results. We found that the triplets of clip pairs in the similarity data do not overlap between presented clip configurations, which prevents the study of learning transitivity. Furthermore, the data has an unsystematic distribution of genres over the test triplets, leaving them very heterogeneous across genres. In informal tests on the MagnaTagATune dataset, subjects found it difficult to make a decision in the odd-one-out scenario, because each of the clips came from a different genre. This concludes our main answers to *rq:4*.

The shortcomings of MagnaTagATune without any alternative datasets encouraged us to start a new dataset collection. Question *rq:5* inquired how large amounts of data can be gained efficiently and with high control even for a web-based application. Chapter 4 introduces the CASimIR framework for collecting media annotations with games as a purpose, which is the basis of the Spot the Odd Song Out game. The framework presented is built with the maxims of open source, open data and modularity. The presented approach of separating survey design and media selection from participant user interface and game logic provides a new methodology for GWAP development. This enables the development and usage of several user interfaces on the basis of an independent central data back-end, thereby acknowledging that different development strategies are optimal for different parts of a GWAP. Our interface framework is built on modern web-standards and enables cross-platform use on any HTML5-ready browser. The resulting API at

the back-end now defines data annotation tasks via a strong specification, providing functionality such as dynamic example choice and user input storage. This enables a survey design that includes dynamic adaptation to a growing number of participants. The code is released as open source and the modularity of the implementation provides a reusable basis for new data collection projects.

Resulting in a collaboration with KTH Stockholm and several publications, the data collected via Spot the Odd Song Out exemplifies the usability of our framework: In Section 4.2.5, we presented an early overview of first data collected, which confirmed the effectiveness of our methods for online data collection. The new CASimIR similarity dataset, although not yet the size of MagnaTagATune, achieved the desired requirements of strong interconnectivity between presented clip triplets and stronger genre conformity. Finally, the ability of CASimIR to integrate with social networks and collect participant attributes allowed for our creation of a first country-specific relative similarity dataset in Section 4.3.1.2 and later comparative experiments. We furthermore provided tempo and rhythm datasets using real-time data of participants tapping to music as well as first analysis results.

In Chapter 5 we discussed acoustic, cultural and metadata features for learning music similarity on large databases. Following an introduction of feature extraction and its function in MIR, we showed how basing analysis on The Echo Nest features instead of raw audio can enable research on large databases including copyrighted recordings. For acoustic features, an aggregation approach was introduced in Section 5.1.1.2, using clustering to represent different harmonic and timbral clusters. We furthermore make use of medium- to high-level features built on The Echo Nest data. Another source of information included in our features consists of tag annotations, as they include contextual information about the music clips. We introduced the genres annotated by the Magnatune label as a valuable addition to the MagnaTagATune dataset, and derived tag features from it, which can be extended by tags from collaborative music platforms.

A novel post-processing approach we introduced is the transformation of features using RBMs. The unsupervised training of a probabilistic model allows for a transformation of features into a new feature space, with freely parametrisable feature

dimensionality. Our later analysis also uses PCA post-processing, which we here introduce as a means of dimensionality reduction and decorrelation of the features. Feature extraction is an influential component of music similarity models, as it determines the representation of music. The different options elaborated allow for a wider examination of *rq:1*. Furthermore, by using features from The Echo Nest, experiments can be reproduced without access to the potentially copyright-restricted audio data, enabling a large-scale evaluation of the methods (*rq:3*).

9.3 Similarity Models: Structural Framework and Training Methods

The central methods in this thesis, providing adaptation of models to similarity data were introduced in Chapter 6. Here, we discuss different model architectures for learning a distance measure as dual representation of the similarity model. We used the abstraction of facet difference vectors to model differences in clip pairs, which are then given as input to the distance measures. MLR and SVM-Light were discussed as two available state-of-the-art methods for metric learning on relative similarity data. At this occasion we contribute a new weighted variant of MLR-WMLR – as well as weighted learning with SVM. Extending this answer of *rq:2* by further alternatives to train models with relative data, we integrated an approach for transferring methods designed for absolute similarity data into our framework of relative similarity learning.

Using this framework, we here present two new algorithms for similarity learning from relative data: RITML is our new algorithm based on the ITML metric learning method, which allows for regularisation towards template Mahalanobis matrices. For relative training data, we thereby present a first method and application of transfer learning on the level of similarity models in W_0 -RITML. Our RDNN method uses a single multilayer perceptron to learn a distance measure from relative similarity data. This is the first model we present that is able to model asymmetric relative similarity data. Further presented applications of the framework enable the application of general regression methods and regression trees to relative similarity learning.

9.4 Implementation and Experiments

In order to evaluate the new and existing methods introduced above, we created the CAMIR framework for culture-aware music information retrieval which was presented in Chapter 7. Implemented in Matlab, the framework provides code for all methods presented in this thesis. This includes analysis of music and similarity datasets, with graph-based visualisation of similarity data, extraction of audio and tag features as well as the definition and training of similarity models. The experiment part of the framework manages a typical similarity learning workflow, including the creation of unbiased data sampling for cross-validation, grid search for feature parameters and concise experiment definition. In addition to our new methods, the framework includes external implementations that were generously provided by other researchers in Matlab, Python or c++. Being awarded the SoundSoftware reproducibility price with publication [pub:2], the framework runs on multiple platforms and consistently integrates measures for source code version control in experiment scripts and results storage, linking created data to the version of producing code.

Finally we presented a comprehensive evaluation of the similarity models and training methods in Chapter 8. This is the first comprehensive evaluation of similarity models adapted to relative data at this scale. In an introductory part, we discussed our evaluation strategy based on cross-validation and responded to peculiarities of our ground truth data (rq:3). We introduced a sampling method specific to relative similarity data that allows for fair comparison of training methods by avoiding transductive learning. Furthermore, many analyses are performed on training sets of increasing size which allowed us to determine the effect of training data quantity and make predictions for larger datasets.

In Section 8.3 we found that all model training algorithms could improve over a Euclidean baseline measure – although by only modest amounts of 2-3% –, reaching up to 70% generalisation with standard features. The substantially higher training set results point to issues with overfitting, in particular for MLR, as well as to similarity learning as a tough generalisation task. The state-of-the-art metric learning algorithms MLR and SVM provided the best, and particularly SVM the fastest and

9.4 Implementation and Experiments

most stable performance due to a simpler model (see *rq:1*). Our new RDNN model also increases performance, but needs further tuning to become competitive. A direct comparison of the new RITML and regression methods was not possible at the time of the general comparison, and is planned for future work.

The choice of input features has significant effects for similarity learning (see Section 8.4). For the MagnaTagATune data, timbre and Slaney08 features had highest performance when used alone, but the combination of many different feature types provides best modelling results: Already the euclidean baseline of combined features outperforms the models trained on any single feature type. The increased performance of the combined features comes at the cost of higher model complexity, as the dimensionality of the feature space grows by a factor of more than 10 times the dimension of timbre features. Although for model analysis as in Section 8.7.2, the individual feature dimensions are needed, dimensionality reduction can help reduce model complexity where such model analysis is not needed: Using the PCA transform with SVM model training, we analysed the model performance with regard to model complexity and feature information. Even with only 12 feature dimensions, and thus fairly reduced complexity of the Mahalanobis distance models, we found that combined features can still reach competitive generalisation performance of about 70%. The comparison to 52-dimensional PCA features showed that dimensionality of features mostly influences training results, with larger dimensionality allowing for better fitting to training data. Looking at the test results, dimensionality can be reduced by a large amount (e.g. to 12 dimensions) without affecting generalisation. The genre features, adding a large number of feature dimensions, may be omitted in order for a better trade-off of model complexity and performance, at a loss of performance of 2-3% which resembles the baseline of features including genre information.

In this thesis, the best results of almost 74% were achieved with our novel high-dimensional RBM features, providing an alternative solution for research question *rq:1*. Using high-dimensional feature spaces the RBM method allows for competitive results even with primitive training using gradient ascent, but lacks potential for analysis of musicological meaning in features. As the features were trained using the whole dataset, this gain might be partially caused by transductive learning

and it is left to future work to distinguish the gains of feature representation and transductive learning using high-performance computing facilities.

Comparing our new sampling method to standard sampling in Section 8.5, we found that the latter, ignoring clip repetition during sampling of cross-validation sets result in bias of up to 5% of increased performance for algorithms like MLR. These algorithms are preferable for transductive learning scenarios, for instance when working with a fixed music dataset where only similarity data is added. This investigation of *rq:3* can inform a better comparison of future experiments with relative similarity data.

WDMLR, our new weighted adaptation of DMLR, proved successful in predicting weighting data of MagnaTagATune on the training data, although with little generalisation success. Comparing results of other methods for weighted learning (our WMLR and existing WSVM), we concluded in Section 8.6.1 that different types of weighted data, reflecting both number of votes and vote uncertainty, are needed for further analysis.

We finalised our experiments with Section 8.7, presenting an application of our new W_0 -RITML method to culture-aware similarity modelling: Using four different similarity datasets divided by country, we introduced the method of transfer learning with similarity models. In this case, a generic “European Similarity” model learnt using RITML was fine-tuned to a specific-country via a second learning step. RITML was created using our model transfer methods with the absolute-similarity data ITML algorithm. These experiments only recently became possible through the new similarity data collected via the Spot the Odd Song Out game, and our method achieved a boost in performance promising better similarity learning when compared to earlier results achieved on MagnaTagATune.

9.5 Main Contributions

In this thesis we addressed learning music similarity measures from relative user data. We discussed existing research relevant to relative similarity learning, present-

ted available data and analysis as well as new and existing methods for modelling music similarity. The main methodical contributions are as follows:

- We derive audio- and tag-based features from the free online APIs The Echo Nest and Last.fm for large datasets (Chapter 5)
- Similarity graph analysis including building and pruning of similarity relation graphs from odd-one-out experiments (Section 3.1.2)
- A thorough analysis of the MagnaTagATune dataset using these methods Section 3.2
- A framework for collecting music annotations via the web and social networks and a game for collecting music similarity data annotated with user attributes (Chapter 4)
- The CASimIR similarity dataset, and a country-annotated dataset derived from it (Chapter 4)
- A general and extensible framework for training and evaluating music similarity models on large scale databases (Chapter 7)
- A number of new and adapted methods for learning from relative data (Chapter 6):
 - A methodical framework for relative similarity learning, allowing for integration of absolute similarity learners (Section 6.3)
 - The WMLR/WDMLR method for learning from weighted relative similarity data (Section 6.2.3)
 - A new approach of using RDNN for similarity learning (Section 6.3.7)
 - The RITML method for learning weighted relative similarity data, and W_0 -RITML for transfer learning with similarity models (Section 6.3.6.1)
- The *inductive sampling* method for unbiased sampling of relative similarity data for cross-validation (Section 8.1)
- A PCA-based evaluation of influence of feature information with constant dimensionality (Section 8.4.1)
- An approach for comparative analysis of culture-based similarity models, using transfer learning with RITML (Section 8.7)

Our evaluation of the discussed modelling strategies resulted in the following central experimental findings:

- Learning of metrics based on relative similarity data from users is possible with the tested features and algorithms. The performance on unseen test data can be significantly improved, depending on the application, the choice of algorithm, and features used.
- Adaptive models can achieve statistically significant improvements of more than 3% over a standard Euclidean metric, yielding an accuracy of almost 70% on test constraints.
- Mahalanobis metrics, and often weighted Euclidean metrics, are sufficiently flexible to model similarity relations in the given data, no gains can yet be found through the more flexible RDNN model.
- For SVM learning on the given dataset, chroma features are least effective, and combinations of different feature types are most effective, independent of dimensionality reduction and clustering versus averaging of timbre and chroma data.
- The best performance of almost 74% fulfilled constraints on MagnaTagATune is delivered by SVM with RBM transformed features.
- Our new RDNN algorithm trains a neural net to relative constraints works but generalisation performance is yet to be improved.
- The test performance of all algorithms leaves considerable room for improvement, which we attribute mostly to the MagnaTagATune dataset used.
- The CASimIR data seems easier to learn than data from MagnaTagATune, when considering acoustic-only features.
- Using biased transductive sampling, the results are up to 5% higher.

We conclude this thesis with an outlook on the opportunities and challenges opened up by the presented data, methods and results.

9.6 Perspectives for Future Work

The methods presented in this thesis allow for the modelling of music similarity models to user data. In a broader sense, we have worked towards the development of more user-centred music retrieval systems, allowing personal or other contextual information to be considered for the systems' responses. In line with the

MIReS roadmap for music technology [83], we here would like to emphasize the potential of user-centred MIR research. In particular, we are interested in developing culture-aware similarity models. Opportunities exist in the combination of research on context-aware models such as listed by Schedl, Flexer and Urbano [77] with relative similarity or other user-reported data. Such models should not only allow for the involvement of cultural indicators of the media and user for similarity predictions, but moreover facilitate comparative research in similarity judgement and perception across cultures. Rather than implicitly following cultural bias in MIR systems, more of it could thereby be made explicit and tangible for interaction with the user.

The CASimIR framework for data collection via GWAPs delivers and enables the collection annotations with rich context information about the data providing participant. We have given an example of a culture-annotated dataset in Section 4.3.1.2, and hope the further data collection via Spot the Odd Song Out will add to this dataset. We also encourage other researchers to contribute to this dataset with different annotation collection modules. In general, the development of more GWAPs and collection of user behaviour data in games for music – within the boundaries of research ethics – will not only help investigate the “glass ceiling” in MIR but also facilitate interdisciplinary research with social sciences, cultural studies, psychology and other disciplines. A comparison of information gained and sources of bias in games versus traditional surveys is yet to be performed and would be very helpful in deciding for appropriate data collections strategies.

The comparison of similarity models for geographic regions exemplifies a first comparative analysis of models, with conclusions relating to the participants’ regions. We particularly are interested how our novel method of W_0 -RITML transfer learning could be further exploited in this scenario. Design of culture-aware models as suggested in Section 4.3.1 will require but also feed back to the understanding of two relationships: the relationship of user attributes to cultural spaces and the relationship of cultures to music (similarity) perception.

For the training of similarity models, our results consistently support the interpretation that the learning performance is limited by the size and the quality of the dataset. Experiments on feature influence showed that a more diverse feature

set promises to increase performance. Already without genre tag features, which are not yet available for CASimIR, the models outperform acoustic-only results from MagnaTagATune (see Table 8.2). Such features can be derived from MusicBrainz data linked to CASimIR. Future research can investigate the data quality in CASimIR, which aims to be more balanced in terms of clip linkage than MagnaTagATune, and extends on presented methods of comparing different similarity datasets.

Apart from more advanced feature design, the different similarity models and training methods presented also encourage further research at the basis of learning similarity from relative data: The representation of clip pairs via their facet differences used in this thesis points out the opportunity of specialised facet difference measures fitted to the underlying feature types. The feature specific measures in Stober and Nürnberger [90] can be extended by further processing of facet difference vectors by for example convolution, the normalised compression distance by Bello [7], or the Kullback–Leibler divergence for tag features.

For the presented training algorithms, MLR as well as RDNN would benefit from a model-selection strategy during experiments, using a validation set to deal with the choice of hyperparameters such as the regularisation trade-off c . This is also an opportunity to optimise the probabilistic RBM feature transformation which proved successful in our experiments. In Section 8.4.3 we tried several instances of probabilistic feature learning, selecting models based on the RBM training set performance. The results here can be validated and possibly improved by using only training sets for model training and a validation set for model selection. Furthermore, using feedback of the final similarity model performance during training, for instance by fine-tuning the RBM via backpropagation, would provide a more unified and potentially more successful training strategy. For the RDNN network, we expect further parameter tuning, and the test of alternative network architectures can allow for achieving competitive results. Transfer learning might also be possible using pre-training of the neural nets. Furthermore, when learning from similarity data allowing for directional similarity evaluation, the ability of the neural net to model asymmetric similarity perception, as suggested by Tversky [94] will become relevant.

Many algorithms that can be adapted to learning from relative data via the method in Section 6.3, such as generic regression or GBRank regression trees, also allow for learning from asymmetric similarity data. We here only evaluate the RITML method, which is restricted to learning a pseudometric, but on the other hand allows us to introduce transfer learning. There is potential in the context information in the similarity triplet which could be exploited more explicitly. It will be interesting to compare the properties and performance of similarity learning paradigms using absolute similarity data with the methods developed here for relative data, including data collection, analysis and user feedback on the final models.

The regression-based and RDNN methods presented in this thesis might allow for usage of similarity data without advance filtering as performed in Section 3.1.2, given we collect similarity data with more informative weighting information. Also, the limitations of assuming symmetric similarity can be overcome using the regression method exemplified in Section 6.3 and the RDNN approach and evaluated using the presented evaluation strategies. Although the CASimIR dataset does contain the sequence of listening in clips, we suggest to collect similarity data using a new module controlling the order of clips for instance by means of presentation.

The application of the presented methods for model training and analysis to big datasets will be pursued in the AHRC-funded project “An Integrated Audio-Symbolic Model of Music Similarity (ASyMMuS)”. Based on an infrastructure for music research on big data, currently developed by the “Digital Music Lab¹” project, similarity models combining acoustic and symbolic data will be developed. The goal is to provide musicologists and others with tools for large-scale collection analysis based on multi-modal concepts of music similarity.

In the 1960’s, Lomax [52] compared cultures on the basis of their songs and music in a large music ethnographic project. Culture-aware MIR systems and models we have today can support comparisons based on further musical attributes such as style, preference and similarity perception, on a much larger scale. The recent advent of more powerful computing techniques requiring less human pre-processing of observations, together with the strong impact of open sourcing of research code

¹<http://dml.city.ac.uk>

and data will enable the development of more user-based systems. This data-centred approach should not lead to a decline of involvement of expert-knowledge in research on music. On the contrary, we hope that this thesis encourages more interdisciplinary work of MIR with disciplines such as musicology, cultural studies, sociology, psychology and others that can help model and compare the structure of cultural knowledge and its integration in music information systems.

References

- [1] Aho, Alfred V., Garey, Michael R. and Ullman, Jeffrey D. "The Transitive Reduction of a Directed Graph". In: *SIAM Journal on Computing* 1.2 (1972), pp. 131–137.
- [2] Akkermans, Vincent, Font, Frederic, Jordi, Funollet, De Jong, Bram, Roma, Gerard, Togias, Stella and Serra, Xavier. "Freesound 2: An Improved Platform for Sharing Audio Clips". In: *Proceedings of ISMIR 2011*. Miami, Florida, USA, 2011.
- [3] Allan, Hamish, Müllensiefen, Daniel and Wiggins, Geraint. "Methodological considerations in studies of musical similarity". In: *8th International Conference on Music Information Retrieval*. 2007, pp. 473–478.
- [4] Bade, Korinna, Garbers, Jörg, Stober, Sebastian, Wiering, Frans and Nürnberger, Andreas. "Supporting Folk-Song Research by Automatic Metric Learning and Ranking". In: *Proceedings of ISMIR 2009*. Kobe, Japan, Oct. 2009, pp. 741–746.
- [5] Barrington, Luke, O'Malley, Damien, Turnbull, Douglas and Lanckriet, Gert R. G. "User-centered design of a social game to tag music". In: *Proceedings of the ACM SIGKDD Workshop on Human Computation*. HCOMP '09. Paris, France: ACM, 2009, pp. 7–10.
- [6] Baumann, Stephan and Halloran, John. "An Ecological Approach to Multimodal Subjective Music Similarity perception". In: *Proceedings of the 1st Conference on Interdisciplinary Musicology (CIM)*. 2004.
- [7] Bello, Juan Pablo. "Grouping Recorded Music by Structural Similarity". In: *Proceedings of ISMIR 2009*. 2009.
- [8] Bello, Juan Pablo. "Measuring Structural Similarity in Music". In: *IEEE Transactions on Audio, Speech & Language Processing* 19.7 (2011), pp. 2013–2025.

-
- [9] Berenzweig, Adam, Logan, Beth, Ellis, Daniel P. W. and Whitman, Brian P. W. "A Large-Scale Evaluation of Acoustic and Subjective Music-Similarity Measures". In: *Computer Music Journal* 28.2 (June 2004), pp. 63–76.
- [10] Bernard, H. Russell. *Research Methods in Anthropology, 5th edition*. AltaMira Press, 2011.
- [11] Bogdanov, Dmitry. "From Music Similarity to Music Recommendation: Computational Approaches Based on Audio Features and Metadata". PhD thesis. Barcelona, Spain: Universitat Pompeu Fabra, 2013.
- [12] Bogdanov, Dmitry, Serrà, Joan, Wack, Nicolas and Herrera, Perfecto. "From Low-level to High-level: Comparative Study of Music Similarity Measures". In: *IEEE International Symposium on Multimedia. Workshop on Advances in Music Information Research (AdMIRe)*. 2009.
- [13] Bosma, Martijn, Veltkamp, Remco C. and Wiering, Frans. "Muugle: A Modular Music Information Retrieval Framework". In: *Proceedings of ISMIR 2006*. 2006.
- [14] Braun, Heinrich. *Neuronale Netze - Optimierung durch Lernen und Evolution*. Springer, 1997, pp. I–XI, 1–279.
- [15] Braun, Heinrich, Feulner, Johannes and Ullrich, Volker. "Learning Strategies for Solving the Planning Problem using Backpropagation." In: *Proceedings of NEURO-Nimes 91, 4th International Conference on Neural Networks and their Applications*. 1991.
- [16] Casey, M.A., Veltkamp, R., Goto, M., Leman, M., Rhodes, C. and Slaney, M. "Content-Based Music Information Retrieval: Current Directions and Future Challenges". In: *Proceedings of the IEEE* 96.4 (2008), pp. 668–696.
- [17] Celma, Oscar. "Music Recommendation and Discovery in the Long Tail". PhD thesis. Barcelona: Universitat Pompeu Fabra, 2008.
- [18] Cheng, Weiwei and Hüllermeier, Eyke. "Learning Similarity Functions from Qualitative Feedback". In: *Proceedings of ECCBR'08*. 2008.
- [19] Dash, Manoranjan and Liu, Huan. "Feature Selection for Classification". In: *Intelligent Data Analysis* 1.1-4 (1997), pp. 131–156.
- [20] Davis, Jason V., Kulis, B., Jain, Prateek, Sra, Suvrit and Dhillon, Inderjit S. "Information-theoretic Metric Learning". In: *Proceedings of the 24th international conference on Machine learning. ICML '07*. Corvallis, Oregon: ACM, 2007, pp. 209–216.

References

- [21] Dieleman, Sander, Brakel, Philémon and Schrauwen, Benjamin. "Audio-based Music Classification with a Pretrained Convolutional Network". In: *Proceedings of the 12th International Society for Music Information Retrieval Conference*. Miami (Florida), USA, 2011, pp. 669–674.
- [22] Ellis, Daniel P. W. and Whitman, Brian. "The Quest for Ground Truth in Musical Artist Similarity". In: *Proceedings of ISMIR 2002*. 2002, pp. 170–177.
- [23] Ferrer, Rafael and Eerola, Tuomas. "Timbral Qualities of Semantic Structures of Music". In: *Proceedings of the 11th International Society for Music*. Utrecht, Netherlands, 2010, pp. 571–576.
- [24] Foster, Peter, Mauch, Matthias and Dixon, Simon. "Sequential Complexity as a Descriptor for Musical Similarity". In: *ArXiv e-prints* (Feb. 2014).
- [25] Friberg, Anders and Hedblad, Anton. "A Comparison of Perceptual Ratings and Computed Audio Features". In: *Proceedings of SMC*. 2011.
- [26] Friedman, Jerome H. "Greedy Function Approximation: A Gradient Boosting Machine". In: *Annals of Statistics* 29 (2000), pp. 1189–1232.
- [27] Frome, Andrea, Singer, Yoram, Sha, Fei and Malik, Jitendra. "Learning Globally-Consistent Local Distance Functions for Shape-Based Image Retrieval and Classification". In: *ICCV*. 2007, pp. 1–8.
- [28] Gamberman, Alex, V., Vladimir and Vapnik, Vladimir. "Learning by Transduction". In: *In Uncertainty in Artificial Intelligence*. Morgan Kaufmann, 1998, pp. 148–155.
- [29] Gentner, Dedre and Markman, A.B. "Structure mapping in analogy and similarity". In: *American Psychologist* 52.1 (1997), pp. 45–56.
- [30] Goto, Masataka, Yoshii, Kazuyoshi, Fujihara, Hiromasa, Mauch, Matthias and Nakano, Tomoyasu. "Songle: A Web Service for Active Music Listening Improved by User Contributions". In: *Proceedings of the 12th International Society for Music Information Retrieval Conference*. Miami (Florida), USA, 2011, pp. 311–316.
- [31] Hadsell, Raia, Chopra, Sumit and Lecun, Yann. "Dimensionality reduction by learning an invariant mapping". In: *In Proc. Computer Vision and Pattern Recognition Conference (CVPR'06)*. IEEE Press, 2006.
- [32] Hahn, Ulrike and Ramscar, Michael. *Similarity and Categorization*. Oxford: Oxford University Press, 2001.
- [33] Hamel, Philippe and Eck, Douglas. "Learning Features from Music Audio with Deep Belief Networks". In: *Proceedings of the 11th Interna-*

- tional Society for Music Information Retrieval Conference*. Utrecht, The Netherlands, 2010, pp. 339–344.
- [34] Hauger, David, Schedl, Markus, Košir, Andrej and Tkalčič, Marko. “The Million Musical Tweets Dataset: What Can We Learn From Microblogs”. In: *Proceedings of the 14th International Society for Music Information Retrieval Conference (ISMIR 2013)*. Curitiba, Brazil, 2013.
- [35] Hinton, Geoffrey E. “Training Products of Experts by Minimizing Contrastive Divergence”. In: *Neural Comput.* 14.8 (Aug. 2002), pp. 1771–1800.
- [36] Hinton, Geoffrey E., Osindero, Simon and Teh, Yee-Whye. “A Fast Learning Algorithm for Deep Belief Nets”. In: *Neural Comput.* 18.7 (July 2006), 1527–1554.
- [37] Hörnel, Dominik. “CHORDNET: Learning and Producing Voice Leading with Neural Networks and Dynamic Programming”. In: *Journal of New Music Research* 33.4 (2004), pp. 387–397.
- [38] Hornik, Kurt, Stinchcombe, Maxwell and White, Halbert. “Multilayer Feedforward Networks are Universal Approximators”. In: *Neural Networks* 2.5 (1989), pp. 359–366.
- [39] Jehan, Tristan. “Creating Music by Listening”. PhD thesis. MA, USA: Massachusetts Institute of Technology, 2005.
- [40] Joachims, Thorsten, Finley, Thomas and Yu, Chun-Nam J. “Cutting-plane training of structural SVMs”. In: *Machine Learning* 77 (1 2009), pp. 27–59.
- [41] Kaminskas, Marius, Ricci, Francesco and Schedl, Markus. “Location-aware Music Recommendation Using Auto-Tagging and Hybrid Matching”. In: *Proceedings of the 7th ACM Conference on Recommender Systems (RecSys 2013)*. Hong Kong, China, 2013.
- [42] Karp, Richard M. “Reducibility Among Combinatorial Problems”. In: *Complexity of Computer Computations*. Ed. by Miller, R. E. and Thatcher, J. W. Plenum Press, 1972, pp. 85–103.
- [43] Kelly, George A. *The Psychology of Personal Constructs*. New York: WW Norton, 1955.
- [44] Khadkevich, Maksim and Omologo, Maurizio. “Large-Scale Cover Song Identification Using Chord Profiles.” In: *Proceedings of ISMIR*. 2013, pp. 233–238.

References

- [45] Law, Edith and Von Ahn, Luis. "Input-agreement: A New Mechanism for Collecting Data Using Human Computation Games". In: *Proceedings of CHI*. ACM Press, 2009.
- [46] Law, Edith, Dalton, Conner, Merrill, Nick, Young, Albert and Gajos, Krzysztof Z. "Curio: A Platform for Supporting Mixed-Expertise Crowdsourcing". In: *Proceedings of HCOMP 2013*. To appear. AAAI Press, 2013.
- [47] Le, Quoc, Ranzato, Marc'Aurelio, Monga, Rajat, Devin, Matthieu, Chen, Kai, Corrado, Greg, Dean, Jeff and Ng, Andrew. "Building high-level features using large scale unsupervised learning". In: *Proceedings of ICML 2012*. 2012.
- [48] Leblanc, Albert. "An Interactive Theory of Music Preference". In: *Journal of Music Therapy* 19.1 (1982), pp. 28–45.
- [49] Lee, Honglak, Grosse, Roger, Ranganath, Rajesh and Ng, Andrew Y. "Convolutional Deep Belief Networks for Scalable Unsupervised Learning of Hierarchical Representations". In: *Proceedings of ICML 2009*. ICML '09. New York, NY, USA: ACM, 2009, 609–616.
- [50] Lerdahl, Fred and Jackendoff, Ray. *A generative theory of tonal music*. Cambridge, MA: The MIT Press, 1983.
- [51] Lim, Daryl, Mcfee, Brian and Lanckriet, Gert R. "Robust Structural Metric Learning". In: *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*. Ed. by Dasgupta, Sanjoy and Mcallester, David. Vol. 28. JMLR Workshop and Conference Proceedings, 2013, pp. 615–623.
- [52] Lomax, Alan. "Song Structure and Social Structure". In: *Ethnology* (1962), pp. 425–451.
- [53] MacCallum, Robert M., Mauch, Matthias, Burt, Austin and Leroi, Armand M. "Evolution of Music by Public Choice". In: *Proceedings of the National Academy of Sciences* (2012), pp. 12081–12086.
- [54] Mahalanobis, Prasanta C. "On the Generalised Distance in Statistics". In: *Proceedings of the National Institute of Sciences of India* 2. MIT Press, 1936, 49–55.
- [55] McDermott, Josh H. "Auditory preferences and aesthetics: Music, voices, and everyday sounds". In: *Neuroscience of Preference and Choice*. Ed. by Sharot and Dolan. 2011.

-
- [56] McFee, Brian, Barrington, L. and Lanckriet, Gert R. G. "Learning Similarity from Collaborative Filters". In: *Proceedings of ISMIR 2010*. Utrecht, 2010, pp. 345–350.
- [57] Mcfee, Brian and Lanckriet, Gert R. G. "Heterogeneous Embedding for Subjective Artist Similarity". In: *Proceedings of the 10th International Society for Music Information Retrieval Conference*. Kobe, Japan, 2009, pp. 513–518.
- [58] McFee, Brian and Lanckriet, Gert R. G. "Heterogeneous Embedding for Subjective Artist Similarity". In: *Proceedings of ISMIR 2009*. 2009.
- [59] McFee, Brian and Lanckriet, Gert R. G. "Hypergraph models of playlist dialects". In: *Proceedings of ISMIR 2012*. 2012.
- [60] Mcfee, Brian and Lanckriet, Gert R. G. "Metric Learning to Rank". In: *Proceedings of the 27th annual International Conference on Machine Learning (ICML)*. 2010.
- [61] McFee, Brian and Lanckriet, Gert R. G. "Partial Order Embedding with Multiple Kernels". In: *Proceedings of the 26th International Conference on Machine Learning (ICML'09)*. 2009, pp. 721–728.
- [62] Medin, DOUGLAS L., Lynch, Elisabeth B., Coley, John D. and Atran, Scott. "Categorization and Reasoning among Tree Experts: Do All Roads Lead to Rome?" In: *Cognitive Psychology* 32 (1997), pp. 49–96.
- [63] Mitrovic, Dalibor, Zeppelzauer, Matthias and Breiteneder, Christian. "Features for Content-Based Audio Retrieval". In: *Advances in Computers Volume 78 Improving the Web*. Elsevier, 2010, pp. 71–150.
- [64] Müller, Meinard, Ellis, Daniel P. W., Klapuri, Anssi and Richard, Gaël. "Signal Processing for Music Analysis". In: *Selected Topics in Signal Processing, IEEE Journal of* 5.6 (2011), pp. 1088–1110.
- [65] Musil, Jason, El-Nusairi, Budr and Müllensiefen, Daniel. "Perceptual dimensions of short audio clips and corresponding timbre features". In: *Proceedings of the 9th International Symposium on Computer Music Modelling and Retrieval (CMMR 2012)*. 2012.
- [66] Nam, Juhan, Ngiam, Jiquan, Lee, Honglak and Slaney, Malcolm. "A Classification-based Polyphonic Piano Transcription Approach Using Learned Feature Representations". In: *Proceedings of the 12th International Society for Music Information Retrieval Conference*. Miami (Florida), USA, 2011, pp. 175–180.

References

- [67] Nam, Juhan, Herrera, Jorge, Slaney, Malcolm and Smith, Julius. "Learning Sparse Feature Representations for Music Annotation and Retrieval". In: *Proceedings of the 13th International Society for Music Information Retrieval Conference*. Porto, Portugal, 2012.
- [68] Novello, Alberto, Mckinney, Martin F. and Kohlrausch, Armin. "Perceptual evaluation of music similarity". In: *Proceedings of ISMIR 2006*. 2006.
- [69] Page, Kevin, Fields, Ben, De Roure, David, Crawford, Tim and Downie, J. Stephen. "Reuse, Remix, Repeat: the Workflows of MIR". In: *Proceedings of ISMIR 2012*. Porto, Portugal, 2012.
- [70] Park, Han-Saem, Yoo, Ji-Oh and Cho, Sung-Bae. "A Context-Aware Music Recommendation System Using Fuzzy Bayesian Networks with Utility Theory". In: *Fuzzy Systems and Knowledge Discovery*. Ed. by Wang, Lipo, Jiao, Licheng, Shi, Guanming, Li, Xue and Liu, Jing. Vol. 4223. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2006, pp. 970–979.
- [71] Paulus, Jouni, Müller, Meinard and Klapuri, Anssi. "Audio-based Music Structure Analysis". In: *Proceedings of the Int. Society for Music Information Retrieval Conference*. 2010.
- [72] Peeters, Geoffroy. *A Large Set of Audio Features for Sound Description (Similarity and Classification) in the CUIDADO Project*. Tech. rep. IRCAM, 2004.
- [73] Pickens, Jeremy. "A Survey of Feature Selection Techniques for Music Information Retrieval". In: *Proceedings of ISMIR 2001*. 2001.
- [74] Riedmiller, Martin and Braun, Heinrich. "A Direct Adaptive Method for Faster Backpropagation Learning: The RPROP algorithm". In: *Proceedings of the IEEE International Conference on Neural Networks*. San Francisco, CA, 1993, pp. 586–591.
- [75] Rolland, Pierre-Yves. "Adaptive User Modelling in a Content-Based Music Retrieval System". In: *Proceedings of ISMIR 2001*. 2001, pp. 27–30.
- [76] Salakhutdinov, Ruslan and Hinton, Geoffrey E. "Deep Boltzmann Machines". In: *Journal of Machine Learning Research - Proceedings Track 5* (2009), pp. 448–455.
- [77] Schedl, Markus, Flexer, Arthur and Urbano, Julián. "The Neglected User in Music Information Retrieval Research". In: *Journal of Intelligent Information Systems* 41 (3 2013), pp. 523–539.

-
- [78] Schedl, Markus, Hauger, David and Urbano, Julián. “Harvesting Microblogs for Contextual Music Similarity Estimation - A Co-occurrence-based Framework”. In: *Multimedia Systems* (2013).
- [79] Schlüter, Jan and Osendorfer, Christian. “Music Similarity Estimation with the Mean-Covariance Restricted Boltzmann Machine”. In: *Proceedings of ICMLA 2011*. Honolulu, USA, 2011.
- [80] Schmidt, Erik, Scott, Jeffrey and Kim, Youngmoo. “Feature Learning in Dynamic Environments: Modeling the Acoustic Structure of Musical Emotion”. In: *Proceedings of the 13th International Society for Music Information Retrieval Conference*. Porto, Portugal, 2012.
- [81] Schultz, M. and Joachims, T. “Learning a Distance Metric from Relative Comparisons”. In: *Advances in Neural Information Processing Systems (NIPS)*. MIT Press, 2003.
- [82] Serra, Xavier. “Data Gathering for a Culture Specific Approach in MIR”. In: *Proceedings of the 21st World Wide Web Conference, WWW 2012, Lyon*. 2012, pp. 867–868.
- [83] Serra, Xavier, Magas, Michela, Benetos, Emmanouil, Chudy, Magdalena, Dixon, Simon, Flexer, Arthur, Gómez, Emilia, Gouyon, F., Herrera, Perfecto, Jordà, S., Paytuvi, Oscar, Peeters, Geoffroy, Schlüter, Jan, Vinet, Hugo and Widmer, Gerraint. *Roadmap for Music Information ReSearch*. 2013.
- [84] Slaney, Malcolm, Weinberger, Kilian Q. and White, William. “Learning a Metric for Music Similarity.” In: *Proceedings of ISMIR 2008*. Ed. by Bello, Juan Pablo, Chew, Elaine and Turnbull, Douglas. 28th Dec. 2009, pp. 313–318.
- [85] Slaney, Malcolm and White, William. “Similarity Based on Rating Data”. In: *Proceedings of ISMIR 2007*. 2007, pp. 479–484.
- [86] Slobin, Mark. “Micromusics of the West: A Comparative Approach”. In: *Ethnomusicology* 36.1 (1992), pp. 1–87.
- [87] Smolensky, Paul. “Information Processing in Dynamical Systems: Foundations of Harmony Theory”. In: *Rumelhart, D. E. and McClelland, J. L., editors, Parallel Distributed Processing: Volume 1: Foundations*. MIT Press, Cambridge, 1986, pp. 194–281.
- [88] Sotiropoulos, Dionysios, Lampropoulos, Aristomenis and Tsihrantzis, George. “MUSIPER: a system for modeling music similarity perception based

References

- on objective feature subset selection". English. In: *User Modeling and User-Adapted Interaction* 18.4 (2008), pp. 315–348.
- [89] Stober, Sebastian. "Adaptive Methods for User-Centered Organization of Music Collections". published by Dr. Hut Verlag, ISBN 978-3-8439-0229-8. PhD thesis. Magdeburg, Germany: Otto-von-Guericke-University, 2011.
- [90] Stober, Sebastian and Nürnberger, Andreas. "An Experimental Comparison of Similarity Adaptation Approaches". In: *Proceedings of AMR 2011*. Barcelona, Spain, 2011.
- [91] Stober, Sebastian and Nürnberger, Andreas. "Similarity Adaptation in an Exploratory Retrieval Scenario". In: *Proceedings of AMR 2010*. Linz, Austria, 2010.
- [92] Tarjan, Robert. "Depth-First Search and Linear Graph Algorithms". In: *SIAM Journal on Computing* 1.2 (1972), pp. 146–160.
- [93] Tsochantaridis, Ioannis, Hofmann, Thomas, Joachims, Thorsten and Al-tun, Yasemin. "Support Vector Machine Learning for Interdependent and Structured Output Spaces". In: *Proceedings of the International Conference on Machine Learning (ICML)*. 2004.
- [94] Tversky, Amos. "Features of Similarity". In: *Psychological Review* 84 (1977), pp. 327–352.
- [95] Typke, Rainer, Hoed, Marc den, Nooijer, Justin de, Wiering, Frans and Veltkamp, Remco C. "A Ground Truth For Half A Million Musical Incipits." In: *JDIM* 3.1 (2005), pp. 34–38.
- [96] Vignoli, Fabio and Pauws, Steffen. "A Music Retrieval System Based on User Driven Similarity and Its Evaluation". In: *Proceedings of the 6th International Conference on Music Information Retrieval*. London, UK, 2005, pp. 272–279.
- [97] Vincent, Pascal, Larochelle, Hugo, Lajoie, Isabelle, Bengio, Yoshua and Manzagol, Pierre-Antoine. "Stacked Denoising Autoencoders: Learning Useful Representations in a Deep Network with a Local Denoising Criterion". In: *J. Mach. Learn. Res.* 11 (Dec. 2010), pp. 3371–3408.
- [98] Von Ahn, Luis and Dabbish, Laura. "Designing games with a purpose". In: *Commun. ACM* 51.8 (Aug. 2008), pp. 58–67.
- [99] Von Ahn, Luis, Kedia, Mihir and Blum, Manuel. "Verbosity: a game for collecting common-sense facts". In: *CHI '06: Proceedings of the SIGCHI conference on Human Factors in computing systems*. Montreal, Quebec, Canada: ACM Press, 2006, pp. 75–78.

-
- [100] Wang, Ju-Chiang, Lee, Hung-Shin, Wang, Hsin-Min and Jeng, Shyh-Kang. "Learning the Similarity of Audio Music in Bag-of-Frames Representation from Tagged Music Data". In: *Proceedings of the 12th International Society for Music Information Retrieval Conference*. Miami (Florida), USA, 2011, pp. 85–90.
- [101] Weinberger, K. Q. and Saul, L. K. "Distance Metric Learning for Large Margin Nearest Neighbor Classification". In: *J. Mach. Learn. Res.* 10 (2009), pp. 207–244.
- [102] Weinberger, Kilian Q. and Saul, Lawrence K. "Distance Metric Learning for Large Margin Nearest Neighbor Classification". In: *The Journal of Machine Learning Research* 10 (2009), pp. 207–244.
- [103] West, Kris, Kumar, Amit, Shirk, Andrew, Zhu, Guojun, Downie, J. Stephen, Ehmann, Andreas F. and Bay, Mert. "The Networked Environment for Music Analysis (NEMA)." In: IEEE Computer Society, 2010, pp. 314–317.
- [104] Weyde, Tillman. *Lern- und wissensbasierte Analyse musikalischer Rhythmen: Konzeption, Entwicklung und Evaluation eines Neuro-Fuzzy-Systems für die Erkennung rhythmischer Strukturen*. Osnabrück: epOs Music, 2003.
- [105] Yang, Liu. *Distance Metric Learning: A Comprehensive Survey*. Tech. rep. 2006, pp. 1–51.
- [106] Zheng, Zhaohui, Chen, Keke, Sun, Gordon and Zha, Hongyuan. "A regression framework for learning ranking functions using relative relevance judgments". In: *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*. SIGIR '07. Amsterdam, The Netherlands: ACM, 2007, pp. 287–294.

10 Appendix

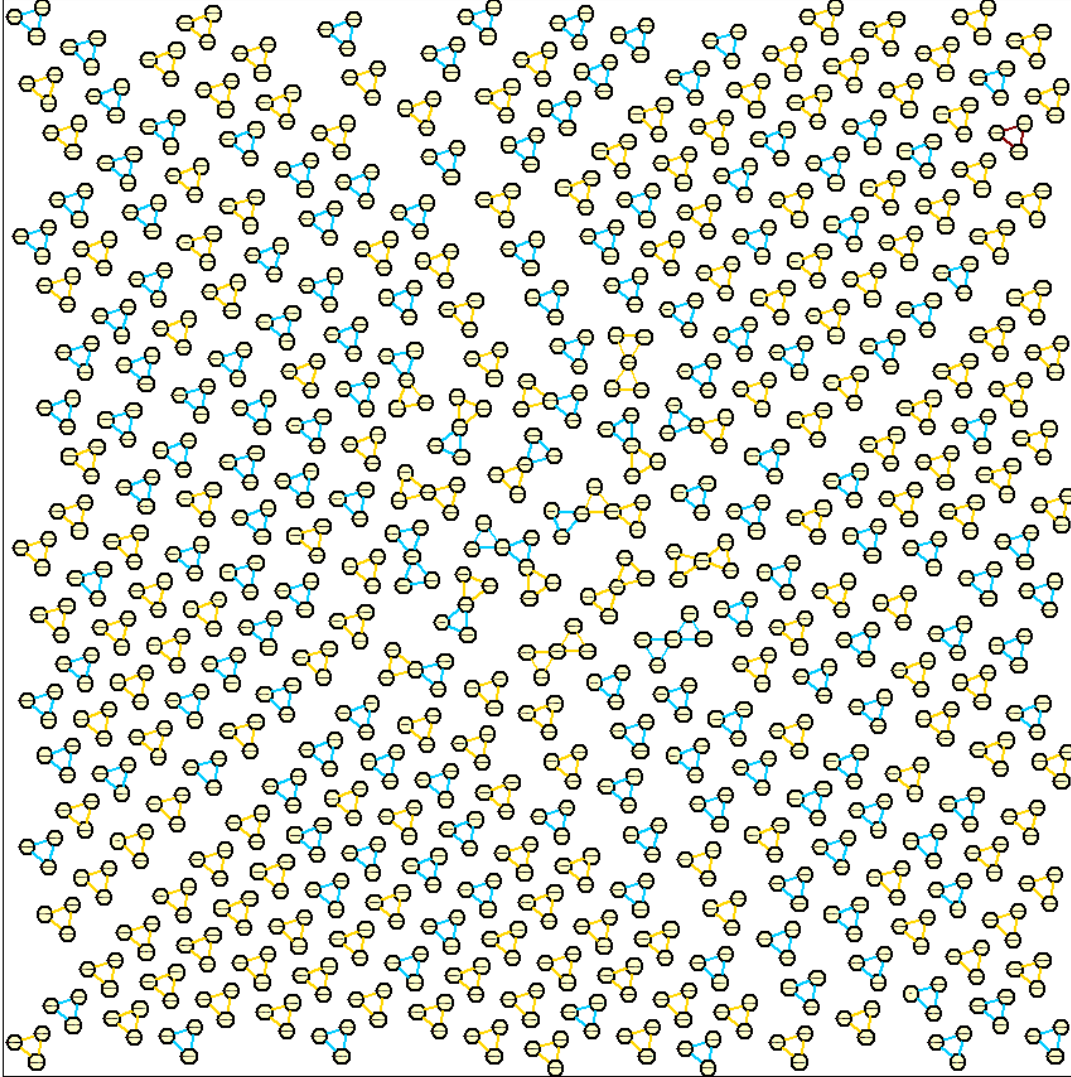


Figure 10.1: ClipComparedGraph (Section 7.2.1) of the MagnaTagATune dataset. Vertices represent clips, undirected edges reflect co-occurrence of the clips in at least one triplet. Question triplets are clearly distinguishable. Only few clips are linked/compared to more than two other clips. Colors indicate the number of permutations a triplet has been presented in (blue=1, yellow=2, red=3). The spatial arrangement minimises edge collisions.

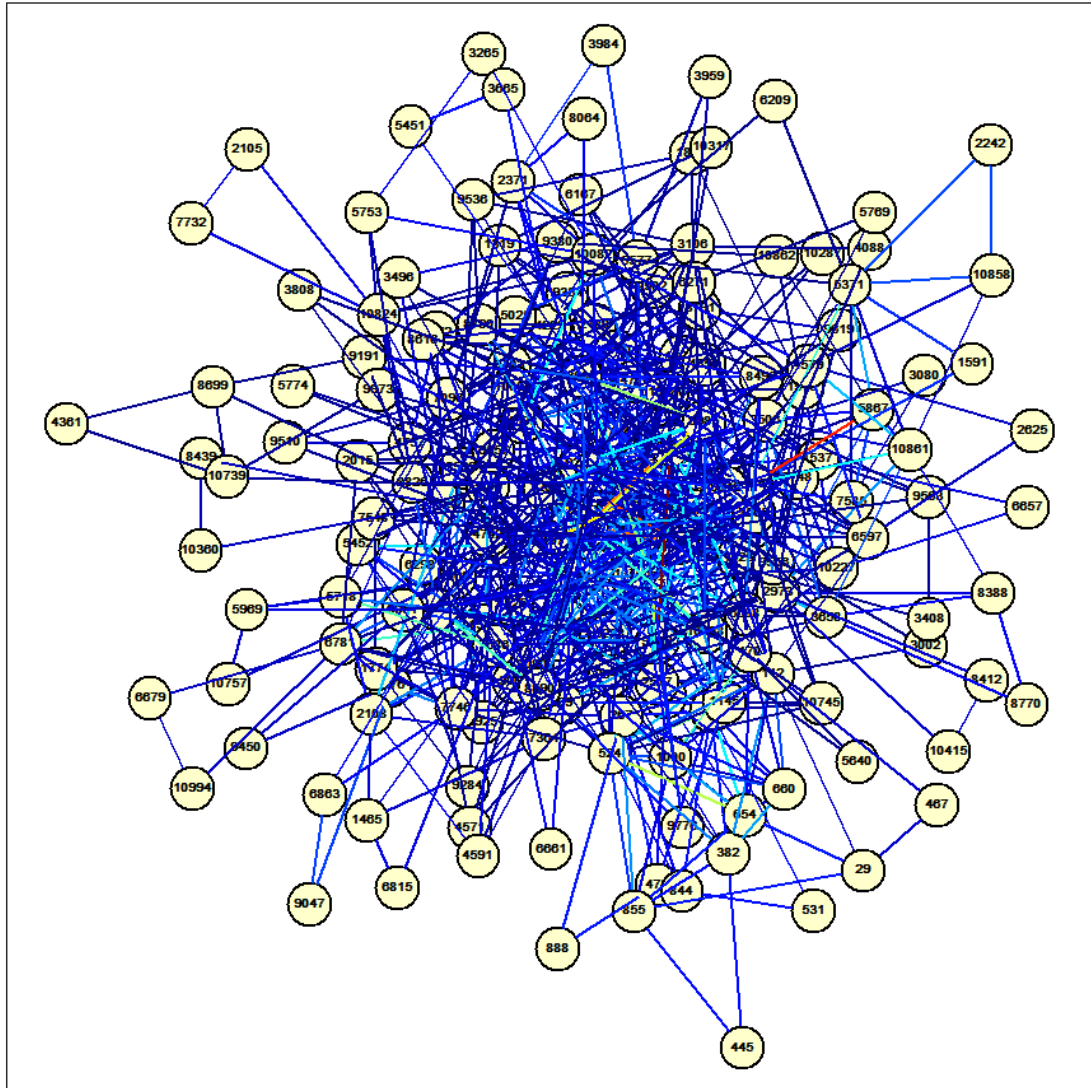


Figure 10.2: ClipComparedGraph (Section 7.2.1) of the current (01/05/2014) CASimIR dataset. Vertices represent clips, undirected edges reflect co-occurrence of the clips in at least one triplet. Less clips exist, but clips are strongly interlinked through questions. The spatial arrangement algorithm fails to unknot the many edges.

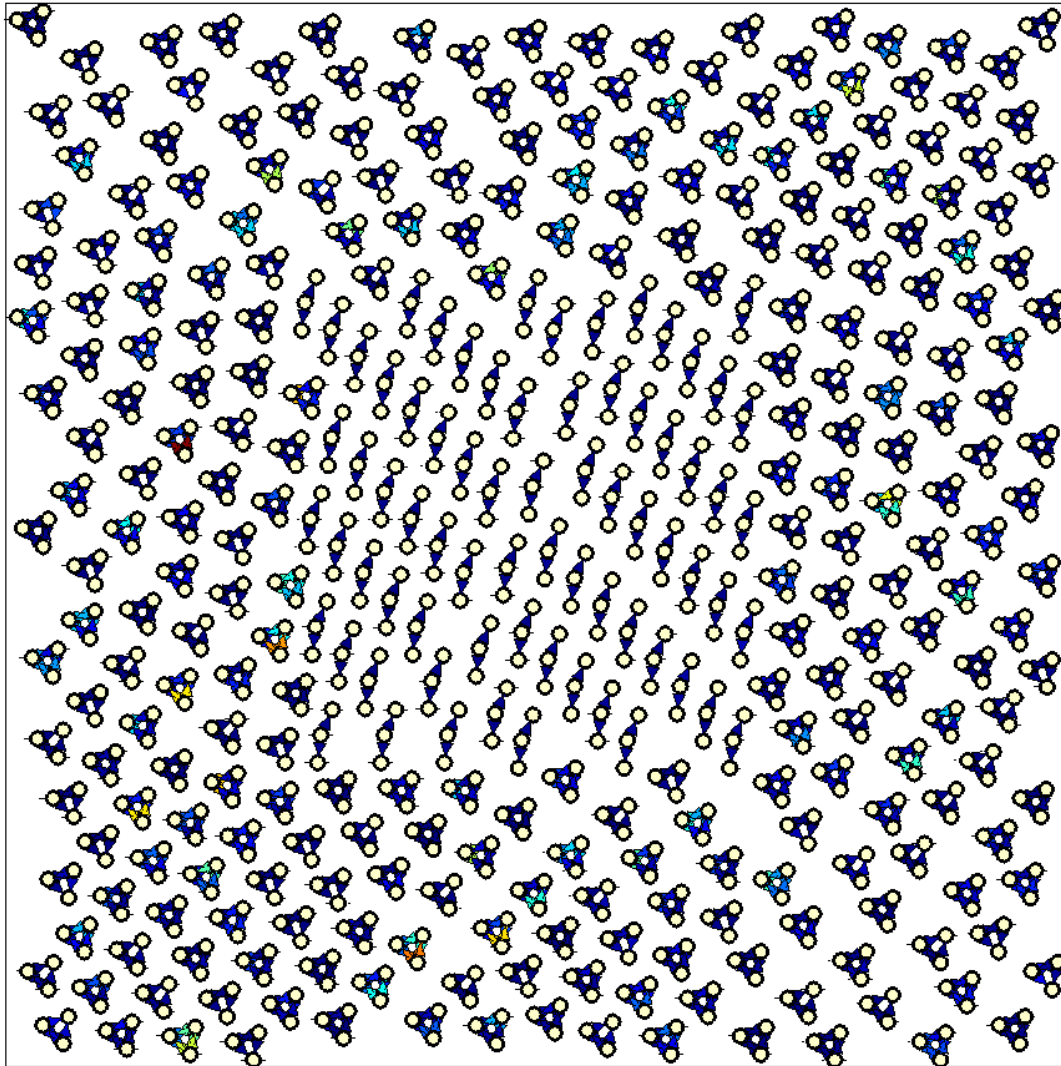


Figure 10.3: MagnaTagATune similarity graph (Sections 3.1 and 7.2.2) before removal of cycles. Vertices represent clip pairs, directed edges represent the relation “more similar than”. The question triplets are already completely separated. The triplets arranged along lines in the centre correspond to triplets with no inconsistent data. Lighter colours correspond to greater edge weights α . The spatial arrangement minimises edge collisions, but is not further related to data.

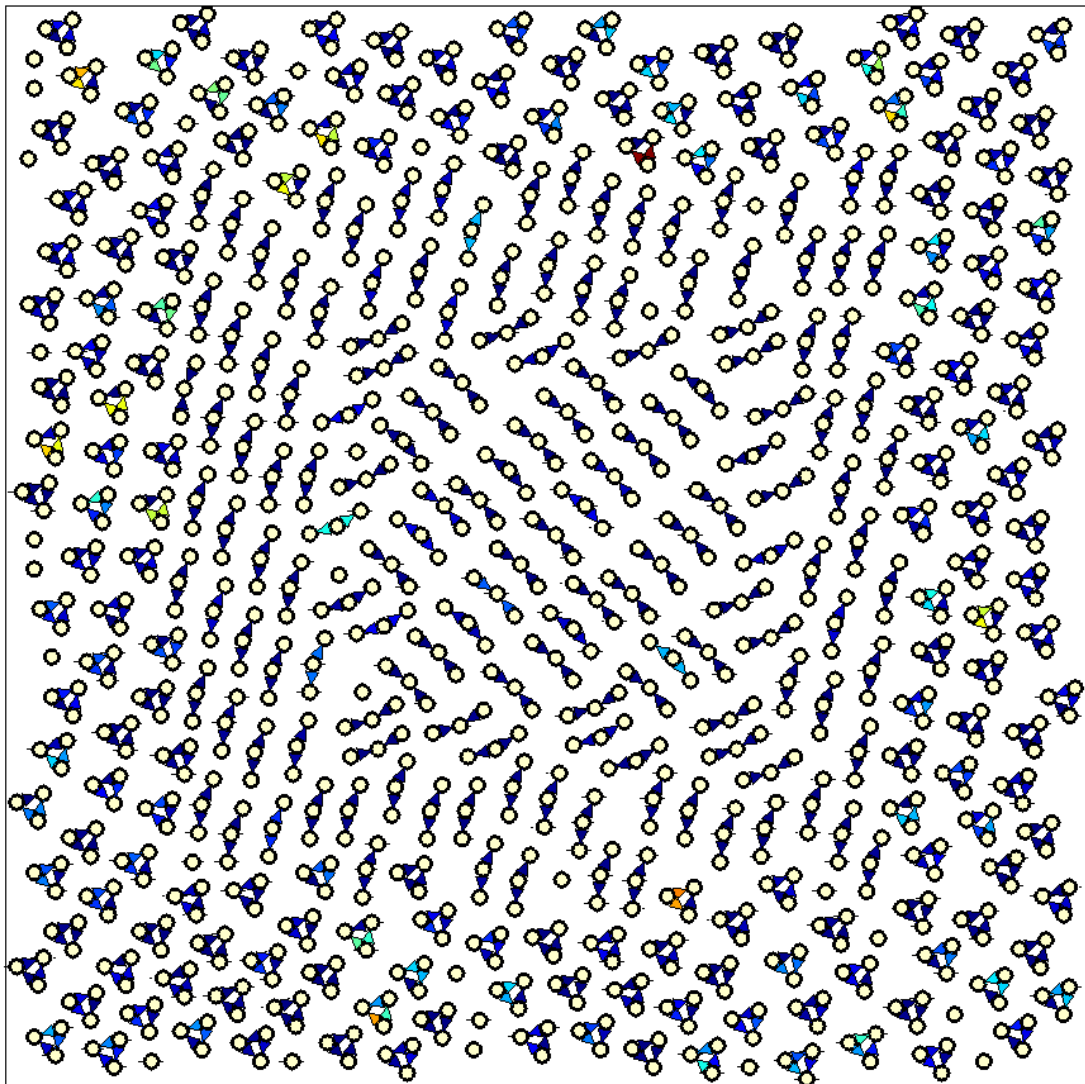


Figure 10.4: MagnaTagATune similarity graph (Sections 3.1 and 7.2.2) **after** removal of cycles. Vertices represent clip pairs, directed edges represent the relation “more similar than”. No inconsistent data is left and even more triplets have only two of three possible connections through edges. Lighter colours correspond to greater edge weights α .

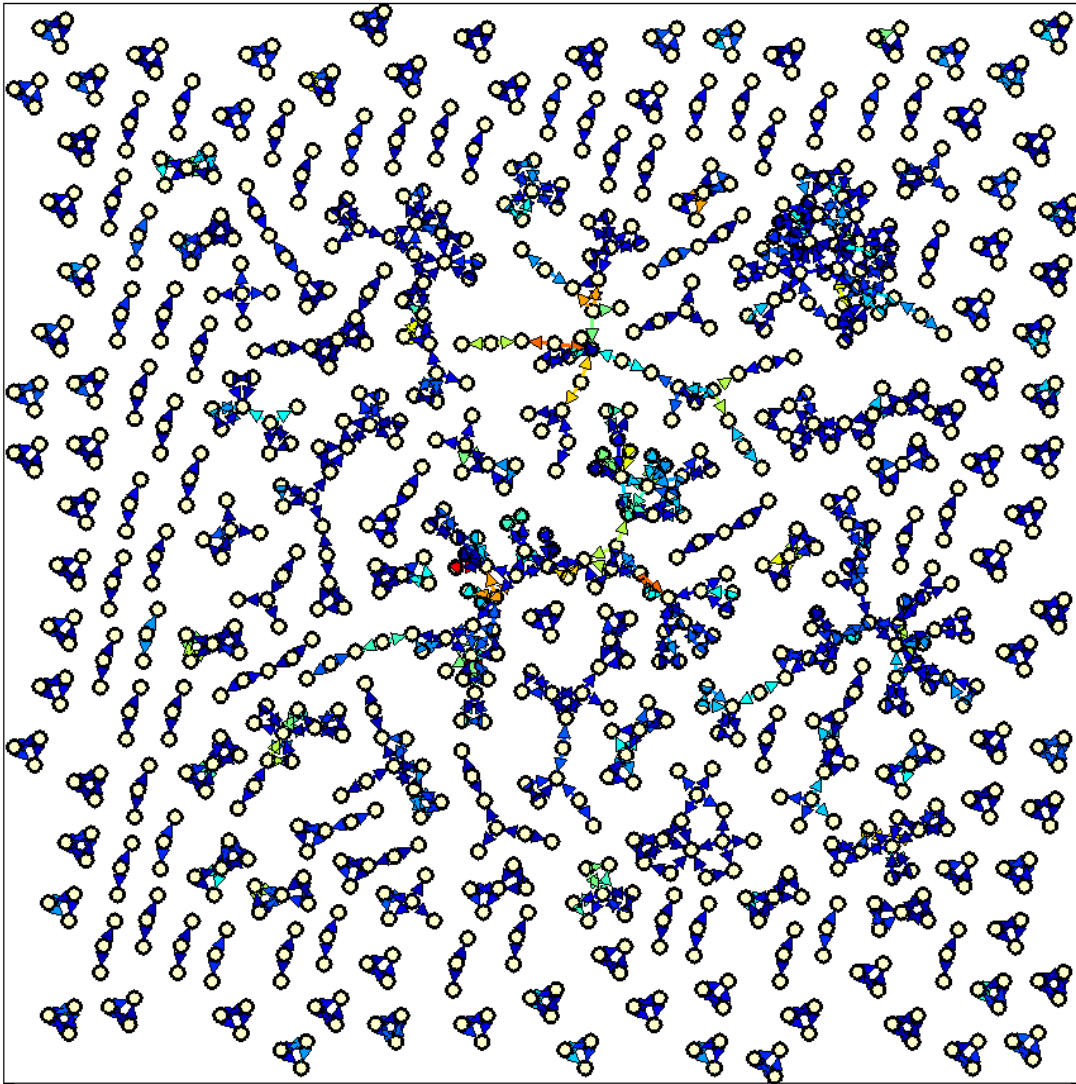


Figure 10.5: CASimIR similarity graph (Sections 3.1 and 7.2.2) before removal of cycles. Vertices represent clip pairs, directed edges represent similarity relations. Several large groups of similarity are linked through transitive relations. Lighter colours correspond to greater edge weights α . The spatial arrangement minimises edge collisions.

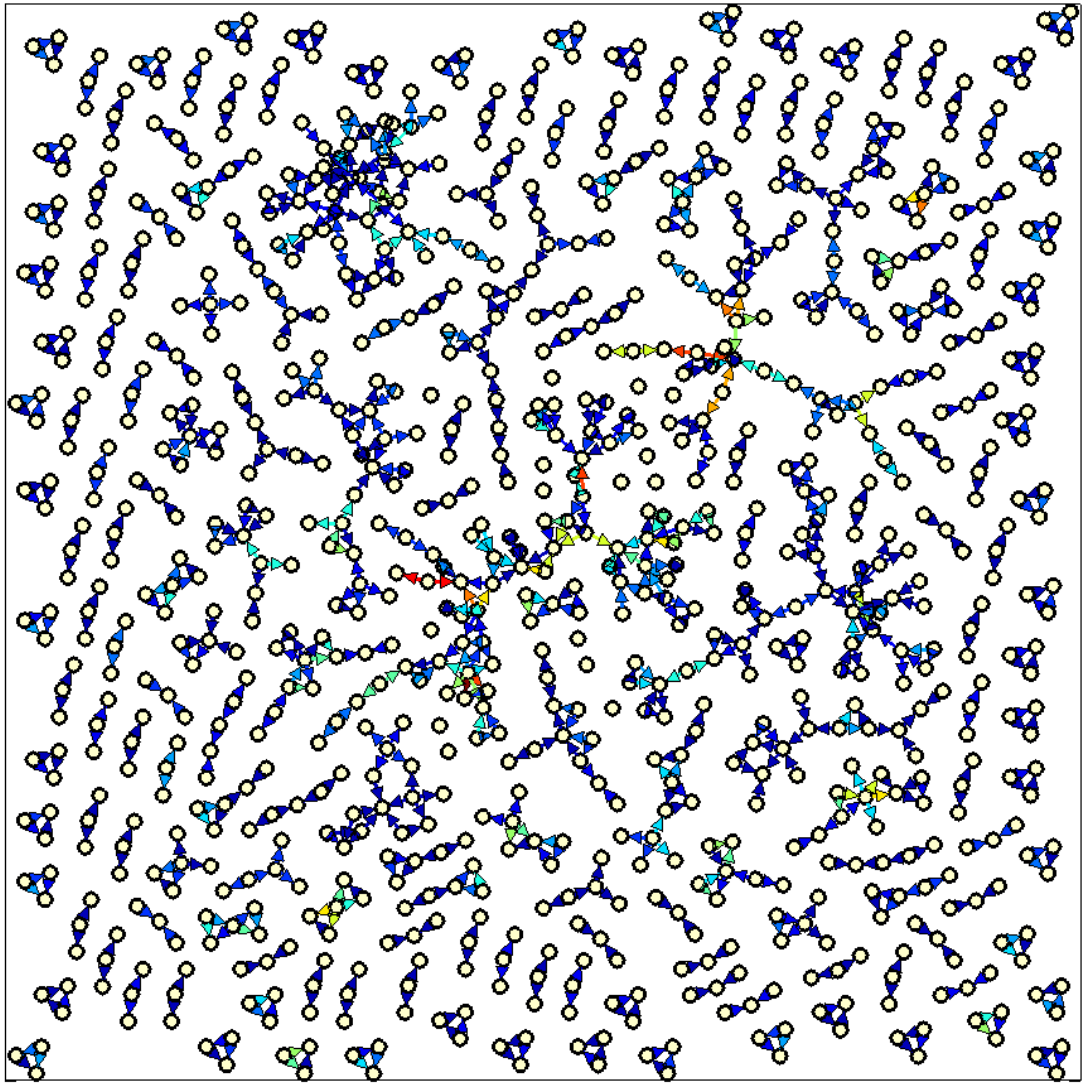


Figure 10.6: CASimIR similarity graph (Sections 3.1 and 7.2.2) **after** removal of cycles. Vertices represent clip pairs, directed edges represent similarity relations. Several large groups of similarity remain linked through transitive relations even after cycle removal. Lighter colours correspond to greater edge weights α .

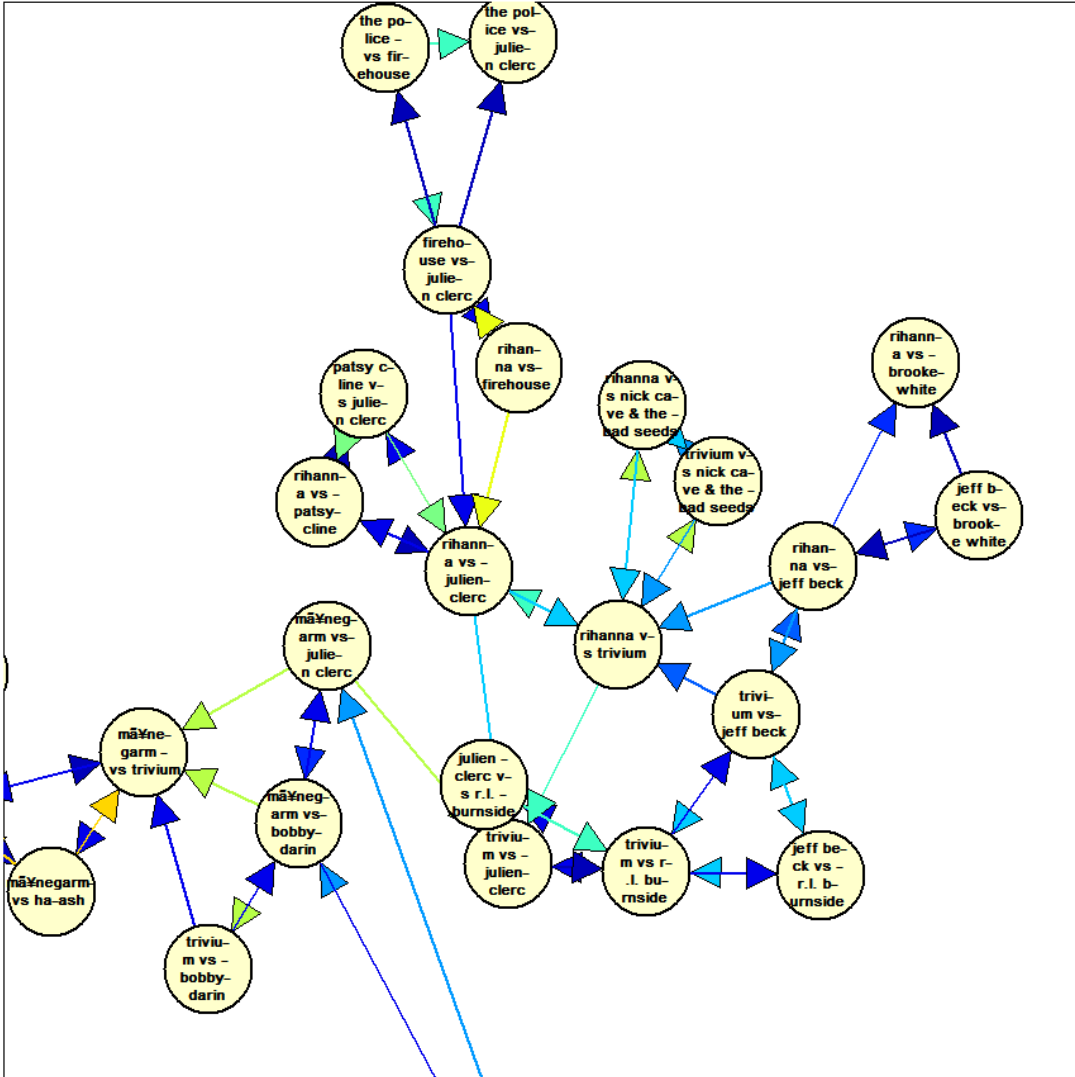


Figure 10.7: Excerpt of the biggest connected component of the CaSimIR similarity dataset before cycle removal. Inconsistent similarity data are represented by two-sided arrows. Vertices represent clip pairs and are tagged with the clips' (artist A vs. artist B) each, directed edges represent the relation "more similar than". Lighter colours correspond to greater edge weights α . The spatial arrangement minimises edge collisions.

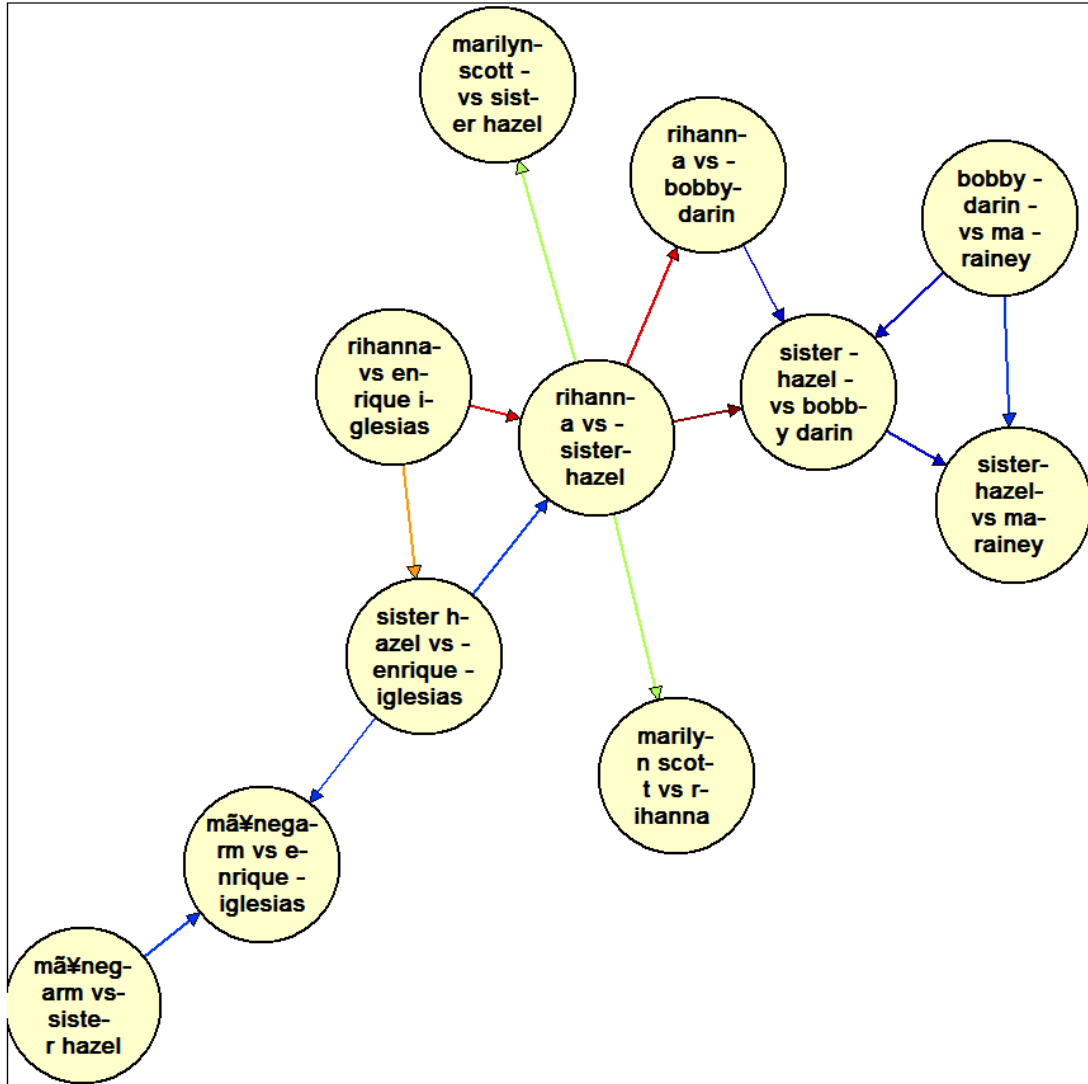


Figure 10.8: Graph with the 11th biggest connected component of the CaSimIR similarity dataset **after** removal of cycles. Vertices represent clip pairs and are tagged with the clips' (artist A vs. artist B) each, directed edges represent the relation "more similar than". Lighter colours correspond to greater edge weights α . The spatial arrangement minimises edge collisions.