Automatic Music Recommendation Based on Acoustic Content and Implicit Listening Feedback

Rodrigo C. Borges (Universidade de São Paulo, São Paulo, São Paulo, Brasil)
rcborges@ime.usp.br
Marcelo Queiroz (Universidade de São Paulo, São Paulo, São Paulo, Brasil)
mqz@ime.usp.br

Abstract: Recommending music automatically isn't simply about finding songs similar to what a user is accustomed to listen, but also about suggesting potentially interesting pieces that bear no obvious relationships to a user listening history. This work addresses the problem known as "cold start", where new songs with no user listening history are added to an existing dataset, and proposes a probabilistic model for inference of users listening interest on newly added songs based on acoustic content and implicit listening feedback. Experiments using a dataset of selected Brazilian popular music show that the proposed method compares favorably to alternative statistical models.
Keywords: Music Recommendation Systems, Cold Start, Collaborative Filtering, Content-based Recommendation, Codeword Bernoulli Average Model, Vector Quantization.
Resumo: Recomendar músicas automaticamente não se resume a encontrar músicas similares às que o usuário está habituado a ouvir, trata-se também de sugerir peças potencialmente interessantes que não guardem relações óbvias com o histórico de escuta do usuário. Esse trabalho aborda o problema do *"cold start"*, onde novas peças sem histórico de escuta são inseridas em uma plataforma existente, e propõe um modelo probabilístico para inferência do interesse de escuta de usuários baseado em conteúdo acústico e *feedback* implícito de escuta. Experimentos usando um conjunto de músicas populares Brasileiras mostram que o método proposto compara-se favoravelmente a modelos estatísticos alternativos.
Palavras-chave: Sistemas de Recomendação Musical, *Cold Start*, Filtragem Colaborativa, Recomendação Baseada em Conteúdo, Modelo *Codeword Bernoulli Average*, Quantização Vetorial.

1. Introduction

Automatic music recommendation entered the mainstream in the last decades after a huge amount of digital media became available through online services (LOGAN, 2004) (ADOMAVICIUS; TUZHILIN, 2005). The most common music recommendation technique used today is called collaborative filtering (HERLOCKER et al., 1999), and it works by matching similar user listening profiles: if users A and B have similar listening profiles, and user A reacted positively to a certain song that hasn't been presented to user B, it is assumed that there is a high probability that user B would react positively to it. The success of a recommendation system may be measured by explicit feedback (e.g. a grade given by the user) and also by implicit feedback (e.g. a user skipping the recommended song) (HU, 2008).

Though very simple and yet very effective, the collaborative filtering approach has a weakness known as the "cold start" problem, which corresponds to the appearance of a new song not yet heard by any existing users, or the appearance of a new user with no listening history (i.e. with a blank listening profile). In the specific case of a new song (or item) added to a dataset with no available information relating this item to existing user profiles, one possible solution would be to extract acoustic information directly from the audio signal, and use it to build acoustic-based statistical representations. These will allow correlations between user profiles and audio items to serve as basis for inference and prediction of existing users potential listening interest on the newly added item.

If the acoustic content of the newly added item is somehow related to the acoustic contents of a user listening history, then it might be the case that this user would have an interest in listening to the newly added item once the system recommends it.

 Recebido em: 27/11/2017 - Aprovado em: 05/03/2018

Previous attempts at using acoustic features combined with collaborative filtering were already reported in the literature, some of them applying probabilistic modeling. YOSHII et al. (2008) have proposed the use of Gaussian Mixture Models for representing Mel-Frequency Cepstrum Coefficients (MFCC), and also explored an e-commerce interaction database as source for social listening data. MFCCs are widely used in MIR-related tasks, being considered a representation that captures relevant timbre-related aspects of audio signals. These authors proposed a group of latent variables corresponding to music pseudo-genres from which the user would have to choose, and the system selected a song using a stochastic method. They considered that pseudo-genres provided recommendation diversity, but forcing users to use this type of categorization is a very different approach from the predictions based on implicit listening feedback that are pursued here.

CAMPOS et al. (2010) propose a hybrid topological model based on Bayesian networks that combines collaborative filtering and content-based techniques. They applied their model to a movie recommender system, but unlike our work, the content-based part of their system was not extracted directly from the media, but instead consisted of textual metadata such as directors, producers, cast, plot keywords and genre.

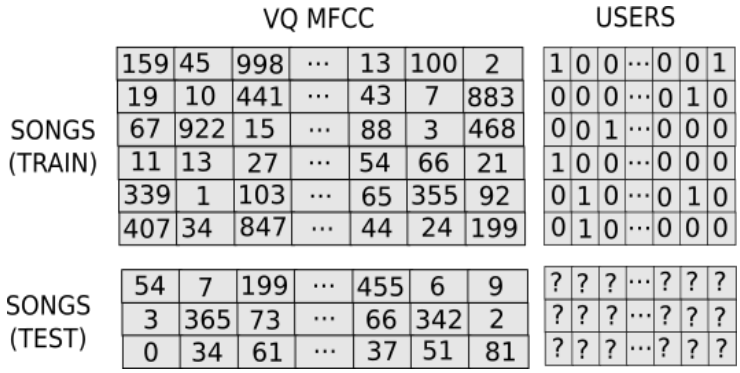

Figure 1: An Illustration of the problem of retrieving implicit listening feedback from acoustic features. The question marks are obtained as probabilities from the VQ MFCC histogram and estimated latent variables.

In this paper we propose the adaptation of the Codeword Bernoulli Average (CBA) model (HOFFMAN et al., 2009) for inferring listening behaviors towards new songs; specifically, to try to predict if a user will listen to the end to a recommended song instead of skipping it (i.e. we want to predict a user's implicit feedback). The CBA model was originally proposed for tagging music with semantic labels, and attempts to predict the probability that a tag applies to a given song, based on information extracted from its audio signal. The model is based on an intermediate representation space associated to a so-called latent variable (a parameter of the Bernoulli statistical model) that links vector-quantized (VQ) representations of the audio signals to corresponding tags, and estimates an optimal value for the latent variable through a supervised training process (using known audio signals with known tags).

In the newly added item form of the cold-start problem, instead of binary tags we consider the implicit feedback associated to a user skipping an item (represented by the value 0) or listening it to the end (represented by the value 1). Our audio representation combines Mel-Frequency Cepstrum Coefficients (MFCC) centroid vectors with the bag-of-words model, producing a histogram-type representation (VQ MFCC histogram) that is independent from audio duration.

Figure 1 illustrates the representation of items (songs) and users implicit feedbacks via two matrices, where each line corresponds to a song, columns on the left matrix repre-

sent VQ MFCC histograms and each column on the right matrix represent a particular user of the system. In the training phase (upper half of Figure 1) the implicit feedback is known, and latent variables (Bernoulli parameters) are estimated independently for each user, in order to adjust the known binary values to the given VQ MFCC histograms. In a subsequent test phase (lower half of Figure 1), probabilities of a user implicit listening feedback being 0 or 1 are calculated by combining the known histograms of the new item with the estimated latent variables to obtain a recommendation.

An experiment was conducted by collecting users implicit listening feedback through a media player app that continuously recommended new songs to users and recorded if they skipped or listened them to the end. The goal of this experiment was to assess the predictive power of the CBA model to obtain the binary values corresponding to a user's implicit feedback. As an alternative prediction method, the well-known and widely used Logistic Regression (LR) model (PARRA, 2011) (WANG, 2012) was used, allowing a comparison of the success rates of both methods applied to the same data.

This text is structured as follows. Section 2 presents the technical details of the Codeword Bernoulli Average model, including the training of the model and its application to test data. Section 3 presents the music dataset used, the media-player application used to collect listening data, and the feature extraction methods used to build the vector-quantized representations for the songs, as well as the experimental methodology used to validate the prediction model based on CBA. Section 4 presents and discusses the results of a comparative study between CBA and LR. Finally, we present conclusions and further work in Section 5.

2. CBA model for music recommendation

Representation of acoustic content

Vector Quantization (VQ) is a technique originally applied to data compression that considers a set of K selected vectors (codewords) that act as representatives for regions of a vector space. A codeword can be seen as a form of approximation or discretized version of other vectors, which may substitute the original vector or serve as a proxy (e.g. by storing the codeword index and the exact residue or deviation from the original vector). When applying VQ to time-varying data $v_1v_2...v_N$, each value v_j in the series may be associated to (or *quantized* as) one of the K available codewords q_j, producing a sequence of codewords $q_1q_2...q_N$. In this representation space, the comparison of two quantized sequences $p_1p_2...p_M$ and $q_1q_2...q_N$ with different lengths is not a trivial task.

One possibility for building a unified feature space for time-varying data with elements of different durations is to use the bag-of-words approach, which corresponds to building histograms of occurrences of each codeword q in a time-varying series. This allows the representation of each item as a histogram indexed by each one of the codewords to describe a large collection, make comparisons and create models based on a unified time-independent representation. For a collection of K codewords, each such histogram has always the same size K, with values (codeword counters) ranging from 0 to N (the temporal length of the time-varying series).

For the specific case of music recommendation, we propose to apply the VQ approach to MFCC vectors, clustering all occurrences of MFCCs in the whole database into K clusters, and using the centroids of these clusters as codewords. Specifically, the algorithm: (i) takes all MFCC data extracted from every song in the database and define K MFCC centroids using the clusterization technique KMeans; (ii) takes each song and quantize each of its

MFCC frames to the nearest MFCC centroid; (iii) produces a VQ MFCC histogram with fixed size *K* for each song, counting how many among all of its MFCC frames are best described by each one of the *K* MFCC centroids. All histograms are normalized by the song length N.

It is worth mentioning that the feature space thus produced is defined at the same time by all MFCC vectors extracted from the dataset (and so it is dependent on the dataset), and also by how exactly these MFCCs are clustered into *K* centroids (and so it is dependent on the clustering algorithm). When a new song is added to the dataset, this feature space should be recalculated, possibly defining a new group of MFCC centroids and corresponding VQ MFCC histograms for each song.

Training phase

Our CBA model assumes a collection of random variables **y**, with $y_{ju} \in \{0, 1\}$ representing the binary implicit feedback indicating whether or not user *u* has listened to song *j* (all the way to its end), or has skipped it. The goal of the training phase is to estimate a set of values for latent variables (Bernoulli parameters) β that will maximize the likelihood $p(y|n, \beta)$, i.e. of observing a certain binary implicit feedback y_{ju}, given VQ MFCC histograms **n** (with n_{jk} available for each song *j* and each centroid *k*) and parameters β (with β_{ku} available for each centroid *k* and each user *u*) using the Expectation Maximization (EM) algorithm.

The matrix β has dimensions given by the number of users (*U*) and the number of centroids (*K*), and represents the statistical relationship between users and MFCC centroids through a Bernoulli distribution. Each parameter β_{ku} acts as a user-dependent weight associated with each MFCC centroid, and one possible interpretation of it corresponds to an attempt of implicitly measuring how much the information encoded in the *k*-th MFCC centroid is worth to user *u*'s appreciation of a song; such an interpretation is not taken to mean that a complex phenomenon such as a user's appreciation can be cast in terms of MFCCs, but rather that such information could be statistically correlated, for inference purposes, to the probability of a user agreeing to listen to a certain song to its end (the validity of which can be tested experimentally).

Each EM iteration has two steps: the first one is the Expectation step, which computes the quantities h_{juk} defined below (HOFFMAN et al., 2009)

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

which are weighted averages representing the fraction corresponding to centroid *k* within the whole VQ histogram for song *j*, according to the weights β_{ku} computed for user *u* and centroid *k*. It should be noted that the weights in equation (1) are used differently according to user *u* having heard song *j* to its end (y_{ju}=1) or skipped it (y_{ju}=0): in the former case the weights are β_{ku}, whereas in the latter case the weights are (1-β_{ku}).

The Expectation step is followed by the Maximization step, which updates the latent variables β_{ku} according to

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

which is a weighted average of the binary implicit feedback y_{ju} over all songs *j*, using the parameters h_{juk} as weights, and could be interpreted as a way of measuring the contribution of a certain MFCC centroid *k* over all songs *j* to a user *u*'s appreciation of each song.

The EM algorithm stops iterating the above two equations when the distance between two consecutive values of the latent variable β is small enough (i.e., when $||\beta-\beta'||<\varepsilon$, using the Euclidean norm and with a very small threshold value $\varepsilon>0$). This results in a matrix which can be interpreted as the optimal value of the latent variables under which the training data corresponding to the known implicit feedback $\boldsymbol{y}$ is more likely to be observed than all possible alternatives (all possible reassignments of the binary implicit feedback values y_{ju}).

Generalization (test phase)

The latent variables β obtained in the training phase can then be used to predict a user implicit listening feedback on new songs by simply multiplying a song VQ MFCC histogram to a column of the β matrix corresponding to the user. The probability of a new song *j* being heard to its end by user *u*, given the song VQ MFCC histogram $\boldsymbol{n}_{j*}$ and the latent variables β_{*u}, is given by

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

which is a weighted average of the latent variables β_{ku} using the centroid counts n_{jk} as weights, and can be used to define recommendations using appropriate thresholds.

The same equation applied to any existing song used in the training phase for that user would give us the optimal probability value obtained by the Expectation-Maximization algorithm. Such optimal values associated to training data are useful to define appropriate thresholds for the recommendation system, in such a way that most songs with positive implicit feedback lie above the threshold and most skipped songs have probabilities lying below this threshold.

3. Listening Data and Prediction Experiments

Music dataset and listening data

The music dataset used in the experimental part of our work is composed of 1199 Brazilian popular songs taken from a selection known as the "*100 best records of Brazilian music*" (ROLLING STONE BRAZIL, 2007), 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 fairly heterogeneous set of music examples that should result in considerably different listening behavior patterns for different listeners participating in the experiment. The listening data was collected between March and December 2017, and was obtained from 13 listeners (ages varying from 25 to 60 years old).

An Android media-player application (Figure 2) was developed specifically for this experiment. When the user opened this application, a randomly selected song from the dataset started playing: the user could decide to listen to it to its end, or to skip it and jump

to the next song. This resulted in a sparse binary matrix $\boldsymbol{y}$ indicating each time a user has listened to a song until its end. For each user u and song j, the value y_{ju} is 1 if the user listened to the song to its end (regardless of how many times), which might be used as indicative of a user's willingness to hear that song, and it is 0 if the user skipped this song (i.e. was unwilling to hear it at that time).

It is important to mention that y_{ju} might have no value at all, indicating that the user had not been given the chance of listening to that song; in that case, no implicit feedback is available and that combination of song and user cannot be used in the training phase (where implicit feedback is supposed to be known in advance).

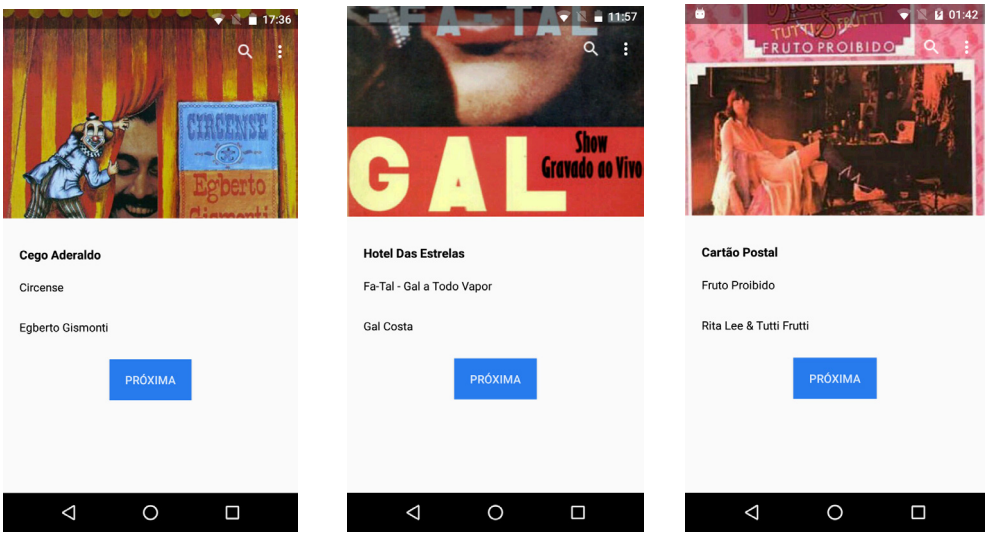

Figure 2: The Android media-player application used for collecting listening data.

It should be pointed out that, at this stage, recommendations had to be randomly produced. Data collected could not be biased towards pleasing the users, since both positive and negative implicit feedback were needed for training the prediction models. Furthermore, having this diversified feedback allows the use of any data collected as part of either the training set or the test set, with the known implicit listening feedback acting as ground-truth, against which the success rates of different recommendation strategies can be objectively measured.

During the period when listening data was being collected, there were around 3000 random song recommendations for the 13 participants (around 231 songs per user on average), and around 2000 complete song reproductions (songs not skipped). In order to allow for a reasonable statistical modeling, a criterium was adopted to ensure that sufficient training data would be available: only those listeners that had listened to at least 10% of the total number of songs (119 out of 1199 songs) in the dataset were considered for statistical analysis at this study. This reduced the number of participants from 13 to 5 subjects that satisfied the above criterium and could be considered in the statistical analysis.

Feature Extraction and VQ MFCC histograms

The MFCC acoustic descriptors were extracted using an open source Python library called Librosa[1]. The MFCC data was extracted with 13 coefficients, using windows of 2048 samples (≈ 46.44 ms) and 75% overlap between windows (these are commonly used values for these parameters: 13 MFCCs is e.g. the default in European Telecommunications Standard ES 201 108, and the segmentation values are the default ones in Librosa). As the

number of windows depend on the duration of the song, and we needed a uniform representation for the whole dataset, we adopted a VQ MFCC histogram representation as explained in Section 2. MFCC data was extracted from all 1199 songs, resulting in 10.844.508 feature vectors. These vectors were grouped in K = 5, 10, 25, 50, 100 and 200 centroids, in order to compare experimentally VQ MFCC histograms of several orders.

CBA implementation and listening feedback prediction

Our CBA implementation follows straightforwardly the description provided in Section 2. The stopping criterion adopted for the CBA corresponded to the distance between two consecutive β matrices being below 1% of the number of centroids (i.e. when $||\beta-\beta'||<\varepsilon$ with $\varepsilon=0.01*K$). Probabilities for each user u listening to a certain song j to its end are computed using equation (3). How to use these probabilities to predict binary values for y_{ju} is actually a very tricky problem, because equation (3) is not guaranteed to produce high values when y_{ju} is known to be equal to 1.

What one would expect from the model is that probabilities $p(y_{ju}=1|n, \beta)$ for y_{ju} known to be 0 should be in general lower than the probabilities $p(y_{ju}=1|n, \beta)$ for y_{ju} known to be 1. The proper threshold for considering a certain probability value to be high enough to predict y_{ju} to be equal to 1 is dependent on the user and also on the training set as a whole.

We defined these thresholds by analysing the Receiver Operating Characteristic (ROC) tradeoff between true and false positives. Specifically, this curve represents the number of true and false positives obtained for each possible threshold value adopted in the prediction process: given a threshold α, all probabilities $p(y_{ju}=1|n, \beta)$ below α produce predictions $\hat{y}_{ju}=0$ and all probabilities above α produce predictions $\hat{y}_{ju}=1$, and then these predictions are compared to the actual y_{ju} values to establish the number of true and false positives (TP/FP) and true and false negatives (TN/FN) which entail specific values of *precision* (TP/(TP+FP)), *recall* (TP/(TP+FN)) and *F-measure* (harmonic mean between precision and recall) for the predictor.

These are standard information retrieval metrics for evaluating binary classifiers (BAEZA-YATES et al., 1999). Thresholds are thus selected in order to maximize the precision of predictions made using the training data (thresholds must be defined using only training data, prior to the testing phase). In this experiment we chose to prioritize precision over recall, which does make sense in a recommendation system: we want to avoid whenever possible recommending songs which would not be appreciated by the user (i.e. we target the avoidance of false positives). This is not meant as a case against recall or the alternative aim of minimizing false negatives (in a recommendation system, false negatives are songs which would be appreciated but have not been recommended). Future work will address these two goals in the context of exploration versus exploitation (XING et al., 2014).

Alternative method: Logistic Regression

In order to have a sensible comparison for the newly proposed application of CBA to music recommendation, we implemented a logistic regression (LR) predictor, which is widely used for binary tagging purposes, using the Scikit[2] Python library. This predictor also associates latent variables β to the feature vector that is used as input for prediction, in our case the VQ MFCC histograms n_{*u}, and associates a logistic function that expresses the probability $p(y_{ju}=1|n, \beta)$ as $1/(1+e^{-f(n)})$ where $f(n)$ is a linear function of the K values of the histogram n_{*u} (weighted by the LR latent variables β).

Prediction experiments

As mentioned earlier, listening data was collected in the form of a sparse binary matrix ***y*** relating users and songs, where each entry of the matrix could be 1 (song listened to the end by the user), 0 (song skipped by the user) or undefined (song not presented to the user). For training and testing the system, only defined entries of this matrix were used, so that predictions of user implicit listening feedback could be compared to available values y_{ju} actually produced by the user. Each of the following was performed independently 20 times and separately for each user (the algorithm is illustrated in Figure 3):

- Data matrix rows were shuffled (producing a random order of songs);
- Matrix rows are reduced to the songs that were played for each specific user;
- Songs are split into training and test subsets corresponding to 80% and 20% of the dataset.
- A LR predictor was calibrated with the training data and LR predictions applied to the test set were recorded in a text file;
- A CBA predictor was calibrated with the training data and CBA predictions applied to the test set were recorded in a text file;
- F-measure, Precision, Recall and AROC performance measures were calculated comparing predictions and true listening feedback values, and were recorded in a text file.

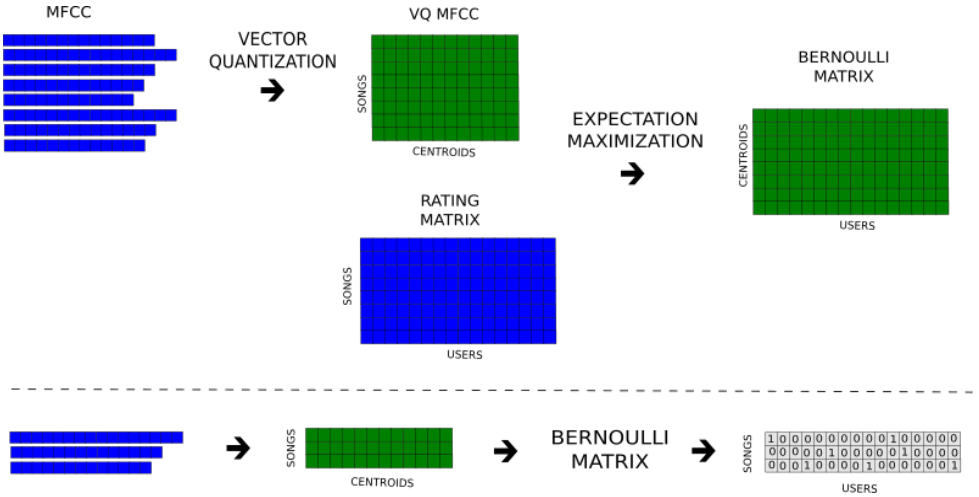

Figure 3: An illustration of the prediction procedure. The first row corresponds to training, the second one to the testing phase. Prior data is in blue colour, posterior in grey, and inner representation in green.

4. Results and Discussion

The results produced by the experiment detailed in the previous section are displayed in the tables below. All values of precision, recall, F-measure and AROC are expressed as means and standard deviations, and are presented for both methods (LR and CBA) as functions of the size *K* of the VQ MFCC histograms.

Table 1 below presents the actual results of the predictions produced by training the models with 80% of the available data and applying the predictors to the remaining 20%, repeating it for 20 independent random splits between training and testing.

The larger recall and F-measure values (recall=0.599 and F-measure=0.67) were obtained for CBA with K=200, and the best precision value (precision=0.792) was obtained

for CBA with K=5. Nevertheless, it should be noted that the differences in all these metrics for CBA are modest, with recall ranging from 0.544 to 0.599, precision ranging from 0.771 to 0.792, and F-measure ranging from 0.636 to 0.67.

For the LR method on the other hand K=5 produces higher values of recall, precision and F-measure, compared to LR using higher histogram sizes. These facts can be better appreciated in Figure 4 which plots these metrics as functions of K.

		Precision	Recall	F_measure	AROC
K=5	Log.Reg.	**0.607 (0.355)**	**0.449 (0.268)**	**0.51 (0.295)**	0.554 (0.08)
	CBA	**0.792 (0.134)**	0.566 (0.084)	0.656 (0.091)	0.551 (0.092)
K=10	Log.Reg.	0.469 (0.382)	0.351 (0.29)	0.399 (0.327)	0.545 (0.074)
	CBA	0.78 (0.117)	0.544 (0.095)	0.636 (0.09)	0.548 (0.079)
K=25	Log.Reg.	0.407 (0.409)	0.314 (0.316)	0.353 (0.355)	**0.557 (0.069)**
	CBA	0.78 (0.138)	0.579 (0.103)	0.659 (0.103)	**0.559 (0.103)**
K=50	Log.Reg.	0.402 (0.414)	0.304 (0.313)	0.345 (0.355)	0.556 (0.069)
	CBA	0.774 (0.141)	0.575 (0.093)	0.655 (0.1)	0.542 (0.092)
K=100	Log.Reg.	0.324 (0.398)	0.243 (0.3)	0.277 (0.34)	0.535 (0.055)
	CBA	0.781 (0.13)	0.582 (0.099)	0.662 (0.099)	0.537 (0.113)
K=200	Log.Reg.	0.133 (0.302)	0.101 (0.23)	0.114 (0.26)	0.517 (0.04)
	CBA	0.771 (0.126)	**0.599 (0.112)**	**0.67 (0.11)**	0.538 (0.098)

Table 1: mean value and standard deviation of recall, precision, area under the receiver-operator curve (AROC), and f-measure for LR and CBA applied to the test data. The highest values for each method are indicated in boldface.

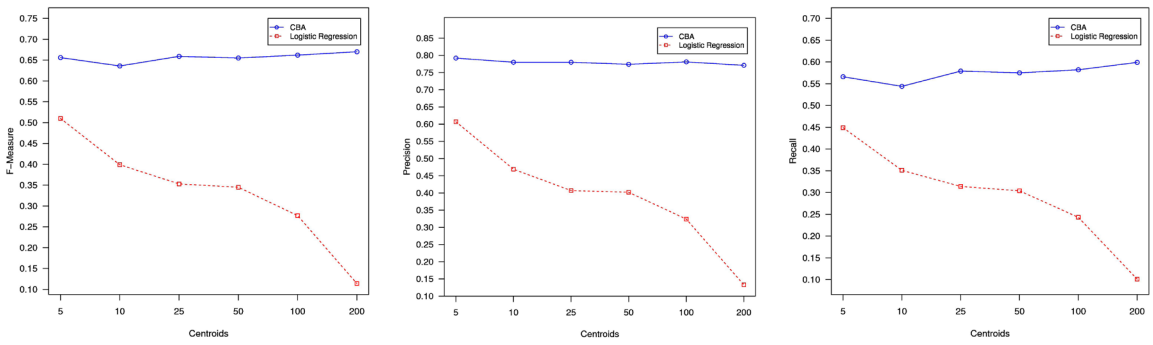

Figure 4: Comparison between F-measure, precision and recall for CBA and Logistic Regression for all values of K (blue circles represent CBA, red squares Logistic Regression)

It is apparent that F-measure and precision values remain relatively stable for CBA as functions of K, and recall does display a slight increasing trend, whereas all three metrics decrease for LR as functions of K. This descent could be related to some form of overfitting of the LR model to a given (training) dataset with higher histogram sizes, making it unable to perform equally well on a different (test) dataset, but it could also be explained in

terms of a poorer fit of the prediction model due to excessive detail in the representation.

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 positives against false positives, with its diagonal meaning random behavior. The highest values achieved for both LR (AROC=0.557) and CBA (AROC=0.559) were obtained for K=25 MFCC centroids.

In order to better understand the potential predictive power of these methods, and to try to explain their differences, we also applied the predictors obtained to the training data itself to produce Table 2 below. It is understood that these numbers are not meant to express the actual performance of the predictors, but simply to provide additional information on the experiment, to provide a glimpse on what the best-case scenario would look like for each prediction method, and also to objectively quantify how well the models reflect the data used for training. These numbers are also displayed graphically in Figure 5.

		Precision	Recall	F_measure	AROC
K=5	Log.Reg.	**0.608 (0.347)**	**0.446 (0.251)**	**0.514 (0.29)**	**0.564 (0.048)**
	CBA	0.792 (0.103)	0.574 (0.048)	0.664 (0.064)	0.574 (0.047)
K=10	Log.Reg.	0.484 (0.39)	0.368 (0.295)	0.418 (0.335)	0.562 (0.063)
	CBA	0.793 (0.102)	0.572 (0.053)	0.663 (0.064)	0.571 (0.053)
K=25	Log.Reg.	0.414 (0.414)	0.313 (0.313)	0.356 (0.356)	0.563 (0.064)
	CBA	0.807 (0.098)	0.594 (0.053)	0.683 (0.064)	0.595 (0.05)
K=50	Log.Reg.	0.4 (0.41)	0.3 (0.308)	0.343 (0.352)	0.557 (0.059)
	CBA	0.804 (0.102)	0.597 (0.053)	0.684 (0.069)	0.595 (0.054)
K=100	Log.Reg.	0.326 (0.399)	0.243 (0.298)	0.278 (0.341)	0.543 (0.055)
	CBA	0.819 (0.092)	0.618 (0.05)	0.703 (0.062)	0.619 (0.049)
K=200	Log.Reg.	0.134 (0.303)	0.099 (0.224)	0.114 (0.258)	0.518 (0.04)
	CBA	**0.836 (0.094)**	**0.654 (0.058)**	**0.734 (0.071)**	**0.653 (0.057)**

Table 2: mean values and standard deviation of recall, precision, area under the receiver-operator curve (AROC), and f-measure for LR and CBA applied to the training data. The highest values for each method are indicated in boldface.

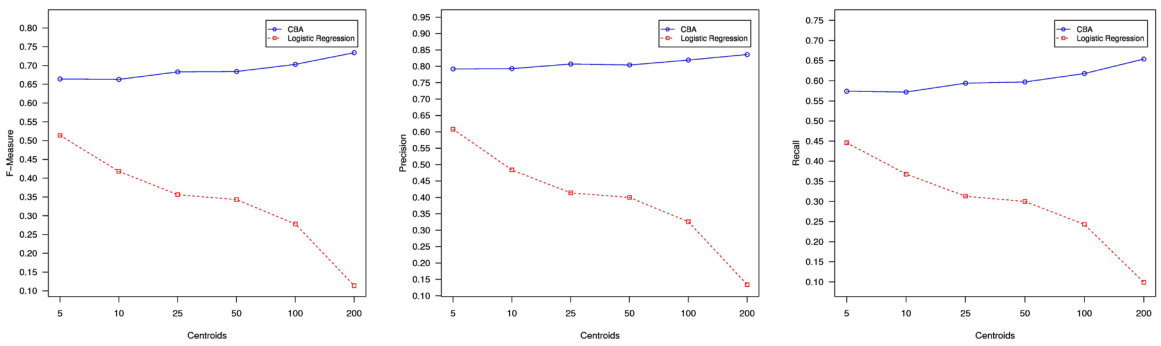

Figure 5: F-measure, precision and recall for CBA and Logistic Regression applied to training data for all values of K (blue circles represent CBA, red squares Logistic Regression)

It is immediately noticeable that all performance metrics improve for CBA as K gets larger, whereas all performance metrics get worse for LR with increasing K. Since each one of these numbers represent the best of possible scenarios in a supervised training setting, because the data used for training and for testing are the same, this suggests that finer VQ MFCC histogram representations (i.e. with larger K and a more detailed description of the actual MFCCs in each song) are useful within the CBA model in the sense that they raise the room for performance improvements (the ceilings for precision, recall, F-measure and AROC get higher).

For LR the opposite appears to be the case, i.e. coarser VQ MFCC histogram representations provide a better training for the Logistic models, with better performance both in training data (best-case) as well as in actual testing data.

On the other hand, the differences between these upper bounds and the actual values obtained in training can be related to the phenomenon of overfitting, since predictors get very good in getting the training data right, but not so good with different (testing) data. This can be seen in Figure 6, which shows the difference between the graphs in Figures 4 and 5. The differential F-measure graph for instance has a more or less stable profile for LR, with a peak in K=10 which could be an artifact due to the small number of users participating in the experiment; the interpretation here is that the best-case and average-case scenario are more or less the same independently of K for this technique, which rules out the hypothesis of overfitting.

For CBA an increasing tendency is visible as a function of K, which translates into a better fitting of the model to the training data and a worse performance when applied to new data. The differential precision and recall graphs have similar interpretations, being relatively constant independently of K for LR and showing an increasing tendency for CBA with increasing K (again with a few artifacts which may be due to the small sample space).

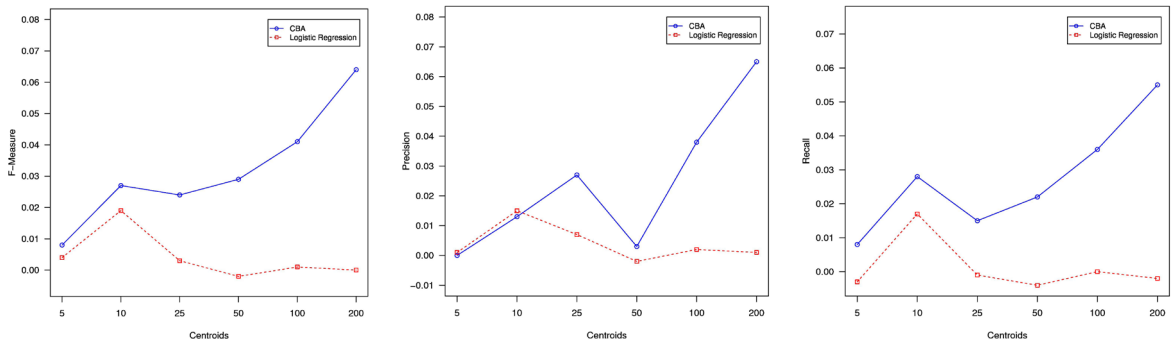

Figure 6: Difference between training and test values of f-measure, precision and recall respectively

5. Conclusions and future work

This article proposed a novel binary predictor for music recommendation based on a Codeword Bernoulli Average (CBA) model applied to binary implicit listening feedbacks from users of a media streaming application built on top of collection of Brazilian popular music dataset.

CBA has proved to be a good model for predicting a user's implicit listening feedback based on VQ MFCC histograms, with relatively small variations in the performance metrics precision, recall and F-measure for different values of K (number of MFCC centroids), being consistently superior to a competing alternative based on Logistic Regression (LR).

Moreover, LR performance consistently degraded as K increased, showing that this method is unable to profit from more detailed representations of the audio signal, in opposition to CBA which performed progressively better over finer VQ MFCC histogram representations (this is observed both in actual testing data as well as in a best-case scenario of applying the predictor directly on training data).

Even though K=200 MFCC centroids appear to produce the best representation scenario for predicting binary implicit listening feedbacks based on VQ MFCC histograms, other aspects of the predictor should be taken into account when deciding on a particular value of K.

Vector-quantization clustering via KMeans turned out to be very computationally expensive for high K values, which suggest that alternative clustering methods should also be explored in the future. The cost for running Expectation Maximization for a high number of centroids is also very high. Even though these algorithms are only run in the training phase and not in the test phase, in a real application every recommendation is followed by an implicit feedback which should be considered to update the model. This suggests that lower values of K are always computationally preferable as long as the performance of the predictor doesn't degrade considerably.

Another important point of the method is the need to have a large amount of labelled data for training, which reduced the number of participants in this study. Repeating the experiment with a larger number of listeners, as well as developing a more robust method for defining an optimal K value should be considered as future work.

Note

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

2 http://scikit-learn.org

References

DOMAVICIUS, Gediminas; TUZHILIN, Alexander. 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.

BAEZA-YATES, Ricardo; RIBEIRO-NETO, Berthier; et al. *Modern information retrieval*, volume 463. ACM press New York., 1999.

CAMPOS, Luis M. de; LUNA, Juan M. Fernández; HUETE, Juan F.; MORALES, Miguel A. Rueda. Combining content-based and collaborative recommendations: A hybrid approach based on bayesian networks. *Int. J. Approx. Reasoning*, 51(7):785–799, September 2010.

HERLOCKER, Jonathan L.; KONSTAN, Joseph A.; BORCHERS, Al; RIEDL, John. 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.

HOFFMAN, Matthew D.; BLEI, David M.; COOK, Perry R.. Easy as CBA: A simple probabilistic model for tagging music. In: 10th INTERNATIONAL SOCIETY FOR MUSIC INFORMATION RETRIEVAL CONFERENCE. 2009.

HU, Yifan; KOREN, Yehuda; VOLINSKY, Chris. Collaborative filtering for implicit feedback datasets. In: ICDM'08. Eighth IEEE International Conference on Data Mining. p. 263-272, 2008.

LOGAN, Beth. Music recommendation from song sets. In: PROCEEDINGS OF THE ISMIR Conference, pages 425–428, 2004.

PARRA, Denis et al. Implicit feedback recommendation via implicit-to-explicit ordinal logistic regression mapping. In: PROCEEDINGS OF THE CARS, 2011.

ROLLING STONE BRAZIL. "Os 100 maiores discos da Música Brasileira" (The 100 greatest records of Brazilian music) - *Rolling Stone Brasil*, october 2007, no. 13, page 109. Electronic Version [Online; accessed 28-December-2017]:

http://rollingstone.uol.com.br/listas/os-100-maiores-discos-da-musica-brasileira/

WANG, Xinxi; ROSENBLUM, David; WANG, Ye. Context-aware mobile music recommendation for daily activities. In: PROCEEDINGS OF THE 20th ACM INTERNATIONAL CONFERENCE ON MULTIMEDIA, p. 99-108, 2012.

XING, Zhe; WANG, Xinxi; WANG, Ye. Enhancing Collaborative Filtering Music Recommendation by Balancing Exploration and Exploitation. In: PROCEEDINGS OF THE INTERNATIONAL SOCIETY FOR MUSIC INFORMATION RETRIEVAL (ISMIR), p. 445-450, 2014.

YOSHII, Kazuyoshi; GOTO, Masataka; KOMATANI, Kazunori; OGATA, Tetsuya; OKUNO, Hiroshi G. 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.

Rodrigo C. Borges – Membro do Grupo de Computação Musical – IME/USP. Atua nos seguintes temas: processamento de sinais sonoros digitais, espacialização Sonora e sistemas musicais interativos.

Marcelo Queiroz – Professor Associado do departamento de Ciência da Computação da Universidade de São Paulo e vice-coordenador do NuSom - Núcleo de Pesquisas em Sonologia da USP. Possui doutorado e livre-docência em Ciência da Computação pela Universidade de São Paulo e graduação em Música (Composição Musical) pela Universidade de São Paulo. Tem experiência de pesquisa nas áreas de Computação Musical e Otimização Contínua, atuando principalmente nos seguintes temas: processamento de sinais sonoros digitais, espacialização sonora, acústica geométrica e sistemas musicais interativos.