SmarterDeals: A Context-aware Deal Recommendation System based on the SmarterContext Engine

Sahar Ebrahimi[1], Norha M. Villegas[1,2,3], Hausi A. Müller[1,2], and Alex Thomo[1]

[1]University of Victoria, Victoria, British Columbia, Canada
[2]IBM Canada CAS Research, Markham, Ontario, Canada
[3]Icesi University, Cali, Valle del Cauca, Colombia

Abstract

Daily-deal applications are popular implementations of on-line advertising strategies that offer products and services to users based on their personal profiles. The current implementations are effective but can frustrate users with irrelevant deals due to stale profiles. To exploit these applications fully, deals must become smarter and context-aware. This paper presents SmarterDeals, our deal recommendation system that exploits users' changing personal context information to deliver highly relevant offers. SmarterDeals relies on recommendation algorithms based on collaborative filtering, and SmarterContext, our adaptive context management framework. SmarterContext provides SmarterDeals with up-to-date information about users' locations and product preferences gathered from their past and present web interactions. For many deal categories the accuracy of SmarterDeals is between 3% and 8% better than the approaches we used as baselines. For some categories, and in terms of multiplicative relative performance, SmarterDeals outperforms related approaches by as much as 173.4%, and 37.5% on average.

1 Introduction

Daily-deal applications are marketing strategies widely used by businesses to advertise products and services using discount coupons. To receive daily-deal coupons, users must create their personal profile by registering their personal information, and selecting relevant product or service categories from the list of available options. Using this information, daily-deal applications send offers that match the user's profile. These solutions use different communication channels such as e-mail, short message services, social networks, mobile applications, and web sites.

Groupon,[1] with approximately 51 million subscribers in 563 cities worldwide, is the most popular provider of coupons on-line [21]. Value creation in Groupon's business model is based on the negotiation of attractive discounts with popular businesses, and the delivery of these discount offers to its subscribers via e-mail. Despite the evident success of daily-deal businesses such as Groupon, their value creation and revenue generation can be considerably more effective by improving the relevance of delivered coupons to users [11]. The delivery of irrelevant offers is a consequence of a lack of knowledge about user situations. To tackle this problem, daily-deal platforms must become smarter, that is, become context-aware.

Context can be defined as any information useful to characterize entities that affect the situation of users [1]. Moreover, context informa-

[1]http://www.groupon.com

tion is highly dynamic since the situations of users change over time [23]. Therefore, to deliver relevant deals to users, the analysis of personal information provided by the user during the registration process is neither enough nor effective. On the one hand, product categories that were relevant to the user at registration-time may become irrelevant over time. On the other hand, demanding from users the manual registration of changes in this information is inconvenient. Dealing with the dynamic nature of context information, in an instinctive way for the user, is a big challenge for businesses to deliver products and services based on the understanding of personal concerns.

To support context-aware user-centric web applications, we implemented the SmarterContext framework, a dynamic context management infrastructure that monitors the interactions of users with web entities (e.g., products offered in an on-line catalog) to gather relevant context [25]. The information gathered by the SmarterContext framework about a particular user is stored into a persistent repository named *personal context sphere (PCS)*. SmarterContext reasons about this information to provide context-aware applications such as daily-deal applications with information useful to understand users' situations and preferences.

This paper introduces SmarterDeals, our deal recommendation system that is aware of changing personal context information to deliver coupons highly relevant to users. To improve the relevance of coupons, SmarterDeals implements an algorithm based on collaborative filtering (CF) that analyzes similarities between users and obtain a set of potential relevant deal categories based on context information (i.e., product and service preferences) of similar users. Then, the accuracy of this list of potential relevant deal categories is improved by correlating these categories with context information about the user's product and service preferences gathered by SmarterContext. Finally, location context is used to filter the recommended categories before their delivery to the user. Thus, our recommendation algorithm contributes to the improvement of the accuracy of recommendations using contextual information provided by our SmarterContext solution.

Our contribution in this paper is a new recommendation algorithm based on traditional user-based CF techniques [3] and the approach used by Bell and Koren [15] in the Netflix competition [7]. In contrast to these approaches, our algorithm uses context information from different users to improve the accuracy of daily-deal recommendations. Furthermore, since SmarterContext infers implicit context facts from explicit context observations, our recommendation algorithm takes into account also categories that could potentially be relevant to the user, even when the user has not marked them explicitly as relevant.

To validate our approach with real data, we used the Yelp academic data set [26] to simulate the context information gathered by SmarterContext. For this, we transformed the 271,418 product and service category ratings and the 65,411 real users of the Yelp data set into Resource Description Framework (RDF) data compliant with context models in SmarterContext. The validation results demonstrate the suitability of our context-aware recommendation approach. For many deal categories the accuracy of SmarterDeals is between 3% and 8% better than the approaches used as baselines. For some categories, and in terms of multiplicative relative performance, SmarterDeals outperforms related approaches by as much as 173.4%, and 37.5% on average.

The remaining sections of this paper are organized as follows. Section 2 provides an overview of our SmarterContext solution. Section 3 explains our case study and the simulation of context information provided by SmarterContext using the Yelp data set. Section 4 introduces the two approaches used as baselines to evaluate our approach. Section 5 explains our recommendation algorithm. Section 6 presents the validation results. Section 7 discusses related work. Finally, Section 8 concludes the paper.

2 SmarterContext

The main components of SmarterContext are (i) the SmarterContext ontology, (ii) the service-oriented software infrastructure, and (iii) the users' PCSs. The SmarterContext ontology, which includes several vocabularies, supports context representation and reasoning [24]. The service-oriented infrastructure provides the software components required to manage the context information lifecycle: context gathering, processing, provisioning, and disposal [25]. PCSs are repositories that store the personal context data of SmarterContext users in the form of RDF statements [23].

2.1 Context Representation and Reasoning

Context representation and reasoning in SmarterContext is supported by our SmarterContext ontology. This ontology exploits RDF [8] and OWL-Lite [22] to represent context types and the relationships among them explicitly, and to infer implicit context facts from these context relationships.

We designed SmarterContext as a modular ontology that supports vertical and horizontal extensibility [24]. Its foundational module, *general context (GC)*, enables context representation and reasoning for any problem domain. The application of the SmarterContext ontology to a particular domain may imply the definition of several hierarchical levels. For example, to support context-awareness in the personal web (PW) [25] we derived from GC the *personal web context (PWC)* module. The PWC module supports context representation and reasoning in any problem domain of the PW. Similarly, to represent and reason about context information in on-line shopping applications, for example in the case study presented in this paper, we derived the *shopping* module from PWC. The namespaces of the main modules of the SmarterContext ontology are *gc*,[2] *pwc*,[3] and *shopping*.[4] Table 1 presents the context entity types and context relationships (i.e., object properties) that are relevant to the SmarterDeals case study.

[2]http://smartercontext.org/vocabularies/gc/v5.0/gc.owl#

[3]http://smartercontext.org/vocabularies/pwc/v5.0/pwc.owl#

[4]http://smartercontext.org/vocabularies/shopping/v5.0/shopping.owl#

The SmarterContext engine processes context using RDFS and OWL-Lite assertions, as well as user-defined rules at different levels of the ontology. The context inference engine is provided by Jena.[5] User-defined rules used by the SmarterContext reasoning engine are based on RDF-S and use the triple representation of RDF descriptions.[6]

2.2 Context Management

A prerequisite for SmarterContext to manage a user's context information is the creation of the user's PCS. For this, users register themselves into the SmarterContext framework by providing some personal context information such as age, gender, preferred location, and preferred payment methods. They may decide to register also web sites or applications compliant with SmarterContext. That is, applications instrumented to interchange context information with the SmarterContext infrastructure.

One of the most relevant mechanisms for gathering context information in SmarterContext is based on the monitoring of users' web interactions. In our smarter commerce case study [25], simple RDF sensors deployed at the context provider side (e.g., an on-line shopping application) keep track of "likes", "wishes", "rankings", and "purchases" interactions performed by the user. From these interactions SmarterContext understands what product and service categories are interesting to the user. Figure 1 represents the RDF graph of a user's ranking interaction gathered by SmarterContext. This graph is composed of two triples. The first one indicates that user Norha, represented by the subject labeled as *norha.rdf#norha*, ranked the product category represented by the object labeled as *deals:LatinRestaurant*. Predicate *pwc:ranked* is the context relationship used to represent ranking (also known as rating) interactions. In the second triple, node *deals:LatinRestaurant* acts as the subject, and the literal with value 4 acts as the object that represents the value given by the user to this category. RDF graphs, serialized as RDF/XML messages, provide the

[5]http://jena.apache.org

[6]http://smartercontext.org/examples/rulesv5.0.rules

Table 1: Context entities and context relationships of the SmarterContext ontology that are relevant to the SmarterDeals application presented in this paper. Column *Description* indicates whether the context type corresponds to a context entity or an object property (context relationship).

Context Type (Class)	Description	Supertype
gc:ContextEntity	Entity. The superclass of any context type.	owl:Thing
gc:LocationContext	Entity. The place of settlement or activity of an object.	gc:ContextEntity
gc:GeoLocation	Entity. The latitude and altitude that describe a physical location.	gc:PhysicalLocation
pwc:PWESite	Entity. Any web site compliant with SmarterContext - e.g., an on-line store.	pwc:WebResource
pwc:User	Entity. Any person registered into SmarterContext.	gc:HumanEntity
shopping:Product ServiceCategory	Entity. A product or service category offered or advertised on-line - e.g., American restaurants.	pwc:WebEntity
gc:locatedIn	Object property. Its value represents the location where the subject (an IndividualContext or LocationContext entity) is located in.	gc:location Relationship
pwc:hasIntegrated	Object property. Its value represents a context entity that has been integrated into a PCS.	gc:association Relationship
pwc:preferredLocation	Object property. Its value defines the preferred location of a user.	gc:location Relationship
pwc:ranked	Object property. An interaction to denote that the user has given a ranking value to a context entity represented by the object.	pwc:userInteraction
shopping:related ProductOrService	Object property. Denotes that two product or service categories are related to each other.	gc:association Relationship

interoperability mechanism to exchange context information between context consumers and providers with the SmarterContext engine.

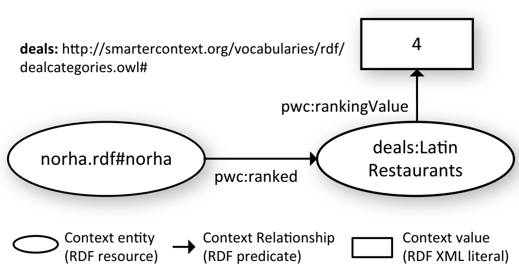

Figure 1: The RDF graph representation of a user's ranking interaction.

3 SmarterDeals: Our Case Study

The case study is inspired by Groupon. Our goal is to demonstrate how the accuracy the relevance of coupons delivered by deal applications such as Groupon can be improved considerably by taking into account the user's personal context information.

3.1 Daily-Deals with Groupon

Groupon delivers daily coupons based on the personal information registered by the user during the sign-up process. This information corresponds to the user's favorite locations, gender, age, and favorite deal categories. Users can edit their personal information at any time through GROUPON's web or mobile applications. GROUPON allows users to share deal recommendations by e-mail, as well as to broadcast them to their social networks.

Even though GROUPON has been effective in the accomplishment of its business goal, the current implementation of its daily-deal application can frustrate users with irrelevant deals due to stale profiles. GROUPON delivers offers of products and services by taking into account only the information registered by the user during the sign-up process. Nevertheless, most of this information gets out-of-date quickly. In daily-deal applications location and preferred deal categories are types of highly dynamic context information. With respect to the user's preferred locations, GROUPON delivers deals related to the whole set of registered locations, which can include different cities. This practice lacks location-aware filtering mechanisms thus compromising the effectiveness of delivered coupons. For example, for users who are frequent travelers, daily deals must be delivered taking into account the users' current location, even if this location is not part of the user's list of favorite locations. Regarding the list of preferred deal categories, sending coupons using only the information registered during the sign-up process is ineffective, since the relevance of deal categories is highly dependent on changing context information such as location or time context. Hence, categories that could have been relevant yesterday, may no longer be relevant today nor in the near future. For example, a user whose kids are children may be interested in children's books today, but probably not in a few years from now. In GROUPON, users must change their personal information and preferences to preserve the relevance of received offers.

3.2 Daily-Deals with SmarterDeals

Figure 2 presents an overview of SMARTERDEALS, our approach to personal context-aware daily-deal recommendations. Our application is composed of two main artifacts: the *recommendation engine* and the *filtering and personalization module*. For SMARTERCONTEXT to provide SMARTERDEALS with personal context information, users must integrate SMARTERDEALS into their PCSs. After completing this prerequisite, our SMARTERCONTEXT framework provides SMARTERDEALS with personal context information about the user's product and service preferences, and locations.

Our recommendation engine exploits context information about the user's product or service preferences to predict daily-deal categories relevant to the user as follows. In the first step our recommendation algorithm correlates similarities among users based on the *Pearson Correlation Coefficient* (PCC) [20]. PCC ranges from -1, which indicates a negative correlation, to +1, which indicates a positive correlation between two users. A value of 0 indicates no correlation. Users who have a PCC equal to or greater than 0.7 are considered similar enough in our approach.[7] In the second step, our algorithm aggregates the ratings of product or service categories given by users similar to the user who will receive the recommendation, to predict the rating of the corresponding product or service category. SMARTERDEALS decides whether a product or service category is relevant to a user using the predicted rating of the corresponding category. In this case study, as in the Yelp data set, ratings range from 1 to 5, where 1 indicates the lowest level of relevance and 5 the highest. Our algorithm recommends categories with predicted ratings equal to or greater than 4. Once the recommendation engine has predicted the list of potential relevant categories, in the third step SMARTERDEALS uses the GROUPON's application programming interface (API) to provide users with business daily-deal offers filtered by the user's locations and preferences.

SMARTERDEALS, supported by SMARTERCONTEXT, improves the relevance of daily product and service recommendations delivered to users by exploiting:

- up-to-date product and service categories gathered from web interactions performed by the users throughout their web experiences, and

[7] Any value between 0.5 and 1 can be used to represent strong association between two variables. We considered 0.7 as a suitable measure for this case study. https://statistics.laerd.com/statistical-guides/pearson-correlation-coefficient-statistical-guide.php

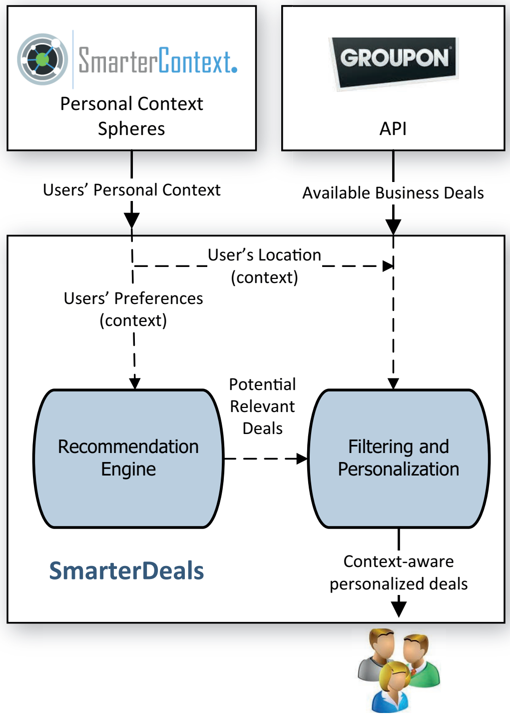

Figure 2: SmarterDeals

- up-to-date information about current and preferred users' locations.

3.3 Simulating Context Data with the Yelp Data Set

PCSs store personal context data in the form of RDF graphs. To validate our approach to smarter deal recommendations with real data, we used the Yelp academic data set [26].

The Yelp data set includes 271,418 ratings that 65,411 real users have given to 6,900 local businesses. Ratings range from 1 to 5, and local businesses have been tagged with one or many of the 365 product and service categories defined by Yelp. We used the information of the Yelp data set to create 65,411 RDF graphs. Each graph simulates the context sphere of a user. Since in the Yelp data set users have rated businesses instead of products or services, we obtained favorite users' product and service categories from the categories associated to the businesses rated by these users. The Yelp businesses are located in 67 different cities across North America. Users' locations were obtained from the locations of the businesses reviewed by each user. Nevertheless, the set of cities related to businesses in Groupon differs from the set of cities related to businesses in Yelp. Thus, to use Groupon's API in SmarterDeals, for each location associated to a user in Yelp, we considered all of the nearby locations in Groupon. Therefore, if Groupon is unavailable in the user's relevant locations, we can still deliver Groupon deals related to nearby locations. For this, we extended the SmarterContext vocabulary that defines geographical locations to include the locations used by Groupon and Yelp.[8]

Groupon defines 633 product and service categories classified into 18 general categories whereas, the Yelp data set contains 433 product or service categories with no hierarchies. 284 of these categories are exactly equal to those in Groupon. Thus, to recommend deals based on the product and service categories defined by Groupon, we mapped manually the remaining 149 Yelp categories into similar Groupon categories. SmarterContext classifies products and services using the Google product taxonomy [12]. For this case study, we extended this taxonomy by creating a complementary ontology, the *deals* ontology,[9] from the set of Yelp product and service categories mapped to Groupon deal categories.

Since some Yelp's users have a very small number of ratings, we reduced the Yelp data set by considering only users who have at least 20 ratings. The reduced data set has 58,069 ratings given to 313 product or service categories by 1,683 users. These 313 categories, now mapped into Groupon categories, belong to 17 parent categories. This set of 17 parent categories was further reduced to 14 parent categories by eliminating categories with less than 50 ratings considered as not statistically significant.

[8] The SmarterContext *geo* vocabulary is available at http://smartercontext.org/vocabularies/rdf/geo.rdf

[9] The *deals* ontology is available at http://smartercontext.org/vocabularies/rdf/dealcategories.owl.

4 Collaborative Filtering

Collaborative filtering is a recommendation technique in which users receive recommendations of items that have been positively rated by other people with similar preferences [19]. The goal of recommendation methods based on collaborative filtering is to predict the unknown rating that a user may give to an item by considering the ratings given to that item by other users.

4.1 Calculating Similarities

Our approach exploits similarities among users to recommend product and service categories. The similarity level between a pair of users is calculated based on similar ratings and preferences. Collaborative filtering techniques based on similarities among users are known as *user-based collaborative filtering* techniques [9, 18, 20]. The similarity between a pair of users can be calculated using different similarity measures such as *correlation*-based (cf. Equation (1)) [20, 18] and *cosine*-based (cf. Equation (2)) [9, 19].

In these equations r_{ui} and $r_{u'i}$ are the ratings given to item i by users u and u', respectively. $I_{uu'}$ is the set of items co-rated by both users u and u'. In Equation (2) each user is defined as a vector of ratings. In this case the similarity between two users is measured by computing the cosine of the angle between the corresponding two vectors. Once similarities between users are calculated, we can consider the top-N similar users, or users having similarities greater than a desired threshold (i.e., 0.7 for this case study) as the users most similar to a given user.

4.2 Rating Prediction

The most important step in collaborative filtering is the prediction of the rating that a particular user would give to an item. A common approach to predict the value of an unknown rating r_{ui} given by user u to item i is the use of an aggregate function of the ratings given to item i by users similar to u. Equation (3) presents three different aggregate functions commonly used for rating prediction in collaborative filtering systems [9, 18, 19]. Equation (3)(a) is known as *simple average*, Equation (3)(b) as *weighted sum*, and Equation (3)(c) as *adjusted weighted sum*. Multiplier k is used as a normalizing factor and usually is defined as $k = 1/\sum_{u' \in \hat{U}} |sim(u,u^{'})|$, with $\hat{U}$ as the set of users similar to user u. The average rating of user u in Equation (3)(c) is defined as $\bar{r}_u = (1/|S_u|)\sum_{i \in S_u} r_{ui}$, with S_u as the set of all items rated by user u [9].

4.3 Baseline Approaches

To evaluate our approach we used two well known recommendation methods as baselines. The first baseline approach is the traditional user-based collaborative filtering method [18, 20]. Consider the user-ratings matrix presented in Fig. 3. Rows correspond to users and columns to items (e.g., product categories). Consequently, each cell represents the rating given by a particular user to the corresponding product category. The goal is to predict the unknown rating given by the active user u to product category i. That is, to calculate $\hat{r}_{ui}$ represented by the highlighted cell. The first step is to find the users that are similar to the active user. Similarity between users is calculated using the Pearson Correlation Coefficient (PCC) method (cf. Equation (1)). Using a threshold of 0.7, we found u_1 and u_3 as the users similar to the active user. To predict the unknown rating this approach uses weighted sum as the aggregation function (cf. Equation (3)(b)). This function aggregates the ratings given to product category i by the users similar to the active user, ratings r_{1i} and r_{3i}.

The second baseline approach is the one used by Koren and Bell (winners of the Netflix prize) [15, 16]. They argue that collaborative filtering data are affected by systematic tendencies for some users to give higher ratings than others, and for some items to be better rated than others. To tackle these effects, they adjust collaborative filtering using *baseline predictors*. A baseline predictor for an unknown rating $r_{u,i}$ denoted by b_{ui} encapsulates these effects that do not involve user-item interaction, and is calculated as:

$$sim(u,u') = \frac{\sum\limits_{i \in I_{uu'}} (r_{ui} - \bar{r}_u)(r_{u'i} - \bar{r}_{u'})}{\sqrt{\sum\limits_{i \in I_{uu'}} (r_{ui} - \bar{r}_u)^2 \sum\limits_{i \in I_{uu'}} (r_{u'i} - \bar{r}_{u'})^2}}$$

Equation 1. Correlation-based user similarity

$$sim(u,u') = cos(\vec{u}, \vec{u'}) = \frac{\vec{u}.\vec{u'}}{||\vec{u}||_2 \times ||\vec{u'}||_2} = \frac{\sum\limits_{i \in I_{uu'}} r_{ui} r_{u'i}}{\sqrt{\sum\limits_{i \in I_{uu'}} r_{ui}^2} \sqrt{\sum\limits_{i \in I_{uu'}} r_{u'i}^2}}$$

Equation 2. Cosine-based user similarity

$$r_{ui} = \frac{1}{N} \sum_{u' \in \hat{U}} r_{u'i}$$

(a) Simple average

$$r_{ui} = k \sum_{u' \in \hat{U}} sim(u,u') \times r_{u'i}$$

(b) Weighted sum

$$r_{ui} = \bar{r}_u + k \sum_{u' \in \hat{U}} sim(u,u') \times (r_{u'i} - \bar{r}_{u'})$$

(c) Adjusted weighted sum

Equation 3. Rating prediction

$$b_{ui} = \mu + b_u + b_i$$

Equation 4. Baseline predictor

μ is the average rating over all of the items rated by all of the users, and b_u and b_i indicate the observed deviations of user u and item i from the average, respectively:

$$b_i = \frac{\sum_{u \in R(i)} (r_{ui} - \mu)}{\lambda_2 + |R(i)|}$$

Equation 5. Item deviation

$$b_u = \frac{\sum_{i \in R(u)} (r_{ui} - \mu - b_i)}{\lambda_3 + |R(u)|}$$

Equation 6. User deviation

Detailed explanations regarding the calculation of b_i and b_u are provided in [16]. Koren and Bell demonstrated that 25 and 10 are typical values for λ_2 and λ_3, respectively. We used the same values for the Yelp data set. Unknown ratings are predicted using similarity measures among items. Since our approach is based on similarity among users, we modified Equation

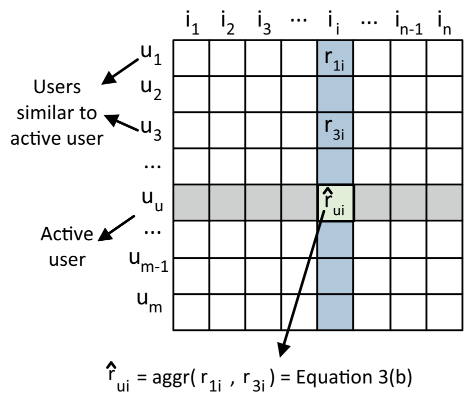

Figure 3: Traditional user-based collaborative filtering approach

5.17 of [16] slightly to calculate unknown ratings as follows:

$$\hat{r}_{ui} = b_{ui} + \frac{\sum_{v \in S^t(u)} S_{uv}(r_{vi} - b_{vi})}{\sum_{v \in S^t(u)} S_{uv}}$$

Equation 7. User-based unknown rating

v refers to any user with a similarity with user u higher than the threshold t (i.e., 0.7), and $S^t(u)$ denotes the set of all users similar to u. We call this approach *Adjusted Collaborative Filtering (ACF)*

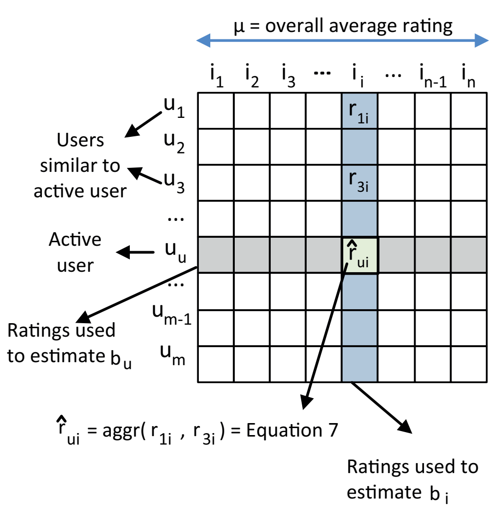

Figure 4: ACF approach

Figure 4 presents an abstraction of the input for the application of ACF with user-based similarities. The ratings used to calculate b_i (cf. Equation (5)) correspond to the ratings given to product i. Similarly, the ratings used to calculate b_u (cf. Equation (6)) correspond to the ratings given by user u. The unknown rating $\hat{r}_{ui}$ is calculated using Equation (7).

5 Our Recommendation Algorithm

Our recommendation algorithm is a variation of ACF, that applies to hierarchies of item categories, where rating prediction is based on item categories that belong to the same parent category. That is, we calculate similarities among items whose immediate category belongs to the same immediate super-category. We hypothesized that by partitioning items according to their parent categories it is possible to improve the accuracy of recommendations. This is because aspects such as the sensitivity of the parent category may affect the ratings given by users. For example, users may rate products or services related to “Health & Fitness” more carefully than those classified as “Nightlife”. In this case study, we partitioned the deal categories to be recommended according to the parent categories defined by Groupon.

Figure 5 illustrates our approach. Instead of having an overall average rating μ, we calculate average ratings for each parent category P. Moreover, we compute the user’s observed deviation b_u for each parent category P. Thus, instead of having a unique b_u per user, we have a b_{uk} for each parent category related to the items rated by the corresponding user:

$$b_{uk} = \frac{\sum_{i \in R_k(u)}(r_{ui} - \mu_k - b_i)}{\lambda_3 + |R_k(u)|}$$

Equation 8. User deviation for P_k

$R_k(u)$ denotes the set of items rated by user u into parent category P_k, and μ_k denotes the average of ratings given to items classified into parent category P_k (cf. the section of the matrix that corresponds to the horizontal arrow in the upper part of Fig. 5).

Similarity between users is calculated using PCC with 0.7 as the threshold (cf. Equation

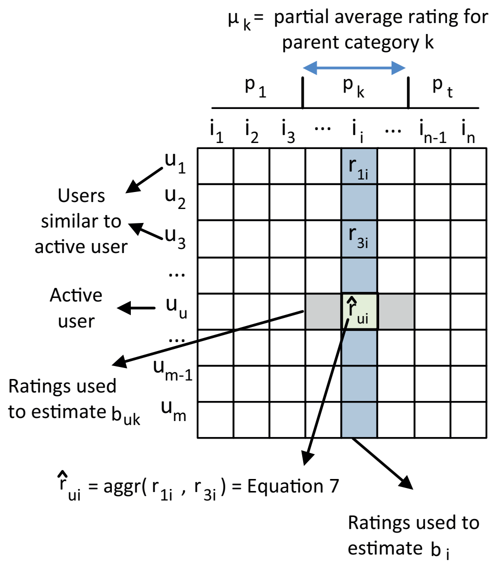

Figure 5: Our recommendation approach

(1)) and b_i is calculated as in ACF. Finally, the unknown rating $\hat{r}_{ui}$ given by the active user u to item i is predicted with our proposed aggregate function (cf. Equation (7)), with $b_{ui} = \mu_k + b_{uk} + b_i$, with $i \in P_k$ (cf. Fig. 5).

6 Validation

To evaluate our approach, for each existing r_{ui} rating, we created a new version of the Yelp dataset, $Y_{-r_{ui}}$, with r_{ui} removed. Then, we applied our approach and the two baseline approaches to predict the deleted rating r_{ui}. The predicted rating $\hat{r}_{ui}$ may or may not be the same as r_{ui}. In general, we are interested to see whether the error $e_{ui} = r_{ui} - \hat{r}_{ui}$ is small. We repeat this procedure for each existing rating r_{ui}, i.e. we ran the algorithms as many times as there are ratings in the dataset. This exhaustive evaluation gives us a precise picture of the quality of each approach. To measure the effectiveness of each approach we used *root mean squared error (RMSE)* as defined in Equation (9), a widely accepted metric to assess the accuracy of the values predicted by a model or an estimator with respect to the values actually observed [13].

$$RMSE = \sqrt{\sum_{(u,i) \in TestSet} \frac{(e_{ui})^2}{|TestSet|}}$$

Equation 9. Root mean squared error

Since our approach relies on partitions based on parent categories, we designed our tests as follows: First, we predict the rates for every single parent category independently. Second, we compute the RMSE for each parent category. Third, we apply the procedure for every recommendation technique to predict deleted ratings.

Table 2 presents our validation results. Column *Parent Category* contains the 14 GROUPON's parent categories included in the reduced Yelp data set. Columns *Classic CF (C)*, *ACF (A)*, and SMARTERDEALS *(S)* present the RMSE for the two baselines—the traditional user-based collaborative filtering method and ACF approach, and our context-driven approach, respectively. Column *(AiC)*, calculated as a percentage $(C - A)/C$, corresponds to the improvement of A over C. That is how better is the error measure (RMSE) of ACF with respect to the traditional user-based recommendation method. Similarly, Column *(SiC)* represents the improvement of SMARTERDEALS (S) over the traditional method (C), and is calculated as a percentage $(C - S)/C$. Column *SiC-AiC* compares the improvement of S over C with respect to the improvement of A over C. Finally, Column *Relative Performance (RP)*, calculated as a percentage $(SiC - AiC)/SiC$, represents the relative improvement of our approach with respect to ACF approach. Figure 6 presents the improvement in terms of accuracy of ACF approach (AiC), and our approach (SiC), with respect to the traditional user-based collaborative filtering method (C). Figure 7 presents the relative performance of SMARTERDEALS with respect to ACF approach.

Our approach is about 8.1% more accurate than classic CF and 1.3% than ACF. For about half of the categories, SMARTERDEALS is more than 2% better than ACF.

To put these results in perspective, it is important to point out that the Netflix competition, which carried a 1 million dollar prize, was about improving the RMSE compared to the

Table 2: Validation results

Parent Category	Classic CF (C)	ACF (A)	SmarterDeals (S)	(AiC)	(SiC)	SiC-AiC	Relative Performance (RP)
Automotive	1.16	1.15	1.12	1.4%	3.9%	2.5%	173.4%
Financial Services	1.67	1.60	1.55	3.9%	6.8%	2.9%	74.8%
Real Estate	1.74	1.68	1.64	3.1%	5.3%	2.3%	73.2%
Health & Fitness	1.09	1.02	0.99	6.2%	9.3%	3.0%	48.4%
Beauty & Spas	1.17	1.08	1.04	7.4%	11.0%	3.5%	47.1%
Education	1.16	1.11	1.09	4.5%	6.6%	2.1%	45.5%
Travel	0.97	0.91	0.89	6.1%	8.4%	2.3%	36.8%
Food & Drink	0.95	0.87	0.85	8.1%	10.3%	2.2%	26.7%
Arts and Entertainment	0.93	0.86	0.84	7.2%	9.0%	1.9%	26.2%
Restaurants	0.95	0.87	0.86	8.7%	9.8%	1.1%	12.7%
Shopping	1.60	1.55	1.55	3.1%	3.3%	0.2%	7.6%
Nightlife	0.85	0.76	0.75	11.6%	12.2%	0.6%	5.3%
Professional Services	0.90	0.78	0.80	12.6%	10.3%	-2.3%	-18.5%
Public Services & Government	1.11	0.98	1.03	11.8%	7.8%	-4.0%	-34.0%
Average	1.16	1.09	1.07	6.8%	8.1%	1.3%	37.5%

recommender system of Netflix by 10%.[10]

In terms of multiplicative relative performance, for some categories the accuracy of SmarterDeals is much better than ACF's (e.g., for Automotive our approach outperforms ACF by 173.4%, and 37.5% on average).

There are two parent categories from the 14 included in the tests, *Professional Services* and *Public Services and Government*, for which our method did not do very well. This seems attributable to the kind of business (service) that the users have rated. We hypothesize that this may be because users rate service-type businesses less carefully than the other ones.

7 Related Work

Several works have been proposed to take advantage of context information in recommendation systems based on collaborative filtering [2, 4, 5, 10, 17]. Adomavicius and Tuzhilin proposed a multidimensional approach that recommends items using contextual information, besides typical information about users' profiles and items [2, 4]. They demonstrated that in most cases the results of recommendation systems are better when considering context information. Moreover, they categorized contextual recommendations into three categories: contextual pre-filtering, contextual post-filtering and contextual modelling. In contextual pre-filtering, context is used to classify ratings according to specified context types before applying the recommendation method [6]. In contextual post-filtering, the recommendation method is applied first and then context information is used to filter the recommendations. In contextual modelling, context is directly integrated into the model [14]. Our SmarterDeals approach exploits contextual pre-filtering by calculating average ratings according to parent categories (cf. Equation (8)). Most importantly, supported by SmarterContext, SmarterDeals exploits users' preferences context gathered throughout their entire web experience. In this way, the calculation of similarities between users, and the prediction of ratings exploit contextual information about users gathered from the interactions of users with web applications.

An approach closely related to SmarterDeals is the framework proposed by Anand and Mobasher [5]. Their approach, based on memory models from cognitive science, proposes a user model based on short-term memory (STM) that stores current interactions of the active user, and

[10] Most of the teams got improvements by less than 2%.

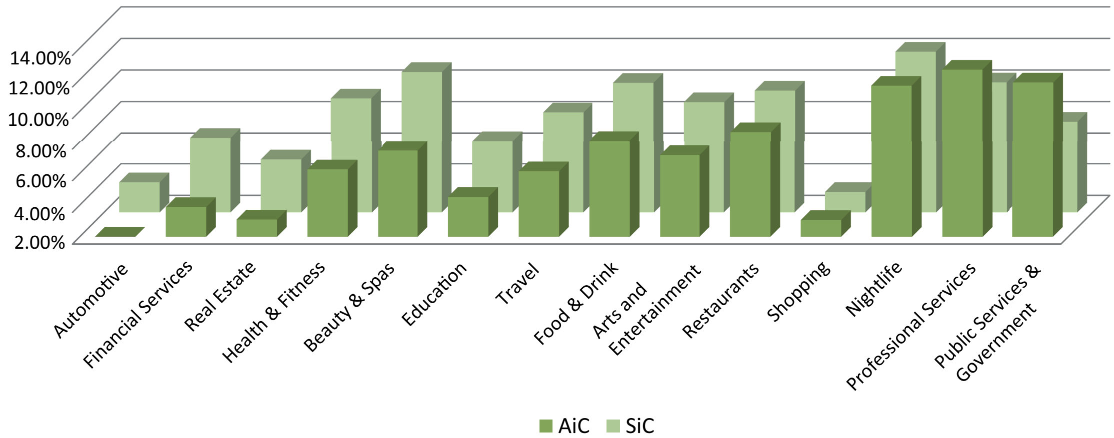

Figure 6: Improvement of ACF (AiC), and SmarterDeals (SiC) with respect to classic CF (C)

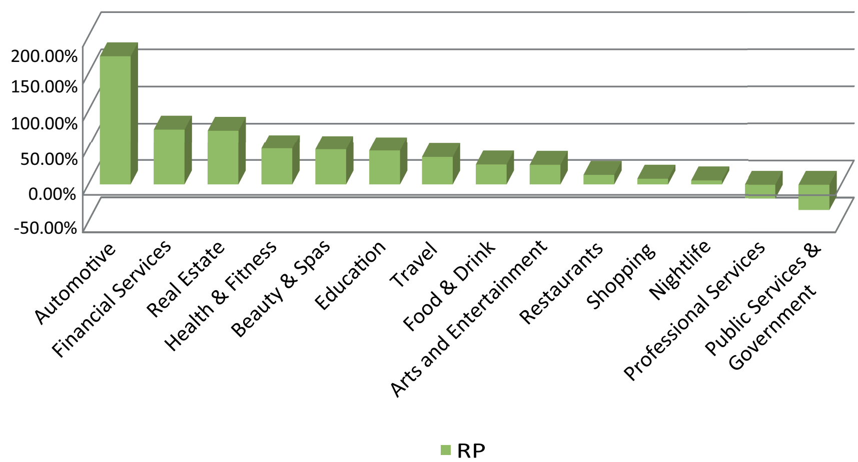

Figure 7: Relative performance of SmarterDeals (S) with respect to ACF (A)

a model based on long-term memory (LTM) that stores previous users' rating interactions, as well as the context of these ratings. Their approach aims to enrich STM models with contextually relevant ratings extracted from LTM models to improve the accuracy of the recommendations. To improve the accuracy of deal recommendations, SmarterDeals not only takes advantage of past and present user interactions with a particular web application (e.g., a daily deal application), but also exploits past and present users' interactions with any related web application (e.g., a shopping web site the user visited previously).

Finally, SmarterDeals centers on users rather than items. Item-based approaches, such as the one used by Koren and Bell, model user preferences based on similarities between items [16, 15]. In contrast, SmarterDeals recommends items based on similarities between users. Moreover, the management of the context information used to improve the accuracy of our recommendation engine is controlled fully by the user assisted by SmarterContext.

8 Conclusions

Even though the recent research on recommendation systems takes advantage of context information, existing e-commerce solutions such as daily-deal applications lack user-specific context-awareness. As a result, users are continuously frustrated with offers of products and services that, although generous, lack pertinence with respect to users' changing preferences, needs and situations.

In this paper we presented SmarterDeals, our deal recommendation system that takes advantage of context information about users to improve the relevance of product and service recommendations delivered to users. Our recommendation engine, built on top of existing recommendation approaches, exploits up-to-date context information about users maintained by our SmarterContext framework to improve the accuracy of deal recommendations. Most importantly, since context information is highly dynamic, the relevance of recommendations with respect to user situations is continuously compromised. We tackle the dynamic nature of context with SmarterContext by guaranteeing context models with up-to-date information. Moreover, since in SmarterContext the user decides about the relevance of context information, the accuracy of our recommendation engine is highly improved.

We demonstrated the suitability of SmarterDeals to recommend products and services based on parent categories defined in Groupon. We did not include other contextual dimensions different than user product preferences and location due to the lack of available real data sets containing user context information. Even though the Yelp data set allowed us to validate our approach partially, it does not contain other context types supported by SmarterContext such as social relationships, calendar events, and time context.

Future work will focus on the validation of our approach in different application domains, and the acquisition of new data sets that provide better support in terms of context information compliant with SmarterContext.

Acknowledgements

This work was funded in part by the National Sciences and Engineering Research Council (NSERC) of Canada under the NSERC Strategic Research Network for Smart Applications on Virtual Infrastructure[11] (SAVI - NETGP 397724-10) and Collaborative Research and Development program (CRDPJ 320529-04 and CRDPJ 356154-07), IBM Corporation, University of Victoria (Canada), and Icesi University (Colombia).

About the Authors

Sahar Ebrahimi is a Master Student and Research Assistant in the Department of Computer Science, University of Victoria, Canada. Her research interests include recommender systems and context-aware software applications. She received a Diploma Degree in Computer Engineering in 2004 from Tarbiat Moallem University of Tehran, Iran and a Master Degree in Computer Management in 2007 from University of Pune, India.

Norha M. Villegas is a PhD candidate in the Department of Computer Science, University of Victoria, Canada. She is a CAS student at the Center for Advanced Studies at the IBM Toronto Laboratory. Her dissertation focuses on the application of dynamic context management techniques for the optimization of self-adaptive software systems. She received a Diploma Degree in Systems Engineering and a Graduate Degree in Organizational Informatics Management in 2002 and 2004, from Icesi

[11] http://www.nsercpartnerships.ca/How-Comment/Networks-Reseaux/SAVI_AIIV-eng.asp

University, Cali, Colombia.

Hausi A. Müller is a Professor, Department of Computer Science and Associate Dean of Research, Faculty of Engineering at University of Victoria, Canada. Dr. Müller's research interests include software engineering, self-adaptive and self-managing systems, context-aware systems, and service-oriented systems. He received a Diploma Degree in Electrical Engineering in 1979 from the Swiss Federal Institute of Technology (ETH), Zürich, Switzerland and MSc and PhD degrees in Computer Science in 1984 and 1986 from Rice University in Houston, Texas, USA.

Alex Thomo is an Associate Professor and graduate advisor in the Department of Computer Science, University of Victoria, Canada. His research interests include theoretical and practical aspects of semi-structured and graph databases, social and biological networks, recommender systems, automata-based techniques, and index structures for textual data. Dr. Thomo received his PhD in Computer Science from Concordia University, Montreal, Canada in 2003.

References

[1] G. D. Abowd, A. K. Dey, P. J. Brown, N. Davies, M. Smith, and P. Steggles. Towards a Better Understanding of Context and Context-Awareness. In *Proceedings 1st International Symposium on Handheld and Ubiquitous Computing (HUC 1999)*, volume 1707 of *LNCS*, pages 304–307. Springer, Sep. 1999.

[2] G. Adomavicius, R. Sankaranarayanan, S. Sen, and A. Tuzhilin. Incorporating Contextual Information in Recommender Systems Using a Multidimensional Approach. *ACM Transactions on Information Systems*, 23(1):103–145, January 2005.

[3] G. Adomavicius and A. Tuzhilin. Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions. *IEEE Transactions on Knowledge and Data Engineering*, 17(6):734–749, June 2005.

[4] G. Adomavicius and A. Tuzhilin. Context-Aware Recommender Systems. In *Proceedings 2008 ACM Conference on Recommender Systems*, RecSys 2008, pages 335–336, New York, NY, USA, 2008. ACM.

[5] S. S. Anand and B. Mobasher. Contextual Recommendation. In B. Berendt, A. Hotho, D. Mladenic, and G. Semeraro, editors, *From Web to Social Web: Discovering and Deploying User and Content Profiles*, pages 142–160. Springer-Verlag, Berlin, Heidelberg, 2007.

[6] L. Baltrunas. Exploiting Contextual Information in Recommender Systems. In *Proceedings 2008 ACM Conference on Recommender systems*, RecSys 2008, pages 295–298, New York, NY, USA, 2008. ACM.

[7] R. M. Bell, K. Yehuda, and C. Volinsky. The BellKor Solution to the Netflix Prize. http://www.netflixprize.com/assets/ProgressPrize2007_KorBell.pdf, 2007.

[8] C. Bizer, T. Heath, and T. Berners-Lee. Linked Data — The Story So Far. *International Journal on Semantic Web and Information Systems*, 5(3):1–22, 2009.

[9] J. S. Breese, D. Heckerman, and C. Kadie. Empirical Analysis of Predictive Algorithms for Collaborative Filtering. In *Proceedings 14th conference on Uncertainty in Artificial Intelligence*, UAI 1998, pages 43–52, San Francisco, CA, USA, 1998. Morgan Kaufmann Publishers Inc.

[10] A. Chen. Context-Aware Collaborative Filtering System: Predicting the User's Preferences in Ubiquitous Computing. In *Proceedings 2005 Human Factors in Computing Systems. Extended abstracts*, CHI EA 2005, pages 1110–1111, New York, NY, USA, 2005. ACM.

[11] Econsultancy - Digital Marketers United. Groupon: Your Email Marketing is Failing your Fans. http://econsultancy.com/us/blog/7291-groupon-your-email-marketing-is-failing-your-audience.

[12] Google Inc. The Google Product Taxonomy. http://support.google.com/merchants/bin/answer.py?hl=en&answer=160081, 2012.

[13] J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl. Evaluating Collaborative Filtering Recommender Systems. *ACM Transactions on Information Systems*, 22:5–53, 2004.

[14] A. Karatzoglou, X. Amatriain, L. Baltrunas, and N. Oliver. Multiverse Recommendation: N-Dimensional Tensor Factorization for Context-Aware Collaborative Filtering. In *Proceedings 4th ACM Conference on Recommender systems*, RecSys 2010, pages 79–86, New York, NY, USA, 2010. ACM.

[15] Y. Koren. Factorization Meets the Neighborhood: a Multifaceted Collaborative Filtering Model. In *Proceedings 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD 2008, pages 426–434, New York, NY, USA, 2008. ACM.

[16] Y. Koren and R. M. Bell. Advances in Collaborative Filtering. In F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor, editors, *Recommender Systems Handbook*, chapter 5, pages 145–186. Springer, Boston, MA, 2011.

[17] U. Panniello, A. Tuzhilin, M. Gorgoglione, C. Palmisano, and A. Pedone. Experimental Comparison of Pre- vs. Post-Filtering Approaches in Context-Aware Recommender Systems. In *Proceedings 3rd ACM Conference on Recommender Systems*, RecSys 2009, pages 265–268, New York, NY, USA, 2009. ACM.

[18] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl. GroupLens: An Open Architecture for Collaborative Filtering of Netnews. In *Proceedings 1994 ACM conference on Computer Supported Cooperative Work*, CSCW 1994, pages 175–186, New York, NY, USA, 1994. ACM.

[19] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl. Item-based Collaborative Filtering Recommendation Algorithms. In *Proceedings 10th International Conference on World Wide Web*, WWW 2001, pages 285–295, New York, NY, USA, 2001. ACM.

[20] U. Shardanand and P. Maes. Social Information Filtering: Algorithms for Automating “Word of Mouth”. In *Proceedings 1995 CHI conference on Human Factors in Computing Systems*, CHI 1995, pages 210–217, New York, NY, USA, 1995. ACM Press/Addison-Wesley Publishing Co.

[21] The Economist. Groupon Anxiety. http://www.economist.com/node/18388904.

[22] The World Wide Web Consortium (W3C). OWL Web Ontology Language Reference. http://www.w3.org/TR/owl-ref/, 2004.

[23] N. M. Villegas and H. A. Müller. Managing Dynamic Context to Optimize Smart Interactions and Services. In M. Chignell, J. Cordy, J. Ng, and Y. Yesha, editors, *The Smart Internet: Current Research and Future Applications*, volume 6400 of *LNCS*, pages 289–318. Springer, 2010.

[24] N. M. Villegas and H. A. Müller. The SmarterContext Ontology and its Application to the Smart Internet: A Smarter Commerce Case Study. In M. Chignell, J. Cordy, J. Ng, and Y. Yesha, editors, *Second Book on the Personal Web*, LNCS. Springer, 2012 (in evaluation).

[25] N. M. Villegas, H. A. Müller, J. C. Muñoz, A. Lau, J. Ng, and C. Brealey. A Dynamic Context Management Infrastructure for Supporting User-driven Web Integration in the Personal Web. In *Proceedings 2011 Conference of the Center for Advanced Studies on Collaborative Research (CASCON 2011)*, pages 200–214, Markham, ON, Canada, 2011. IBM Corp.

[26] Yelp Inc. Yelp’s Academic Dataset. http://www.yelp.com/academic_dataset.