On the Impact of Item Representation in the Quality of Recommendations

Andre Carvalho
Universidade Estadual de Campinas
andrescarvalho@students.ic.unicamp.br

Hendrik Macedo
Universidade Federal de Sergipe
hendrik@dcomp.ufs.br

ABSTRACT: *Most of data about an item in recommendation algorithms actually consists of meta-information. We argue that meta-information heavily affects the recommendation quality. This paper analyses the impact of deeper representations and more elaborate similarity measures in the quality of the recommendation of music. Firstly, we propose a innovative approach to extract musical features from audio files and use it to deepen song representation. Next, we propose a newer metric for measuring diversification. Finally, we evaluate both diversification and accuracy for the recommendation of songs. Results show that the inclusion of these deeper features on the representation of the item improves the accuracy and greatly improves diversification.*

Keywords: Music recommendation, Item representation, Accuracy, Diversification

Received: 14 April 2017, Revised 18 May 2017, Accepted 30 May 2017

1. Introduction

Automatic information filtering techniques have arisen in order to lessen the problem of deciding what information in the Web are relevant and what are not. Recommendation Systems (RS) are filtering systems that attempt to reproduce human recommendations automatically. Due to the subjectivity of the informally accepted definition of RS, varying RS systems have emerged in the literature: the recommendation of the best item within a subgroup, the recommendation of a list of items having higher quality than the expected, or the recommendation that escapes the desire of the user [28], [15].

Mainstream recommendation algorithms often make use of meta-information about an item, a shallow representation of the item to be recommended [1]. Collaborative filtering, for instance, only considers users evaluation in order to perform recommendations. This raises the question of how much the item representation affects the recommendation quality. Unfortunately, there are few datasets widely available that provides a deep representation of the item to be recommended.

Providing proper evaluation of the recommendation quality is not a trivial task. Different RS systems employ a sort of metrics when evaluating different aspects of a system; and while the literature clearly focuses on the evaluation of the accuracy, improvement on accuracy alone might not necessarily imply in the overall improvement of recommendation quality [25, 16, 18, 40, 22].

This paper analyses the impact of deeper representations and more elaborate similarity measures in the quality of the recommendation of music. Recommendation of music has recently been the target of great academic interest [32, 31, 6, 30]. Firstly, we propose a innovative approach to extract musical features from audio files and use it to deepen song representation. Next, we propose a newer metric for measuring diversification. Finally, we evaluate both diversification and accuracy for the recommendation of songs.

We present common approaches to recommending items in section 2. In section 3 we propose a method for extracting the set of musical instruments of MIDI files as well as a proper similarity function to deal with it. Section 4 discusses the problem of providing proper evaluation to the quality of recommendations; we also propose a new diversification metric. Experiments an results are presented in section 5. Finally, in section 6, we present some concluding remarks.

2. Recommendation Algorithms

There are varying approaches to recommendation systems [23, 5, 9]. One of the most prominent mathematical models treats the recommendation problem as a regression problem [17, 1]. Consider a set of items I and a set of users U, the relation $p(i, u)$ represents the preference of a user u for the item i: a numerical value expressing how much the user liked that item. A recommendation algorithm predicts the preference of user u for each of the items $i \in I'$, where $I' \subset I$ is the set of items to which the user has never set a preference before. We represent this prediction as $\hat{p}(i, u)$. The items are then sorted according to $\hat{p}(i, u)$ value. The goal of the recommendation system is thus to suggest the N items with the highest $\hat{p}(i, u)$ value (*TOP-N* method). There are other methods, one such being *Item Diversification* [40].

Two most well known recommendation approaches are the *content-based* and the *collaborative-based*. Content-based approach considers the information of an item in conjunction to the collected user preferences in order to generate the recommendations. Collaborative-based approach, in contrast, takes the preferences set by users that are known to have similar tastes and use them to generate the recommendations: no information on items are needed.

Collaborative algorithms assume that users with similar tastes will provide similar preferences to the same set of items. In other words, $p(i, u)$ is correlated to $p(i, w)$ for $u \neq w$ [27]. The most popular collaborative algorithms in the literature are those based on neighborhood, k-Nearest Neighbor (kNN), where the prediction $\hat{p}(i, u)$ uses the preference $p(i, w)$ of a group of similar users to u [21, 27]. If U_u are the users closest to u, $\overline{p}_u$ is the average of the values defined by $p(u, i)$ and $s(u, v)$ is the similarity between users, we have:

$$\hat{p}(i, u) = \overline{p}_u + \frac{\sum_{v \in Uu} (p(i, u) - \overline{p}_u)\, s(u, v)}{\sum_{u \in Uu} s(u, v)} \qquad (1)$$

Because different users may express their preferences using different scales of values, prediction is treated as a deviation of the average $\overline{p}_u$ of user preferences. It is important to highlight that the prediction uses no information on items. The similarity of users u and v may be derived from a correlation between items whose relation $p(i, x)$ is explicit to both. To this end, Pearson correlation is often used (equation 2):

$$s(u, v) = \frac{\sum_{i \in I_{u,v}} (\overline{p}_u - p(i, u))\, (\overline{p}_v - p(i, v))}{\sqrt{\sum_{i \in I_{u,v}} (\overline{p}_u - p(i, u))^2 \sum_{i \in I_{u,v}} (\overline{p_{uk}} - p(i, u))^2}} \qquad (2)$$

where $I_{u,v}$ is a subset of the items for which the point $p(i, x)$ is defined by u and v.

Content-based algorithms can also use the neighborhood to predict user preferences [11, 27], deriving a formula almost similar to equation 1:

$$\hat{p}(i,u) = \frac{\sum_{j \in Iu,i} p(j,u)\, s(i,j)}{\sum_{i \in Iu,i} s(i,j)} \quad (3)$$

In this equation, $s(i, j)$ represents the similarity between two items. An advantage of this approach is that it can be used to represent items even they do not behave well in Euclidean spaces, since we can treat similarity as a *kernel*. Hence, we may represent an item as a feature vector $i = (f_1, f_2, ..., f_n)$.

There are several different methods to compute similarity among items. Equations 4, 5 and 6 are some of them [29]. Equation 4, for instance, concerns the cosine *similarity*. It computes the angle between two feature vectors i and j of numerical values, as vectors of TFIDF weights. Equation 5 is the similarity of the *adjusted cosine*, in which the correlation between preferences $p(i, u)$ and $p(i, j)$ to all users $u \in U$ that have evaluated both i and j; the correlation is adjusted by the averages of the users preferences. Note that this similarity does not use any of the information on the item in question, only the preferences set by the users. Equation 6 is the Jaccard similarity, often used to compute the similarity between categorical attributes (*e.g*. keywords, genre).

$$s(i,j) = cos(i,j) = \frac{i \,.\, j}{\| i \| 2 \| j \| 2} \quad (4)$$

$$s(i,j) = \frac{\sum_{u \in U} (p(i,u) - \overline{R}_u)\,(p(j,u) - \overline{R}_u)}{\sum u \in U\,(p(i,u) - \overline{R}_u)^2 \;\sum_{u \in U} (p(j,u) - \overline{R}_u)^2} \quad (5)$$

$$s(i,j) = \frac{|\, i \cap j \,|}{|i \cup j|} \quad (6)$$

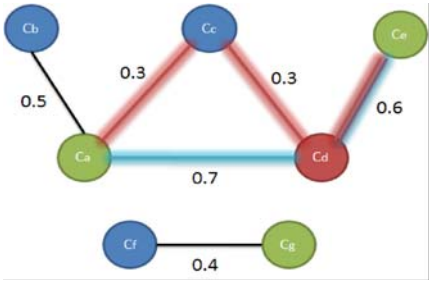

Figure 1. Demonstrates the use of various paths for the deduction of the relationship

Both approaches are complementary with regards to their advantages and disadvantages [33, 1]. The collaborative approach (1) does not depend on the content of the item, and can be easily used even with items whose representation is very complex such as video files, (2) depends on the success of usage by the community and needs a great sort of users; an individual whose preference greatly diverges from that of the community will not receive useful recommendations and, finally, (3) for a specific item to be recommended it is necessary that it has received at least one rating by one of its users.

The content-based filter (1) does not require a community of users; a lone user with a set of preferences is enough, (2) usually over-fits its recommendations, recommending very similar items, (3) requires a great amount of preferences set in order to properly learn the user profile and (4) depends on the representation of an item. A noteworthy fact for the success of both approaches is that the prediction depends on the definition of the similarity measure adopted.

3. The Impact of Item Representation

The representation of an item as well as the measure of similarity are central to content-based approach. Considering c_i and c_j

two categories (e.g. key word, genre, tag, etc.) that represent the item, naively we may presume that they do not hold any relation (*i.e.* $s(c_i, c_j) = 1$ com $c_i = c_j$, $s(c_i, c_j) = 0$ if on the contrary). However, this representation may not be precise; and there may be a correlation between both categories. Additionally, some of these relations may be interpreted differently by distinct users (e.g. musical genres) and so it is desired that these relations may be open to personalization.

3.1 Modeling User Preference

In order to mold the relations in similarity amongst categories, we used a simple graph, in which the edges, with weights, indicate the similarity between the categories.

Be $G(V, E, v)$, such that $\upsilon \in V$ represents a category, $e = (\upsilon, w) \in E : V \times V$ represents an edge (i.e. the existence or absence of a relation) and $v: E \rightarrow L_E$ function that labels the edges, in which $x \in L_E$, $0 \leq x \leq 1$. The value of $k = v(\upsilon, w)$ indicates the similarity between the categories υ and w.

For the case in which the relation between the two categories c_i and c_j is not known (i.e. are not neighbors in G), we may deduce a relation between them through the path $P_n = \{a_1, ..., a + n\}$. Such deduced relation is composed by the combination of all relations $v(a_i, a_{i+1})$ of the vertexes contained in the path. An example of such a combination is expressed in equation 7, which utilizes an average weighted by the path length. Since there may be various paths between c_i and c_j in G the deduced relationship may change according to the chosen path. An example of this phenomenon is illustrated in figure 1, in which there are distinct paths between the vertexes c_a and c_e with distinct values.

$$v_i(c_i, c_j) = \frac{\sum_{i=1}^{n-1} v(a_i, a_{i+1})}{n} \tag{7}$$

Due to the mechanism for deduction, it is advantageous that G be the most possible connection. The relations (i.e. edges) are built through a list of evaluations. Be L a list of points $(i, p(i, u))$ representing the evaluations associated to their respective items and be $p(c_i)$ the expressed preference for the category in L. Relationships are built through the complement of the absolute difference between the expressed preference of two categories, as described in equation 8. $p(c_i)$ may be calculated as an average of the evaluations $p(i, u)$ of the items i that are classified within the category c_j .

1-4 bytes Δ-time	1 byte Tipo e canal	1-4 bytes Tamanho	<tamanho> bytes Conteúdo

Figure 2. Format of a MIDI message

$$v_t(c_i, c_j) = 1 - |p(c_i) - p(c_j)| \tag{8}$$

If the relationship already exists, it is important that the model resist to a dramatic change in the current relationship and modify the value of the relationship gradually. Equation 9 defines the update as a weighted average between the new relationship $v_n(v, w)$ and the current $v(v, w)$; the weights w_e and w_n define the speed in which the model should adapt itself to changes in relationships.

$$v(\upsilon, w) = \frac{v_n(\upsilon, w)w_n + v(\upsilon, w)w_e}{w_n + w_e} \tag{9}$$

3.2 Extraction of Musical Characteristics

Songs have been chosen for this work due to their potential for feature extraction. Usually, the extraction of sonorous signals is relative to low level characteristics of the signal such as the rate of passage by zero, MFCC (Mel-Frequency Cepstral Coefficient), among others [12, 24]. Still, the extraction of sonorous signal itself is not capable of obtaining features considered to be high level, such as the rhythm, tone, and genre.

From data present in a MIDI file, it is possible to obtain high level features. One such feature is the set of musical instruments that compose the song. For this feature, we have considered the number of notes per instrument of a song, in a relative manner.

Figure 2 presents the model of a default MIDI message. The first field corresponds to the time elapsed since its last message. The second field indicates the type of message and the channel to which it is directed; in some messages, this field only represents the type of message. The third field contains the size of the message content and the fourth contains the message content itself.

The extraction of instruments from a song is done following the temporal order of the song, while monitoring the instruments pertaining to each channel. Each note sent to a channel is accounted for the counting of the notes for the active instrument belonging to the channel at the moment of the note dispatch. The result is a list of instruments $Ins = \{(in_1, p_1), ..., (in_n, p_n)\}$ with a number of notes p_i associated to each instrument in_i. This quantity is presented in a relative manner so that it may be used later in the calculation of similarity.

The algorithm traverses the MIDI messages in the temporal order. If the message is a *note*, the counting of the notes of the active instrument of the channel to which the message is sent is incremented as well as the total of notes read. If the message is a *change in instrument*, the reference to the active instrument of that channel is altered. Lastly, all counting of notes are divided by the total in order to obtain the relative value of each instrument.

The *Ins* set fits the definition of the features or that of the multivalued variables [4, 3], that are sets of categories or numbers associated to a taxonomy. The Ichino method for the comparison of multivalued variables [19] calculates the dissimilarity between objects composed by two components: one called *free of context*, used for monovalued features and the other, *context dependent*, for multivalued features. The method is described by equations 10, 11 and 12, but we will explore only equation 12, which corresponds to a context dependent component.

$$\phi(x, u) = \sum_{i=1}^{p} \phi_{cf}(x_i, u_i) + \phi_{cd}(x_i, u_i) \tag{10}$$

$$\phi_{cf}(x_i, u_i) = \frac{|\overline{X_i} \cap \overline{U_i} \cap (X_i \oplus U_i)|}{|X_i \oplus U_i|} \tag{11}$$

$$\phi_{cd}(x_i, u_i) = 1/2 \left[\sum_{k/x_k \in X_i \cap \overline{U_i}} w_k + \sum_{m/u_m \in \overline{X_i} \cap U_i} W_m \right] \tag{12}$$

This context dependent component sums up the values of the instruments that only appear on one of the sides, dividing everything by 2. Because this method calculates dissimilarity, we calculate the dissimilarity using the instruments that repeat themselves in both sides of the comparison. This is equivalent to add the averages of the instruments that are common to two songs.

The Ichino method does not give support to groups of instruments: only identical instruments count to similarity. In an extreme case, a song containing only guitar sounds would not be similar to another containing only acoustic guitar sounds, despite the fact that both instruments belong to the same group.

Due to the limitations of the similarity method, we have proposed a new method for similarity. It is capable of taking better advantage of the information found in the set of instruments. This method consists in forming pairs of instruments from two distinct songs with a posterior selection of pairs. These pairs are called *relationships* and the sets are created from the Cartesian product between the instruments found in the songs. In this case, the set R of relationships of the type $r(i_A, i_B)$ is the set:

$$R = I_A \times I_B \tag{13}$$

where I_X represents the set of instruments for a determined song X and i_X is an element of this set.

Each relationship has a value $r.v$ that is given by:

$$\forall r(i_A, i_B) \in R, r.v = \frac{\min(i_A, i_B)}{\max(i_A, i_B)} * f \tag{14}$$

In equation 14, f corresponds to the group factor, whose value varies from 0 to 1. For different instruments within a single group, this variable may receive an intermediary value. For identical instruments, f has the value of 1 and for different instruments that do not belong to a single group the value is 0.

The similarity measure between instruments $s(a, b)$ is calculated incrementally according to equations 15 and 16:

$$s(a, b) = \frac{S_1 + S_2 + \ldots + S_n}{n} \quad (15)$$

$$S_i = \max(R_i) \quad (16)$$

where $\max(R_i)$ is the relationship in which $r.\upsilon$ is the maximum in set R_i. At each interaction i, the relationship having the greater value is selected and so a new set R_{i+1} is created in accordance to equations 17 and 18.

$$R_{i+1} = R_i - E_i \quad (17)$$

$$E_i = \{r(x, y) \in R_i | \exists z, w, max(R_i) = r(z, w), x = z \vee y = w\} \quad (18)$$

The set E_i represents all relationships of set R_i that share at least 1 instrument with $\max(R_i)$. This set is subtracted from set R_i, thus forming set R_{i+1}, from which the process repeats itself iteratively. At the end of all interactions, the average of the values of the maximum relationships selected during the course of the process is calculated. The calculation starts with $R_1 = R$ and ends when the set R_n is such that $R_{n+1} = \phi$.

Another difference between the methods is found in the calculation of similarity between instruments. The proposed method makes use of a ratio, whereas the Ichino method uses an average. In table 1, we demonstrate how this difference influences similarity. The table shows a percentage of the participation of an instrument in two songs and the impact of the use of an average, or that of a ratio, in similarity between songs, considering only sets of instruments. As an example, in the first line, the participation of a given instrument is high in both songs, and the similarity between them would be found to be high using either an average or a ratio. In practical terms, using the Ichino Method, a pair of songs containing an infrequently played instrument would result in low similarity, given the minor participation of the instrument in question. However, using the proposed method, the similarity of this instrument would be high, since the participation in both songs would be similar.

Porcentagem		**Similaridade**	
Song A	**Song B**	**Average**	**Ratio**
High	High	High	High
High	Low	Average	Low
Low	Low	Low	High

Table 1. Similarities using averages and ratios

4. Evaluating Recommendation Systems

The evaluation of recommendation system has generated much discussion in the literature [2, 15, 34, 35]. And although it is natural that metrics of accuracy, as with the Mean Absolute Error (MAE), are used for the evaluation of the quality of recommendation, there is much evidence that accuracy is not sufficient in it of itself for the evaluation of the utility of the recommendations provided [26, 13].

Among the aspects desired of a recommendation are:

- Accuracy - indicates the accuracy of a prediction in terms of the relevancy of the items ($\hat{p}$);
- Novelty and Serendipity - recommendation with aspects of novelty represent recommendations that are not obvious for the

user [8, 38]. An example is that of unpopular items *i*, the recommendation of the item *i* is obvious, once it is possible that the user has knowledge of the item *i*. Recommendations with the aspect of surprise are recommendations of items that escape the standards of the user but are still sufficiently relevant.

- Diversification - indicates to what extent are the system's produced recommendations diverse.
- Dependability - measures the reliability of how useful a recommendation truly is;
- Leaning rate - measures the speed at which the system learns; and this is an important aspect of usability since user evaluations are costly to be collected.

The importance or significance of an aspect in a system depends on the task and the goal of the SR [16].

One of the metrics most commonly used in evaluating system accuracy is the MAE, defined in equation 19, where N is the number of evaluations $p(i, u)$ considered in the calculation. This metric indicates an average deviation in user given grades in relation to the grades predicted by the system. The MAE metric is simple in use and interpretation, and also makes using tests to form direct comparisons between algorithms from statistical significance possible [10, 17].

$$|\overline{E}| = \frac{\sum_{i=1}^{N} |\hat{p}(i, u) - p(i, u)|}{N} \tag{19}$$

The use of accuracy metrics (e.g. MAE) does not indicate quality of the produced recommendations. The results presented by Lathia et al [22] shows that the MAE obtained while utilizing different measurements of similarity in collaborative KNN were comparable to the results using random similarity. This indicates that the measures of similarity used were not sufficiently significant to represent the relationships amongst users. Another possibility is that the database used (i.e. Movielens [14]) is the cause of this phenomenon, since it has the *Long Tail* property, or be it, a small subset of items is evaluated by a majority of the community, however, the number of items that have received little evaluation is much greater in number.

The MAE metric allows for the execution of tests that verify statistical significance. We are able to calculate the necessary number of samples needed for comparisons using equation 20 derived from the inequality of the extremes of intervals of confidence of two samples [20]:

$$n \geq (z \frac{(S_y + S_x)}{\overline{x} - \overline{y}})^2 \tag{20}$$

Being random variables x and y, s_x and s_y the sample standard deviations, and z the value of the desired quantile of the normal distribution for the due interval of confidence. We can also calculate the number of necessary samples according to the accuracy formulated in equation 21, where r is the accuracy necessary.

$$n = (\frac{100 z S_x}{r\overline{x}}) \tag{21}$$

The Pearson correlation takes into account the absolute values of two variables. Other correlations such as the Kendall τ (equation 22) and the Spearman correlations may only be used in relative analyses. In the Kendall τ we have that n_c is the number of concurring pairs (i.e. the number of pairs of items that appear in the same order in two compared lists), n_d the number of incongruent pairs and $n = n_c + n_d$ (i.e. the total number of items).

$$c = \frac{n_c - n_d}{1/2n\,(n - 1)} \tag{22}$$

Recommendations solely based on accuracy tend to consistently recommend similar items when the set of evaluations of a user tends to be limited diversified [39, 25]. Some of the metrics found in the literature have emerged from the intent of measuring the diversification of a recommendation list. One such metric is the *Intra List Similarity* [40]. The measure is defined in equation 23, where R is a set containing the items of the list, and $s(i_k, i_l)$ is a function of the similarity among items

$$ILS(R) = \frac{\sum_{ik \in R} \sum_{il \in R,\, ik \neq il} s(i_k, i_l)}{2} \tag{23}$$

Equation 23 requires the use of similarity measure for the calculation of diversification. The choice of the measure may, thus, affect the value of the metric. In order to avoid the creation of a similarity function which might favour a similarity measure in the algorithm, we can just analyze the concentration of a given category at a histogram. For that, we compute the percentage contribution p_w for each distinct w in the list R. We define then the 24; the closer to zero is c, less concentrated in a category the list is. This is useful to domains with well separated categories.

$$c = \sum_{w} p^2{}_w \tag{24}$$

The diversification measures must not be used in detriment of other aspects, this because a recommendation algorithm is capable of producing very diverse recommendations. The goal of diversification is to avoid monotonous recommendations, however, relevancy towards the item is important within the process of recommendation. Be that as it may, in general accuracy and diversification are conflicting aspects [7, 37].

Another measure of similarity is *Item Novelty* [7, 36], defined in equation 25, where R is the list of items, i and j are the items, d is a measure of distance or dissimilarity and $p = |R|$. This measure calculates the *novelty* of an item in accordance to an R; and if applied to the user list of evaluations this measure shows how new the item is to the user.

$$n_R(i) = \frac{1}{p - 1} \sum_{j \in R} d(i, j) \tag{25}$$

5. Experiments and Results

5.1 Methodology

Algorithms are evaluated in accordance to the previously presented metrics: MAE (equation 19), kendall-tau (equation 22) and the measures for diversification: ILS (equation 23) and the metric proposed in this paper (equation 24). The set of evaluations is divided in two ways: 80%/20% and 50%/50% for training and testing purposes, respectively. The amount of neighbors considered K in the prediction $\hat{p}$ is varied in the experiments.

The representation of music in the system includes two types of features: metadata and extracted data. The representation includes:

- Genre - represents the classification of the musical genre for a specific song. It is represented by set *Ge*;
- Artist/Composer - represents the name of a composer or an artist representing the song. It is represented by set *Ar*;
- Year - represents the year of publication or composition of the song. It is represented by set *An*;
- Instruments - represents the set of instruments that are part of the song composition represented in section 3.

For each feature, we have used different measures of similarity, as described in equations 26 to 30; the subscript s is the characteristic function which computes the similarity (e.g. s_{Ge} represents the similarity between genres). Equation 26 refers to the similarity of equality and is used for artist/composer as well as for genres. Equation 27 uses the relationships v_i of the created graphs G_{ge} and G_{ar} for genres and artists/composers respectively as described in section 3.

$$s^b{}_{ar}(i, j) = s^b{}_{ge}(i, j) \{i = j \& 1 i \neq j \& 0 \tag{26}$$

$$s^a{}_{ge}(i, j) = s^a{}_{ar}(i, j) = v_i(i, j) \tag{27}$$

For the year, we have used the distance between decades:

$$s_{an}(a_i, a_j) = 1 - \frac{|dec(a_i) - dec(a_j)|}{100} \tag{28}$$

where $dec(i)$ represents the decade of the year i.

For the musical instruments, we considered two distinct measures of similarity: equations 15 and 12 presented previously:

$$s^{a}_{ins}(ins_i, ins_j) = S(ins_i, ins_j) \quad (29)$$

$$s^{b}_{ins}(ins_i, ins_j) = \phi_{cd}(ins_i, ins_j) \quad (30)$$

The similarity of the item is described as a weighted average of the similarity of each feature that pertains to the item:

$$s(i_a, i_b) = \frac{\sum_{j=1}^{n\_carac} s(i_a.j.j_b.j)}{ncarac} \quad (31)$$

The database considered in this assessment was set to aid the extraction of musical features. It consists of a total of 337 item evaluations. Considering the importance of the properties exposed in [16], following we depict some database aspects.

• Domain characteristics - the domain used is the musical domain, on which there are characteristics extracted from the musical composition in the representation of the item. The cost of a false positive is relatively low, since songs are short and their cost consumption (considering digital distribution) is low to none. However, a false positive does exclude the appreciation of the item by the user in the order of preference. As a consequence, it is symbolically high because the user would not consume a relevant item.

• Inherent characteristics - evaluations were feed explicitly by the users and varied on a scale of 0-10 in real numbers. However, few users used values that really took advantage of the density scale. The base contains information about items as noted in the previous sections. The items were given to users to be evaluated in a random manner.

• Sample characteristics - The base contains a total of 17 users, 202 items and a total of 337 evaluations generating an average of 19 evaluations per user. All evaluated users have at least 12 evaluations. And as a general review, 162 items were evaluated by at least one user and 40 were not evaluated by any user. Because the items evaluated were chosen at random, the distribution of ratings per item is approximately equal. The distribution of genre is also roughly equivalent.

The database may be considered difficult to manage due to the small quantity of data to undergo both the collaborative method (i.e. low amount of users) as well as with the content-based method (i.e. low amount of reviews by users).

5.2 Results and Discussion

Table 2 shows the results of applying the initial accuracy metrics. The first column shows the combination of features considered and their similarity measures, except for the collaborative method which is indicated by *Col*.

On the first line, for instance, ($s^{a}_{g\,e}$, s^{a}_{ar}), only the genre and artist/composer features are considered and by using the models considered, the similarity measure used for both is equation 27.

The columns show the metric applied and the division of the examples of the applied system (*e.g*. MAE(80%/20%) means that the metric MAE is divided as 80% for training and 20% for tests applied to each of the users in the system). The title of the table indicates the amount of neighbors considered (e.g. K = 3 means three neighbors) and the size of the list of evaluations $|L|$, which affects the construction of the similarities of the model presented in section 3.

Each obtained value for the metrics in table 2 is the average of 100 samples collected. Each collected sample is the average error of all system users. Using equation 21 we obtained the accuracy of r = 1:14% which in absolute values is r = 0:02. As the accuracy is less than the difference between the averages, this amount of samples is enough for most of the comparisons; subsequent tables follow the same organization, including the samples, unless otherwise stated.

Observing the results presented in table 2 we take some conclusions. The collaborative method (*Col* in the table) obtained better accuracy than all methods based on content. However, when a smaller number of evaluations is used (50% for training) methods based on content improve their performance. The success of the collaborative approach can be explained by the fact that there is little diversity among the community of users in the base. Methods for similarity of multivalued feature (e.g. s^{a}_{ins}, s^{b}_{ins} obtained similar results, but the method of similarity proposed in section 3 obtained slight improvement.

Table 2 shows accuracy results considering only one type of similarity measure. Table 3 shows accuracy arising from the

combination of different similarity measures/features. There was a clear improvement for combinations that include the similarity of equality ((s^b_{ge}, s^b_{ar})). Other features complement binary similarity when the binary similarity fails (i.e. when binary similarity results in 0), and this demonstrates the importance of the similarity measure in the quality recommendations.

Table 4 shows the impact of increasing the number of reviews per session considered. This parameter affects only the methods that use the model presented in section 3. There was a general improvement in accuracy with the increased number of reviews per session.

The previous tables demonstrated the accuracy of the system with the number of neighbors $K = 3$. Table 5 shows the results of the most promising metrics within the tables considering $K = 5$. The collaborative method showed a significant improvement in accuracy once the number of neighbors was increased. Content-based methods showed no such improvement.

Table 6 shows the results obtained from the application of the metrics of diversity 23 and 24 applied to a recommendation list of size 10. The value shown is the average of the values obtained from each user. The similarity measure used in equation 23 was the same used to make the recommendation; as for the collaborative method, similarities (s^b_{ge}, s^b_{ar}) were used.

For equation 24, the concentrations of genres and artists/composers were analyzed.

Features	**MAE (80%/20%)**	**Kendall (80%/20%)**	**MAE (50%/50%)**	**Kendall (50%/50%)**
(s^a_{ge}, s^a_{ar})	2,05±1,30	0,42±0,36	2,20±1,14	0,44±0,16
(s^b_{ge}, s^b_{ar})	4,48±2,26	0,36±0,36	4,88±1,81	0,41±0,20
(s^a_{ins})	2,25±1,34	0,40±0,36	2,37±1,16	0,46±0,18
(s^b_{ins})	2,31±1,36	0,43±0,37	2,46±1,20	0,47±0,19
(s_{an})	2,33±1,46	0,44±0,37	2,34±1,18	0,48±0,19
(*Col*)	1,58±1,14	0,28±0,33	2,23±1,48	0,32±0,19

Table 2. Metrics of Accuracy X Features set ($K = 3$, $|L| = 4$)

Features	**MAE (80%/20%)**	**Kendall (80%/20%)**	**MAE (50%/50%)**	**Kendall (50%/50%)**
(s^a_{ge}, s^a_{ar}, s_{an})	2,12±1,34	0,40±0,37	2,28±1,15	0,48±0,20
(s^b_{ge}, s^b_{ar}, s_{an})	2,30 ±1,48	0,40±0,37	2,38±1,18	0,50±0,20
(s^a_{ge}, s^a_{ar}, s^a_{ins})	2,07±1,28	0,38±0,36	2,27±1,10	0,47±0,19
(s^b_{ge}, s^b_{ar}, s^a_{ins})	2,09±1,30	0,38±0,36	2,30±1,14	0,46±0,19
(s^b_{ge}, s^b_{ar}, s^b_{ins})	2,16±1,32	0,36±0,35	2,45±1,20	0,46±0,19
(s^a_{ge}, s^a_{ar}, s_{an}, s^a_{ins})	2,10 ± 1,30	0,44 ±0,36	2,23±1,11	0,48 ± 0,15
(s^b_{ge}, s^b_{ar}, s_{an}, s^a_{ins})	2,19 ± 1,34	0, 46 ±0,36	2,28 ,±1,15	0,48,±0,14

Table 3. Accuracy Metrics X Features set ($K = 3$, $|L| = 4$)

Features	MAE (80%/20%)	Kendall (80%/20%)	MAE (50%/50%)	Kendall (50%/50%)
(s^a_{ge}, s^a_{ar})	1,91±1,26	0,39± 0,36	2,10 ±,07	0,42 ±0,16
$(s^a_{ge}, s^a_{ar}, s^a_{ins})$	1,89±1,15	0,41±0,36	2,10 ±1,05	0,45 ±0,15
$(s^a_{ge}, s^a_{ar}, s_{an}, s^a_{ins})$	1,95±1,23	0,41±0,36	2,23±1,11	0,48 ±0,15

Table 4. Accuracy Metrics X Features set (K = 3, $|L|$ = 6)

Features	MAE (80%/20%)	Kendall (80%/20%)	MAE (50%/50%)	Kendall (50%/50%)
(s^a_{ge}, s^a_{ar})	1,94 ± 1,24	0,42 ± 0,36	2,09 ± 1,03	0,41 ± 0,16
(s^a_{ins})	2,16 ± 1,29	0,45 ± 0,35	2,27 ± 1,17	0,47 ± 0,13
(s^b_{ins})	2,24 ± 1,29	0,44 ± 0,35	2,38 ± 1,21	0,47 ± 0,16
$(s^a_{ge}, s^a_{ar}, s^a_{ins})$	1,93 ± 1,18	0,42 ± 0,36	2,12 ± 1,06	0,44 ± 0,15
$(s^b_{ge}, s^b_{ar}, s^a_{ins})$	1,95 ± 1,23	0,41± 0,36	2,16 ±1,13	0,44± 0,15
$(s^b_{ge}, s^b_{ar}, s_{an}, s^a_{ins})$	2,13±1,33	0,48± 0,36	2,21 ± 1,13	0,49 ± 0,14
$(s^a_{ge}, s^a_{ar}, s_{an}, s^a_{ins})$	2,00 ±1,26	0,46±0,36	2,14 ± 1,08	0,45 ± 0,14
(Col)	0,58 ± 0,48	0,14 ± 0,23	0,86 ± 0,75	0,13 ± 0,11

Table 5. Accuracy Metrics X Features set (K = 5, $|L|$ = 6)

Characteristcs	Hist. (GABnero)	Hist. (Artista)	ILS
(s^a_{ge}, s^a_{ar})	0,78 ± 0,24	0,67 ± 0,30	3,32 ± 1,01
$(s^a_{ge}, s^a_{ar}, s^a_{ins})$	0,48 ± 0,25	0,43 ± 0,25	1,33 ± 0,86
$(s^b_{ge}, s^b_{ar}, s^a_{ins})$	0,68 ± 0,28	0,58 ± 0,33	1,77 ± 0,97
(s^a_{ins})	0,17 ± 0,05	0,16 ± 0,05	0,02 ± 0,01
(Col)	0,16 ± 0,04	0,14 ± 0,03	0,68 ± 0,15

Table 6. Accuracy Metrics X Features set (K = 5, $|L|$ = 6)

The use of similarity s^a_{ins} produced good results in respect to diversification when used isolated and combined. This helps to solve the problem of over fitting of the content based approach.

6. Conclusion

In this work we propose a model that captures the similarity between a set of categories used to represent an item, a method of extracting high level musical features using MIDI files and a metric for the evaluation of diversification.

As expected, the use of more elaborate similarities and representations implied in an improvement for both accuracy and diversification, especially in the case of extracted features; its inclusion in the features set resulted in a considerable improvement in diversification. This is important since monotonicity is a huge problem in a content-based approach. It is possible that the

problem can be bypassed with the inclusion of a set with a sufficiently large quantity of features. While it is not possible to verify if the use of the proposed metrics is more useful than that of the ILS, it has the advantage of not needing a measure of similarity. The database used is, unfortunately, insufficient for the execution of a more thorough analysis, especially for analysis of the collaborative method. There is, however, a lack of an evaluation database containing extractions of deeper characteristics of items that are not textual.

References

[1] 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*, 17:734–749, 2005.

[2] Charu C Aggarwal. (2002). Evaluating recommender systems. In Recommender Systems, pages 225–254. Springer, 2016. [3] Byron L. D. Bezerra. Estudo de algoritmos de filtragem de informação baseados em conteúdo. Trabalho de graduação, Centro de Informática - UFPE, Recife

[4] Lynne Billard. (2006). Symbolic data analysis: what is it? In Compstat 2006-Proceedings in Computational Statistics, pages 261–269. Springer

[5] Jesús Bobadilla, Fernando Ortega, Antonio Hernando, and Abraham Gutiérrez. (2013). Recommender systems survey. Knowledge-based systems, 46 109–132

[6] Dmitry Bogdanov, MartíN Haro, Ferdinand Fuhrmann, Anna Xambó, Emilia Gómez, and Perfecto Herrera. (2013). Semantic audio content-based music recommendation and visualization based on user preference examples. Information Processing & Management, 49(1) 13–33

[7] Pablo Castells, Neil J Hurley, and Saul Vargas. (2015). Novelty and diversity in recommender systems. In Recommender Systems Handbook, pages 881–918. Springer

[8] Pablo Castells, Saúl Vargas, and Jun Wang. (2011). Novelty and diversity metrics for recommender systems: choice, discovery and relevance.

[9] Cazella, S.C., M. Nunes, and E. Reategui. A ciência da opinião: Estado da arte em sistemas de recomendação. André Ponce de Leon F. de Carvalho; Tomasz Kowaltowski..(Org.). Jornada de Atualização de Informática-JAI, pages 161–216, 2010.

[10] Tianfeng Chai and Roland R Draxler. Root mean square error (rmse) or mean absolute error (mae)?–arguments against avoiding rmse in the literature. Geoscientific Model Development, 7(3):1247–1250, 2014.

[11] Suthee Chaidaroon. Content-based recommendation. 2016.

[12] Namrata Dave. Feature extraction methods lpc, plp and mfcc in speech recognition. International journal for advance research in engineering and technology, 1(6):1–4, 2013.

[13] Mouzhi Ge, Carla Delgado-Battenfeld, and Dietmar Jannach. Beyond accuracy: evaluating recommender systems by coverage and serendipity. In Proceedings of the fourth ACM conference on Recommender systems, pages 257–260. ACM, 2010.

[14] Grouplens. Movielens data set, 2011.

[15] Asela Gunawardana and Guy Shani. Evaluating recommender systems. In Recommender Systems Handbook, pages 265–308. Springer, 2015.

[16] Jonathan L. Herlocker, Joseph A. Konstan, Loren G. Terveen, and John T. Riedl. Evaluating collaborative filtering recommender systems. ACM Trans. Inf. Syst., 22:5–53, January 2004.

[17] Jonathan Lee. Herlocker. Understanding and improving automated collaborative filtering systems. PhD thesis, University of Minnesota, 2000.

[18] Félix Hernández del Olmo and Elena Gaudioso. Evaluation of recommender systems: A new approach. Expert Syst. Appl., 35:790–804, October 2008.

[19] M Ichino and H Yaguchi. Generalized minkowsky metrics for mixed feature type data analysis. In IEEE Transactions system, Man and Cybernetics, volume 24, pages 698–708, 1994.

[20] Raj Jain. The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation, and Modeling. John Wiley and Sons, 1991.

[21] Yehuda Koren and Robert Bell. Advances in collaborative filtering. In Recommender systems handbook, pages 77–118. Springer, 2015.

[22] Neal Lathia, Stephen Hailes, and Licia Capra. The effect of correlation coefficients on communities of recommenders. In Proceedings of the 2008 ACM Symposium on Applied Computing, SAC '08, pages 2000–2005, New York, NY, USA, 2008. ACM.

[23] Jie Lu, Dianshuang Wu, Mingsong Mao, Wei Wang, and Guangquan Zhang. Recommender system application developments: a survey. Decision Support Systems, 74:12–32, 2015.

[24] Martin F. McKinney and Jeroen Breebaart. Features for audio and music classification. In Proceedings of the International Symposium on Music Information Retrieval, pages 151–158, 2003.

[25] Sean M McNee, John Riedl, and Joseph A Konstan. Being accurate is not enough: how accuracy metrics have hurt recommender systems. In CHI'06 extended abstracts on Human factors in computing systems, pages 1097–1101. ACM, 2006.

[26] Denis Parra and Shaghayegh Sahebi. Recommender systems: Sources of knowledge and evaluation metrics. In Advanced Techniques in Web Intelligence-2, pages 149–175. Springer, 2013.

[27] Patrice Perny and Jean-Daniel Zucker. Preference-based search and machine learning for collaborative filtering:the "film-conseil" movie recommendation system, 2001.

[28] Alan Said and Alejandro Bellogín. Comparative recommender system evaluation: benchmarking recommendation frameworks. In Proceedings of the 8th ACM Conference on Recommender systems, pages 129–136. ACM, 2014.

[29] Badrul Sarwar, George Karypis, Joseph Konstan, and John Reidl. Item-based collaborative filtering recommendation algorithms. In Proceedings of the 10th International Conference on World Wide Web,WWW'01, pages 285–295, New York, NY, USA, 2001. ACM.

[30] Mohammad Soleymani, Anna Aljanaki, Frans Wiering, and Remco C Veltkamp. Content-based music recommendation using underlying music preference structure. In Multimedia and Expo (ICME), 2015 IEEE International Conference on, pages 1–6. IEEE, 2015.

[31] Aaron Van den Oord, Sander Dieleman, and Benjamin Schrauwen. Deep content-based music recommendation. In Advances in neural information processing systems, pages 2643–2651, 2013.

[32] Xinxi Wang and Ye Wang. Improving content-based and hybrid music recommendation using deep learning. In Proceedings of the 22nd ACM international conference on Multimedia, pages 627–636. ACM, 2014.

[33] Meng-LunWu, Chia-Hui Chang, and Rui-Zhe Liu. Integrating content-based filtering with collaborative filtering using co-clustering with augmented matrices. Expert Systems with Applications, 41(6):2754–2761, 2014.

[34] Wen Wu, Liang He, and Jing Yang. Evaluating recommender systems. In Digital Information Management (ICDIM), 2012 Seventh International Conference on, pages 56–61. IEEE, 2012.

[35] Zied Zaier, Robert Godin, and Luc Faucher. Evaluating recommender systems. In Automated solutions for Cross Media Content and Multi-channel Distribution, 2008. AXMEDIS'08. International Conference on, pages 211–217. IEEE, 2008.

[36] Fuzheng Zhang, Kai Zheng, Nicholas Jing Yuan, Xing Xie, Enhong Chen, and Xiaofang Zhou. A novelty-seeking based dining recommender system. In Proceedings of the 24th International Conference on World Wide Web, pages 1362–1372. ACM, 2015.

[37] Mi Zhang and Neil Hurley. Avoiding monotony: improving the diversity of recommendation lists. In Proceedings of the 2008 ACM Conference on Recommender Systems, RecSys '08, pages 123–130, New York, NY, USA, 2008. ACM.

[38] Yuan Cao Zhang, Diarmuid Ó Séaghdha, Daniele Quercia, and Tamas Jambor. Auralist: introducing serendipity into music recommendation. In Proceedings of the fifth ACM international conference on Web search and data mining, pages 13–22. ACM, 2012.

[39] Tao Zhou, Zoltán Kuscsik, Jian-Guo Liu, Matúš Medo, Joseph RushtonWakeling, and Yi-Cheng Zhang. Solving the apparent diversity-accuracy dilemma of recommender systems. Proceedings of the National Academy of Sciences, 107(10):4511–4515, 2010.

[40] Cai-Nicolas Ziegler, Sean M. McNee, Joseph A. Konstan, and Georg Lausen. Improving recommendation lists through topic diversification. In Proceedings of the 14th International Conference on World Wide Web, WWW'05, pages 22–32, New York, NY, USA, 2005. ACM.