Chapter 2
Recommender Systems

In the following we will describe systematically and formally the most important problems related to recommender systems and give some references to actual solutions. Our focus here is to describe the general recommender systems setting as a base for social recommender systems. See [11, 3] for a more general introduction to recommender systems and a more thorough overview of the state-of-the-art, respectively.

2.1 Rating and Item Prediction

The two most basic recommendation problems are rating prediction and item prediction.

In rating prediction, there are users that rate items (e. g., movies, books, electronic devices, articles, resources in the terminology of social systems etc.) explicitely on some scale, say with the numbers 1 to 5, where 1 denotes the least preferred item and 5 the most preferred one. Given such ratings we would like to predict ratings of users for items they did not rate yet. In the most basic scenario, users and items are treated as entities about which nothing else is known, i. e., as IDs or nominal levels. Formally, there are given

- a set U of users,
- a set I of items,
- a set $\mathcal{R} \subseteq \mathbb{R}$ of ratings, e. g., $\mathcal{R} := \{1, 2, 3, 4, 5\}$,
- a set $\mathcal{D}^{\text{train}} \subseteq U \times I \times \mathcal{R}$ of (user, item, rating) triples,
- a (rating) loss function $\ell : \mathcal{R} \times \mathbb{R} \to \mathbb{R}$ where $\ell(r, \hat{r})$ quantifies how bad it is to predict rating $\hat{r}$ if the actual rating is r. A typical choice for the loss is absolute error or squared error:

$$\ell_{AE}(r, \hat{r}) := |r - \hat{r}|, \quad \ell_{SE}(r, \hat{r}) := (r - \hat{r})^2$$

Sought is the prediction of the rating for a user and item, i. e.,

L.B. Marinho et al., *Recommender Systems for Social Tagging Systems*, SpringerBriefs in Electrical and Computer Engineering, DOI 10.1007/978-1-4614-1894-8_2, © The Author(s) 2012

$$\hat{r} : U \times I \to \mathbb{R}$$

s. t. for some test set $\mathcal{D}^{\text{test}} \subseteq U \times I \times \mathcal{R}$ of (user, item, rating) triples (from the same unknown distribution as the train set and not available for the construction of $\hat{r}$) the test risk

$$\text{risk}(\hat{r}; \mathcal{D}^{\text{test}}) := \frac{1}{|\mathcal{D}^{\text{test}}|} \sum_{(u,i,r) \in \mathcal{D}^{\text{test}}} \ell(r, \hat{r}(u,i))$$

is minimal.

In the second problem scenario, item prediction, there are no ratings, but just co-occurrences of users and items, e. g., users may view or buy some of the items. Formally, there are given

- a set U of users,
- a set I of items,
- a set $\mathcal{D}^{\text{train}} \subseteq U \times I$ of (user, item) co-occurrences,
- a (ranking) loss function $\ell : \mathcal{P}(I) \times \mathbb{R}^I \to \mathbb{R}$, e. g., recall at k

$$\text{recall}_k(J, \hat{r}) := \frac{1}{|J|} |J \cap \operatorname*{argmax}^{k}_{i' \in I} \hat{r}(i')|, \quad J \subseteq I$$

Sought is for every user a ranking of the items, i. e., a score function

$$\hat{r} : U \to \mathbb{R}^I \quad \text{or equivalently } \hat{r} : U \times I \to \mathbb{R}$$

s. t. for some test set $\mathcal{D}^{\text{test}} \subseteq U \times I$ of (user, item) co-occurrences (from the same unknown distribution as the train set and not available for the construction of $\hat{r}$) the test risk

$$\text{risk}(\hat{r}; \mathcal{D}^{\text{test}}) := \frac{1}{|U(\mathcal{D}^{\text{test}})|} \sum_{u \in U(\mathcal{D}^{\text{test}})} \ell(I_u(\mathcal{D}^{\text{test}}), \hat{r}(u)),$$

$$\text{with } U(\mathcal{D}) := \{u \in U \mid \exists i \in I : (u,i) \in \mathcal{D}\}, \quad I_u(\mathcal{D}) := \{i \in I \mid (u,i) \in \mathcal{D}\}$$

is minimal.

If the score function is injective (or made injective by breaking ties at random), it defines for each user u a linear order over the items by

$$i \prec_u j \quad :\Leftrightarrow \quad \hat{r}(u,i) > \hat{r}(u,j), \quad i,j \in I$$

2.2 Rating Prediction as Regression Problem

Recommendation problems such as rating and item prediction can be viewed as instances of broader problem classes. The rating prediction problem basically is an (ordinal) regression problem where an (ordinal/numeric) target variable (rating) should be predicted based on two nominal variables (user, item). Among the specific characteristics of the rating prediction is (i) the high number of levels of each of the two nominal variables (many users, many items) and consequently (ii) the extreme sparsity, i. e., that ratings are observed for only very few user/item pairs. Regression problems of this type are also described lucidly as matrix completion problems where rows and columns of the matrix are indexed by the nominal levels of the two variables and cell values are the ratings, most cells being not observed (see e. g., [6]).

Treating the rating prediction problem in a naive way, say, with binary indicator variables for the nominal levels of users and items and a linear model on these variables, leads to a very simple model

$$\hat{r}(u,i) := \mu + \mu^U(u) + \mu^I(i), \quad \mu \in \mathbb{R}, \mu^U : U \to \mathbb{R}, \mu^I : I \to \mathbb{R}$$

where μ denotes a global average rating and μ^U and μ^I model independent user and item effects (often called user and item bias). This model is unsuited for personalized predictions as it does not catch any user/item interactions. If one would use it, again in a naive way, for ranking items for a given target user, then this ranking would be the same for all users. On the other hand, adding an explicit interaction effect between user and item indicator to the model would lead to $|U| \times |I|$ parameters, as many as there are observations in the completed rating matrix.

Therefore, historically, researchers were looking for other methods to model the rating prediction regression problem. For its simplicity, especially the nearest neighbor model got a lot of attention [7, 20, 23]. Here, rating prediction is viewed as a separate problem for each item. Then the user indicator variable is the only variable remaining. Between users a similarity measure is defined based on their rating vectors, e. g., the Pearson correlation of their jointly rated items

$$\text{sim}(u,v) := \text{corr}_{\text{pearson}}(r|_{u,I_u \cap I_v}, r|_{v,I_u \cap I_v})$$

with

$$\begin{aligned} I_u &:= \{i \in I \mid \exists r : (u,i,r) \in \mathcal{D}^{\text{train}}\}, \quad u \in U \\ r|_{u,J} &:= (r(u,j))_{j \in J} \in \mathbb{R}^{|J|}, \quad J \subseteq I \\ \text{corr}_{\text{pearson}}(x,y) &:= \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2}\sqrt{\sum_{i=1}^n (y_i - \bar{y})^2}}, \quad x, y \in \mathbb{R}^n \end{aligned}$$

For each target item, then the k nearest neighbors having the target item rated are determined and the rating of a given target user for a given target item is predicted by a k-nearest-neighbor rule, e. g.,

$$\hat{r}(u,i) := \mu^U(u) + \sum_{v \in N_{u,i}} \frac{\text{sim}(u,v)}{\sum_{v' \in N_{u,i}} \text{sim}(u,v')} (r(v,i) - \mu^U(v))$$

with

$$N_{u,i} := \underset{v \in U:\ i \in I_v}{\overset{k}{\operatorname{argmax}}}\ \operatorname{sim}(u,v)$$

Alternatively, one could swap the roles of users and items, i. e., decompose the problem into separate problems for each item, define a similarity measure between items based on their rating vector by users and predict ratings by a k nearest neighbor rule on items. Nearest neighbor models often are called collaborative filtering or memory-based models in the context of recommender system problems. If users are the instances, they are called user-based, otherwise item-based.

At the end of the 90s, probabilistic latent class models, especially the aspect model [9, 10] have been developed that allowed a richer modelling of the user/item effects through a set of non-observed classes. These models nowadays can be understood as regularized sparse low-rank matrix factorization models [25]. In sparse low-rank matrix factorization models one associates a latent feature vector ϕ with every level of each nominal variable and models the interaction between two such variables by a function of their latent feature vectors, e. g., by their scalar product

$$\hat{r}(u,i) := \mu + \mu^U(u) + \mu^I(i) + \langle \phi^U(u), \phi^I(i) \rangle, \quad \phi^U : U \to \mathbb{R}^k, \phi^I : I \to \mathbb{R}^k$$

These models are called sparse matrix factorization models, because when identifying the rating and latent feature functions with the matrices of their values, the rating matrix can be reconstructed by the product of the feature matrices:

$$\hat{r} := \mu \mathbb{I} + \mu^U \mathbb{1}^T + \mathbb{1}(\mu^I)^T + \phi^U(\phi^I)^T,$$

where $\mathbb{I}$ denotes the $|U| \times |I|$ matrix containing only 1's and $\mathbb{1}$ the vector containing $|U|$ many 1's (or I many 1's, respectively). As most entries of this matrix are not observed, one measures the reconstruction error only on the sparse submatrix of observed entries, i. e.,

$$\ell(r, \hat{r}) := ||W^{\text{train}} \odot (r - \hat{r})||$$

where $W^{\text{train}} \in \mathbb{R}^{|U| \times |I|}$ is a weight matrix, usually

$$W^{\text{train}}_{u,i} := \delta(\exists r :\ (u,i,r) \in \mathcal{D}^{\text{train}}), \quad u \in U, i \in I$$

and $|| \cdot ||$ a matrix norm, e.g.,

$$||A|| := \sum_{i=1}^{n} \sum_{j=1}^{m} A_{i,j}^2, \quad A \in \mathbb{R}^{n \times m}$$

and $\odot$ denotes element-wise matrix multiplication. They are called low-rank because the dimension k of the feature vectors (and thus the rank of the resulting reconstruction $\hat{r}$) is small compared to the dimensions $|U|, |I|$ of the original matrix. The models are called regularized, as not just the training loss is minimized, but a combination of training risk and a regularization term, e. g., Tikhonov regularization

$$\text{min. } f(\phi^U, \phi^I) := \text{risk}(\hat{r}; \mathcal{D}^{\text{train}}) + \lambda(||\phi^U||^2 + ||\phi^I||^2), \quad \lambda \in \mathbb{R}_0^+$$

Sparse low-rank matrix factorization models are the state-of-the-art models at the time of writing [13]. They usually provide better performance than other models, do not require to have all training data available at prediction time and are easy to train. For training, different learning methods have been researched. Extremely simple and among the fastest training methods is stochastic gradient descent [13]. Here, one triple (u, i, r) at a time is sampled from the training data and the features are updated along the negative gradient with some learning rate $\eta \in \mathbb{R}^+$ until convergence:

$$\begin{aligned}
\phi^U(u) :=& \phi^U(u) - \eta \frac{\partial \ell}{\partial \hat{r}}(r, \hat{r}(u,i)) \frac{\partial \hat{r}}{\partial \phi^U(u)}(u,i) - 2\eta\lambda\phi^U(u) \\
\phi^I(i) :=& \phi^I(i) - \eta \frac{\partial \ell}{\partial \hat{r}}(r, \hat{r}(u,i)) \frac{\partial \hat{r}}{\partial \phi^I(i)}(u,i) - 2\eta\lambda\phi^I(i)
\end{aligned}$$

So for example, for the squared error loss this simply yields

$$\begin{aligned}
\phi^U(u) :=& \phi^U(u) - 2\eta(r - \hat{r}(u,i))\phi^I(i) - 2\eta\lambda\phi^U(u) \\
\phi^I(i) :=& \phi^I(i) - 2\eta(r - \hat{r}(u,i))\phi^U(u) - 2\eta\lambda\phi^I(i)
\end{aligned}$$

Extensions of the simple matrix factorization model also can cope with the ordinal level of the rating variable [15].

To demonstrate the usefulness of personalized models such as nearest neighbor models or matrix factorization models in some specific domain, one usually compares them with non-personalized models. More exactly, non-personalized models are models that are constant, either the globally constant model

$$\hat{r}(u,i) := \mu, \quad \mu \in \mathbb{R}$$

or constant w. r .t. one of the user or item variable, i. e., user or item averages:

$$\begin{aligned}
\hat{r}(u,i) &:= \mu + \mu^U(u), \quad \mu \in \mathbb{R}, \mu^U : U \to \mathbb{R} \\
\hat{r}(u,i) &:= \mu + \mu^I(i), \quad \mu \in \mathbb{R}, \mu^I : I \to \mathbb{R}
\end{aligned}$$

2.3 Item Prediction as Ranking Problem

The item prediction problem often is viewed as a set-valued classification problem (usually called a multi-label classification problem). As such it could be described as a set of dependant binary classification problems, one for each item. A naive Perceptron model again using binary indicator variables for each level of the user and item variable looks like this:

$$\hat{r}(u,i) := \mu + \mu^U(u) + \mu^I(i), \quad \mu \in \mathbb{R}, \mu^U : U \to \mathbb{R}, \mu^I : I \to \mathbb{R}$$

It suffers from the very same defect as the naive rating prediction model: it is not personalized, i. e., there is no interaction term for users and items and such a term cannot be inserted into the model for all interactions as these parameters are exactly the output one is trying to learn.

Also for item prediction, nearest neighbor models have been used very successfully very early on. Similarities no longer can be described by the correlation of the joint rating vectors, but, e. g., by measures of the overlap of the item sets for two users, e. g., the Jaccard coefficient

$$\mathrm{sim}(u,v) := \frac{|I_u \cap I_v|}{|I_u \cup I_v|}$$

The neigborhood of a user now does not depend on him having rated the target item, but just on similarity, and the nearest-neighbor rule counts the fraction of neighbors with the target item:

$$\hat{r}(u,i) := \sum_{v \in N_u} \frac{\mathrm{sim}(u,v)}{\sum_{v' \in N_u} \mathrm{sim}(u,v')} \delta((v,i) \in \mathcal{D}^{\mathrm{train}})$$

with

$$N_u := \underset{v \in U}{\overset{k}{\mathrm{argmax}}}\, \mathrm{sim}(u,v)$$

To better understand the item prediction problem, in our opinion three ideas have been crucial: (i) the problem has been tackled by a probabilistic latent class model, the aspect model, closely related to the one used for rating prediction [9, 10]. (ii) the decomposition by a binary classification model per item has been found not to work, as typical recommender data sets have disjoint train and test item sets for the same user, i. e., no repeating items, while pairwise decompositions have been shown to work well [22]. (iii) the matrix factorization approach and the direct optimization of a ranking loss have been applied to the item prediction problem [26]. Nowadays the simplest and most elegant formulation of the item prediction problem is not as a classification problem, but as a ranking problem using pairs of positive items (in the train set) and negative items (not in the train set) as pairwise input, optimizing a simple ranking loss such as AUC

$$\ell_{\mathrm{AUC}}(J,\hat{r}) := \frac{1}{|J||I \setminus J|} \sum_{j \in J, i \in I \setminus J} \delta(\hat{r}(j) > \hat{r}(i)), \quad J \subseteq I$$

and using matrix factorization as the ranking function [19].

By approximating the discontinuous step function δ, e. g., by the logistic function

$$\sigma(x) := \frac{1}{1+e^{-x}}$$

one gets a differentiable (logarithmic) loss

$$\ell(r,\hat{r}) := \tau(r-\hat{r}), \quad \tau(x) := \ln\sigma(x)$$

that can be optimized directly using a stochastic gradient algorithm on triples (u, j, i) of users $u \in U$, positive items $j \in I_u$ and negative items $i \in I \setminus I_u$.

$$\begin{aligned}
\phi^U(u) :=& \phi^U(u) - \eta\tau'(\hat{r}(j)-\hat{r}(i))(\phi^I(j)-\phi^I(i)) - 2\eta\lambda\phi^U(u)\\
\phi^I(j) :=& \phi^I(j) - \eta\tau'(\hat{r}(j)-\hat{r}(i))\phi^U(u) - 2\eta\lambda\phi^I(j)\\
\phi^I(i) :=& \phi^I(i) + \eta\tau'(\hat{r}(j)-\hat{r}(i))\phi^U(u) - 2\eta\lambda\phi^I(i)
\end{aligned}$$

with

$$\tau'(x) := 1-\sigma(x)$$

This model is known as Bayesian Personalized Ranking (BPR) [19].

2.4 User and Item Attributes

In practice, the assumption that users and items are entities about which there is nothing else known often is too restrictive and does not make use of some descriptive information about them. For example, in e-commerce, there are a lot of attributes of the items (products) easily available and there are some known attributes of users (customers). In these scenarios we say we have attributes of users or items. We model them by functions

$$a^U : U \to \mathbb{R}^{n_U}, \quad \text{and} \quad a^I : I \to \mathbb{R}^{n_I}$$

respectively.

Early on models have been developed that partition the recommendation problem into (independent) subproblems for each user, trying to predict the rating or item choice based solely on the item attributes (content-based filtering [5]). As this completely disregards all collaborative information from other users, such models provide useful results only in specific settings where no such information is available (see the new item problem in Section 2.5) or as component models in ensembles (sometimes called hybrid models in the context of recommender systems).

Nowadays, user and item attributes are understood as a second auxiliary relation in a multi-relational setting. As the user and item attribute relation share a nominal variable with many levels with the target relation, it makes sense to factorize both relations and share the features of the shared variable.

The resulting models are called multi-relational matrix factorization models [24, 14]. For the case of user and item attributes, the loss of such a model looks like

$$\ell(\phi^U, \phi^I, \phi^{AU}, \phi^{AI}) := \ell(r, \phi^U(\phi^I)^T) + \lambda_{AU}\ell(a^U, \phi^U(\phi^{AU})^T) + \lambda_{AI}\ell(a^I, \phi^I(\phi^{AI})^T), \quad \lambda_{AU}, \lambda_{AI} \in \mathbb{R}_0^+$$

Such a model could be easily learned again by stochastic gradient descent, sequentially sampling tuples from the different input matrices. The effect of factorizing an auxiliary relation could be understood as data-dependent regularization as we push the latent features ϕ^U in a direction where they can be used to reconstruct not just the target rating or item choice relation, but also the auxiliary user attribute relation. The weights λ_{AU} and λ_{AI} determine how strong this regularization effect should be. As any other regularization parameters they have to be learned as hyperparameters.

Auxiliary information such as user and item attributes obviously are only useful when there is not too much primary information, i. e., ratings or item choices. For some large real recommendation scenarios it has been shown experimentally that collaborative models based on 10 ratings about an item provide better predictions than content-based models on thousands of attributes [16]. So user and item attributes mostly are been useful for users and items that recently joined a system and for whom/which only little rating/item choice information is available (recent user / recent item problems) or none at all (new user / new item problems; see next section).

2.5 New User and New Item Problems

A specific class of problems in recommender systems are the so-called new user or new item problems (also called cold-start problems). A new user problem describes the situation of a new user entering the system, so that this user did not yet have rated or choosen any items. In this case obviously none of the personalized models discussed so far could provide any recommendations. In practice these problems cover important cases: new users should not be scared away by getting bad or no recommendations in the beginning, and new items should not have to wait until they are found and taken up by users by chance.

For new users, one could resort to content-based filtering, i. e., to build a separate model for each item that predicts the rating / item choice as function of the user attributes. Better models have been researched where such content-based models are mediated by collaborative models for the ratings / item choices [21].

Besides the question how to deal with new users and new items, the active learning scenario is of interest for recommender systems, i. e., ratings about which items to ask a new or recent user so that his preferences could be learned as quickly as possible [17, 8].

2.6 Context-aware and Multi-Mode Recommendations

Traditionally, recommender systems describe an interaction between two entities, users and items. In many scenarios further entities may moderate this interaction and influence a users preference for an item, e. g., the mood of the target user, the actual location, the actual time, the task the target user is pursueing, the group the target user is with, etc. These further circumstances or modes initially have been described in the literature by different names (location-aware recommendations, time-aware recommendations, group recommendations, etc.), but now often collectively are called context-aware recommendations, where the context could be the mood, location, time, etc. Abstracting from the names of the different entities, this problem could be described as a multi-mode recommendation problem (in the literature usually called multidimensional [2, 4]). Formally, the multi-mode rating prediction is as follows: given

- a set $\mathcal{E}$ of entity classes, where each entity class $E \in \mathcal{E}$ is a set of entity instances (i. e., a set of users, items, moods, etc.),
- a set $\mathcal{R} \subseteq \mathbb{R}$ of ratings, e. g., $\mathcal{R} := \{1, 2, 3, 4, 5\}$,
- a set $\mathcal{D}^{\text{train}} \subseteq \prod_{E \in \mathcal{E}} E \times \mathcal{R}$ of (entity$_1$, entity$_2$, ..., entity$_{|\mathcal{E}|}$, rating) tuples,
- a (rating) loss function $\ell : \mathcal{R} \times \mathbb{R} \to \mathbb{R}$ where $\ell(r, \hat{r})$ quantifies how bad it is to predict rating $\hat{r}$ if the actual rating is r.

Sought is the prediction of the rating for an entity instance of each class, i. e.,

$$\hat{r} : \prod_{E \in \mathcal{E}} E \to \mathbb{R}$$

s. t. for some test set $\mathcal{D}^{\text{test}} \subseteq \prod_{E \in \mathcal{E}} E \times \mathcal{R}$ of (entity$_1$, entity$_2$, ..., entity$_{|\mathcal{E}|}$, rating) tuples (from the same unknown distribution as the train set and not available for the construction of $\hat{r}$) the test risk

$$\text{risk}(\hat{r}; \mathcal{D}^{\text{test}}) := \frac{1}{|\mathcal{D}^{\text{test}}|} \sum_{(e_1, \ldots, e_{|\mathcal{E}|}, r) \in \mathcal{D}^{\text{test}}} \ell(r, \hat{r}(e_1, \ldots, e_{|\mathcal{E}|}))$$

is minimal.

The item prediction problem can be generalized in the same way to a multi-mode item prediction problem. Specifically for item prediction, it sometimes is interesting to predict another mode than the item mode. As we have seen in section 1.2, in social tagging systems, tags can be described as a third mode, and there it will be interesting to predict for a given user and item (resource), which tags he is likely to use for the item. Any multi-modal item prediction model obviously can be used to predict any mode by just subsituting mode names.

If all modes are nominal (as the user and item modes), a multi-mode recommendation problem can be modeled by means of factorization models of higher order, so called tensor factorization models [18, 12].

Related to, but different from the multi-mode rating recommendation problem is the multi-criteria recommendation problem [1]. Here, the rating is not just a single (ordinal) overall rating, but a compound rating reflecting different criteria or different aspects of the item individually, e. g., for movies suspense, emotion and humor, or for holiday accommodations location, comfort, friendliness, etc. The difference between multi-criteria recommendation problems and multi-mode recommendation problems is that in the latter all modes are observed for a test case, but for multi-criteria recommendations ratings for other criteria are not observed, i. e., not available to base the prediction upon.

If a problem is described as a multi-mode or a multi-criteria problem reflects a specific requirement of the application. For example, item recommendation in a social tagging system has useful (but different) applications as both, as multi-mode recommendation problem and as multi-criteria recommendation problem. As multi-mode recommendation problem we try to predict for a given user and a given set of tags, which items the user may be looking for. Here, the tags may describe the context in which the user is looking for items. On the other hand, as multi-criteria recommendation problem we are treating the tags as a (nominal) rating, so for a given user we are looking for interesting items (and eventually in parallel for tags he may later associate with that item).

For reference in the remaining chapters, we instantiate the context-aware multi-mode item recommendation problem for tags in social systems, for short called tag recommendation: given

- sets U, R, and T of users, resources, and tags, respectively,
- a set $Y := \mathcal{D}^{\text{train}} \subseteq U \times R \times T$ of user/resource/tag triples,
- a (ranking) loss function $\ell : \mathcal{P}(T) \times \mathbb{R}^T \to \mathbb{R}$, e. g., recall at k

$$\text{recall}_k(S, \hat{s}) := \frac{1}{|S|} |S \cap \underset{t' \in T}{\overset{k}{\operatorname{argmax}}}\, \hat{s}(t')|, \quad S \subseteq T$$

Sought is for every user/resource pair a ranking of the tags, i. e., a score function

$$\hat{s} : U \times R \to \mathbb{R}^T \quad \text{or equivalently } \hat{s} : U \times R \times T \to \mathbb{R}$$

s. t. for some test set $\mathcal{D}^{\text{test}} \subseteq U \times R \times T$ of user/resource/tag triples (from the same unknown distribution as the train set and not available for the construction of $\hat{s}$) the test risk

$$
\begin{aligned}
\mathrm{risk}(\hat{s};\mathcal{D}^{\mathrm{test}}) :=& \frac{1}{|U\!R(\mathcal{D}^{\mathrm{test}})|}\sum_{(u,r)\in U\!R(\mathcal{D}^{\mathrm{test}})} \ell(T_{u,r}(\mathcal{D}^{\mathrm{test}}),\hat{s}(u,r)),\\
\text{with } U\!R(\mathcal{D}) :=& \{(u,r)\in U\times R \mid \exists t\in T:(u,r,t)\in\mathcal{D}\},\\
T_{u,r}(\mathcal{D}) :=& \{t\in T\mid (u,r,t)\in\mathcal{D}\}
\end{aligned}
$$

is minimal.

References

1. G. Adomavicius and Y. O Kwon. New recommendation techniques for multicriteria rating systems. *IEEE Intelligent Systems*, page 48–55, 2007.
2. G. Adomavicius and A. Tuzhilin. Multidimensional recommender systems: a data warehousing approach. *Electronic Commerce*, page 180–192, 2001.
3. G. Adomavicius and A. Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE transactions on knowledge and data engineering*, page 734–749, 2005.
4. G. Adomavicius and A. Tuzhilin. Context-aware recommender systems. In *Proceedings of the 2008 ACM conference on Recommender systems RecSys 08*, volume 16, 2008.
5. M. Balabanović and Y. Shoham. Fab: content-based, collaborative recommendation. *Communications of the ACM*, 40(3):66–72, 1997.
6. E. J Candès and B. Recht. Exact matrix completion via convex optimization. *Foundations of Computational Mathematics*, 9(6):717–772, 2009.
7. D. Goldberg, D. Nichols, B. M Oki, and D. Terry. Using collaborative filtering to weave an information tapestry. *Communications of the ACM*, 35(12):61–70, 1992.
8. A. S Harpale and Y. Yang. Personalized active learning for collaborative filtering. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, page 91–98, 2008.
9. T. Hofmann and J. Puzicha. Latent class models for collaborative filtering. In *International Joint Conference on Artificial Intelligence*, volume 16, page 688–693, 1999.
10. Thomas Hofmann. Latent semantic models for collaborative filtering. *ACM Transactions on Information Systems*, 22(1):89–115, January 2004.
11. Dietmar Jannach, Markus Zanker, Alexander Felfernig, and Gerhard Friedich. *Recommender Systems: An Introduction*. Cambridge University Press, 1 edition, September 2010.

12. A. Karatzoglou, X. Amatriain, L. Baltrunas, and N. Oliver. Multiverse recommendation: n-dimensional tensor factorization for context-aware collaborative filtering. In *Proceedings of the fourth ACM conference on Recommender systems*, page 79–86, 2010.
13. Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):37, 30, 2009.
14. C. Lippert, S. H Weber, Y. Huang, V. Tresp, M. Schubert, and H. P Kriegel. Relation-prediction in multi-relational domains using matrix-factorization. In *NIPS Workshop: Structured Input-Structured Output*, 2008.
15. Ulrich Paquet, Blaise Thomson, and Ole Winther. A hierarchical model for ordinal matrix factorization. *Statistics and Computing*, June 2011.
16. I. Pilászy and D. Tikk. Recommending new movies: even a few ratings are more valuable than metadata. In *Proceedings of the third ACM conference on Recommender systems*, page 93–100, 2009.
17. A. M Rashid, I. Albert, D. Cosley, S. K Lam, S. M McNee, J. A Konstan, and J. Riedl. Getting to know you: learning new user preferences in recommender systems. In *Proceedings of the 7th international conference on Intelligent user interfaces*, page 127–134, 2002.
18. Steffen Rendle. *Context-Aware Ranking with Factorization Models*. Springer Berlin Heidelberg, 1st edition, November 2010.
19. Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Schmidt-Thieme Lars. BPR: Bayesian personalized ranking from implicit feedback. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, UAI '09, pages 452–461, Arlington, Virginia, United States, 2009. AUAI Press.
20. P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl. GroupLens: an open architecture for collaborative filtering of netnews. In *Proceedings of the 1994 ACM conference on Computer supported cooperative work*, page 175–186, 1994.
21. A. I Schein, A. Popescul, L. H Ungar, and D. M Pennock. Methods and metrics for cold-start recommendations. In *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, page 253–260, 2002.
22. L. Schmidt-Thieme. Compound classification models for recommender systems. In *Fifth IEEE International Conference on Data Mining (ICDM'05)*, pages 378–385, Houston, TX, USA, 2005.
23. U. Shardanand and P. Maes. Social information filtering: algorithms for automating "word of mouth". In *Proceedings of the SIGCHI conference on Human factors in computing systems*, page 210–217, 1995.
24. Ajit P. Singh and Geoffrey J. Gordon. Relational learning via collective matrix factorization. In *Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 650–658, Las Vegas, Nevada, USA, 2008. ACM.

25. N. Srebro and T. Jaakkola. Weighted low-rank approximations. In *Proceedings of the 25th international conference on Machine learning*, volume 20, pages 720–727, Helsinki, Finland, 2003.
26. M. Weimer, A. Karatzoglou, Q. Le, and A. Smola. Cofi rank-maximum margin matrix factorization for collaborative ranking. *Advances in neural information processing systems*, 20:1593–1600, 2008.

Springer

http://www.springer.com/978-1-4614-1893-1

Recommender Systems for Social Tagging Systems

Balby Marinho, L.; Hotho, A.; Jäschke, R.; Nanopoulos, A.; Rendle, S.; Schmidt-Thieme, L.; Stumme, G.; Symeonidis, P.

2012, IX, 111p., Softcover
ISBN: 978-1-4614-1893-1