# Automatic Classification of Musical Artists based on Web-Data

**Peter Knees[1], Elias Pampalk[2], Gerhard Widmer[1,2]**

[1]**Department of Computational Perception, JKU Linz**
Altenberger Str. 69, A-4040 Linz

[2]**Austrian Research Institute for Artificial Intelligence**
Freyung 6/6, A-1010 Wien

peter.knees@jku.at, elias@oefai.at, gerhard.widmer@jku.at

## Abstract

The organization of music is one of the central challenges in times of increasing distribution of digital music. A well-tried means is the classification in genres and/or styles. In this paper we propose the use of text categorization techniques to classify artists present on the Internet. In particular, we retrieve and analyze webpages ranked by search engines to describe artists in terms of word occurrences on related pages. To classify artists we primarily use support vector machines.
Based on a previously published paper and on a master's thesis, we present experiments comprising the evaluation of the classification process on a taxonomy of 14 genres with altogether 224 artists, as well as an estimation of the impact of daily fluctuations in the Internet on our approach, exploiting a long-term study over a period of almost one year. On the basis of these experiments we study (a) how many artists are necessary to define the concept of a genre, (b) which search engines perform best, (c) how to formulate search queries best, (d) which overall performance we can expect for classification, and finally (e) how our approach is suited as a similarity measure for artists.

## Introduction

Organizing music is a challenging task. Nevertheless, the vast number of available pieces of music requires ways to structure them. One of the most common approaches is to classify music into genres and styles. Genre usually refers to high-level concepts such as jazz, classical, pop, blues, and rock. On the other hand, styles are more fine-grained such as drum & bass and jungle in the genre electronic music. In this work, we do not distinguish between the terms genre and style. We use the term genre in a very general way to refer to categories of music which can be described using the same vocabulary.
Although even widely used genre taxonomies are inconsistent (for a detailed discussion see, e.g. [14]), they are commonly used to describe music. For example, genres can help locating an album in a record store or discovering similar artists. One of the main draw-

backs of genres is the time-consuming necessity to classify music manually. However, recent work (e.g. [20, 24, 1, 11]) suggests that this can be automatized.
Several approaches exist to describe music by extracting features. One flexible but challenging is to analyze the audio signal directly. A complementary approach is to analyze cultural features, also referred to as *community metadata* [23]. Community metadata includes data extracted through collaborative filtering, co-occurrence of artists in structured, readily available metadata (such as CDDB) [15], and artist similarities calculated from web-based data with text-retrieval methods [24, 2, 4]. In the following, we will not distinguish between the terms community metadata, cultural metadata, and web-based metadata.
In this paper, we extract features for artists from web-based data and classify the artists with support vector machines (SVMs). In particular, we query Internet search engines with artist names combined with constraints such as *+music +review* and retrieve the top ranked pages. The retrieved pages tend to be common web pages such as fan pages, reviews from online music magazines, or music retailers. This allows us to classify any artist present on the web using the Internet community's collective knowledge. In the experiments conducted we classify 224 artists into 14 genres (16 artists per genre). Some of these genres are very broad such as classical, others are more specific such as punk and alternative rock. We compare the performances of Google and Yahoo, as well as 3 different types of queries. One of the main questions is the number of artists necessary to define a genre such that new artists are correctly classified. Furthermore, we demonstrate the possibility of using the extracted descriptors also for a broader range of applications, such as similarity-based organization and visualization. Finally, we investigate the impact on the results of fluctuations over time of the retrieved content. For this experiment we retrieved the top ranked pages from search engines for 12 artists every fourth day for a period of 11 months.
The remainder of this paper is organized as follows. In the next section, we briefly review related work. Then, we describe the methods we use. After that, we describe our experiments and present the results. In the last section, we will discuss limitations of the approach and point out future directions.

## Related Work

Basically, related work can be classified into two groups, namely, artist similarity from metadata, and genre classification from audio. First, we review metadata-based methods. In [15] an approach is presented to compute artist and song similarities from co-occurrences on samplers and radio station playlists. From these similarities rough genre structures are derived using clustering techniques. The finding that groups of similar artists (similar to genres) can be discovered in an unsupervised manner by considering only cultural data was further supported by [1]. While the above approaches focus on structured data, [23, 2] also consider information available on common web sites. The main idea is to retrieve top ranked sites from Google queries and apply standard text-processing techniques like n-gram extraction and part-of-speech tagging. Using the obtained word lists, pairwise similarity of a set of artists is computed. The applicability of this approach to classify artists into 5 genres (heavy metal, contemporary country, hardcore rap, intelligent dance music, R&B) was shown by Whitman and Smaragdis [24] using a weighted k-NN variant. One of the findings was that community metadata works well for certain genres (such as intelligent dance music), but not for others (such as hardcore rap). This is dealt with by combining audio-based features with community metadata. Since metadata-based and audio signal-based methods are not directly related, we just want to give a brief overview of the classification categories used in systems based on audio signal analysis. In one of the first publications on music classifica-

tion, Tzanetakis [21] used 6 genres (classic, country, disco, hip hop, jazz, and rock), where classic was further divided into choral, orchestral, piano, and string quartet. In [20] this taxonomy was extended with blues, reggae, pop, and metal. Furthermore, jazz was subdivided into 6 subcategories (bigband, cool, fusion, piano, quartet, and swing). In the experiments, the subcategories were evaluated individually. For the 10 general categories, a classification accuracy of 61% was obtained. In [3], a hierarchically structured taxonomy with 13 different musical genres is proposed. Other work usually deals with smaller sets of genres. In [25] and [19] 4 categories (pop, country, jazz, and classic) are used with a classification accuracy of 93%, respectively 89%. In [11] 7 genres (jazz, folk, electronic, R&B, rock, reggae, and vocal) are used and the overall accuracy is 74%. In the present paper, we will demonstrate how we achieve up to 93% for 14 genres.

## Method

For each artist, we search the web either with Google or Yahoo. The query string consists either solely of the artist's name as an exact phrase or of the name extended by the keywords *+music +review* (+MR) as suggested in [21] or *+music +genre +style* (+MGS). Without these constraints searching for groups like Sublime would result in many unrelated pages. From the retrieved sites, we remove all HTML markup tags, taking only the plain text content into account. We use common English stop word lists to remove frequent terms (e.g. a, and, or, the). In our first results published at ISMIR 2004 [8] due to various reasons (e.g. server not responding) on average we were only able to retrieve about 40 from the top 50 ranked pages successfully. In our current version we retrieve more than the top 50 to fill the missing gaps, which has significantly improved performance [7]. For each artist *a* and each term *t* appearing in the retrieved pages, we count the number of occurrences $tf_{ta}$ (term frequency) of term t in documents relating to *a*. Furthermore, we count $df_t$ the number of pages on which the term occurred (document frequency). These are combined using the *term frequency* × *inverse document frequency* (*tf*×*idf*) function (we use the ltc variant [18]). The term weight per artist is computed as,

$$w_{ta} = \begin{cases} (1+\log_2 tf_{ta}) \log_2 \frac{N}{df_t}, & \text{if } tf_{ta} > 0, \\ 0, & \text{otherwise,} \end{cases}$$

where *N* is the total number of pages retrieved. A web crawl with 200 artists might retrieve more than 200,000 different terms. Most of these are unique typos or otherwise irrelevant, and thus we remove all terms which do not occur in at least 5 of the up to 50 pages retrieved per artist. As a result, between 3,000 and 10,000 different terms usually remain. Note that one major difference to previous approaches such as [23, 2] is that we do not search for n-grams or perform part-of-speech tagging. Instead we use every word (with at least 2 characters) which is not in a stop word list.

From a statistical point of view it is problematic to learn a classification model given only a few training examples described by several thousand dimensions. To further reduce the number of terms we use the $\text{chi}^2$-test which is a standard term selection approach in text classification (e.g. [26]). The $\text{chi}^2$-value measures the independence of *t* from category *c* and is computed as

$$c_{tc}^2 = \frac{N(AD-BC)^2}{(A+B)(A+C)(B+D)(C+D)}$$

where *A* is the number of documents in *c* which contain *t*, *B* the number of documents not in *c* which contain *t*, *C* the number of documents in *c* without *t*, *D* the number of documents not in *c* without *t*, and *N* is the total number of retrieved documents. Since *N* is equal for all terms, it can be ignored. The terms with highest $\mathrm{chi}^2_{tc}$-values are selected because they are least independent from *c*. Given $\mathrm{chi}^2_{tc}$-values for every term in each category there are different approaches to select one global set of terms to describe all documents. For our experiments, we select the *n* highest for each category and join them into a global list. We got best results using the top 100 terms for each category, which gives us a global term list of up to 14×100 terms (if there is no overlap in top terms from different categories). Table 1 gives a typical list of the top 100 terms in the genre reggae. Note that we do not remove words which are part of the queries. We use the notation $C_n$ to describe the strategy of selecting *n* terms per category. In case of $C_8$ we do not remove any terms based on the $\mathrm{chi}^2_{tc}$-values and thus do not require prior knowledge of which artist is assigned to which category. (This is of particular interest when using the same representation for similarity measures.) After term selection each artist is described by a vector of term weights. The weights are normalized such that the length of the vector equals 1 (Cosine normalization), to reduce the influence of repeated word occurrences in longer documents. To classify the artists we primarily use support vector machines [22]. SVMs solve high-dimensional problems extremely efficiently and are a particularly good choice for text categorization (e.g. [6]). In our experiments we used a linear kernel as implemented in the Matlab OSU Toolbox[1]. In addition to SVMs we use k-nearest neighbors (k-NN) for classification to evaluate the performance of the extracted features in similarity based applications.

To visualize the artist data space we use self-organizing maps [10], an unsupervised clustering technique. The SOM maps high-dimensional vectors onto a 2-dimensional map such that similar vectors are located close to each other. While the SOM requires a similarity measure, it does not require any training data where artists are assigned to genres. Thus, we can use the algorithm to find the inherent structure in the data and, in particular, to automatically organize and visualize music collections (e.g. [16,17]). For our experiments we use the Matlab SOM Toolbox[2].

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 100 | reggae | 19 | vibration | 11 | aswad | 9 | *grant |
| 83 | *marley | 18 | *eddy | 11 | yuh | 9 | tubby |
| 68 | dancehall | 18 | isaacs | 11 | bounty | 9 | ghetto |
| 57 | jamaica | 18 | wailer | 11 | cliff | 8 | tra |
| 50 | *shabba | 18 | babylon | 11 | makers | 8 | sponji |
| 48 | wailers | 17 | dragonfly | 11 | loverman | 8 | caan |
| 47 | *uhuru | 17 | vp | 11 | beres | 8 | dennis |
| 46 | *capleton | 15 | toots | 11 | soca | 8 | cedella |
| 45 | *ziggy | 15 | cocoa | 10 | maxi | 8 | labour |
| 45 | *ub40 | 15 | *bob | 10 | luciano | 8 | damian |
| 45 | *shaggy | 15 | dread | 10 | lexxus | 8 | tuff |
| 39 | jamaican | 15 | ganja | 10 | hotshot | 8 | rastafarian |
| 37 | jah | 15 | riddims | 10 | sly | 8 | minott |
| 31 | rasta | 15 | maytals | 10 | prophet | 8 | nuh |
| 31 | buju | 15 | selassie | 10 | gregory | 8 | gal |
| 30 | sizzla | 14 | boombastic | 10 | spear | 8 | mikey |
| 30 | banton | 14 | dem | 9 | exodus | 8 | wicked |
| 29 | kingston | 14 | greensleeves | 9 | dunbar | 8 | kaya |
| 27 | beenie | 14 | rastaman | 9 | duckie | 8 | jo'anna |
| 24 | ragga | 14 | vibes | 9 | sinsemilla | 7 | africa |
| 24 | ras | 13 | bunny | 9 | inna | 7 | demus |
| 23 | tosh | 12 | yellowman | 9 | elephant | 7 | tok |
| 23 | dub | 12 | zion | 9 | trojan | 7 | abyssinians |
| 23 | *ranks | 12 | pon | 9 | gong | 7 | eek |
| 21 | riddim | 12 | augustus | 9 | ting | 7 | puma |

**Table 1:** 100 terms with highest $\mathrm{chi}^2$-values for reggae defined by Black Uhuru, Bob Marley, Capleton, UB40, Shaggy, Eddy Grant, Shabba Ranks, Ziggy Marley using +MR. * marks words from search queries; values normalized (highest score=100).

[1] http://www.ece.osu.edu/~maj/osu_svm

[2] http://www.cis.hut.fi/projects/somtoolbox

## Experiments

In the experiments conducted, we classify 224 artists from 14 partly overlapping genres (16 artists per genre). Furthermore, the results of an experiment over time where the same queries were sent to a search engine every fourth day over a period of 11 months to measure the variance in the results are presented.

#### Experiment with 224 Artists

To evaluate our approach on a larger dataset we use the 14 genres from [8]. To each genre, 16 artists are assigned. The complete list is available online[3]. For each artist, we compute the *tf×idf* representation as described before. The classification accuracies are estimated via 50 hold out experiments. For each run, 2, 4, or 8 artists out of the 16 per genre are randomly selected to define the concept of the genre. The remaining ones are used for testing. The main classification results are listed in Table 2 and Table 3. The reason why we experiment with defining a genre using only 2 artists is the following application scenario. A user has an MP3 collection structured by directories which reflect genres to some extent. For each directory, we extract the artist names from the ID3 tags. Any new MP3s added to the collection should be (semi)automatically assigned to the directory they best fit into based on the artist classification. Thus, we are interested in knowing how well the system can work given only few examples. Using SVMs and 8 artists to define a genre we get up to 93% accuracy which is quite impressive given a baseline accuracy of only 7%. Generally the results of Yahoo are significantly worse. We assume that the reason is that Yahoo does not strictly enforce the constraints if many search terms are given. We observe that the +MR constraint generally performs better than +MGS and, at least for Google, better than queries with no constraints. We would also like to point out that, using only 2 artists to define a genre we get surprisingly good results of up to 79% accuracy using SVMs. The confusion matrix for an experiment with Google +MR (SVM, t8, $C_{100}$) is shown in Figure 1. Classical music and jazz are not confused with the other genres. Also reggae is classified very accurately. In contrast to the results published in [24] rap/hiphop is nearly perfectly distinguished. Some of the main errors are that alternative/indie is wrongly classified as electronica, and punk is confused with alternative and heavy metal/hard rock (all directions). The latter errors "make sense", the former needs further investigation.
In addition to the results using SVMs we also investigated the performance using k-NN (without $\mathrm{chi}^2$ cut-off) to estimate how well our approach is suited as a similarity measure. Similarity measures have a very broad application range. For example, we would like to apply a web-based similarity measure to our islands of music approach were we combine different views of music for interactive browsing [16]. Accuracies of up to 83% are very encouraging. However, one remaining issue is the limitation to the artist level, while we would prefer a more fine-grained similarity measure at the song level. Since the accuracy increases with growing number of training examples, it would be interesting to evaluate similarity on the full set of artists. Therefore, we conducted leave-one-out cross-validations for every artist and determined the most similar with a 1-NN classifier using Euclidean distance. We did this for the features acquired by Google with +MR and +MGS. The results are strongly enforcing the usage of web-based features for similarity: for +MR 82% of the artists are most similar to an artist from the same genre, for +MGS even 87%.

[3] http://www.cp.jku.at/people/knees/artistlist224.html

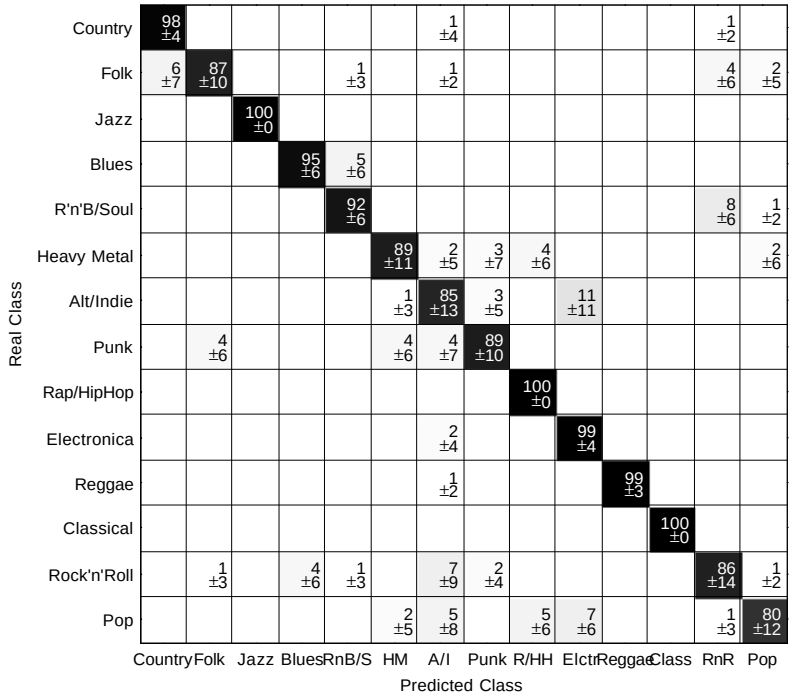


**Figure 1:** Confusion matrix of classification results using a SVM with Google +MR $C_{100}$ data using 8 artists per category for training. Values are given in percent. The lower value in each box is the standard deviation computed from 50 hold out experiments.

| | Google | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | *no constraints* | | | music genre style | | | music review | | |
| | t2 | t4 | t8 | t2 | t4 | t8 | t2 | t4 | t8 |
| SVM $C_{100}$ | 71±3.7 | 80±3.2 | 87±2.6 | 79±4.0 | 87±2.7 | 90±2.3 | 77±4.1 | 87±2.5 | 93±2.0 |
| SVM $C_{200}$ | 67±4.8 | 80±3.0 | 87±2.7 | 79±4.4 | 86±2.4 | 91±2.3 | 76±3.8 | 87±2.6 | 93±2.2 |
| SVM $C_8$ | 69±4.1 | 79±3.2 | 84±2.8 | 78±4.1 | 86±2.7 | 90±2.1 | 77±4.3 | 87±2.8 | 92±2.3 |
| 3-NN $C_8$ | 53±5.6 | 64±4.9 | 71±2.7 | 66±5.0 | 78±3.7 | 83±3.3 | 60±6.7 | 73±5.7 | 79±4.6 |
| 7-NN $C_8$ | 35±7.1 | 64±4.9 | 74±3.5 | 43±8.1 | 76±3.3 | 83±2.8 | 41±8.9 | 72±5.6 | 81±4.8 |

**Table 2:** Classification results achieved with Google on a set of 224 artists evaluated against a genre association as ground truth. The first value in each cell is the mean accuracy from 50 hold out experiments. The second value is the standard deviation. Values are given in percent. The number of artists (size of the training set) used as training examples is labelled with t2, t4, t8.

| | Yahoo! | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | *no constraints* | | | music genre style | | | music review | | |
| | t2 | t4 | t8 | t2 | t4 | t8 | t2 | t4 | t8 |
| SVM $C_{100}$ | 71±3.9 | 81±3.0 | 88±3.1 | 73±3.6 | 80±2.7 | 84±2.4 | 66±5.3 | 78±3.5 | 86±2.4 |
| SVM $C_{200}$ | 66±4.9 | 80±3.8 | 89±2.9 | 69±3.7 | 79±2.4 | 84±2.8 | 63±4.8 | 76±3.7 | 85±3.0 |
| SVM $C_8$ | 69±4.1 | 79±3.7 | 87±3.1 | 68±3.5 | 78±3.0 | 83±2.9 | 63±5.8 | 75±3.6 | 83±3.3 |
| 3-NN $C_8$ | 51±6.2 | 64±5.0 | 72±4.1 | 54±5.1 | 67±3.7 | 72±2.9 | 45±5.7 | 58±4.5 | 66±3.6 |
| 7-NN $C_8$ | 33±7.9 | 63±5.4 | 75±4.1 | 32±10. | 66±4.4 | 75±2.8 | 38±8.1 | 59±4.6 | 67±3.8 |

**Table 3:** Classification results achieved with Yahoo on a set of 224 artists evaluated against a genre association as ground truth. Labels as in Table 2.

| | | | | | |
|---|---|---|---|---|---|
| CLASSICAL (16) | JAZZ (7) | RNBSOUL (10) | rnbsoul (4)<br>pop (3)<br>folk (1)<br>raphiphop (1)<br>electro (1) | RAPHIPHOP(14) | REGGAE (15) |
| JAZZ (6)<br>rocknroll (1) | jazz (1) | altindie (1)<br>pop (1) | POP (6)<br>reggae (1) | raphiphop (1)<br>pop (1) | electro (1)<br>pop (1) |
| ROCKNROLL (5)<br>jazz (1)<br>rnbsoul (1) | ROCKNROLL (5)<br>blues (1)<br>rnbsoul (1) | country (1) | pop (3) | electro (2)<br>pop (1) | ELECTRO (12)<br>altindie (1) |
| BLUES (15) | COUNTRY (12)<br>folk (1) | FOLK (10)<br>country (3)<br>jazz (1) | ALTINDIE (5)<br>ROCKNROLL (5)<br>folk (4)<br>punk (1) | PUNK (5)<br>altindie (3)<br>heavymetal (2) | HEAVYMETAL (14)<br>PUNK (10)<br>ALTINDIE (6) |

**Figure 2:** SOM trained on 224 artists. The number of artists from the respective genre mapped to the unit is given in parenthesis. Upper case genre names emphasize units which represent many artists from one genre.

To further test the applicability as a similarity measure and to visualize the similarities of the features, we trained a SOM on all artists (Figure 2). The features from Google with the +MGS constraint have been chosen as underlying data, since they achieved the best results in the k-NN classification. We did not use the $\mathrm{chi}^2$ cut-off as this would require knowledge of the genre of each artist which we do not assume to be given in most scenarios. Classical artists (upper left) are all mapped to one unit. This shows, that classical artists are very similar to each other and very dissimilar to others. Also artists from reggae (upper right) and blues (lower left) are very well distinguishable from others (15 out of 16 artists on one unit). Furthermore, it can be seen that also the group of rap/hiphop artists has a high consistency (14 artists on one unit; next to reggae). In the lower right, the map gives the insight that heavy metal/hard rock, punk, and alternative/indie overlap strongly. Three units in that region contain 46 out of 48 artists from these genres. Furthermore, an overlap of alternative/indie with rock'n'roll occurs.

An interesting characteristic of the SOM is the overall order. This gives at least an impression of the similarity of the groups. For being able to quantize the correctness of the similarities between genres, explicit relations and similarities must be given in advance. From the overall order it can be seen that, among all other genres, jazz is the most similar to classical. Another impressive observation is, that the cluster with heavy metal/hard rock, punk, and alternative/indie has the maximum distance possible to classical. Also the neighbourhood of blues with rock'n'roll on one hand and country on the other seems reasonable, since these (together with the neighboring folk) influenced each other in the 60's and 70's. The last aspect that we want to focus on is the scattering of pop artists in the map. It can be seen well, that pop lies in the centre of all other genres and spreads to a lot of adjacent units. This can be seen as evidence that there exists no sharp contoured

definition for pop and also that pop artists do not necessarily form a homogeneous group. It is particularly interesting to see, to which units pop spreads. Roughly spoken, pop can be seen somewhere between the rock-affine genres, r'n'b/soul, rap/hiphop, and electronic, which is, considering the (US) billboard charts of the last 15 years, quite correct.

**Measuring the time dependency of features**

It is well known that contents on the Internet are not persistent (e.g. [9, 12]), and the top ranked pages of search engines are updated frequently. As a consequence, it is a realistic threat that querying to a certain time might lead to a biased view of the web. To measure how this influences the *tf×idf* representations we sent repeated queries to Google over a period of eleven months every fourth day (79 times) starting on December 18th, 2003. The queries comprise 12 randomly chosen artists from different genres. For each artist we sent queries with +MR. For all available pages from the top 50, the *tf×idf*-vectors (without $\mathrm{chi}^2$ term selection) are calculated. For reasons of computational efficiency, only every fourth day is evaluated, although data for (almost) every day would be available. This leads to the 79 data points per artist in time. In case of unavailable data (e.g. due to a not responding search engine) data from the next day is used instead. To visualize the variance we trained a SOM with all vectors. The results can be seen in figure 3. For example, it comes out, that all 79 tf×idf-vectors for the artists Robbie Williams, Youssou N'Dour, Daft Punk, Strokes, Marshall Mathers, and Mozart are mapped to almost one unit. This is a relatively strong indicator, that the data remains consistent over a period of 11 months. The vectors for Eminem and Marshall Mathers (Eminem's real name) are neighboring. It is worth mentioning, that there are no overlaps between artists. This means, that every unit is at most representing vectors from one artist. For a more detailed discussion of the results, see [7].

Finally, we can state that there are significant variations in the underlying sites. The SOM shows that the impact of these variations is not dramatic for the presented approach since they do not cause any overlaps or confusion. Changes in the data are no negative phenomenon that has to be marginalized by any means. Rather, they are one of the main reasons, why the exploration of web-based data is that attractive. As the public opinion on an artist changes and evolves, the representation should also change. Based on the impressions of the long-term study, this seems possible without suffering from the obligation of unforeseeable results. Thus, we can conclude that the feature extraction is not excessively negatively influenced by data variations over time, although further research is needed to study the impact on larger sets of artists.

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| Youssou N'Dour (79) | | | Stacie Orrico (79) | | Robbie Williams (79) | | Daft Punk (79) |
| | | | | | | | Strokes (79) |
| | | | Alicia Keys (78) | Alicia Keys (1) | | | |
| Mozart (79) | | | | | | | Michael Jackson (79) |
| | | | | | Eminem (9) | | |
| Pulp (77) | Pulp (2) | Sublime (3) | Sublime (76) | | Eminem (70) | | Marshall Mathers (79) |

**Figure 3:** SOM trained on data retrieved with +MR over a period of 11 months. The number below the artists abbreviation is the number of results from different days mapped to the same unit.

## Limitations and further improvements

Although first results are very encouraging, with the web-based data we face several limitations. One of the main problems is that our approach heavily relies on the underlying search engines and the assumption that the suggested webpages are highly related to the artist. Although some approaches to estimating the "quality" of a webpage have been published (e.g. [2]), it is very difficult to identify off-topic websites without detailed domain knowledge. For example, to retrieve pages for the band Slayer, we queried Google with *"slayer" +music +genre +style* and witnessed unexpectedly high occurrences of the terms *vampire* and *buffy*. In this case a human might have added the constraint *-buffy* to the query to avoid retrieving sites dealing with the soundtrack of the tv-series "Buffy The Vampire Slayer". Similar problems have been discussed in [30] (e.g. bands with common word names like War or Texas are more susceptible to confusion with unrelated pages).
Furthermore, as artists or band names occur on all pages, they have a strong impact on the lists of important words (e.g. see Table 1). This might cause trouble with band names such as Daft Punk, where the second half of the name indicates a totally different musical style. In addition, also artists with common names can lead to confusion. For example, for pop artists like Michael Jackson and Janet Jackson, pages include the term *jackson* very frequently. The same is valid for artists such as country artist Alan Jackson, who will be susceptible to confusion with them. A variation of the same problem is e.g, rap artist Nelly, whose name is a substring of ethno-pop artist Nelly Furtado. One approach to overcome these problems would be to use noun phrases (as already suggested in [23]) or to treat artist names not as words but as special identifiers. We plan to address these issues in future work using n-grams and other more sophisticated content filtering techniques. First experiments with filters similar to those suggested in [26] did not achieve big performance boosts (see [7] for details).
Further options for future research are for example using the information from the Google ranks (the first page should be more relevant than the 50th) and the use of additional information like song titles.

## References

[1] J.-J. Aucouturier and F. Pachet, "Musical genre: A survey," *Journal of New Music Research*, vol. 32, no. 1, 2003.

[2] S. Baumann and O. Hummel, "Using cultural metadata for artist recommendation," in *Proc. of Wedel-Music*, 2003.

[3] J.J. Burred and A. Lerch, "A Hierarchical Approach to Automatic Musical Genre Classification," in *Proc. of the International Conf. on Digital Audio Effects*, 2003.

[4] W.W. Cohen and Wei Fan, "Web-collaborative filtering: Recommending music by crawling the web," *WWW9 / Computer Networks*, vol. 33, no. 1-6, pp. 685–698, 2000.

[5] F. Debole and F. Sebastiani, "Supervised term weighting for automated text categorization," in *Proc. of the ACM Symposium on Applied Computing*, 2003.

[6] T. Joachims, "Text categorization with support vector machines: Learning with many relevant features," in *Proc. of the European Conf. on Machine Learning*, 1998.

[7] P. Knees, "Automatische Klassifikation von Musikkünstlern basierend auf Web-Daten," Diplomarbeit, Technische Universität Wien, 2004.

[8] P. Knees, E. Pampalk, G. Widmer, “Artist Classification with Web-based Data,” in *Proc. of the International Conf. on Music Information Retrieval*, 2004.

[9] W. Koehler, “A longitudinal study of web pages continued: A consideration of document persistence,” *Information Research*, vol. 9, no. 2, 2004.

[10] T. Kohonen, *Self-Organizing Maps*, Springer, 2001.

[11] M.F. McKinney and J. Breebaart, “Features for audio and music classification,” in *Proc. of the International Conf. on Music Information Retrieval*, 2003.

[12] S. Lawrence and C. L. Giles, “Accessibility of Information on the Web,” in *Nature*, vol. 400, no. 6740, pp. 107–109, 1999.

[13] B. Logan and A. Salomon, “A music similarity function based on signal analysis,” in *Proc. of the IEEE International Conf. on Multimedia and Expo*, 2001.

[14] F. Pachet and D. Cazaly, “A taxonomy of musical genres,” in *Proc. of RIAO Content-Based Multimedia Information Access*, 2000.

[15] F. Pachet, G. Westerman, and D. Laigre, “Musical data mining for electronic music distribution,” in *Proc. of WedelMusic*, 2001.

[16] E. Pampalk, S. Dixon, and G. Widmer, “Exploring music collections by browsing different views,” *Computer Music Journal*, vol. 28, no. 3, pp. 49–62, 2004.

[17] E. Pampalk, A. Rauber, and D. Merkl, “Content-based organization and visualization of music archives,” in *Proc. of ACM Multimedia*, 2002.

[18] G. Salton and C. Buckley, “Term-weighting approaches in automatic text retrieval,” *Information Processing and Management*, vol. 24, no. 5, pp. 513–523, 1988.

[19] Xi Shao, C. Xu, and M.S. Kankanhalli, “Unsupervised classification of music genre using hidden markov model,” in *Proc. of the IEEE International Conf. of Multimedia Expo*, 2004.

[20] G. Tzanetakis and P. Cook, “Musical genre classification of audio signals,” *IEEE Transactions on Speech and Audio Processing*, vol. 10, no. 5, pp. 293–302, 2002.

[21] G. Tzanetakis, G. Essl, and P. Cook, “Automatic musical genre classification of audio signals,” in *Proc. of the International Symposium on Music Information Retrieval*, 2001.

[22] V. Vapnik, *Statistical Learning Theory*, Wiley, 1998.

[23] B. Whitman and S. Lawrence, “Inferring descriptions and similarity for music from community metadata,” in *Proc. of the International Computer Music Conf.*, 2002.

[24] B. Whitman and P. Smaragdis, “Combining musical and cultural features for intelligent style detection,” in *Proc. of the International Conf. on Music Information Retrieval*, 2002.

[25] C. Xu, N.C. Maddage, Xi Shao, and Qi Tian, “Musical genre classification using support vector machines,” in *Proc. of the International Conf. of Acoustics, Speech & Signal Processing*, 2003.

[26] Y. Yang and J.O. Pedersen, “A comparative study on feature selection in text categorization,” in *Proc. of the International Conf. on Machine Learning*, 1997.