Usage Context-based Object Recommendation

Katja Niemann
Fraunhofer Institute for Applied Information Technology FIT
Schloss Birlinghoven
Sankt Augustin, Germany
www.fit.fraunhofer.de
katja.niemann@fit.fraunhofer.de

In this paper, we present a new way of discovering semantic relations between items by comparing their usage contexts in online portals. This way, we do not need to consider the content of the items to find semantic similarities. Additionally, we explore the use of these usage context-based similarities for enhancing recommender systems.

recommender systems, collaborative filtering, semantic similarity, usage data, context

1. INTRODUCTION

Recommender systems are taking on a steadily growing importance in an expanding number of domains (movies, music, books, etc.) to filter relevant items for users. The most well-known approaches are collaborative filtering (i.e. neighbourhood-based and matrix factorization techniques) and content-based filtering (Adomavicius and Tuzhilin (2005), Koren et al. (2009)). With the evolution of recommender systems, the expectations regarding the recommendations also rise, e.g. in terms of accuracy, diversity, and novelty and several systems have been created that deal with these expectations (Adamopoulos and Tuzhilin (2011), Adomavicius and Kwon (2012), Vargas and Castells (2011)).

A remaining problem is sparsity. In this paper, we deal with datasets coming from the area of Technology Enhanced Learning (TEL) and Movie Recommendation. Datasets from the TEL domain tend to be very sparse (Verbert et al. (2011)) which can make it impossible or considerably more difficult to recommend items at all. In areas like movie recommendation big datasets are available, however, it is still challenging to recommend niche items, i.e. items from the long tail, since they do not hold extensive usage data (Goel et al. (2010)).

In this paper, we focus on inferring information from the given usage data to enhance recommender systems. We got inspiration from corpus-driven lexicology that comes with techniques to find semantic relations between words based on their

usage in sentences and apply them to items used in sessions. One way to find semantically related words is the use of paradigmatic relations. Words stand in linear orders, e.g. in speech or in written texts. The usage context of a word can thus be defined by the words that occur before and after it. If two words have very similar contexts, they are said to be paradigmatically related (Saussure (1986)). As example: The words "car" and "vehicle" share a similar usage context holding e.g. the words "driver" and "highway". Thus, paradigmatic relations lead us to semantic relations, this is to say, context similarity correlates with content relatedness.

Similar to words used in sentences, data objects are used in sessions and we can create usage contexts for objects similar to those for words. Thus, a usage context of an object holds all objects it significantly often co-occurs with. Hereby, the definition of the terms session and usage context depends on the application domain and the available information, e.g. all events conducted by a user between logging in and out in a web portal. This leads us to our first research question:

RQ 1: Can we detect semantic similarities of objects by comparing their usage contexts?

If so, the question arises if and how these semantic similarities can be utilized to be supportive for the users which leads to the second research question:

RQ 2: Can these similarities be used to enhance current recommender systems?

This paper is structured as follows. Chapter 2 deals with RQ 1 and chapter 3 with RQ 2 by explaining the approaches, the experimental set-ups and the results of the experiments conducted to investigate the research questions. Finally, chapter 4 gives a conclusion and an outlook on future work.

2. USAGE CONTEXT-BASED OBJECT SIMILARITY

2.1. Approach

2.1.1. Significance of Co-occurrences

We define two items to be co-occurrences if they co-occur in at least one session and use statistical association measures to calculate their significance. Basic association measures calculate a significance score by comparing the observed frequency of a co-occurrence with its expected frequency, e.g. MI (mutual information) or z-scores (Evert (2004)). These association measures are sufficient for many applications, though they also have some limitations as they e.g. tend to fail when calculating the significance for a frequent and an infrequent object.

More sophisticated association measures and independence tests are always based on a cross-classification of a set of items, e.g. using contingency tables. These measures compare the expected and the observed frequencies as well. In contrast to the more simple approaches, they do not only consider the expected co-occurrence frequency of the two objects but compute the expected frequencies for all cells in the contingency table. Examples are log likelihood, Poisson and χ^2 (Evert (2004)).

2.1.2. Detection of a Suitable Threshold

After the calculation of the co-occurrences' significance values, the most significant ones must be selected for each object. There are two ways to do so, i.e. by ranking and selecting only the n most significant co-occurrences or by using a threshold. Since there is no standard scale of measurement to draw a clear distinction between significant and non-significant occurrences, the calculation of a suitable n or a threshold is an exploratory investigation.

2.1.3. Object Similarity Calculation

We calculate the similarity for each item pair using the cosine similarity at which each item is described by its most significant co-occurrences including their significance values.

2.2. Experimental Setup

We test our approach on two data sets collected in the MACE¹, respectively the Travel Well² Portal.

¹<http://mace-project.eu/>

²<http://lreforschools.eun.org/web/guest/travelwell-all>

The MACE project relates learning resources about architecture. While interacting with the MACE portal, users are monitored and their activities, e.g. accessing, rating or tagging an object are recorded. The events used for the evaluation were conducted by 620 users on 12.176 resources over a period of three years and hold user and item identifier as well as the event type and a timestamp. The TravelWell portal makes open educational resources available from more than 20 providers. The dataset contains information about the rating and tagging behaviour of 98 registered users over a period of six months. For each event, the date, user and item identifier as well as the tag, respectively the rating is stored. As there are no timestamps but only the dates, a session comprises all events conducted in one day.

In order to evaluate our approach we use the semantic metadata similarities of the objects, which are calculated using tags and classifications. We admit that in both data sets the semantic metadata are not perfectly accurate or even approximately complete. However, the data sets are comparatively good since large parts have been tagged by experts.

We calculate the usage context-based similarity for all item pairs in MACE and TravelWell using MI, log likelihood, and χ^2 . We select the significant co-occurrences for each item by a) varying the vector sizes and b) calculating an item-specific threshold, i.e. the average significance values of all its co-occurrences. In order to prove our hypothesis that usage-context similarity correlates with content similarity, we calculate the Pearson correlation coefficient between these similarity distributions.

2.3. Results

Figure 1 and figure 2 show a) the Pearson correlation coefficients for the semantic metadata-based and the usage context-based similarities that are calculated with the different association measures and varying vector sizes (count: a fixed n was chosen, avg: an object specific threshold, i.e. the average significance value of its co-occurrences, was used) as well as b) the amount of item pairs for which a similarity score could be calculated.

As could be assumed, the more co-occurrences are used to describe an object, the more usage context-based similarities can be detected between object pairs. Additionally, the correlation with the semantic metadata-based similarity increases with an increasing co-occurrence vector size, whereas from a certain vector size on, the amount of found similar item pairs and the correlation get stable.

The log likelihood measure with the average threshold performs best in terms of correlation.

Figure 1: MACE: Pearson correlation coefficient and amount of considered item pairs

Figure 2: TravelWell: Pearson correlation coefficient and amount of considered item pairs

However, it can only calculate similarity values for a subset of the available objects. When comparing the association measures MI and corrected χ^2 with the average threshold, MI performs significantly better.

For the MACE dataset, the coefficients are significantly higher than for the TravelWell dataset, we assume this is due to the fact that the MACE dataset holds more detailed usage data, e.g. each item access is tracked and not only the rating and tagging activities as in TravelWell. Additionally, the sessions are more accurate in the MACE dataset since the MACE usage data offer timestamps.

The association measures MI, log likelihood and χ^2 behave quite similarly compared to each other for the MACE and the TravelWell dataset. For both datasets, the best performing set-up in terms of correlation coefficient and amount of object pairs is the log likelihood measure in combination with a "large" vector size, whereat the meaning of "large" must be defined depending on dataset. For MACE and TravelWell, a vector size of about 7% of the number of distinct objects in the dataset is recommended. This set-up is followed by MI in combination with the average threshold which has the advantage that no parameter for the vector size must be defined.

The correlation coefficients can be described as medium and as we regard a large sample of objects, the coefficients can said to be representative although no separate tests for bivariate normality have been undertaken (Bortz (1993)). Moreover, because of the big sample size the correlations

are significant on the 5%-level ($p=.05$). A further important point is the fact that the considered metadata can only be interpreted as a shallow content representation. Thus, it is possible that the correlations represent a lower bound for the real correlation between content and usage context. For a more detailed evaluation, please see Niemann and Wolpers (2013b).

3. UTILIZATION OF USAGE CONTEXT-BASED SIMILARITY FOR RECOMMENDATIONS

3.1. Usage Context-boosted Collaborative Filtering

3.1.1. Approach

Melville et al. (2002) introduced content-boosted collaborative filtering, i.e. a hybrid recommender system that uses the given rating history of users and content information of items to predict the missing ratings in a user-item rating matrix using a content-based recommender. This matrix is then used as input for collaborative filtering (CF) techniques.

In the previous section we show that the usage context-based similarity gives an indication for the semantic similarity of objects pairs. Here, we use a usage context-based recommender to predict the missing ratings in the user-item rating matrix. We compute the expected rating $p(u, i)$ on a item i for a user u by averaging the ratings given by the user to the other items in her profile $P(u)$ while each rating is weighted by the corresponding similarity $sim(i, j)$, see equation 1. The usage context-based similarity is calculated using the association measure MI in combination with the average threshold.

$$p(u, i) = \frac{\sum_{j \in P(u), i \neq j} (sim(i, j) * r(u, j))}{\sum_{j \in P(u), i \neq j} |sim(i, j)|} \quad (1)$$

The filled-in matrix can then be used as input for several recommendation approaches. Thus, the usage context-based boosting is similar to the content-based boosting but doesn't require any content information.

3.1.2. Experimental Setup

We perform a 5-fold cross-validation to compare the baseline recommendation approaches with their usage context-boosted versions for the MACE and the TravelWell dataset and calculate the predictive evaluation metric Root Mean Squared Error (RMSE). Additionally, we present the percentage of ratings in the test set for which predictions could be created.

The baselines algorithms are a) item-based CF (with adjusted cosine similarity), b) user-based

CF (with Pearson correlation based similarity), c) Single Value Decomposition (SVD) from the PREA toolkit (Lee et al. (2012)), and d) Biased Matrix Factorization (BMF) from the MyMediaLite library (Gantner et al. (2011)). The SVD and the BMF were chosen because they performed best on the given data sets in comparison to the other matrix factorization techniques implemented in the libraries. In order to ensure a fair comparison, the most suitable neighbour size (for the neighbourhood-based approaches), respectively the degree of normalization and the learning rate (for the matrix factorization techniques) were determined by cross-validation.

3.1.3. Results

Table 1 shows the results for the MACE dataset. Due to the sparsity (98,35%), the baseline algorithms IBCF, UBCF and SVD are only able to predict ratings for a small subset (14,8% - 24,9%) of the ratings the test sets. BMF predicts a rating for each possible user-item combination. For all underlying algorithms, but the IBCF, the boosted approach is able to decrease the RMSE up to 16,39%. Additionally, up to 4,5 times as much ratings are predicted with the boosting.

Table 1: MACE: Comparison of the approaches

	Baseline		Boosted	
	RMSE	Quantity	RMSE	Quantity
IBCF	1,0090	14,8%	1,0562	67,9%
UBCF	1,0816	14,8%	1,0792	67,9%
SVD	1,2975	24,9%	1,0848	70,4%
BMF	1,1144	100%	1,0583	100%

Table 2 shows the results for the TravelWell data set (Sparsity: 98,17%). Similarly to MACE, the baseline algorithms, with BMF as exception, can only create predicted ratings for a subset of the ratings in the test set whereas all boosted approaches create predicted ratings for about 3 times as much user-item combinations. For the IBCF, SVD and BMF algorithms, the usage context-boosted approach is able to decrease the RMSE up to 2,81%.

The MACE data set offers more extensive usage data than the TravelWell data set and additionally timestamps can be used to create more accurate sessions, thus, as also shown in chapter 2, the MACE usage data is better suited to create usage context-based similarities that imply semantic similarities. As result, all usage context-boosted approaches perform better for the MACE dataset.

For both datasets, the MF methods SVD and BMF profit from the usage context-boosting approach, whereas IBCF and UBCF are more sensible

Table 2: TravelWell: Comparison of the approaches

	Baseline		Boosted	
	RMSE	Quantity	RMSE	Quantity
IBCF	0,8909	31,5%	0,8769	94,7%
UBCF	0,8946	31,5%	0,9255	94,7%
SVD	0,9095	35,8%	0,8839	94,8%
BMF	0,8656	100%	0,8533	100%

concerning expected ratings in the user-item rating matrix that differ from the true user ratings.

To conclude, boosting can be recommended for the use in learning portals, especially in combination with Biased Matrix Factorization. If the collected usage data is fine-grained, i.e. all events concerning an item are stored and timestamps are given and/or only sparse semantic metadata information is given for the items, the use of the usage context can even outperform the use of semantic metadata (which is not shown here due to size constraints).

3.2. Usage Context-based Collaborative Filtering

3.2.1. Approach

When analysing usage data from learning portals we have a clear context definition. This is different when recommending movies. The fact that two movies were rated in the same session does not imply that these movies were "used", i.e. watched together. Additionally, movies are often not watched in a row. Thus, we need another context definition considering e.g. the weekday or the company. Since this information is often not available, we use the whole user profile as context and consider items as co-occurrences if they co-occur in a profile with a rating difference below a given threshold.

Apart from the different context definitions, the approach is the same as described in the previous chapters. First, we calculate the significance values for all co-occurrences and define which are the most significant ones. Then, we calculate the usage context-based similarities and the rating predictions. Here, we don't use the rating predictions to fill in the user-item rating matrix but directly use the predictions for recommendations.

When analysing how the usage context-based collaborative filtering approach (UC-BCF) works when these simplifications are conducted, one can see, that it combines features of the user- and the item-based approach:

- 1) If two items share a significant amount of users that rated them similarly, they are described through similar item vectors because they often co-occur with the same items in a user profile.

2) If two items are rated similarly by a significant amount of similar users (not necessarily the same users), they are also described by similar item vectors because similar users co-rated the same items similarly.

Additionally, there is a third feature not contained in the user- or item-based approach:

3) If two items are rated dissimilarly by a significant amount of dissimilar users, they hold similar item vectors, because dissimilar users co-rated the same items dissimilarly.

3.2.2. Experimental Setup

In the previous experiments we focus on the recommendations' accuracy and on being able to create recommendations at all. Here, with more data available using the MovieLens³ (1M) and the Netflix Prize⁴ data set, we do not only focus on accuracy but on diversity and novelty as well. Due to size constraints, we only show the results for accuracy and aggregate diversity. Following Adomavicius and Kwon (2012), we present the aggregate diversity as the total number of distinct items recommended to all users. Due to the fact that we need a uniform number of recommended items to evaluate our approach in respect to the aggregate diversity, we use the accuracy in top- n approach to calculate the recommender's accuracy instead of the commonly used measures precision and recall.

We perform a 5-fold cross-validation. Users that hold less than 20 highly rated items in the test set are removed from it and added to the training set, because the recommendation lists will have a size up to 20 and we want to distinguish between relevant (rated with at least 4 out of 5 stars) and irrelevant items. The UC-BCF is tested with the association measures χ^2 -corr (Chi), log-likelihood (Log), and Poisson-based (Poisson). We vary the co-occurrence vector size from 10 to 100 and the recommendation list size n from 5 to 20. Altogether, we run 240 experiments combining the different features. Since all algorithms perform quite similar for the different sizes of n , we only present the results for $n=10$.

3.2.3. Results

Fig. 3 shows the aggregate diversity, i.e. the total amount of distinct recommended items. The baseline algorithms perform dissimilar for the two data sets, e.g. the BMF method is the best performing baseline on the MovieLens data set while on the sparser Netflix data set, it is the worst performing one. Overall, the user-based collaborative filtering

Figure 3: Aggregate Diversity in Top-10

Figure 4: Accuracy in Top-10

approach performs best in terms of aggregate diversity.

Fig. 4 shows the comparison of the classification accuracy in the top-10 recommendation lists for the different approaches. Here, the baseline algorithms perform similarly on both data sets. As expected, the MF methods outperform the collaborative filtering techniques with SVD being the best and UBCF being the worst performing approach for both data sets.

The goal of the UC-BCF approach is to increase the aggregate diversity by pushing items from the long tail into the users' top- n recommendations without decreasing the classification accuracy of the recommendations. Therefore, we need to choose a co-occurrence vector size that is as small as possible to increase the aggregate diversity. Additionally, the size needs to be large enough to not decrease the classification accuracy. These considerations result in a vector size between 20 and 25 depending on the used association measure and the data set. Overall, χ^2 -corr with vector size 25 is the most promising association measure to be used with UC-BCF.

For the MovieLens data set, the usage of the UC-BCF approach with χ^2 -corr and vector size 25 raises the amount of recommended items up to 21.7% compared to the UBCF approach (from 919 to 1118) and at least by 5.45% compared to the BMF approach (from 1060 to 1118). In terms of accuracy, the SVD approach receives a 0.86% better value than the UC-BCF approach. Nonetheless, the UC-BCF approach outperforms all other baseline algorithms by 0.41%-1.91%.

³<http://www.grouplens.org/node/73>

⁴<http://www.netflixprize.com/>

For the Netflix data set, the amount of recommended items is improved up to 76.52% compared to the BMF approach (from 2101 to 3709 items) and at least by 42.37% compared to the UBCF approach (from 2605 to 3709 items) when using the UCBCF approach with χ^2 -corr and vector size 25. The accuracy is slightly increased as well (from 0.1% (SVD) to 2.5% (UBCF)). The improvement of the aggregate diversity is more significant for Netflix, which can be accounted for by the fact that Netflix holds more rarely rated objects than MovieLens that benefit from the new approach. For a more detailed evaluation, please see Niemann and Wolpers (2013a).

4. CONCLUSION AND FUTURE WORK

In this paper, we present a new approach to infer semantic information from analysing the usage contexts of items in online portals. Items are assumed to be similar if they often occur in similar usage contexts, but not necessarily together in the same sessions or user profiles. The experiments show that the usage context-based similarity correlates with the semantic metadata-based similarity that was calculated using tags and classifications. The usage context-based similarity was then used to predict user ratings on items. These predicted ratings can directly be used to recommend items or to enhance the user-item rating matrix in a feature augmented hybrid recommender system. This way, the accuracy of recommendations can be increased especially for seldom used items which results in a higher aggregate diversity. So far, we evaluated our approach in terms of accuracy, aggregate diversity and novelty. We will further research if the inferred semantic information can also be used to avoid over-specialisation.

We will expand our experiments to more areas and context definitions. A possible application area is music recommendation where we have an obvious usage context definition, i.e. songs that are listened to in a row. As stated before, we also aim to use additional contextual information to enrich the context definition e.g. for movies. Given this information, a session can e.g. comprise all movies a user watched at a weekend with her minor children or all movies she watched at a weekday with friends.

REFERENCES

- Adamopoulos, P., and Tuzhilin, A. (2011) *On unexpectedness in recommender systems: Or how to expect the unexpected*. DiveRS 2011, ACM RecSys Workshop on Novelty and Diversity in Recommender Systems.
- Adomavicius, G., and Tuzhilin, A. (2005) *Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions*. IEEE Transaction on Knowledge and Data Engineering 17(6).
- Adomavicius, G., and Kwon, Y. (2012) *Improving aggregate recommendation diversity using ranking-based techniques*. IEEE Transactions on Knowledge and Data Engineering 24.
- Bortz, J.: *Statistics*. Springer, Heidelberg (1993)
- Evert, S. (2004) *The Statistics of Word Cooccurrences: Word Pairs and Collocations* (Dissertation). University Stuttgart.
- Gantner, Z., Rendle, S., Freudenthaler, C., and Schmidt-Thieme, L. (2011) *MyMediaLite: A free recommender system library*. Proc. of the 5th ACM Conference on Recommender Systems.
- Goel, S., Broder, A. Z., Gabrilovich, E., and Pang, B. (2010) *Anatomy of the long tail: ordinary people with extraordinary tastes*. Proc. of the 3rd ACM Int. Conference on Web Search and Data Mining.
- Koren, Y. and Bell, R. and Volinsky, C. (2009) *Matrix Factorization Techniques for Recommender Systems*. Computer 42(8).
- Lee, J., Sun, M., and Lebanon, G. (2012) *A Comparative Study of Collaborative Filtering Algorithms*. ArXiv e-prints.
- Melville, P., Mooney, R.J., Nagarajan, R. (2002) *Content-boosted collaborative filtering for improved recommendations*. Proc. of the 18th National Conference on Artificial Intelligence.
- Niemann, K. and Wolpers, M. (2013) *A New Collaborative Filtering Approach for Increasing the Aggregate Diversity of Recommender Systems*. Proc. of the 9th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD).
- Niemann, K. and Wolpers, M. (2013) *Usage Context-boosted Filtering for Recommender Systems in TEL*. Proc. of the 8th European Conference on Technology Enhanced Learning (EC-TEL).
- Saussure, F. de (re-print 1986) *Course in General Linguistics*. Open Court Publishing, Chicago.
- Vargas, S., and Castells, P. (2011) *Rank and relevance in novelty and diversity metrics for recommender systems*. Proc. of the 5th ACM Conference on Recommender systems.
- Verbert K., Drachsler, H., Manolis, N. (2011) *Dataset-driven research for improving tel recommender systems*. Proc. of the 1st ACM International Conference on Learning Analytics and Knowledge.