A Probabilistic Model For Recommending Music Based on Acoustic Features and Social Data

Rodrigo C. Borges[1] **, Marcelo Queiroz**[1*]

[1]Grupo de Computação Musical - IME - USP
Av. Prof. Luciano Gualberto, 158, tv. 3 – 05508-900 São Paulo, SP

rcborges@ime.usp.br, mqz@ime.usp.br

Abstract

The "Cold Start" problem is a well known issue in Collaborative Filtering recommendation systems, associated to the moment when a new item or user is added to a given collection, because the system has no historical information of interaction between existing and new elements and it still need to incorporate these elements into the recommendation algorithm. This work addresses one possible solution for the case where new songs are added to a dataset of a music recommendation system, by proposing a probabilistic model for inference based on the songs' acoustic/timbre features. This model was first proposed for tagging music with semantic labels but is here suggested as being suitable for predicting user interactions with new songs. The experiments were conducted using a selection of Brazilian popular music and the results show that the proposed method compares favorably to Logistic Regression.

1. Introduction

Recommending music automatically has became a popular issue in the last decades since a huge amount of digital media became available as on-line services [1, 2]. The most common technique used today for this purpose is called Collaborative Filtering [3], and it works by matching similar user listening profiles. If user A listened to a song that user B with a similar listening profile hasn't, it is assumed that there is a high probability that user B would react positively to this song. But this approach has a weakness known as the "cold start" problem, which corresponds to a new song with no records of having being listened by existing users or a new user with no records of having listened to existing songs.

One of the possible solutions for this problem is to combine acoustic features extracted from the songs and past listening behavior, hopefully finding a statistical representation of the timbre content of the listened songs [4] that would somehow correlate with listening habits. If a new song is added to the set and its acoustic content is close enough to what has been learned by the recommender as part of a user's listening habit, then it might be considered as a recommendation candidate.

In this paper we apply a probabilistic model named Codeword Bernoulli Average Model [5] for predicting listening behaviors to new songs. This model was first proposed for tagging music with semantic labels, and attempts to predict the probability that a binary tag applies to a song, based on a vector-quantized representation of that song's audio (Figure 1). This is achieved through automatically learning a latent variable that represents some statistical relationship between the audio and the tag, which in this case are timbre representations and user listening behaviors.

This text is structured as follows. We start briefly presenting previous probabilistic models proposed for the same problem. The dataset used in our experiment is then described, namely the songs presented to the listeners and from which acoustic features were extracted. Then we present the experimental methodology, detailing the procedure of collecting listening data, the application used and how this information was stored. Feature extraction is explained in detail as well as the vector-quantized representation used for the model. The Codeword Bernoulli

*The second author acknowledges funding received from CNPq.

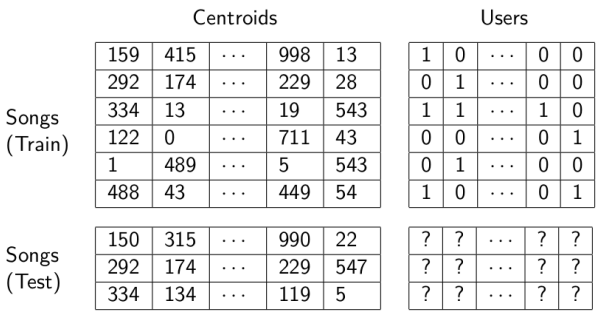

Figure 1: An Illustration of the problem of retrieving listening data from acoustic features.

Average Model is then briefly described, and we discuss how to apply it to sparse listening data. The evaluation of the model is presented next, along with a comparison to a Logistic Regression baseline, discussing how far this approach can be further explored.

2. Related Work

Some efforts on using acoustic features combined with collaborative filtering were already reported, some of them applying probabilistic modeling. Yoshii et al. [6] has proposed using GMM for representing the MFCC information, and also an e-commerce interaction database as corresponding to social data. A group of latent variables were proposed as corresponding to genres, among which the user would choose, and from which a piece of music was stochastically selected. Pseudo-genres are considered as providing recommendation diversity but also as differing from the kind of prediction desired here.

Campos et al. [4] propose a topological model based on Bayesian networks from where the degree of each recommending technique was automatically selected. This model could operate exclusively as collaborative-filtering or as content-based, using it for finding good items as well as for predicting user ratings.

Codeword Bernoulli Average Model was first presented in 2009 as a technique for automatically tagging music [5]. In this context the challenge was separated in two parts: annotating music that has no associated tags, and retrieving songs from a given tag. In both cases a subset should be returned sorted by relevance of the returned items, where the ordering had no relationship whatsoever with the binary tags.

3. Data Set

The dataset we used in the experiment is composed of 1199 Brazilian popular songs taken from a selection known as "*100 best records of Brazilian music*" [7], published in 2007 by the specialized music magazine Rolling Stone and representing the opinions of 60 music researchers, producers and journalists based on how influential they thought these records were to others artists. The recordings release dates vary from 1950 to 2003, which configures a heterogeneous group of music examples that should result in considerably different listening behavior patterns.

4. Listening Data

The listening data was collected between March and May 2017, having 10 listeners with ages varying from 25 to 60 years old. An Android application (Figure 2) was developed specifically for this experiment and when initialized, it selected randomly any song from the dataset and started playing. The user could listen to it or jump to the next song. This resulted in a sparse matrix counting how many times each user listened to each music until its end. It should be made clear that a count of 0 could either mean that the user was never exposed to a song, or that she or he skipped the song before reaching its end.

There were around 1000 complete song reproductions during this period, but only four listeners listened to a reasonable sample of the whole dataset (at least 10% of the number of songs), and for this reason these were the only subjects considered in the analysis.

The listening counting matrix was used to define a binary matrix representing which user had listened to which song to the end (regardless of how many times). This matrix relates users and songs through a binary correspondence that might be used as indicative of a user's willingness to hear to a song; again a value of 0 should

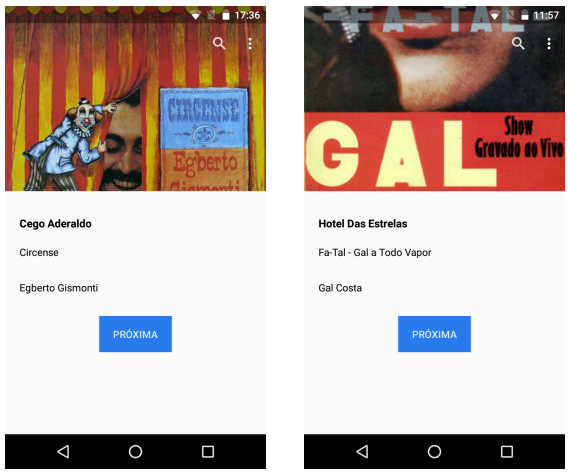

Figure 2: The application for collecting listening data.

not be understood as any negative disposition between a user and a song, because the user might not have been given the chance of listening to that song. This ambiguity is considered in the discussion and will be addressed in future work.

5. Feature Extraction

The MFCC acoustic descriptor was considered as a suitable representation for timbre aspects of songs, and was extracted using an open source Python library called Librosa [1]. The MFCC data was extracted with 13 coefficients, using windows of 2048 samples and 75% overlap between windows. As the number of windows depend on the duration of the song, and we needed a uniform representation for the dataset, we adopted a Vector Quantized Representation to solve this problem.

5.1. Vector Quantized (VQ) Representation

VQ is a technique first applied in data compression that considers a group of selected items for building a general feature space, building histograms to describe a large collection according to the number of members best described by each selected item. For this specific case, VQ: (i) takes all MFCC data extracted from every song in the database and define K centroids using Kmeans; (ii) takes each song and verify for each of its MFCC frames which of the K centroids is closest to it; (iii) produces a fixed size histogram for each song, representing how many MFCC frames are best described by each centroid.

One thing that it is worth mentioning is that the feature space is defined by all MFCC vectors extracted from the dataset, and how these MFCCs are clustered in K centroids. When a new song is added to the dataset, this feature space should be recalculated, possibly defining a new group of centroids and corresponding histograms for each song.

6. Codeword Bernoulli Average (CBA) Model

Our CBA model assumes a collection of binary random variables $\mathbf{y}$ with $y_{ju} \in \{0, 1\}$, indicating whether or not user u has listened to song j. The goal is to estimate a set of values for a Bernoulli parameters β that will maximize the likelihood $p(\mathbf{y}|\mathbf{n}, \beta)$ of the listened song associated to the VQ centroids counts $\mathbf{n}$ and the parameters β. It uses the Expectation Maximization (EM) algorithm for maximum likelihood estimation. Each EM iteration has two steps: first the Expectation that corresponds to:

$$h_{juk} = \begin{cases} \dfrac{n_{jk}\beta_{ku}}{\sum_{i=1}^{K} n_{ji}\beta_{iu}} & if \quad y_{ju} = 1 \\ \dfrac{n_{jk}(1-\beta_{ku})}{\sum_{i=1}^{K} n_{ji}(1-\beta_{iu})} & if \quad y_{ju} = 0, \end{cases} \tag{1}$$

followed by the Maximization step which corresponds to:

$$\beta_{ku} = \frac{\sum_j h_{juk} y_{ju}}{\sum_j h_{juk}}. \tag{2}$$

EM should stop iterating when the difference between two consecutive β matrices reaches a threshold value. When this happens we have found a β under which the training data has become more likely.

This results in a β matrix with dimensions given by the number of users (U) and the number of centroids (K), representing the statistical relationship between users and centroids through a Bernoulli distribution.

[1] https://github.com/librosa/librosa

6.1. Generalization

The result can then be generalized to new songs by simply multiplying its VQ representation vector by the corresponding column from the β matrix. The probability of a new song being heard given its feature vector is given by:

$$p(y_{ju} = 1|\mathbf{n}_j, \boldsymbol{\beta}) = \frac{1}{N_j} \sum_k n_{jk}\beta_{ku} \qquad (3)$$

where the normalization N_j represents how many MFCC windows were extracted from this specific song.

7. Evaluation

MFCC data was extracted from all 1199 songs, resulting in 10.844.508 feature vectors. These vectors were grouped in 5, 10, 25, 50, 100 and 200 centroids, in order to compare experimentally VQ representations of several orders. The listening information was then concatenated with each timbre vector in order to have all data contained in one single matrix.

Each of the following was performed independently 20 times:

- Data matrix rows were shuffled;
- Data is then split in training and test subsets corresponding to 80% and 20% of the whole set. Listening information is separated and acoustic features data is normalized by the number of MFCCs extracted of each song;
- For the Logistic Regression the first subset was used to train the model, and the second for testing. Predictions were recorded in a text file;
- For the CBA the training subset was used for learning the β matrix, and the test subset for generalizing through equation 3;
- Predictions were recorded in a text file;
- F-measure, Precision, Recall and AROC values were calculated comparing predictions and true test values. The threshold for considering the predicted probability as 0 or 1 was learned from the training data;
- A random vector was generated and also compared to true values;
- Performance measurements were recorded in a log file.

CBA should stop iterating the learning loop once the difference between two consecutive matrices was above 1% of the number of centroids. Logistic Regression was chosen as a baseline reference for comparison purposes only, and the results are presented in Table 1.

7.1. Results Discussion

Recall, precision, f-measure and area under the receiver-operator curve (AROC) are standard metrics for evaluating binary classifiers [8]. Recall is obtained as the relation between true positive and the sum of true positives and false negatives (R=tp/(tp+fn)); Precision is the relation between true positives and sum of true positives and false positives (P=tp/(tp+fp)); and f-measure is the harmonic mean between both (F=2PR/(P+R)). AROC is the area under the curve representing true positive rate against the false positive rate.

The larger recall and f-measure values were obtained for the case of CBA with the lowest value for K. Here it means that 5 centroids is the best scenario where the relationship between MFCC representations and user behaviors was best captured in the β matrix, possibly meaning that a better prediction model would not use so many latent variables (the MFCC centroids) to express the recomendation as a function of the MFCC histograms.

We can also see that both f-measure and recall values decrease as functions of K, for both CBA and Logistic Regression. This might be interpreted in terms of a form of overfitting of the model to a given (training) dataset which is not able to perform equally well on a different (test) dataset. Overfitting would certainly explain the monotonicity of these metrics, where both f-measure and recall get progressively worse as more MFCC centroids are used to model each user's listening preference.

Precision, on the other hand does not display a very clear trend, although it is marginally higher for Logistic Regression than it is for CBA. The

		Recall	Precision	F-Measure	AROC
K=5	Log. Reg.	0.516 (0.117)	0.125 (0.016)	0.201	**0.525** (0.023)
	CBA	**0.677** (0.136)	0.119 (0.016)	**0.203**	0.513 (0.028)
K = 10	Log. Reg.	0.405 (0.070)	0.113 (0.013)	0.177	0.517 (0.022)
	CBA	**0.677** (0.119)	0.112 (0.010)	0.192	0.509 (0.045)
K = 25	Log. Reg.	0.300 (0.063)	0.115 (0.021)	0.166	0.513 (0.027)
	CBA	0.621 (0.121)	0.113 (0.015)	0.191	0.504 (0.027)
K = 50	Log. Reg	0.236 (0.051)	0.119 (0.018)	0.158	0.505 (0.028)
	CBA	0.612 (0.113)	0.113 (0.012)	0.190	0.513 (0.021)
K = 100	Log. Reg.	0.187 (0.053)	0.117 (0.023)	0.144	0.508 (0.031)
	CBA	0.572 (0.105)	0.110 (0.012)	0.184	0.511 (0.031)
K = 200	Log. Reg.	0.166 (0.049)	**0.133** (0.025)	0.148	0.515 (0.028)
	CBA	0.390 (0.085)	0.114 (0.018)	0.177	0.506 (0.026)

Table 1: A table presenting mean value and standard deviation of recall, precision, area under the receiver-operator curve (AROC), and f-measure for both settings: Logistic Regression, CBA. K represents the number of centroids used for the histograms.

upward jump of precision in Logistic Regression with K=200 explains the increase in f-measure for that method with this single value, denying the general decreasing trend; this was also the highest precision value for this experiment.

These values could be also compared to a baseline of a purely random recommendation. Since there were 533 complete song reproductions out of a recommendation matrix with 4 listeners and 1199 songs, the density of 1's is $533/(1199 * 4) \approx 0.111$ in the ground-truth, and so generating a uniformly random binary matrix would theoretically produce 0.111 of precision, 0.5 of recall, and an f-measure of 0.182, independently of K. This has also been established by numerical simulations, not reproduced here to save space.

Another baseline which might be interesting to consider is a pure recommendation of 1's (or simply stated "just listen to everything") or a pure recommendation of 0's ("don't listen to anything"). In the first case the theoretical precision would again be 0.111 for this data, and the theoretical recall would be 1.0, with an f-measure of 0.2, whereas the second approach produces recall $R = 0$ and precision and f-measure are not defined.

Every setting used in the experiment surpassed the 50% threshold under the ROC curve. This is the curve for defining the configuration of classifiers in terms of true positive against false positive, with its diagonal meaning random behavior. The highest value achieved was for Logistic Regression operating with 5 centroids.

8. Conclusion

CBA has proved to be a good model for predicting sparse listening data for small amounts of MFCC centroids in a Vector Quantized representation. It reached its best f-measure value when MFCC data was represented by only 5 centroids. The hidden β matrix represents the distribution between the centroids and listeners taste, and the results point to the possibility of representing these tastes in 5 dimensions, which may be due to the small number of listeners who participated in the experiment. Repeating the experiment with a larger number of listeners, as well as a more robust method for defining an optimal K value are considered as future work.

Vector-quantization turned out to be expensive in terms of memory consumption for high K values, and this should be also tackled in the future. The cost for running Expectation Maximization for a high number of centroids is also very high, and because of this, good results for low values of K are computationally preferable.

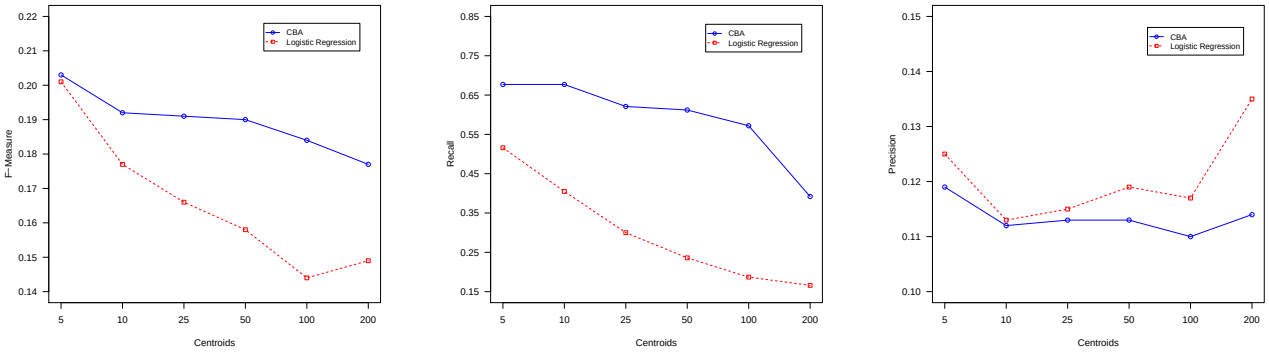

Figure 3: Comparison between F-measure, recall and precision for CBA and Logistic Regression for all values of K

References

[1] Beth Logan. Music recommendation from song sets. In *In Proc ISMIR*, pages 425–428, 2004.

[2] Gediminas Adomavicius and Alexander Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering*, 17(6):734–749, 2005.

[3] Jonathan L. Herlocker, Joseph A. Konstan, Al Borchers, and John Riedl. An algorithmic framework for performing collaborative filtering. In *Proceedings of the 22Nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '99, pages 230–237, New York, NY, USA, 1999. ACM.

[4] Luis M. de Campos, Juan M. Fernández-Luna, Juan F. Huete, and Miguel A. Rueda-Morales. Combining content-based and collaborative recommendations: A hybrid approach based on bayesian networks. *Int. J. Approx. Reasoning*, 51(7):785–799, September 2010.

[5] Matthew D. Hoffman, David M. Blei, and Perry R. Cook. Easy as cba: A simple probabilistic model for tagging music. In *10th International Society for Music Information Retrieval Conference*. 2009.

[6] Kazuyoshi Yoshii, Masataka Goto, Kazunori Komatani, Tetsuya Ogata, and Hiroshi G. Okuno. An efficient hybrid music recommender system using an incrementally trainable probabilistic generative model. In *IEEE Transaction on Audio Speech and Language Processing*, pages 435–447, 2008.

[7] Wikipedia. Lista dos 100 maiores discos da música brasileira pela Rolling Stone Brasil. `https://pt.wikipedia.org/wiki/Lista_dos_100_maiores_discos_da_musica_brasileira_pela_Rolling_Stone_Brasil`, 2007. [Online; accessed 28-May-2017].

[8] Ricardo Baeza-Yates, Berthier Ribeiro-Neto, et al. *Modern information retrieval*, volume 463. ACM press New York., 1999.