Top-N recommendations on Unpopular Items with Contextual Knowledge

Paolo Cremonesi
Politecnico di Milano, Italy
paolo.cremonesi@polimi.it

Paolo Garza
Politecnico di Milano, Italy
garza@elet.polimi.it

Elisa Quintarelli
Politecnico di Milano, Italy
quintare@elet.polimi.it

Roberto Turrin
Moviri srl, Italy
roberto.turrin@moviri.com

ABSTRACT

Traditional recommender systems provide recommendations of items to users; recently, some of them also consider the context related to predictions. In this paper we propose a technique that relies on classical recommendation algorithms and post-filters recommendations on the basis of contextual information available for them. Association rules are exploited to identify the most significant correlations among context and item characteristics. The mined rules are used to filter the predictions performed by traditional recommender systems to provide contextualized recommendations. Our experimental results show that the proposed approach allows improving the output of classical algorithms proposed in the literature, especially in the case of unpopular items.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous

General Terms

Algorithms, Experimentation

Keywords

Recommender system, Association rules, Contextual data

1. INTRODUCTION AND RELATED WORK

Recommender systems help people in retrieving potentially useful information or products in a huge set of choices by using the knowledge of the individual's past ratings, but also the ratings of all the system users.

However, other aspects, such as the situation, location, and time can influence the user ratings and thus, can be used to improve the accuracy of recommendations. For example, the restaurants and cuisine types chosen during the winter can be different from those chosen in the summer; moreover, the age and location of a user can have a great impact in the choices related to the food domain. The use of contextual information in the area of recommendation systems has been considered in the recent literature [1, 7, 8, 13, 14]. In [13] the notion of context is integrated in the customer's behavior model for improving the prediction of their behavior; however, in [8] the authors show that collecting contextual information relevant for recommendation purposes is a hard task and they develop a special-purpose browser to gather such information. The proposals related to contextual recommendation systems can be classified into three main groups: *pre-filtering*, *post-filtering*, and *contextual modeling* [2]. In [1] the context is used for rating estimation in multidimensional recommender systems, where contextual information are added to the standard dimensions related to users and items. In [7] a proposal which uses contextual values as virtual items used together to standard ones is presented; the advantage is that classical recommendation algorithms can be used without any modification. In [4] an approach to pre-filter the item ratings on the basis of the possible values of a unique contextual perspective, that is the season when the rating was expressed, is described. [14] contains a comparison between the pre-filtering and the post-filtering approaches and identifies which method dominates the other and under which circumstances, since, in general, there is not an approach better than the others.

In this paper we introduce a generic contextual post-filtering technique, not specifically adapted to a target scenario; however, to clarify the exposition, in the examples we will use the movie domain. The proposed approach exploits association rules to identify the most significant correlations among context and item characteristics and uses them to filter the predictions performed by a traditional recommender system in order to provide contextualized recommendations. The performed experiments show the effectiveness of the proposed approach in the context of top-N recommendations, in particularly when unpopular items are considered.

The paper is organized as follows. In Section 2 the proposed approach is described, while in Section 3 the obtained experimental results are discussed. Finally, Section 4 draws conclusions and discusses future work.

CARS-2011, October 23, 2011, Chicago, Illinois, USA.

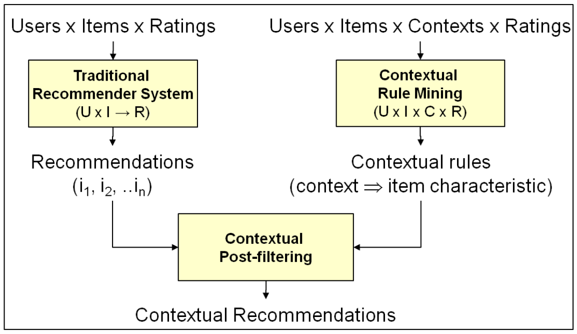


Figure 1: The Context-RS recommender system

2. PROPOSED ALGORITHM

The notion of context has emerged with different interpretations in various fields of research like psychology, philosophy, or computer science. A widely accepted definition is the one proposed in [6], where the context is considered as any information useful to characterize the situation of an entity, in our work an access to an item in the movie domain. In general, contextual information can be explicitly declared by the user, e.g., the user is interested in dramatic movies when he/she is alone, and in horror movies when he/she is with friends, or implicitly inferred by the system, as in the case of temporal information, or by sensors, e.g., the location. The contextual information available for our scenario includes static demographic information, i.e., *Age*, *Gender*, *Occupation*, and *ZIP code* (of the residence address) of the users. However, the generality of our proposal allows us to extend our experiments to domains where a wider set of contextual and dynamic values is traced. For example, we could consider the user *situation* (he/she is alone or with friends) and his/her *location* (e.g., at home, at office).

2.1 The Context-RS recommender system

We propose a new recommender system, called Context-RS, that combines contextual knowledge, traditional recommender systems, and association rules to improve the quality of top-N recommendations. Context-RS is a post-filtering approach (see Figure 1) based on the available knowledge about users (U), items (I), contexts (C), and ratings (R). It exploits association rules to find useful correlations between contextual knowledge and item characteristics and uses the extracted rules to filter uninteresting items from the items recommended by a traditional recommender system.

Given an arbitrary user u in the context c, the proposed approach works as follows to select the top-N items to recommend to u.

1. for each unrated item i a traditional (collaborative) recommender system is used to predict the appropriate rating r_{ui} of user u
2. the subset of items having a set of characteristics related to the current context c, according to the extracted contextual association rules, is selected
3. the top-N items of the subset of items selected during the previous step are recommended to user u

The main difference between Context-RS and the previous post-filtering approaches is given by the adopted selection technique (step 2). To decide which types of items are related to a context c, and hence which items must be considered, we propose to exploit association rules, and in particular the correlations between context and item characteristics. Association rules [3] are rules of the form $X \Rightarrow Y$, where both X and Y are sets of objects. In our approach, X is a set of predicates representing a context and the consequent is a characteristic of the available items. For instance, $(gender = M) \wedge (age = [20-25]) \Rightarrow (genre = horror)$ is an association rule representing a correlation between the context $(gender = M) \wedge (age = [20-25])$ and the *horror* genre mined from the available data (users, items, contexts, and ratings). Association rules have been successfully applied to market basket analysis [3] and automatic data classification [11]. In this work we show that they can be profitably exploited also in the context of recommender systems.

Two measures are usually used to assess the quality of association rules: support (sup) and confidence ($conf$). The support represents the frequency of the rule in the analyzed data, while the confidence estimates the conditional probability of the consequent given the antecedent.

The proposed approach infers which kinds of items are more of interest for each context c by extracting association rules from the available training ratings, by taking into consideration also the context in which the ratings have been given and the characteristics of the rated items. In particular, it extracts all the association rules of the form *context* ⇒ *item charactetistic* with support and confidence higher than the minimum thresholds $minsup$ and $minconf$, respectively. We remark that the characteristics of the considered users are also part of the contextual knowledge. Hence, in the antecedent of the extracted rule predicates on the characteristics of the users can be present (e.g., a predicate such as $gender = M$).

After the set of contextual rules has been mined, for each context c the consequents of the association rules having c as antecedent are used to form a set S_c composed of the item characteristics which are more frequently related to c. Only those items which are characterized by a characteristic that belongs to S_c can be recommended when the user current context is c. For example, suppose the following two rules are extracted for the context $(gender = M)$:

$(gender = M) \Rightarrow (genre = horror), sup = 5\%, conf = 50\%$
$(gender = M) \Rightarrow (genre = action), sup = 4\%, conf = 40\%$

It follows that S_c is equal to $\{(genre = horror), (genre = action)\}$ for the context $(gender = M)$. Hence, only horror and action movies can be recommended to male users when our approach is adopted.

2.1.1 Contextual rule mining

In order to be able to apply traditional rule mining algorithms to mine contextual rules, the available data (ratings and the related knowledge) must be represented in the transactional data format [3]. In the context of association rule a transaction is defined as a set of items, and a transactional dataset as a set of transactions [3]. For the extraction of the contextual rules exploited by our approach, a transactional dataset D is generated by applying a data transformation on the rating data taking into consideration also contextual knowledge and item characteristics. For each rating four pieces of information are known: the rated item (i), the user who rated the item (u), the context in which the rating has been given (c), and the rating value (r_{ui}). Each rating $<i,u,c,r_{ui}>$ given by user u to item i in the context c is transformed in a transaction (i.e., set of pairs) composed of the following pairs:

- one pair (i.characteristic=value) for each characteristic of the considered item i
- one pair (u.characteristic=value) for each characteristic of the considered user u
- one pair (c.characteristic=value) for each characteristic of the considered context c

For instance, consider a rating given by a male in the situation "with friends" and in the location "at home" to a fantasy movie produced in year 2001. The set $\{(gender = M), (situation = with\ friends), (position = at\ home), (genre = fantasy), (year = 2001)\}$ is included in D.

In order to extract only rules related to positive user experiences we have considered only the positive user ratings (i.e., 4 and 5 out of 5).

From the generated dataset D, contextual association rules are efficiently mined by means of an implementation of the FP-growth algorithm downloaded from the FIMI website [9].

3. EXPERIMENTAL RESULTS

Since large datasets with contextual information are not available, in our initial evaluation of the proposed approach we used the Movielens dataset and we considered as context ("static" context) the available demographic data.

3.1 Testing methodology, datasets, and algorithms

Our study considered two neighborhood (CorNgbr and NNCosNgbr) and two latent-factor (AsySVD and PureSVD) collaborative algorithms, along with two non-personalized algorithms used as baseline (TopPop and MovieAvg).

The two non-personalized algorithms recommend static lists of items regardless the collected user ratings: *TopPop* (Top Popular) suggests the most rated items (i.e., the most popular), while *MovieAvg* (Movie Average) suggests the highest rated items (i.e., the most liked).

Neighborhood collaborative algorithms are based on the similarity relationships among either users or items, in terms of collected ratings. *CorNgbr* (Correlation Neighborhood) computes item-item similarity by means of the Pearson linear correlation coefficient [10]. Similarly, *NNCosNgbr* (Non-normalized Cosine Neighborhood) computes item-item similarity by means of the cosine coefficient [5].

Latent factor collaborative models represent users and items in a common low-dimensional 'latent factor' space. *AsySVD* (Asymmetric SVD) is a matrix factorization model that reported an RMSE of 0.9000 in the Netflix context [10]. *PureSVD* is a latent factor algorithm recently proposed [5], whose rating estimation rule is based on the conventional SVD, where unknown ratings have been treated as zeros.

In this work we have considered the 1-million-rating Movielens dataset [12], which publishes user ratings along with movie genres (e.g., comedy, horror, ...) and demographics (e.g., user gender, age, ...). According to the methodology adopted in [5], known ratings are split into a training set $\mathcal{M}$ and a test set $\mathcal{T}$. The test set $\mathcal{T}$ contains only 5-star ratings, i.e., the items relevant to the respective users. Therefore, we have firstly trained the algorithm over the ratings in $\mathcal{M}$. Then, for each rating in $\mathcal{T}$ given by user u to item i, (i) we have randomly selected 1,000 additional items unrated by user u, (ii) we have predicted the ratings for the test item i and the additional 1,000 items, and (iii) we have formed a top-N recommendation list composed by the N items with the highest predicted ratings. Finally, the recommendation quality has been measured in terms of *recall(N)*, defined as the percentage of tested items that appear in the top-N list [5].

Algorithm		**All items**	**Long tail items**
PureSVD	standard	28.90	14.14
	with context	29.32	16.17
AsySVD	standard	13.47	4.40
	with context	14.26	6.96
NNCosNgbr	standard	25.75	7.75
	with context	26.33	9.60
CorNgbr	standard	7.59	2.91
	with context	8.63	3.97
TopPop	standard	14.35	0.03
	with context	16.07	0.06
MovieAvg	standard	0.50	0.27
	with context	1.77	0.42

Table 1: Recall% at N=3

Similarly to [5], an advanced analysis used to compute the quality on long-tail items has been performed by selecting from $\mathcal{T}$ the subset of ratings related to unpopular items. In our tests, we have considered only items with less than 990 ratings (the 95% of items) which refer to the 67% of ratings.

3.2 Recall analysis

In this section, we analyze the recall achieved by contextual and not contextual algorithms on the Movielens dataset. Initially, we report a high-level analysis related to the recall achieved by setting the number of recommended items N to 3 (real systems such as IMDB and Amazon usually show from 2 to 5 recommended items per page depending on their layout). Then a deeper discussion on the recall achieved by varying N on the long tail items is reported. We recall that, in the performed experiments, the context is given by age and gender of the users, while the considered item characteristic is the movie genre. For the contextual rule mining step, we set the minimum support threshold to 0.01%, while the minimum confidence was initially set to 0% (the effects of the enforced minimum confidence threshold is analyzed in the following). Table 1 summarizes the recall at N=3 for the considered algorithms. For each algorithm we report the recall obtained by the traditional version of the algorithm (standard version) and that achieved by the contextualized version (with context). We report the recall obtained on all items and that achieved on the long tail items. The exploitation of contextual rules allows improving the performance of all the considered algorithms in both cases (all items and long tail items). The recall improvement is on the average higher when non-personalized algorithms are used (e.g., +1.72% for TopPop on all items) and less relevant when state of the art latent factor based recommender systems are considered (e.g., +0.42% for PureSVD on all items). Even if contextual knowledge allows improving recall also when all items are considered, its positive effect is more evident when long tail items are considered. In this more difficult situation the information provided by the automatically extracted contextual rules allows improving more significantly also the recall of the better performing algorithms (+2.03% for PureSVD and +1.85% for NNCosNgbr). The post-filtering operation, performed by means of contextual rules, allows focusing on the most interesting movie genres for each context discarding irrelevant items from the top-N prediction list. The improvement of recall on the long tail items is an interesting property. In fact, a good recall on the long tail items means that the proposed approach is able to properly recommend also not best seller items, providing more novelty.

3.2.1 *Prediction of long tail items*

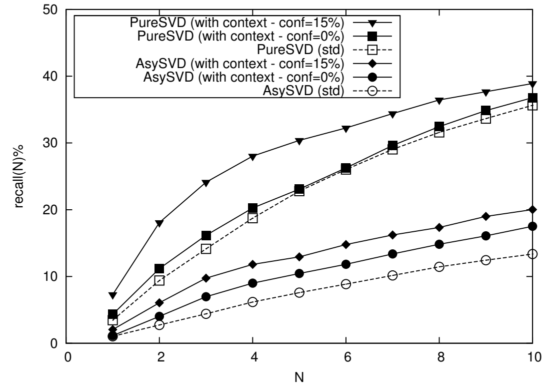
Figures 2(a)-2(c) report the recall-at-N when varying the value of N. For the contextual algorithms we report the results obtained by using two different settings during the contextual rule mining step: $minconf$=0% and $minconf$=15%. The second configuration ($minconf$=15%) is more selective. Hence, for each context a fewer genres are selected. However, as discussed in the following, the second configuration allows achieving higher recall values.

The contextual versions of the considered algorithms are significantly better than the not contextual ones, independently of the considered algorithm. The more selective configuration, i.e., the one with $minconf$=15%, shows to be the most effective (+7.6% at N=5 with respect to standard version for PureSVD). We performed also a set of experiments by setting $minconf$ equal to 20%. However, in that configuration the filter becomes too tight and recall decreases.

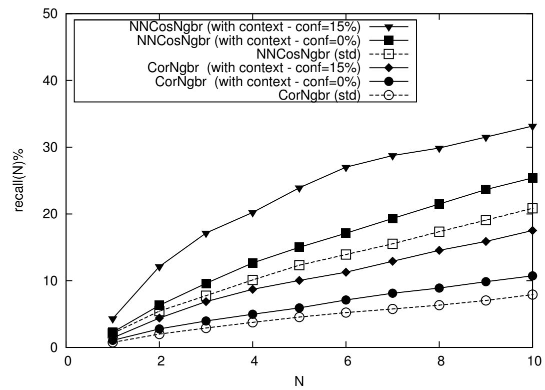
The use of contextual rules allows improving significantly the recall of the non-personalized algorithm TopPop (+18.6% at N=10 with respect to standard TopPop). The recall of the contextual version of TopPop is comparable to that achieved by the standard versions of many other more complex algorithms. In particular, the recall achieved by the contextualized version of TopPop is comparable to that of standard NNCosNgbr and even better than standard AsySVD.

3.2.2 *Recommendations for new users*

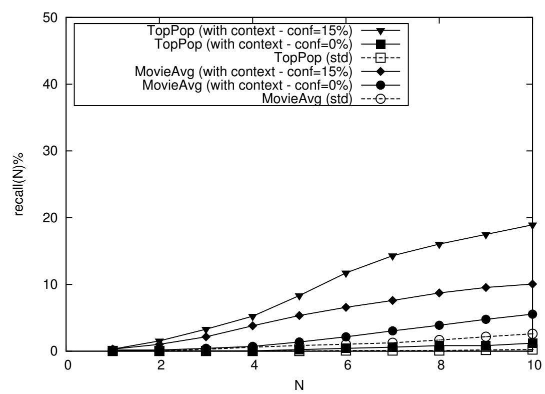
The good performance achieved by the contextualized version of TopPop is particularly of interest in presence of new users. When recommender systems have to deal with new users, the state of the art collaborative approaches cannot be used because the history of new users is empty. In this situation only non-personalized algorithms, such as TopPop and MovieAvg, can be used. Since the contextualized version of TopPop needs only to know the current context and the profile of the new user (e.g., Age and Gender), contextualized TopPop can be used to recommend items to new users. The recall at N=10 of the contextualized TopPop is 18.9% while that of the standard (not contextualized) TopPop is 0.25%. Hence, our approach is a very interesting solution when new users have to be managed.



(a) Latent factor algorithms



(b) Neighborhood algorithms



(c) Non-personalized algorithms

Figure 2: Movielens: recall-at-N on long tail (95% of items)

3.3 Explanation provided by contextual rules

The main goal of Context-RS is the recall improvement. However, the proposed approach adds also expressiveness to traditional collaborative recommender systems. In fact, the extracted contextual rules can be used to explain to end users the reason behind the performed recommendations. A merely list of items is usually less appealing than a set of items with an explanation. All the items recommended by Context-RS are selected because (i) they are ranked high by the used standard collaborative filter and (ii) their characteristics are considered interesting for the current context, according to the extracted contextualized rules. By simply showing together to each recommended item also the contex-

Context	$\Rightarrow$	Item characteristic	Support% (Absolute Support)	Confidence
$(gender = F) \wedge (age = [35 - 44])$	$\Rightarrow$	$(genre = Drama)$	1.72% (17204)	34.69%
$(gender = F) \wedge (age = [35 - 44])$	$\Rightarrow$	$(genre = Comedy)$	1.59% (15903)	32.16%
$(gender = F) \wedge (age = [35 - 44])$	$\Rightarrow$	$(genre = Romance)$	0.87% (8702)	17.62%
$(gender = F) \wedge (age = [35 - 44])$	$\Rightarrow$	$(genre = Children's)$	0.38% (3801)	7.62%
$(gender = F) \wedge (age = [35 - 44])$	$\Rightarrow$	$(genre = Musical)$	0.24% (2401)	4.89%
$(gender = F) \wedge (age = [35 - 44])$	$\Rightarrow$	$(genre = Animation)$	0.22% (2200)	4.36%
$(gender = F) \wedge (age = [35 - 44])$	$\Rightarrow$	$(genre = Mystery)$	0.19% (1900)	3.85%
$(gender = F) \wedge (age = [35 - 44])$	$\Rightarrow$	$(genre = Fantasy)$	0.15% (1500)	2.98%
$(gender = M) \wedge (age = [35 - 44])$	$\Rightarrow$	$(genre = Action)$	3.40% (34007)	22.76%
$(gender = M) \wedge (age = [35 - 44])$	$\Rightarrow$	$(genre = Thriller)$	2.42% (24205)	16.18%
$(gender = M) \wedge (age = [35 - 44])$	$\Rightarrow$	$(genre = Sci - Fi)$	2.15% (21504)	14.41%
$(gender = M) \wedge (age = [35 - 44])$	$\Rightarrow$	$(genre = Adventure)$	1.72% (17204)	11.51%
$(gender = M) \wedge (age = [35 - 44])$	$\Rightarrow$	$(genre = War)$	1.03% (10302)	6.91%
$(gender = M) \wedge (age = [35 - 44])$	$\Rightarrow$	$(genre = Horror)$	0.90% (9002)	6.05%
$(gender = M) \wedge (age = [35 - 44])$	$\Rightarrow$	$(genre = Mystery)$	0.52% (5201)	3.48%
$(gender = M) \wedge (age = [35 - 44])$	$\Rightarrow$	$(genre = Western)$	0.33% (3301)	2.19%
$(gender = M) \wedge (age = [35 - 44])$	$\Rightarrow$	$(genre = Film - noir)$	0.30% (3001)	2.01%
$(gender = M) \wedge (age = [35 - 44])$	$\Rightarrow$	$(genre = Documentary)$	0.12% (1200)	0.78%

Table 2: Contextual rules for two representative contexts

tual rule that has been used to select it we can improve the confidence of the user on the provided recommendations. As an example of the knowledge provided by contextual rules, the sets of rules for two representative contexts are reported in Table 2. The first context is "female in the range 35-44 years old", while the second one is "male in the range 35-44 years old". In both contexts, only a subset of the 18 available genres is automatically selected by means of the extracted contextual rules.

4. CONCLUSIONS AND FUTURE WORK

In this paper, we showed how contextual rules, representing frequent relationships between context information and the characteristics of the rated items, allow increasing the recall of state-of-the-art collaborative recommender systems. Our approach can be profitably applied to both personalized and non-personalized recommender systems.

As future work, we are focusing our attention on the cross-domain problem. In particular, we are investigating the possible application of contextual rules, extracted from a dataset, to another dataset. For example, the set of rules mined from a movie dataset could be applied to a book dataset. Obviously a mapping function is needed to map the rules mined from one domain to the domain of the other dataset. This approach could be useful (i) when past contextual history is available only for one dataset or (ii) in presence of noisy datasets. Contextual rules are mined from the first, non-noisy, dataset and are used to perform contextualized recommendations for the second dataset.

5. REFERENCES

[1] G. Adomavicius, R. Sankaranarayanan, S. Sen, and A. Tuzhilin. Incorporating contextual information in recommender systems using a multidimensional approach. *ACM Trans. Inf. Syst.*, 23(1):103–145, 2005.

[2] G. Adomavicius and A. Tuzhilin. Context-aware recommender systems. In *ACM RecSys'08*, pages 335–336, 2008.

[3] R. Agrawal, T. Imieliński, and A. Swami. Mining association rules between sets of items in large databases. In *ACM SIGMOD'93*, pages 207–216, 1993.

[4] L. Baltrunas and F. Ricci. Context-based splitting of item ratings in collaborative filtering. In *ACM RecSys'09*, pages 245–248, 2009.

[5] P. Cremonesi, Y. Koren, and R. Turrin. Performance of recommender algorithms on top-n recommendation tasks. In *ACM Recsys'10*, pages 39–46, 2010.

[6] A. Dey. Understanding and using context. *Personal Ubiquitous Computing*, 5(1):4–7, 2001.

[7] M. A. Domingues, A. M. Jorge, and C. Soares. Using contextual information as virtual items on top-n recommender systems. In *Workshop on Context-Aware Recommender Systems (CARS-2009)*, pages 1–5, 2009.

[8] M. Gorgoglione, C. Palmisano, and A. Tuzhilin. Personalization in context: Does context matter when building personalized customer models? In *ICDM'06*, pages 222–231, 2006.

[9] R. J. B. Jr., B. Goethals, and M. J. Zaki. Frequent itemset mining implementations repository, 2004. `http://www.cs.rpi.edu/~zaki/Workshops/FIMI/`.

[10] Y. Koren. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *ACM SIGKDD'08*, pages 426–434, 2008.

[11] B. Liu, W. Hsu, and Y. Ma. Integrating classification and association rule mining. In *ACM SIGKDD'98*, pages 80–86, 1998.

[12] B. Miller, I. Albert, S. Lam, J. Konstan, and J. Riedl. MovieLens unplugged: experiences with an occasionally connected recommender system. In *International conference on Intelligent user interfaces*, pages 263–266, 2003.

[13] C. Palmisano, A. Tuzhilin, and M. Gorgoglione. Using context to improve predictive modeling of customers in personalization applications. *IEEE Trans. Knowl. Data Eng.*, 20(11):1535 –1549, 2008.

[14] 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 *ACM RecSys'09*, pages 265–268, 2009.