Topic extraction for music genre using lyrical content

Sharon Gieske
6167667
sharongieske@gmail.com

David van Erkelens
10264019
daviddvanerkelens@gmail.com

Elise Koster
5982448
koster.elise@gmail.com

December 23, 2014

Abstract

Genre classification and music recommendations are an important service of online music providers. Since manual genre classification is very time-consuming, especially with ever larger quantities of music available, an accurate classifier for music genres enhances the user experience of music providers and provides insight into themes that differentiate genres from each other. This paper proposes an extension of LDA to model topics over genres instead of documents, and shows that this extension outperforms regular LDA on classification tasks.

1 Introduction

Online music services consist of large databases of meta-data for songs and artists, which are used to suggest artists to users based on their earlier listening habits. However, most of the information on such websites is crowd sourced and thus error-prone. Moreover, obscure artists and releases are often unlabeled. The main problem with manual labeling is that the process costs both time and money. Consequently, automated classification of music genres would provide music services with a cost efficient way of labeling. Furthermore, a generative model approach may provide new insight into the themes present in different music genres that elude casual fans.

Music genre classification has been researched mostly using audio signals [8]. While this approach uses the *defining* characteristic of most music genres, namely their musical structure, an other approach is to use word-based analysis, which has the advantage of smaller datasets, since text files are smaller than audio files. Some earlier research has been done in the field by classifying songs using their lyrics, but often different lyric-features like syntactic structure [1], the amount of words per minute and the average word length [6], recognition of structures in the lyrics [5] or a combination of audio signals and lyrics [6] have been used. Little research has been done on classifying song using only the words of the lyrics [2], with topic models using Gibbs sampling. Other research has used LDA for artist assignment instead of genre classification [3], however, this research took audio features into account as well. However, since different music genres tend to deal with different themes and thus lyrics, can an extension of LDA over music genres instead of documents be used to classify music genres?

This paper proposes that extending LDA over genres instead of documents may prove useful in classifying music into genres. In order to classify data using this new model, a dataset was built, since there are no such datasets readily available. To achieve this, a crawler was written to automatically retrieve documents and labels. After building the dataset, it was cleaned and split into a training set and a test set (using 5-fold cross-validation). When running the algorithm, the topics were randomly initialized and Gibbs sampling was run for a set number of iterations. After this, the topic distribution for each genre from the test set was calculated to build a genre profile, and these profiles were used to train an SVM to predict the most probable genre of a new document.

The complete outline of the problem can be found in section 2, a detailed description of the approach used can be found in section 3 and the results of this approach can be found in section 4. The conclusions and discussion can be found in section 5 and the team responsibilities in section 6.

2 Problem

Automated genre classification based on lyrics has been attempted before [1], using a large range of features[1]. However, none of these features used topic modeling or an extension thereof. Moreover, topic modeling *has* been used for music analysis, but not in the context of genre classification [4] [3]. Thus, no research has been done using topics extracted using LDA (or an extension thereof) for genre classification.

Topic modeling is attempted partly because of the observations that different genres deal with different kinds of subjects. So much so, that stereotypes exist of fans of different music genres. For example, within the scope of the *Heavy Metal* genre, *Death Metal* is a genre that often focuses around death, gore, rot and murder, from either the victim or the perpetrator's point of view, while, *Black Metal* usually alludes to satanism, anti-Christianity and misanthropy. On the other hand, a genre like *Rap* has the stereotype of revolving around sex, drugs, gangs and violence, whereas *Reggae* is usually regarded to be about Rastafari culture and smoking marijuana. Moreover, certain music genres are named after the content of their lyrics; the *Holiday* genre primarily deals with Christmas, whereas the *Religious* genre contains all kinds of music, as long as the music deals with religious themes. An important side note is that certain music genres, such as *Techno*, *Trance* and *House* (with the exception of their *Vocal* subgenres) usually don't contain any lyrics. As such, the approach used in this paper is unsuited for classification of these types of music.

On the other hand, topic modeling is a *generative* approach, meaning that the models built can be used to create new 'songs' fitting of a genre. However, since the model used in this paper is a bag-of-words model, the resulting songs will be unstructured. Also, since low-information, common words, are assumed to not contribute much to topics, they are left out, and thus the generated songs will lack these.
The approach used in this paper assumes *a)* that words are significant markers of a music genre, *b)* that there is a significant difference between words in different music genres, *c)* that topic distributions are accurate representations of lyrical content within a genre and *d)* that common words[2] are low-information and can thus be left out of the dataset.

3 Approach

For this research the following framework is used:

1. Gather and label a dataset using crawler
2. Preprocessing (removal of stop words, non-English documents)
3. Model topics and extract topic distribution for genres
4. Train SVM on array of topic-probabilities per document (according to its genre label)
5. Generate 'song' of length n for genre g: select topic $k \sim \theta_g$ then select word $w \sim \varphi_k$, repeat n times

In the following subsections the dataset is described in section 3.1 and the extended LDA model is explained in section 3.2. The classification and evaluation methods are described in section 3.3 and the method to generate a song is displayed in section 3.4

3.1 Data acquisition

Before any model could be created and tested, data had to be collected and processed. To do this, a crawler was built that collects song lyrics from *Lyricsmode*[3] by going through the alphabet (with one added 'letter' for numbers $0 - 9$), finding the top 100 artists starting with that letter and collecting the first five songs (alphabetically) by that artist. After collecting these lyrics, genre-information was retrieved from *Allmusic*[4]. The choice of these websites was primarily based on earlier research [1] that used these websites because of their consistency.

[1] n-grams, vocabulary, syntactic structure, semantics and more

[2] for example, 'I', 'was', 'to'

[3] `http://www.lyricsmode.com`

[4] `http://www.allmusic.com`

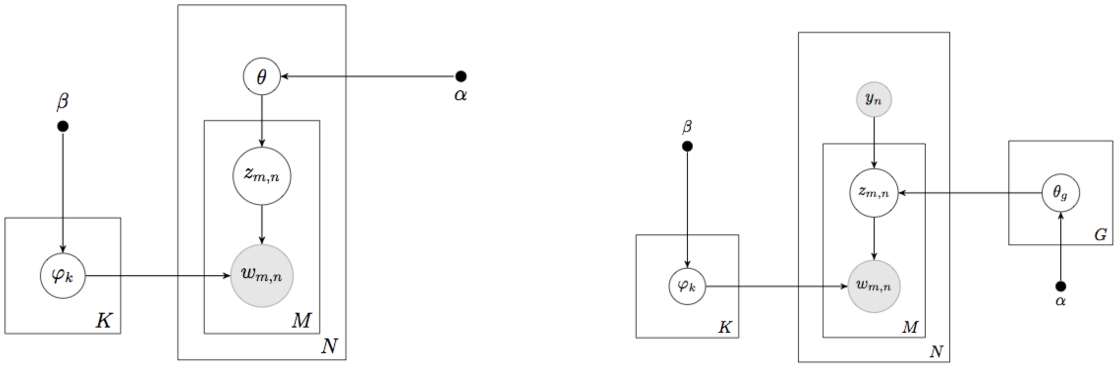
After collecting a dataset of 12.102 lyrics, the data was pre-processed to filter out *a)* punctuation marks and capitalization *b)* non-English documents and *c)* low-content, common words[5], resulting in a dataset of 9.728 lyrics. These pre-processing steps were done under the assumption that for topic modeling, including common words would create low-information topics that would hardly contribute to the model. Another assumption was that punctuation and capitalization, while containing useful information regarding language, were not relevant to the kinds of topics that would be found (as topic modeling is usually a bag-of-words approach). Non-English lyrics were filtered because extracting topics over multiple language would add a lot of noise to the model. Note however, that not *all* non-English words were filtered out (only songs that were in a different language entirely): in many cases, the use of *some* non-English words can say a lot about a genre[6].

3.2 Extended LDA model

Latent Dirichlet Allocation (LDA) treats every document like a collection of words, and presumes that each word in a document is generated by a certain topic. A hidden distribution over topics is assumed to exist for every document. The goal of LDA is to *a)* find a distribution over topics for each document and *b)* find a distribution over words for each topic. Thus, we're looking for two sets of distributions: $\theta_i \sim \text{Dirichlet}(\alpha)$, or a distribution of topics over document i, and $\varphi_k \sim \text{Dirichlet}(\beta)$, or a distribution of words over topic k. Thus, LDA assumes that some generative model generates a document: for each word position i, j from $i \in$ documents and $j \in \text{words}_i$, choose a topic $z_{ij} \sim \theta_i$, then choose a word $w_{ij} \sim \varphi_{z_{ij}}$.

In this paper, LDA is extended: instead of finding distributions over topics for every document, topic distributions are learned over genres, where a genre is a non-overlapping set of documents. The difference between regular LDA and genre-LDA can be seen in the graphical models found in figure 1a (original LDA) and 1b (LDA extended over genres).

To infer the different distributions, collapsed Gibbs sampling is used (where the distributions θ_i and φ_k are integrated out). The derivation for genre-LDA is similar to, but not exactly the same as the derivation for original LDA. Both the derivations as well as the symbol descriptions for extended LDA can be found in section 7.1.



(a) Graphical model for regular LDA

(b) Graphical model for extended LDA

3.2.1 Gibbs sampling

The calculation of the total probability of the model is computationally difficult because it marginalizes by integrating over a joint distribution. Therefore, the collapsed Gibbs sampling algorithm is applied to the (extended) LDA model. This algorithm samples from a conditional distribution which asymptotically approaches the correct distributions and is computationally simple compared to the marginalization of the total probability. The formula for the conditional probability used in Gibbs sampling algorithm for the extended LDA model can be found in section 7.1. As can be seen

[5] For example, 'a', 'by', 'I', 'off', 'were'

[6] The Latin genre, for example, turns out to use both English and Spanish words quite a lot

in this formula, the conditional probability in the extended LDA model also depends on the counts of topics that occur in documents with the same genre instead of only the document in question (the latter being the case in the original LDA model). The algorithm for collapsed Gibbs sampling in the extended LDA model is described in algorithm 0.

```
Data: Topic Model inferred by extended DA, genre $G$, number of documents $N$, words in document i $M_i$
Result: word-topic and topic-genre distribution
Initialize: randomly assign words to topics
assign topics to genre given genre of doc in which word is found
for Document i where $i \in 1 \dots N$ do
    for word j where $j \in 1 \dots M_i$ do
        calculate conditional probability distribution for $z_{i,j}$
        sample topic from distribution $k$
        update word-topic distrubtion given $z_{ij} = k$
        update topic-genre distrubtion given $G$
    end
end
return word-topic distribution, topic-genre distribution
```

Algorithm 0: Gibbs sampling for extended LDA

3.3 Classification

In order to classify documents into their genres, a multi-class Support Vector Machine (SVM) classifier is implemented with the use of python module Scikit-Learn [7]. The SVM classifier is a state-of-the-art classification algorithm for supervised learning which constructs a hyper-plane in high dimensional spaces with largest distance to the nearest training data points of any class. In this research a multi-class SVM is used which incorporates a One-vs-All strategy where it fits one classifier for each class.

As input for the SVM, the topic distribution for each document in the training set is calculated and its class (i.e. genre) is passed to the multi-class SVM for training. Then, for each document in the test set, its topic distribution is computed, after which the classifier can predict its genre. The F1-score (section 3.3.1) is used to evaluate the predictions made by the classifier.

3.3.1 F1-score

The classifications are evaluated using the F1-score. The F1-score is a measure which takes into account the precision and recall scores of the model and can be seen as a weighted average. The precision score computes the number of true positives divided by the true and false positives of retrieved documents. The recall score computes the number of number of true positives divided by the all positive documents that should have been retrieved. The calculation of the F1 score is displayed below.

$$F1 = 2 * \frac{precision * recall}{precision + recall} \tag{1}$$

3.4 Generative model

Since LDA is a generative model, it can also be used to create new datapoints (lyrics, in this case). To create a ‘song’ of length n for a genre G, for each word position from 0 to n, a topic k is sampled from G’s topic distribution. Then, a word is sampled from k’s word distribution (see also algorithm 1).

```
Data: Topic Model inferred by extended LDA, genre $G$, length $n$
Result: Song for genre $G$ of length $n$
initialize: song = ‘’
while counter $< n$ do
    sample $k \sim \theta_G$
    sample $w \sim \varphi_k$
    append $w$ to song
    counter ++
end
return song
```

Algorithm 1: Song generation

4 Experiments/Empirical Evaluation

The experiments were run on a dataset consisting of 9.728 documents, automatically labeled by genre. Table 1 shows the number of documents in the dataset for each genre.
There are a few different parameters that may influence the accuracy of the classifier.These are α, the Dirichlet parameter for θ, β, the Dirichlet parameter for φ, the number of topics to be extracted from the dataset and the number of iterations for Gibbs sampling. For validation, k-fold cross-validation has been used with $k = 5$ (as described in section 4.1), to ensure an unbiased test.

Genre	Reggae	Latin	Holiday	Stage & Screen	Electronic	Children’s	Country	Jazz	Vocal	Easy Listening	New age	R&B	Avant-Garde	Pop/Rock	Folk	Rap	International	Blues	Comedy/Spoken	Religious	Classical	Total
# Documents	65	67	11	121	330	60	679	138	259	28	21	785	6	5470	129	720	186	101	35	470	47	9728

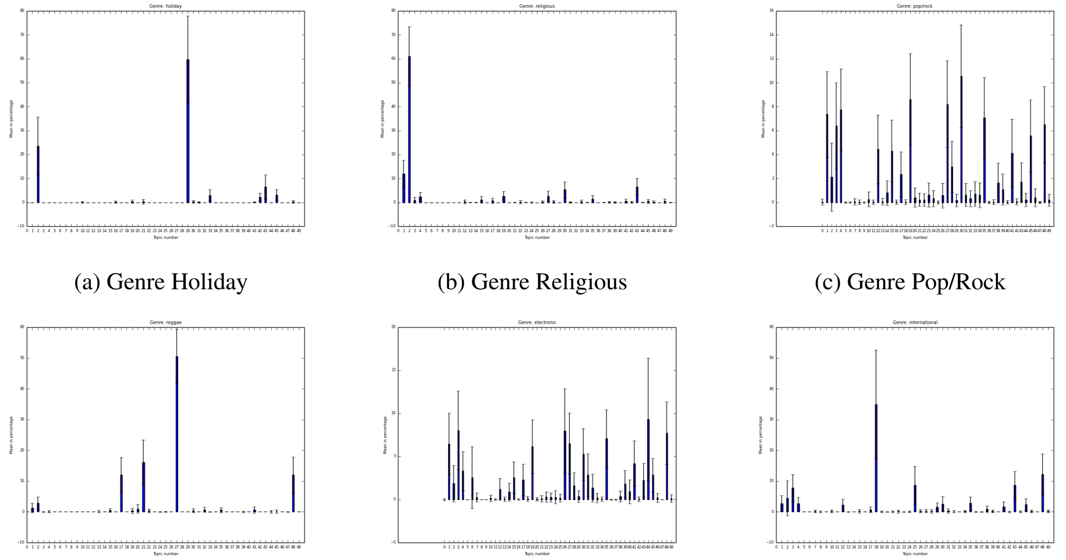
Table 1: Number of documents per genre

4.1 Cross-validation

Cross-validation is a validation technique used to asses how well the results of a given model will generalize. In this research, stratified 5-fold cross-validation was used. In 5-fold cross-validation, the dataset is split into 5 subsets of equal size and the algorithm is performed 5 times (or folds) in which each time, a different subset of the 5 subsets is used as validation data. Each experiment is thus run 5 times where the training set entails 80% of the total data set and the validation set entails 20% of the total dataset. Due to the variation in population for each genre in the dataset, a stratified sampling of training and validation set for the subsets is used. This sampling technique preserves the percentage of samples for each class in all subsets.

4.2 Results

Figure 2 presents genre profiles for different genres. Each profile is created by summing topic counts for every document in the genre, and averaging these. For certain genres, such as *Holiday*, *Religious* and *Reggae*, these profiles are strongly dominated by one or two topics. For other genres, such as *Electronic* or *Pop/Rock*, these profiles are more evenly distributed over topics. This is not surprising, as *Holiday* and *Religious* are genres named after the themes of their songs, instead of musical composition. Moreover, *Pop/Rock* contains music styles ranging from pop music to brutal death metal, resulting in a genre profile that covers a wide range of topics. Results of the classification contrasting the extended LDA and original LDA model can be found in table 2. These results show that by applying more topics to the extended model, the performance of the classifier increases. The classifiers have scores of around 50%. However, after closer inspection of the classifications, it was discovered that the classifiers almost always predicted the genre *Pop/Rock*. Since more than half of the dataset consists of data labeled with *Pop/Rock*, it is unsurprising that the classifier didn’t work too well.



(a) Genre Holiday

(b) Genre Religious

(c) Genre Pop/Rock

(d) Genre Reggae

(e) Genre Electronic

(f) Genre International

Figure 2: Topic distibution for different genres

Type of LDA	**Average Precision**	**Average Recall**	**Average F1**
Extended (50 topics)	0.48137806154784157	0.57399794450154162	0.46618418532122646
Extended (100 topics)	0.50310266811037774	0.59352517985611508	0.48576895832520584
Original (100 topics)	0.31608099763575387	0.56166495375128467	0.40451708979848938

Table 2: Classification including pop/rock genre, Topics = 50 or 100, alpha=0.1, beta=0.1

Since the genre *Pop/Rock* dominates the dataset, the same experiment was re-run, leaving out the data labeled *Pop/Rock*. [7] After removing documents labeled *Pop/Rock*, the dataset consisted of 4.258 documents. The results of running extended and original LDA without *Pop/Rock* documents can be found in table 3. These results show that extended LDA outperforms original LDA. Moreover, the most distinctive topics were *R&B* and *Rap*. Higher results were expected for *Holiday* and *Religious* and *Reggae*, due to their distinctive genre profiles. However, the size of *Holiday* and *Reggae* was relatively small (11 and 65 documents). On the other hand, *Religious* was a rather large part of the dataset. Further analysis showed that most documents belonging to the genre *Religious* were predicted to be *Country*, which is a larger subset of the dataset. Since *Country* is a genre associated with the conservative American South, it might share a lot of themes with *Religious* (allusions to God and Heaven, for example).

Folds	**1**	**2**	**3**	**4**	**5**	**Average**
avant-garde	0	0	0	0	0	0
blues	0	0.0011	0.0023	0	0	0.0007
children's	0	0	0.0012	0.0023	0	0.0007
classical	0	0	0.0023	0	0	0.0005
comedy/spoken	0	0	0.0023	0	0	0.0005
country	0	0.1357	0.0982	0.0983	0.0962	0.0857
easy listening	0	0	0	0	0	0
electronic	0	0.0093	0.0548	0.0681	0.0047	0.0274
folk	0	0	0.0023	0	0.0047	0.0014
holiday	0	0	0	0	0	0
international	0	0.0070	0.0117	0.0232	0.0163	0.0116
jazz	0	0	0.0023	0	0.0209	0.0046
latin	0	0	0.0023	0.0023	0	0.0009
new age	0	0	0	0	0.0023	0.0005
r&b	0.3112	0.1658	0.1275	0.1068	0.1379	0.1698
rap	0	0.2595	0.2487	0.2215	0.2435	0.1946
reggae	0	0	0.0023	0	0	0.0005
religious	0	0.0459	0.0614	0.0746	0.0571	0.0478
stage & screen	0	0	0	0.0023	0.0023	0.0009
vocal	0	0	0.0140	0.0023	0.0094	0.0051
weighted total	0.0573	0.2669	0.3153	0.2936	0.2879	0.2442

Table 3: Extended LDA - F1-score for 20 runs, 50 topics, alpha & beta = 0.1, no poprock

Folds	**1**	**2**	**3**	**4**	**5**	**Average**
avant-garde	0	0	0	0	0	0
blues	0	0	0	0	0	0
children's	0	0	0	0	0	0
classical	0	0	0	0	0	0
comedy/spoken	0	0	0	0	0	0
country	0	0.0346	0.2138	0.2508	0.0392	0.1077
easy listening	0	0	0	0	0	0
electronic	0	0	0	0.0023	0	0.0005
folk	0	0	0	0	0	0
holiday	0	0	0	0	0	0
international	0	0	0	0	0	0
jazz	0	0	0	0	0	0
latin	0	0	0	0	0	0
new age	0	0	0	0	0	0
r&b	0.3112	0.0459	0.0961	0.0046	0.2808	0.1477
rap	0	0	0	0	0	0
reggae	0	0	0	0	0	0
religious	0	0	0	0	0	0
stage & screen	0	0	0	0	0	0
vocal	0	0	0	0	0	0
weighted total	0.1023	0.2669	0.0887	0.0487	0.0800	0.1173

Table 4: Original LDA - F1-score for 20 runs, 50 topics, alpha & beta = 0.1, no poprock

Table 5: Extended versus original LDA

To test the effects of α and β, experiments were run to contrast different values for these parameters. The results of these experiments can be found in table 4.2. This table also displays the evaluation scores of the original LDA model[8] which are used as a baseline. The best F1-score is achieved using $\alpha = 0.5$ and $\beta = 0.1$ for the extended LDA model. This might be because a high α-value ensures overlapping topic distributions, resulting in a large mixture of different topics over genres. A low β-value creates less overlap in words over topics, creating high-information topics that are more distinctive.

alpha	**beta**	**Average Precision**	**Average Recall**	**Average F1**
0.1*	0.1*	0.0628	0.1740	0.0754
0.1	0.1	0.2899	0.3034	0.2441
0.5	0.1	0.3559	0.3273	0.2925
0.1	0.5	0.2597	0.2474	0.2010
0.5	0.5	0.2693	0.3074	0.2306

Table 6: Classification without pop/rock genre, Topics = 50, alpha=0.1, beta=0.1
$\star$: baseline

When running the algorithm with $\alpha = 0.5$, $\beta = 0.1$, and 50 topics, the distribution of the words over topics shows a clear split between different genres. After 20 runs, the results are as follows:

- Top topic for genre *Latin*:

```
want, bad, que, sleep, bla, got, la, tu, cant, y, loco, te, de, lets, jiyoon, alabao,
bongo, por, king, macarena, yo, night, quiero, haebwa, anjullae
```

[7] This only concerns the experiment with 50 topics due to time constraints.

[8] The baseline classifier for original LDA, with $\alpha = 0.1$ and $\beta = 0.1$

A distinct feature of the *Latin* genre is the combination of English en foreign, mostly Spanish, words. This topic demonstrates that combination, since it contains both English words like *bad*, *sleep* and *king*, but also Spanish words like *quiero*, *que* and *macarena*.

- Top topic for genre *Holiday*:

  ```
  la, star, fa, o, noel, closer, oli, reindeer, night, ye, moving, ije, pearl, mal, rudolph,
  guide, ho, wowork, light, hago, annabella, israel, inch, muskura, royal
  ```

  The holiday genre is dominated by Christmas songs, which can clearly be seen from this topic. It contains words which are typical for Christmas, like noel, reindeer and rudolph. With Christmas being a Christian holiday, the word israel also belongs in this topic, since Israel is closely related to certain Bible stories.

- Top topic for genre *Rap*:

  ```
  st, na, got, fk, bh, ns, nigga, yall, back, aint, hook, yo, wanna, get, shit, black, verse,
  work, bang, like, check, life, fuckin, gotta, em
  ```

  The rap genre is dominated by stereotypical *Rap* words: a lot of swearing and slang.

- Top topic for genre *Religious*:

  ```
  god, oh, know, lord, love, let, like, life, chorus, jesus, see, never, heart, praise,
  take, go, things, way, us, grace, wanna, verse, give, one, yeah
  ```

  The religious genre contains songs solely based upon praising God, Jesus and everything about the Bible, which can be seen from this topic since it contains typical words like god, lord, jesus, praise and grace.
  However, not every topic is representative for its genre.

- This is demonstrated by the assignment of the following topic to the genre *Classical*:

  ```
  love, oh, know, baby, yeah, feel, wanna, girl, right, see, come, make, want, give,
  tonight, never, let, way, cause, take, ooh, time, chorus, gonna, life
  ```

  A lot of these words are not representative for classical music and are more likely to originate from lyrics with genres other than classical.

- Furthermore, the genre *Electronic* was, among others, represented by the following topic:

  ```
  love, heart, oh, find, never, night, got, alone, always, need, back, see, eyes, christmas,
  mine, way, tell, like, one, time, comes, say, think, waiting, kiss
  ```

  Words such as *heart*, *always*, *christmas* and *kiss* are words that are not representative for the genre *Electronic*.

Original LDA outputs different topics than extended LDA. For example, after 20 iterations, the top words of the top topic assigned to *Reggae* were

```
yeah, get, got, right, man, like, little, time, say, hey
```

using regular LDA, whereas extended LDA assigned

```
mi, fight, police, gonna, say, jah, whatcha, dem, ya, yuh, burnin, ah
```

This probably causes the better performance of the classifier trained on the extended LDA model. The differences between the F1 scores of regular LDA and extended LDA can be seen in table 5
Furthermore, the results show that genres with a large role for vocals (thus, genres where lyrics are important), like *Holiday*, *Rap* and *Religious* tend to have a higher F1-score than genres containing fewer lyrics, like *Classical* and *Electronic*. This may have two causes: on one hand, these genres have more lyrics and thus a richer dataset, and on the other, their lyrics may also be more distinct for the genre.
As a demonstration of the generative capabilities of the extended LDA model, algorithm 1 has been used to generate a *Rap* ‘song’ of 40 words. The output of this algorithm is shown in figure 3.

cause leader sun nigga bastard push verse
end account high mountain ballin trash yall
pen tears writing instrumental gucci like chorus
rule best clean legit line curtains way heroin
proven nothin yo find 1 aint oh
ride every rising gambino

Figure 3: Song generated in genre *rap* with $\alpha = \beta = 0.7$, 20 topics and 30 iterations

5 Discussion and Conclusions

5.1 Challenges

The main challenge in this research was caused by the dataset: more than half of the dataset is labeled as *Pop/Rock*, a label that unites both easy-to-listen top-40 songs and brutal death metal songs. The diversity of this genre makes it quite hard to classify, since such a broad genre contains a lot of different words and therefore a pretty even distribution over topics.

Another challenge was the fact that some words, such as *love*, occur often in a lot of different genres. Words that are common in many different genres make topic distributions more even, and are thus low-information words. These should be filtered from the dataset along with other low-information words (such as ‘girl’ and ‘baby’).

5.2 Conclusion

Concluding, the extended version of LDA as proposed in this paper forms sensible topics for different genres, especially those genres named after their lyrical themes. However, due to the very uneven distribution of documents over genres (some genres dominated the dataset by sheer number), classification did not work as well as expected. Extended LDA did outperform regular LDA, and is thus a better choice for genre classification tasks, which could also be applied to, for example, literary genres.

The question researched in this paper cannot be answered conclusively: preliminary results show that the classifier does not perform very well, even for the better version of LDA. However, it is assumed that this underperformance is largely due to a poorly labeled dataset.

5.3 Future work

If reseach is continued with the current dataset, the *Pop/Rock* genre has to be split into more (sub)genres. This could be achieved by taking the topic distribution of subgenres, and merging subgenres containing similar topic distributions (for example - *Heavy Metal* and *Speed Metal* are usually not far apart thematically). This way, subgenres like *Death Metal* and *Punk Rock* should be separated from, for example, top 40 songs.

Furthermore, words that occur a lot in different genres should be removed from the dataset. A proposed method for achieving this is by storing the number of occurrences of a word for each genre, and if the word occurs in more than a threshold percentage of the genres, remove it from the dataset.

Another proposed addition to this research is to perform a 10-fold cross-validation (instead of 5-fold cross-validation) and run it over multiple experiments. Due to time constraints, this extensive measuring was not within the scope of this research, but it can strengthen the conclusions drawn from the evaluations of the classifications.

More tests have to be run, especially tests where more topics are extracted. Since the experiment with $\alpha = 0.5$ and $\beta = 0.1$ provided the best results for different parameter settings, and the experiment with 100 topics outperformed experiments with less topics, it is expected that a run with 100 topics instead of 50 with α set to 0.5 and β set to 0.1 will yield even better results.

6 Team responsibilities

Component	**Name**
Writing crawler, implementation, report	Elise
Collecting data, implementation, report	David
Pre-processing, derivations for LDA, implementation, report	Sharon

References

[1] Michael Fell and Caroline Sporleder. Lyrics-based analysis and classification of music.

[2] Ricci Kaminskas. Contextual music information retrieval and recommendation: State of the art and challenges.

[3] Ogihara Li. Music artist style identification by semi-supervised learning from both lyrics and content.

[4] Alen Lukic. A comparison of topic modeling approaches for a comprehensive corpus of song lyrics.

[5] Maxwell. Exploring the music genome: Lyric clustering with heterogeneous features.

[6] Rauber Mayer, Neumayer. Combination of audio and lyrics features for genre classification in digital audio collections.

[7] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[8] George Tzanetakis and Perry Cook. Musical genre classification of audio signals. *Speech and Audio Processing, IEEE transactions on*, 10(5):293–302, 2002.

7 Appendices

7.1 Appendix A. Gibbs sampling for extended LDA

Total probability of the extended LDA model:

$$
\begin{align}
P(w, z, \varphi, \theta | \alpha, \beta, y) = P(\varphi|\beta)P(\theta|\alpha)P(z|\theta, y)P(w|\varphi, z) \tag{2} \\
\tag{3}
\end{align}
$$

The formulas for the separate probabilities are given here:

$$
\begin{align}
P(\varphi|\beta) &= \prod_{k=1}^{K} P(\varphi_k|\beta) \tag{4} \\
P(\theta|\alpha) &= \prod_{g=1}^{G} P(\theta_g|\alpha) \tag{5} \\
P(z|\theta, y) &= \prod_{i=1}^{N} \prod_{j=1}^{M_i} P(z_{i,j}|\theta_{y_i}) \tag{6} \\
P(w|\varphi, z) &= \prod_{i=1}^{N} \prod_{j=1}^{M_i} P(w_{i,j}|\varphi_{z_{i,j}}) \tag{7} \\
\tag{8}
\end{align}
$$

Fill the separate probabilities into the model formula:

$$
P(w, z, \varphi, \theta | \alpha, \beta, y) = \prod_{k=1}^{K} P(\varphi_k|\beta) \prod_{g=1}^{G} P(\theta_g|\alpha) \prod_{i=1}^{N} \prod_{j=1}^{M_i} P(z_{i,j}|\theta_{y_i}) P(w_{i,j}|\varphi_{z_{i,j}}) \tag{9}
$$

Integrate φ and θ out of the total probability:

$$
\begin{align}
\int\int P(w, z, \varphi, \theta | \alpha, \beta, y) d\varphi d\theta &= \int \prod_{k=1}^{K} P(\varphi_k|\beta) \prod_{i=1}^{N} \prod_{j=1}^{M_i} P(w_{i,j}|\varphi_{z_{i,j}}) d\varphi \tag{10} \\
&\times \int \prod_{g=1}^{G} P(\theta_g|\alpha) \prod_{i=1}^{N} \prod_{j=1}^{M_i} P(z_{i,j}|\theta_{y_i}) d\theta \tag{11}
\end{align}
$$

Since all θ's and φ's are independent to each other, we can derive these integrals separately.

1) Derivation of $\int \prod\limits_{k=1}^{K} P(\varphi_k|\beta) \prod_{i=1}^{N} \prod_{j=1}^{M_i} P(w_{i,j}|\varphi_{z_{i,j}}) d\varphi$
Use $C(k, w)$ as the number of times word w is assigned to topic k in any document.

$$
\prod_{k=1}^{K} P(\varphi_k|\beta) \prod_{i=1}^{N} \prod_{j=1}^{M_i} P(w_{i,j}|\varphi_{z_{i,j}}) d\varphi \tag{12}
$$

$$= \prod_{k=1}^{K} \int \frac{\Gamma(\sum\limits_{w=1}^{V} \beta)}{\prod\limits_{w=1}^{V} \Gamma(\beta)} \prod_{w=1}^{V} \varphi_k(w)^{\beta-1} \prod_{i=1}^{N} \prod_{j=1}^{M_i} \varphi_{z_{i,j}}(w_{i,j}))d\varphi \tag{13}$$

$$= \prod_{k=1}^{K} \int \frac{\Gamma(V\beta)}{(\Gamma(\beta))^V} \prod_{w=1}^{V} \varphi_k(w)^{\beta-1} \prod_{w=1}^{V} \varphi_k(w)^{C(k,w)} d\varphi \tag{14}$$

$$= \prod_{k=1}^{K} \int \frac{\Gamma(V\beta)}{(\Gamma(\beta))^V} \prod_{w=1}^{V} \varphi_k(w)^{\beta+C(k,w)-1} d\varphi \tag{15}$$

$$= \prod_{k=1}^{K} \frac{\Gamma(V\beta)}{(\Gamma(\beta))^V} \frac{\prod\limits_{w=1}^{V} \Gamma(C(k,w)-1)}{\Gamma(\sum\limits_{w=1}^{V} C(k,w)-1))} \int \frac{\Gamma(\sum\limits_{w=1}^{V} C(k,w)-1))}{\prod\limits_{w=1}^{V} \Gamma(C(k,w)-1} \prod_{w=1}^{V} \varphi_k(w)^{\beta+C(k,w)-1} d\varphi \tag{16}$$

$$= \prod_{k=1}^{K} \frac{\Gamma(V\beta)}{(\Gamma(\beta))^V} \frac{\prod\limits_{w=1}^{V} \Gamma(C(k,w)-1)}{\Gamma(\sum\limits_{w=1}^{V} C(k,w)-1))} \tag{17}$$

$$\propto \prod_{k=1}^{K} \frac{\prod\limits_{w=1}^{V} \Gamma(C(k,w)-1)}{\Gamma(\sum\limits_{w=1}^{V} C(k,w)-1))} \tag{18}$$

2) Derivation of $\prod\limits_{g=1}^{G} P(\theta_g|\alpha) \prod_{i=1}^{N} \prod_{j=1}^{M_i} P(z_{i,j}|\theta_{y_i}) d\theta$

Use $C(g,k)$ as the number of times topic k is assigned to genre g in any document.

$$\prod_{g=1}^{G} P(\theta_g|\alpha) \prod_{i=1}^{N} \prod_{j=1}^{M_i} P(z_{i,j}|\theta_{y_i}) d\theta \tag{19}$$

$$= \prod_{g=1}^{G} \int \frac{\Gamma(\sum\limits_{k=1}^{K} \alpha)}{\prod\limits_{k=1}^{K} \Gamma(\alpha)} \prod_{k=1}^{K} \theta_g(k)^{\alpha-1} \prod_{i=1}^{N} \prod_{j=1}^{M_i} \theta_{y_i}(k_{i,j}))d\theta \tag{20}$$

$$= \prod_{g=1}^{G} \int \frac{\Gamma(K\alpha)}{(\Gamma(\alpha))^K} \prod_{k=1}^{K} \theta_g(k)^{\alpha-1} \prod_{k=1}^{K} \theta_g(k)^{C(g,k)} d\theta \tag{21}$$

$$= \prod_{g=1}^{G} \int \frac{\Gamma(K\alpha)}{(\Gamma(\alpha))^K} \prod_{k=1}^{K} \varphi_k(w)^{\alpha+C(g,k)-1} d\theta \tag{22}$$

$$= \prod_{g=1}^{G} \frac{\Gamma(K\alpha)}{(\Gamma(\alpha))^K} \frac{\prod\limits_{k=1}^{K} \Gamma(C(g,k)-1)}{\Gamma(\sum\limits_{k=1}^{K} C(g,k)-1))} \int \frac{\Gamma(\sum\limits_{k=1}^{K} C(g,k)-1))}{\prod\limits_{k=1}^{K} \Gamma(C(g,k)-1} \prod_{k=1}^{K} \varphi_k(w)^{\alpha+C(g,k)-1} d\theta \tag{23}$$

$$= \prod_{g=1}^{G} \frac{\Gamma(K\alpha)}{(\Gamma(\alpha))^K} \frac{\prod\limits_{k=1}^{K} \Gamma(C(g,k)-1)}{\Gamma(\sum\limits_{k=1}^{K} C(g,k)-1))} \tag{24}$$

$$\propto \prod_{g=1}^{G} \frac{\prod\limits_{k=1}^{K} \Gamma(C(g,k)-1)}{\Gamma(\sum\limits_{k=1}^{K} C(g,k)-1))} \tag{25}$$

Calculate the collapsed Gibbs sampling
For the collapsed Gibbs sampling the conditional probability $P(z_{i,j} = k | Z_{\neg w_{i,j}}, \alpha, \beta, W, Y)$ is used.
Use $\neg w_{i,j}$ as collection that does not include the jth word in document i. Use $\widetilde{C}$ as a count that does not include the jth word in document i.

$$P(z_{i,j} = k | Z_{\neg w_{i,j}}, \alpha, \beta, W, Y) \propto \frac{\beta + \widetilde{C}(k, w_{i,j})}{V\beta + \widetilde{C}(k)} \times \frac{\alpha + \widetilde{C}(y_i, k)}{K\alpha + \widetilde{C}(y_i)} \tag{26}$$

7.1.1 Symbol description

SYMBOL	DESCRIPTION
K	# of topics
G	# of genres
N	# of documents
M_i	# of words in document i
y_i	genre of document i
$w_{i,j}$	word in document i at position j
$z_{i,j}$	topic assignment for word $w_{i,j}$
θ_g	probabilities of topics given genre g
φ_k	probability of words given topic k
α	Dirichlet prior for genre distribution
β	Dirichlet prior for topic distribution