Recommending hotels based on multi-dimensional customer ratings

Dietmar Jannach[a], Fatih Gedikli[a], Zeynep Karakaya[a], and Oliver Juwig[b]

[a] Fakultät für Informatik
TU Dortmund, Germany
{firstname.lastname}@tu-dortmund.de

[b] HRS Hotel Reservation Service
50676 Cologne, Germany
oliver.juwig@hrs.de

Abstract

Recommender Systems (RS) have shown to be a valuable means to support the traveller or tourist in his pre-trip information search and decision making processes. These systems often rely on rating information provided by the user community to make recommendations for individual users. In classical application domains such as movie or book recommendation, users provide one overall rating for each item. Customers in the travel and tourism domain however are often allowed to evaluate their hotel or holiday packages along several dimensions after the trip. In this work, we show through an empirical evaluation based on a real-world data set from the tourism domain that the predictive accuracy of an RS can be significantly improved when the multi-dimensional rating information is taken into account. In particular, we demonstrate that regression-based methods and in particular the novel combination of user- and item-based models leads to more accurate recommendations than previous approaches. In addition, we show that not all dimension (criteria) ratings are equally valuable for the prediction process and that a careful selection of rating dimensions can help to further increase the quality of the recommendations.

Keywords: recommender systems, collaborative filtering, multi-criteria ratings.

1 Introduction

Recommender Systems (RS) are information filtering and decision support tools, which have been successfully applied to support the online customer in various domains. In the travel and tourism domain, such systems are for example developed to help the customer in the pre-trip information search and decision making process. Examples of recent research include knowledge-based and conversational approaches to filter destinations and select travel packages (Jannach et al., 2009), (Jannach et al., 2007), (Zanker et al., 2008), context-aware recommendation of places of interest (Baltrunas et al., 2011), mobile recommenders (Ricci, 2011), or the development of more intelligent user interaction strategies (Mahmood et al., 2009).

Recommendation in the tourism domain has some specific particularities and challenges, which are not present in more classical RS application domains, in which especially collaborative filtering (CF) techniques have been successfully applied in the past. Customers in the tourism domain, for example, do not purchase items as frequently as customers of an online book store or movie rental system. Thus, the amount of user feedback and the buying history available for building systems based on collaborative filtering techniques may be limited, which is why conversational approaches are often chosen. Furthermore, the context of the traveller or tourist is particularly relevant. Think, e.g., of making recommendations for a group of people traveling together. Also, the type of the trip (business or private) or seasonal aspects can be important when recommending tourism products.

With regard to the sparsity of the data note that while the number of past transactions and ratings per user is relatively low when compared to other domains, one particular aspect in the tourism domain is that customers are often not only allowed to assign an overall rating to a hotel or travel package, but can rate the product or service along several dimensions. In the hotel domain, the typical dimensions or criteria include "room quality", "service quality", or "value-for-money", see Figure 1.

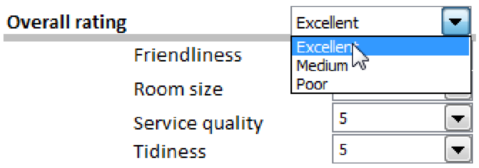

Figure 1. Example rating screen.

The value of such multi-criteria ratings for improving the accuracy of CF recommender systems has been analyzed in the past, e.g., by Adomavicius and Kwon (2007), who also proposed different algorithms that take these detailed ratings into account in the prediction process, see also (Adomavicius et al., 2011).

In this work, we build on previous work of Adomavicius and Kwon and propose a CF recommendation approach based on Support Vector regression (Drucker et al., 1997), which combines user- and item-models in a weighted approach to further improve the accuracy of the recommendations. We evaluate our method on a real-world data set provided by a major European tourism platform and compare it with state-of-the-art baseline algorithms based on matrix factorization. In addition, we aim to explore in this work whether or not all available criteria ratings are equally valuable to increase the predictive accuracy or if only a subset of the ratings should be exploited.

2 Improving RS accuracy based on multi-criteria ratings

2.1 Single-rating and multi-criteria rating algorithms

In the classical setting for collaborative filtering, the input to the RS is a user-item rating matrix and a recommender system can be seen as a function that returns a rating prediction (or relevance score) for a given user-item combination. In Figure 2, the problem for example consists of predicting the rating of User1 for Item4.

Over the last fifteen years, a variety of algorithms have been proposed to calculate these relevance scores from the known ratings, starting with from nearest-neighbour methods over probabilistic approaches to recent matrix factorization algorithms[1].

	Item1	Item2	Item3	Item4
User1	5	5	5	?
User2	4	3	5	5
User3	2	1	-	2
User4	4	-	4	5

Figure 2. CF based on single ratings.

In multi-criteria rating settings, however, we assume that we know more about the user's preferences than just his overall evaluation for the item. In particular, we assume that the user has additionally expressed his opinion on several different aspects of the item, e.g., the friendliness of the hotel staff or the tidiness of the rooms. The general idea of multi-criteria is thus to take that additional information into account in the recommendation process. If we, for example, know that the room tidiness is particularly important for a customer (by analysing the relation between the overall rating and the rating for tidiness), we could try to recommend more hotels which have received a high rating for tidiness also by other customers.

2.2 Aggregation function based multi-criteria RS

The most precise approaches to generate recommendations based in multi-criteria ratings presented in the study by Adomavicius and Kwon are called "aggregation function based". The main goal of such approaches is to determine how the individual criteria ratings r_1 to r_k are related to the overall rating r_0 of an item and use that information as a prediction function for the overall rating as shown in Equation (1).

$$r_0 = f(r_1, \ldots, r_k) \tag{1}$$

In the movie domain, this for example would mean to learn a function that describes in which way the ratings for the "story" and other aspects are related to the overall rating. In (Adomavicius and Kwon, 2007), the authors propose to determine the aggregation function by learning a regression function of the following form for each item, where the w_i's are the learned weights for the dimensions r_i and c is a constant.

$$r_0 = w_1 * r_1 + w_2 * r_2 + \ldots + w_k * r_k + c \tag{2}$$

Once the (estimated) values of the function are known, the overall rating for item i and user u can be predicted by parameterizing the regression function for item i with u's ratings for the different criteria. The problem of course is that u's criteria ratings

[1] See (Jannach et al., 2010) for a recent overview.

for item *i* are also not known. In (Adomavicius and Kwon, 2007), it is therefore proposed to predict those dimension ratings for a given user-item combination using, for example, standard CF methods in a first phase. For the Yahoo!Movies (http://movies.yahoo.com) data set used in their experiments, they for example use a nearest-neighbour approach to predict *u*'s ratings for the dimensions "action", "story" and so forth. We will follow this approach also in our experimental study.

Figure 3 illustrates the approach. The entry 5(4,3,4,5) in the table, for example, denotes that the user gave an overall rating of 5 and the criteria ratings 4, 3, 4 and 5 respectively. In the first phase, the "neighbours" for User1 with respect to the last rating dimension are determined (let us assume User2 and User4). Their predictions are then used to predict the rating for the last dimension of Item4, for which we also seek an overall recommendation. Once all criteria ratings are determined, the overall rating is estimated based on the learned regression function from Equation (2).

Phase 2

	Item1	Item2	Item3	Item4
User1	5(4,3,4,5)	3(3,3,4,5)	5(4,5,5,4)	?(?,?,?,?)
User2	4(4,3,3,4)	3(3,3,4,5)	5(4,4,5,5)	5(5,5,4,5)
User3	2(2,3,3,1)	1(1,1,2,1)	-	2(2,1,2,1)
User4	4(4,5,4,2)	-	4(4,3,3,4)	5(5,5,4,3)

Phase 1

Figure 3. Aggregation function based approach.

2.3 Proposed enhancements

We propose the following improvements and extensions compared with previous works.

- *Use a weighted combination of user and item models.* In (Adomavicius and Kwon, 2007), aggregation functions as shown in Equation (2) are learned per movie. The goal is therefore to determine the importance of the individual dimensions for the overall rating for each movie across all users. However, it is also possible to learn such a function also per user, that is, we could try to learn how important the "action" aspect is for a particular user across all movies. Since we assume that both regression functions carry valuable information, we propose in this work to combine the two predictions obtained from the different models in a weighted approach.
- *Use support vector regression.* Instead of using least squares regression as done in previous works, we propose to use Support Vector (SV) regression to calculate the regression functions, because this method has shown to be more accurate than other regression methods also in other recommender applications (Sen et al., 2009).

- *Selection of rating dimensions.* In our real-world data set from the tourism domain used in this study, customer ratings are available for more than a dozen different aspects for each hotel. Our hypothesis is that not all rating information is equally valuable to predict the overall rating and that some of the ratings might be better predictors than others. We will therefore evaluate different settings in which we intentionally omit parts of the available rating information.

3 Empirical evaluation

We conducted several offline experiments on a data set provided by HRS.com, a major European platform for hotel reservations, and compared the predictive accuracy of our newly proposed techniques with top-performing, recent recommendation algorithms. In order to demonstrate the general usability of the weighted regression approach also for other domains, we ran further experiments on a data set obtained from Yahoo!Movies, which also contains multi-dimensional ratings and which was also used in the studies by Adomavicius and Kwon.

3.1 Data sets

Hotel ratings. In order to analyse how the density of the rating information affects the recommendation accuracy, we created variations of the original data set. To that purpose, we varied the constraints *minimum number of ratings per user* and *item* (see the columns *R/U* and *R/I* respectively in Table 1) on the data set and removed users or items for which not sufficient data was available, thus increasing the density of the data set. Table 1 lists the different quality levels used for evaluation. In the `hotel-high` data set, for example, each user has rated at least 5 hotels and each hotel has at least 5 ratings attached. Note that the hotel rating data set is extremely sparse when compared to other data sets from literature as most users have rated only very few hotels. In the well-known MovieLens data set, for example, each user has rated at least 20 movies, which we consider to be an unrealistic constraint in the tourism domain.

Beside the variations with respect to the rating sparsity, we also varied the set of dimension ratings to be taken into account for the prediction task. Table 2 shows an overview of the characteristics of the used data sets, which differ with respect to the considered subset of dimension ratings.

Dataset/Constraint	*R/U*	*R/I*
`hotel-high`	5	5
`hotel-medium`	3	3
`hotel-low`	1	1

Table 1. Hotel data set quality levels.

Overall, the described density variations listed in Table 1 and the variation with respect to the criteria ratings listed in Table 2 lead to a total number of 12 data set variations which we analysed in our evaluation.

Name	**Description**
`dim-all`	All dimensions are taken into account.
`dim-chi-squared`	The most relevant dimensions according to a chi-squared statistic test are used.
`dim-price-performance`	Only the price-performance dimension is used.
`dim-domain-knowledge`	A set of seven dimensions determined by a domain expert are used.

Table 2. Different subset combinations of the rating dimensions in the hotel data set.

In order to determine the most relevant dimensions for the `dim-chi-squared` data set, we computed the value of the chi-squared statistic with respect to the overall rating for each dimension. We retained the 14 dimensions with the highest weights since the weight of the next dimensions was significantly lower. In the `dim-price-performance` data set, we based the prediction of the overall rating only on the customer's rating on the price performance ratio aspect of the hotel. Note that with the emergence of Web-portals, the hotel market became very transparent, open, and competitive. Our hypothesis is thus that the rating on the price performance ratio strongly correlates with the overall rating. For the `dim-domain-knowledge` data set, finally, the 7 dimensions taken into account were selected based on domain expertise. Note that four of them were also in the top-7 list determined based on chi-squared statistics.

Yahoo!Movies data set. Based on the movie rating data set provided by Yahoo! Research, we collected additional criteria rating information from the Yahoo!Movie web site, where users can rate the available movies along four different dimensions (Acting, Story, Visuals, Directing). In order to make our results comparable to previous research, we applied the constraint that each user and item must have at least 10 ratings. We refer to the resulting data set as `YM-10-10`.

3.2 Algorithms compared in the evaluation

Single-rating baseline algorithms.

- `slope-one`: Slope One is a family of prediction schemes proposed in (Lemire and Maclachlan, 2005). It was designed to be an algorithm which is comparably easy to implement, supports the incorporation of new ratings, has a reasonable accuracy and is efficient at run-time. Slope One predictors have the form f(x) = x + b and are based on rating differences between items. In our experiments we used the (basic) Slope One scheme because the results are on a par with slower memory-based schemes such as nearest-neighbour approaches often used in comparisons.

- `funk-svd`: Over the last years, matrix factorization techniques have shown to be a good basis to develop highly accurate recommender systems. In these approaches, the goal is to automatically identify a set of latent semantic features (aspects or factors) which characterize the available items using for example Singular Value Decomposition (SVD) or probabilistic approaches. Predictions can then be made by determining and combining the information about the position of each item and user interest profile in this reduced feature space. In our evaluation, we used an algorithm recently proposed in the context of the Netflix Prize by Simon Funk (pen name)[2]. For the SVD-based recommender, we used 30 latent features, a number which we determined empirically.

Multi-criteria algorithms.

- `item-based SV regression`: This algorithm corresponds to the idea of item-based regression models as proposed in (Adomavicius and Kwon, 2007). In our work, however, we use Support Vector (SV) regression to learn the regression function.
- `user-based SV regression`: The regression model in which we learn the coefficients for each user. Again we use SV regression.
- `weighted SV regression`: This method combines the estimates of item- and user-based SV regression using the harmonic mean of the two estimates which yielded the best results in our evaluation.
- `item-based lin-regression`: This algorithm is similar to the original regression approach proposed in (Adomavicius and Kwon, 2007) and relies on Ordinary Least Squares regression instead on SV regression.

Note that customers in some cases only gave ratings for some but not all dimensions. For the multi-criteria algorithms, we approximated these missing ratings by taking the average of the user's other dimension ratings for an item.

For the SV regression methods, we furthermore used the empirically determined penalty parameter $c = 0.15$ and $p = 0.1$ for epsilon in the loss function. Note that we also made experiments with the Ordinary Least Squares regression algorithm `item-based lin-regression` on the hotel data set. This method is however not applicable for sparse data sets with many rating dimensions as the method requires that the number of data points is higher than the number of regression coefficients. Note that in the hotel data set we have more than a dozen criteria but most hotels and users have much less than 12 ratings. Experiments on the Yahoo!Movies data set, however, showed that SV regression outperforms the Least Squares regression approach.

All algorithms were implemented in our Java-based *Recommender Suite* framework, which also includes components to run offline experiments, do cross-validation and

[2] http://sifter.org/˜simon/journal/20061211.html

measure various metrics such as accuracy or coverage. For the SV regression calculations we used the `libsvm` library[3].

Evaluation Metrics. We rely on standard metrics to evaluate the accuracy of our method. For determining the *Root Mean Squared Error* (RMSE) the available rating information was split into training data (95%) and test data (5%). In order to factor out effects of randomness, we used random subsampling, repeated the experiments appropriately and report the average value of all runs.

Beside the RMSE, we also report the standard Information Retrieval (IR) metrics precision, recall, and F1 and rely on the procedure described in (Nakagawa and Mobasher, 2003). In this approach, the numerical rating predictions are transformed into binary "like" and "dislike" statements. In order to measure precision and recall, we compare the number of existing like statements (ELS) in the test set with the number of predicted like statements (PLS) returned by the algorithms. Precision and recall are measured as follows: $\text{Precision} = \frac{|PLS \cap ELS|}{|PLS|}$; $\text{Recall} = \frac{|PLS \cap ELS|}{|ELS|}$.

F1 is calculated as the harmonic mean of the two. When determining these IR metrics, we used five-fold cross-validation.

3.3 Results and observations

Accuracy results. Tables 3 to 6 show the RMSE numbers for the different hotel data sets. Note that the overall rating (indicated by the user-specified general "feel good factor") in the hotel data sets is given on a 1-to-3 rating scale.

Algorithm	**hotel-high**	**hotel-medium**	**hotel-low**
`weighted SV regression`	0.561	0.575	0.640
`item-based SV regression`	0.613	0.614	0.722
`funk-svd`	0.633	0.640	0.668
`user-based SV regression`	0.642	0.676	0.683
`slope-one`	0.725	0.718	0.786

Table 3. RMSE results for the hotel data set (`dim-all`).

The following observations were made for all hotel data set variations. Firstly, the results confirm the general value of exploiting multi-criteria ratings for recommender algorithms. Moreover, the herein proposed `weighted SV regression` method significantly outperforms both single-rating methods `funk-svd` and `slope-one` as well as the individual regression approaches throughout all hotel data set variations.

3 http://www.csie.ntu.edu.tw/~cjlin/libsvm

Algorithm	hotel-high	hotel-medium	hotel-low
`weighted SV regression`	0.520	0.528	0.677
`item-based SV regression`	0.581	0.577	0.758
`user-based SV regression`	0.594	0.629	0.757
`funk-svd`	0.632	0.638	0.668
`slope-one`	0.749	0.717	0.789

Table 4. RMSE results for the Hotel data set (`dim-chi-squared`).

Algorithm	hotel-high	hotel-medium	hotel-low
`weighted SV regression`	0.521	0.533	0.640
`item-based SV regression`	0.576	0.578	0.768
`user-based SV regression`	0.581	0.614	0.694
`funk-svd`	0.634	0.638	0.653
`slope-one`	0.725	0.723	0.788

Table 5. RMSE results for the Hotel data set (`dim-price-performance`).

Algorithm	hotel-high	hotel-medium	hotel-low
`weighted SV regression`	0.515	0.528	0.533
`item-based SV regression`	0.565	0.582	0.758
`user-based SV regression`	0.587	0.610	0.757
`funk-svd`	0.633	0.638	0.668
`slope-one`	0.721	0.718	0.789

Table 6. RMSE results for the Hotel data set (`dim-domain-knowledge`).

The individual regression-based approaches alone are also able to outperform the single-rating approaches except for the setting in which all dimensions are taken into account (Table 3) and where user-based regression works slightly worse than `funk-svd`. Item-based regression on the other hand is consistently better than the single-rating approaches in all settings and dataset variations.

Note that even though the weighted approach is already the best performing one, we see potential for further gains with respect to accuracy, e.g., by optimizing the parameter settings for the regression learner or by combining the user- and item-based predictions in a different way.

We can see from the evaluation data that increasing the quality level by adding constraints on the minimum number of ratings per user and item the predictive accuracy increases. Intuitively, the RMSE values thus decrease when more rating information per user and item is available. Note however, that the different data sets are not fully comparable with respect to the number of hotels and items. The "hotel-high"-data sets are for example derived from the same raw data set as the "hotel-medium"-ones, which means that they comprise less hotels, items and ratings. Still, we can observe that the RMSE values are significantly better in the "hotel-high" data set even though less training data was available.

Table 7 shows the results for the standard IR metrics precision, recall, and F1 on the hotel data sets `hotel-high` and `hotel-low` (`dim-chi-squared`). We can see that the results for the IR metrics are comparable to the RMSE results reported above and that our weighted approach outperforms the other algorithms also on this measure. In that context, note that the single-rating method `slope-one` performs extremely poor on the highly sparse data set.

Algorithm	**hotel-high**			**hotel-low**		
	F1	Prec.	Recall	F1	Prec.	Recall
`weighted SV regression`	93.47	93.47	93.47	70.97	73.24	68.84
`item-based SV regression`	92.97	92.97	92.97	69.51	71.79	68.74
`user-based SV regression`	92.68	92.68	92.68	70.87	73.14	67.39
`funk-svd`	92.53	92.53	92.53	69.23	71.51	67.10
`slope-one`	46.13	49.29	43.40	08.31	10.17	07.03

Table 7. F1, precision, and recall results for the hotel data set `dim-chi-squared`.

Impact of feature selection. The tables showing the RMSE values (Table 3 to Table 6) also highlight the importance of choosing an adequate subset of dimensions as this selection can significantly affect the performance of a multi-criteria recommender algorithm. However, the results we obtained so far are not conclusive and will be analysed in more detail as a part of our future work.

Our results indicate that feature selection based on statistics or machine learning paid off for the high quality data set (`hotel-high`). On the other hand, for the low-quality data set (`hotel-low`), which is more realistic in real world scenarios, feature selection by a human expert with domain knowledge was the most successful approach. Based on the observation that the price-performance ratio is one of the most important factors in the domain and overpriced hotels are quickly penalized with low reviews by customers, we can see from the results in Table 5 that relying only on this detailed rating information can lead to comparably good results.

Finally, as mentioned above, the user-based regression approach does not work better than `funk-svd` when all available dimension ratings are taken into account (Table 3). This again stresses the importance of selecting the most important features for the prediction process.

Accuracy results for the movie domain. We conclude this section with a discussion of the results obtained on the Yahoo!Movies data set. These experiments have been conducted to determine whether or not the weighted regression approach also works in domains other than hotel recommendation. Note that in this data set, the overall ratings were originally given on an A+ to F score and have been transformed to the usual 1-to-5-scale. Therefore, the RMSE values have to be interpreted differently than the values for the hotel data set.

Algorithm	**RMSE**	**F1**	**Precision**	**Recall**
`weighted SV regression`	0.625	90.69	90.69	90.69
`user-based SV regression`	0.652	88.18	88.18	88.18
`item-based SV regression`	0.694	89.93	89.93	89.93
`item-based lin-regression`	0.779	88.14	88.14	88.14
`funk-svd`	0.871	83.29	83.29	83.29
`slope-one`	0.888	82.64	82.64	82.64

Table 8. Accuracy results for the `YM-10-10` data set.

Table 8 shows the accuracy results (RMSE, Precision, Recall) for the different algorithms. Similar to the hotel domain, the weighted regression approach performs best on all metrics. In contrast to the hotel data sets, however, all regression-based approaches are consistently better than the best single-rating based approach `funk-svd`. A comparison with the Least Squares regression method `item-based lin-regression` also indicates that SV regression is favourable for multi-criteria recommendation also for this domain.

4 Conclusion and outlook

In contrast to other traditional application domains of recommender systems, multidimensional ratings play an important role on today's online tourism platforms. In this work, we have proposed to use Support Vector regression and the combination of user-based and item-based model to implement highly accurate multi-criteria recommenders. We analysed the predictive accuracy of our method in the domain of hotel recommendation based on a real-world data set. Our experiments showed that in particular the weighted combination of user- and item-based prediction models leads to the best performance measured in terms of standard accuracy metrics and that the results are better than those that can be achieved with state-of-the-art matrix factorization methods.

Beside the analysis of the accuracy for different density levels, we also ran experiments in which we varied the number of rating dimensions to be taken into account for the recommendation process. Although we could not develop a final guidance of how to select the best subset of dimensions, our initial results show that the selection measurably influences the recommendation accuracy. Our future work therefore includes the investigation of additional strategies of selecting the best subset of dimensions for the recommendation task. In addition, we plan to develop other schemes of combining the predictions returned by the user- and the item-based approach, e.g., based on the aspect of model quality. Currently, we are investigating an approach in which we try to learn individual weights for the item- and user-based component for each item and user as was done in (Gedikli et al., 2011).

An analysis of whether or not the existing information about the context of the customer (e.g., is it a business trip or a private travel) can be exploited to further increase the accuracy of the recommendations is part of our on-going work. In addition, future work in the area could also include the design of new approaches to extract additional customer opinions (on different aspects) from the free-text reviews. A first review of the data from a real-world platform in that context however revealed that most probably specific new algorithms are required as standard sentiment analysis methods might have difficulties in analysing text reviews that in many cases only consist of fragmentary sentences and individual keywords.

References

Adomavicius, G. and Kwon, Y. (2007). New recommendation techniques for multicriteria rating systems. *IEEE Intelligent Systems*, 22:48–55.

Adomavicius, G., Manouselis, N., and Kwon, Y. (2011). Multi-criteria recommender systems. In Ricci, F., Rokach, L., Shapira, B., and Kantor, P. B., editors, *Recommender Systems Handbook*, pages 769–803. Springer US.

Baltrunas, L., Ludwig, B., Peer, S., and Ricci, F. (2011). Context-aware places of interest recommendations for mobile users. In Proc. HCI 2011), pages 531–540, Orlando, FL.

Drucker, H., Chris, Kaufman, B. L., Smola, A., and Vapnik, V. (1997). Support vector regression machines. *In Adv. in Neural Inf. Proc. Systems*, volume 9, pages 155–161.

Gedikli, F., Bagdat, F., Ge, M., Jannach, D.: RF-Rec (2011). Fast and Accurate Computation of Recommendations based on Rating Frequencies, Proc. *IEEE CEC 2011*, Luxembourg.

Jannach, D., Zanker, M., Felfernig, A., and Friedrich, G. (2010). *Recommender Systems – An Introduction*. Cambridge University Press.

Jannach, D., Zanker, M., and Fuchs, M. (2009). Constraint-based recommendation in tourism: A multiperspective case study. International Journal of Information Technology and Tourism, 11(2):139–155.

Jannach, D., Zanker, M., Jessenitschnig, M., and Seidler, O. (2007). Developing a conversational travel advisor with ADVISOR SUITE. In Proceedings *ENTER 2007*, pages 43–52, Ljubljana, Slovenia.

Lemire, D. and Maclachlan, A. (2005). Slope one predictors for online rating-based collaborative filtering. In Proc. *SIAM Intl. Conf. on Data Mining*, pages 471–480, Newport Beach, CA.

Mahmood, T., Ricci, F., and Venturini, A. (2009). Improving recommendation effectiveness: Adapting a dialogue strategy in online travel planning. *International Journal of Information Technology and Tourism*, 11(4):285–302.

Nakagawa, M. and Mobasher, B. (2003). A hybrid web personalization model based on site connectivity. In Proceedings of the *Workshop on Web Mining and Web Usage Analysis (WebKDD'03)*, pages 59–70, Washington, DC, USA.

Ricci, F. (2011). Mobile recommender systems. *International Journal of Information Technology and Tourism*, 12(3):205–231.

Sen, S., Vig, J., and Riedl, J. (2009). Tagommenders: Connecting users to items through tags. In Proc. *WWW'09*, pages 671–680, Madrid, Spain.

Zanker, M., Fuchs, M., Höpken, W., Tuta, M., and Müller, N. (2008). Evaluating recommender systems in tourism - a case study from Austria. In Proceedings ENTER 2008, pages 24–34, Amsterdam, Netherlands.