Including Item Characteristics in the Probabilistic Latent Semantic Analysis Model for Collaborative Filtering

Martijn Kagie, Matthijs van der Loos and Michiel van Wezel

ERIM REPORT SERIES *RESEARCH IN MANAGEMENT*	
ERIM Report Series reference number	ERS-2008-053-MKT
Publication	August 2008
Number of pages	21
Persistent paper URL	http://hdl.handle.net/1765/13180
Email address corresponding author	kagie@few.eur.nl
Address	Erasmus Research Institute of Management (ERIM) RSM Erasmus University / Erasmus School of Economics Erasmus Universiteit Rotterdam P.O.Box 1738 3000 DR Rotterdam, The Netherlands Phone: + 31 10 408 1182 Fax: + 31 10 408 9640 Email: info@erim.eur.nl Internet: www.erim.eur.nl

Bibliographic data and classifications of all the ERIM reports are also available on the ERIM website:
www.erim.eur.nl

ERASMUS RESEARCH INSTITUTE OF MANAGEMENT

REPORT SERIES

RESEARCH IN MANAGEMENT

ABSTRACT AND KEYWORDS	
Abstract	We propose a new hybrid recommender system that combines some advantages of collaborative and content-based recommender systems. While it uses ratings data of all users, as do collaborative recommender systems, it is also able to recommend new items and provide an explanation of its recommendations, as do content-based systems. Our approach is based on the idea that there are communities of users that find the same characteristics important to like or dislike a product. This model is an extension of the probabilistic latent semantic model for collaborative filtering with ideas based on clusterwise linear regression. On a movie data set, we show that the model is competitive to other recommenders and can be used to explain the recommendations to the users.
Free Keywords	Algorithms, Recommender Systems, Probabilistic Latent Semantic Analysis, Hybrid Recommender Systems
Availability	
Classifications	

Including Item Characteristics in the Probabilistic Latent Semantic Analysis Model for Collaborative Filtering

MARTIJN KAGIE, MATTHIJS VAN DER LOOS and MICHIEL VAN WEZEL
Erasmus University Rotterdam

We propose a new hybrid recommender system that combines some advantages of collaborative and content-based recommender systems. While it uses ratings data of all users, as do collaborative recommender systems, it is also able to recommend new items and provide an explanation of its recommendations, as do content-based systems. Our approach is based on the idea that there are communities of users that find the same characteristics important to like or dislike a product.

This model is an extension of the probabilistic latent semantic model for collaborative filtering with ideas based on clusterwise linear regression. On a movie data set, we show that the model is competitive to other recommenders and can be used to explain the recommendations to the users.

Categories and Subject Descriptors: H.3.5 [**Information Systems**]: Online Information Systems—*Web-based services*

General Terms: Algorithms

Additional Key Words and Phrases: Recommender Systems, Probabilistic Latent Semantic Analysis, Hybrid Recommender Systems.

1. INTRODUCTION

Nowadays, a lot of electronic commerce companies assist their customers online in buying their product using recommender systems [Resnick and Varian 1997]. In general, two different types of these systems can be identified.

A large group of recommender systems, so-called collaborative filtering systems [Goldberg et al. 1992], recommend items based on similar preferences between users. When customers (henceforth called users) A and B liked the same products (henceforth called items) in the past, collaborative systems assume that a good item to recommend to A would be an item that user B appreciated, but which A is unfamiliar with. Often, this idea is incorporated in the collaborative filtering method by modelling communities of users having similar taste. A popular collaborative filtering algorithm with a strong probabilistic foundation that uses this idea is the probabilistic latent semantic analysis model for collaborative filtering (pLSA-CF) [Hofmann 2004]. A disadvantage of these collaborative type recommender systems is that they neglect item characteristics, such as genre and keywords for a movie, when making recommendations, whereas these may carry useful information. Another disadvantage is that handling new items is difficult, since new items have not been appreciated by anyone yet. This is called the 'cold start item problem'.

In contrast, content-based recommender systems [Pazzani and Billsus 2007], use only characteristics of the items to provide a recommendation, and recommend

Author's address: Econometric Institute, Erasmus University Rotterdam, PO Box 1738, 3000DR Rotterdam, The Netherlands; e-mail: {kagie,mvanderloos}@few.eur.nl, mvanwezel@acm.org.

items that are similar, based on these characteristics, to items that the user liked in the past. The advantage that this approach has over collaborative methods, is that it is also able to recommend items that are not yet rated by any user, since item characteristics are known upfront. A disadvantage is that such systems only use earlier ratings from the active user when providing a recommendation, whereas ratings by other users are discarded. (The active user is the user receiving the recommendations.)

To combine some of the strengths of both approaches, hybrid recommender systems have emerged [Burke 2002], which integrate collaborative and content-based techniques. Section 2 discusses these systems in more detail. In this paper, we introduce a new hybrid recommendation approach which exploits the idea that communities of users exist which find the same item characteristics important to like or dislike a product.

The basis of our method is pLSA-CF [Hofmann 2004] mentioned above. pLSA-CF attempts to find 'hidden' user communities that rate items similarly. For each community, pLSA-CF estimates expectations of the ratings for all items. For a specific user, the expected rating for an item is computed as a weighted average over the community ratings for that item. The weighting coefficients, which are also estimated, express to what degree a user belongs to each community. We will briefly describe the pLSA-CF model in Section 3.1 below.

We extend pLSA-CF by conditioning the expected ratings on item characteristics. This is achieved using a separate linear regression function that models the relationship between item characteristics and ratings for each user community. For each user, an individual regression function can then be constructed by taking a weighted average over the the community regression functions. This idea borrows from, but is not identical to so-called clusterwise linear regression (CLR) [DeSarbo and Cron 1988]. We discuss CLR and our extension to pLSA-CF in Sections 3.2 and 3.3 below.

Our model, which we call the latent class regression recommender system (LCR-RS), is evaluated on a movie recommendation data set. This data set was constructed by combining two well-known databases: The Netflix database[1], which contains user ratings of movies, and the IMDb database[2], which contains movie characteristics. We compare prediction performance of our model with a set of standard models and pLSA-CF.

There are a number of potential advantages that our method has over pLSA-CF: (1) Since the regression coefficients are shared among items, it contains far fewer parameters, making it less susceptible to overfitting; (2) The regression coefficients add to the interpretability since they can help in explaining *why* the system thinks that items are highly/poorly rated; (3) Since items are recommended based on characteristics, our method does not suffer from the cold-start item problem.

The remainder of the paper is organized as follows. In the next section, we discuss recommender systems with an emphasis on hybrid recommendation approaches. Section 3 describes pLSA-CF, CLR, and LCR-RS. In Section 4, we describe the movie data that is used in the experiments that are shown in Section 5. Finally,

[1] `http://www.netflix.com`
[2] `http://www.imdb.com`

we draw conclusions and give directions for future research in Section 6.

2. RECOMMENDER SYSTEMS

As mentioned in the introduction, there exist three types of recommender systems [Prasad 2003; Adomavicius and Tuzhilin 2005]: Collaborative filtering (CF) [Goldberg et al. 1992], content-based [Pazzani and Billsus 2007], and hybrid systems [Burke 2002], combining these latter two approaches.

Breese et al. [1998] and Adomavicius and Tuzhilin [2005] make a distinction between two types of collaborative filtering systems. The first type are the memory-based recommenders, which use the entire user-item ratings matrix each time recommendations are computed. Often, these methods compute similarities between users based on their rating behavior and then apply some nearest neighbor approach to provide recommendations, an idea first applied in the GroupLens system [Resnick et al. 1994]. Similarity between items based on the ratings they received can also be used. This technique is called item-based CF [Sarwar et al. 2001] and is applied at e.g., Amazon.com [Linden et al. 2003]. Model-based systems on the other hand, fit a model which provides the recommendations. The ratings matrix is only used during model fitting, not during deployment. Popular model-based collaborative recommenders include EigenTaste [Goldberg et al. 2001], latent Dirichlet allocation [Blei et al. 2003], and probabilistic latent semantic analysis for collaborative filtering [Hofmann 2004], which is the basis of our algorithm.

Content-based recommenders make recommendations based on the characteristics of items that the user has liked in the past. Also in this kind of systems, which originate from information retrieval, nearest neighbor methods have been very popular. When the items at hand are text documents, the term frequency - inverse document frequency is often used to construct the characteristics vector of the items [Adomavicius and Tuzhilin 2005]. Besides the nearest neighbor method, Pazzani and Billsus [1997] show that also other classifiers as naive Bayes, decision trees, and neural networks can be used as content-based recommender systems.

Hybrid recommender systems combine content-based and collaborative filtering methods. Adomavicius and Tuzhilin [2005] made a distinction between four types of hybrid combinations of content-based and collaborative filtering systems.

The most basic hybrid recommender systems combine two separate recommender systems, a content-based and a collaborative filtering system. The final recommendation is then based on a linear combination [Claypool et al. 1999] or a voting scheme [Pazzani 1999]. Also, a quality metric can be used to select the best prediction of the systems as is done by, for instance, Billsus and Pazzani [2000] and Tran and Cohen [2000].

Another type of hybrid recommenders includes content-based characteristics to collaborative filtering methods. A way in which this can be done, is, for instance, by augmenting the vector of user ratings with additional ratings, that are computed using a content-based method as is done by Melville et al. [2002]. Similar approaches were taken by Balabanović and Shoham [1997], Good et al. [1999], and Pazzani [1999]. Soboroff and Nicholas [1999] took the opposite approach by adding collaborative characteristics, using latent semantic indexing [Deerwester et al. 1990], to a content-based recommender system.

Finally, there is a group of hybrid recommender systems in which content-based and collaborative filtering are combined into one model. This has be done, for example, using rule-based classifiers [Basu et al. 1998] or Bayesian mixed-effect regression models [Condliff et al. 1999; Ansari et al. 2000]. Popescul et al. [2001] and Schein et al. [2002] integrated item characteristics (words contained in a document) in the original probabilistic latent semantic analysis model for binary targets (observed – not observed) [Hofmann 1999; 2001].

In our method, which we discuss in detail in Section 3.3, we integrate both methods in a different way. Where in collaborative filtering systems similar users are determined by similar rating patterns, we define similar users as users who find the same characteristics of an item important and, thus, incorporate content-based information in the model. Since our approach is an extension of the probabilistic latent semantic analysis model for collaborative filtering (pLSA-CF) [Hofmann 2004], we first discuss this model in Section 3.1.

2.1 Explanations in Recommender Systems

Although providing good recommendations to users is essential for a recommender system, users are more likely to accept the recommendations of the system, when it gives a good explanation why it gives certain recommendations [Herlocker et al. 2000]. In an overview of explanation strategies in recommender systems, Tintarev and Masthoff [2007] distinguish seven aims for explanations in recommender systems. In our approach, we will focus on the aim of transparency, that is, explaining why a certain recommendation is given by the system, an approach which is, for example, taken by McSherry [2005]. In a small user study, Sinha and Swearingen [2002] showed that users liked transparent recommender systems over nontransparent ones. In Section 3.3, we show how our system can be used to provide transparency on how it provides its recommendations.

3. MODELS

This section describes the latent class regression recommender system (LCR-RS). Before describing LCR-RS itself, we first describe the models it is built on: the pLSA-CF model for collaborative filtering and the CLR model for clusterwise regression.

3.1 Probabilistic Latent Semantic Analysis for Collaborative Filtering

pLSA-CF, introduced in [Hofmann 2004], models unobserved (latent) classes of users which give similar ratings to items. To discuss this model more formally we introduce some notation. Consider a set of items $\mathcal{Y} = \{y_1, \ldots, y_n\}$, a set of users $\mathcal{U} = \{u_1, \ldots, u_m\}$, and a set of numerical ratings $\mathcal{V}$. A typical collaborative data set contains N entries $\langle u, y, v \rangle$, that is, ratings v of a certain item y by a certain user u. However, this set of entries is not complete, since by far not all items are rated by each user.

pLSA-CF assumes that there is also a set $\mathcal{Z} = \{z_1, \ldots, z_K\}$ of latent classes of users with similar preferences. This dependency structure is depicted in Figure 1. Intuitively, these classes can be interpreted as communities of users with similar preferences. (An example could be the 'Trekkies', interested in science fiction.) Since these classes are unobserved we do not know to which class a user belongs.

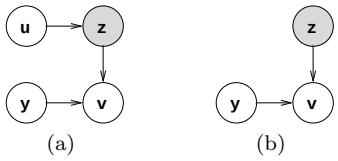

Fig. 1. Graphical model representations of 1(a): pLSA-CF model and 1(b): Gaussian mixture model.

However, we do know that the fact that a user belongs to a certain class influences his ratings. Although class memberships are unobserved, we can still estimate class membership probabilities using the EM algorithm [Dempster et al. 1977], as we will see shortly.

First, consider the joint probability density function, including the hidden variable z

$$P(u, v, y, z) = P(u)P(y)P(z|u)P(v|z, y) \tag{1}$$

and the marginal density, obtained by summing over all possible values of z

$$P(u, v, y) = P(u)P(y)\sum_{k=1}^{K} P(z_k|u)P(v|z_k, y) \ . \tag{2}$$

$P(u)$ and $P(y)$ are constants, depending on the number of votes by from user u and for item y. They will be discarded henceforth. $P(z|u)$ denotes the class membership probabilities for user u. For each k, $P(z_k|u)$ represents the probability that user u will behave according to class z_k when making a vote. $P(z_k|u)$ is just a scalar and $\sum_{k=1}^{K} P(z_k|u) = 1$. In the pLSA-CF model, $P(v|z_k, y)$ is taken to be Gaussian

$$P(v|z_k, y) = P(v|\mu_{k,y}, \sigma_{k,y}) = \mathcal{N}(v|\mu_{k,y}, \sigma_{k,y}) = \frac{1}{\sqrt{2\pi}\sigma_{k,y}} \exp\left[-\frac{(v-\mu_{k,y})^2}{2\sigma_{k,y}^2}\right] \ . \tag{3}$$

This Gaussian distribution, with a class/item specific mean and standard-deviation, specifies the distribution of votes given that a user votes for item y and behaves according to class z_k. In total there are $K \times (m + 2n)$ parameters in the model. These parameters are collectively denoted θ.

The joint density thus becomes

$$P(u, v, y, z|\theta) \propto \prod_{k=1}^{K} P(z_k|u)^{I(z=z_k)} \mathcal{N}(v|\mu_{k,y}, \sigma_{k,y})^{I(z=z_k)} \ , \tag{4}$$

where $I()$ denotes the indicator function returning 1 if its argument is true, 0 otherwise. The marginal density (for the observed data) is

$$P(u, v, y|\theta) \propto \sum_{k=1}^{K} P(z_k|u)\mathcal{N}(v|\mu_{k,y}, \sigma_{k,y}) \ . \tag{5}$$

Although this seems similar to a Gaussian mixture model, there is an important

difference: The mixture proportions are user-dependent. A GMM lacks this dependency, as can be seen from Figure 1.

Based on the observed data, the log-likelihood is

$$L(\theta) = \sum_{\langle u,v,y \rangle} \log \left\{ \sum_{k=1}^{K} P(z_k|u) \mathcal{N}(v|\mu_{k,y}, \sigma_{k,y}) \right\} \quad . \tag{6}$$

This function should be optimized with respect to the parameters θ. Unfortunately, this is difficult because of the summation inside the logarithm.

Consider instead the complete data log-likelihood

$$L^c(\theta) = \sum_{\langle u,v,y,z \rangle} \sum_{k=1}^{K} I(z = z_k) \left\{ \log P(z_k|u) + \log \mathcal{N}(v|\mu_{k,y}, \sigma_{k,y}) \right\} \quad . \tag{7}$$

This function is fairly easy to optimize, because the logarithm appears inside the summation. Unfortunately, the z are unobserved, so direct optimization of this likelihood is impossible since the values of the indicator function are unknown. We can, however, compute the expected value of the indicator function for each k, given certain values for the parameters θ and observation $\langle u, v, y \rangle$. This would then amount to the posterior probability that $\langle u, v, y \rangle$ was generated by class k given the parameters θ. This is done in the expectation step of the EM-algorithm, where we estimate $E[I(z = z_k)|u, y, v, \theta] = P(z_k|u, y, v, \theta)$ [Hofmann 2004]

$$P(z_k|u, y, v, \theta) = \frac{P(z_k|u) P(v|\mu_{k,y}, \sigma_{k,y})}{\sum_{k'=1}^{K} P(z_{k'}|u) P(v|\mu_{k',y}, \sigma_{k',y})} \quad . \tag{8}$$

These expectations can now be plugged into Equation (7) giving the expected likelihood

$$E_z[L^c(\theta)] = \sum_{\langle u,v,y \rangle} \sum_{k=1}^{K} P(z_k|u, y, v, \theta) \left\{ \log P(z_k|u) + \log \mathcal{N}(v|\mu_{k,y}, \sigma_{k,y}) \right\} \quad . \tag{9}$$

This function is then minimized w.r.t. θ in the minimization step of the EM algorithm. This leads to the following update equations for the user-specific mixing weights $P(z_k|u)$ [Hofmann 2004]

$$P(z_k|u)^{\text{new}} = \frac{\sum_{\langle u',y,v \rangle : u'=u} P(z_k|u', y, v, \theta^{\text{old}})}{|\{\langle u', y, v \rangle : u' = u\}|} \tag{10}$$

and for $\mu_{k,y}$ and $\sigma_{k,y}$

$$\mu_{k,y}^{\text{new}} = \frac{\sum_{\langle u,y',v \rangle : y'=y} v P(z_k|u, y', v, \theta^{\text{old}})}{\sum_{\langle u,y',v \rangle : y'=y} P(z|u, y', v, \theta^{\text{old}})} \tag{11}$$

$$\sigma_{k,y}^{2,\text{new}} = \frac{\sum_{\langle u,y',v \rangle : y'=y} (v - \mu_{k,y'}^{\text{new}})^2 P(z_k|u, y', v, \theta^{\text{old}})}{\sum_{\langle u,y',v \rangle : y'=y} P(z_k|u, y', v, \theta^{\text{old}})} \quad . \tag{12}$$

These updated estimates for $P(z_k|u)$, $\mu_{k,y}$, and $\sigma_{k,y}$ are then plugged into Equation (8) and the process continues until convergence, i.e., until the expected log-likelihood (9) does not increase anymore. It can be shown (see e.g., [Bishop 2006]) that this end-point is also a local optimum of the observed data likelihood (6).

Notice that Equation (10) cannot be applied when a user has not cast any votes yet. This can easily be overcome by e.g., adding a term a/K to the numerator and adding a term a to the denominator, for a suitable integer a. This corresponds to assigning equal prior class probabilities to users. The larger a, the slower the actual class probabilities for a user will be learned from ratings data. Alternatively, the constants can be constructed such that prior class membership probabilities for new users match average observed class membership probabilities probabilities over the users with sufficient votes. This would solve the cold start user problem.

After the model has been trained, we can simply predict a new rating as

$$v_{u,y} = \sum_{k=1}^{K} P(z_k|u)\mu_{k,y} \quad . \tag{13}$$

A drawback of the pLSA-CF model is that, like all CF methods, it is unable to predict ratings for items not rated by any user in the system. Therefore, we extend the pLSA-CF model using item characteristics to solve this cold-start item problem. The pLSA-CF model is summarized in Algorithm 1.

Algorithm 1 pLSA-CF algorithm

procedure pLSA_CF(entries $\langle u, y, v\rangle$,K)

▷ Initialization

for all u **do**
Generate random $P(z_k|u)$ such that $\sum_k P(z_k|u) = 1$.
end for
for all y, z_k **do**
Generate random $\mu_{k,y}$.
Generate random $\sigma_{k,y}$.
end for
Set $err = \infty$.

▷ EM-loop

repeat
$prevErr = err$.
for all $\langle u, y, v\rangle, z_k$ **do**
Compute $P(z_k|u, y, v)$ using (8).
end for
for all u, z_k **do**
Compute $P(z_k|u)$ using (10).
end for
for all y, z_k **do**
Compute μ_{y,z_k} using (11).
Compute σ_{y,z_k} using (12).
end for
Compute $err = -E[L^c(\theta)]$ using (9).
until $prevErr - err < \epsilon$
end procedure

3.2 Clusterwise Linear Regression

Often, we can represent a certain item by a vector $\mathbf{x}_y$ containing a set of characteristics (such as keywords and genres in the case of movies). A simple way to build a content-based recommender system, is to estimate a linear regression model. When

we specify $\mathbf{v}_u$ as the set of ratings of user u and matrix $\mathbf{X}_u$ as the matrix containing the characteristic vectors of the items rated by user u, we can estimate user specific regression weights by

$$\mathbf{b}_u = (\mathbf{X}_u'\mathbf{X}_u)^{-1}\mathbf{X}_u'\mathbf{v} \ . \tag{14}$$

A rating for an item unrated by user u then can simply be predicted by

$$E[v|u, y] = \mathbf{x}_y'\mathbf{b}_u \ . \tag{15}$$

Using this simple content-based method we could easily predict ratings for items that have not been rated yet, since we only need the characteristics of the new item. Moreover, it would also be possible to explain why certain recommendations are given using the regression weights. Clearly, these two properties make the regression approach worth considering.

Constructing user-specific regressions may be problematic since we need at least a substantial number of ratings from a user if we want to estimate this user's regression coefficients. As an alternative, one could 'pool' all ratings generated by all user for all items, and estimate a 'population-regression' with this pooled data set. Such a single model is likely to perform poorly, since people have varying preferences.

Hence, an intermediate solution would be not to specify a regression for each user separately or for the total population at once, but to create regression models for different groups of users having similar preferences. We will discuss such a method below in Subsection 3.3. First we discuss method called clusterwise linear regression (CLR) proposed by DeSarbo and Cron [1988]. This method, although not directly suitable for the recommendation problem, forms the basis for the method in Subsection 3.3.

CLR and its extensions are very popular in various areas of marketing, such as conjoint analysis [DeSarbo et al. 1992; Vriens et al. 1996] and estimating marketing mix elasticities [Ramaswamy et al. 1993]. Especially the connection between conjoint analysis and item recommendation is very clear. In conjoint analysis subjects rate potential products which are described by a limited set of product characteristics. Then a model is estimated that explains which product characteristics are important to make a product popular. When a model based on CLR is used, one can differentiate between market segments. Besides in marketing, models based on CLR were also successfully applied in other fields of business research as accounting [Larcker and Richardson 2004], finance [Pennings and Garcia 2004], and strategic management [DeSarbo et al. 2005]. But also in other research disciplines as psychology [Kriegler and Zimprich 2006] and climate research [Camargo et al. 2007], CLR has been used.

In the original CLR the dependent variable (in our case the rating variable v) is modeled as a mix of K conditional univariate densities with cluster-specific regression coefficients and standard deviations

$$P(v|y) = \sum_{k=1}^{K} \lambda_k \mathcal{N}(v|\mathbf{x}_y'\mathbf{b}_k, \sigma_k) \ . \tag{16}$$

CLR is optimized using the EM-algorithm [Dempster et al. 1977]. In the E-step we estimate the posterior probabilities for each vote v that this vote belongs to class

z_k

$$P(z_k|v,y,\theta) = \frac{\lambda_k \mathcal{N}(v|\mathbf{x}_y'\mathbf{b}_k,\sigma_k)}{\sum_{k'=1}^{K}\lambda_{k'}\mathcal{N}(v|\mathbf{x}_y'\mathbf{b}_{k'},\sigma_{k'})} \ , \tag{17}$$

where y denotes the item that vote v concerned. Next, in the M-step, we compute the cluster-weights λ_k

$$\lambda_k = \frac{\sum_{\langle y,v\rangle} P(z_k|y,v,\theta)}{N} \ . \tag{18}$$

Here, the summation is over all item/vote pairs in the dataset. In the rest of the M-step, we estimate the cluster-specific regression weights $\mathbf{b}_k$ and variances σ_k^2 using weighted least squares (WLS). Following Ramaswamy et al. [1993], the estimate for $\mathbf{b}_k$ is given by

$$\begin{aligned} \mathbf{b}_k^{\text{new}} &= (\mathbf{X}'\mathbf{P}_k\mathbf{X})^{-1}\mathbf{X}'\mathbf{P}_k\mathbf{v} \ , \\ &\text{where } \mathbf{P}_k = \text{diag}(P(z_k|v,y,\theta^{\text{old}}))) \end{aligned} \tag{19}$$

and matrix $\mathbf{X}$ contains the characteristics for each item y of all item/value pairs $\langle y,v\rangle$ and vector $\mathbf{v}$ contains the corresponding rating. Following DeSarbo and Cron [1988], the estimate for σ_k^2 is

$$\sigma_k^{2,\text{new}} = \frac{\sum_{\langle y,v\rangle} P(z_k|y,v,\theta^{\text{old}})(v-\mathbf{x}_y'\mathbf{b}_k)^2}{\sum_{\langle y,v\rangle} P(z_k|y,v,\theta^{\text{old}})} \ . \tag{20}$$

Although CLR provides us with a way to model cluster-specific regression coefficients, it has some shortcomings due to which it cannot be directly used for our problem: Contrary to the pLSA-CF model, the class memberships do not depend on the user. Thus, latent classes in this model lack a clear interpretation in terms of user communities. In fact, CLR is an extension of a Gaussian mixture model, which also lacks that ability (see Figure 1). Although we can compute these user communities in CLR afterwards, the model is not optimized for these user communities. Therefore, we integrate the idea of cluster-specific regression coefficients in the pLSA-CF framework with its user specific mixing weights.

3.3 Latent Class Regression Recommender System

As mentioned above our method, which we call the latent class regression recommender system (LCR-RS), integrates the idea of CLR into the pLSA-CF framework. Informally, this implies the following changes w.r.t. pLSA-CF: we replace item means $\mu_{y,k}$ for each latent class z_k by one vector of regression coefficients $\mathbf{b}_k$. Also, the item-specific standard deviations $\sigma_{y,k}$ for each class z_k are replaced by one class-specific standard deviation σ_{z_k}. Hence, the number of parameters becomes independent of the number of items. When there are many items, this could lead to a substantial reduction.

Therefore, we adapt the Gaussian density function (3) to

$$P(v|z_k,y) = P(v|\mathbf{x}_y'\mathbf{b}_k,\sigma_k) = \mathcal{N}(v|\mathbf{x}_y'\mathbf{b}_k,\sigma_k) \ , \tag{21}$$

where $\mathbf{x}_y$ is a vector containing the characteristics of item y. Again, the model is

fitted with EM. The E-step then becomes

$$P(z_k|u,y,v,\theta^{\text{old}}) = \frac{P(z_k|u)^{\text{old}} P(v|\mathbf{x}'_y \mathbf{b}_k^{\text{old}}, \sigma_k^{\text{old}})}{\sum_{k'=1}^{K} P(z_{k'}|u)^{\text{old}} P(v|\mathbf{x}'_y \mathbf{b}_k^{\text{old}}, \sigma_{k'}^{\text{old}})} \ . \tag{22}$$

In the M-step, we first compute new values for the user specific mixture weights $P(z_k|u)$. Because of the dependence on u, this is done using Equation (10) from the pLSA-CF model. Here, the difference w.r.t. CLR becomes clear, where general cluster weights were computed using Equation (18), rather than user-specific ones.

In the remainder of the M-step we compute new values for the regression coefficients $\mathbf{b}_k$, and the standard deviations σ_k. The $\mathbf{b}_k$ are computed using WLS

$$\begin{aligned} \mathbf{b}_k^{\text{new}} &= (\mathbf{X}'\mathbf{P}_k\mathbf{X})^{-1}\mathbf{X}'\mathbf{P}_k\mathbf{v} \ , \\ &\text{where } \mathbf{P}_k = \text{diag}(P(z_k|u,y,v,\theta^{\text{old}})) \ , \end{aligned} \tag{23}$$

where $\text{diag}(P(z|u,y,v,\theta^{\text{old}}))$ is a diagonal matrix containing probabilities $P(z|u,y,v,\theta^{\text{old}})$ for all entries $\langle u,y,v\rangle$, $\mathbf{v}$ contains the ratings v of all entries, and $\mathbf{X}$ contains for all entries the characteristics of the corresponding item y. Finally, we estimate σ_k for each latent class

$$\sigma_k^{2,\text{new}} = \frac{\sum_{\langle u,y,v\rangle} (v - \mathbf{x}_y \mathbf{b}_k)^2 P(z_k|u,y,v,\theta^{\text{old}})}{\sum_{\langle u,y,v\rangle} P(z_k|u,y,v,\theta^{\text{old}})} \ . \tag{24}$$

After convergence, we can use the model to predict the rating for a certain item y by user u

$$E[v|u,y] = \sum_{k=1}^{K} P(z_k|u)\mathbf{x}'_y \mathbf{b}_k \ . \tag{25}$$

Notice that we can also use this equation to predict ratings for items that have not yet been rated by any user as long as we have the characteristics of this item. Also, the computation of a rating is very fast, that is, of order $O(Kp)$, where K is the number of classes and p the number of characteristics.

As mentioned in the introduction, LCR-RS can also easily be used to explain the recommendations. This can be done by creating the personal regression vector $\mathbf{b}_u$ in the following way

$$\mathbf{b}_u = \sum_{k=1}^{K} P(z_k|u)\mathbf{b}_k \ . \tag{26}$$

Then, using the product characteristics vector $\mathbf{x}_y$ of a highly rated product, the coefficients of the regression vector can be multiplied by the values of the characteristics. The product is then an identifier of how important that characteristic was in the recommendation. Note that when all characteristics are binary, that is have a value of 0 or 1, this multiplication is not necessary and one only has to sort the coefficients of the characteristics with a value of 1. In the same way, we can also provide insights in the user communities.

Finally, note that the number of parameters in the LCR-RS model is $K \times (m + p + 1)$. (p denotes the number of characteristics in the $\mathbf{x}$ vector.) When there are many items, this is typically much smaller than the number of parameters in the

pLSA-CF model, since the regression weights and the variances within each latent class are shared among all items rather than being item-specific. Thus, the LCR-RS model is likely to be less prone to overfitting.

Algorithm 2 LCR-RS algorithm

procedure LCR_RS(entries $\langle u, y, v\rangle$,$\mathbf{X}$,K)

▷ Initialization

for all u **do**
Generate random $P(z_k|u)$ such that $\sum_k P(z_k|u) = 1$.
end for
for all y, z_k **do**
Generate random $\mathbf{b}_k$.
Generate random σ_k.
end for
Set $err = \infty$.

▷ EM-loop

repeat
$prevErr = err$.
for all $\langle u, y, v\rangle, k$ **do**
Compute $P(z_k|u, y, v)$ using (22).
end for
for all u, z_k **do**
Compute $P(z_k|u)$ using (10).
end for
for all y, z_k **do**
Compute $\mathbf{b}_k$ using (23).
Compute σ_k using (24).
end for
Compute $err = -E[L^c(\theta)]$ using (9).
until $prevErr - err < \epsilon$
end procedure

4. DATA

We evaluate our new algorithm on a data set concerning movie ratings. Since there is no publicly available data set containing both movie ratings and movie characteristics, we combine two publicly available data sets for this purpose.

The first data set we use is the Netflix data set. Netflix[3] is an online company that provides a DVD movie rental service. To provide their customers with personalized DVD recommendations, their web site makes use of a recommender system called Cinematch. Currently, Netflix has organized a contest, the Netflix Prize[4], to challenge the public to improve the prediction accuracy of Cinematch with ten percent. For this contest a data set was made available containing over 100 million movie ratings on a 5 point scale.

The Netflix data set does not contain characteristics about the movies except the title and the year of release. Therefore, we combined it with the IMDb data set. The Internet Movie Database (IMDb)[5] is a web site that contains information about

[3] `http://www.netflix.com`
[4] `http://www.netflixprize.com`
[5] `http://www.imdb.com`

approximately 900,000 titles (movies, TV series, etc.). It offers information such as plots, keywords that describe a title, genre classification, and participating actors and actresses. The IMDb database is available for download from the website.

When we combined both data sets and only include ratings corresponding to a movie also described in the IMDb data set, we were left with approximately 80 million ratings by about 480,000 users of 9,620 movies. Since it is practically impossible to work and experiment with such a large data set, we use a subset of these data in our evaluation.

To construct this reduced data set, first users with more than 50 ratings were selected. Next a random sample of .25% was taken from the selected users. The ratings of these sampled users form the foundation of the reduced data set. As characteristics of the movies we used dummy variables for each genre and for the 100 most often used keywords. This results in a data set of 191,059 ratings by 741 users of 6,305 movies. On average the users have rated 258 movies and a movie is on average rated by 30 users. The 6,302 movies have on average 2.50 genres and 28.2 keywords (of the 100) connected to them. We provide more statistics on the item characteristics in Table I. The table shows both the number and percentage of occurrence both per rating in the data set and per item in the data set. One should especially note that there are several genres, such as Adult, Reality-TV, and News, which have only a very small number of ratings and items connected to them. Therefore, results for these genres may be unreliable and regression coefficients may be insignificant.

5. EXPERIMENTS & RESULTS

In this section, we first describe the experiment setup used to evaluate the LCR-RS. Then, we discuss the findings of these experiments and, finally, we give an interpretation of the parameters of the LCR-RS model and discuss how these can be used to explain the recommendation to the user.

5.1 Experiment Setup

To evaluate the performance of the LCR-RS, we compared the prediction performance of LCR-RS to several other recommendation methods on the data described in the previous section.

The first method we compared with is a simple baseline method, which was also used in Hofmann [2004]. The prediction for an item $\langle u, y \rangle$ is simply the mean rating for item y in the training data.

We also used a simple memory-based collaborative and three simple content-based recommendation methods as comparison. The collaborative method is the Pearson correlation method. And as content-based methods we use two approaches based on linear regression. The first approach uses a single linear regression model on the complete data set and provides the same predictions for all users. The second approach fits a linear regression model for each user, only using the items rated by this user as input. For users that do not have rated enough items to be able to train a linear regression model, the global linear regression model is used for prediction. Finally, we also tested a Nearest Neighbor algorithm [Cover and Hart 1967] using the Hamming distance as distance measure. We report the results of using 10 neighbors, since this lead to the best results.

Table I. Item Characteristics of the data set.

Characteristic	per rating #	per rating %	per item #	per item %
Intercept	191.059	100,00%	6.302	100,00%
Genres				
Comedy	84.844	44,41%	2.310	36,66%
Animation	4.553	2,38%	209	3,32%
Family	14.888	7,79%	550	8,73%
Biography	8.386	4,39%	239	3,79%
Sci-Fi	19.238	10,07%	504	8,00%
Mystery	17.021	8,91%	449	7,12%
Musical	4.557	2,39%	256	4,06%
Thriller	61.383	32,13%	1.451	23,02%
Film-Noir	388	0,20%	50	0,79%
Adult	62	0,03%	13	0,21%
Romance	45.530	23,83%	1.200	19,04%
Sport	7.668	4,01%	179	2,84%
Drama	108.071	56,56%	3.497	55,49%
Adventure	33.434	17,50%	751	11,92%
Horror	13.631	7,13%	626	9,93%
Western	3.509	1,84%	165	2,62%
Action	49.939	26,14%	1.129	17,91%
Documentary	2.032	1,06%	190	3,01%
Fantasy	21.297	11,15%	474	7,52%
War	9.078	4,75%	271	4,30%
Short	175	0,09%	29	0,46%
Reality-TV	11	0,01%	2	0,03%
History	4.588	2,40%	136	2,16%
Music	5.246	2,75%	163	2,59%
Crime	36.925	19,33%	921	14,61%
News	154	0,08%	1	0,02%

Characteristic	per rating #	per rating %	per item #	per item %
Keywords				
blockbuster	53.957	28,24%	437	6,93%
murder	51.462	26,94%	1.206	19,14%
title-spoken-by-char.	41.820	21,89%	510	8,09%
based-on-novel	41.707	21,83%	1.399	22,20%
char.-name-in-title	36.899	19,31%	996	15,80%
twist-in-the-end	31.177	16,32%	463	7,35%
independent-film	32.998	17,27%	1.821	28,90%
police	30.310	15,86%	685	10,87%
helicopter	28.847	15,10%	452	7,17%
friendship	28.453	14,89%	468	7,43%
flashback-sequence	28.046	14,68%	476	7,55%
love	27.120	14,19%	508	8,06%
father-son-rel.	26.151	13,69%	433	6,87%
violence	25.354	13,27%	482	7,65%
gun	25.000	13,08%	393	6,24%
marriage	23.904	12,51%	489	7,76%
death	23.364	12,23%	462	7,33%
new-york-city	22.703	11,88%	404	6,41%
revenge	22.900	11,99%	441	7,00%
father-daughter-rel.	22.143	11,59%	337	5,35%
dog	22.165	11,60%	422	6,70%
blood	22.466	11,76%	409	6,49%
vulgarity	22.496	11,77%	314	4,98%
hospital	21.231	11,11%	397	6,30%
mother-son-rel.	20.176	10,56%	320	5,08%

The final benchmark model we use is the original pLSA-CF model of Hofmann [2004]. Since the LCR-RS model uses a lot less parameters (informally, the LCR-RS model estimates the $\mu_{y,z}$ for each latent class z by a linear function), we do not expect the LCR-RS model to improve the predictions of pLSA-CF. However, LCR-RS has the advantage that it can also predict ratings for new items.

We compare the models using the data described in the previous section. We divide the data in a training and test set by randomly selecting 10% of the rated items per user (with a minimum of 1) and adding these to the test set. All other data are used in the training set. Both pLSA-CF and LCR-RS also need a validation set to stop the algorithm. This validation set is extracted from the training set in the same way as the test set was extracted from the complete data set. We repeat this approach in total 20 times with different random seeds. As measure for prediction error, we use the mean absolute error

$$MAE = \frac{\sum_{i=1}^{N^*} |\tilde{v}_i - v_i|}{N^*} \ , \tag{27}$$

where N^* is the size of the test set and $\tilde{v}_i$ is the predicted rating and v_i the given rating. Besides as measure for the prediction error, the MAE is also used as measure on the validation set for stopping the pLSA-CF and LCR-RS algorithms.

Both in the pLSA-CF and LCR-RS algorithm the number of classes K is set to 40. Hofmann [2004] suggests to use this value in the pLSA-CF algorithm and because of the similarity between both algorithms we think this will also be a good value to use in the LCR-RS. Furthermore, also similar to Hofmann's approach [Hofmann 2004], we fit both models on ratings v' standardized by subtracting the mean user rating and dividing by the user specific standard deviation

$$v' = \frac{v - \mu_u}{\sigma_u} \ . \tag{28}$$

This standard deviation is smoothed in the following way suggested by Hofmann [2004]

$$\sigma_u = \sqrt{\frac{\sum_{\langle u=u',y,v\rangle}(v - \mu_u)^2 + 5\bar{\sigma}^2}{N_u + 5}} \ , \tag{29}$$

where μ_u is the user specific mean, $\bar{\sigma}^2$ the overall standard deviation, and N_u is the number of ratings done by the user.

Following Hofmann [2004], we also implemented tempered EM [Hofmann 2001] in pLSA-CF in which Equation (8) is replaced by

$$P(z_k|u,y,v,\theta) = \frac{(P(z_k|u)P(v|\mu_{k,y},\sigma_{k,y}))^\beta}{\sum_{k'}(P(z_{k'}|u)P(v|\mu_{k',y},\sigma_{k',y}))^\beta} \ , \tag{30}$$

where β is a tempering parameter used for regularization and, therefore, avoiding overfitting when set to a value < 1. In our experiments, setting β to .95 lead to the best results for pLSA-CF. Unfortunately, using tempered EM did not lead to improved results for LCR-RS.

Table II. Prediction accuracy of the different recommendation methods.

Method	MAE	Improvement
Baseline	0.8309	–
Collaborative Filtering		
Pearson	0.7083	14.8%
pLSA-CF	0.7035	15.3%
Content Based Filtering		
Linear Regression	0.8671	-4.4%
User Based Linear Regression	0.9731	-17.1%
10 Nearest Neighbor	0.7782	6.3%
Latent Class Regression Recommender System		
LCR-RS	0.7254	12.7%

5.2 Experiment Results

Table II shows the test MAE's (averaged over 20 runs) for the different recommendation methods. As can be seen 4 out of 6 methods have a higher accuracy than the baseline method. Only the two content based methods based on linear regression perform worse. Also, the third pure content based method, the nearest neighbor method performs not that good, only improving the baseline by six percent. On the other hand, the collaborative methods were able to improve the accuracy compared to the baseline by 15%. Although pLSA-CF performed somewhat better, the difference with the Pearson correlation is very small. The error made by LCR-RS is much smaller than the error made by the content-based methods, although it is somewhat higher than the error of the collaborative methods as we expected. It is remarkable that, although linear regression itself seems to be a poor performing content-based model, it works well in our framework. When comparing LCR-RS to the collaborative methods, LCR-RS only performs 3% worse than these methods. However, only LCR-RS has the ability to provide recommendations of new items.

5.3 Model Interpretation

An extra advantage of the LCR-RS is that it provides an explanation of the recommendations given. Furthermore, the model can be used to get a better insight in the user communities.

For each latent class, which can be seen as some user community, the LCR-RS model contains a class specific regression vector, indicating which characteristics have a high positive or negative effect on the predicted rating. Table III shows the regression coefficients for two classes of our model for the 10 most occurring genres and keywords. As can be seen in this example, movie preferences can differ a lot among user communities. For example, for these two classes one can see that users belonging to class 15 consider it to be quite positive that a movie belongs to the science fiction or fantasy genre, while people in class 14 do not seem to prefer these movies. However, sometimes interpretation can still be quite complicated, since some characteristics are correlated which each other. For instance, the genre romance and the keyword love, will occur quite a lot together. When looking at the regression coefficients for both classes for these characteristics, we see that the one has a negative coefficient for love and a positive one for romance, where for the

Table III. Regression coefficients for two user communities.

Characteristic	Class 14	Class 15
Intercept	0.9545	0.5255
Drama	0.0889	0.0929
Comedy	-0.0774	-0.0237
Thriller	-0.0919	0.0248
Romance	-0.0403	0.0095
Action	-0.0981	-0.0403
Crime	-0.0354	-0.1090
Horror	-0.2306	-0.4084
Family	-0.0494	-0.0200
Sci-Fi	**-0.1251**	**0.0635**
Fantasy	**-0.0001**	**0.1348**
independent-film	0.0232	-0.0342
based-on-novel	0.0257	0.1692
murder	0.0116	0.0429
char.-name-in-title	0.0145	0.0390
police	-0.0239	-0.0057
title-spoken-by-char.	0.0438	-0.0215
love	0.0756	-0.0367
marriage	0.0124	0.0905
violence	-0.0449	0.0991
flashback-sequence	0.0172	-0.0181

Table IV. Recommendation order per user community for ten selected movies.

Movie Title	Class 14		Class 15	
	Abs	Rel	Abs	Rel
Bridget Jones' Diary	572	7	131	1
Die Hard	1568	9	190	4
Harry Potter and the Sorcerer's Stone	55	3	220	5
I Know What You Did Last Summer	5745	10	2167	8
Reservoir Dogs	16	1	1364	7
Saving Private Ryan	38	2	352	6
Seven	125	4	3051	9
The Matrix	1459	8	147	2
Titanic	290	5	6265	10
Unforgiven	474	6	188	3

other class it is opposite. Therefore, it is quite difficult to see which community would like these kind of movies more. When one wants to overcome this problem of correlated characteristics one could use feature selection or extraction methods. However, although this may increase the usefulness of the explanations, it might also reduce prediction performance.

To gain somewhat more insight in the user communities in class 14 and 15 of the model, we have estimated ratings for all movies in the data based on the regression coefficients of these classes and ordered the recommendations based on these predicted ratings. In Table IV, we show the recommendation rank for ten well-known movies for both classes. We see that the order is often in line with the results shown

Table V. Regression coefficients of most important positive and negative characteristics for User 369.

Positive Characteristics	Coefficient	Negative Characteristics	Coefficient
News	0.4785	Adult	-0.2894
Short	0.3920	Horror	-0.2282
Documentary	0.2425	Sci-Fi	-0.0907
sequel	0.2176	Film-Noir	-0.0820
famous-score	0.1868	male-nudity	-0.0809

Table VI. Regression coefficients of most important positive and negative characteristics for User 369 that determine the rating for the movie 'Schindler's List'.

Positive Characteristics	Coefficient	Negative Characteristics	Coefficient
famous-score	0.1868	sex	-0.0584
blockbuster	0.1801	female-nudity	-0.0438
History	0.1042	premarital-sex	-0.0409
redemption	0.1005	rescue	-0.0275
Drama	0.0802	blood	-0.0202

in Table III, such as the higher ranks for the Sci-Fi movie 'The Matrix' and action movie 'Die Hard' in class 15. However, sometimes movies get a very high or low rating due to a not that common characteristic not shown in Table III. This might for instance be the case for the movie 'Reservoir Dogs' in class 14.

The same regression coefficients that were shown for the user communities can be computed for individual users using Equation (26). We computed these individual regression coefficients for a random user, namely user 369. For this user, the most important positive and negative characteristics are listed in Table V. As can be seen, the model expects that the user would like to see news, short movies, and documentaries, while he would not like adult, horror, and science-fiction movies.

These kind of regression coefficients can be used for explanation in, for example, the following way. Top recommendation for this user is the movie 'Schindler's List'. With respect to this movie, we can determine which characteristics have the most influence on this rating, that is, we look for characteristics that have a value of 1 for this movie and then look at the highest coefficients for these characteristics. These coefficients for the five most important positive and negative characteristics that determine the rating for user 369 for the movie 'Schindler's List' are shown in Table VI. Note that the most important positive characteristics have a far more higher coefficient than the negative ones, since this is a movie that the user is expected to like. Based on these coefficients a recommender system should be able to provide an explanation of the recommendation made, that should look something like

> We think you would like the movie **Schindler's List**, because it has a **famous score** and is considered to be a **blockbuster**. Also, the movie is about **History**, **redemption**, and it belongs to the **Drama** genre. However, there are also some things that you may not like about the movie. The movie contains **sex** and **female nudity**. Also, the movie contains **premarital sex**, a **rescue**, and **bloody** scenes.

As we said earlier, we think that the explanations of the system might be improved by using some feature selection or extraction method. Not only will such a

method eliminate correlating characteristics, also characteristics that do not occur a lot and are therefore insignificant might be excluded. As can be seen in Table V, some of these characteristics now have high coefficients in the model, while these are probably highly unreliable due to the limited number of positive examples they are based on.

6. CONCLUSIONS

In this paper, we presented a new hybrid recommendation approach. This approach is based on the idea that there are communities of users that find the same characteristics important to like or dislike a product. To construct this model, which we call the latent-class regression recommender system (LCR-RS), we extended Hofmann's probabilistic latent semantic analysis model for collaborative filtering (pLSA-CF) [Hofmann 2001]. This extension is based on clusterwise regression [DeSarbo and Cron 1988] and leads to a model containing community specific regression vectors. Advantages of this model are that it is able to recommend new items and provides a framework to explains its recommendations, while it is also able to use the ratings of all users.

We applied this approach on a data set containing movie ratings and movie characteristics (genres and keywords). On these data, prediction performance of LCR-RS was slightly worse than collaborative methods, such as Pearson correlation and pLSA-CF, but better than content-based methods. Although performance of LCR-RS is somewhat worse than the collaborative methods, this method has in favour that it can recommend new items and provides a way to explain its recommendations, which was also illustrated for the movie data.

We see some directions to future research based on the experiments with the pLSA-CF algorithm. For example, we expect that explanations of the LCR-RS might be improved using feature extraction or selection methods excluding correlated and insignificant characteristics. This can maybe done by integrating stepwise regression methods or lasso regression [Tibshirani 1996].

Also, the LCR-RS model might be adapted to model other type of user evaluations such as a binary 'like - don't like' evaluation. This can be done using generalized linear models [McCullagh and Nelder 1989], such as logistic or probit regression, in combination with latent classes [Wedel and DeSarbo 1995].

REFERENCES

Adomavicius, G. and Tuzhilin, A. 2005. Towards 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.

Ansari, A., Essegaier, S., and Kohli, R. 2000. Internet recommendation systems. *Journal of Marketing Research 37,* 3, 363–375.

Balabanović, M. and Shoham, Y. 1997. Fab: Content-based, collaborative recommendation. *Communications of the ACM 40,* 3, 66–72.

Basu, C., Hirsh, H., and Cohen, W. 1998. Recommendation as classification: Using social and content-based information in recommendation. In *Proceedings of the Fifteenth National/Tenth Conference on Artificial Intelligence/Innovative Applications of Artificial Intelligence*. AAAI Press, Menlo Park, CA, USA, 714–720.

Billsus, D. and Pazzani, M. J. 2000. User modeling for adaptive news access. *User Modeling and User-Adapted Interaction 10*, 147–180.

Bishop, C. M. 2006. *Pattern Recognition and Machine Learning.* Springer.

Blei, D. M., Ng, A. Y., and Jordan, M. I. 2003. Latent Dirichlet allocation. *Journal of Machine Learning Research 3*, 993–1022.

Breese, J. S., Heckerman, D., and Kadie, C. 1998. Empirical analysis of predictive algorithms for collaborative filtering. In *Proceedings of the Fourteenth Annual Conference on Uncertainty in Artificial Intelligence*. Morgan Kaufmann, San Francisco, CA, USA, 43–52.

Burke, R. 2002. Hybrid recommender systems: Survey and experiments. *User Modeling and User-Adapted Interaction 12*, 331–370.

Camargo, S. J., Robertson, A. W., Gaffney, S. J., Smyth, P., and Ghil, M. 2007. Cluster analysis of typhoon tracks. part I. general properties. *Journal of Climate 20,* 14, 3635–3653.

Claypool, M., Gokhale, A., Miranda, T., Murnikov, P., Netes, D., and Sartin, M. 1999. Combining content-based and collaborative filtering in an online newspaper. In *Proceedings of ACM SIGIR Workshop on Recommender Systems: Algorithms and Evaluation*. ACM Press.

Condliff, M., Lewis, D., Madigan, D., and Posse, C. 1999. Bayesian mixed-effects models for recommender systems. In *Proceedings of ACM SIGIR Workshop on Recommender Systems: Algorithms and Evaluation*. ACM Press.

Cover, T. and Hart, P. 1967. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory 13,* 1, 21–27.

Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., and Harshman, R. 1990. Indexing by latent semantic analysis. *Journal of the American Society for Information Science 41,* 6, 391–407.

Dempster, A. P., Laird, N. M., and Rubin, D. B. 1977. Maximum likelihood from incomplete data via the E-M-algorithm. *Journal of the Royal Statistical Society Series B 39*, 1–38.

DeSarbo, W. S. and Cron, W. L. 1988. A maximum likelihood methodology for clusterwise linear regression. *Journal of Classification 5,* 2, 249–282.

DeSarbo, W. S., Di Benedetto, C. A., Song, M., and Sinha, I. 2005. Revisiting the miles and snow strategic framework: uncovering interrelationships between strategic types, capabilities, environmental uncertainty, and firm performance. *Strategic Management Journal 26,* 1, 47–74.

DeSarbo, W. S., Wedel, M., Vriens, M., and Ramaswamy, V. 1992. Latent metric conjoint analysis. *Marketing Letters 3,* 3, 273–288.

Goldberg, D., Nichols, D., Oki, B. M., and Terry, D. 1992. Using collaborative filtering to weave an information tapestry. *Communications of the ACM 35,* 12, 61–70.

Goldberg, K., Roeder, T., Gupta, D., and Perkins, C. 2001. Eigentaste: A constant time collaborative filtering algorithm. *Information Retrieval 4*, 133–151.

Good, N., Schafer, J. B., Konstan, J. A., Borchers, A., Sarwar, B. M., Herlocker, J. L., and Riedl, J. T. 1999. Combining collaborative filtering with personal agents for better recommendations. In *Proceedings of the Sixteenth National/Eleventh Conference on Artificial Intelligence/Innovative Applications of Artificial Intelligence*. AAAI Press, Menlo Park, CA, USA, 439–446.

Herlocker, J. L., Konstan, J. A., and Riedl, J. 2000. Explaining collaborative filtering recommendations. In *Proceedings of the 2000 ACM Conference on Computer Supported Cooperative Work*. ACM Press, New York, 241–250.

Hofmann, T. 1999. Probabilistic latent semantic analysis. In *Proceedings of the Fifteenth Annual Conference on Uncertainty in Artificial Intelligence*. Morgan Kaufmann, San Francisco, CA, USA, 289–296.

Hofmann, T. 2001. Unsupervised learning by probabilistic latent semantic analysis. *Machine Learning 42*, 177–196.

Hofmann, T. 2004. Latent semantic models for collaborative filtering. *ACM Transactions on Information Systems 22*, 89–115.

Kriegler, M. and Zimprich, D. 2006. Predictors of cognitive complaints in older adults: A mixture regression approach. *European Journal of Ageing 2,* 1, 13–23.

Larcker, D. F. and Richardson, S. A. 2004. Fees paid to audit firms, accrual choices, and corporate governance. *Journal of Accounting Research 42,* 3, 625–658.

Linden, G., Smith, B., and York, J. 2003. Amazon.com recommendations: Item-to-item collaborative filtering. *IEEE Internet Computing 7,* 1, 76–80.

McCullagh, P. and Nelder, J. A. 1989. *Generalized Linear Models*, 2nd ed. Monographs on Statistics and Applied Probability, vol. 37. Chapman & Hall, Boca Raton.

McSherry, D. 2005. Explanation in recommender systems. *Artificial Intelligence Review 24*, 179–197.

Melville, P., Mooney, R. J., and Nagarajan, R. 2002. Content-boosted collaborative filtering for improved recommendations. In *Proceedings of the Eighteenth National Conference on Artificial Intelligence*. AAAI Press, Menlo Park, CA, USA, 187–192.

Pazzani, M. J. 1999. A framework for collaborative, content-based and demographic filtering. *Artificial Intelligence Review 13*, 393–408.

Pazzani, M. J. and Billsus, D. 1997. Learning and revising user profiles: The identification of interesting web sites. *Machine Learning 27*, 313–331.

Pazzani, M. J. and Billsus, D. 2007. Content-based recommendation system. In *The Adaptive Web*. Lecture Notes in Computer Science, vol. 4321. Springer, Heidelberg, 325–341.

Pennings, J. M. and Garcia, P. 2004. Hedging behavior in small and medium-sized enterprises: The role of unobserved heterogeneity. *Journal of Banking and Finance 28,* 5, 951–978.

Popescul, A., Ungar, L. H., Pennock, D. M., and Lawrence, S. 2001. Probabilistic models for unified collaborative and content-based recommendation in sparse-data environments. In *Proceedings of the 17th Conference on Uncertainty in Artificial Intelligence*. 437–444.

Prasad, B. 2003. Intelligent techniques for e-commerce. *Journal of Electronic Commerce Research 4,* 2, 65–71.

Ramaswamy, V., DeSarbo, W. S., Reibstein, D. J., and Robinson, W. T. 1993. An emperical pooling approach for estimating marketing mix elasticities with PIMS data. *Marketing Science 12,* 1, 103–124.

Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., and Riedl, J. T. 1994. GroupLens: An open architecture for collaborative filtering of netnews. In *Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work*. ACM Press, New York, NY, USA, 175–186.

Resnick, P. and Varian, H. R. 1997. Recommender systems. *Communications of the ACM 40,* 3, 56–58.

Sarwar, B. M., Karypis, G., Konstan, J. A., and Riedl, J. T. 2001. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the Tenth International Conference on World Wide Web*. ACM Press, New York, NY, USA, 285–295.

Schein, A. I., Popescul, A., Ungar, L. H., and Pennock, D. M. 2002. Methods and metrics for cold-start recommendations. In *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM Press, New York, 253–260.

Sinha, R. and Swearingen, K. 2002. The role of transparency in recommender systems. In *CHI '02 extended abstracts on Human factors in computing systems*. ACM Press, New York, 830–831.

Soboroff, I. and Nicholas, C. 1999. Combining content and collaboration in text filtering. In *Proceedings of IJCAI Workshop Machine Learning for Information Filtering*.

Tibshirani, R. 1996. Regression shrinkage and selection via the Lasso. *Journal of the Royal Statistical Society. Series B (Methodological) 58,* 1, 267–288.

Tintarev, N. and Masthoff, J. 2007. A survey of explanations in recommender systems. In *Proceedings of the 23rd International Conference on Data Engineering Workshops*, V. Oria, A. Elmagarmid, F. Lochovsky, and Y. Saygin, Eds. IEEE Computer Society, Los Alamitos, 801–810.

Tran, T. and Cohen, R. 2000. Hybrid recommender systems for electronic commerce. In *Proceedings of Knowledge-Based Electronic Markets, Papers from the AAAI Workshop*. AAAI Press.

Vriens, M., Wedel, M., and Wilms, T. 1996. Metric conjoint segmentation methods: A monte carlo comparison. *Journal of Marketing Research 33,* 1, 73–85.

Wedel, M. and DeSarbo, W. S. 1995. A mixture likelihood approach for generalized linear models. *Journal of Classification 12,* 1, 21–55.

Publications in the Report Series Research* in Management

ERIM Research Program: "Marketing"

2008

Experts' Stated Behavior
Youssef Boulaksil and Philip Hans Franses
ERS-2008-001-MKT
http://hdl.handle.net/1765/10900

The Value of Analogical Reasoning for the Design of Creative Sales Promotion Campaigns: A Case-Based Reasoning Approach
Niek A.P. Althuizen and Berend Wierenga
ERS-2008-006-MKT
http://hdl.handle.net/1765/11289

Shopping Context and Consumers' Mental Representation of Complex Shopping Trip Decision Problems
Benedict G.C. Dellaert, Theo A. Arentze and Harry J.P. Timmermans
ERS-2008-016-MKT
http://hdl.handle.net/1765/11812

Modeling the Effectiveness of Hourly Direct-Response Radio Commercials
Meltem Kiygi Calli, Marcel Weverbergh and Philip Hans Franses
ERS-2008-019-MKT
http://hdl.handle.net/1765/12242

Choosing Attribute Weights for Item Dissimilarity using Clikstream Data with an Application to a Product Catalog Map
Martijn Kagie, Michiel van Wezel and Patrick J.F. Groenen
ERS-2008-024-MKT
http://hdl.handle.net/1765/12243

The Effect of Superstar Software on Hardware Sales in System Markets
Jeroen L.G. Binken and Stefan Stremersch
ERS-2008-025-MKT
http://hdl.handle.net/1765/12339

Sales Growth of New Pharmaceuticals Across the Globe: The Role of Regulatory Regimes
Stefan Stremersch and Aurélie Lemmens
ERS-2008-026-MKT
http://hdl.handle.net/1765/12340

When Intelligence is (Dys)Functional for Achieving Sales Performance
Willem J. Verbeke, Frank D. Belschak, Arnold B. Bakker, and Bart Dietz
ERS-2008-034-MKT
http://hdl.handle.net/1765/12633

Path Dependencies and the Long-term Effects of Routinized Marketing Decisions
Paul Farris, Willem J. Verbeke, Peter Dickson and Erjen van Nierop
ERS-2008-035-MKT
http://hdl.handle.net/1765/12634

Does Irritation Induced by Charitable Direct Mailings Reduce Donations?
Merel van Diepen, Bas Donkers and Philip Hans Franses
ERS-2008-036-MKT
http://hdl.handle.net/1765/12704

Brain Mechanisms of Persuasion: How "Expert Power" Modulates Memory and Attitudes
Vasily Klucharev, Ale Smidts and Guillén Fernández
ERS-2008-038-MKT
http://hdl.handle.net/1765/12784

Moderating Factors of Immediate, Dynamic, and Long-run Cross-Price Effects
Csilla Horváth and Dennis Fok
ERS-2008-042-MKT
http://hdl.handle.net/1765/12901

Why, How and When Do Prices Land? Evidence from the Videogame Industry
Carlos Hernandez-Mireles, Dennis Fok and Philip Hans Franses
ERS-2008-044-MKT
http://hdl.handle.net/1765/12900

Situation-Based Shifts in Consumer Web Site Benefit Salience: The Joint Role of Affect and Cognition
Sonja Wendel and Benedict G.C. Dellaert
ERS-2008-050-MKT
http://hdl.handle.net/1765/13179

Including Item Characteristics in the Probabilistic Latent Semantic Analysis Model for Collaborative Filtering
Martijn Kagie, Matthijs van der Loos and Michiel van Wezel
ERS-2008-053-MKT
http://hdl.handle.net/1765/13180

Cross-National Logo Evaluation Analysis: An Individual Level Approach
Ralf van der Lans, Joseph A. Cote, Catherine A. Cole, Siew Meng Leong, Ale Smidts, Pamela W. Henderson, Christian Bluemelhuber, Paul A. Bottomley, John R. Doyle, Alexander Fedorikhin, M. Janakiraman, B. Ramaseshan, and Bernd H. Schmitt
ERS-2008-055-MKT
http://hdl.handle.net/1765/13181

Sales and Sincerity: The Role of Relational Framing in Word-of-Mouth Marketing
Mirjam A. Tuk, Peeter W.J. Verlegh, Ale Smidts and Daniel H.J. Wigboldus
ERS-2008-056-MKT
http://hdl.handle.net/1765/13183

Interpersonal Relationships Moderate the Effect of Faces on Person Judgments
Mirjam A. Tuk, Peeter W.J. Verlegh, Ale Smidts and Daniel H.J. Wigboldus
ERS-2008-057-MKT
http://hdl.handle.net/1765/13185

---

* A complete overview of the ERIM Report Series Research in Management:
https://ep.eur.nl/handle/1765/1

ERIM Research Programs:
LIS Business Processes, Logistics and Information Systems
ORG Organizing for Performance
MKT Marketing
F&A Finance and Accounting
STR Strategy and Entrepreneurship