ARTIST CLASSIFICATION WITH WEB-BASED DATA

Peter Knees[1], *Elias Pampalk*[1], *Gerhard Widmer*[1,2]
[1]Austrian Research Institute for Artificial Intelligence
Freyung 6/6, A-1010 Vienna, Austria
[2]Department of Medical Cybernetics and Artificial Intelligence
Medical University of Vienna, Austria

ABSTRACT

Manifold approaches exist for organization of music by genre and/or style. 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.

We present 3 experiments in which we address the following issues. First, we study the performance of our approach compared to previous work. Second, we investigate how daily fluctuations in the Internet affect our approach. Third, on a set of 224 artists from 14 genres 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.

Keywords: genre classification, community metadata, cultural features

1. 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 paper, 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. [18]), they are commonly used to describe music. For example, genres can help located an album in a record store or discover similar artists. One of the main drawbacks of genres is the time-consuming necessity to classify music manually. However, recent work (e.g. [25, 29, 2, 15]) suggests that this can be automatized.

A closely related topic is overall perceived music similarity (e.g. [11, 17, 1, 22, 4]). Although music similarity and genre classification share the challenge of extracting good features, the evaluation of similarity measures is significantly more difficult (for recent efforts in this direction see, e.g. [10, 5, 9, 20]).

Several approaches exist to extract features to describe music. One flexible but challenging approach is to analyze the audio signal directly. A complementary approach is to analyze cultural features, also referred to as community metadata [28]. Community metadata includes data extracted through collaborative filtering, co-occurrence of artists in structured, readily available metadata (such as CDDB) [19], and artist similarities calculated from web-based data with text-retrieval methods [29, 3, 7]. 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*.

We present 3 experiments. First, we compare our approach to previously published results on a set of 25 artists classified into 5 genres using web-based data [29].

Second, 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 other day for a period of 4 months.

Third, 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 2 different constraints on the queries.

One of the main questions is the number of artists necessary to define a genre such that new artists are correctly classified. Finally, we demonstrate the possibility of using the extracted descriptors also for a broader range of applications, such as similarity-based organization and visualization.

The remainder of this paper is organized as follows. In the next section we briefly review related work. In Section 3 we describe the methods we use. In Section 4 we describe our experiments and present the results. In Section 5 we draw conclusions and point out future directions.

2. 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 [19] 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 [2].

While the above approaches focus on structured data, [28, 3] 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 [29] 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 classification, Tzanetakis [26] used 6 genres (classic, country, disco, hip hop, jazz, and rock), where classic was further divided into choral, orchestral, piano, and string quartet. In [25] 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 [6] a hierarchically structured taxonomy with 13 different musical genres is proposed.

Other work usually deals with smaller sets of genres. In [30] and [24] 4 categories (pop, country, jazz, and classic) are used with a classification accuracy of 93%, respectively 89%. In [15] 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 87% for 14 genres.

3. METHOD

For each artist we search the web either with Google or Yahoo. The query string consists of the artist's name as an exact phrase extended by the keywords *+music +review* (+MR) as suggested in [28] or *+music +genre +style* (+MGS). Without these constraints searching for groups such as *Sublime* would result in many unrelated pages.

We retrieve the 50 top-ranked webpages for each query and 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).

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 the term occurred in (document frequency). These are combined using the term frequency × inverse document frequency ($tf \times idf$) function (we use the *ltc* variant [23]). The term weight per artist is computed as,

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

where N is the total number of pages retrieved. Note that 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.

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 [28, 3] 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 (in the experiments below we use up to 112) described by several thousand dimensions. To further reduce the number of terms we use the χ^2 test which is a standard term selection approach in text classification (e.g. [31]). The χ^2-value measures the independence of t from category c and is computed as,

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

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. As N is equal for all terms, it can be ignored. The terms with highest χ^2_{tc} values are selected because they are least independent from c.

Note that the idf part of $tf \times idf$ can be replaced with the χ^2_{tc}-value in text classification as suggested in [8]. However, in our experiments this did not improve the results.

Given χ^2_{tc}-values for every term in each category there are different approaches to select one global set of terms to describe all documents. A straightforward approach is to select all terms which have the highest sum or maximum value over all categories, thus using either terms which perform well in all categories, or using those which perform well for one category.

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 2 gives a typical list of the top 100 terms in the genre heavy metal/hard rock. 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$_\infty$ we do not remove any terms based on the χ^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). This removes the influence that the length of the retrieved webpages would otherwise have. (Longer documents tend to repeat the same words again and again which results in higher term frequencies.)

To classify the artists we primarily use support vector machines [27]. SVMs are based on computational learning theory and solve high-dimensional problems extremely efficiently. SVMs are a particularly good choice for text categorization (e.g. [12]). In our experiments we used a linear kernel as implemented in LIBSVM (version 2.33) with the Matlab OSU Toolbox. [1,2]

In addtion 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 [14] which belong to the larger group of unsupervised clustering techniques. 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. [22, 21]). For our experiments we used the Matlab SOM Toolbox. [3]

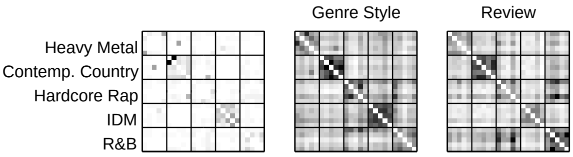


Figure 1. Distance matrix for the 25 artists. On the left is the matrix published in [29], the other two matrices we obtained using $tf \times idf$ (with C$_\infty$). Black corresponds to high similarity, white to high dissimilarity. The diagonals of the matrices are set to the largest distance to improve the contrast. Note that the overall differences in brightness are due to the two extreme outlier values in contemporary country (thus the grayscale in the right matrix needs to cover a larger range). However, for k-NN classification not the absolute values but merely the rank is decisive.

	1-NN	3-NN	5-NN	7-NN
Whitman & Smaragdis	68	80	76	72
Google Music Genre Style	96	92	96	92
Google Music Review	80	76	84	80

Table 1. Results for k-nearest neighbor classification for 25 artists assigned to 5 genres. The values are the percentage of correctly classified artists computed using leave-one-out cross validation.

4. EXPERIMENTS

We ran three experiments. First, a very small one with 25 artists for which genre classification results have been published by Whitman and Smaragdis [29]. Second, an experiment over time where the same queries were sent to search engines every second day over a period of almost 4 months to measure the variance in the results. Third, a larger one with 224 artists from 14 partly overlapping genres which are more likely to reflect a real world problem.

4.1. Whitman & Smaragdis Data

Although the focus in [29] was not on genre classification Whitman and Smaragdis published results which we can compare to ours. They used 5 genres to which they assigned 5 artists each. The distance matrix they published is shown graphically in Figure 1. Using the distance matrix we apply k-NN to compare our $tf \times idf$ approach to describe artist similarity. The classification accuracies are listed in Table 1.

As pointed out in [29], and as can be seen from the distance matrix, the similarities work well for the genres contemporary country and intelligent dance music (IDM). However, for hardcore rap, heavy metal, and R&B the results are not satisfactory. Whitman and Smaragdis presented an approach to improve these by using audio similarity measures.

As can be seen in Table 1 our results are generally better. In particular, when using the constraint +MGS in the Google queries we only get one or two wrong classifications. *Lauryn Hill* is always misclassified as hardcore rap

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

[2] http://www.csie.ntu.edu.tw/˜cjlin/libsvm

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

Sub 56			RW 56		SO 54		YND 56
					SO 2		
Moz 56							AK 21
			MJ 56				AK 35
MM 56	Em 24	Em 32		DP 56		Pulp 56	Stro 56

Figure 2. SOM trained on data retrieved over a period of about 4 months. The full artist names are listed in Figure 3. The number below the artists abbreviation is the number of results from different days mapped to the same unit.

instead of R&B. Furthermore, *Outkast* tends to be misclassified as IDM or R&B instead of hardcore rap. Both errors are forgivable to some extent.

When using +MR as constraint in the Google queries the results do not improve consistently but are on average 6 percentage points better than those computed from the Whitman and Smaragdis similarity matrix. The distance matrix shows that there is a confusion between hardcore rap and R&B.

The big deviations between the constraint +MGS and +MR are also partly time dependent. We study the variations over time in the next section.

4.2. Experiment measuring Time Dependency

It is well known that contents on the Internet are not persistent (e.g. [13, 16]) and the top ranked pages of search engines are updated frequently. To measure how this influences the $tf \times idf$ representations we sent repeated queries to Google over a period of almost 4 months every other day (56 times) starting on December 18th, 2003.

We analyzed 12 artists from different genres (for a list see Figure 3). For each artist we used the constraints +MR or +MGS. We retrieved the 50 top ranked pages and computed the $tf \times idf$ vectors (without χ^2 term selection).

We studied the variance by training a SOM on all vectors. The resulting SOM (using the +MGS constraint) is shown in Figure 2. For example, all 56 $tf \times idf$ vectors for *Sublime* are mapped to the upper left corner of the map. The vectors for *Eminem* and *Marshall Mathers* are located next to each other. Note that there is no overlap between artists (i.e. every unit represents at most one artist). This indicates that the overall structure in the data is not drastically effected.

In addition we measured the variation over time by computing the following. Given 56 vectors $\{\mathbf{v}_{ad}\}$ for an artist a where d denotes the day the pages were retrieved we compute the artist's mean vector $\bar{\mathbf{v}}_a$. For each artist we measure the daily distance from this mean as $d_{ad} = ||\mathbf{v}_a - \mathbf{v}_{ad}||$. The results for +MGS and +MR are shown in Figure 3. We normalize the distances so that the mean distance between *Eminem* and *Marshall Mathers* (Eminem's real name) equals 1.

The results show that in general the deviations from the mean are significantly smaller than 1 for all artists. However, there are some exceptions. For example, for the +MGS constraint some of the queries for *Michael Jackson* are quite different from the mean. We assume that the recent court case and its attention in the media might be one of the reasons for this.

We obtained the best results with the smallest variance for the African artist *Youssou N'Dour* who is best known for his hit Seven Seconds (released 1994). The hypothesis that this might be because N'Dour has not done anything which would have attracted much attention from December 2003 to April 2004 does not hold as this would also apply, for example, to the alternative ska-punk band *Sublime* who have significantly more variance but disbanded in 1996 after their lead singer died.

Another observation is that the variances are quite different for the 2 constraints. For example, *Pulp* has a very low variance for +MR (median deviation is about 0.45) and a high one for +MGS (median deviation is above 0.6). However, looking at all 12 artists both constraints have a similar overall variance.

We can conclude that there are significant variations in the retrieved pages. However, as we can see from the SOM visualizations, these variations are so small that they do not lead to overlaps between the different artists. Thus, we can expect that the classification results are not greatly influenced. Further research is needed to study the impact on larger sets of artists.

4.3. Experiment with 224 Artists

To evaluate our approach on a larger dataset we use 14 genres (country, folk, rock'n'roll, heavy metal/hard rock, alternative rock/indie, punk, pop, jazz, blues, R&B/soul, rap/hiphop, electronic, reggae, and classical). To each genre we assigned 16 artists. The complete list of 224 artists is available online. [4]

For each artist we compute the $tf \times idf$ representation as described in Section 3. Table 2 lists the top 100 words for heavy metal/hard rock selected using the χ^2 test. Note that neither of the constraint words (review and music) are in the list.

The top 4 words are all (part of) artist names which were queried. However, many artists which are not part of the queries are also in the list, such as Phil Anselmo (Pantera), Hetfield, Hammett, Trujillo (Metallica), and Ozzy Osbourne.

Furthermore, related groups such as Slayer, Megadeth, Iron Maiden, and Judas Priest are found as well as album names (Hysteria, Pyromania, ...) and song names (Paranoid, Unforgiven, Snowblind, St. Anger, ...) and other descriptive words such as evil, loud, hard, aggression and heavy metal.

The main classification results are listed in Table 3. The classification accuracies are estimated via 50 hold out experiments. For each run from the 16 artists per genre ei-

[4] http://www.oefai.at/˜elias/ismir04

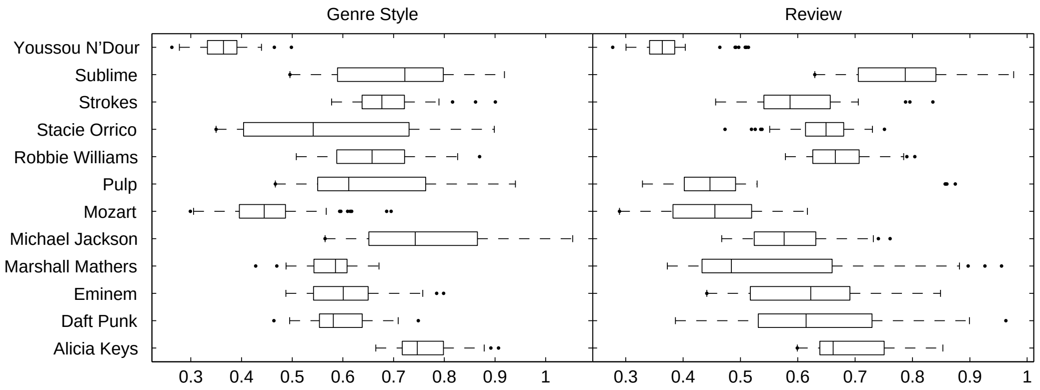


Figure 3. Boxplots showing the variance of the data over time. The x-axis is the relative distance between the mean per artist over time and each day, normalized by the average distance between the vectors of *Eminem* and *Marshall Mathers*. The boxes have lines at the lower quartile, median, and upper quartile values. The whiskers are lines extending from each end of the box to show the extent of the rest of the data (the maximum length is 1.5 of the inter-quartile range). Outliers are data with values beyond the ends of the whiskers.

100	*sabbath	26	heavy	17	riff	12	butler
97	*pantera	26	ulrich	17	leaf	12	blackened
89	*metallica	26	vulgar	17	superjoint	12	bringin
72	*leppard	25	megadeth	17	maiden	12	purple
58	metal	25	pigs	17	armageddon	12	foolin
56	hetfield	24	halford	17	gillan	12	headless
55	hysteria	24	dio	17	ozzfest	12	intensity
53	ozzy	23	reinventing	17	leps	12	mob
52	iommi	23	lange	16	slayer	12	excitable
42	puppets	23	newsted	15	purify	12	ward
40	dimebag	21	leppards	15	judas	11	zeppelin
40	anselmo	21	adrenalize	15	hell	11	sandman
40	pyromania	21	mutt	15	fairies	11	demolition
40	paranoid	20	kirk	15	bands	11	sanitarium
39	osbourne	20	riffs	15	iron	11	*black
37	*def	20	s&m	14	band	11	appice
34	euphoria	20	trendkill	14	reload	11	jovi
32	geezer	20	snowblind	14	bassist	11	anger
29	vinnie	19	cowboys	14	slang	11	rocked
28	collen	18	darrell	13	wizard	10	drummer
28	hammett	18	screams	13	vivian	10	bass
27	bloody	18	bites	13	elektra	9	rocket
27	thrash	18	unforgiven	13	shreds	9	evil
27	phil	18	lars	13	aggression	9	loud
26	lep	17	trujillo	13	scar	9	hard

Table 2. The top 100 terms with highest χ^2_{tc}-values for heavy metal/hard rock defined by 4 artists (Black Sabbath, Pantera, Metallica, Def Leppard) using the +MR constraint. Words marked with * are part of the search queries. The values are normalized so that the highest score equals 100.

ther 2, 4, or 8 are randomly selected to define the concept of the genre. The remaining are used for testing.

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 87% accuracy which is quite impressive given a baseline accuracy of only 7%. Generally the results for Google are slightly better than those for Yahoo. For +MGS 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. In contrast to the findings of the dataset with 25 artists (Section 4.1) we observe that the +MR constraint generally performs better than +MGS.

We would also like to point out that using only 2 artists to define a genre we get surprisingly good results of up to 71% accuracy using SVMs with C_{100}. Performance is only slightly worse when using the top 200 words per genre (C_{200}) or even when not using the χ^2 test to select terms (C_∞).

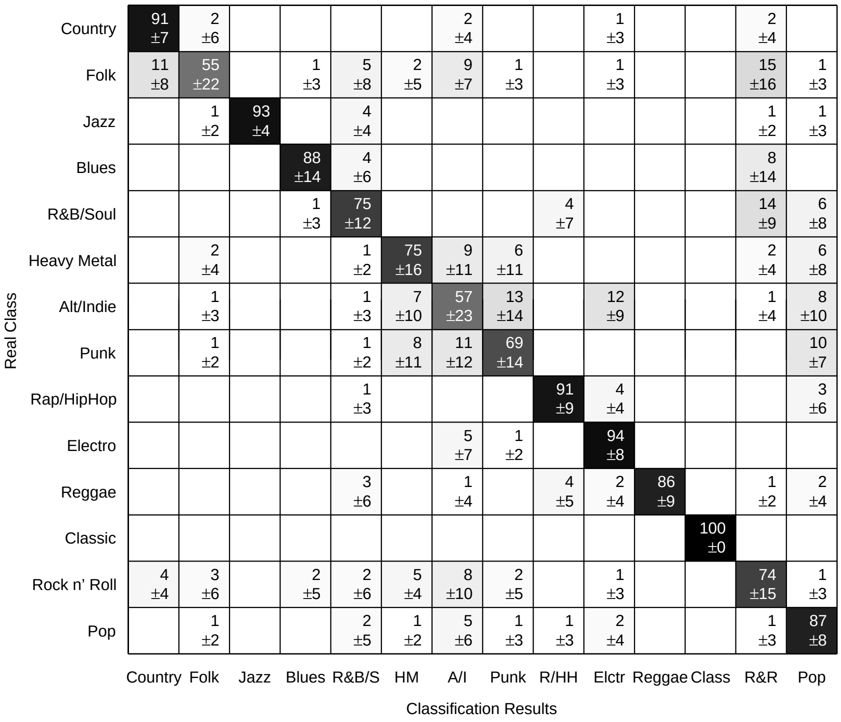
The confusion matrix for an experiment with Google +MR (SVM, t4, C_{100}) is shown in Figure 4. Classical music is not confused with the other genres. In contrast to the results published in [29] hip hop/rap is also very well distinguished. Some of the main errors are that folk is wrongly classified as rock'n'roll, and punk is confused with alternative and heavy metal/hard rock (all directions). Both errors "make sense". On the other hand, any confusion between country and electronic (even if only marginal) needs further investigation.

In addition to the results using SVMs we also investigated the performance using k-NN (without χ^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 [21]. Accuracies of up to 77% 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.

To further test the applicability as a similarity measure, we trained a SOM on all artists (Figure 5). We did not

	Google						Yahoo						
	Genre Style			Review			Genre Style			Review			
	t2	t4	t8	t2	t4	t8	t2	t4	t8	t2	t4	t8	Mean
SVM C_{100}	70±3.7	80±2.9	86±2.3	71±4.3	81±3.1	87±3.0	61±4.3	72±3.1	79±2.9	65±4.9	78±2.9	87±2.6	76±3.3
SVM C_{200}	67±3.8	78±3.0	85±2.7	68±4.3	79±3.3	86±2.6	56±4.4	69±3.3	78±3.0	62±4.5	75±3.2	85±3.2	74±3.4
SVM C_{∞}	67±3.8	77±3.1	84±3.0	69±4.7	79±3.5	84±2.7	56±4.8	67±3.7	74±3.1	65±4.9	76±2.1	85±3.1	73±3.5
3-NN C_{∞}	54±6.9	66±4.6	73±3.8	56±4.6	68±4.3	74±3.3	39±6.1	51±5.6	58±3.9	51±6.9	62±4.7	71±3.7	60±4.9
7-NN C_{∞}	39±7.7	67±3.7	75±3.0	43±8.2	68±4.5	77±3.7	31±9.0	51±5.5	62±3.7	40±8.5	63±5.2	73±3.7	57±5.5
Mean	59±5.2	74±3.5	81±3.0	61±5.2	75±3.7	81±3.0	49±5.7	62±4.2	70±3.3	57±5.9	71±3.6	80±3.3	
Mean	71±3.9			73±4.0			60±4.4			69±4.3			

Table 3. Classification results on the 224 artist dataset. 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 to define a genre is labeled with t2, t4, t8.



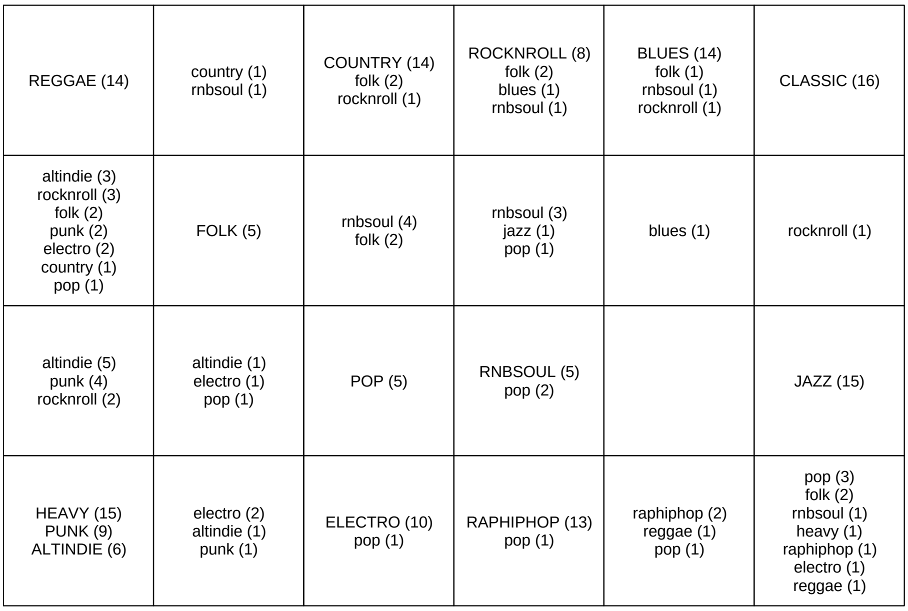
	Country	Folk	Jazz	Blues	R&B/S	HM	A/I	Punk	R/HH	Elctr	Reggae	Class	R&R	Pop
Country	91 ±7	2 ±6					2 ±4			1 ±3			2 ±4	
Folk	11 ±8	55 ±22		1 ±3	5 ±8	2 ±5	9 ±7	1 ±3		1 ±3			15 ±16	1 ±3
Jazz		1 ±2	93 ±4		4 ±4								1 ±2	1 ±3
Blues				88 ±14	4 ±6								8 ±14	
R&B/Soul				1 ±3	75 ±12				4 ±7				14 ±9	6 ±8
Heavy Metal		2 ±4			1 ±2	75 ±16	9 ±11	6 ±11					2 ±4	6 ±8
Alt/Indie		1 ±3			1 ±3	7 ±10	57 ±23	13 ±14		12 ±9			1 ±4	8 ±10
Punk		1 ±2			1 ±2	8 ±11	11 ±12	69 ±14						10 ±7
Rap/HipHop					1 ±3				91 ±9	4 ±4				3 ±6
Electro							5 ±7	1 ±2		94 ±8				
Reggae					3 ±6		1 ±4		4 ±5	2 ±4	86 ±9		1 ±2	2 ±4
Classic												100 ±0		
Rock n' Roll	4 ±4	3 ±6		2 ±5	2 ±6	5 ±4	8 ±10	2 ±5		1 ±3			74 ±15	1 ±3
Pop		1 ±2			2 ±5	1 ±2	5 ±6	1 ±3	1 ±3	2 ±4			1 ±3	87 ±8

Figure 4. Confusion matrix of classification results using a SVM with Google +MR C_{100} data using 4 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.

use the χ^2 cut-off as this would require knowledge of the genre of each artist which we do not assume to be given in the islands of music scenario. The SOM confirms some of the results from the confusion matrix. Classic (upper right) is clearly separated from all others. Jazz and reggae are also very well distinguished. Heavy metal, punk, and alternative overlap very strongly (lower left). Folk is very spread out and overlaps with many genres. An interesting characteristic of the SOM is the overall order. Notice that blues and jazz are located closer to classical music while electronic is close to alternative. Furthermore, the SOM offers an explanation of the confusion between electronic and folk. In particular, 2 artists from electronic and from folk together with artists from many other genres are mapped to the same unit (in the 2nd row, 1st column). The main reason for this is that some of the artists we assigned to each genre are very “mainstream” and thus their $tf \times idf$ representations are more similar to other mainstream artists than to typical members of their genre that are not so popular.

5. CONCLUSIONS

In this paper we have presented an approach to classifying artists into genres using web-based data. We conducted 3 experiments from which we gained the following insights. First, we showed that our approach outperformed a previously published approach [29]. Second, we demonstrated that the daily fluctuations in the Internet do not significantly interfere with the classification. Third, on a set of 224 artists from 14 genres we showed that classification accuracies of 87% are possible. We conclude that in our experiments Google outperformed Yahoo. Furthermore,



REGGAE (14)	country (1) rnbsoul (1)	COUNTRY (14) folk (2) rocknroll (1)	ROCKNROLL (8) folk (2) blues (1) rnbsoul (1)	BLUES (14) folk (1) rnbsoul (1) rocknroll (1)	CLASSIC (16)
altindie (3) rocknroll (3) folk (2) punk (2) electro (2) country (1) pop (1)	FOLK (5)	rnbsoul (4) folk (2)	rnbsoul (3) jazz (1) pop (1)	blues (1)	rocknroll (1)
altindie (5) punk (4) rocknroll (2)	altindie (1) electro (1) pop (1)	POP (5)	RNBSOUL (5) pop (2)		JAZZ (15)
HEAVY (15) PUNK (9) ALTINDIE (6)	electro (2) altindie (1) punk (1)	ELECTRO (10) pop (1)	RAPHIPHOP (13) pop (1)	raphiphop (2) reggae (1) pop (1)	pop (3) folk (2) rnbsoul (1) heavy (1) raphiphop (1) electro (1) reggae (1)

Figure 5. SOM trained on 224 artists. The number of artists from the respective genre mapped to the unit is given in parentheses. Upper case genre names emphasize units which represent many artists from one genre.

we achieved best results using the constraint *+music +review* in the search engine queries. A particularly interesting insight we obtained was that defining a genre with only 2 artists results in accuracies of up to 71%. Finally, we demonstrated that the features we extract are also well suited for direct use in similarity measures.

Nevertheless, 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 estimate the "quality" of a webpage have been published (e.g. [3]), 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". Similarly, as already pointed out in [28], 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 2). This might cause trouble with band names such as *Daft Punk*, where the second half of the name indicates a totally different genre. In addition, also artists with common names can lead to misclassification. For example, if the genre pop is defined through *Michael Jackson* and *Janet Jackson*, any page including the term *jackson* (such as those from country artist *Alan Jackson*) will be more likely to be classified as pop. 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 [28]) 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 as suggested in [3].

Further, we plan to investigate classification into hierarchically structured genre taxonomies similar to those presented in [6]. Other plans for future work include using the information from the Google ranks (the first page should be more relevant than the 50th), experimenting with additional query constraints, and combining the web-based similarity measure with our islands of music approach to explore different views of music collections [21].

6. ACKNOWLEDGEMENTS

This research was supported by the EU project SIMAC (FP6-507142) and by the Austrian FWF START project Y99-INF. The Austrian Research Institute for Artificial Intelligence is supported by the Austrian Federal Ministry for Education, Science, and Culture and by the Austrian Federal Ministry for Transport, Innovation, and Technology.

7. REFERENCES

[1] J.-J. Aucouturier and F. Pachet, "Music similarity measures: What's the use?," in *Proc. of the International Conf. on Music Information Retrieval*, 2002.

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

[3] S. Baumann and O. Hummel, “Using cultural metadata for artist recommendation,” in *Proc. of WedelMusic*, 2003.

[4] A. Berenzweig, D. Ellis, and S. Lawrence, “Anchor space for classification and similarity measurement of music,” in *Proc. ot the IEEE International Conf. on Multimedia and Expo*, 2003.

[5] A. Berenzweig, B. Logan, D. Ellis, and B. Whitman, “A large-scale evaluation of acoustic and subjective music similarity measures,” in *Proc. of the International Conf. on Music Information Retrieval*, 2003.

[6] 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.

[7] 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.

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

[9] J.S. Downie, “Toward the scientific evaluation of music information retrieval systems,” in *Proc. of the International Conf. on Music Information Retrieval*, 2003.

[10] D. Ellis, B. Whitman, A. Berenzweig, and S. Lawrence, “The quest for ground truth in musical artist similarity,” in *Proc. of the International Conf. on Music Information Retrieval*, 2002.

[11] J.T. Foote, “Content-based retrieval of music and audio,” in *Proc. of SPIE Multimedia Storage and Archiving Systems II*, 1997, vol. 3229.

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

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

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

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

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

[17] 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.

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

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

[20] E. Pampalk, S. Dixon, and G. Widmer, “On the evaluation of perceptual similarity measures for music,” in *Proc. of the International Conf. on Digital Audio Effects*, 2003.

[21] 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.

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

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

[24] 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.

[25] 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.

[26] 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.

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

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

[29] 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.

[30] 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.

[31] 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.