Dimensions as Virtual Items: Improving the predictive ability of top-\(N\) recommender systems

Marcos Aurélio Domingues\(^a,b,^*\), Alípio Mário Jorge\(^b,d\), Carlos Soares\(^a,c\)

\(^a\)INESC TEC – INESC Technology and Science, Rua Dr. Roberto Frias, 378, 4200-465 Porto, Portugal
\(^b\)DCC-FCUP, University of Porto, Rua do Campo Alegre, 1021/1055, 4169-007 Porto, Portugal
\(^c\)Faculty of Economics, University of Porto, Rua Dr. Roberto Frias, 4200-464 Porto, Portugal
\(^d\)LIAAD/INESC TEC, Rua de Ceuta, 118, Andar 6, 4050-190 Porto, Portugal

1. Introduction

Most web sites offer a large number of information resources. Finding relevant content has, thus, become a challenge for users. Recommender systems have emerged in response to this problem. A recommender system for the web is an information filtering technology which can be used to recommend a set of items (e.g., movies, musics, books, news, images, web pages, etc.) that are likely to be of interest to the user (Resnick & Varian, 1997; Sarwar, Karypis, Konstan, & Riedl, 2000a). One of the best illustrations for such a recommender system is the one deployed by the Amazon web site,\(^1\) which informs a user that “Customers Who Bought This Item Also Bought . . .” or “Customers Viewing This Page May Be Interested in These Sponsored Links . . .” (Linden, Smith, & York, 2003; Schafer, Konstan, & Riedl, 2001).

Traditionally, the data that are most often available for recommender systems are web access data that represent accesses from users to pages. Therefore, the most common recommender systems focus on these two dimensions. Based on access
used by data warehouses and OLAP applications. More formally, given the dimensions of the multidimensional data model, the recommendation space for these dimensions as a Cartesian product \( D_1 \times D_2 \times \ldots \times D_t \), where \( t > 2 \), as

\[
M' : D_1 \times D_2 \times \ldots \times D_t \rightarrow R.
\]

In this paper we present a multidimensional approach, called DaVI (Dimensions as Virtual Items), that consists in inserting contextual and background information as new user–item pairs. The main advantage of this approach is that it can be applied in combination with several existing two-dimensional recommendation algorithms in order to improve the predictive ability of those algorithms. The DaVI approach was introduced in an earlier workshop paper (Domingues, Jorge, & Soares, 2009), where preliminary ideas of the approach were described focusing on contextual information and context-aware recommender system. Later, we realized that the DaVI approach could also be used, not only with contextual information but also with background information. Then, in Domingues, Jorge, and Soares (2011), we formalized the DaVI as a multidimensional approach that can make uses of contextual and background information, and presented a preliminary result of its use. In this paper we discuss the DaVI approach in a significantly greater depth and extension. In particular we propose a new classification for multidimensional recommender systems, we analyze the computational complexity and scalability of our approach, and we also make a depth empirical evaluation of it, including, a comparison of our approach against one more multidimensional algorithm.

1.1. Contributions of the paper

As part of this work, we have adapted the categorization of context-aware recommender systems proposed by Adomavicius and Tuzhilin (2008) to multidimensional recommendation approaches. We have also formalized our DaVI approach and demonstrated how it can be applied on two different top-N recommendation techniques, Item-based Collaborative Filtering and Recommendation Based on Association Rules. We also analyzed the computational complexity of our approach using these two recommendation techniques.

An important issue is which dimensions should be used by the DaVI approach, given that some dimensions are more informative than others. To address this issue, in this work we have proposed three different algorithms. The first one, called DaVI-BEST, evaluates and selects the best dimension in a data set to build the multidimensional recommendation model. The second algorithm, called DaVI-FS, combines the DaVI-BEST with a sequential forward selection algorithm in order to select the best combination of dimensions to build the model. The last algorithm, called DaVI-ALL, consists in the simple idea of applying the DaVI approach on all existing dimensions in a data set, at the same time, to build the multidimensional model.

Finally, we ran an extensive set of experiments in three different real world data sets to evaluate the effectiveness of the DaVI approach and its three algorithms. We also compared our approach against two approaches proposed in the literature: combined reduction-based approach (Adomavicius et al., 2005) and weight post-filtering approach (Panniello, Tuzhilin, Gorgoglione, Palmisano, & Pedone, 2009). Additionally, we analyzed the scalability of the three DaVI algorithms with respect to the number of dimensions available in a data set for building a multidimensional model.

1.2. Organization of the paper

The paper is organized as follows. In Section 2 we review the main multidimensional recommendation approaches proposed in the literature. In Section 3 we describe our multidimensional approach. In the proposed approach, it is important to accurately determine which dimensions should be included in the model, given that some dimensions are more informative than others. In Section 4 we address this issue by proposing three different algorithms: DaVI-BEST, DaVI-FS and DaVI-ALL. In Section 5 we empirically evaluate the three algorithms to answer three research questions: (1) is DaVI-BEST algorithm able to take advantage of the useful information in multidimensional data to achieve better predictive ability than
a two-dimensional recommender algorithm?, (2) Does the use of more than one additional dimension (DaVI-ALL algorithms) provide better predictive ability than using the single best dimension (DaVI-BEST algorithm)?, and (3) Does the DaVI-BEST algorithm present better predictive ability than other multidimensional algorithms proposed in the literature? Finally, we conclude the paper with a summary of the work, its contributions and possible paths for further development (Section 6).

2. Related work

In this section, we review some of the main multidimensional recommendation approaches proposed in the literature. We characterize them according to an adaptation of the taxonomy proposed by Adomavicius and Tuzhilin (2008). We divide methods into filtering, algorithmic and transformational. Filtering methods handle different values of the dimensions separately and are further divided into pre-and post-filtering methods. In pre-filtering, the additional dimensions are used to filter out irrelevant items before building the recommendation model. In post-filtering, the values of the dimensions are used to reorder or filter out recommendations after building the recommendation model. Algorithmic approaches consist of changing the recommendation algorithm, such that it is able to handle the additional dimensions. Finally, transformational approaches transform the original data taking the values of the dimensions into account.

2.1. Filtering approaches

The first multidimensional approach for recommender systems has been proposed by Adomavicius and Tuzhilin (2001a, 2001b) and extended in greater depth in Adomavicius et al. (2005). This is a pre-filtering approach, called combined reduction-based, which uses additional dimensions as labels for segmenting sessions. Segmented sessions are used to build the recommendation models. Here, a segment is defined as a subset of the overall set of sessions selected according to the values of attributes of an additional dimension or combinations of these values. Basically, the approach segments the sessions and determines for each segment, whether the extra-dimensional information outperforms the traditional recommendation method. Then, given a particular active session, it chooses the best dimensional segment and applies the two-dimensional recommendation algorithm on the segment to build a model and make the predictions. By segmenting sessions, the reduction-based approach reduces the problem of multidimensional recommendation to the traditional two-dimensional recommendation problem. Thus, all previous two-dimensional recommendation algorithms can be used for multidimensional recommendation.

Some pre-filtering approaches have been combined with OLAP. In Adomavicius and Tuzhilin (2001b) and Adomavicius et al. (2005), OLAP multidimensional data handling capabilities are integrated into recommender systems by defining three basic concepts for recommender systems: (1) multiple dimensions, (2) profiling capabilities, and (3) aggregation capabilities. Based on these three multidimensional concepts, Weng, Lin, and Chen (2009) implement a multidimensional recommendation structure and evaluate it on a movie web site. Additionally, the authors define the multi-facet concept for their multidimensional structure and use it to explain the ratings at multiple levels of OLAP hierarchies. In Li, Wang, Geng, and Dai (2007), the multidimensional collaborative filtering approach proposed by Adomavicius et al. (2005) is used to provide top-N recommendations in a framework for mobile commerce (m-commerce).

In Panniello et al. (2009), the authors analyze the use of additional dimensions (contextual information) in pre- and post-filtering approaches. For pre-filtering, the authors have used the approach proposed by Adomavicius et al. (2005). For post-filtering, they proposed two approaches: Weight and Filter. In post-filtering, we first ignore the additional dimensions in the data set and apply a traditional two-dimensional algorithm to build the recommendation model. Then, we compute the probability $P_d(u,i)$, with which a user $u$ accesses an item $i$ under the additional dimension $d$. This can be computed as the number of neighbors (users similar to $u$) who access the same item under the same dimension, divided by the total number of neighbors (Panniello et al., 2009). Finally, the probability $P_d(u,i)$ is used to reorder (Weight approach) or filter out (Filter approach) the two-dimensional recommendations.

2.2. Algorithmic approaches

Lu, Zhou, Qiu, and Deng (2008) propose a multidimensional recommendation model based on the Resource Space Model (RSM), which is defined as a semantic model for uniformly specifying and organizing resources in normal forms (Zhuge, 2004, 2007). The authors also propose a collaborative filtering approach based on reduction–aggregation to predict ratings, and a multidimensional recommendation operation language (MROL) to exploit their multidimensional model.

An attribute-aware Item-based Collaborative Filtering algorithm is proposed by Tso and Schmidt-Thieme (2005). The algorithm exploits the additional attributes/dimensions by changing the distance function, which computes the similarity between pairs of items, to include such attributes. In Cho, Lee, Jang, and Choi (2006), the authors propose a technique to measure the similarity between ratings allocated for additional dimensions (contextual information) and ratings allocated for items. In the recommendation process, the calculated similarities are used as weights for items to reorder the recommendations.
2.3. Transformational approaches

In Baltrunas and Ricci (2010), the authors propose a prediction approach that splits the ratings of an item into two subsets according to the value of a contextual variable (additional dimension). They claim that the split can be beneficial if the ratings within each newly obtained subset are more homogeneous and/or if the two new subsets are significantly different. The approach first splits the ratings of the items into two subsets, creating two new artificial items from each original item. The split is based on the additional dimension and the ratings in each resulting subset correspond to a certain value of the dimension. For instance, the ratings may have been obtained in “winter” or in “summer”, and in this case, the additional dimension is the season. Then, the approach tests if the two new items are significantly different. If this is the case, the original item in the ratings matrix is replaced by the two new items. When predicting a rating for an active session, the corresponding value of the active session is considered and can then be used.

In Hosseini-Pozveh, Nematbakhsh, and Movahhedinia (2009), the authors exploit additional dimensions (contextual information) in order to create new ratings data sets for recommender systems. Firstly, the different contexts are clustered according to the usage patterns of the users. For example, if the same item is accessed in two different contexts, these contexts will be regarded as similar. Thus, a cluster can contain the ratings of users under different contexts. Next, the ratings from a user to an item under different contexts are aggregated and a new user representing the aggregated ratings is created. Finally, all new users are selected in order to create a new data set that is used to build a two-dimensional model and generate the recommendations.

2.4. Summary

In Table 1, we summarize the classification of the multidimensional recommender systems discussed above plus our own method presented in this paper. As we can see, most proposals are pre-filtering approaches. Our proposal (DaVI), which is described in the next section, is classified as transformational, since we transform the initial dataset by combining the usage information with the additional dimensions.

3. Dimensions as Virtual Items (DaVI)

In this section, we present our approach, called DaVI (Dimensions as Virtual Items), that exploits multidimensional data using existing two dimensional recommender systems. The idea is to treat additional dimensions as virtual items, using them together with the regular items in a recommender system. Here, we assume that virtual items are only used to build the recommendation model. On the other hand, regular items are used to build the model and they can also be recommended.

Let \( p \) be the number of users \( U = \{u_1, u_2, \ldots, u_p\} \) and \( q \) the number of all possible items that can be recommended \( I = \{i_1, i_2, \ldots, i_q\} \). In addition, we have other dimensions (e.g., contextual or background information), \( D = \{D_1, D_2, \ldots, D_t\} \), where each dimension \( D \) comprehends a set of values, i.e., \( D = \{d_1, d_2, \ldots, d_j\} \). For example, the dimension \( Hour \) can define a set of integer values from 1 to 24. Now, let \( j \) be the number of historical multidimensional sessions in a web site \( S = \{s_1, s_2, \ldots, s_j\} \). Each session \( s \) is a tuple defined by a user \( u \in U \), a set of accessed items \( I_f \subseteq I \) and a set \( D_f \subseteq D_1 \cup D_2 \cup \cdots \cup D_t \) containing all the dimension values associated with the session \( s \), i.e., \( s = (u, I_f, D_f) \).

A multidimensional session can have two types of dimensions in terms of granularity: session-based dimensions and item-based dimensions. If a single dimension \( D \) is session-based, a session \( s = (u, I_f, D_f) \) has a single dimension value (virtual item) \( d \in D_f \) associated to the session \( s \). Here, the dimension value \( d \) can represent, for example, the hour or location from where the session is accessed. On the other hand, if the dimension \( D \) is item-based, a session \( s = (u, I_f, D_f) \) has the dimension values (virtual items) \( d_1, \ldots, d_q \) associated to respective items.
where the dimension values in $S'$ are indicated by $i_1, \ldots, i_q$ in the session $s'$. For example, if the dimension values $d_1, \ldots, d_q$ represent the genre of songs in a music web site, we will have the values associated to songs (items) in the session and not directly to the session as presented in the first case.

The DaVI approach consists in converting each multidimensional session $s' = (u, I, D)$ into an extended two-dimensional session $s'' = (u, I, D, D_v)$, where the values of the additional dimensions in $D_v$ are used as virtual items together with the regular items in $I$. The DaVI approach can also be applied to a subset of dimensions or even to a single dimension. For example, a multidimensional session $s' = (u, I, D_1) = (u, \{i_1, \ldots, i_q\}, \{d_1, \ldots, d_q\})$, with a single dimension $D_v \subseteq D_1$, can be converted into an extended two-dimensional session $s'' = (u, I, D_v) = (u, \{i_1, \ldots, i_q, d_1, \ldots, d_q\})$. Thus, we have defined the DaVI approach as an operator that converts a set of multidimensional sessions into a set of extended two-dimensional sessions,

$$S'' = \text{DaVI}(S', \hat{D}),$$

where $S''$ is the set of extended two-dimensional sessions, $S'$ is the set of multidimensional sessions and $\hat{D} \subseteq D$ is a set indicating which dimension values in $S'$ must be converted to virtual items.

Once we have a set of extended two-dimensional sessions $S''$, building/learning a multidimensional recommendation model $M'$ consist in applying a two-dimensional recommender algorithm on $S''$. We illustrate the learning process using the DaVI approach in Fig. 1, where the values of the additional dimension Hour are used as virtual items.

Finally, to generate the recommendations, we use the multidimensional model $M'$ providing to it with the items and additional dimensions (transformed in virtual items by the DaVI approach) from the active user session $s''_u = (u, I_u \cup D_u)$ as follows:

$$R = M'(I_u \cup D_u),$$

where $I_u \cup D_u$ is referred to a set of observable items $O$, and it contains the items ($I_u$) and dimension values ($D_u$) which are, respectively, the regular and virtual items from the active user session $s''_u$. $R$ is a set of items/recommendations, such that $R \subseteq I$ and $R \cap O = \emptyset$, that are the most relevant/interesting for the user $u$, according to the model $M'$. As stated before, virtual items cannot be recommended. Thus, we apply a filter on the recommendations generated by the model $M'$ in order to guarantee that the model will never recommend virtual items.

One important advantage of our approach is that it can be applied to different recommendation methods. This means that DaVI makes it easy to apply existing recommender algorithms to multidimensional data and obtain multidimensional models without changing the algorithms. In the following sections we demonstrate how to apply DaVI with two different top-$N$ recommender algorithms: Item-based Collaborative Filtering and Association Rules based.

### 3.1. Item-Based Collaborative Filtering

The Item-based Collaborative Filtering technique analyzes web items in order to identify relations among them (Karypis, 2001). Here, the two-dimensional recommender model $M$ is a matrix representing the similarities between all the pairs of

![Fig. 1. Illustration of the learning process using the DaVI approach.](image)

<table>
<thead>
<tr>
<th>Table 2</th>
<th>Item-item similarity matrix.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$i_1$</td>
</tr>
<tr>
<td>$i_1$</td>
<td>1</td>
</tr>
<tr>
<td>$i_2$</td>
<td>$\text{sim}(i_2,i_1)$</td>
</tr>
<tr>
<td>$i_3$</td>
<td>$\text{sim}(i_3,i_1)$</td>
</tr>
</tbody>
</table>
items, according to a similarity metric. An abstract representation of a similarity matrix is shown in Table 2. Each item $i \in I$ is an accessed item, for example, a web page.

According to Deshpande and Karypis (2004a), the properties of the model and consequently the effectiveness of this recommendation algorithm depend on the method used to calculate the similarity among the items. To calculate the similarity between pairs of items, for example, $i_1$ and $i_2$, we first isolate the users who have rated both of these items, and then, we apply a metric on the ratings to compute the similarity $\text{sim}(i_1, i_2)$ between $i_1$ and $i_2$. In Sarwar, Karypis, Konstan, and Riedl (2001), the authors present three metrics to measure similarity between pairs of items: cosine angle, Pearson’s correlation and adjusted cosine angle. In this paper, we use the cosine angle metric, defined as

$$
\text{sim}(i_1, i_2) = \cos(\overrightarrow{i_1}, \overrightarrow{i_2}) = \frac{i_1 \cdot i_2}{\|i_1\| \cdot \|i_2\|},
$$

(4)

where $\overrightarrow{i_1}$ and $\overrightarrow{i_2}$ are rating vectors with as many positions as existing users in the set $U$. The operator “$\cdot$” denotes the dot-product of the two vectors. In our case, the rating vectors are binary. The value 1 means that the users accessed the respective item. The value 0 is the opposite.

Once we obtain the recommendation model, we can generate the recommendations. Given an active session $s_a$ containing a user $u_a$ and its set of observable items $O \subseteq I$, the model generates the $N$ recommendations as follows. First, we identify the set of candidate items for recommendation $C$ by selecting from the model all items $i \notin O$. Then, for each candidate item $c \in C$, we calculate its similarity to the set $O$ as

$$
\text{sim}_{i,O} = \frac{\sum_{i \in K_c \cap O} \text{sim}(c, i)}{\sum_{i \in K_c} \text{sim}(c, i)},
$$

(5)

where $K_c$ is a set with the $k$ most similar items (the nearest neighbors) to the candidate item $c$.

Finally, we select the $N$ candidate items with the highest similarity to the set $O$ and recommend them to the user $u_a$.

When we apply DaVI to the Item-based Collaborative Filtering algorithm, it first adds extra user–item pairs in the data set, where the item represents a dimension value (virtual item). Then, using this new data set as input, the recommendation algorithm creates a similarity matrix with rows and columns for each regular and virtual item, and calculates the similarity values between all the pairs of items.

A representation of a similarity matrix with the additional dimension $\text{Hour} = \{d_{10}, d_{24}, \ldots, d_{24}\}$ is shown in Table 3. The recommendations are generated as described above. For an active session $s_a$ occurring, e.g., at hour 10, the recommendations are the set of items that are the most similar to its set of observable items $O \subseteq I_{10} \cup d_{10}$, which contains regular items ($I_{10}$) and the hour 10 ($d_{10}$) as a virtual item. Although the multidimensional data are used by the model, only items are recommended.

A rationale behind DaVI with the Item-based Collaborative Filtering is that the similarity between a given item and a dimension value is higher if the item tends to be accessed at that dimension value. This way, the relation between items and the dimension values (virtual items) is captured. For example, the similarity between a given item and a particular hour is higher if the item tends to be accessed at that hour. When a recommendation is made for an active session, the value of the dimension on that particular session is used to provide the additional information. For example, the hour of the day the active session is taking place is also used to generate the recommendations.

### 3.2. Recommender system based on Association Rules

A two-dimensional recommender model $M$ based on Association Rules is a set of rules. Each rule $m$ has the form $m:X \rightarrow Y$, where $X \subseteq I$ and $Y \subseteq I$ are sets of items and $X \cap Y = \emptyset$. Each association rule is characterized by two metrics: support and confidence. The support of a rule in a data set $S$, where $S$ is a collection with $j$ sets of items (or sessions), is defined as

$$
\text{support}(X \rightarrow Y) = \frac{|X \cup Y|}{j},
$$

(6)

where $|X \cup Y|$ is the number of sessions in $S$ that contain all items in $X \cup Y$ and $j$ is the number of sessions in $S$.

The confidence of a rule is the proportion of the number of sessions which contain $X \cup Y$ with respect to number of sessions that contain $X$, and can be formulated as

<table>
<thead>
<tr>
<th>$i_1$</th>
<th>$i_2$</th>
<th>$d_1$</th>
<th>$d_24$</th>
<th>$d_{24}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$i_1$</td>
<td>1</td>
<td>$\text{sim}(i_1, i_2)$</td>
<td>$\text{sim}(i_1, d_1)$</td>
<td>$\text{sim}(i_1, d_{24})$</td>
</tr>
<tr>
<td>$i_2$</td>
<td>$\text{sim}(i_2, i_1)$</td>
<td>1</td>
<td>$\text{sim}(i_2, d_1)$</td>
<td>$\text{sim}(i_2, d_{24})$</td>
</tr>
<tr>
<td>$d_1$</td>
<td>$\text{sim}(d_1, i_1)$</td>
<td>$\text{sim}(d_1, i_2)$</td>
<td>1</td>
<td>$\text{sim}(d_1, d_{24})$</td>
</tr>
<tr>
<td>$d_{24}$</td>
<td>$\text{sim}(d_{24}, i_1)$</td>
<td>$\text{sim}(d_{24}, i_2)$</td>
<td>$\text{sim}(d_{24}, d_1)$</td>
<td>1</td>
</tr>
</tbody>
</table>
\[
\text{confidence}(X \rightarrow Y) = \frac{|X \cup Y|}{|X|}.
\]  

(7)

Discovering all association rules from a data set \( S \) consists in generating all rules whose support and confidence are greater than or equal to the corresponding minimal thresholds, called \( \text{minsup} \) and \( \text{minconf} \). The classical algorithm for discovering association rules is Apriori (Agrawal & Srikant, 1994).

To build the recommender model \( M \) using association rules, each session is represented as a set of pairs \( (s,i) \) with the same \( s \), where \( s \) and \( i \) respectively identify the session and the accessed item. These sessions are used as input to an association rules algorithm to generate a set of rules. Once we have the model, we can make recommendations, \( R \), to a new session. Given an active session \( s_a \) containing a user \( u_a \) and its set of observable items \( O \), we build the set \( R \) as follows (Jorge, Alves, & Azevedo, 2002, 2003):

\[
R = \{ \text{consequent}(m) | m \in M \text{ and } \text{antecedent}(m) \subseteq O \text{ and } \text{consequent}(m) \neq O \}.
\]  

(8)

To obtain the top-\( N \) recommendations, we select from \( R \) the \( N \) distinct recommendations corresponding to the rules with the highest confidence values.

Extending association rules to handle additional dimensions by applying \text{DaVI} consists in including extra pairs \( (s,d) \) into the former set of sessions, where the item represents a dimension value (virtual item). For example, to use the dimension \( \text{Hour} = \{1,2,\ldots,24\} \), we add extra pairs \( (s,d) \) to the respective sessions, where \( d \) represents the hour of the day when the sessions occurred. The set of augmented sessions is used as input by an association rules algorithm to generate extended rules such as

\[
\{i_1, i_2, 10\} \rightarrow \{i_4\},
\]  

(9)

which means a person who accesses the items \( i_1 \) and \( i_2 \) at around 10 tends to access the item \( i_4 \).

Once we have a set of extended rules \( M' \), we can output the set of recommendations \( R \) using Eq. (8). In this case, the set of observables \( O \) contains regular and virtual items. Moreover, although some rules can contain virtual items in their consequent, a filter is applied on the rules to guarantee that only regular items are recommended.

### 3.3. Analysis of complexity

In this section we analyze the computational complexity of the \text{DaVI} approach using the Item-based Collaborative Filtering (CF) and Association Rules based (AR) algorithms as base recommenders. According to Deshpande and Karypis (2004a), the complexity of the CF algorithm for building a recommendation model is \( O(q^2 \times j) \), where \( q \) is the number of items and \( j \) is the number of sessions. Using CF as base recommender, the computational complexity of the \text{DaVI} approach is \( O((q + v)^2 \times j) \), where \( v \) denotes the sum of the quantity of different values in each dimension. Therefore, in terms of number of dimension values, the complexity is \( O(v^2) \). Fig. 2a illustrates the expected computational behavior of the \text{DaVI} approach. For this figure, we define a small number of items \( (q = 20) \) and sessions \( (j = 100) \) in order to emphasize the behavior of the \text{DaVI} approach with respect to the number of values \( v \). In Fig. 2b, we use a higher number of items \( (q = 200) \) and sessions \( (j = 1000) \) to confirm that the impact of \( v \) is reduced if \( q \) and \( j \) are much larger than \( v \).
Regarding the AR technique, the complexity is \( O(b \times j \times 2^t) \), where \( j \) is the number of sessions, \( t \) is the length of the longest frequent itemset, and \( b \) is the number of maximal frequent itemsets (Zaki, 2004). Assuming \( t \) as the number of dimensions in the longest frequent itemset, we can say that the computational complexity for the DaVI approach using the AR technique is \( O(b \times j \times 2^t) \). Thus, the complexity with respect to the number of dimensions is \( O(2^t) \). In Fig. 3, we can see the expected computational behavior of the DaVI approach using the AR technique. We estimate the values \( b = 14, t = 7 \) and \( j = 30,000 \) based on information of our largest data set (Section 5.1).

Although the complexity of the DaVI approach using the AR technique is exponential, in practice, we usually have a small number of dimensions in the longest frequent itemset (e.g., from 1 to 3). Moreover, the complexity is more favorable when the data sets are sparse (Angiulli, Ianni, & Palopoli, 2004). We have confirmed empirically this fact in Section 5.3.4.

4. DaVI based algorithms

As stated before, an important issue with respect to the DaVI approach is to determine which dimensions should be included in a recommendation model, given that some dimensions are more informative than others. This is related to the problem of feature selection that has been extensively addressed in data mining (Liu & Motoda, 1998). In this section, we address this problem by proposing three different algorithms. The first one, called DaVI-BEST, evaluates and selects the best dimension in a data set to build the multidimensional recommendation model. The second algorithm, called DaVI-FS, combines DaVI-BEST with a sequential forward selection algorithm in order to select the best combination of dimensions to build the multidimensional model. The last algorithm, called DaVI-ALL, consists in the simple idea of applying the DaVI approach simultaneously on all existing dimensions in a data set to build the multidimensional model.

4.1. DaVI-BEST algorithm

To determine the best dimension for a given top-\( N \) recommender algorithm \( A \), the DaVI-BEST algorithm first applies the DaVI approach on each candidate dimension and builds its respective multidimensional recommendation model. Then, it evaluates each model and selects the best dimension, the one whose recommendation model presents the best performance. The original set of training data is split into two parts: one which is used to learn the model (training set) and another to evaluate it, called the validation set.

To evaluate each of the candidate recommendation models, we need an evaluation metric. There are several metrics which are used to evaluate the performance of recommender algorithms, such as Mean Absolute Error (MAE), Mean Squared Error (MSE), Precision, Recall, F1, ROC Curves, Coverage, Learning Rate, and Novelty (Herlocker, Konstan, Terveen, & Riedl, 2004). From these metrics, Precision, Recall and F1 are the most common metrics used to evaluate top-\( N \) recommender algorithms, since they focus on recommending high quality items (Adomavicius et al., 2005; Basu, Hirsh, & Cohen, 1998; Billsus & Pazzani, 1998; Huang, Zeng, & Chen, 2007; Kwon, 2008; Rendle, Marinho, Nanopoulos, & Schmidt-Thieme, 2009; Sarwar et al., 2000a, Sarwar, Karypis, Konstan, & Riedl, 2000b; Symeonidis, Nanopoulos, & Manolopoulos, 2009; Tso & Schmidt-Thieme, 2006; Zanker, 2008). Although both Precision and Recall are very important for the quality judgment of top-\( N \) recommendations, we do not use them directly here because they are often conflicting in nature (Cleverdon, Mills, & Keen, 1966). For instance, increasing the number of recommendations \( N \) tends to increase Recall but decrease Precision, and vice versa. Therefore, instead of Precision and Recall, we use the F1 metric. This metric combines Precision and Recall with equal
weights in a harmonic mean and has a simple interpretation. It ranges from 0 to 1 and higher values indicate better recommendations.

We evaluate each multidimensional recommendation model using the All But One protocol (Breese, Heckerman, & Kadie, 1998) with the n-fold cross validation technique (Mitchell, 1997). The sessions in the data set are randomly partitioned into n subsets. For each fold, we use n – 1 of those subsets of data for training and the remaining data for validation. The training set is used to build the recommendation model. For each session in the validation set, we randomly hide one regular item (never a virtual item), referred to as the singleton set \( H \). The remaining items represent the set of observables, \( O \), based on which the top-N recommendations are generated. The F1 metric is computed by comparing, for each session in the validation set, the set of recommendations generated \( R \) against the singleton set \( H \) for that session, as follows:

\[
\text{Precision} = \frac{|R \cap H|}{|R|},
\]

\[
\text{Recall} = \frac{|R \cap H|}{|H|},
\]

\[
F1 = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}.
\]

Once we have F1 values for each candidate dimension and fold, we can select the best one. Firstly, we select the dimensions whose F1 values are significantly higher than F1 values of the pure two-dimensional recommendation model (without additional dimensions). To do that, we apply the paired t-test, with a 95% confidence level, to the n folds. Then, we compute the average of the F1 values for each candidate dimension selected in the previous step, and select the one with the highest F1 value to build the final multidimensional recommendation model on the whole data set. The DaVI-BEST algorithm is presented in Algorithm 1.

**Algorithm 1. DaVI-BEST algorithm**

**Input:** A set of multidimensional sessions \( S' = \{ s'_1, s'_2, \ldots, s'_t \} \), where each session \( s' \) is a tuple defined by a user \( u \in U \), a set of accessed items \( I_u \subseteq I \) and a set of dimension values \( D_u \subseteq D_1 \cup D_2 \cup \cdots \cup D_l \); \( A \), a top-N recommender algorithm; \( n \), the number of folds which are used to evaluate the multidimensional models; \( N \), the number of recommendations generated during the evaluation of the models.

**Output:** \( \overline{M} \), an object containing the final two-dimensional or multidimensional recommendation model.

1. \( \mu_s := \emptyset \); \{F1 values, for each fold, calculated using the two-dimensional models\}
2. \( \mu_{s,d}^f := \emptyset \); \{F1 values, for each fold and dimension, calculated using the multidimensional models\}
3. \( D^+ := \emptyset \); \{Set of pairs (dimension,F1 values) for informative dimensions\}
4. \( \mathcal{F} := \text{create-folds}(S', n) \);
5. **for all** folds \( F \in \mathcal{F} \) **do**
6. \( M_F := A(\mathcal{F} - F) \);
7. \( \mu_F := \text{eval} (M_F,F) \);
8. **end for**
9. **for all** dimensions \( D \in D \) **do**
10. **for all** folds \( F \in \mathcal{F} \) **do**
11. \( M_{F,D} := A(\text{DaVI}(\mathcal{F} - F, D)) \);
12. \( \mu_{F,D} := \text{eval} (M_{F,D}, \text{DaVI}(F, D)) \);
13. **end for**
14. **if** t-test \( \mu_{F,D} > \mu_F, \alpha = 0.05 \) **then**
15. \( D^+ := D^+ \cup < D, \mu_{F,D} > ; \)
16. **end if**
17. **end for**
18. **if** \( D^+ \neq \emptyset \) **then**
19. \( D^* := \arg \max_{D \in D^+} \{ F1(D^+) \} \);
20. \( \overline{M} := A(\text{DaVI}(S', D^*)) \);
21. **else**
22. \( \overline{M} := A(S') \);
23. **end if**
24. **return** \( \overline{M} \);
Firstly, the algorithm sets to “∅” (empty set) the variables \( μ_\emptyset \), \( μ_{x,\emptyset} \), and \( D^+ \). The variables \( μ_\emptyset \) and \( μ_{x,\emptyset} \) denote the values of the evaluation measure (F1, in this case) for each fold in \( F \) and dimension in \( D \). The variable \( D^+ \) stores the informative dimensions and respective F1 values in the form \((\text{dimension}, \text{F1 values})\). Here, a dimension is informative if its respective multidimensional model presents an F1 value significantly higher than the F1 value of the two-dimensional model. In line 4, the function \(_\text{create-folds}\) partitions the sessions into \( n \) folds which are used to evaluate the dimensions through their respective multidimensional recommendation models.

The next step (lines 5–8 of the Algorithm 1) consists of building \( n \) two-dimensional models from all folds but one and evaluating them on the remaining fold. To build a model, we can use any two-dimensional top-N recommender algorithm. The function \(_A\) builds a model and the function \(_\text{eval}\) evaluates it against the validation set, calculating the F1 metric as described before. These F1 values will be used as reference to evaluate the performance of the multidimensional models in lines 14–16.

The evaluation and selection of informative dimensions are performed in the lines 9–17 of the Algorithm 1. Lines 10–13 build the multidimensional models for each fold and evaluate them on the corresponding validation data (using the function \(_\text{eval}\) to compute the F1 metric). The function \(_\text{DaVI}\) converts a set of multidimensional sessions into a set of extended two-dimensional sessions, as described in Section 3. Lines 14–16 analyze whether the F1 values of the multidimensional models are significantly higher than the F1 values of their respective pure two-dimensional models (without additional dimensions) or not. The analysis is performed using the function \(_t\)-test that computes a one-sided paired \(_t\)-test with a 95% confidence level (significance level \( x = 0.05 \)). In line 15, the dimensions that are informative are stored in the set \( D^+ \) with their respective F1 values.

Finally, we test whether the set \( D^+ \) is empty or not. If it is not, we use the function \(_\text{argmax}\) to return the dimension \( D^+ \in D^+ \) which provides the highest F1 value (line 19). Then, using the dimension \( D^+ \), we apply the \(_\text{DaVI}\) approach on the whole set of sessions \( S' \) to build the final multidimensional recommendation model (line 20). If the set \( D^+ \) is empty, we use the whole set of sessions \( S' \) to build the two-dimensional recommendation model (line 22). The final multidimensional or two-dimensional recommendation model is returned in line 24 of Algorithm 1.

### 4.2. \(_\text{DaVI}-\text{FS}\) algorithm

The \(_\text{DaVI}-\text{BEST}\) algorithm identifies one single informative dimension: the best one. In this section, we propose \(_\text{DaVI}-\text{FS}\), which employs forward selection (Jain & Zongker, 1997; Liu & Motoda, 1998) to identify the best combination of dimensions. \(_\text{DaVI}-\text{FS}\) is a generalization on \(_\text{DaVI}-\text{BEST}\) and also uses the \(_\text{DaVI}\) operator. The sequential forward selection algorithm applied is a simple greedy search algorithm that starts from an empty set of selected dimensions, \( \bar{C} = \emptyset \), and sequentially adds the dimension \( D \) that results in the greatest improvement for an objective function \( J(\bar{C} + D) \), where \( \bar{C} \) represents the previously selected dimensions. The \(_\text{DaVI}-\text{FS}\) algorithm is presented in Algorithm 2.

Similarly to the \(_\text{DaVI}-\text{BEST}\) algorithm, we use variables \( μ_\emptyset \), \( μ_{x,\emptyset} \), and \( \bar{C} \) to store, respectively, the F1 values for the two-dimensional recommendation models, the F1 values for the multidimensional models (using combinations of dimensions) and the pair \((\text{combination of dimensions}, \text{F1 values})\) for the best combination. In line 5, the function \(_\text{create-folds}\) generates the \( n \) folds which are used to evaluate the combinations of dimensions through their respective multidimensional recommendation models.

#### Algorithm 2. \(_\text{DaVI}-\text{FS}\) algorithm

Input: A set of sessions \( S' = \{s'_1, s'_2, \ldots, s'_l\} \), where each session \( s'_i \) is a tuple defined by a user \( u \in U \), a set of accessed items \( I_u \subseteq I \) and a set of dimension values \( D_{s'_i} \subseteq D_1 \cup D_2 \cup \ldots \cup D_M \); \( A \), a top-N recommender algorithm; \( n \), the number of folds which are created to evaluate the multidimensional models; \( N \), the number of recommendations generated during the evaluation of the models.

Output: \( M_\emptyset \), an object containing the final two-dimensional or multidimensional recommendation model.

1: \( μ_\emptyset := \emptyset \); \((\text{F1 values, for each fold, calculated using the two-dimensional models})\)
2: \( μ_{x,\emptyset} := \emptyset \); \((\text{F1 values, for each fold and combination of dimensions, calculated using the multidimensional models})\)
3: \( \bar{C} := \emptyset \); \((\text{A pair } (\text{combination of dimensions}, \text{F1 values}) \text{ which stores the best combination})\)
4: \( \text{stop} := \text{false}; \)
5: \( F := _\text{create-folds}(S', n); \)
6: \( \text{for all folds } F \in F \text{ do} \)
7: \( M_F := A(F - F); \)
8: \( μ_F := _\text{eval}(M_F, F); \)

(continued on next page)
9: end for
10: do
11: \( C^+ := \emptyset \); (Set of pairs \( \text{combination of dimensions,F1 values} \) for informative combinations)
12: for all dimensions \( D \mid D \in \mathcal{D} \) and \( D \notin \widetilde{C} \) do
13: for all folds \( F \in \mathcal{F} \) do
14: \( \widetilde{M}'_{F, \widetilde{C} + D} := A(\text{DaVI}(\mathcal{F} - F, \widetilde{C} + D)) \);
15: \( \mu'_{F, \widetilde{C} + D} := \text{eval}(\widetilde{M}'_{F, \widetilde{C} + D}; \text{DaVI}(F, \widetilde{C} + D)) \);
16: end for
17: if \( t\text{-test} \left( \mu'_{F, \widetilde{C} + D} > \mu_F, \alpha = 0.05 \right) \) then
18: \( C^+ := C^+ \cup \langle \widetilde{C} + D, \mu'_{F, \widetilde{C} + D} > \);
19: end if
20: end for
21: \( \tilde{C} := \text{argmax}_{C^+ \in C^+} \{F1(C^+)\} \);
22: if \( \text{mean}(C^+) > \text{mean}(\tilde{C}) \) then
23: \( \tilde{C} := C^+ \);
24: else
25: stop := true;
26: end if
27: while stop = false
28: if \( \tilde{C} \neq \emptyset \) then
29: \( \tilde{M} := A(\text{DaVI}(\mathcal{S}', \tilde{C})) \);
30: else
31: \( \tilde{M} := A(\mathcal{S}') \);
32: end if
33: return \( \tilde{M} \);

Lines 6–9 of the Algorithm 2 build two-dimensional models (without the combination of additional dimensions) and evaluate them for the \( n \) folds. Again, we can use any two-dimensional top-\( N \) recommender algorithm in order to build the models. Here, we also use the function \( \text{eval} \) to evaluate the models, which consists in calculating the F1 metric. These F1 values will be used as reference to analyze the statistical significance of the gains obtained with the multidimensional models for each combination of dimensions (lines 17–19).

The sequential forward selection algorithm is applied in lines 10–27 of the Algorithm 2. Lines 12–20 build the multidimensional models by combining the current set of selected dimensions and each of the remaining ones. The models are evaluated for each fold by calculating their F1 values. Here, the function \( \text{DaVI} \) converts a set of multidimensional sessions into a set of extended two-dimensional sessions, as described in Section 3. The new combinations of dimensions whose respective recommendation models are significantly better than the two-dimensional model are stored in the set \( C^+ \) with their respective F1 values. In line 21, the function \( \text{argmax} \) returns the combination \( C^+ \in C^+ \) which provides the highest value for the objective function, i.e., F1 value.

Given that sequential forward selection is a computationally expensive algorithm, we define a constraint that must be tested before moving to the next iteration. In lines 22–26 of the Algorithm 2, we test whether the mean F1 value for the current best combination of dimensions is higher than the mean F1 value for the previous best combination or not. If the F1 value for the current best combination is higher, we will move to the next iteration (lines 10–27). Otherwise, we stop the algorithm and return the final recommendation model. Although such a constraint can improve the time performance of the algorithm, it also increases the possibility of selecting a locally optimal combination instead of a globally optimal one.

As in \( \text{DaVI-BEST} \), the algorithm may return a multidimensional or a two-dimensional model (lines 28–33).

4.3. \( \text{DaVI-ALL} \) algorithm

Finally, we define an algorithm that uses all the dimensions without any selection strategy, which is presented in Algorithm 3. In line 1, we test whether there are dimensions to build a multidimensional model or not. If there are dimensions, the function \( \text{DaVI} \) converts all of them into virtual items simultaneously to build the final multidimensional recommenda-
tion model (line 2). Otherwise, the algorithm builds a pure two-dimensional recommendation model (line 4). The final model is returned in line 6.

**Algorithm 3. DaVI-ALL algorithm**

**Input:** A set of sessions \( S' = \{s'_1, s'_2, \ldots, s'_j\} \), where each session \( s'_j \) is a tuple defined by a user \( u \in U \), a set of accessed items \( I_u \subseteq I \) and a set of dimension values \( D_u \subseteq D_1 \cup D_2 \cup \ldots \cup D_t \); \( A \), a top-N recommender algorithm.

**Output:** \( M \), an object containing the final two-dimensional or multidimensional recommendation model.

1: if \( D \neq \emptyset \) then
2: \( M := A(\text{DaVI}(S', D)) \);
3: else
4: \( M := A(S') \);
5: end if
6: return \( M \);

### 4.4. Generating top-N recommendations with multidimensional models

Once we have a multidimensional recommender model in \( M \), we can generate the recommendations. Algorithm 4 generates recommendations using a multidimensional model built by the DaVI-BEST, the DaVI-FS or the DaVI-ALL algorithm.

**Algorithm 4. Algorithm for top-N recommendations with a DaVI multidimensional model**

**Input:** An object \( M \) containing a recommender model; an active session \( s'_u = \langle u, I_u, D_u \rangle \) defined by a user \( u \in U \), a set of accessed items \( I_u \subseteq I \) and a set of dimension values \( D_u \subseteq D_1 \cup D_2 \cup \ldots \cup D_t \); the number of recommendations \( N \) that will be generated.

**Output:** \( R \), the top-N recommendations for the active session \( s'_u \).

1: if \( M \) contains a multidimensional model then
2: \( M' := M \);
3: \( s''_u := \text{DaVI}(s'_u, D_u) \); \( \{ \text{An extended session } s''_u = \langle u, I''_u \cup D''_u \rangle, \text{ where } I''_u \cup D''_u \text{ is referred to a set of observable items, } O, \text{ containing the regular } (I''_u) \text{ and virtual } (D''_u) \text{ items} \} \)
4: \( R := \text{filter}(M'(I''_u \cup D''_u)) \);
5: else
6: \( M := M \);
7: \( R := M(I'_u) \);
8: end if
9: return \( R \);

In line 1, the algorithm analyzes whether \( M \) contains a multidimensional or a two-dimensional model. If it contains a multidimensional model \( M' \), we retrieve it to generate the set of \( N \) recommendations \( R \) for the active session \( s''_u \) (lines 2–4). In this case, regular and virtual items are used to generate the recommendations but the function \( \text{filter} \) guarantees that only regular items are recommended. On the other hand, if \( M \) contains a two-dimensional model \( M \), we retrieve it to generate the \( N \) recommendations using only regular items as input (lines 6–7). Line 9 of the Algorithm 4 returns the top-N recommendations \( R \) for the active session \( s''_u \).

### 5. Empirical evaluation

In this section we empirically evaluate the ability of the DaVI approach to improve the recommendations of the Item-based Collaborative Filtering (Section 3.1) and Association Rules based (Section 3.2) algorithms. Basically, we answer three research questions:

1. Is the DaVI-BEST algorithm able to take advantage of the useful information in multidimensional data to achieve better predictive ability than a two-dimensional recommender algorithm?
2. Does the use of more than one additional dimension (DaVI-FS and DaVI-ALL algorithms) provide better predictive ability than using the single best dimension (DaVI-BEST algorithm)?
3. Does the DaVI-BEST algorithm present better predictive ability than other multidimensional algorithms proposed in the literature?

Additionally, we also evaluate the scalability of the three DaVI based algorithms with respect to the number of dimensions available in a data set for building a multidimensional model.

5.1. Data sets

A major challenge in multidimensional recommendation research is the lack of large scale annotated data sets (Li et al., 2010; Palmisano, Tuzhilin, & Gorgoglione, 2008; Verbert et al., 2012). Previous studies usually experiment on a small data set collected through user studies. Although undoubtedly useful, this approach is limited because the user studies are usually very expensive and their scale small.

In this paper the evaluation is carried out on three different real world data sets. The first two data sets come from Palco Principal, a web site of Portuguese music. The first one, called Listener, contains accesses to music tracks of the site. Each session in this data set represents all accesses from a user to music tracks since the first enrollment of the user in the site. The data set has 62,208 accesses, 6428 different items (music tracks) and 9740 sessions. In addition, it presents a minimum of 2 items, a mean of 6.3 items and a maximum of 997 items per session. The second data set, called Playlist, represents the set of music tracks explicitly selected by registered users to include in their individual playlist. Here, each session corresponds to a playlist and contains the music tracks selected for the playlist. The data set has 37,022 accesses, 5428 different items (music tracks) and 4417 sessions. Furthermore, it presents a minimum of 2 items, a mean of 8.3 items and a maximum of 798 items per session.

The additional dimensions for both data sets are presented in Table 4.

The first group of dimensions is related to the time and location of the accesses and it is obtained by pre-processing the web access data. The second one consists of domain specific information and it is collected from the content management system (CMS) of the web site. The web site does not register time and location information for the Playlist, so, for this data set, we only have dimensions which are domain specific information collected from the CMS of the web site.

Besides the previous two new data sets, the evaluation is also carried out on a third data set, called Entree. This is a public data set that contains a record of user’s interactions with the Entree Chicago restaurant recommender system. The users interact with the system by stating their preferences with respect to a given restaurant, and the system recommends restaurants that are adequate for the users based on their preferences. A session in this data set represents all accesses from a user to music tracks since the first enrollment of the user in the site. The data set has 149,849 accesses, 639 different items (restaurants) and 31,440 sessions. It also has a minimum of 2 items, a mean of 4.7 items and a maximum of 47 items per session. The additional dimensions for both data sets are presented in Table 4.

The first group of dimensions is related to the time and location of the accesses and it is obtained by pre-processing the web access data. The second one consists of domain specific information and it is collected from the content management system (CMS) of the web site. The web site does not register time and location information for the Playlist, so, for this data set, we only have dimensions which are domain specific information collected from the CMS of the web site.

5.2. Experimental setup and evaluation metrics

To measure the predictive ability of the recommender systems, we calculate the Precision, Recall and F1 metrics using the All But One protocol with 10-fold cross validation as described in Section 4.1. Then, for each metric, the 10 global values are summarized using mean and standard deviation. To compare two recommendation algorithms, we apply the two-sided paired t-test, with a 95% confidence level, on the 10 global values of each metric. For the comparison, the null hypothesis considers that the algorithms are equal (in terms of Precision, Recall or F1 metric). The alternative hypothesis considers that the algorithms are different. We run the experiments for $N$ equal 1, 2, 3, 5 and 10, where $N$ is the number of items to be recommended by the top-$N$ recommender systems.

With respect to recommendation algorithms, we use the Item-based Collaborative Filtering (CF) and the Association Rules based (AR), which were described in Sections 3.1 and 3.2, respectively. In CF, the $N$ recommendations are generated based on their four most similar items (the four nearest neighbors). We ran a first set of experiments using different numbers of neighbors and analyzed the F1 metric for these experiments. We observed in our data sets that the values for F1 tend to increase from 2 to 4 neighbors and decrease from 4 to 5. We can see this behavior, for the Listener data set, in Fig. 4. Therefore, in the following experiments, we have chosen the four most similar items to generate the recommendations. In AR, the recommendation models are built using a minimum support value determined to keep at least 50% of the items in the data set. The minimum confidence values are defined as being the support value of the third most frequent item in the data set. This allows the generation of at least three rules without antecedent that can be used by default, in the case that no other rules applies.

---

2 http://www.palcoprincipal.pt/.
5.3. Empirical results

In this section, we answer the previous three research questions. We present empirical results (i.e., Precision, Recall and F1 values) and discuss them in order to answer each of them.

**Table 4**
Additional dimensions for the Listener and Playlist data sets. The first group of dimensions is related to the time and location of the accesses and the second one consists of domain specific information collected from a content management system (CMS).

<table>
<thead>
<tr>
<th>Dimension</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>day</td>
<td>Day of each access (from 01 to 31)</td>
</tr>
<tr>
<td>month</td>
<td>Month of each access (from 01 to 12)</td>
</tr>
<tr>
<td>week_day</td>
<td>Week day of each access (from Monday to Sunday)</td>
</tr>
<tr>
<td>work_day</td>
<td>If the accesses were made during the week (from Monday to Friday) or weekend (Saturday or Sunday)</td>
</tr>
<tr>
<td>hour</td>
<td>Hour of each access (from 01 to 24)</td>
</tr>
<tr>
<td>work_hour</td>
<td>If the accesses were made during working hours (from 8 a.m. to 6 p.m.) or not</td>
</tr>
<tr>
<td>location</td>
<td>Location where the accesses were made (country/city)</td>
</tr>
<tr>
<td>band</td>
<td>The band which plays a music track</td>
</tr>
<tr>
<td>music_genre</td>
<td>The genre of a music track (pop, rock, jazz, and so forth)</td>
</tr>
<tr>
<td>instrumental</td>
<td>If a music track is instrumental or not</td>
</tr>
</tbody>
</table>

**Table 5**
Additional dimensions for the Entree data set.

<table>
<thead>
<tr>
<th>Dimension</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>day</td>
<td>Day of each access (from 01 to 31)</td>
</tr>
<tr>
<td>month</td>
<td>Month of each access (from 01 to 12)</td>
</tr>
<tr>
<td>week_day</td>
<td>Week day of each access (from Monday to Sunday)</td>
</tr>
<tr>
<td>work_day</td>
<td>If the accesses were made during the week (from Monday to Friday) or weekend (Saturday or Sunday)</td>
</tr>
<tr>
<td>hour</td>
<td>Hour of each access (from 01 to 24)</td>
</tr>
<tr>
<td>work_hour</td>
<td>If the accesses were made during working hours (from 8 a.m. to 6 p.m.) or not</td>
</tr>
<tr>
<td>intention</td>
<td>The intention of navigation in a restaurant recommendation system (for example, the search for a restaurant cheaper, closer, more traditional, more creative, and so forth)</td>
</tr>
</tbody>
</table>

![Fig. 4. Analyzing the DaVI-BEST algorithm using the CF technique with different number of neighbors in the Listener data set.](image)

5.3. Empirical results

In this section, we answer the previous three research questions. We present empirical results (i.e., Precision, Recall and F1 values) and discuss them in order to answer each of them.
To do this, we have implemented Algorithms 1 and 4, using the CF and AR recommendation techniques, and tested them on information in multidimensional data to achieve better predictive ability than a two-dimensional recommender algorithm? The analysis shows that the model were implemented using internal 5-fold cross validation in order to reduce the computational time. Additionally, we have analyzed which additional dimensions are selected by the \textit{DaVI}-BEST algorithm. All the values are statistically significant. For \textit{Listener} and \textit{Playlist} data sets, we have F1 average gains of 26.16%, 13.66% and 4.58%, respectively.

5.3.1. Evaluating the \textit{DaVI}-BEST algorithm

In this section, we answer our first research question: \textit{Is the DaVI}-BEST algorithm able to take advantage of the useful information in multidimensional data to achieve better predictive ability than a two-dimensional recommender algorithm? To do this, we have implemented Algorithms 1 and 4, using the CF and AR recommendation techniques, and tested them on the three data sets. In Algorithm 1, the procedures that select the best additional dimension to build the multidimensional model were implemented using internal 5-fold cross validation in order to reduce the computational time.

When the base recommender is CF, the \textit{DaVI}-BEST algorithm is significantly better than the two-dimensional algorithm (\textit{user \times item}). This can be observed in Table 6. For the first two data sets, the \textit{DaVI}-BEST algorithm presents F1 gains ranging from 11.9% to 33.7% (\textit{Listener}), and from 7.6% to 25.4% (\textit{Playlist}). In the \textit{Entree} data set, the improvement of the models using the \textit{DaVI}-BEST algorithm is quite small (F1 average gain of 1.1%) although they are also statistically significant.

Additionally, we have analyzed which additional dimensions are selected by the \textit{DaVI}-BEST algorithm to build the final multidimensional models. The dimension \textit{band} is always selected in the \textit{Listener} and \textit{Playlist} data sets, independently of fold and value of \textit{N}. On the other hand, in the \textit{Entree} data set, the selected dimension varies depending on the fold and value of \textit{N}. The analysis shows that the \textit{DaVI}-BEST algorithm selects the dimension intention or week\textunderscore day to build the multidimensional models, and that sometimes it does not select any dimension in some folds. In these cases, the \textit{DaVI}-BEST algorithm outputs the two-dimensional model. In any case, for CF, the \textit{DaVI}-BEST algorithm is able to identify and exploit informative dimensions.

With respect to the AR models, our results also show that the \textit{DaVI}-BEST algorithm is significantly better than the two-dimensional algorithm (\textit{user \times item}). Table 7 presents the values (i.e., Precision, Recall and F1 metric) obtained using the \textit{DaVI}-BEST algorithm. All the values are statistically significant. For \textit{Listener}, \textit{Playlist} and \textit{Entree} data sets, we have F1 average gains of 26.16%, 13.66% and 4.58%, respectively.

For the AR technique, we have also analyzed which additional dimensions are selected by the \textit{DaVI}-BEST algorithm to build the final multidimensional models. Again, the dimension \textit{band} is always selected to build the multidimensional models for the \textit{Listener} and \textit{Playlist} data sets, independently of fold and value of \textit{N}. Unlike what has been observed with the CF technique, in the \textit{Entree} data set, the dimension \textit{intention} is always selected to build the multidimensional models with the AR technique. Again, \textit{DaVI}-BEST identifies and exploits informative dimensions.

In summary, the experiments show that the \textit{DaVI}-BEST algorithm is able to exploit additional dimensions to improve the predictive ability of top-\textit{N} recommender systems. Moreover, the good results obtained with the \textit{Listener} and \textit{Playlist} data sets indicate that the algorithm can be used to improve the music recommendation in the Palco Principal

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>N</th>
<th>\textit{Listener}</th>
<th>\textit{Playlist}</th>
<th>\textit{Entree}</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Precision</td>
<td>Recall</td>
<td>F1</td>
</tr>
<tr>
<td>\textit{DaVI}-BEST</td>
<td>1</td>
<td>0.175</td>
<td>0.175</td>
<td>0.175</td>
</tr>
<tr>
<td>\textit{DaVI}-BEST</td>
<td>2</td>
<td>0.207</td>
<td>0.207</td>
<td>0.207</td>
</tr>
<tr>
<td>\textit{DaVI}-BEST</td>
<td>3</td>
<td>0.136</td>
<td>0.271</td>
<td>0.181</td>
</tr>
<tr>
<td>\textit{DaVI}-BEST</td>
<td>4</td>
<td>0.081</td>
<td>0.244</td>
<td>0.122</td>
</tr>
<tr>
<td>\textit{DaVI}-BEST</td>
<td>5</td>
<td>0.102</td>
<td>0.307</td>
<td>0.153</td>
</tr>
<tr>
<td>\textit{DaVI}-BEST</td>
<td>6</td>
<td>0.053</td>
<td>0.265</td>
<td>0.088</td>
</tr>
<tr>
<td>\textit{DaVI}-BEST</td>
<td>7</td>
<td>0.069</td>
<td>0.344</td>
<td>0.115</td>
</tr>
<tr>
<td>\textit{DaVI}-BEST</td>
<td>8</td>
<td>0.028</td>
<td>0.283</td>
<td>0.051</td>
</tr>
<tr>
<td>\textit{DaVI}-BEST</td>
<td>9</td>
<td>0.038</td>
<td>0.38</td>
<td>0.069</td>
</tr>
</tbody>
</table>
web site. The answer to our first questions is therefore positive: the DaVI-BEST algorithm is able to take advantage of the useful information in multidimensional data to achieve better predictive ability than a two-dimensional recommender algorithm.

5.3.2. Evaluating the DaVI-FS and DaVI-ALL algorithms against the DaVI-BEST algorithm

The second question, Does the use of more than one additional dimension (DaVI-FS and DaVI-ALL algorithms) provide better predictive ability than using the single best dimension (DaVI-BEST algorithm)?, is answered in this section. To do that, we have implemented Algorithms 1–3 for DaVI-BEST, DaVI-FS and DaVI-ALL algorithms, respectively, in order to build the multidimensional recommender models. We have also implemented Algorithm 4 to generate the recommendations for the three DaVI algorithms. In Algorithms 1 and 2, the procedures which select the best additional dimension and the best combination of dimensions, respectively, are implemented using internal 5-fold cross validation.

Using the CF technique, both the DaVI-FS and DaVI-ALL algorithms are usually not significantly different from the DaVI-BEST algorithm. This can be observed in Table 8.

For the DaVI-FS algorithm, we have confirmed that it is significantly better than the DaVI-BEST in only 1 out of 10 comparisons for Precision, Recall and F1 metric. Moreover, the DaVI-FS algorithm timed-out in all comparisons carried out in the Listener data set (symbol “.”” in Table 8). Additionally, we have also analyzed the number of dimensions selected by DaVI-FS algorithm to build a final model. In the Playlist data set, the algorithm has selected combinations of two dimensions in 69% of the experiments. The usage of only one dimension is the most frequent situation in the Entree data set and it occurs in 32% of the experiments.

With respect to the DaVI-ALL algorithm, we have confirmed that it is significantly better than the DaVI-BEST in 5 out of 15 comparisons. In 1 out of 15, it is significantly worse than the DaVI-BEST, and in other nine comparisons, both algorithms obtain equivalent performance. Here, the DaVI-ALL algorithm uses all the regular and virtual items (dimension values) to

### Table 8
Comparing the DaVI-FS and DaVI-ALL algorithms against the DaVI-BEST algorithm using the CF technique in the Listener, Playlist and Entree data sets. Values in boldface are statistically significant.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>N</th>
<th>Listener</th>
<th></th>
<th>Playlist</th>
<th></th>
<th>Entree</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Precision</td>
<td>Recall</td>
<td>F1</td>
<td></td>
<td>Precision</td>
<td>Recall</td>
</tr>
<tr>
<td>DaVI-BEST</td>
<td>1</td>
<td>0.309</td>
<td>0.309</td>
<td>0.309</td>
<td></td>
<td>0.429</td>
<td>0.429</td>
</tr>
<tr>
<td>DaVI-FS</td>
<td>1</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td></td>
<td>0.426</td>
<td>0.426</td>
</tr>
<tr>
<td>DaVI-ALL</td>
<td>1</td>
<td>0.307</td>
<td>0.307</td>
<td>0.307</td>
<td></td>
<td>0.426</td>
<td>0.426</td>
</tr>
<tr>
<td>DaVI-BEST</td>
<td>2</td>
<td>0.203</td>
<td>0.405</td>
<td>0.27</td>
<td></td>
<td>0.253</td>
<td>0.506</td>
</tr>
<tr>
<td>DaVI-FS</td>
<td>2</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td></td>
<td>0.251</td>
<td>0.502</td>
</tr>
<tr>
<td>DaVI-ALL</td>
<td>2</td>
<td>0.203</td>
<td>0.405</td>
<td>0.27</td>
<td></td>
<td>0.251</td>
<td>0.501</td>
</tr>
<tr>
<td>DaVI-BEST</td>
<td>3</td>
<td>0.154</td>
<td>0.463</td>
<td>0.231</td>
<td></td>
<td>0.181</td>
<td>0.542</td>
</tr>
<tr>
<td>DaVI-FS</td>
<td>3</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td></td>
<td>0.18</td>
<td>0.539</td>
</tr>
<tr>
<td>DaVI-ALL</td>
<td>3</td>
<td>0.155</td>
<td>0.464</td>
<td>0.232</td>
<td></td>
<td>0.18</td>
<td>0.54</td>
</tr>
<tr>
<td>DaVI-BEST</td>
<td>5</td>
<td>0.104</td>
<td>0.519</td>
<td>0.173</td>
<td></td>
<td>0.116</td>
<td>0.579</td>
</tr>
<tr>
<td>DaVI-FS</td>
<td>5</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td></td>
<td>0.116</td>
<td>0.578</td>
</tr>
<tr>
<td>DaVI-ALL</td>
<td>5</td>
<td>0.105</td>
<td>0.522</td>
<td>0.174</td>
<td></td>
<td>0.116</td>
<td>0.578</td>
</tr>
<tr>
<td>DaVI-BEST</td>
<td>10</td>
<td>0.057</td>
<td>0.567</td>
<td>0.103</td>
<td></td>
<td>0.061</td>
<td>0.614</td>
</tr>
<tr>
<td>DaVI-FS</td>
<td>10</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td></td>
<td>0.062</td>
<td>0.615</td>
</tr>
<tr>
<td>DaVI-ALL</td>
<td>10</td>
<td>0.058</td>
<td>0.574</td>
<td>0.104</td>
<td></td>
<td>0.061</td>
<td>0.614</td>
</tr>
</tbody>
</table>

### Table 9
Comparing the DaVI-FS and DaVI-ALL algorithms against the DaVI-BEST algorithm using the AR technique in the Listener, Playlist and Entree data sets. Values in boldface are statistically significant.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>N</th>
<th>Listener</th>
<th></th>
<th>Playlist</th>
<th></th>
<th>Entree</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Precision</td>
<td>Recall</td>
<td>F1</td>
<td></td>
<td>Precision</td>
<td>Recall</td>
</tr>
<tr>
<td>DaVI-BEST</td>
<td>1</td>
<td>0.207</td>
<td>0.207</td>
<td>0.207</td>
<td></td>
<td>0.255</td>
<td>0.255</td>
</tr>
<tr>
<td>DaVI-FS</td>
<td>1</td>
<td>0.208</td>
<td>0.208</td>
<td>0.208</td>
<td></td>
<td>0.255</td>
<td>0.255</td>
</tr>
<tr>
<td>DaVI-ALL</td>
<td>1</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td></td>
<td>0.255</td>
<td>0.255</td>
</tr>
<tr>
<td>DaVI-BEST</td>
<td>2</td>
<td>0.136</td>
<td>0.271</td>
<td>0.181</td>
<td></td>
<td>0.151</td>
<td>0.301</td>
</tr>
<tr>
<td>DaVI-FS</td>
<td>2</td>
<td>0.136</td>
<td>0.272</td>
<td>0.181</td>
<td></td>
<td>0.151</td>
<td>0.301</td>
</tr>
<tr>
<td>DaVI-ALL</td>
<td>2</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td></td>
<td>0.15</td>
<td>0.3</td>
</tr>
<tr>
<td>DaVI-BEST</td>
<td>3</td>
<td>0.102</td>
<td>0.307</td>
<td>0.153</td>
<td></td>
<td>0.107</td>
<td>0.32</td>
</tr>
<tr>
<td>DaVI-FS</td>
<td>3</td>
<td>0.104</td>
<td>0.311</td>
<td>0.155</td>
<td></td>
<td>0.107</td>
<td>0.322</td>
</tr>
<tr>
<td>DaVI-ALL</td>
<td>3</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td></td>
<td>0.107</td>
<td>0.322</td>
</tr>
<tr>
<td>DaVI-BEST</td>
<td>5</td>
<td>0.069</td>
<td>0.344</td>
<td>0.115</td>
<td></td>
<td>0.07</td>
<td>0.352</td>
</tr>
<tr>
<td>DaVI-FS</td>
<td>5</td>
<td>0.07</td>
<td>0.349</td>
<td>0.116</td>
<td></td>
<td>0.071</td>
<td>0.354</td>
</tr>
<tr>
<td>DaVI-ALL</td>
<td>5</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td></td>
<td>0.071</td>
<td>0.355</td>
</tr>
<tr>
<td>DaVI-BEST</td>
<td>10</td>
<td>0.038</td>
<td>0.38</td>
<td>0.069</td>
<td></td>
<td>0.039</td>
<td>0.388</td>
</tr>
<tr>
<td>DaVI-FS</td>
<td>10</td>
<td>0.039</td>
<td>0.388</td>
<td>0.071</td>
<td></td>
<td>0.04</td>
<td>0.392</td>
</tr>
<tr>
<td>DaVI-ALL</td>
<td>10</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td></td>
<td>0.04</td>
<td>0.392</td>
</tr>
</tbody>
</table>
build a multidimensional model. However, the CF technique generates recommendations based on the k most similar items to a candidate recommendation. Therefore, although the model contains all the virtual items (dimension values), only a few of them are used to generate the recommendations.

When the base recommender is AR, the DaVI-FS and DaVI-ALL algorithms are usually also not significantly different from the DaVI-BEST algorithm (see Table 9).

In Table 9, the DaVI-FS algorithm is significantly better than the DaVI-BEST algorithm in 3 out of 15 comparisons for Precision, Recall and F1 metric. In the other twelve comparisons, the algorithms are not significantly different. With respect to the number of dimensions selected by DaVI-FS algorithm, in the Listener data set the algorithm has selected combinations of two dimensions in 80% of the experiments. In the Playlist data set, the combinations of two dimensions are used in 58% of the experiments. Finally, in the Entree, the usage of only one dimension to build the final model occurs in 40% of the experiments, and combinations of two dimensions in others 40%.

Regarding the DaVI-ALL algorithm, it is better in 1 out of 10 comparisons, worse in 2 out of 10, and equivalent to DaVI-BEST in the other seven comparisons. In all comparisons performed in the Listener data set, the DaVI-ALL algorithm timed-out (symbol “-” in Table 9). Here, the AR technique, which is used as base recommender by the DaVI-ALL algorithm, selects a few regular and virtual items as frequent in order to generate the rules and, consequently, the recommendations. Therefore, although the algorithm can use all the virtual items (dimension values), only a few of them are selected to take part of the multidimensional model.

Based on the results presented here, the answer for our second question is: No, the use of more than one additional dimension (DaVI-FS and DaVI-ALL algorithms) does not provide better predictive ability than using the single best dimension (DaVI-BEST algorithm). We can see by the experiments that both DaVI-FS and DaVI-ALL, with CF and AR techniques as base recommenders, do not provide more accurate recommendations than the DaVI-BEST algorithm. In most of the cases presented in Tables 8 and 9, DaVI-FS and DaVI-ALL are not significantly different than DaVI-BEST. Additionally, we also observe that DaVI-FS using the CF technique and DaVI-ALL using the AR technique timed-out in the Listener data set. This happened because this data set has a large number of dimensions, which make it difficult for both algorithms to process the large data set generated. Finally, we also verified that although the DaVI-ALL algorithm can use all the dimensions in a data set to build a multidimensional model and generate recommendations, it only uses a few of them for this purpose. All these facts make us believe that the DaVI-BEST algorithm is a better option to build a multidimensional model than the DaVI-FS and DaVI-ALL algorithms.

5.3.3. Evaluating the DaVI-BEST algorithm against other algorithms proposed in the literature

In this section, we answer our last research question, Does the DaVI-BEST algorithm present better predictive ability than other multidimensional algorithms proposed in the literature? To answer this question, we have compared the DaVI-BEST against two other algorithms proposed in the literature.

5.3.3.1. DaVI-BEST versus combined reduction-based algorithm

The first algorithm is the combined reduction-based (Adomavicius et al., 2005). To the best of our knowledge, it is considered the first algorithm for multidimensional recommender systems and has been used in the literature as a baseline to evaluate multidimensional recommendation algorithms (Baltrunas & Ricci, 2009, 2010; Lu et al., 2008; Panniello et al., 2009). To carry out the evaluation, we have implemented Algorithm 1 for the DaVI-BEST algorithm, and the algorithm presented in Adomavicius et al. (2005) for the combined reduction-based algorithm. We have combined both algorithms with the CF and AR recommendation techniques, and tested them on the three data sets used so far. For the DaVI-BEST algorithm (Algorithm 1), the procedures that select the best additional dimension to build the final multidimensional model were implemented using internal 5-fold cross validation. In order to have a fair evaluation, in the combined reduction-based algorithm (Adomavicius et al., 2005), the procedures that select the best segments were also implemented using internal 5-fold cross validation.

With the CF technique, the DaVI-BEST algorithm presents better results (i.e., Precision, Recall and F1 metric) than the combined reduction-based algorithm in the Listener and Playlist data sets. We can observe this fact in Table 10.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>N</th>
<th>Listener</th>
<th></th>
<th></th>
<th>Playlist</th>
<th></th>
<th></th>
<th>Entree</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Precision</td>
<td>Recall</td>
<td>F1</td>
<td>Precision</td>
<td>Recall</td>
<td>F1</td>
<td>Precision</td>
<td>Recall</td>
<td>F1</td>
</tr>
<tr>
<td>C. Reduction</td>
<td>1</td>
<td>0.231</td>
<td>0.231</td>
<td>0.231</td>
<td>0.342</td>
<td>0.342</td>
<td>0.342</td>
<td>0.218</td>
<td>0.218</td>
<td>0.218</td>
</tr>
<tr>
<td>DaVI-BEST</td>
<td>1</td>
<td>0.309</td>
<td>0.309</td>
<td>0.309</td>
<td>0.429</td>
<td>0.429</td>
<td>0.429</td>
<td>0.22</td>
<td>0.22</td>
<td>0.22</td>
</tr>
<tr>
<td>C. Reduction</td>
<td>2</td>
<td>0.169</td>
<td>0.338</td>
<td>0.226</td>
<td>0.219</td>
<td>0.439</td>
<td>0.293</td>
<td>0.17</td>
<td>0.34</td>
<td>0.227</td>
</tr>
<tr>
<td>DaVI-BEST</td>
<td>2</td>
<td>0.203</td>
<td>0.405</td>
<td>0.27</td>
<td>0.253</td>
<td>0.506</td>
<td>0.337</td>
<td>0.169</td>
<td>0.339</td>
<td>0.226</td>
</tr>
<tr>
<td>C. Reduction</td>
<td>3</td>
<td>0.132</td>
<td>0.396</td>
<td>0.198</td>
<td>0.161</td>
<td>0.484</td>
<td>0.242</td>
<td>0.142</td>
<td>0.426</td>
<td>0.213</td>
</tr>
<tr>
<td>DaVI-BEST</td>
<td>3</td>
<td>0.154</td>
<td>0.463</td>
<td>0.231</td>
<td>0.181</td>
<td>0.542</td>
<td>0.271</td>
<td>0.141</td>
<td>0.422</td>
<td>0.211</td>
</tr>
<tr>
<td>C. Reduction</td>
<td>5</td>
<td>0.051</td>
<td>0.456</td>
<td>0.152</td>
<td>0.107</td>
<td>0.334</td>
<td>0.178</td>
<td>0.106</td>
<td>0.528</td>
<td>0.176</td>
</tr>
<tr>
<td>DaVI-BEST</td>
<td>5</td>
<td>0.104</td>
<td>0.519</td>
<td>0.173</td>
<td>0.116</td>
<td>0.579</td>
<td>0.193</td>
<td>0.105</td>
<td>0.527</td>
<td>0.176</td>
</tr>
<tr>
<td>C. Reduction</td>
<td>10</td>
<td>0.051</td>
<td>0.509</td>
<td>0.092</td>
<td>0.057</td>
<td>0.572</td>
<td>0.104</td>
<td>0.052</td>
<td>0.629</td>
<td>0.114</td>
</tr>
<tr>
<td>DaVI-BEST</td>
<td>10</td>
<td>0.057</td>
<td>0.567</td>
<td>0.103</td>
<td>0.061</td>
<td>0.614</td>
<td>0.112</td>
<td>0.063</td>
<td>0.634</td>
<td>0.115</td>
</tr>
</tbody>
</table>
In the Listener data set, the values presented in Table 10 represent Precision gains ranging from 11.7% to 33.7%, Recall gains from 11.3% to 33.7%, and F1 gains from 11.9% to 33.7%. In the same table, for the Playlist data set, we have gains of Precision ranging from 7% to 25.4%, Recall from 7.3% to 25.4%, and F1 from 7.6% to 25.4%. With respect to the Entree data set, we can see in Table 10 that the algorithms present Precision, Recall and F1 metric which are generally not significantly different. This is true for N taking values 1, 2, 3 and 5. For N = 10, the DaVI-BEST algorithm presents values which are better than the combined reduction-based algorithm. The values represent small gains of 1.6%, 0.8% and 0.87% for Precision, Recall and F1 metric, respectively, but they are all statistically significant.

An interesting fact is that the results in Listener and Playlist are equal in Tables 6 and 10. This fact occurs because the models built with the segments do not outperform the pure two-dimensional model, and, therefore, the combined reduction-based algorithm generates its recommendations based on the pure two-dimensional recommender model (user × item). This means that combined reduction-based algorithm is not able to use the information in the dimensions to improve the recommendations of CF on these data sets, while DaVI-BEST is. In the Entree data set, the segments browser (from the dimension intention) and week day (from the dimension work_day) outperform the two-dimensional recommender model and generate more accurate recommendations.

With respect to the AR models, we see in Table 11 that the DaVI-BEST algorithm is significantly better than the combined reduction-based algorithm in the Listener and Entree data sets. In the Playlist data set, the algorithms are equivalent.

In Table 11, the values in the Listener data set provide gains of Precision ranging from 5% to 11.7%, Recall from 5% to 10.7%, and F1 from 5% to 11.2%. In the Entree data set, we have gains in Precision ranging from 1.3% to 7.4%, in Recall from 0.6% to 7.4%, and in F1 from 0.7% to 7.4%. Regarding the Playlist data set, the DaVI-BEST algorithm presents worse results than the combined reduction-based algorithm. However, according to the paired t-test, these losses are not statistically significant.

An interesting fact regarding the Listener and Playlist data sets is that the dimension band, which is widely selected by the DaVI-BEST algorithm, is not used by the combined reduction-based recommender algorithm. In the Listener data set, the combined reduction-based recommender algorithm usually selects segments from the dimensions work_day, work_hour, location, music_genre and instrumental. In the Playlist data set, it selects segments from the dimensions music_genre and instrumental. Thus, we observe that each algorithm uses a different set of dimensions to improve the accuracy of its recommendations.

Thus, from our experiments, we can assume that the DaVI-BEST presents better predictive ability than the combined reduction-based algorithm. The DaVI-BEST has better performance in 4 out of 6 experiments (i.e., data set × base recommender). In the other two, the performance is equivalent.

5.3.3.2. DaVI-BEST versus weight post-filtering algorithm. The second algorithm with which we have compared to the DaVI-BEST is the weight post-filtering (PoF) (Panniello et al., 2009). Again, we have combined both algorithms with the CF and AR recommendation techniques, and tested them on the three data sets used so far.

When the base recommender is CF, the DaVI-BEST is significantly better than the weight post-filtering algorithm (Weight PoF) in the Listener and Playlist data sets. This can be observed in Table 12. For these two data sets, the DaVI-BEST algorithm presents F1 gains ranging from 10.7% to 32.1% (Listener), and from 6.6% to 8.7% (Playlist). In the Entree data set, we have a F1 gain of 0.9% for N = 10. For N equal to 2 and 3, we have losses of 2.2% and 1.4%, respectively. Finally, for N equal to 1 and 5, DaVI-BEST has statistically non-significant losses.

Regarding the AR models, we see in Table 13 that the DaVI-BEST algorithm is significantly better than the weight post-filtering algorithm in all data sets. In Table 13, the values for the Listener data set provide gains of Precision ranging from 17.6% to 35.7%, Recall from 17.6% to 33.3%, and F1 from 17.6% to 32.7%. For the Playlist data set, we have gains in Precision, Recall and F1 between 9% and 13%. Finally, in the Entree data set, we have gains between 12% and 20% for Precision, Recall and F1 metric.

These results indicate that the DaVI-BEST algorithm also presents better predictive ability than the weight post-filtering algorithm. On these data sets, it has better performance in 5 out of 6 experiments (i.e., data set × base recommender).

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>N</th>
<th>Listener Precision</th>
<th>Recall</th>
<th>F1</th>
<th>Playlist Precision</th>
<th>Recall</th>
<th>F1</th>
<th>Entree Precision</th>
<th>Recall</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Precision</td>
<td></td>
<td></td>
<td>Recall</td>
<td></td>
<td></td>
<td>F1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C. Reduction</td>
<td>1</td>
<td>0.197</td>
<td>0.197</td>
<td>0.197</td>
<td>0.262</td>
<td>0.262</td>
<td>0.262</td>
<td>0.324</td>
<td>0.324</td>
<td>0.324</td>
</tr>
<tr>
<td>DaVI-BEST</td>
<td>1</td>
<td>0.207</td>
<td>0.207</td>
<td>0.207</td>
<td>0.255</td>
<td>0.255</td>
<td>0.255</td>
<td>0.348</td>
<td>0.348</td>
<td>0.348</td>
</tr>
<tr>
<td>C. Reduction</td>
<td>2</td>
<td>0.127</td>
<td>0.254</td>
<td>0.17</td>
<td>0.153</td>
<td>0.305</td>
<td>0.203</td>
<td>0.226</td>
<td>0.452</td>
<td>0.301</td>
</tr>
<tr>
<td>DaVI-BEST</td>
<td>2</td>
<td>0.136</td>
<td>0.271</td>
<td>0.181</td>
<td>0.151</td>
<td>0.301</td>
<td>0.201</td>
<td>0.239</td>
<td>0.478</td>
<td>0.319</td>
</tr>
<tr>
<td>C. Reduction</td>
<td>3</td>
<td>0.095</td>
<td>0.285</td>
<td>0.143</td>
<td>0.109</td>
<td>0.327</td>
<td>0.164</td>
<td>0.179</td>
<td>0.536</td>
<td>0.268</td>
</tr>
<tr>
<td>DaVI-BEST</td>
<td>3</td>
<td>0.102</td>
<td>0.307</td>
<td>0.153</td>
<td>0.107</td>
<td>0.32</td>
<td>0.16</td>
<td>0.186</td>
<td>0.557</td>
<td>0.279</td>
</tr>
<tr>
<td>C. Reduction</td>
<td>5</td>
<td>0.063</td>
<td>0.317</td>
<td>0.106</td>
<td>0.071</td>
<td>0.356</td>
<td>0.119</td>
<td>0.13</td>
<td>0.65</td>
<td>0.217</td>
</tr>
<tr>
<td>DaVI-BEST</td>
<td>5</td>
<td>0.069</td>
<td>0.344</td>
<td>0.115</td>
<td>0.07</td>
<td>0.352</td>
<td>0.117</td>
<td>0.132</td>
<td>0.661</td>
<td>0.22</td>
</tr>
<tr>
<td>C. Reduction</td>
<td>10</td>
<td>0.034</td>
<td>0.343</td>
<td>0.062</td>
<td>0.039</td>
<td>0.388</td>
<td>0.071</td>
<td>0.078</td>
<td>0.784</td>
<td>0.142</td>
</tr>
<tr>
<td>DaVI-BEST</td>
<td>10</td>
<td>0.038</td>
<td>0.38</td>
<td>0.069</td>
<td>0.039</td>
<td>0.388</td>
<td>0.07</td>
<td>0.079</td>
<td>0.789</td>
<td>0.143</td>
</tr>
</tbody>
</table>
In conclusion, the answer to our last research question is: Yes, the DaVI-BEST algorithm presents better predictive ability than other multidimensional algorithms proposed in the literature. We have shown that it has better performance than the combined reduction-based algorithm in 4 out of 6 experiments, and better performance than weight post-filtering algorithm in 5 out of 6 experiments.

5.3.4. Scalability of the DaVI Based Algorithms

In this section, we analyze the scalability of the DaVI-BEST, DaVI-FS and DaVI-ALL algorithms with respect to the number of dimensions available in a data set for building a multidimensional model. We analyze the time that the algorithms take to build a model when the data set contains only one dimension, two dimensions, three dimensions and so forth. To do that, we have first determined a random sequence for the dimensions in our three data sets. Then, we have measured the time required by each algorithm to build its multidimensional model considering the first dimension in the sequence, the first two dimensions, the first three dimensions and so forth. To do that, we have summed the time taken to build all internal models and also the final one. The same scale parameters are used for all data sets.

For the sake of simplicity, we have only computed the time spent to build the multidimensional models (i.e., internal models for evaluation of the dimensions and the final model for recommendation), excluding the time for any other task. For the DaVI-BEST algorithm, we have summed the time taken to build all internal models and also the final one. The same is done for the DaVI-FS. Here, we ran all iterations of this algorithm (without applying the stopping criterion defined in Section 4.2). For both the DaVI-BEST and DaVI-FS algorithms, we have also simplified the execution by running them using an internal All But One protocol without cross validation, and randomly selecting the best dimension. Finally, for the DaVI-ALL, we have summed the time spent building the multidimensional model with all dimensions together. The results presented in Figs. 5 and 6 were obtained using an Intel Core i7 920 PC with a CPU clock rate of 2.66 GHZ, 12 GB of main memory, and running the Ubuntu Linux operating system.

In Fig. 5, we observe that DaVI-BEST scales linearly with the number of dimensions, DaVI-FS scales exponentially with the number of dimensions, and DaVI-ALL remains roughly constant with a small increase in time as the number of dimensions grows.

With respect to the Listener data set, the first dimension chosen randomly is week_day. This dimension will add seven new rows and columns to the similarity matrix, independently of the DaVI algorithm. On the other hand, when we have all the ten dimensions, the DaVI-BEST algorithm can add up to 2296 new rows and columns to the matrix by using the dimension band, and the DaVI-FS and DaVI-ALL will add 2471 new rows and columns to the matrix (i.e., the sum of the number of values avail-

### Table 12
Comparing the DaVI-BEST algorithm against the Weight Post-Filtering algorithm using the CF technique in the Listener, Playlist and Entree data sets. Values in boldface are statistically significant.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>N</th>
<th>Listener</th>
<th></th>
<th></th>
<th></th>
<th>Playlist</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th>Entree</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Precision</td>
<td>Recall</td>
<td>F1</td>
<td>Precision</td>
<td>Recall</td>
<td>F1</td>
<td>Precision</td>
<td>Recall</td>
<td>F1</td>
<td>Precision</td>
<td>Recall</td>
<td>F1</td>
<td>Precision</td>
<td>Recall</td>
<td>F1</td>
</tr>
<tr>
<td>Weight PoF</td>
<td>1</td>
<td>0.234</td>
<td>0.234</td>
<td>0.234</td>
<td>0.398</td>
<td>0.398</td>
<td>0.398</td>
<td>0.221</td>
<td>0.221</td>
<td>0.221</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DaVI-BEST</td>
<td>1</td>
<td>0.309</td>
<td>0.309</td>
<td>0.309</td>
<td>0.429</td>
<td>0.429</td>
<td>0.429</td>
<td>0.22</td>
<td>0.22</td>
<td>0.22</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Weight PoF</td>
<td>2</td>
<td>0.17</td>
<td>0.34</td>
<td>0.227</td>
<td>0.233</td>
<td>0.466</td>
<td>0.31</td>
<td>0.173</td>
<td>0.346</td>
<td>0.231</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DaVI-BEST</td>
<td>2</td>
<td>0.203</td>
<td>0.405</td>
<td>0.27</td>
<td>0.253</td>
<td>0.506</td>
<td>0.337</td>
<td>0.169</td>
<td>0.339</td>
<td>0.226</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Weight PoF</td>
<td>3</td>
<td>0.132</td>
<td>0.397</td>
<td>0.199</td>
<td>0.168</td>
<td>0.504</td>
<td>0.252</td>
<td>0.142</td>
<td>0.427</td>
<td>0.214</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DaVI-BEST</td>
<td>3</td>
<td>0.154</td>
<td>0.463</td>
<td>0.231</td>
<td>0.181</td>
<td>0.542</td>
<td>0.271</td>
<td>0.141</td>
<td>0.422</td>
<td>0.211</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Weight PoF</td>
<td>5</td>
<td>0.091</td>
<td>0.457</td>
<td>0.152</td>
<td>0.108</td>
<td>0.541</td>
<td>0.18</td>
<td>0.106</td>
<td>0.528</td>
<td>0.176</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DaVI-BEST</td>
<td>5</td>
<td>0.104</td>
<td>0.519</td>
<td>0.173</td>
<td>0.116</td>
<td>0.579</td>
<td>0.193</td>
<td>0.105</td>
<td>0.527</td>
<td>0.176</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Weight PoF</td>
<td>10</td>
<td>0.051</td>
<td>0.509</td>
<td>0.093</td>
<td>0.058</td>
<td>0.378</td>
<td>0.105</td>
<td>0.063</td>
<td>0.629</td>
<td>0.114</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DaVI-BEST</td>
<td>10</td>
<td>0.057</td>
<td>0.567</td>
<td>0.103</td>
<td>0.061</td>
<td>0.614</td>
<td>0.112</td>
<td>0.063</td>
<td>0.634</td>
<td>0.115</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table 13
Comparing the DaVI-BEST algorithm against the Weight Post-Filtering algorithm using the AR technique in the Listener, Playlist and Entree data sets. Values in boldface are statistically significant.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>N</th>
<th>Listener</th>
<th></th>
<th></th>
<th></th>
<th>Playlist</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th>Entree</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Precision</td>
<td>Recall</td>
<td>F1</td>
<td>Precision</td>
<td>Recall</td>
<td>F1</td>
<td>Precision</td>
<td>Recall</td>
<td>F1</td>
<td>Precision</td>
<td>Recall</td>
<td>F1</td>
<td>Precision</td>
<td>Recall</td>
<td>F1</td>
</tr>
<tr>
<td>Weight PoF</td>
<td>1</td>
<td>0.176</td>
<td>0.176</td>
<td>0.176</td>
<td>0.23</td>
<td>0.23</td>
<td>0.23</td>
<td>0.29</td>
<td>0.29</td>
<td>0.29</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DaVI-BEST</td>
<td>1</td>
<td>0.207</td>
<td>0.207</td>
<td>0.207</td>
<td>0.255</td>
<td>0.255</td>
<td>0.255</td>
<td>0.348</td>
<td>0.348</td>
<td>0.348</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Weight PoF</td>
<td>2</td>
<td>0.112</td>
<td>0.223</td>
<td>0.149</td>
<td>0.135</td>
<td>0.27</td>
<td>0.18</td>
<td>0.204</td>
<td>0.407</td>
<td>0.271</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DaVI-BEST</td>
<td>2</td>
<td>0.136</td>
<td>0.271</td>
<td>0.181</td>
<td>0.151</td>
<td>0.301</td>
<td>0.201</td>
<td>0.239</td>
<td>0.478</td>
<td>0.319</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Weight PoF</td>
<td>3</td>
<td>0.081</td>
<td>0.244</td>
<td>0.122</td>
<td>0.098</td>
<td>0.293</td>
<td>0.146</td>
<td>0.16</td>
<td>0.48</td>
<td>0.24</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DaVI-BEST</td>
<td>3</td>
<td>0.102</td>
<td>0.307</td>
<td>0.153</td>
<td>0.107</td>
<td>0.32</td>
<td>0.16</td>
<td>0.186</td>
<td>0.537</td>
<td>0.279</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Weight PoF</td>
<td>5</td>
<td>0.053</td>
<td>0.266</td>
<td>0.089</td>
<td>0.064</td>
<td>0.318</td>
<td>0.106</td>
<td>0.116</td>
<td>0.579</td>
<td>0.193</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DaVI-BEST</td>
<td>5</td>
<td>0.069</td>
<td>0.344</td>
<td>0.115</td>
<td>0.07</td>
<td>0.352</td>
<td>0.117</td>
<td>0.132</td>
<td>0.661</td>
<td>0.22</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Weight PoF</td>
<td>10</td>
<td>0.028</td>
<td>0.285</td>
<td>0.052</td>
<td>0.035</td>
<td>0.345</td>
<td>0.063</td>
<td>0.07</td>
<td>0.7</td>
<td>0.127</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DaVI-BEST</td>
<td>10</td>
<td>0.038</td>
<td>0.38</td>
<td>0.069</td>
<td>0.039</td>
<td>0.388</td>
<td>0.07</td>
<td>0.079</td>
<td>0.789</td>
<td>0.143</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
able for all dimensions). For the Playlist data set, when the number of dimensions is equal to 1, we have the dimension music_genre that adds 32 rows and columns to the similarity matrix. With a number of dimensions equal to 3, DaVI-BEST can add up to 1862 rows and columns to the matrix by using the dimension band, and the other two algorithms will add 1896 rows and columns to the matrix (i.e., the number of values available for the three dimensions, band, music_genre and instrumental). Finally, for the Entree, the first dimension chosen randomly is hour, which will add 24 rows and columns to the similarity matrix. When we have all the seven dimensions, DaVI-BEST can use the dimension day and add up to 31 rows and columns to the matrix. The DaVI-FS and DaVI-ALL will add 85 rows and columns to the similarity matrix (i.e., the number of values available for all the seven dimensions).

Fig. 5. Scalability of the DaVI-BEST, DaVI-FS and DaVI-ALL algorithms, using the CF technique, with respect to the number of dimensions.
In Fig. 6, where the AR technique is used as base recommender, we also observe that DaVI-BEST scales linearly, DaVI-FS scales exponentially and DaVI-ALL remains roughly constant. The AR models are built using the same minimum support and confidence values defined for the previous experiments. In Fig. 6, we also see that the DaVI-FS and DaVI-ALL algorithms timed-out in the Listener data set with nine and ten dimensions. Finally, it is interesting to note that the curves have similar shapes in both cases (i.e., CF and AR).

6. Conclusion and future work

In this paper we proposed a multidimensional recommendation approach, called DaVI (Dimensions as Virtual Items). It consists in using the values of the additional dimensions (e.g., contextual or background information) as (virtual) items to
enable the application of existing two-dimensional recommender algorithms for the generation of recommendations using the additional dimensions. The main advantage of this approach is that it can be applied on different off-the-shelf two-dimensional recommender algorithms.

We have implemented three variants of the DaVI approach. The first one, the DaVI-BEST algorithm, automatically selects the best additional dimension (from a set of dimensions), transforms the original data set with this extra information and builds a multidimensional model to generate the recommendations. The second algorithm, called DaVI-FS, combines the DaVI-BEST with a sequential forward selection algorithm in order to select the best combination of dimensions to build the multidimensional model. Finally, the DaVI-ALL algorithm applies the DaVI approach on all existing dimensions in a data set at the same time.

The results of our empirical evaluation showed that the DaVI-BEST algorithm improves the predictive ability of top-N recommender systems and is able to identify and exploit informative dimensions for recommendations. It is followed by the DaVI-ALL and DaVI-FS algorithms, which obtain similar performance to the DaVI-BEST algorithm but requiring more time and/or memory. The DaVI-BEST has also presented better performance than two multidimensional algorithms proposed in the literature, the combined reduction-based and the weight post-filtering. An empirical analysis of the scalability of our algorithms, relative to the number of dimensions, indicates that DaVI-ALL is nearly constant, the execution time of DaVI-BEST increases linearly and the increase for DaVI-FS is exponential.

There are several directions to be explored in the future. The DaVI approach can be tried on other recommender algorithms, such as Markov Models (Deshpande & Karypis, 2004b) and SVD approaches (Brand, 2003). There are other multidimensional algorithms which could be compared with our approach. They have been presented in Section 2, but implementing some of them will take some time. Therefore, we have chosen as reference two state-of-the-art algorithms. It would also be important to further challenge the DaVI approach with new data sets, in order to validate the conclusions of this paper. However, there is a lack of large scale annotated data sets for multidimensional recommendation researches (Li et al., 2010; Palmisano et al., 2008; Verbert et al., 2012). Therefore, it would be important to devote some time investigating new ways for building multidimensional data sets.

Acknowledgements

The financial support from Fundação para a Ciência e e Tecnologia (FCT) under the PhD grant SFRH/BD/22516/2005, FCT project Rank! (PTDC/EIA/81178/2006) and QREN-AdI Palco3.0/3121 PONORTE is gratefully acknowledged.

References


