Improving Multidimensional Recommender Systems Using Dimensions as Virtual Items

Marcos Aurélio Domingues[1], Alípio Mário Jorge[2], Carlos Soares[2], Solange Oliveira Rezende[1]

[1]Institute of Mathematics and Computer Science
University of São Paulo
São Carlos, SP, Brazil

{mad, solange}@icmc.usp.br

[2]University of Porto
Porto, Portugal

amjorge@fc.up.pt, csoares@fe.up.pt

Abstract. *The first multidimensional algorithm for recommender systems is the well known combined reduction-based, which treats additional dimensions as labels for segmenting/filtering sessions, using the segmented sessions to build the recommendation model. This algorithm only uses the additional dimensions when it outperforms the traditional two-dimensional algorithm. Otherwise, it reverts to the traditional two-dimensional algorithm to generate the top-N recommendations. In this paper, we propose to improve the combined reduction-based algorithm by using the* ***DaVI*** *approach, which handles additional dimensions as virtual items. Incorporating the* ***DaVI*** *approach into the combined reduction-based, the multidimensional algorithm uses the additional dimensions not only as labels for segmenting sessions but also as virtual items to improve the recommendation model. The empirical results demonstrate that our proposal reduces the needs of reverting to the traditional two-dimensional algorithm to generate the top-N recommendations, increasing the performance of the combined reduction-based algorithm.*

1. Introduction

As the massive information accessible via Internet grows exponentially, users have more difficulties to reach the information they really need. 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 output a set of items/recommendations that are likely to be of interest to the user [Resnick and Varian 1997, Ricci et al. 2011].

Usually a recommender system is divided into a two-stage process [Anand and Mobasher 2003]. The first stage is carried out offline. Data representing the behavior of users of the web site, which were previously collected, are mined and a model is generated for use in future online interactions. The second stage is carried out in real time with a new user interacting with the web site. Data from the current user session are used as input by the model to generate the recommendations.

One of the best illustrations for such a recommender system is the one deployed in the Amazon web site, 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 et al. 2003].

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 data that relate these dimensions, a recommendation model can be built and used to identify a set of N pages that are expected to be of interest to a certain user. However, other dimensions, such as time and type of content (e.g., genre of music the user is listening to a music portal) of the accesses, can be used as additional information. They may capture the context or background information in which recommendations are being made to improve their performance. For example, in a news delivery web site the date/time when the recommendation is made is important because on weekdays a user might prefer to read world news in the morning and stock market reports in the evening, and on weekends, sport news may be more adequate.

To the best of our knowledge, the first multidimensional algorithm for recommender systems is proposed in [Adomavicius and Tuzhilin 2001a, Adomavicius and Tuzhilin 2001b]. This algorithm, called combined reduction-based, uses additional dimensions as labels for segmenting/filtering sessions, using the segmented sessions to build the recommendation model. This algorithm only uses the additional dimensions when it outperforms the traditional two-dimensional algorithm. Otherwise, it reverts to the traditional two-dimensional algorithm to generate the top-N recommendations.

In this paper, we propose to improve the combined reduction-based algorithm by using the **DaVI** approach, which treats additional dimensions as virtual items into the recommendation model [Domingues et al. 2009, Domingues et al. 2013]. We extend the combined reduction-based algorithm to use the **DaVI** approach. Thus, the multidimensional algorithm uses the additional dimensions not only as labels for segmenting sessions but also as virtual items to improve the recommendation model. Our proposal reduces the needs of reverting to the traditional two-dimensional algorithm to generate the top-N recommendations, increasing the performance of the combined reduction-based algorithm. We evaluate our proposal by using two real world data sets and compare it against to the previously introduced combined reduction-based algorithm [Adomavicius and Tuzhilin 2001a, Adomavicius and Tuzhilin 2001b]. Additionally, we also compare our proposal against the traditional two-dimensional recommendation algorithm in order to demonstrate the advantage of using additional information to make recommendations.

The paper is organized as follows. In Section 2, we present multidimensional recommender systems, which incorporate additional dimensions (e.g., time, location, etc) to build the recommendation model. In Section 3, we present our **DaVI** approach, which treats dimensions as virtual items. Our proposal is presented in Section 4. In Section 5, we empirically evaluate our proposal. Finally, we conclude the paper and present some ideas for further developments (Section 6).

2. Multidimensional Recommender Systems

A traditional two-dimensional recommender system for the web is an information filtering technology which can be used to predict preference ratings of items (e.g., events, movies, music, books, news, images, web pages, etc) not currently rated by the user [Breese et al. 1998, Ricci et al. 2011], and/or to output a set of items/recommendations that are likely to be of interest to the user [Resnick and Varian 1997, Ricci et al. 2011].

In this work, we focus on the task of selecting the top-N items/recommendations which are of interest to a user. We formalize this task as follows:

> Let p be the number of users $U = \{u_1, u_2, ..., u_p\}$ and q the number of all possible items that can be recommended $I = \{i_1, i_2, ..., i_q\}$. Now, let j be the number of historical sessions in a web site $S = \{s_1, s_2, ..., s_j\}$. Each session $s = \langle u, I_s \rangle$ is a tuple defined by a user $u \in U$ and a set of accessed items $I_s \subseteq I$. The set S is used to build a top-N recommendation model M.
>
> Given an active session s_a defined by an active user u_a and a set of observable items $O \subset I$, the recommendation model M uses the set O to identify the interest of the user u_a and recommend N items from the set of items/recommendations R, such that $R \subset I$ and $R \cap O = \varnothing$, that are believed to be the top preferences of the user u_a.

As we can see, the most common two-dimensional recommender systems focus on two dimensions, users and items, to make the recommendations. However, other dimensions, such as time and type of content (e.g., type of music that a page concerns in a music portal) of the accesses, can be used as additional information, capturing the context or background information in which recommendations are made in order to improve their performance. For example, the type of book that a user looks for in Amazon[1] during working hours is probably different from the books searched for during leisure hours, or the songs recommended by a music web site (e.g., Last.fm[2]) to a user who likes rock music should be different from the songs recommended to a user who likes pop music.

According to [Adomavicius et al. 2005, Domingues et al. 2013], multidimensional recommender systems extend traditional two-dimensional recommender systems to handle multiple dimensions following the multidimensional data model used by data warehouses and OLAP applications. More formally, given the dimensions $d_1, d_2, ..., d_t$, where each dimension d represents a set of values of attributes (e.g., users, items, days and/or months of the accesses, etc), we can define the recommendation space for these dimensions as a Cartesian product $d_1 \times d_2 \times ... \times d_t$. Moreover, let $\Re$ be a set of recommendations R, where each R is a set of items $i \in I$. Then, we can define the multidimensional recommendation model M' over the space $d_1 \times d_2 \times ... \times d_t$, where $t > 2$, as

$$M' : d_1 \times d_2 \times ... \times d_t \rightarrow \Re. \qquad (1)$$

In a multidimensional model, 2 out of t dimensions always represent users (U) and items (I). Usually, the other additional dimensions represent contextual or background

[1] `http://www.amazon.com`

[2] `http://www.last.fm`

information about a web access/session. Therefore, we can redefine the multidimensional model as

$$M' : U \times I \times ... \times d_t \rightarrow \Re. \quad (2)$$

The first multidimensional algorithm for recommender systems is proposed in [Adomavicius and Tuzhilin 2001a, Adomavicius and Tuzhilin 2001b] and extended in greater depth in [Adomavicius et al. 2005, Adomavicius and Tuzhilin 2011]. This algorithm, called combined reduction-based, uses additional dimensions as labels for segmenting/filtering sessions, using the segmented sessions to build the recommendation model. Here, a segment is defined as a subset of the overall set of sessions which is selected based on the values of attributes of an additional dimension or combinations of these values.

In this paper, we extend the combined reduction-based algorithm by using the **DaVI** approach, which handles additional dimensions as virtual items to build the multidimensional model [Domingues et al. 2009, Domingues et al. 2013]. In Section 3, we present the **DaVI** approach. In Section 4, we extend the combined reduction-based algorithm by using the **DaVI** approach.

3. Dimensions as Virtual Items

In this section, we present the **DaVI** (*Dimensions as Virtual Items*) approach [Domingues et al. 2009, Domingues et al. 2013]. The approach treats additional dimensions as virtual items, using them together with the regular items in a recommender system. Virtual items are used to build/improve the recommendation model but they can not be recommended. 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 $U = \{u_1, u_2, ..., u_p\}$ and q the number of all possible items that can be recommended $I = \{i_1, i_2, ..., i_q\}$. In addition, we also have other dimensions (e.g., contextual or background information), $D = d_1 \cup d_2 \cup ... \cup d_t$, where each dimension d defines a set of values of attributes. 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 $S' = \{s'_1, s'_2, ..., s'_j\}$. Each session s' is a tuple defined by a user $u \in U$, a set of accessed items $I_{s'} \subseteq I$ and a set $D_{s'} \subseteq D$ containing all the dimension values associated with the session s', i.e., $s' = \langle u, I_{s'}, D_{s'} \rangle$.

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 $s' = \langle u, I_{s'}, d_{s'}.v \rangle$ has the dimension value (virtual item) $d_{s'}.v$ associated to the session s'. Here, the dimension value $d_{s'}.v$ 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 $s' = \langle u, I_{s'}, d_{s'} \rangle = \langle u, \{i_1, ..., i_q\}, \{d_{s'}.v_1, ..., d_{s'}.v_q\} \rangle$ has the dimension values (virtual items) $d_{s'}.v_1, ..., d_{s'}.v_q$ associated to respective items $i_1, ..., i_q$ in the session s'. For example, if the dimension values $d_{s'}.v_1, ..., d_{s'}.v_q$ 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 of converting each multidimensional session $s' = \langle u, I_{s'}, D_{s'} \rangle$ into an extended two-dimensional session $s'' = \langle u, I_{s''} \cup D_{s''} \rangle$, where the values of the additional dimensions in $D_{s''}$ are used as virtual items together with the regular items in $I_{s''}$. The **DaVI** approach can also be applied to a subset of dimensions or even to a single dimension. For example, a multidimensional session $s' = \langle u, I_{s'}, d_{s'} \rangle = \langle u, \{i_1, ..., i_q\}, \{d_{s'}.v_1, ..., d_{s'}.v_q\} \rangle$, with a single dimension d, can be converted into an extended two-dimensional session $s'' = \langle u, I_{s''} \cup d_{s'} \rangle = \langle u, \{i_1, ..., i_q, d_{s'}.v_1, ..., d_{s'}.v_q\} \rangle$. Thus, we have defined the **DaVI** approach as a function that converts a set of multidimensional sessions into a set of extended two-dimensional sessions,

$$S'' = \mathbf{DaVI}(S', \widehat{D}), \tag{3}$$

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

4. Our Proposal

The combined reduction-based algorithm was developed to predict ratings [Adomavicius et al. 2005, Adomavicius and Tuzhilin 2011]. Given our focus on top-N recommender systems, we present a version of this algorithm for top-N recommendations. The algorithm consists of two phases, which are presented in Algorithms 1 and 2. First, using historical data, we determine which dimensional segments outperform the traditional recommendation method. Second, to generate the top-N recommendations, we choose the best dimensional segment for a particular active session and apply the two-dimensional recommendation algorithm on this dimensional segment. In order to improve this algorithm, we extend it by applying the **DaVI** approach on the dimensional segment before building the multidimensional recommendation model. We describe in detail each of these phases below, emphasizing our proposal.

In Algorithm 1, we present the first phase which is a pre-processing phase usually performed offline. Line 1 of the algorithm determines all the “large” dimensional segments, i.e., the segments where the number of sessions belonging to the segment exceeds a pre-determined threshold γ. Here, the algorithm groups the sessions in segments, based on the dimension value of each session, and then selects only the segments which the number of sessions exceeds γ. In [Adomavicius et al. 2005], the authors use a threshold of 20%, which means that only segments containing at least 20% of the sessions in the data set S' are selected as “large” dimensional segments and stored in the set $SEGM(S')$.

Next, line 2 of the algorithm, for each “large” segment L, we first apply the **DaVI** approach on the dimensional segment L, then we run algorithm Θ using the extended sessions S'' of the segment L to build a recommendation model and determine its performance $\mu_{\Theta,S''}(S'')$. This can be done, for example, with the n-fold cross validation technique [Shani and Gunawardana 2011], as described in Section 5.2. With respect to the algorithm Θ, we can use any top-N recommendation algorithm. In this work, we use the Item-based Collaborative Filtering algorithm [Deshpande and Karypis 2004]. In this algorithm, the recommendation model is a matrix representing the similarities between all the pairs of items according to a similarity measure (in our case, the cosine angle).

We also run the algorithm Θ using the set S' (after excluding from S' the sessions which represent the test set of L) and compute its performance $\mu_{\Theta,S'}(L)$ on the test set of L. Then, we compare both results $\mu_{\Theta,S''}(S'')$ and $\mu_{\Theta,S'}(L)$ using, for example, a one-sided paired t-test with a 95% confidence level. We keep in $SEGM(S')$ only segments L which the performance of the algorithm Θ on the segment significantly outperforms the performance of Θ on the full set S'. Using the **DaVI** approach, we believe that more dimensional segments can outperform the full set S'.

Finally, in line 3, we remove from $SEGM(S')$ those segments L, for which there exists a strictly more general segment Q where the algorithm Θ performs better. Then, the set of high-performance dimensional segments $\overline{SEGM(S')}$ is returned (line 4).

Algorithm 1 The algorithm for determining high-performing dimensional segments. Adapted from: [Adomavicius et al. 2005].

Require: S', a set of sessions for a multidimensional recommendation space; μ, a performance metric function; γ, a threshold defining the minimal number of sessions for a "large" segment; $\widehat{D}$, a set indicating which dimension values in S' must be converted to virtual items; Θ, a top-N recommender algorithm; N, the number of recommendations which are generated.

Ensure: $\overline{SEGM(S')}$, a set of dimensional segments on which the reduction-based approach on the algorithm Θ significantly outperforms the pure algorithm Θ.

1: Let $SEGM(S')$ initially be the set of all large dimensional segments for the set S';

2: For each segment $L \in SEGM(S')$ compute $S'' = \textbf{DaVI}(L, \widehat{D})$, $\mu_{\Theta,S''}(S'')$ and $\mu_{\Theta,S'}(L)$, and keep only those segments $L \in SEGM(S')$ for which $\mu_{\Theta,S''}(S'')$ is better than $\mu_{\Theta,S'}(L)$;

3: Among the segments remaining in $SEGM(S')$ after Step 2, discard any segment L for which there exists a different segment Q such that $L \subset Q$ and $\mu_{\Theta,Q}(Q)$ is better than $\mu_{\Theta,L}(L)$. The remaining segments form the set $\overline{SEGM(S')}$ with the high-performance segments;

4: **Return** $\overline{SEGM(S')}$;

Once we have the set of high-performance dimensional segments $\overline{SEGM(S')}$, we can perform the second phase of the combined reduction-based algorithm (see Algorithm 2) to generate top-N recommendations in real time. Given an active session s'_a for which we want to generate the top-N recommendations, we first go over the dimensional segments $\overline{SEGM(S')} := \{L_1, ..., L_f\}$, ordered in the decreasing order of their performance metric (i.e., F1 measure), and select the best segment to which the active session s'_a belongs to (line 2). Then, we use the recommendation algorithm Θ on that segment, and return the top-N recommendations (lines 8 and 10). Again, before running the algorithm Θ, we apply the **DaVI** approach on the selected segment (lines 6 and 7). If s'_a does not belong to any segment, we use the pure two-dimensional recommendation algorithm Θ (i.e., trained on the set S') to generate the top-N recommendations (lines 4 and 10). In line 2, the active session s'_a belongs to a segment L_e if it has the same dimension value of L_e and all the items in s'_a are contained in the segment L_e.

As we can see, our proposal consists of applying the **DaVI** approach on the dimensional segment before building the multidimensional model. In this way, we extend the combined reduction-based algorithm to use the additional information as virtual items, using them together with the regular items in order to improve the recommendation model.

Algorithm 2 The combined approach for top-N recommendations. Adapted from: [Adomavicius et al. 2005].

Require: $\overline{SEGM(S')} := \{L_1, ..., L_f\}$, where segments L_1 through L_f are arranged in decreasing order with respect to μ, i.e., $\mu_{\Theta,L_1}(L_1) > ... > \mu_{\Theta,L_f}(L_f)$; $\widehat{D}$, a set indicating which dimension values in S' must be converted to virtual items; $M_{\Theta,S''}$, a recommendation model based on algorithm Θ and the extended segment S''; $M_{\Theta,S'}$, a recommendation model based on algorithm Θ and the set S'; N, the number of recommendations which are generated; s'_a, an active session for which we want to generate the top-N recommendations.
Ensure: R, the top-N recommendations for the active session s'_a.

$\phi := 0$;
For active session s'_a compute: $\phi := min_{e:=1,...f}\{e|s'_a \in L_e\}$;
if $\phi = 0$ **then**
    $R := M_{\Theta,S'}(s'_a)$;
else
    $S'' =$ **DaVI**$(L_\phi, \widehat{D})$;
    $s''_a =$ **DaVI**$(s'_a, \widehat{D})$;
    $R := M_{\Theta,S''}(s''_a)$;
end if
Return R;

5. Empirical Evaluation

We use our multidimensional recommendation strategy, described in the previous section, with the Item-based Collaborative Filtering [Deshpande and Karypis 2004], and evaluate its effects on two real world data sets of music. The data sets come from Palco Principal[3], a start-up company that holds a web site of Portuguese music since 2007. Besides music recommendations, the site also provides services like news, advertisements, social networking and an application for users to access the services of the site through their mobile phone.

5.1. Data Sets

The first data set, called *Dataset1*, contains accesses to music tracks from the Palco Principal web site. Each session 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, 6,428 different items (music tracks) and 9,740 sessions. The second data set (called *Dataset2*) represents the set of music 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, 5,428 different items and 4,417 sessions.

The additional information/dimension for the data sets are presented in Table 1. The dimensions *band* and *music_genre* indicate the type of content a user is looking for when he/she navigates through a web site.

[3] `http://www.palcoprincipal.com`

Table 1. Additional information for *Dataset1* and *Dataset2*.

Dimension	Description
day	Day of each access (from 01 to 31).
month	Month of each access (from 01 to 12).
week_day	Week day of each access (Monday to Sunday).
work_day	Accesses made on the week or the weekend.
hour	Hour of each access (from 01 to 24).
work_hour	Accesses made during working hours or not.
location	Location where the accesses were made.
band	The band which plays a music track.
music_genre	The genre of a music track (pop, rock, etc).

5.2. Experimental Setup and Evaluation Measures

As already stated, we use our recommendation strategy with the Item-based Collaborative Filtering algorithm [Deshpande and Karypis 2004]. In this algorithm, the recommender model is a matrix representing the similarities between all the pairs of items according to a similarity measure (in our case, the cosine angle). The top-N recommendations are generated based on the 4 most similar items (the 4 nearest neighbors). To tune the algorithm, we ran a first set of experiments using different numbers of neighbors and analyzed the F1 measure. We observed that the F1 values tend to increase from 2 to 4 neighbors. For 5 neighbors, the values were a bit worse than for 4 neighbors. Therefore, we have chosen the 4 most similar items to make the recommendations. Here, we setup $\widehat{D}$ to indicate that all dimension values in S' must be converted to virtual items.

To measure the predictive ability of the recommender systems, we use the All But One protocol with 10-fold cross validation and calculate the Precision (the number of recommended items that are relevant), Recall (the number of relevant items that are recommended) and F1 measure (the harmonic mean of the previous measures) for the recommendations [Shani and Gunawardana 2011]. To do this, the sessions in a data set are randomly partitioned into 10 subsets. For each fold, we use $n-1$ of those subsets of data for training and the remaining one for testing. The training set is used to build the recommendation model. For each session in the test set, we randomly hide one regular item, referred to as the singleton set H. The remaining items represent the set of observables, O, based on which the recommendation is made. The Precision, Recall and F1 are calculated by comparing, for each session in the test set, the set of recommendations R that the system makes, given the set of observables O, against the set H:

$$Precision = \frac{|R \cap H|}{|R|}, \tag{4}$$

$$Recall = \frac{|R \cap H|}{|H|}, \tag{5}$$

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall}. \tag{6}$$

We obtain global Precision, Recall and F1 by averaging individual values from each session. Then, for each measure, the 10 values are summarized by using mean and standard deviation. To compare two recommendation algorithms, we apply the two-sided paired t-test with a 95% confidence level. We ran the experiments for N equal $1, 2, 3, 5$ and 10, where N is the number of items to be recommended by the models.

5.3. Results

In Table 2, we present the mean of Precision, Recall and F1 measure for our proposal (DaVI-C. Reduction), for the original combined reduction-based algorithm (C. Reduction), and for the two-dimensional recommender system based on the Item-based Collaborative Filtering algorithm ($User \times Item$). The table allows us to compare our proposal against the C. Reduction and $User \times Item$ algorithms.

Table 2. Comparing our extended multidimensional recommendation algorithm (DaVI-C. Reduction) against the combined reduction-based (C. Reduction) and the traditional two-dimensional ($User \times Item$) algorithms. Values in boldface are statistically significant.

		Dataset1			Dataset2		
Algorithm	N	Precision	Recall	F1	Precision	Recall	F1
$User \times Item$	1	0.231	0.231	0.231	0.342	0.342	0.342
C. Reduction	1	0.231	0.231	0.231	0.342	0.342	0.342
DaVI-C. Reduction	1	**0.253**	**0.253**	**0.253**	**0.363**	**0.363**	**0.363**
$User \times Item$	2	0.169	0.338	0.226	0.219	0.439	0.293
C. Reduction	2	0.169	0.338	0.226	0.219	0.439	0.293
DaVI-C. Reduction	2	**0.173**	**0.347**	**0.231**	**0.221**	**0.442**	**0.295**
$User \times Item$	3	0.132	0.396	0.198	0.161	0.484	0.242
C. Reduction	3	0.132	0.396	0.198	0.161	0.484	0.242
DaVI-C. Reduction	3	**0.134**	**0.401**	**0.201**	0.162	0.485	0.243
$User \times Item$	5	0.091	0.456	0.152	0.107	0.534	0.178
C. Reduction	5	0.091	0.456	0.152	0.107	0.534	0.178
DaVI-C. Reduction	5	**0.092**	**0.461**	**0.154**	0.107	0.534	0.178
$User \times Item$	10	0.051	0.509	0.092	0.057	0.572	0.104
C. Reduction	10	0.051	0.509	0.092	0.057	0.572	0.104
DaVI-C. Reduction	10	**0.052**	**0.524**	**0.095**	0.057	0.573	0.104

The C. Reduction algorithm presents results (i.e., Precision, Recall and F1 measure) similar to the $User \times Item$ algorithm for both data sets (see Table 2). This fact occurs because the models built with the dimensional segments do not outperform the traditional two-dimensional model. Therefore, the C. Reduction algorithm makes its recommendations based on the traditional recommendation model.

On the other hand, we observe in Table 2 that the DaVI-C. Reduction algorithm is better than the C. Reduction and $User \times Item$ algorithms for both data sets. For the *Dataset1*, the DaVI-C. Reduction algorithm presents Precision gains ranging from 1.1% to 9.5%, Recall from 1.1% to 9.5% and F1 ranging from 1.3% to 9.5%. For the *Dataset2*, the Precision gains ranging from 0.9% to 6.1%, Recall from 0.7% to 6.1% and F1 ranging from 0.7% to 6.1%. These gains occur because our proposal is able to generate more dimensional segments that outperform the full data set, by using dimension values as virtual items into the recommendation models.

6. Final Remarks

Most web sites offer a large number of information resources (i.e., events, books, music, web pages, etc) to their users. Finding relevant content has, thus, become a challenge for users. The recommender systems have emerged in response to this problem.

In this paper, we extended the combined reduction-based algorithm to use the **DaVI** approach, which treats additional dimensions as virtual items into the recommendation model. With such an extension, this multidimensional algorithm can use the additional dimensions not only as labels for segmenting sessions but also as virtual items to improve the recommendation model. In the empirical evaluation, we obtained gains of 9.5% (*Dataset1*) and 6.1% (*Dataset2*) for Precision, Recall and F1 measure. The empirical evaluation shows that with our proposal, the combined reduction-based algorithm reduces the needs of reverting to the traditional two-dimensional algorithm to generate the top-N recommendations, increasing the accuracy of its recommendations.

As future work, we will expand our findings by using other data sets as well as by defining some strategy to setup the parameter $\widehat{D}$, that indicates which dimension values in S' must be converted to virtual items. We will also compare our proposal against other algorithms for multidimensional recommendation.

Acknowledgments. This work is supported by the grant 2012/13830-9, São Paulo Research Foundation (FAPESP). This work is also partially funded by FCT/MEC through PIDDAC and ERDF/ON2 within project NORTE-07-0124-FEDER-000059. The first author is supported by a research grant from CAPES/BRAZIL.

References

Adomavicius, G., Sankaranarayanan, R., Sen, S., and Tuzhilin, A. (2005). Incorporating contextual information in recommender systems using a multidimensional approach. *ACM Transactions on Information Systems*, 23(1):103–145.

Adomavicius, G. and Tuzhilin, A. (2001a). Extending recommender systems: A multidimensional approach. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI-01), Workshop on Intelligent Techniques for Web Personalization (ITWP2001)*, Seattle, Washington.

Adomavicius, G. and Tuzhilin, A. (2001b). Multidimensional recommender systems: A data warehousing approach. In *WELCOM '01: Proceedings of the Second International Workshop on Electronic Commerce*, pages 180–192, London, UK. Springer-Verlag.

Adomavicius, G. and Tuzhilin, A. (2011). Context-aware recommender systems. In Ricci, F., Rokach, L., Shapira, B., and Kantor, P. B., editors, *Recommender Systems Handbook*, pages 217–253. Springer US.

Anand, S. S. and Mobasher, B. (2003). Intelligent techniques for web personalization. In *Intelligent Techniques for Web Personalization (ITWP 2003), LNCS 3169*, pages 1–36.

Breese, J. S., Heckerman, D., and Kadie, C. M. (1998). Empirical analysis of predictive algorithms for collaborative filtering. In *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence*, pages 43–52.

Deshpande, M. and Karypis, G. (2004). Item-based top-n recommendation algorithms. *ACM Transaction on Information System*, 22(1):143–177.

Domingues, M. A., Jorge, A. M., and Soares, C. (2009). Using contextual information as virtual items on top-n recommender systems. In *ACM RecSys'09 Workshop on Context-Aware Recommender Systems (CARS-2009)*.

Domingues, M. A., Jorge, A. M., and Soares, C. (2013). Dimensions as virtual items: Improving the predictive ability of top-n recommender systems. *Inf. Process. Manage.*, 49(3):698–720.

Linden, G., Smith, B., and York, J. (2003). Amazon.com recommendations: Item-to-item collaborative filtering. *IEEE Internet Computing*, 7(1):76–80.

Resnick, P. and Varian, H. R. (1997). Recommender systems. *Communications of the ACM*, 40(3):56–58.

Ricci, F., Rokach, L., Shapira, B., and Kantor, P. B., editors (2011). *Recommender Systems Handbook*. Springer.

Shani, G. and Gunawardana, A. (2011). Evaluating recommendation systems. In Ricci, F., Rokach, L., Shapira, B., and Kantor, P. B., editors, *Recommender Systems Handbook*, pages 257–297. Springer US.