

Music Similarity Clustering

A Content-Based Approach in Clustering
Music Files According to User Preference

Thesis-Project for the Degree

Bachelor of Science in Cognitive Science

Bastian Tenbergen

(tenberge@oswego.edu)

Human Computer Interaction M.A. Program
Oswego State University

formerly

Cognitive Science B.Sc. Program
University of Osnabrück, Germany

Outline

- Introduction
 - Related Work
 - This approach
- Clustering Music Files
 - Feature Extraction using GAs
 - Similarity Clustering using SOMs
 - Implementation
- Experiments
 - Experimental Set-up
 - Results
- Discussion
- Future Work

1. Introduction

- Large number of music files on end-user computers
- Acquired by
 - network based peer-to-peer networks
 - Music online stores
 - Internet radio stations (i.e. Pandora)
- Problems:
 - Keeping an overview over the database
 - Disambiguating information on mutual database
 - Information retrieval for new and/or unknown files

1.1 Related Work

- Many researchers have addressed these problems
- Many different approaches
- Two basic forms:
 - Pre-interpreted music representation
 - MIDI-files, scores, etc.
(Unal, Narayanan & Chew, 2004;
Uitdenbogers & Zobel, 2004;
Pardo, Shlfrin, Birmingham, 2004)
 - Real world audio data / raw data
 - Mp3, CDDA, ogg-VOBIS, flac, etc

1.1 Related Work

- Two basic approaches:
 - Mathematical / statistical approach
 - (Mierswa & Morik, 2005; Pardo et al., 2004)
 - Neurophysiological approach
 - (Tzanetakis, Essl & Cook, 2001)
- Both possibly involving neural networks
 - (Schedl, Pampalk & Widmer, 2004)

1.1 Related Work

- Query-by-humming approach
 - Synthetic error model
- vs
- Singer based error model

Good results, even with synthetic error model

- Problems:
 - MIDI files rather uncommon
 - Poor singing ability of user

1.1 Related Work

- Genre Classification
 - Genetic Algorithm searches for optimal feature extraction method
 - Support Vector Machine classifies genres:
 - pop/classic, pop/techno, pop/hiphop
- Very Good Results!
- Uses real-world Mp3 files!
- Problem:
 - Very slow

1.2 This Approach

- To the best of my knowledge, no research has been conducted that aims at dividing music files into preference groups.
- similarity measurement necessary
- How represent files?
 - > Extract Features.
- What features should be extracted?

1.2 This Approach

- Use GA approach of Mierswa and Morik
- Given a vector representation of files:
 - Similar files are closer to each other in vector space
 - **Problem:** Vector space of very high dimensionality
 - > curse of dimensionality
- More Problems:
 - Similarity clustering is ill-defined!
 - Number and size of clusters not known a priori

1.2 This Approach

- Special requirements for framework:
 - Deal with high dimensional data
 - Simplify high dimensional data
 - Deal with no a priori knowledge
- SVM not suitable
 - Classes not pre-defined
- Solution: **Self-Organizing Map**
 - maps high dimensional data on low dimensional representation
 - Number of clusters emerges from training

2. Clustering Music Files

Goals:

- Find music that a user likes
 - Do so on the basis of the content, not meta data
- Structure a music database
 - On basis other than genres, artists, albums
- Disambiguate a music database
 - Filter duplicates, retrieve info for songs
- Narrow search space for manual comparison

2. Clustering Music Files

Considerations:

- The larger the variety of genre and style, the larger the scope and variety of features
 - > different features might be necessary for different songs
- Number of clusters emerges from training
 - > the more songs in the corpus, the more clusters might be found

2.1 Feature Extraction using GAs

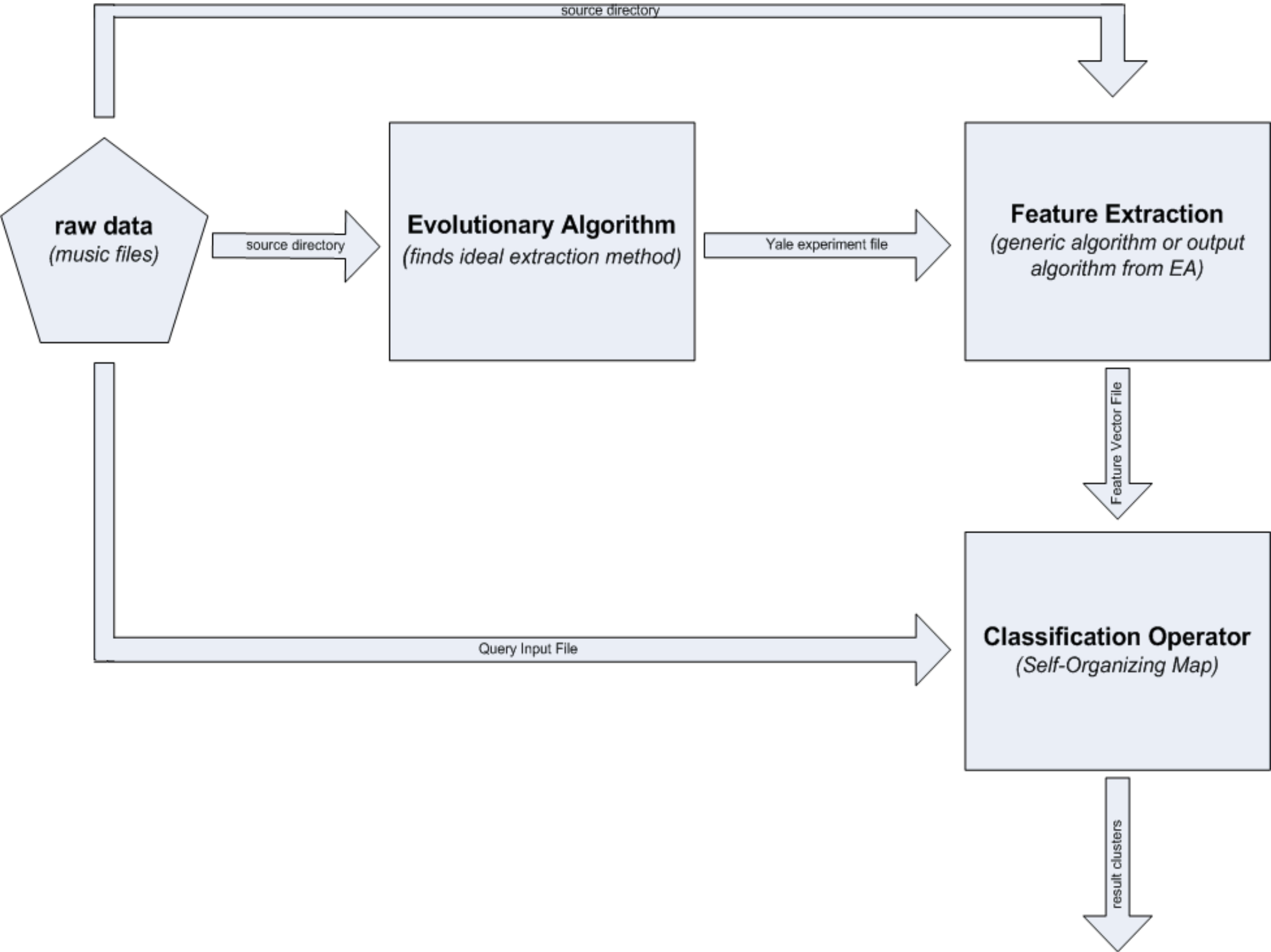
- Use Genetic Algorithm to find ideal feature extraction algorithm on the basis of a given test corpus
 - Idea adopted from Mierswa and Morik, **but corpus not pre-structured**
- Result is a extraction algorithm
- Extraction algorithm creates feature vector for each song

2.2 Similarity Clustering using SOMs

- Feature-vectors with d dimensions
 - Depending on the amount of extraction methods in the extraction algorithm
 - Each extraction method is one dimension
- Nice side effect: visualize input space

2.3 Implementation

- Plug-in for the Machine Learning Environment “YALE”
- Input:
 - Source folder of music files
(MPEG Layer III at 44.1KHz, see ISO/IEC11172-3)
 - Query File
(file in the source folder according to which clustering is performed)
 - Settings for GA and SOM
- Output:
 - Feature vector file
 - A list of music files in the same cluster as the Query File



Tree

XML

Box

Results

Monitor

Root
Experiment

Self-Organizing Map
Self-Organizing Map

Key	Value
Feature Vector File	<input type="text"/> ...
Query Input File	<input type="text"/> ...
Number Neuron-Columns	7
Number Neuron-Rows	5
Closed Topology	<input type="checkbox"/>
Training Runs	1.000
Learning Rate Winner	1
Learning Rate Neighbor	0,25
Maximal Value	-1
Metric	Euclidean
Filter Patterns	<input checked="" type="checkbox"/>
Normalize Patterns	<input type="checkbox"/>
Verbosing	<input checked="" type="checkbox"/>

Navigation icons: back, forward, home, search, delete, refresh.

3.1 Experimental Set-up

- Three-step process:
 - 1. find best extraction algorithm
 - 2. extract features
 - 3. cluster with SOM
- Three corpus sizes:
 - Three small corpora: approx 20 files
 - Three medium corpora: approx 100 files
 - One large corpus: 1,554 files

Overall corpus: 1,914 files
8,07 Gigabyte
160 hours playtime

German Top-100 from 1990 to 2005

3.1 Experimental Set-up

- GA settings:
- 50 generations
 - Early stopping criterion:
stop if no significant change in 10 successive generations
- Individuals per gen:
 - Three times the corpus size
 - Minimally 100 inds
- Mutation probability: 0.2
- cross-over probability: 0.5

3.1 Experimental Set-up

- SOM settings:
- $n \times m$ matrix, $n, m > 1$
- Open topology
- 1000 training runs
- Learning rate winner neuron: 1
- Learning rate neighbors: 0.25

3.1 Experimental Set-up

- Distance metric:

$$d(\mathbf{x}, \mathbf{y}) = \sqrt[\lambda]{\sum_{i=1}^n |x_i - y_i|^\lambda} \quad \lambda \in \mathbb{R}$$

Equation 1. Minkowski Distance.

With $\lambda = 1$: Manhattan Metric, $\lambda = 2$: Euclidean Distance.

$$d(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^n (1 + m) \quad m = \begin{cases} 0 & x_i = y_i \\ 1 & \text{else} \end{cases}$$

Equation 2. Nominal Distance.

(according to Schedl et al, 2005, no superiority of a metric exists)

3.1 Experimental Set-up

- User was asked to evaluate performance of clustering process
- User was given the resulting suggestions (i.e. The list of files which are potentially perceptually similar to the Query File)
- 22 year old German college student with no background in music or music theory
- Ask to evaluate each song in list according to it's similarity to the Query File
- 6-point scale

3.2 Results

- Very good results!
- 77% peak accuracy
- 67.3% average accuracy
 - Files received a rating of at least 4 or 5 by human user
- 34.3% false positive
 - Files received a rating of less than 3
- 17% average false negatives (forgotten files)
 - Files, not in the cluster of the QF, but received a rating of 4 or higher
- Clustering performance increases with increasing corpus size

3.2 Results

- GA did not contribute to good performance
- Generic feature sets provided comparable results
- Best SOM performance when number of neurons is 30% of the number of files

Similar Songs in Cluster (ranking):	Similar songs not in cluster (ranking):
Elton John & George Michael Don't Let The Sun Go Down on Me (5)	Band Ohne Namen Take my Heart (5)
Garland Jeffreys Hail Hail Rock 'n Roll (5)	Highland Bella Stella (5)
Michael Jackson Heal the World (5)	Echt Weinst Du (5)
Youssou Ndour & Neneh Cherry 7 Seconds (5)	Die Ärzte Wie Es Geht (5)
Meat Loaf I'd Do Anything for Love (5)	Die 3. Generation Ich will, dass Du mich liebst (5)
Young Deenay Walk On By (5)	Laura Immer wieder (5)
Thomas D Liebesbrief (5)	Reamonn Supergirl (4)
Christina Aguilera Beautiful (5)	Orange Blue She's Got That Light (4)
Coldplay Speed Of Sound (5)	R Kelly If I could Turn back Hands of Time (4)
Hypertraxx The Darkside (4)	Sisqo The Thong Song (3)
Nelly Furtado I'm Like a Bird (4)	Rednex Spirit of The Hawk (3)
Atomic Kitten It's OK (4)	Santana Maria Maria (3)
Daniel Bedingfield If You're Not The One (4)	Madonna Music (3)
Nomad I wanna give you devotion (4)	Madonna American Pie (3)
Salt 'n Pepa Lets Talk about Sex (3)	Sting Desert Rose (3)
Ace of Base Don't Turn Around (3)	Gabrielle Rise (3)
Dune Hardcore Vibes (3)	Anastasia I'm Outa Love (3)
Chris Brown Run It (3)	Manu Chao Bongo Bong (3)
Dru Hill How Deep is Your Love (2)	Music Instructor feat. Dean Super Fly (3)
J-Kwon Tipsy (2)	(results truncated)
Color Me Badd I Wanna Sex You Up (1)	Corpus Size : 1,554
Cher Believe (1)	Neurons : 15x15
Interactive Living Without Your Love (0)	Filtering : No
Wolfgang Petry Die Längste Single der Welt (0)	Normalization : No
	Total # of Clusters : 143
	Similar files : 24
	Target Song: Metallica Nothing else Matters

Table 1. Similarity Cluster for "Nothing else Matters" by Metallica.

4. Discussion

- Very promising results
- Allows for decrease in search space for manual file comparison
- Search space dynamically structured
- Design allows for ad hoc-clustering
 - (i.e. Extracting features of new files and adding them to the cluster without repeating the whole clustering process)

4. Discussion

- Poor GA performance!
- -> ga can be left out of the clustering process
-> decreases time and space complexity
- But:
 - Generic feature set should be designed for SOM clustering specifically
 - SVM approach still outperforms in classification tasks, yet clustering tasks are promising

4. Discussion

- System is useful to
 - Disambiguate music libraries
 - Help research in music theory
 - Find similarity relationships between songs
 - Find more music according to user preference
 - Visualize music style distribution of cross-over genres, artists and alike

5. Future Work

- Conduct more thorough user rating
- Design a more stream-lined extraction algorithm for SOM clustering
- Test performance on wider range of music style
- Develop a standalone application from plug-in

Any Questions?

Thank you for your attention!