

Distributional Semantic Pre-filtering in Context-Aware Recommender Systems

Victor Codina

vcodina@lsi.upc.edu

Software Department, Technical University of Catalonia, Girona 1-3, K2M 201, 08034 Barcelona, Spain

Francesco Ricci

fricci@unibz.it

Faculty of Computer Science, Free University of Bozen-Bolzano, Piazza Domenicani 3, 39100 Bolzano, Italy

Luigi Ceccaroni

luigi@1000001labs.org

1000001 Labs, c. Alzina 52, 08024 Barcelona, Spain

Abstract

Context-aware recommender systems improve context-free recommenders by exploiting the knowledge of the contextual situation under which a user experienced and rated an item. They use data sets of contextually-tagged ratings to predict how the target user would evaluate (rate) an item in a given contextual situation, with the ultimate goal to recommend the items with the best estimated ratings. This paper describes and evaluates a pre-filtering approach to context-aware recommendation, called Distributional-Semantics Pre-filtering (DSPF), which exploits in a novel way the distributional semantics of contextual conditions to build more precise context-aware rating prediction models. In DSPF, given a target contextual situation (of a target user), a matrix-factorization predictive model is built by using the ratings tagged with the contextual situations most similar to the target one. Then, this model is used to compute rating predictions and identify recommendations for that specific target contextual situation. In the proposed approach, the definition of the similarity of contextual situations is based on the distributional semantics of their composing conditions: situations are similar if they influence the user's ratings in a similar way. This notion of similarity has the advantage of being directly derived from the rating data; hence it does not require a context taxonomy. We analyze the effectiveness of DSPF varying the specific method used to compute the situation-to-situation similarity. We also show how DSPF can be further improved by using clustering techniques. Finally, we evaluate DSPF on several contextually-tagged data sets and demonstrate that it outperforms state-of-the-art context-aware approaches.

Keywords

Context-Awareness, Recommender Systems, Distributional Semantics, Collaborative Filtering, Matrix Factorization, Pre-Filtering, Clustering

1 Introduction

Context-Aware Recommender Systems (CARSs) differ from traditional recommenders because they predict the rating of a target user u for an item i , not only by using an existent data set of ratings but also exploiting: a) the knowledge of the contextual situations under which the ratings were acquired, and b) the knowledge of the (real-time) contextual situation of the target user asking for a recommendation.

The notion of context is ubiquitous to several scientific disciplines and has several interpretations (Bazire and Brezillon 2005), but, in CARSs, context is commonly defined as all the information that characterizes the situation of a user, an item, and the experience (interaction between a user and an item) that the user is evaluating. Dourish (2004) introduces a taxonomy of contexts by classifying contextual information as either pertaining to the “interactional” or “representational” views. According to Dourish, the *interactional* view assumes that the user behavior is induced by an underlying context, but the context itself is not necessarily observable. Therefore, no enumeration of contextual conditions is possible beforehand, since the scope of contextual information is defined dynamically. In contrast, the *representational view* assumes that context is defined with a predefined set of observable conditions, which can be separated from the activity and the structure of which does not change significantly over time. In other words, using this view a set of contextual factors and their conditions are identifiable, they are known a priori and, therefore, can be represented beforehand. In this paper, as in the majority of the researches on context-aware recommender systems, we follow the representational view of context. In particular, by using the term contextual *factor* we refer to a specific type of contextual information (e.g., weather), and with contextual *condition* we refer to a specific value of a contextual factor (e.g., sunny). The term contextual *situation* refers to a combination of elementary contextual conditions that describe the context in which the user experienced the item, e.g., “today is sunny and it is holiday”.

The main motivation for introducing CARS techniques is that, when context matters, the rating data acquired in the target contextual situation should be more relevant for predicting what to recommend in that situation. However, a common limitation of these techniques is the data-sparsity problem; in order to generate accurate recommendations CARSs need large data sets of contextually-tagged ratings. These are ratings for items provided in several contextual situations, which are encountered by a user while experiencing an item.

A solution to that data-sparsity problem, when making recommendations in a specific situation, is to consider as relevant background data, not only the ratings provided by the users exactly in that situation, but also to consider ratings provided in *similar* situations. For instance, if we want to predict the rating for a place of interest, e.g., the South Tyrol Museum of Archaeology (in Bolzano, Italy), and the target contextual situation includes a condition such as, “group composition is two adults and two children”, ratings acquired when the “group composition is two adults and three children” may also be useful for computing the rating prediction model, and consequently the recommendations for the target situation. But, what about ratings acquired while the “weather” was “sunny”? A “sunny” day and a group of “two adults and two children” do not seem to be similar contextual conditions although, depending on the recommendation domain, they may be highly related. These cross-factor semantic associations are not easy to identify at design stage. But, following with the same example in the tourism domain, if we analyze how a “sunny” day and a group of “two adults and two children” influence the users’ ratings, we may discover that they actually have a similar influence pattern: they both tend to increase the user’s ratings for outdoor places like castles and decrease them for indoor places like museums. Therefore, based on the similarity of their influence patterns (i.e., their distributional semantic similarities) we may consider them as semantically similar.

In distributional semantics, the meaning of a concept is based on its distributional properties, which are automatically derived from the data corpus where the concept is used. The fundamental idea supporting this approach to extract semantic similarities between domain concepts is the so-called distributional hypothesis (Rubenstein and Goodenough 1965): *concepts repeatedly co-occurring in the same context or usage tend to be related*. Originally, distributional semantics has been introduced in the context of applications that require semantic processing of text (Turney and Pantel 2010; Molino

2013). Later on, in the Recommender Systems (RSs) field, some researchers have exploited distributional semantics for overcoming some of the limitations of content-based recommenders. For instance, Musto et al. (2014) presented a content-based context-aware recommendation framework that adopts a novel distributional semantics representation to extract latent relationships between item’s concepts, which are then exploited to improve the system prediction accuracy. Differently from that previous research, in this paper we employ distributional semantics to infer similarities between contextual conditions based on an analysis of how similarly two contextual conditions affect the users’ rating behavior.

We present a reduction-based pre-filtering approach, called *Distributional-Semantics Pre-filtering* (DSPF), which computes similarities between situations based on the distributional semantics of their contextual conditions, i.e., assuming that two situations are similar if they are defined by elementary contextual conditions that influence users’ ratings in a similar way. Given a target contextual situation, DSPF uses ratings tagged with contextual situations similar to the target one to generate more precise recommendations. In order to determine if a candidate situation is similar enough to the target one, DSPF uses a global *similarity threshold* that specifies the minimum similarity required for situations to be considered as relevant and useful to build a predictive model for the target situation: the larger the threshold, i.e., the higher is the required similarity, the sharper the contextualization is. This implies that fewer situations, i.e., the ratings collected in these situations, are used to build the rating prediction model adapted to the target contextual situation. For that reason we say that the predictive model is (more) “local”.

Early variants of DSPF were introduced by Codina et al. (2013a; 2013b). Codina et al. (2013a) presented an initial definition of the approach and evaluated two situation-to-situation similarity computation methods based on the pair-wise evaluation of the similarities between the conditions that are composing the contextual situations. Codina et al. (2013b) provided a more sophisticated method to obtain the distributional semantics of contextual conditions and to estimate the situation-to-situation similarity. Moreover, the performance of DSPF was compared to other state-of-the-art approaches using several contextually-tagged rating data sets.

In this paper, we introduce an improved and scalable variant of DSPF that reduces the number of the generated local models by using clustering strategies. Furthermore, we include the results of a comprehensive experimental evaluation of the proposed approach by using six data sets with different types of contextual information and rating sparsity. The evaluation results show that DSPF outperforms state-of-the-art context-free and context-aware approaches both in terms of rating prediction (MAE) and ranking (NDCG) accuracy.

The remainder of this paper is organized as follows. Section 2 positions our work with respect to the state of the art. Section 3 presents the details of DSPF and the proposed variants. Section 4 presents the experimental evaluation of DSPF on the considered data sets as well as a detailed analysis of the effect of the similarity threshold on the system performance. Finally, section 5 draws the main conclusions and describes the future work.

2 Related work

CARS are generally classified into three paradigms (Adomavicius and Tuzhilin 2011; Adomavicius et al. 2011):

- (1) *contextual pre-filtering*, where context is used for selecting a set of contextually relevant rating data that is then exploited for generating target context-dependent recommendations (using a context-free model);
- (2) *contextual post-filtering*, where context is used to adjust (filter) recommendations generated by a context-free model; and
- (3) *contextual modeling*, in which contextual information is directly exploited in the adopted context-aware recommendation model.

In contextual pre-filtering, context is used for data pre-processing, that is, for discarding rating data that is not relevant for the target situation the recommender is facing. The remaining ratings are used to learn a local model for rating prediction and recommendation. Adomavicius et al. (2005) proposed a straightforward approach to implement this idea known as reduction-based. The first variant of this approach was *Exact Pre-filtering*, which strictly implements the pre-filtering idea and builds a local context model for each target situation. This is achieved by using exactly the ratings tagged with that situation. The main limitation of this approach is its rigidity: it does not reuse any rating acquired in situations even slightly different from the target one, regardless of the number of the remaining training ratings that can be used for learning the local model, which may not be sufficient for that task. To mitigate this shortcoming the authors proposed *Generalized Pre-filtering*, which first determines the optimal aggregation of ratings tagged with hierarchically related contextual situations (i.e., optimal segments), and then builds a collection of prediction models using the ratings belonging to each segment. However, the performance of this solution depends on the pre-defined context hierarchical taxonomy, which may not optimally partition the contextual situations in a collection of local models (one for each segment) having jointly the highest rating prediction accuracy.

A different approach was introduced in Baltrunas and Ricci (2009; 2014), which is known as *Item Splitting*. The idea here is to split the rating vector of a given item into two virtual item vectors using a specific contextual factor. So for instance the full set of ratings of a music track may be split in the set of ratings for the track collected when the user was happy and another set of ratings acquired when the user was not happy (assuming in this example that *happiness* is the contextual factor). Then a predictive model is trained by considering all the ratings organized in the extended set of items generated by splitting the items that satisfy a given statistical test, which is essentially measuring if the two virtual items generated by the splitting have significantly different ratings. In *Item Splitting*, filtering is selectively carried out item by item for the most relevant contextual condition. Baltrunas and Amatriain (2009) proposed a variant of this approach, which is called *User Splitting* that, instead of splitting items, splits users into several sub-profiles, each representing the (ratings of the split) user in a particular context. Then, similarly to the previous approach, a global predictive model is built using all the ratings but in the modified set of users. Zheng et al. (2013a) also explored a combination of the two previous variants, *UI-Splitting*, which yielded even a better prediction accuracy in a movie recommendation problem.

Another pre-filtering approach, proposed by Zheng et al. (2012; 2013b), is differential context modeling, which tries to break down a predictive model into different functional components to which specific optimal contextual constraints are applied in order to maximize the performance of the whole algorithm. The authors proposed two variants of this pre-filtering approach used in combination with the traditional user-based CF technique: *Differential Context Relaxation*, which discards the contextual factors with small relevance for a given function component; and *Differential Context Weighting*, which weights the contribution of each factor according to their relevance for the given component.

As we mentioned above, contextual post-filtering exploits contextual information to discard some (irrelevant) recommendations, after an initial recommendation set is determined by a context-free predictive model. Panniello et al. (2009) proposed a probabilistic post-filtering approach that first estimates the probability of an item to be relevant for the user in a given context, and then uses these probabilities to penalize the items estimated as not relevant to the target context. Two variants are presented: *Weight Post-filtering*, which reorders the recommended items by weighting the predicted ratings according to their estimated probability; *Filter Post-filtering*, which filters out the recommended items that have a probability to be relevant lower than a specific threshold. Hayes and Cunningham (2004) presented a content-based post-filtering approach that focuses on finding common item features (e.g., preferred actors to watch) for a given user in a given context, and then uses these features to adjust the recommendations.

Approaches based on contextual modeling extend context-free predictive models by directly taking into account the influence of context on the rating prediction model, i.e., by typically adding new model parameters that represent the contextual information. Currently, two major approaches based on extending Matrix Factorization (MF) techniques have been proposed in the literature: *Tensor*

Factorization (TF) and *Context-Aware Matrix Factorization* (CAMF). Tensor Factorization consists of extending the two-dimensional MF problem into a multi-dimensional version, where the rating tensor is factored into a lower-dimensional vector space. In this way, the interactions between users, items, and contextual factors are represented as latent factor vectors. Several authors have proposed variants of this approach: some of them are optimized for rating prediction, such as the *Multiverse Recommendation* (Karatzoglou et al. 2010) and *Factorization Machines* (Rendle et al. 2011), while others are optimized for ranking-based recommendation, such as *iTALS* (Hidasi and Tikk 2012), and *TFMAP* (Shi et al. 2012). The main limitation of TF is its computational complexity. In fact, the number of model parameters to be learnt grows exponentially with the number of contextual factors.

CAMF is a more scalable contextual modeling approach that was proposed by Baltrunas et al. (2011b; 2012). CAMF extends MF by using context-aware baseline predictors to represent the interactions of contextual information with the items or users. In this way a smaller number of parameters, compared with TF, is used. It generalizes the time-aware baseline predictor that was initially proposed by Koren (2010) to incorporate the temporal dynamics associated to rating data. This technique was proved to be effective when combined with MF on the Netflix data set (Koren and Bell 2011; Campos et al. 2014). Baltrunas et al. (2011b, 2012) proposed different variants of CAMF that model the influence of contextual conditions at different granularities. *CAMF-C* models the influence of a condition in a global way, i.e., assuming that it has the same effect on every user and item. *CAMF-CI* models the influence of a contextual condition uniformly on each item, i.e., assuming that it does not depend on the user. Finally, *CAMF-CC* assumes that context influences uniformly the ratings for all the items of the same type (i.e., item categories). More recently, Odić et al. (2013) proposed a variant of CAMF that models the influence of contextual conditions with respect to the users (*CAMF-CU*).

Recent empirical analyses indicate that there is no single best approach, among pre-filtering, post-filtering and contextual-modeling, and the best performing one depends on the recommendation task and the application domain (Panniello et al. 2014). Their analysis shows that the accuracy of all considered CARS techniques decreases when the contextual information has a finer granularity and hence fewer ratings tagged with the target situation are available. In this article, we claim that it is possible to overcome such limitation by exploiting the distributional similarity of contextual situations directly in the context modeling phase.

Distributional-Semantics Pre-filtering (DSPF), the method proposed in this article, is analogous to *Generalized Pre-filtering*: it is a reduction-based approach, but instead of searching for the optimal segmentation of the ratings, it exploits similarities between situations to generate segments that aggregate the ratings tagged with situations similar to the target one. Hence, the key difference is that our approach leverages the knowledge of the situation-to-situation similarity of contextual conditions instead of relying on the usually limited condition-to-condition hierarchical relationships defined in a context taxonomy. As we will show later, our approach supports a more flexible and effective aggregation of ratings and thus yields a better accuracy, especially when the contextual situations considered in the application are very specific, i.e., defined by the conjunctions of several contextual conditions.

3 Distributional-Semantics Pre-Filtering

To better understand DSPF, it is worth recalling how reduction-based pre-filtering operates. When it is requested to compute recommendations in a target contextual situation, reduction-based pre-filtering executes two procedural steps:

- firstly, a subset of the training ratings, which are judged as relevant to that contextual situation, is selected from the training set;
- secondly, a predictive model is trained on the selected ratings, which is then used to make predictions to users exactly in that situation.

We say that this model is “local” because it is not based on the full set of available ratings but exploits only a subset of more relevant ratings, which is estimated to produce better predictions in that particular contextual situation.

The key step of this process is therefore the selection of the ratings, i.e., the estimation of what ratings are relevant to better model the users' rating behavior in the target contextual situation. In DSPF, in addition to the ratings acquired exactly in the target situation, ratings acquired in situations "similar" (enough) to the target one are also used. DSPF uses a custom definition of similarity that will be described in the next section. The selection of the "similar" contextual situations is also determined by a *similarity threshold* (t), which is a global parameter that must be tuned to the data set; it determines the minimum similarity score between two situations to make one reusable when the target contextual situation is defined by the other. The larger the threshold is, i.e., the closer is to 1 (maximum similarity), the less contextual situations are selected and consequently the more the rating prediction model fits the target contextual situation. In particular, when $t=1$ the prediction model is very local and it is equal to the one built by *Exact Pre-filtering*: only the ratings acquired in the target contextual situation are used.

As in other machine learning tasks, it may not be the case that a model fitting the most specific training data (the ratings provided in the target contextual condition) provides the best predictions on future data, i.e., not used to train the model: *overfitting* the local contextual situations may jeopardize the overall system behavior especially when ratings data are scarce. Hence, one must detect the optimal middle point between a global model based on all user ratings (i.e., a context-free model) and a strict local model, which is just fitting the user ratings in a specific context. In DSPF we have designed an approach to find the right level of contextualization for a given data, i.e., to learn the similarity threshold that maximizes rating prediction accuracy.

Making the above discussion more precise, given a target contextual situation s^* and a similarity threshold t , in DSPF the local training set R_{s^*} , which is the set of the ratings acquired in situation s^* , is expanded by adding all the ratings acquired in all the situations s where $\text{Sim}(s, s^*) \geq t$. This expanded training set is then used for building the local rating prediction model for the target situation s^* . Denoting with R_s the set of ratings acquired in situation s , the set X_{s^*} of the training data for the local model of s^* is therefore:

$$X_{s^*} = \bigcup_{s: \text{Sim}(s, s^*) \geq t} R_s \quad (1)$$

Figure 1 illustrates the generation of the training data for a target contextual situation in DSPF. In this example it is shown that only the ratings tagged with situation s^* and s_1 are selected. We note that, if more than one rating for a given user and item are available in the selected contextual situations similar to the target one, an average of these ratings is computed in order to generate a unique rating for a given user, item and contextual situation. Using this procedure we reduce the original multidimensional rating data to a two-dimensional rating matrix that can be used for training any CF context-free prediction model.

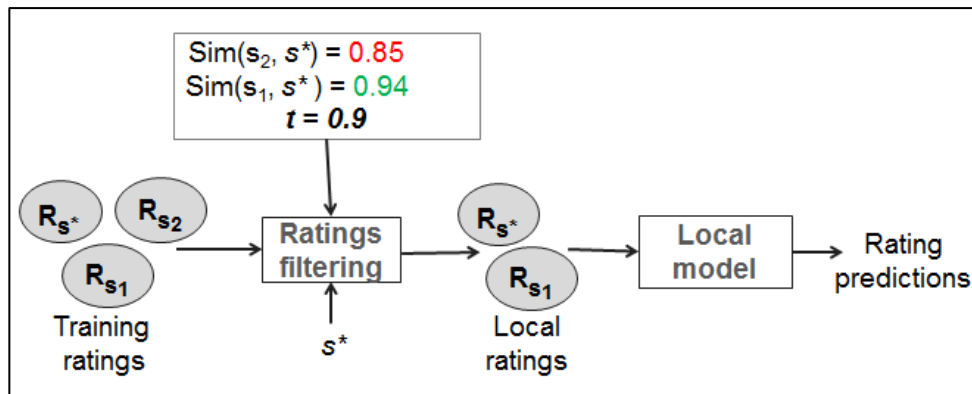


Figure 1. Example of the ratings selection process adopted by DSPF

3.1 Distributional semantics of contextual conditions

In the previous section we referred to the usage of a similarity function between contextual situations in order to determine which ratings must actually be used to generate context-dependent recommendations for any given target contextual situation. In this section, we describe how a viable similarity function can be computed. The proposed situation-to-situation similarity is based on the distributional semantics of contextual conditions, i.e., it is rooted on the assumption that two situations are similar if their composing conditions influence users' ratings in a similar way. To represent the distributional semantics of concepts we use a semantic-vector space with the goal to define the similarity between concepts in terms of their proximity in a high-dimensional vector space.

In this article we propose to model a contextual condition by describing its influence on the average rating either of the items or of the users. Hence, the dimensionality of the resulting semantic vectors is equal to either the number of items or users. In more detail, our method exploits rating information to measure the influence of a condition as the produced deviation between the observed ratings when the condition holds (r_{uic}), and the predicted context-free rating (\hat{r}_{ui}). If we use a *user-based* perspective, then the influence of a condition c on a user u , which is denoted by w_{cu} , is calculated as follows:

$$w_{cu} = \frac{1}{|R_{uc}| + \beta} \sum_{r_{uic} \in R_{uc}} (r_{uic} - \hat{r}_{ui}) \quad (2)$$

where R_{uc} is the set of ratings of the user u in condition c ; and β is a decay factor used to cope with the lack of reliability of the proposed measure w_{cu} when only a limited number of user u ratings are available in a given contextual condition c . The decay factor has the effect of decreasing the estimated deviation w_{cu} when $|R_{uc}|$ is small. The rationale is that the smaller the number of ratings is, the less reliable the estimated deviation is (hence it must be closer to 0). In our experiments we obtained the best results when $\beta \in [0,10]$; the exact value depends on the data set.

In this article we use as context-free predictive model the baseline predictor presented in Koren and Bell (2011) that optimizes the model parameters by using stochastic gradient descent and whose prediction formula is: $\hat{r}_{ui} = \mu + b_u + b_i$, where μ is the overall rating average, b_u is the bias associated to the user u , and b_i the bias associated to the item i . We also tested more sophisticated context-free predictive models but no significant performance differences were observed.

Analogously to what it is shown above, the measure of the impact of a contextual condition can also be based on its effect on the ratings for an item. If R_{ic} denotes the set of ratings for item i in condition c , then in the *item-based* perspective the influence of the condition c on the item i , which is denoted by w_{ci} , is defined as follows:

$$w_{ci} = \frac{1}{|R_{ic}| + \beta} \sum_{r_{uic} \in R_{ic}} (r_{uic} - \hat{r}_{ui}) \quad (3)$$

Using either formula (2) or (3) we can build a semantic-vector representation of each condition with respect to either the users or the items respectively.

Figure 2 shows the semantic vectors of three contextual conditions in a hypothetical scenario where only six items are present in the recommender system. This exemplifies the application of the *item-based* perspective. In such a representation, a positive value (of w_{ci}) means that the condition tends to increase the ratings given to the item, a negative value means that in that condition the item tends to be rated lower, and zero indicates that the condition has overall no effect on the item ratings: the larger the value, the larger the impact of the condition.




Condition		Item 1	Item 2	Item 3	Item 4	Item 5	Item 6
sunny		0.8	-0.9	0.2	0.9	0.1	-0.8
family		0.1	-1.1	0.9	0.1	0.9	0
happy		0.6	-0.7	1.2	1.1	0.9	-0.7

Figure 2. Semantic vectors of three conditions with respect to six items

3.2 Situation-to-situation similarity

Relying on the previously described semantic representation of contextual conditions, we can measure the semantic similarity between two contextual conditions and then between two generic contextual situations. We recall that a contextual situation is defined by the conjunction of one or more conditions (e.g., a contextual situation may be defined by temperature=hot, season=summer and mood=happy).

If the compared situations are defined by only one condition, we define the situation-to-situation similarity as the condition-to-condition similarity between the candidate c and the target condition c^* , which is calculated as the cosine of the angle between their respective semantic vectors, denoted by w_c and w_{c^*} respectively (l is the dimensionality of the semantic representation of a contextual condition):

$$sim(c, c^*) = \frac{w_c^T w_{c^*}}{\sqrt{\sum_{i=0}^l w_{ci}^2} \times \sqrt{\sum_{i=0}^l w_{c^*i}^2}} \quad (4)$$

Note that cosine similarity ranges in $[-1,1]$, but in our experiments we never considered negative values as acceptable similarity thresholds. In this paper, we evaluate two strategies to measure situation-to-situation similarity when situations are defined by the conjunction of several conditions:

- **Aggregative Similarity**, aggregating the pair-wise similarities of the conditions belonging to the compared situations. In this case the similarity between a target situation s^* and a candidate situation s is calculated by comparing *all-pairs* of conditions in the two situations, as follows:

$$sim(s, s^*) = \frac{1}{|s| \times |s^*|} \sum_{c \in s} \sum_{c^* \in s^*} sim(c, c^*) \quad (5)$$

- **Direct Similarity**, which directly measures the similarity of two situations by representing the situations, similarly to the conditions, as vectors of influence scores on the items or users. Here the similarity of two situations is calculated by defining first a vector representation of a situation, and then comparing these vector representations. The semantic vector representation of a contextual situation is defined as the centroid of the semantic vectors representing its known conditions:

$$w_s = \frac{1}{|s|} \sum_{c \in s} w_c \quad (6)$$

Then, the similarity between a target situation s^* and a candidate situation s is estimated as the cosine of the angle between their corresponding semantic vectors:

$$sim(s, s^*) = \frac{w_s^T w_{s^*}}{\sqrt{\sum_{i=0}^l w_{si}^2} \times \sqrt{\sum_{i=0}^l w_{s^*i}^2}} \quad (7)$$

Figure 3 shows the semantic vectors of three contextual situations defined by the conjunctions of two conditions as in the example shown in **Figure 2**.

Situation	Item 1	Item 2	Item 3	Item 4	Item 5	Item 6
$S_1 = \langle \text{sunny, family} \rangle$	0.5	-1	0.6	0.5	0.5	-0.4
$S_2 = \langle \text{family, happy} \rangle$	0.4	-0.9	1.1	0.6	0.9	-0.4
$S_3 = \langle \text{sunny, happy} \rangle$	0.7	-0.8	0.7	1	1	-0.8

Figure 3. Semantic vectors in a situation-to-item influence matrix

3.3 Improving the scalability by using clustering techniques

A potentially limitation of DSPF, which is common to all the reduction-based approaches, is that DSPF needs to learn a local rating prediction model for each target contextual situation that the system may face. This means that, depending on the number of possible situations and the size of each local prediction model, DSPF can be more memory-demanding than other context-aware techniques where a unique global prediction model is needed (e.g., contextual post-filtering and contextual modeling). In the worst case scenario, when the complexity of each local model is almost the same as the global model (i.e., the local models have the same number of parameters as the global one) the memory consumption may be significantly larger than other methods, because for each contextual situation a model (of the same size of the global one) must be stored on the system secondary memory. However, as we will show in Section 4.6, the system’s identified local models are trained on a small subset of the training rating data (i.e., only with the ratings tagged with situations strongly similar to the target one). Hence the local models complexity, expressed by the number of parameters, does not need to be as large as that required by other global context-free and context-aware approaches, such as Tensor Factorization (Karatzoglou et al. 2010; Rendle et al. 2011; Hidasi and Tikk 2012; Shi et al. 2012) and CAMF (Baltrunas et al. 2011b; 2012).

Moreover, during the initial system test we conjectured that many of the local models produced by DSPF may be very similar to each other, hence “merging” together models trained on almost the same ratings may be an effective approach to save space without paying too much in terms of prediction accuracy. Therefore, in a set of experiments illustrated in this article, we have studied the effectiveness of clustering techniques in reducing the number of local models used by DSPF. We have developed and evaluated two different clustering strategies with the goal of reducing the number of local models while preserving the prediction accuracy of the original, non-clustered, DSPF:

- *K-means-based clustering strategy*, first uses the k-means clustering algorithm (Rajaraman and Ullman 2012) in order to find k optimal groups of contextual situations, and then, for all the situations in a cluster (and for each cluster) it selects, as relevant set of training ratings, those acquired in the situations that belong to the cluster.
- *Hierarchical clustering strategy* uses bottom-up hierarchical clustering by initially considering as clusters all the local set of ratings (i.e., the ratings associated to each possible target contextual situation), and then iteratively merging the most similar pairs of clusters.

Note that these clustering strategies employ different similarity functions: the *k-means*-based strategy estimates situation-to-situation similarities by comparing their representative semantic vectors, whereas the *hierarchical* strategy compares two situations by considering the ratings previously identified as relevant for the given situations.

We implemented the *k-means*-based clustering strategy using the standard algorithm (also known as Lloyd’s algorithm). **Figure 4** shows the pseudo-code of the full learning process. The process has two parameters as input: the set of contextual situations to be clustered, and the exact number of clusters to produce. The algorithm begins initializing the clusters by randomly assigning each semantic vector of

a situation to one of the possible k clusters. Then, it follows an iterative process, which ends when the clustering converges, that is, all the situations are assigned to the nearest cluster. At each iteration, two main operations are carried out:

- (1) The centroid of each cluster is updated computing the average situation representation (i.e., a semantic vector) over the situations currently in the cluster.
- (2) Each situation is moved to the nearest cluster, based on the Euclidean distance between the centroid of the cluster and the semantic vector of the situation.

In this algorithm we used the Euclidean distance, rather than the cosine situation-to-situation similarity (see Eq. 7), because the standard k-means assumes a Euclidean space and hence using other distance metrics do not ensure the convergence of the algorithm (Rajaraman and Ullman 2012). Once the clusters of contextual situations are generated, for each cluster a local rating prediction model is built by using as training set only the ratings acquired in the situations that belong to the cluster. In this case, instead of using the similarity threshold, the level of expansion of the local models is controlled by the hyper-parameter k : the larger k , the fewer contextual situations are aggregated per cluster. As in the similarity threshold, the optimal k for each data set must be determined experimentally (cross-validation).

```

Input
  P: initial set of contextual situations to be clustered (i.e.
  semantic vectors representing the contextual situations)
  k: exact number of clusters to produce
Output
  models: k local prediction models (one for each cluster)

clusters = getRandomInitialGroups(P, k);
movement = true;
while movement do
  movement = false;
  centroids = computeMeans(clusters);
  foreach  $p \in P$  do
    nearestClusterID = findNearestCluster(p, centroids);
    if (p.clusterID  $\neq$  nearestClusterID) then
      assignPointToNearestCluster(p, clusters, nearestClusterID);
      movement = true;
    endif
  endfor
endwhile
foreach  $c \in clusters$  do
   $R_c = \text{getRelevantRatings}(c)$ ;
   $M_c = \text{buildLocalModel}(R_c)$ ;
  addModelTo(models,  $M_c$ );
endfor
return models

```

Figure 4. Model learning algorithm using the k-means--based clustering strategy

The *hierarchical* clustering strategy builds clusters of contextual situations by merging the ratings associated to each possible situation. We have implemented a bottom-up approach that gradually merges highly similar pairs of ratings sets. We measure the similarity between ratings sets using the Jaccard similarity, which measures how well two sets overlap. Being R_{s_1} and R_{s_2} the ratings sets tagged with the contextual situations s_1 and s_2 respectively, the computation is defined as follows:

$$\text{sim}(R_{s_1}, R_{s_2}) = \frac{R_{s_1} \cap R_{s_2}}{R_{s_1} \cup R_{s_2}} \quad (8)$$

Figure 5 shows the pseudo-code of the full learning process using the proposed hierarchical clustering strategy. It takes as input a set of ratings for each possible target contextual situation, and the minimum Jaccard similarity required for merging two sets of ratings. The algorithm follows an iterative process that ends when no more sets can be merged in a new cluster. At each iteration, the most similar pairs of sets are merged if their similarity is larger than the specified threshold. When this happens, the original sets of ratings are removed and the merged one is added. The resulting sets of ratings correspond to the new clusters, each one referring to the contextual situations that were associated with the merged sets. As in the previous strategy, the final step consists of building a local rating prediction model for each resulting cluster of ratings.

```

Input
  ratingSets: Initial sets of ratings (one set for each possible target
  contextual situation)
   $t_m$ : Jaccard similarity threshold
Output
  models: local prediction models (one for each cluster)
  merge = true;
  while merge do
    merge = false;
    foreach  $R_s \in \text{ratingSets}$  do
       $R_{s'} = \text{findMostSimilarSet}(R_s, \text{ratingSets})$ ;
      if ( $\text{JaccardSim}(R_s, R_{s'}) > t_m$ ) then
         $R_c = \text{union}(R_s, R_{s'})$ ;
        addSetTo(ratingSets,  $R_c$ ) ;
        removeSetFrom(ratingSets,  $R_s$ );
        removeSetFrom(ratingSets,  $R_{s'}$ );
        merge = true;
      else
        addSetTo(ratingSets,  $R_s$ );
      endfor
    endwhile
    foreach  $R_c \in \text{ratingSets}$  do
       $M_c = \text{buildLocalModel}(R_c)$ ;
      addModelTo(models,  $M_c$ );
    endfor
  endwhile
  return models

```

Figure 5. Model learning algorithm using the bottom-up hierarchical clustering strategy

4 Experimental evaluation

In order to evaluate DSPF, we have considered six contextually-tagged data sets of ratings with different characteristics. **Table 1** illustrates some descriptive statistics of the data sets.

- The *Music* data set contains ratings for music tracks collected by an in-car music recommender developed by Baltrunas et al. (2011a). In this data set a user may have rated the same track more than once in different contextual situations, which are described by one condition only. Eight factors are used here, e.g., *driving style*, *mood* and *landscape*, and each factor can have different conditions (e.g., *active*, *passive*, *happy* and *sad* are possible conditions of the *mood* factor).
- The *Tourism* data set contains ratings for places of interest in the region of South Tyrol. It was collected using a mobile tourist application called South Tyrol Suggests¹, developed at the Free University of Bolzano. In this data set, ratings are acquired in contextual situations described

¹ South Tyrol Suggest is a mobile application currently available on the Google Play Store. See [https://play.google.com/store/apps/details?id=it.unibz.sts.android] (accessed June 4th, 2014).

by the conjunction of several conditions, using 14 different factors, such as weather, companion and travel goal. Possible travel goal conditions are, for instance, business, education, fun and social event. The contextual factors and conditions were identified in Baltrunas et al. (2012).

- The *Adom* data set is derived from the movie data set used by Adomavicius et al. (2005). The ratings were collected in a survey of college students who also provided information about the context of the movie-watching experience. In this data set, conditions are expressed using four contextual factors: *companion*, *day of the week*, *movie venue*, and if it was on the *opening weekend*. As in the previous data set, the contextual situations are described by several conditions (e.g., a situation could be defined by *summer*, *home* and *alone*).
- The *Comoda* movie-rating data set was collected and used by Odić et al. (2013). As in the previous data set, it contains ratings acquired in situations defined by the conjunction of several conditions, expressed using 12 different contextual factors, such as mood, time of the day, and weather.
- The *Movie* rating data set was collected for the MovieLens recommender². In this case, we used the tags provided by the users to the movies as contextual situations. We observe that here user tags provide a contextual clue of why the movie is important for the user. However, given the inherent noise of user-generated tags, we only used those tags that have a statistically significant impact on the user's rating behavior and have been used by a minimum of 5 users. As significance test we used Pearson's chi-squared that has been proven to be an effective method for identifying relevant contextual information (Odić et al. 2013). We selected the tags that are dependent on the ratings (at 99% confidence level) and we obtained 29 tags (contextual conditions). In this case, factors are Boolean, i.e., a tag can appear or not in the definition of a contextual situation.
- The *Library* book-rating data set was collected from the LibraryThing website³ and augmented with user tags. Even in this data set tags are used as contextual conditions. But, given the large number of ratings and tags, we used a stricter tag filtering criteria: in order to be selected a tag must influence the ratings (at the 99% confidence level) and must have been used by a minimum of 200 users. After the filtering process, 149 tags were kept as relevant.

Table 1. Data set's statistics ($\text{sparsity} = 1 - \frac{\#ratings}{\#users \times \#items}$)

Data set	#ratings	rating scale	#users	#items	sparsity	#factors	#conditions	#possible situations
<i>Music</i>	4013	1-5	43	139	84%	8	26	26
<i>Tourism</i>	1358	1-5	121	101	85%	14	57	375
<i>Adom</i>	1464	1-13	84	192	90,9%	4	14	134
<i>Comoda</i>	2296	1-5	121	1197	98,6%	12	49	1939
<i>Movie</i>	2190	1-10	428	1115	99,6%	-	29	114
<i>Library</i>	609K	1-10	7192	37K	99,8%	-	149	39K

We evaluated DSPF in terms of its rating and ranking prediction accuracy. We measured the accuracy of DSPF by conducting a per-user evaluation protocol known as *all-but-n* because, as noted by Shani and Gunawardana (2011), it is better than standard n-fold cross-validation for assessing the user perceived system performance, since users with many and few ratings count equally. Using this protocol, for each user, *n* ratings are randomly selected as *test* set (we used *n*=5 in *Library* and *n*=3 in the other data sets) and all the remaining ratings are used for *training*. We note that the training ratings were also used to compute the distributional *situation-to-situation* similarities. We measured the Mean

² See [http://www.movielens.org] (accessed October 27th, 2013).

³ See [http://www.librarything.com] (accessed October 27th, 2013)

Absolute Error (MAE), the Normalized Discounted Cumulative Gain (NDCG), and the catalog coverage of the context-aware predictions on the test set. All the reported results are averages of per-user evaluations, and the statistical significance of the differences between the evaluated models has been calculated using the paired Wilcoxon signed-rank test.

For all the data sets we tested the system performance only for target contextual situations that had in the training set at least three ratings. Applying the proposed evaluation method to the Library data set, around 1000 target contextual-situations were used for testing, which implies that, at each execution, 1000 local MF prediction models had to be built. In order to make testing more efficient using this data, we tested reduced subsets of target situations. Particularly, we found that using 100 contextual situations (randomly chosen) the measured system's performance was already stable.

The context-free prediction model that we used for building the local prediction models is the bias-based Matrix Factorization (MF) proposed by Koren (2010), since it is one of the best-performing ones, especially when dealing with highly sparse data. Nonetheless, any available context-free rating prediction model can be used in DSPF to build the local models. The selected MF rating prediction model generates the rating estimation for the user u and item i as the sum of the user and item biases and the dot product between their corresponding vectors of latent features (p_u) and (q_i):

$$\hat{r}_{ui} = \mu + b_u + b_i + q_i^T p_u \quad (9)$$

To learn the model parameters we minimized the *regularized error* using stochastic gradient descent, which has been proven to be an effective approach (Koren and Bell 2011).

4.1 Meta-parameter optimization

DSPF has some meta-parameters that need to be fine-tuned in order to maximize its effectiveness: (1) the global similarity threshold and (2) those controlling the learning process of the local MF models (e.g., learning rate, number of latent factors and regularization parameters).

We optimized these meta-parameters relying only on the training set data. By using the Nelder and Mead (1965) simplex search algorithm, a widely-used meta-optimizer method in the RSs field (Koenigstein et al. 2011), we searched for the best configuration of all the meta-parameters. This method begins with a set of points, each one representing a specific meta-parameters configuration and which, together, form the initial working simplex. At each iteration, the method performs a sequence of transformations of the simplex with the goal of finding a new configuration that decreases the MAE on the validation set (a subset of the training set) with respect to the previously evaluated configurations. In our experiments the algorithm converged, on average, after 15 iterations.

The validation set was generated by applying again the per-user splitting protocol mentioned above. In particular, for each user, we randomly selected n ratings among the training ratings as *validation* set (as in the first training-test split, $n=5$ in *Library* and $n=3$ in the other data sets). Moreover, the ratings selected for the validation were selected among those tagged with contextual situations that appeared at least once in the test set. That makes the target situations in the *validation* set similar to the one in the *test* set. Hence, in conclusion the meta-parameters were optimized on a set of ratings different from those in the test set but with similar target contextual situations.

We followed a two-stage process for learning the meta-parameters. Firstly, we found the optimal meta-parameters for the global, context-free MF model, and we then used that configuration also for learning the local MF models produced by DSPF. Therefore, one can expect that a better optimization of the parameters for DSPF could be even possible. Then, we run again the Nelder and Mead algorithm just for determining the optimal global similarity threshold (in the validation set) while using the Matrix Factorization meta-parameters optimized for the global model. In Section 4.6 we analyze and discuss in detail the impact of the similarity threshold to DSPF's performance.

4.2 Semantic-vector representations

In this section we compare the performance of DSPF when the different methods, described in Section 3.1, are used for obtaining the distributional semantics of contextual conditions, i.e., the semantic vectors. To simplify the performance comparison, here we only show the performance of DSPF using the *direct* situation-to-situation similarity measure. In fact, the performance results obtained by using the *aggregative* similarity measure are pretty similar and do not yield a different selection of the best semantic vector (user or item based) for each data set.

Figure 6 shows the performance comparison of the adoption of the item-based and user-based perspectives in DSPF (MAE reduction) in comparison with *MF*:

- *DSPF-MF-UB* denotes DSPF when it uses the user-based perspective for measuring the influence of conditions (defined in Eq. 2) and building the semantic vectors.
- *DSPF-MF-IB* denotes DSPF when it uses the item-based perspective, defined in Eq. 3, to build the semantic vectors.

It can be observed that there is not a clear winner, since the item-based perspective performs better in *Tourism* and *Movie*, while the user-based one achieves better results in *Adom*, *Music*, *Comoda* and *Library*. In that respect, DSPF is similar to other context-aware approaches, such as CAMF (Baltrunas et al. 2011b) and *User-Item Splitting* (Baltrunas and Ricci 2009; Zheng et al. 2013a), where different variants of the same technique can be generated depending on whether the context effect is measured with respect to the users or items (e.g., CAMF-CU and CAMF-CI). Even in these cases it is difficult to decide a priori which variant is better at the design stage.

However, it must be noted that in some data sets, such as *Adom* and *Library*, the difference in the performance of the two perspectives is not significant, which means that not always this decision is crucial for the effectiveness of DSPF. Similarly to *User-Item Splitting* (Zheng et al. 2013a), we also experimented with a DSPF variant that models the context with respect to items and users at the same time, but it did not improve the results.

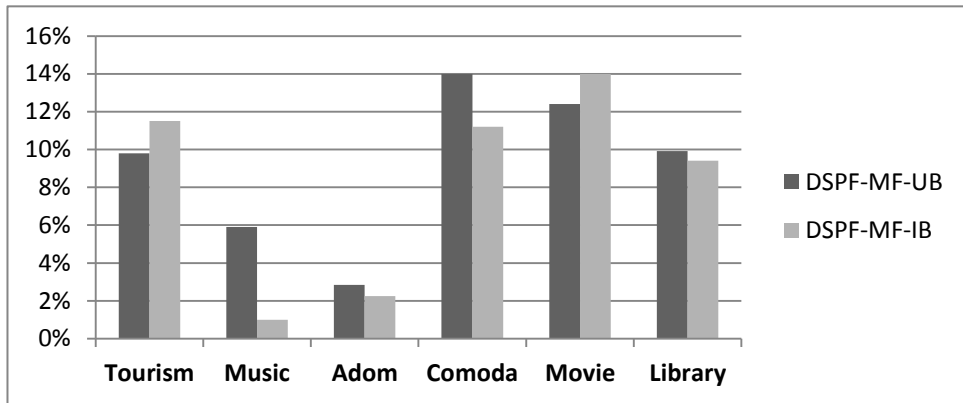


Figure 6. MAE reduction of DSPF-MF-UB and DSPF-MF-IB with respect to *MF*

In order to better understand when a variant between *DSPF-MF-UB* and *DSPF-MF-IB* is better, we have analyzed and compared the *variance* of the user- and item-based semantic vectors. We conjectured that the larger the variance of the semantic vectors is, the better the performance is. The intuition is that the larger is the variance of the entries in the semantic vector, the more the semantic vector tells about the specific impact of a contextual condition on the ratings of the available items or users.

Table 2 shows the average variance of the semantic vectors computed according to the two perspectives in the six contextually-tagged data sets. The perspective that produced the lowest MAE in each data set is in bold font. As it can be observed, our hypothesis is confirmed; the best perspective is always that producing the most diverse semantic vectors. Hence, the variance of the semantic vectors is a good indicator (easily computed at design stage) of which perspective (user- or item-based) will perform better.

Table 2. Variance of the semantic vectors. Bold font means that the perspective is better than the other

Dataset	Best variant	Variance	
		User Based	Item Based
Tourism	<i>DSPF-MF-IB</i>	0.048	0.053
Music	<i>DSPF-MF-UB</i>	0.062	0.041
Adom	<i>DSPF-MF-UB</i>	0.304	0.262
Comoda	<i>DSPF-MF-UB</i>	0.136	0.06
Movie	<i>DSPF-MF-IB</i>	5.51	7.14
Library	<i>DSPF-MF-UB</i>	0.152	0.052

4.3 Situation-to-situation similarity measures

In this section we evaluate the effectiveness of the two *situation-to-situation* similarity measures described in Section 3.2:

- *DSPF-MF-AG* corresponds to DSPF using the *aggregative* similarity defined in Eq. 5 (Codina et al. 2013a).
- *DSPF-MF-DR* corresponds to DSPF using the *direct* similarity defined in Eq. 7 (Codina et al. 2013b).

Table 3 shows the MAE of *DSPF-MF-AG* and *DSPF-MF-DR* when they use the best-performing semantic vector representation of contextual conditions in each considered data set (as shown in **Table 2**; **Error! Marcador no definido.**). In the comparison we have also included two baseline algorithms:

- A context-free Matrix Factorization model (*MF*), which generates rating predictions without taking into account the context, i.e., all predictions are based on a global prediction model learnt by using all the training ratings (whose estimation formula is defined in Eq. 9).
- *Exact Pre-filtering*, the reduction-based approach proposed by Adomavicius et al. (2005) (see Section 2) in combination with the Matrix Factorization predictive model (*Pref-MF-Exact*).

Table 3. MAE of DSPF variants using different situation-to-situation similarity measures. Results in bold are statistically significant better (95% confidence level) than the baseline algorithms. Underlined results are also significantly better than the others.

Model	Tourism	Music	Adom	Comoda	Movie	Library
<i>MF</i>	1.00	1.00	2.25	.76	1.27	1.26
<i>Pref-MF-Exact</i>	1.02	1.21	2.21	.83	1.13	1.19
<i>DSPF-MF-AG</i>	.91	.93	2.20	.74	1.10	1.16
<i>DSPF-MF-DR</i>	.88	.93	2.19	<u>.65</u>	1.10	<u>1.14</u>

We can see that DSPF with *direct* similarity (*DSPF-MF-DR*) systematically achieves better results, and outperforms the baseline algorithms and *DSPF-MF-AG*⁴. As expected, in the *Music* data set, where situations are defined only by one condition, the two similarity functions coincide and the results are equal. The differences in the performance of the two similarities are larger in the data sets with fine-grained context granularity, i.e., when the average number of conditions per situation is larger. For instance, in the *Comoda* data set, where contextual situations are defined on average by 12 conditions, *DSPF-MF-DR* improves the accuracy by 12% with respect to *DSPF-MF-AG*. In the other data sets, where the contextual situations are defined on average by 3 conditions, the improvement is not that large (3% on average).

Although it is not common to observe a large number of conditions per situation, *DSPF-MF-DR* has also a smaller time complexity compared with *DSPF-MF-AG*. In fact, in the worst case, computing the *all-pairs* aggregation of condition-to-condition similarities for two contextual situations is $O(n^2)$, whereas the cost of computing the averaged semantic vector for two situations is at most $O(n)$; n being the number of contextual factors.

4.4 Clustering strategies

In this section we illustrate the effectiveness of the clustering strategies described in Section 3.3. The two DSPF variants that use the clustering strategies are:

- *DSPF-MF-kmeans*, which is the variant using the k-means clustering method;
- *DSPF-MF-hierarchical*, the variant using the bottom-up hierarchical clustering method.

The MAE of the best-performing DSPF variants is shown in **Table 4**. In addition to the baseline algorithms, in this table we also show the MAE of the best-performing non-clustered DSPF variant, which we denote as *DSPF-MF*. We recall that the goal of the proposed clustering methods is to reduce the number of local models generated by DSPF while avoiding a possible reduction of the prediction accuracy. Even though we initially did not expect an improvement of the rating prediction accuracy, as a matter of fact, we can observe that *DSPF-MF-hierarchical* slightly outperforms *DSPF-MF* in *Tourism*. This demonstrates that merging similar sets of ratings, i.e., sets that largely overlap, in addition to the beneficial effect of reducing the number of local models, can also improve the prediction accuracy. In contrast, the k-means method (*DSPF-MF-kmeans*) always causes a loss of accuracy with respect to *DSPF-MF*. Therefore, in terms of prediction accuracy *DSPF-MF-hierarchical* is the preferable clustering method.

Table 4. MAE of DSPF using the proposed clustering strategies

Model	Tourism	Music	Adom	Comoda	Movie	Library
<i>MF</i>	1.00	1.00	2.25	.76	1.27	1.261
<i>Pref-MF-Exact</i>	1.02	1.21	2.21	.83	1.13	1.193
<i>DSPF-MF</i>	.88	.93	2.14	.65	1.10	1.14
<i>DSPF-MF-kmeans</i>	.92	.96	2.15	.69	1.10	1.15
<i>DSPF-MF-hierarchical</i>	.86	.95	2.14	.65	1.10	1.14

⁴ Note that this does not depend on the fact that the best semantic vector representation in each data set was selected (as described in the previous section) by using the *direct* measure of the similarity. In fact, the best semantic vector representation, i.e., either the user-based or the item-based (as mentioned previously) does not change if the similarity measure is changed.

¡Error! No se encuentra el origen de la referencia. shows the number of local models produced by DSPF with and without the model clustering strategies. It can be observed that, even if *DSPF-MF-kmeans* is not the best in terms of MAE performance, in its best configuration produces a smaller number of local MF models than *DSPF-MF-hierarchical*. In particular, we found that the optimal number of clusters is equal to 2 in *Tourism*, *Music* and *Comoda*, 4 in *Adom*, 5 in *Movie* and 8 in *Library*.

Table 5. Number of local MF models produced by each DSPF variant

Model	Tourism	Music	Adom	Comoda	Movie	Library
<i>DSPF-MF</i>	103	26	31	90	18	73
<i>DSPF-MF-kmeans</i>	2	2	4	2	5	8
<i>DSPF-MF-hierarchical</i>	31	23	28	51	12	60

Figure 7 illustrates how the *normalized MAE* (NMAE) of *DSPF-MF-kmeans* is affected by different values of k (i.e., number of local MF models). Besides, the optimal configuration of *DSPF-MF-hierarchical* is obtained when the Jaccard similarity of the merged sets of ratings is larger than 0.8. This clustering method, which is based on merging similar rating sets, is less effective and produces a larger number of local models. Using this method, the most significant reduction of the number of local models is produced in the *Tourism* (70% reduction with respect to *DSPF-MF*) and *Comoda* (43% reduction with respect to *DSPF-MF*) data sets.

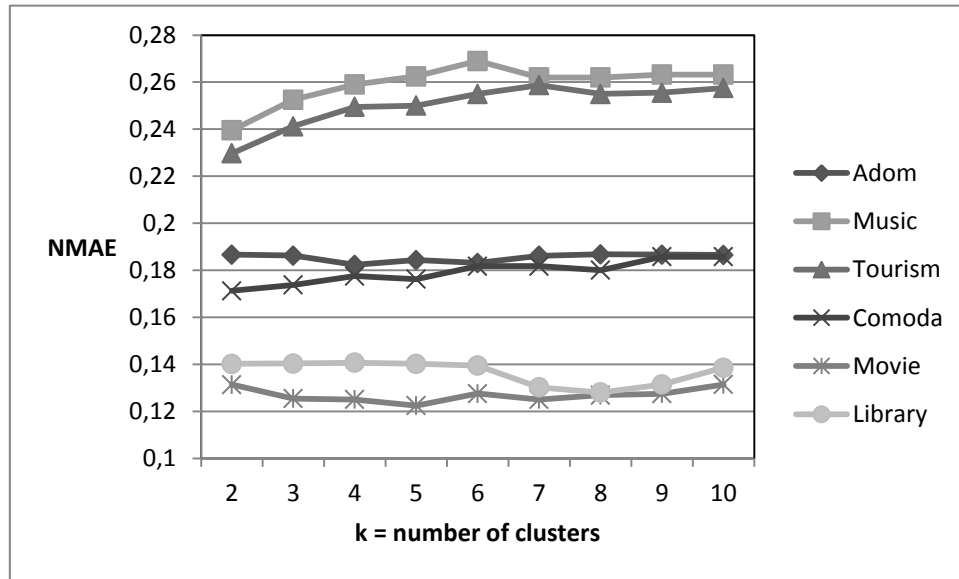


Figure 7. NMAE results of *DSPF-MF-kmeans* as a function of the number of clusters (k)

We have analyzed the time and space complexity of *DSPF-MF-kmeans* and *DSPF-MF-hierarchical* in comparison to the non-clustered DSPF and context-free MF in the *Tourism* data set. We implemented the algorithms in the Java programming language and executed the experiments on a laptop machine with two cores with clock frequency 2.4 GHz. To speed up the learning process of the considered DSPF variants we parallelized the algorithms using multi-threading. The execution time and the run-time memory consumed by each prediction model are shown in **Table 6**. As expected, *DSPF-MF-kmeans*, with k equal to 2, is clearly more efficient than *DSPF-MF* and *DSPF-MF-hierarchical*. In this particular case it is even faster than *MF* (because of the parallel processing) but *DSPF-MF-kmeans* uses slightly more run-time memory. The run-time memory usage of *DSPF-MF-kmeans* is slightly larger than *MF* because, although they use fewer training ratings, the two local MF models generated by *DSPF-MF-kmeans* have a similar number of parameters to the global model of *MF*. Comparing *DSPF-MF-kmeans* to *DSPF-MF-hierarchical*, we can observe that the *hierarchical* clustering method consumes more memory (more local models are generated), but still this is smaller than the memory used by *DSPF-MF* (84% less) and it is two times faster than *DSPF-MF*.

Table 6. Execution time and run time memory of the DSPF variants and MF in the *Tourism* data set

Model	Time (seconds)	RAM memory (MB)
<i>MF</i>	5	10
<i>DSPF-MF</i>	57	173
<i>DSPF-MF-kmeans</i>	2	16
<i>DSPF-MF-hierarchical</i>	25	28

4.5 Comparing DSPF to the state of the art

In this section we compare the performance of the best DSPF variant in each data set, which we denote here simply as *DSPF-MF*, with two state-of-the-art context-aware models: *CAMF*, a contextual modeling approach proposed by Baltrunas et al. (2011b; 2012), and *UI-Splitting*, a novel splitting-based pre-filtering method proposed by Zheng et al. (2013a) (both methods are described in more detail in Section 2). *CAMF* was proved to outperform TF approaches, especially in small-medium-size rating data sets like *Adom* (Baltrunas et al. 2011b), and *UI-Splitting* was proved to outperform *CAMF* on the *Comoda* data set (Zheng et al. 2013a).

4.5.1 Rating prediction accuracy

In **Figure 8** we show the MAE reduction of the three evaluated rating prediction models in comparison with *MF*. In this chart, we show for each prediction model its best-performing variant in each data set. For *CAMF* the best variants are the following: *CAMF-CC* in *Tourism* and *Music*, *CAMF-CI* in *Adom* and *Comoda*, and *CAMF-CU* in *Movie* and *Library*. For *DSPF-MF* the best results are obtained using *DSPF-MF-IB* in *Tourism*⁵ and *Movie*, and *DSPF-MF-UB* in *Music*, *Comoda*, *Adom* and *Library*. Finally, similarly to the experimental evaluation presented by Zheng et al. (2013a), for *UI-Splitting* the best results are obtained when using the chi-square significance test at 95% confidence level as splitting criteria, and *MF* as context-free prediction model.

⁵ In *Tourism* we used a slightly different variant that employs Singular Value Decomposition to reduce the dimensionality of the original item-based semantic vectors because, in this particular case, it improved significantly the results. (See Codina et al. (2013b) for more details about this variant).

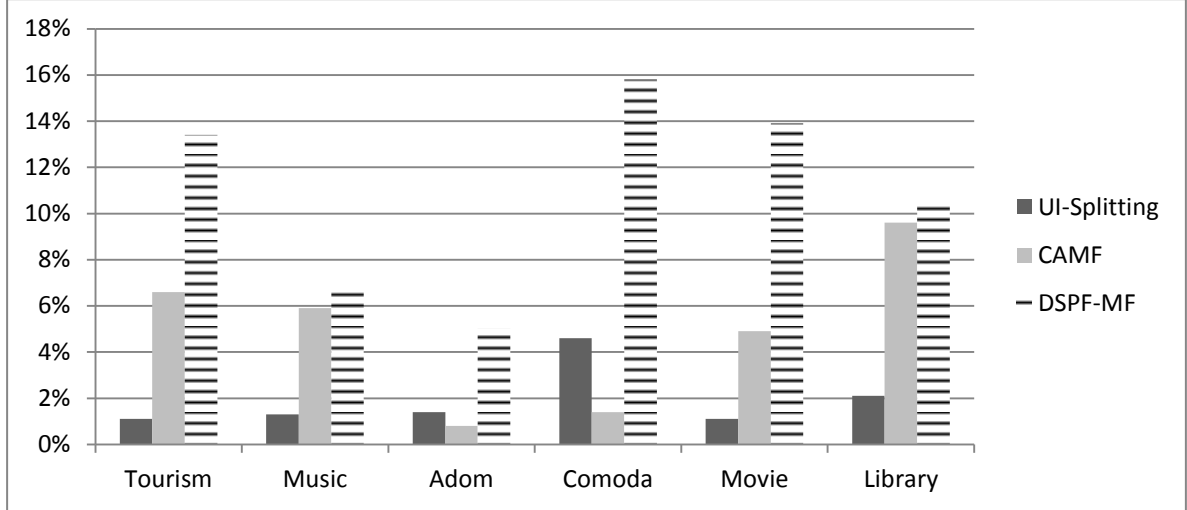


Figure 8. MAE reduction with respect to MF (the context-free baseline) of DSPF-MF compared to two state-of-the-art context-aware approaches (CAMF and UI-Splitting)

As it can be observed, the three context-aware prediction models significantly outperform *MF* in all the data sets, confirming that context-aware methods are significantly more accurate (when context matters). *UI-Splitting* outperforms *CAMF* in *Adom* and *Comoda* data sets, but in the remaining data sets *CAMF* is clearly superior. On the other hand, *DSPF-MF* outperforms *CAMF* and *UI-Splitting* in all the data sets. The improvement is evident in *Tourism* (7% gain w.r.t. *CAMF*), *Comoda* (12% gain w.r.t. *UI-Splitting*) and *Movie* (9% gain w.r.t. *CAMF*). In the next section, we will present an in-depth analysis of the system performance on the evaluated data sets, which aims to shed light on data set characteristics that we have found to be correlated to the DSPF performance and which can illustrate why DSPF performs better in some data sets and less in others.

4.5.2 Data-set characteristics influencing DSPF performance

Here we present a thorough analysis of the data set characteristics that favor and hinder the performance of DSPF, compared to the other context-aware approaches, which we have considered in this article, i.e., *CAMF* and *UI-Splitting*. We would like to explain why DSPF can offer a better performance compared to competing solutions. We conjectured that the performance differences can be related to the sparsity of the data; in fact, DSPF was originally conjectured to perform better than standard pre-filtering techniques because, when computing rating predictions in a target contextual situation, it is able to exploit ratings collected in other, similar, contextual situations, hence reducing the sparsity of the data in the target context.

To perform this analysis, we calculated different sparsity-related metrics and computed the correlation between each metric and the improvement achieved by DSPF compared to the competing methods. We wanted to check if a metric is correlated with the improvement and, therefore, if from the knowledge of the metric in a data set one can predict whether DSPF will have a better performance compared to the competing approaches.

The sparsity metrics are defined as follow:

1. **Item rating count in condition (RI)**, which measures the average item rating count in condition over all conditions, as follows:

$$RI = \frac{1}{|C|} \sum_{c \in C} \frac{1}{|I|} \sum_{i \in I} n_{ic} \quad (10)$$

where: I and C are the set of items and set of possible conditions, respectively, and n_{ic} is the number of ratings in condition c given to item i . In this metric, larger numbers mean lower sparsity, i.e., there are more ratings in each contextual condition.

2. **IB co-occurrence matrix sparsity (IMS)**, which measures the overall sparsity of the item-based co-occurrence matrix, i.e., the proportion of non-zero entries in the condition-item matrix. This metric is defined as:

$$IMS = 1 - \frac{nNonZeroEntries}{|C| \cdot |I|} \quad (11)$$

3. **UB co-occurrence matrix sparsity (UMS)**, which measures the proportion of non-zero entries in the condition-user matrix:

$$UMS = 1 - \frac{nNonZeroEntries}{|C| \cdot |U|} \quad (12)$$

Again, a larger value for these two metrics (IMS and UMS) indicates a sparser data set.

4. **Context granularity (CG)**, calculated as the average number of conditions per situation. It measures how specific the definition of context in the data set is. The larger the number of condition per situation, the finer-grained is the contextual information.

Table 7 shows the computed values of the four considered sparsity metrics. For all of them we have found a strong correlation between the sparsity metric and the improvement of the DSPF performance.

Table 7. Metrics for each data set and improvement of DSPF w.r.t. CAMF and UI-Splitting.

	Sparsity-related metrics				Improvement w.r.t.	
	RI	UMS	IMS	CG	CAMF	UI-Splitting
Tourism	1.6	0.9	0.9	3	7	12.8
Comoda	1.3	0.6	0.7	12	14	11
Movie	1.5	1	1	2	9	13
Music	1.5	0.4	0.5	1	1	5.5
Adom	2.6	0.4	0.5	3	4	3.6
Library	2.2	0.9	1	4	1	8.5

In **Table 8**, we show the correlation coefficients of each metric with the improvement of *DSPF* w.r.t. *CAMF* and *UI-Splitting*. *DSPF* improvement is measured as the percentage of the reduction of the error (MAE) with respect to the error of the competing approach. So, for instance, in the Tourism data set, *DSPF* has an error that is 7% smaller than the error of *CAMF*. One can observe that RI is negatively correlated with the improvement of *DSPF*. This means that in data sets with high RI (larger than 2 in these data sets) the gain achieved by our approach is lower, or, in other words, that *DSPF* is more effective than the other methods in the data sets where a smaller number of ratings is available for each item and contextual condition on average.

On the other hand, IMS and UMS are positively correlated with the improvement of *DSPF*, particularly when it is compared to *UI-Splitting* (0.77 and 0.82, respectively). This indicates that the higher the sparsity of the condition-item and condition-user co-occurrence matrix, the better *DSPF* performs compared to these approaches; in contrast, in data sets with low sparsity, the improvement is smaller.

Finally, also CG is positively correlated with the improvement of *DSPF*, especially when compared to *CAMF* (0.74). This shows that in the data sets with high CG, like in *Comoda*, *DSPF* will more largely outperform *CAMF*, whose performance considerably decays in such cases.

Overall, based on the above analysis of these sparsity-related measures and the relationship of these metrics with the performance of *DSPF*, we conclude that *DSPF* is especially suitable for data sets with high sparsity (i.e., low RI) and with high context granularity (i.e., high CG). Furthermore, the positive correlations of IMS and UMS metrics with *DSPF* improvement indicate that the proposed distributional semantics model is still effective even when there is low overlap between semantic vectors.

Table 8. Correlations between metrics and DSPF improvement

Metrics	Vs. CAMF	Vs. UI-Splitting
RI	-0.57	-0.63
UMS	0.22	0.82
IMS	0.24	0.77
CG	0.74	0.26

4.5.3 Ranking and Coverage

In addition, we have evaluated the performance of *DSPF* in terms of ranking precision using *Normalized Discounted Cumulative Gain* (NDCG) at the ranking cutoff of 20. In this case, we have followed the evaluation protocol known as *one plus random* (Cremonesi et al. 2010). This method consists of first selecting, for each user, a set of highly relevant items and target contextual situations. Then, for each relevant item a ranking is produced considering the context in which the user rated it, which includes the relevant item plus a set of randomly selected candidates. Finally, NDCG is measured on the base of the final position of the relevant item in the ranking: the higher, the better. We selected the relevant items by simply choosing the items rated with more than 3 stars, in 1-5 rating scale data sets, and more than 6 stars in data sets with 1-10 rating scale.

Figure 9 shows the NDCG of the considered prediction models. Again *DSPF* is the best performing model in all the data sets. Similarly to what we observed for the MAE, the largest gain with respect to the second best model is found in *Tourism* (57% gain), *Adom* (44% gain), *Comoda* (29%), and *Movie* (25%).

Ranking the considered algorithms by their NDCG does not produce in all the data sets the same order as ranking them by their MAE. For instance, *CAMF* has a lower NDCG in *Tourism* and *Music*, being clearly worse than *MF* and *IU-Splitting*. Another difference can be seen in *Comoda*, where *IU-Splitting* is slightly worse than *CAMF* in terms of NDCG but not for MAE.

We have also analyzed the catalog coverage of the prediction models at the ranking cutoff of 20. As it can be seen in **Figure 10**, apart from *Library*, where no significant differences among models are observed, in the rest of data sets, *DSPF* is the model that achieves a larger coverage.

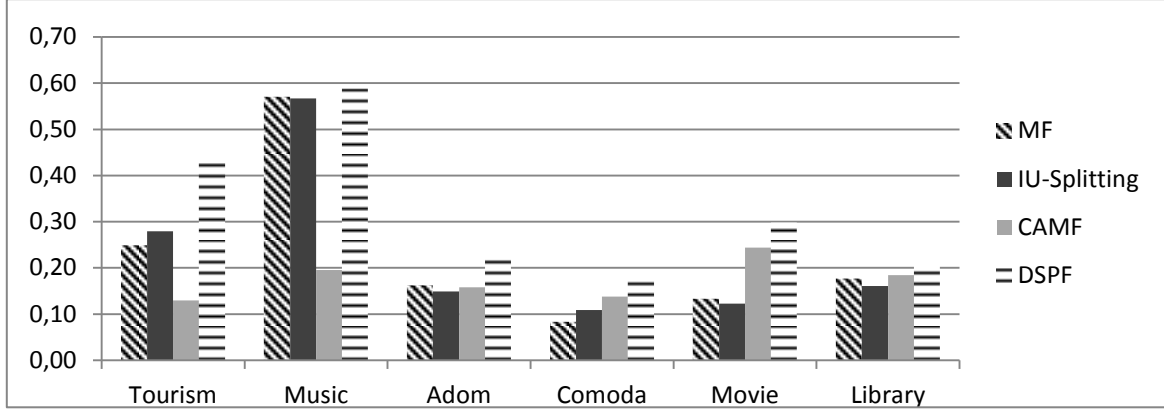


Figure 9. NDCG@20 results of the evaluated prediction models

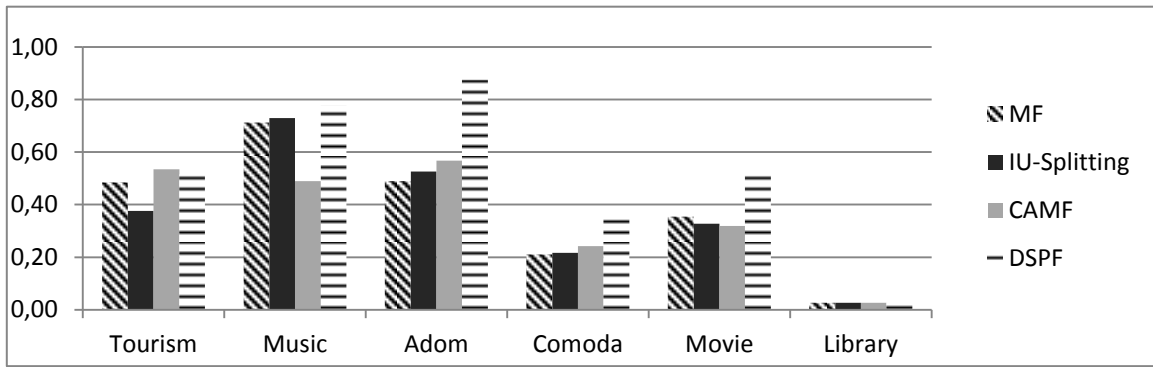


Figure 10. Coverage@20 of the evaluated prediction models

4.6 Impact of the similarity threshold

We recall that DSPF, when is requested to compute a rating prediction in a target contextual situation, instead of using a model trained exactly with the ratings acquired in that situation, it expands the training data by using ratings acquired in similar situations. In the previous sections, we have illustrated the performance of several variants of DSPF; all of them were using the optimal similarity threshold for each data set, i.e., the one that yields better prediction accuracy. As mentioned previously in Section 4.1, we experimentally found the optimal value for each data set.

We note that the impact of the similarity threshold on the prediction accuracy is similar to the effect of the *user-to-user* similarity threshold in user-based Collaborative Filtering (CF): the lower the threshold value, the larger the user's neighborhood size. In user-based CF, as in our case, it is important to find the optimal level of locality (neighborhood size), namely, that yielding the best prediction accuracy for the given data. We were not able to identify this optimal level on the base of the data set characteristics, and therefore it has to be determined through experimentation.

Table 9 shows the level of contextual expansion of the local models produced by DSPF using the optimal similarity threshold. We measure the level of contextual expansion applied by DSPF as the amount of added ratings (on average for all the possible situations) to the ratings used for building the strict local models targeted to a specific contextual situation (as those built by *Exact Pre-filtering*). Hence, we say that there is a zero expansion when no additional ratings are added (i.e., when DSPF is using *Exact Pre-filtering*) and 100% expansion when all the ratings in the data set are added independently of the target situation (i.e., when using a context-free global model). In practice, for a given target situation (s), we measure the expansion level (l_s) applied by DSPF to learn its local model

as follows (note that the final percentage is estimated by averaging the l_s of all the tested target situations):

$$l_s = \frac{\#ratings(DSPF) - \#ratings(Exact)}{\#ratings(Global) - \#ratings(Exact)} \quad (13)$$

As it can be observed in **Table 9**, the precise value of the similarity threshold itself is not providing an indication of the level of contextual expansion applied to a data set, because the expansion depends on the distribution of the *situation-to-situation* similarities, which is data-dependent. In fact, for a given similarity threshold, if the average similarity between situations is smaller, then fewer situations are aggregated. For example, although the optimal threshold in *Adom* and *Comoda* is equal (0.9), the contextual expansion is much lower in *Adom* than in *Comoda*. The reason is that in *Adom* the average situation-to-situation similarity is 0.38, whereas it is 0.86 in *Comoda*. A similar situation is found in *Movie* and *Library*, which have similar levels of contextual expansion but significantly different optimal thresholds.

Table 9. Optimal level of contextualization for each data set using DSPF

	Tourism	Music	Adom	Comoda	Movie	Library
Similarity threshold	0.3	0	0.9	0.9	0.05	0.35
Contextual expansion	31%	90%	2%	40%	10%	8%

Figure 11 shows the MAE and the expansion percentage as functions of the similarity threshold in the three movie rating data sets: *Movie*, *Comoda* and *Adom*. We show here the MAE and the contextual expansion only for positive values of the similarity threshold (from 0 to 1). We have observed that negative similarity thresholds yield bad results. In the *Movie* data (top figures) the smallest MAE (i.e., the global minimum) is obtained when the threshold is near to 0.05, which causes a level of contextual expansion above 10%. In *Comoda* (charts in the middle) the best MAE is obtained when the similarity threshold is close to 0.9, which implies a level of contextual expansion around 40%. In this case the accuracy suddenly increases when the threshold is over 0.75, and then drastically drops when the contextual expansion is lower than 30%. Finally, the charts at the bottom show the results obtained in *Adom*. Similarly to *Comoda*, the best MAE is obtained when the threshold is set to 0.9, which corresponds to a very low contextual expansion (2%). Note that the expansion decreases linearly as a function of the threshold.

Looking at the left hand side charts one can see that in all the data sets there is a global minimum where thresholds values close to it obtain the best results. We have observed that, in general, thresholds 0.1 higher or lower than the optimal value do not cause a significant loss of performance. However, when the threshold is higher/lower than 0.1 from the optimal value, the accuracy decreases significantly though the DSPF performance is still better than that of the global MF model.

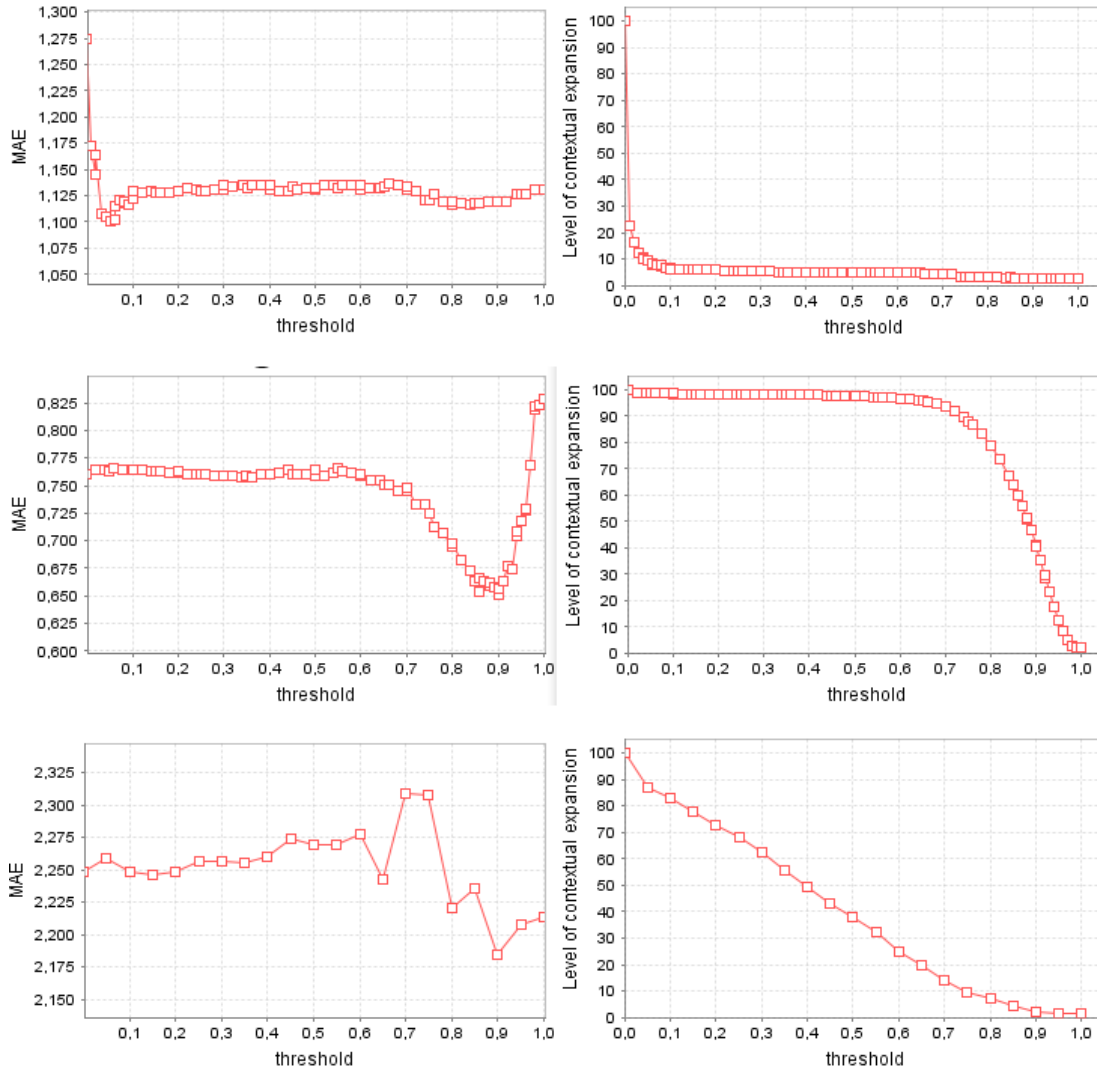


Figure 11. MAE (left line chart) and contextual expansion (right line chart) as functions of the similarity threshold (from top to down we show the results in *Movie*, *Comoda* and *Adom* data sets)

5 Conclusions and Future Work

In this paper we have described *Distributional-Semantics Pre-Filtering* (DSPF), a novel contextual, pre-filtering approach that tackles the data-sparsity problem of Context-Aware Recommender Systems (CARSs). DSPF improves state-of-the-art CARS techniques by exploiting semantic similarities between contextual situations during local context modeling. DSPF is a reduction-based approach that employs a situation-to-situation similarity function to accurately select the right level of contextualization for a given data; it builds local rating prediction models trained with ratings collected in a target contextual situation and similar situations, i.e., with a similarity larger than a data-dependent similarity threshold.

Accurate assessments of the optimal similarity threshold allow fine-tuning the rating prediction model. The situation-to-situation similarity is based on a distributional semantics approach; two situations are similar if their known conditions influence users' ratings in a similar way. Although the effectiveness of this proposed pre-filtering approach strongly depends on the size of the training data, it does not require a context taxonomy as *Generalized Pre-filtering* does. Such a taxonomy may be difficult to obtain and may not ultimately yield better rating predictions.

The experimental evaluation that we have carried out on six contextually-tagged data sets shows that DSPF outperforms state-of-the-art CARS techniques when used in combination with a bias-based MF rating prediction model. The results show that our approach obtains better results in data sets where contextual situations have a finer granularity and high data sparsity, demonstrating that DSPF is especially effective under these conditions.

Although our method uses specific solutions to improve the reliability of distributional similarities, data sparsity can still be a major issue for DSPF. We conjecture that in very sparse data sets, with small training sets, the use of an ontology-based similarity measure (based on explicit semantic features of context) could improve the precision of the similarity assessments and thus the effectiveness of DSPF. This aspect must be assessed in a future analysis.

Another potential limitation of DSPF is the method used for computing the influence of contextual conditions and building the semantic vectors, which, by design, works only with explicit rating data. Since it is based on the aggregations of rating deviations from context-free predictions, it cannot be used in data sets formed by implicit-feedback signals. Therefore, an interesting line of future research is to investigate other methods to measure the influence of contextual conditions that do not depend on explicit ratings and baseline rating predictors.

Also, the performance of DSPF could be further improved on rating data sets where the granularity of target contextual situations considerably differs. In these scenarios, by using a fine-grained tuning of the similarity threshold, i.e., a threshold that varies situation by situation, the overall performance may improve. We also conjecture that DSPF could be further improved if the meta-parameters of the Matrix Factorization (MF) local prediction models are fitted to the local training data instead of relying on the configuration optimized for the context-free MF. However, the main difficulty of this fine-tuning procedure is finding the optimal thresholds and meta-parameters without overfitting the training data, especially in small-medium rating data sets. This is a major limitation of DSPF as in other local model techniques.

Finally, we leave as future work the evaluation of a more sophisticated method for computing the semantic vector representations, measuring the influence of a contextual condition on *groups* of items or users. The groups could be based on explicit category labels or rating similarities, as in item-based and user-based CF approaches.

Appendix

The following table describes the abbreviations used to identify the considered prediction model variants.

Table 10. Description of abbreviations used to identify the evaluated model variants

Abbreviation	Description
MF	Context-free MF prediction model
Pref-MF-Exact	Exact Pre-filtering using MF local models
DSPF-MF-AG	DSPF using MF and the <i>aggregative</i> situation-to-situation similarity measure
DSPF-MF-DR	DSPF using MF and the <i>direct</i> situation-to-situation similarity measure
DSPF-MF-IB	DSPF using MF, <i>direct</i> measure and the <i>item-based</i> influence perspective
DSPF-MF-UB	DSPF using MF, <i>direct</i> measure and the <i>user-based</i> influence perspective
DSPF-MF-kmeans	DSPF with MF, best distributional similarity and k-means clustering method
DSPF-MF-hierarchical	DSPF with MF, best distributional similarity and hierarchical clustering
CAMF-CC	CAMF modeling the influence of context with respect to items' categories
CAMF-CI	CAMF modeling the influence of context with respect to items
CAMF-CU	CAMF modeling the influence of context with respect to users

Acknowledgments

The research described in this paper is partly supported by the SuperHub and the Citclops European projects (FP7-ICT-2011-7, FP7-ENV-308469), and the Universitat Politècnica de Catalunya – BarcelonaTech (UPC) under an FPI-UPC grant. The opinions expressed in this paper are those of the authors and are not necessarily those of SuperHub or Citclops projects' partners.

References

- Adomavicius, G., Mobasher, B., Ricci, F., & Tuzhilin, A. (2011). Context-aware recommender systems. *AI Magazine*, 32(3), 67–80.
- Adomavicius, G., Sankaranarayanan, R., Sen, S., & Tuzhilin, A. (2005). Incorporating contextual information in recommender systems using a multidimensional approach. *ACM Transactions on Information Systems (TOIS)*, 23(1), 103–145.
- Adomavicius, G., & Tuzhilin, A. (2011). Context-aware recommender systems. In *Recommender Systems Handbook*, pp. 217–256.
- Baltrunas, L., & Amatriain, X. (2009). Towards time-dependant recommendation based on implicit feedback. In *Proceedings of the 1st Workshop on Context-Aware Recommender Systems (CARS'09)*. October 22-25, 2009, New York City, USA.
- Baltrunas, L., Kaminskas, M., Ludwig, B., Moling, O., Ricci, F., Aydin, A., Lüke, K., & Schwaiger, R. (2011). InCarMusic: Context-aware music recommendations in a car. In *Proceedings of 12th International Conference (EC-Web'11)*, pp. 89–100. August 30 – September 1, 2011, Toulouse, France.
- Baltrunas, L., Ludwig, B., Peer, S., & Ricci, F. (2012). Context relevance assessment and exploitation in mobile recommender systems. *Personal and Ubiquitous Computing*, 16(5), pp. 507–526.

- Baltrunas, L., Ludwig, B., & Ricci, F. (2011). Matrix Factorization Techniques for Context Aware. In *Proceedings of the 5th ACM conference on Recommender systems (RecSys'11)*, pp. 301–304. October 23–27, 2011, Chicago, USA.
- Baltrunas, L., & Ricci, F. (2009). Context-dependent items generation in collaborative filtering. In *Proceedings of the 3th ACM conference on Recommender system (RecSys'09)*, pp. 245–248. October 22–25, 2009, New York City, USA.
- Baltrunas, L., & Ricci, F. (2014). Experimental evaluation of context-dependent collaborative filtering using item splitting. *User Modeling and User-Adapted Interaction*, 24(1-2), 7–34. doi:10.1007/s11257-012-9137-9
- Bazire, M., & Brezillon, P. (2005). Understanding Context Before Using It. In *Proceedings of the 5th International Conference on Modeling and Using Context(CONTEXT'05)*, LNCS vol. 3554, pp.113–192. July 5–8, 2005, Paris, France. Campos, P., Díez, F., & Cantador, I. (2014). Time-aware recommender systems: a comprehensive survey and analysis of existing evaluation protocols. *User Model User-Adapted Interaction*, 24(1-2), 67–119.
- Codina, V., Ricci, F., & Ceccaroni, L. (2013a). Exploiting the Semantic Similarity of Contextual Situations for Pre-filtering Recommendation. In S. Carberry, S. Weibelzahl, A. Micarelli, & G. Semeraro (Eds.), *Proceedings of the 21th International Conference on User Modeling, Adaptation, and Personalization (UMAP'13)*, pp. 165–177. June 10–14, Rome, Italy: Springer, Berlin Heidelberg.
- Codina, V., Ricci, F., & Ceccaroni, L. (2013b). Local Context Modeling with Semantic Pre-filtering. In *Proceedings of the 7th ACM conference on Recommender systems (RecSys'13)*, pp. 363–366. October 14–16, 2013, Hong Kong: ACM New York, NY, USA.
- Cremonesi, P., Koren, Y., & Turrin, R. (2010). Performance of recommender algorithms on top-n recommendation tasks. In *Proceedings of the 4th ACM conference on Recommender systems (RecSys'10)* pp. 39–46. September 23–26, Barcelona, Spain.
- Dourish, P. (2004). What we talk about when we talk about context. *Personal and Ubiquitous Computing*, 8(1), pp. 19–30.
- Hayes, C., & Cunningham, P. (2004). Context boosting collaborative recommendations. *Knowledge-Based Systems*, 17(2-4), 131–138.
- Hidasi, B., & Tikk, D. (2012). Fast ALS-Based Tensor Factorization for Context-Aware Recommendation. In *Proceedings of the 2012 European conference on Machine Learning and Knowledge Discovery in Databases (KDD'12)*, pp. 67–82. August 12–16, 2012 Beijing, China.
- Karatzoglou, A., Amatriain, X., Baltrunas, L., & Oliver, N. (2010). Multiverse Recommendation: N-dimensional Tensor Factorization for Context-aware Collaborative Filtering. In *Proceedings of the 4th ACM conference on Recommender systems (RecSys'10)*, pp. 79–86. September 23–26, Barcelona, Spain.
- Koenigstein, N., Dror, G., & Koren, Y. (2011). Yahoo! Music Recommendations: Modeling Music Ratings with Temporal Dynamics and Item Taxonomy. In *Proceedings of the 5th ACM conference on Recommender systems (RecSys'11)*, pp. 165–172. October 23–27, 2011, Chicago, USA.
- Koren, Y. (2010). Collaborative filtering with temporal dynamics. *Communications of the ACM*, 53(4), 89. doi:10.1145/1721654.1721677
- Koren, Y., & Bell, R. (2011). Advances in collaborative filtering. In *Recommender Systems Handbook*, pp. 145–186.
- Kurucz, M., Benczúr, A., & Csalogány, K. (2007). Methods for large scale SVD with missing values. In *Proceedings of KDD Cup and Workshop (held during KDD-2007)*, pp. 31–38. San Jose, California, USA.

- Molino, P. (2013). Semantic Models for Answer Re-ranking in Question Answering. In Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'13), Dublin, Ireland
- Musto, C., Semeraro, G., Lops, P., & de Gemmis, M. (2014). Combining Distributional Semantics and Entity Linking for Context-Aware Content-Based Recommendation. In User Modeling, Adaptation, and Personalization (UMAP'14), pp. 381–392. Aalborg, Denmark
- Nelder, J., & Mead, R. (1965). A simplex method for function minimization. *The Computer Journal*, 7(4), 308–313.
- Odić, A., Tkalčič, M., Tasic, J., & Košir, A. (2013). Predicting and detecting the relevant contextual information in a movie-recommender system. *Interacting with Computers*, 1–17.
- Panniello, U., Tuzhilin, A., & Gorgoglione, M. (2014). Comparing context-aware recommender systems in terms of accuracy and diversity: which contextual modeling, pre-filtering and post-filtering methods perform the. *User Model User-Adapted Interaction*, 24(1-2), pp. 35–65.
- Panniello, U., Tuzhilin, A., Gorgoglione, M., Palmisano, C., & Pedone, A. (2009). Experimental comparison of pre-vs. post-filtering approaches in context-aware recommender systems. In *Proceedings of the 3th ACM conference on Recommender systems (RecSys'09)*, pp. 265–268. October 22-25, 2009, New York City, USA.
- Rajaraman, A., & Ullman, J. (2012). Clustering. In *Mining of massive datasets*, pp. 239–278.
- Rendle, S., Gantner, Z., Freudenthaler, C., & Schmidt-Thieme, L. (2011). Fast context-aware recommendations with factorization machines. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval (SIGIR '11)*, pp. 635–644. July 24-27, New York, USA: ACM Press.
- Rubenstein, H., & Goodenough, J. B. (1965). Contextual Correlates of Synonymy. *Commun. ACM*, 8(10), pp. 627–633.
- Shani, G., & Gunawardana, A. (2011). Evaluating Recommendation Systems. In *Recommender Systems Handbook*, pp. 257–297.
- Shi, Y., Karatzoglou, A., Baltrunas, L., & Larson, M. (2012). TFMAP : Optimizing MAP for Top-N Context-aware Recommendation. In *Proceedings of the 35th international ACM SIGIR conference on Research and development in Information Retrieval (SIGIR '12)*, pp. 155–164. August 12-16, Portland, USA.
- Turney, PD. & Pantel, P. (2010). From Frequency to Meaning: Vector Space Models of Semantics. *J. Artif. Int. Res.*, 37(1), 141-188.
- Zheng, Y., Burke, R., & Mobasher, B. (2012). Optimal feature selection for context-aware recommendation using differential relaxation. In *RecSys'12 Workshop on Context-Aware Recommender Systems (CARS'12)*. Dublin, Ireland.
- Zheng, Y., Burke, R., & Mobasher, B. (2013a). Recommendation with differential context weighting. In *Proceedings of the 21th International Conference on User Modeling, Adaptation, and Personalization (UMAP'13)*, pp. 152–164. June 10-14, 2013, Rome, Italy.
- Zheng, Y., Burke, R., & Mobasher, B. (2013b). The Role of Emotions in Context-aware Recommendation. In *RecSys'13 Workshop on Human Decision Making in Recommender Systems*, pp. 21–28. October 14-16, 2013, Hong Kong: ACM New York, NY, USA.

Author Biographies

(1) Dr. Victor Codina:

Technical University of Catalonia, Software department, Girona 1-3, K2M 201, 08034 Barcelona, Spain

Dr. Victor Codina is currently an associate researcher at Barcelona Digital Technology Center, Spain. His current research interests include context-aware recommender systems, mobile systems and applications of machine learning. He received his BSc in Computer Science from the Technical University of Catalonia in January 2008, and he completed his PhD in Artificial Intelligence at the same institution in June 2014. This paper summarizes the main contributions of his thesis work on context-aware recommender systems.

(2) Prof. Francesco Ricci:

Faculty of Computer Science, Free University of Bozen-Bolzano, Piazza Domenicani 3, 39100 Bozen-Bolzano, Italy.

Francesco Ricci is a professor of computer science at the Free University of Bozen-Bolzano, Italy. His current research interests include recommender systems, intelligent interfaces, mobile systems, machine learning, case-based reasoning, and the applications of ICT to health and tourism. He has published more than one hundred of academic papers on these topics. He is the editor in chief of the Journal of Information Technology & Tourism and on the editorial board of User Modeling and User Adapted Interaction.

(3) Dr. Luigi Ceccaroni:

1000001 Labs, c. Alzina 52, 08024 Barcelona, Spain

Dr. Luigi Ceccaroni is founder and research lead of 1000001 Labs. He obtained a BSc degree in Environmental Sciences, an MSc degree in Information-Technology Languages and Systems; and completed his PhD in Artificial Intelligence at the Technical University of Catalonia in December 2001. His main research interests combine: biochemical-hazards forecasting in oceans, seas and coasts; citizen science; decision support systems participatory science; ontologies; recommendation systems; semantic tools; the semantic Web; personalization; and application of artificial intelligence to healthcare and environmental sciences. Dr. Ceccaroni coordinated several large European research projects in the Environment, and Information and Communication Technologies areas.