ENGINEERING SERENDIPITY IN MUSIC RECOMMENDATION USING LDA TOPIC MODELLING

Thomas Nuttall
Music Technology Group, Universitat Pompeu Fabra, Barcelona
thomasnuttall01@estudiant.upf.edu

ABSTRACT

Of particular interest in the development of modern recommenders is the introduction of diverse, unexpected and unusual suggestions. Here we present a novel approach to music recommendation by means of latently representing album reviews using LDA topic modelling. Modern feature extraction techniques are ensured and our model is tuned and trained on the MARD dataset. We empirically evaluate our model using a blind AB test against a chance recommender. Comparison to current state-of-the-art is also included.

1. RELATED WORK

Here we use the principles outlined in [5] to fuel our serendipitous approach. Taramigkou et al. applies LDA topic modelling to user preference data on Last.fm to learn distinct patterns in listener behaviour and then uses this latent representation to suggest new artists to a group of 25 music lovers, ultimately producing a result that listeners on average agreed helped them find artists they wouldn't have found easily on their own and which they would like to listen to from now on.

User preference data however is not always available to recommenders and is indeed the issue underlying the cold-start problem in recommendation. Here we apply the same techniques as Taramigkou to textual album review data and compare the serendipity of our results to this study.

2. METHODOLOGY

Our framework is implemented in Python using the `gensim`, `nltk` and `spacy` libraries. Marked improvements in efficiency are observed when using Pythons `multiprocessing` for textual processing and `gensims` multi core implementation of LDA for modelling. [1]

2.1 Dataset

We use the MARD dataset [7] for our analysis. A collection of 265,525 Amazon reviews of 65,566 albums. The dataset is enriched with metadata from MusicBrainz and AcousticBrainz.

For our analysis we are concerned only with album name and the textual review.

2.2 Text Processing

Each review is subject to 3 processing steps.

2.2.1 Data Cleaning

- Apostrophes are removed
- Words are forced to lower case
- Stopwords are removed using `nltks` stop word list
- Words of less than 3 characters are removed
- Words not in the dictionary are removed (as queried against `nltks` English dictionary)

2.2.2 Compute Word Grams

Words that appear frequently together are grouped, the maximum permitted in one group is 3. For this we use `gensims` n-gram model with a minimum count of 5 and threshold of 100. These parameters are tuned by trial and error. Example's of words joined are soothes soul (to soothes-soul) and highly recommended (to highly-recommended).

2.2.3 Lemma Computation

Lemmatization converts a word to it's root word. For example beats becomes beat, wobbling becomes wobble. `spacy` has functionality for this.

After processing, album reviews are represented as a bag of words and grouped together such that each album is represented by one bag of all of it's processed reviews combined.

2.3 Modelling

For our modelling we build a reproducible experiment framework that trains multiple LDA models and stores the results in a CSV under a unique run id generated using Python's UUID. The modelling pipeline can be summarized in 3 steps.

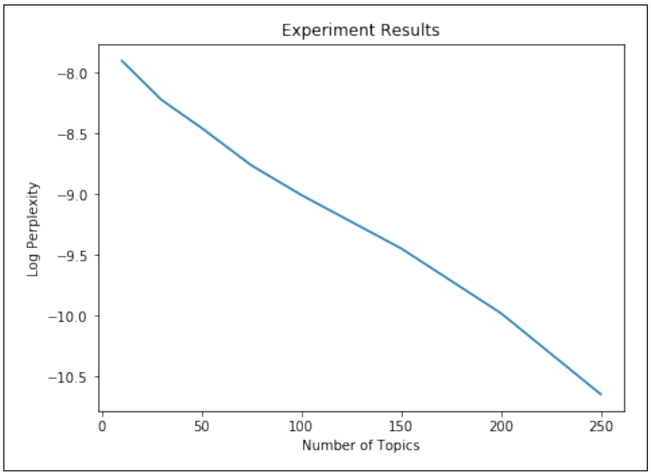

Figure 1. Log Perplexity and Number of Topics

2.3.1 Train/Test Split

Our data is split into train and test of proportions 80% and 20% respectively. This provides us with 52,452 reviews to train on and 13,113 reviews to test on.

2.3.2 Train Multiple Models

A model is trained to identify varying number of topics, k. Values of k experimented with are [0, 30, 50, 75, 100, 150, 200, 250]. We use an online variational bayes optimizer [4] on 20 passes of the data with a maximum of 1500 iterations for each chunk.

2.3.3 Compute Log Perplexity

For each run we calculate the log perplexity [3] on our test dataset in a hope to identify a natural choice of k. Figure 1 demonstrates that no such natural choice exists.

With no analytical justification for our choice for number of topics we select our value manually by inspecting our topics and making a judgement on whether we believe sufficient information is captured, this is common practice for in training LDA models and is in fact the method selected by Taramigkou et al in [5]. Our final model describes 150 topics.

3. MODEL

It is impractical to outline here all 150 topic distributions. Instead we select a few important topics and include alongside this paper our final model with code to scrutinise it more thoroughly.

Figures 2-4 demonstrate 3 interesting topics. Hopefully the reader can identify semantic trends in music description that transcend genre/traditional music classification methods. It is by representing our albums with these latent patterns that we hope to make interesting and novel recommendations.

4. RECOMMENDATION

We use our trained model to predict the topic distribution of each review in our original corpus, hence representing every album as a vector of 150 features. We compute the cosine similarity between each pairwise combination of al-

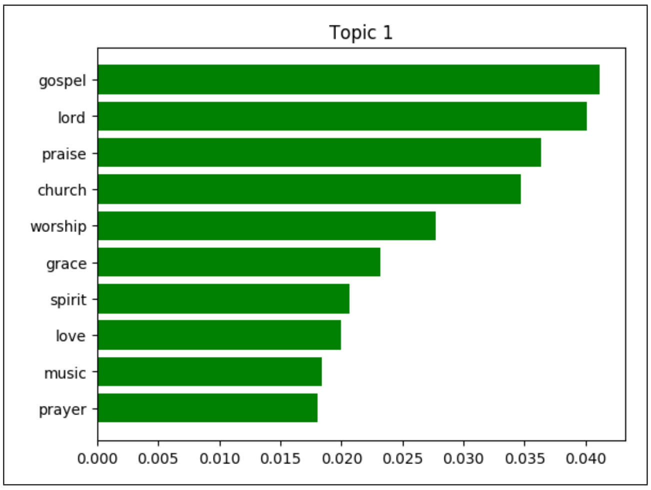

Figure 2. Term Distribution for Topic 1

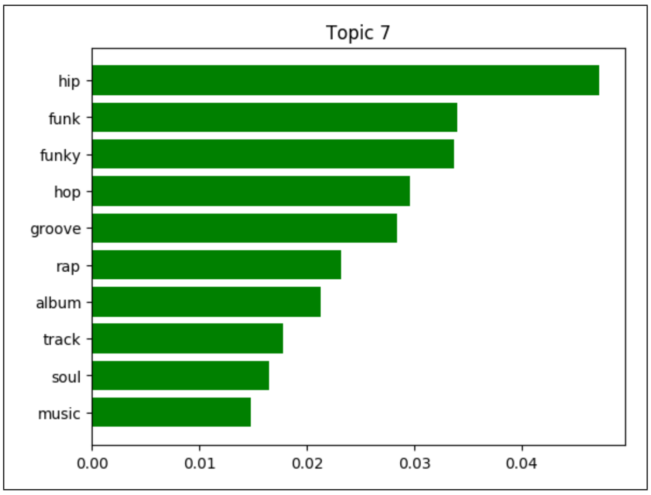

Figure 3. Term Distribution for Topic 8

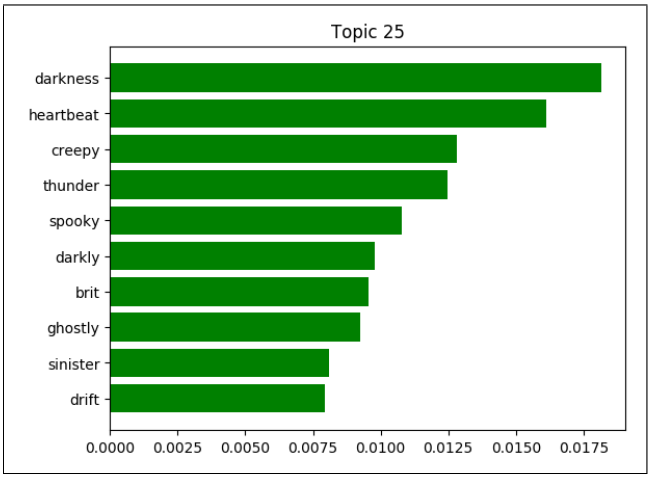

Figure 4. Term Distribution for Topic 25

bum reviews and by doing so create a ranked list of most similar albums for each data point. For a given album we can now recommend top n most similar albums.

We notice that our recommender does a good job at suggesting recommendations within the genre of the target album. This is traditionally considered a good sign however somewhat impedes us on our quest for serendipity. Particularly insular genres are Classical, Latin Music and Metal. Albums of the Hip Hop or R&B genre tend to generate good recommendations outside of their own genre, this is desirable.

The following sub sections outline the top 5 recommendations for a selection of 3 albums.

4.1 Good Recommendation

This is typical of most of our recommendation, we are returned similar albums in the genre. Good but not interesting.

The Beatles - The Beatles [Rock]

Pink Floyd - Wish You Were Here [Rock]
The Beatles - Abbey Road [Rock]
The Beatles - The Beatles [Rock]
Pink Floyd - The Final Cut [Rock]
Pink Floyd - The Wall [Rock]

4.2 Bad Recommendation

Here our model is confused, it has noticed similarity in that all albums are compilations, actually this would be a bad recommendation.

Various Artists - Soul Satisfaction 4 [R&B]

Gilbert & Sullivan - Greatest Hits [Pop]
Sonora Carruseles - Boogaloo Is Here [Latin Music]
Washboard Riddem - Washboard Riddem [Reggae]
Various Artists - Super Hits of The 80s [Pop]
The Felix Culpa - Thought Control [Alternative Rock]

4.3 Serendipitous Recommendation

This is what we are aiming for. Alice Coltrane's World Galaxy is string focused (both bowed and plucked), rather ambient and cosmic - all qualities that can be found in the recommendations for it.

Alice Coltrane - World Galaxy [Jazz]

Andrew Violette - Sonata for the Creation of the World [Classical]
Yelena Eckemoff - Flying Steps [Jazz]
Kori Linae Carothers - Trillium [New Age]
Esa-Pekka Salonen - Wing on Wing [Classical]
Ken Townshend - Gentle Beauty [New Age]

5. EVALUATION

We evaluate our recommender for both serendipity and general recommendation.

5.1 Bayesian AB Testing

With no obvious baseline to compare too for general recommendation, we evaluate our model with an AB test against a chance recommender, that is a recommender that selects 5 albums randomly and offers them as a recommendation. The tester chooses which set of recommendations are best for a given track and the results are stored.

150 sets of recommendation were compared. Our serendipitous recommender won 133 comparisons with chance winning 27. A Bayesian AB test is performed and determines that our recommender is 8.8 times better than a chance recommender with 100% probability. [6]

5.2 Comparison to State of the Art

It is perhaps not surprising that our recommender outperforms chance, after all, even the most basic semantic connection is likely to offer better results than random picking. We wish to compare also to Taramigkou et al. in [5]. Taramigkou does not use her findings for recommendation explicitly, she instead presents the learnt latent space to a group of 25 participants who evaluate the serendipity of their observations by answering the following question on a five point Likert scale [2]:

> *"Did you find artists you wouldnt have found easily on your own and which you would like to listen to from now on?"*

We too evaluate our learnt representation by answering this same question for 100 recommendations. Our group size is 4 (we would like to improve this in future with increased time and resources).

Our final result was a score of 2.2 (std, 1.3). Considerably worse than [5] who's result was a score of 3.8 (std, 0.8).

6. LIMITATIONS

Due to the reviews being customer submitted from Amazon, many of them are structured poorly, contain errors or are just plain uninformative. It is only a very small proportion of them that use the sort of descriptive language required to learn useful semantic trends. It would be interesting in future apply this approach to professional critical reviews.

Another drawback is that our model is not able to distinguish between desirable and undesirable trends, for example, matching albums because they are both compilations is sub-ideal, matching them because they are both described as *dark* or *technical* is desirable. Potentially hand selecting which topics to prioritise might help with this in future.

7. CONCLUSION

We have demonstrated that it is indeed possible to build a significantly better than chance recommender on music reviews alone. Though that it is almost certainly not a sensible final solution for music recommendation. Subject to the improvements outlined in Limitations, a useful application of this methodology could be found in combining the results with more traditional techniques like collaborative filtering or audio-based approaches as a means of injecting diversity, or even serendipity into the final offering!

8. REFERENCES

[1] Michael I. Jordan D. M. Blei, Andrew Y. Ng. Latent dirichlet allocation. In *Journal of Machine Learning Research 3*, pages 993–1022, 2003.

[2] C. A. Seaman I. E. Allen. Likert scales and data analyses. In *Quality Progress, 40*, pages 64–65, 2007.

[3] C. A. Seaman I. E. Allen. Hlt '10 human language technologies: The 2010 annual conference of the north american chapter of the association for computational linguistics. In *Quality Progress, 40*, pages 100–108, 2010.

[4] F. Bach M. D. Hoffman, David M. Blei. Online learning for latent dirichlet allocation. 2010.

[5] K. Christidis D. Apostolou G. Mentzas M. Taramigkou, E. Bothos. Escape the bubble: Guided exploration of music preferences for serendipity and novelty. In *7th ACM conference on Recommender systems*, pages 335–338, 2013.

[6] Evan Miller. *Formulas for Bayesian A/B Testing*. `https://www.evanmiller.org/bayesian-ab-testing.html`, 2015.

[7] Lawlor A. Serra X. Saggion H. Oramas S., Espinosa-Anke L. Exploring customer reviews for music genre classification and evolutionary studies. 17th international society for music information retrieval conference (ismir'16). 2016.