Improving Collaborative Filtering Recommendations Using External Data

Akhmed Umyarov
New York University
aumyarov@stern.nyu.edu

Alexander Tuzhilin
New York University
atuzhili@stern.nyu.edu

Abstract

This paper describes an approach for incorporating externally specified aggregate ratings information into certain types of collaborative filtering (CF) methods. For a statistical model-based CF approach, we formally showed that this additional aggregated information provides more accurate recommendations of individual items to individual users. Furthermore, theoretical insights gained from the analysis of this model-based method suggested a way to incorporate aggregate information into the heuristic item-based CF method. Both the model-based and the heuristic item-based CF methods were empirically tested on several datasets, and the experiments uniformly confirmed that the aggregate rating information indeed improves CF recommendations. These results also show the power of theory by demonstrating how the insights gained from theoretical developments can shed light on proper selection of good heuristic methods. We also showed the way to introduce scalability and parallelization into the estimation procedure and reported the running time for steps of the estimation procedure for large datasets.

1 Introduction

Consider a Netflix recommender system [7] and assume that it is augmented with the aggregate ratings from the IMDB database [13], such as the one specifying that females in the age category of 18 to 29 gave an average rating of 6.9 (out of 10) to the movie "Madagascar." Can such additional aggregate rating information, provided from the external sources, improve the quality of individual ratings? More generally, a traditional recommender system providing individual ratings to individual users can be supplemented with an *externally* provided OLAP-based [1] system of aggregate ratings, such as the aggregate ratings for "Madagascar" provided by females vs. provided by females in the age category of 18 to 29 years.

[20] studied this question for the special type of a statistical model reported in the marketing literature [5] and showed that these externally obtained aggregate ratings can be converted into additional constraints on the model estimating individual ratings. [20] also theoretically demonstrated that these additional constraints provide for better estimation of unknown ratings.

Although useful and having nice statistical properties, the model from [5] is not widely used in the field of recommender systems. Therefore, in this paper, we study how the aggregate ratings information can be used in the *collaborative filtering (CF)* systems constituting the "bread-and-butter" of recommender systems. To this end, we show in this paper that these *external aggregate* ratings can be used for providing better recommendations of *individual* items to *individual* users for a certain class of collaborative filtering systems.

More specifically, we consider two types of CF systems: *model-based* and classical *item-to-item based CF* [17]. We selected the model-based CF approach because it was grounded in fundamental statistical theory, and, therefore, we wanted to gain some insights on how to handle aggregate ratings based on that theory. We also selected the classical heuristic based [17] item-to-item CF method because of its popularity in the recommender systems community and because it is used in some popular systems, including Amazon's recommender system [15]. We also show how to expand the theoretical insights obtained from the model-based approach to developing better recommendation methods for the heuristic-based CF approach.

We made the following contributions in the paper: for the model-based and the CF models approaches we

- showed how to apply the aggregate rating information to estimating individual ratings for individual users
- theoretically and experimentally showed that the aggregate ratings information indeed helps to provide better recommendations for the the model-based CF
- applied the insights gained from our studies of the model-based CF system to the heuristics-based item-

1550-4786/08 $25.00 © 2008 IEEE
DOI 10.1109/ICDM.2008.44

to-item CF approach to develop a new aggregation-based item-to-item CF method

- experimentally showed that these insights indeed lead to better recommendations for the classical item-to-item CF. This result shows how theoretical analysis can provide further insights for improving some heuristic-based methods
- showed how to scale the proposed methods to large datasets also taking advantage of highly parallelizable nature of the algorithm

The usage of aggregate ratings has been previously studied in the recommender systems literature. An idea of using an OLAP-based multidimensional approach to recommender systems was proposed by [2]. This approach was subsequently extended by [1] by incorporating additional contextual information on ratings, such as when, how and with whom the movie was watched. Also, [16] presents a method for providing recommendations to a group of users. [14] discusses new issues that arise when one considers web-based personalization involving groups for a certain subclass of group recommender systems. Both methods deal with the bottom-up approach to recommendations that used aggregate ratings as a basis for recommendations to groups of users. In contrast to this, [9] presents a top-down method for using aggregate information about traversal of hypertext pages by a group of users in order to provide better recommendations of hypertext pages to individual members of the group. [6] presents a two-level rating estimation method where at the lower level ratings are estimated using collaborative filtering deploying local scale neighborhood information. At the upper level, [6] uses SVD-style factorization based on global scale information to improve predictions. However, this work does not use any information on prespecified taxonomy of users or items, nor does it use externally specified aggregate ratings. [4] uses pre-existing taxonomy of webpages and advertisements in order to better estimate the click-through rate and combat the sparsity of the data. However, this work is only tangentially related to recommender systems, also does not use any externally specified aggregate information and does not deal with aggregate ratings. [20] presents the preliminary work on aggregate ratings model based on [5] and analyses it at the theoretical level. However, none of this prior work focused on using aggregate ratings information to improve the CF methods considered in this paper.

The rest of the paper is organized as follows. In Section 2 we describe a model-based approach to collaborative filtering from [19] and the estimation procedure that we implemented in order to estimate unknown parameters of the model. In Section 3 we describe our new method of adding aggregate information into this model, get theoretical insights on how this information is introduced and prove theoretically that the aggregate information improves performance. In Section 4 we use these theoretical insights obtained from the model-based approach in order to introduce the aggregate information into classical heuristic item-based collaborative filtering. In Sections 5 and 6, we show empirically on several datasets that the significant improvement is achieved for both model-based CF and item-based CF when the aggregate information is introduced using the suggested method. In Section 7, we address scalability issues for our method.

2 Model Specification

First, we present a model-based approach to CF that is grounded in the fundamentals of statistical theory. In particular, we follow [19] in this section when describing such an approach. Then the insights from this theoretical development are applied to handle the classical item-based CF in Section 4.

Assume we have a set of N users and M items. Denote r_{ij} an observed *or* unobserved rating by user i for item j. Moreover, for a specific item j, denote the vector $\boldsymbol{r}_j = (r_{1j}, r_{2j}, \ldots, r_{Nj})'$ a vector of ratings of all N users for item j. We assume that all vectors $\boldsymbol{r}_j$ are i.i.d draws from a multivariate normal distribution with some unknown mean vector $\boldsymbol{\mu}$ and unknown covariance matrix Σ:

$$\boldsymbol{r}_j \sim N(\boldsymbol{\mu}, \Sigma) \quad (1)$$

We also assume that for each j, we do not observe the vector $\boldsymbol{r}_j$ completely, but only observe some subset of ratings explicitly provided by some subset of users $K(j)$.

The goal of this recommender system is to estimate an unobserved rating r_{ij} from the set of observed ratings $\{r_{kl}\}$ and parameters of the model $\boldsymbol{\mu}$, Σ. According to [8], the least mean squared error unbiased estimator for (1) is:

$$\hat{r}_{ij} = E[r_{ij} \mid \text{observed } \{r_{kl}\}, \boldsymbol{\mu}, \Sigma]$$

For item j, consider the vector of observed ratings $\boldsymbol{r}_{Kj}$, where $K = K(j)$ is a set of users whose ratings we have observed for item j, and the vector of unobserved ratings $\boldsymbol{r}_{Uj}$, where $U = U(j)$ is a set of users whose ratings we have not observed. From the assumption that $\boldsymbol{r}_j$ is drawn from multivariate normal distribution, we conclude that $(\boldsymbol{r}_{Uj}, \boldsymbol{r}_{Kj})$ is also drawn from the following multivariate normal distribution:

$$\begin{pmatrix} \boldsymbol{r}_{Uj} \\ \boldsymbol{r}_{Kj} \end{pmatrix} \sim N\left(\begin{pmatrix} \boldsymbol{\mu}_1 \\ \boldsymbol{\mu}_2 \end{pmatrix}, \begin{pmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{pmatrix} \right) \quad (2)$$

As it is shown in [11], the conditional expected value has the following form:

$$\hat{\boldsymbol{r}}_{Uj} = E[\boldsymbol{r}_{Uj} | \boldsymbol{r}_{Kj} = \boldsymbol{y}] = \boldsymbol{\mu}_1 + \Sigma_{12}\Sigma_{22}^{-1}(\boldsymbol{y} - \boldsymbol{\mu}_2) \quad (3)$$

We call the estimator $\hat{\boldsymbol{r}}_{Uj}$ *unconstrained rating estimator (URE)* for the reasons that will become clear below when we introduce aggregate information as a constraint.

Equation (3) provides us a direct method for computing the estimator of unobserved ratings $\boldsymbol{r}_{Uj}$. However, we must take into account that the parameters $\boldsymbol{\mu}$ and Σ of the model (1) are unobserved as well. Following [19] and [12], they can be estimated using our prior beliefs about the parameters and the observed ratings as follows.

We follow the standard assumption in Bayesian statistics [12] that our prior beliefs are conjugate priors on $\boldsymbol{\mu}$ and Σ:

$$\Sigma \sim \text{Inv-Wishart}_{\nu_0}(\Lambda_0^{-1})$$

$$\boldsymbol{\mu}|\Sigma \sim N\left(\boldsymbol{\mu}_0, \frac{1}{k_0}\Sigma\right)$$

where ν_0, Λ_0, $\boldsymbol{\mu}_0$, k_0 are hyper-parameters of the model, that is, parameters specifying our prior belief about parameters Σ and $\boldsymbol{\mu}$ before observing the data. The scalar hyper-parameter ν_0 describes the degrees of freedom and the matrix Λ_0 describes the scale of inverse-Wishart distribution. The vector hyper-parameter $\boldsymbol{\mu}_0$ is the prior mean and the scalar k_0 is the scaling of prior variance.

In order to find the point estimates of unobserved parameters $\boldsymbol{\mu}$ and Σ, we find the values $\boldsymbol{\mu}^*$ and Σ^* that maximize the posterior probability $P(\boldsymbol{\mu}, \Sigma|\text{ observed } \{r_{kl}\})$:

$$\underbrace{P(\boldsymbol{\mu}, \Sigma|\text{ observed}\{r_{kl}\})}_{\text{posterior belief}} \propto \underbrace{P(\text{observed}\{r_{kl}\}|\boldsymbol{\mu}, \Sigma)}_{\text{likelihood}} \underbrace{P(\boldsymbol{\mu}, \Sigma)}_{\text{prior belief}}$$

In [19], the parameters were estimated using expectation-maximization algorithm. However, that approach did not work well on our data, since their algorithm converged very slowly to a local optimum for us.

Therefore, in this paper we developed the following alternative method for estimating parameters $\boldsymbol{\mu}$ and Σ for model (1) by following the ideas from [12] and taking into account our likelihood function (1). After some algebra, we find that the negative logarithm of posterior distribution corresponds to the following expression (up to a constant term):

$$\begin{aligned} -\log P(\boldsymbol{\mu}, \Sigma|\text{ observed}\{r_{kl}\}) &= \left(\frac{\nu_0 + N}{2} + 1\right)\log|\Sigma| + \\ &+ \frac{1}{2}\text{tr}\left(\Lambda_0\Sigma^{-1}\right) + \frac{k_0}{2}(\boldsymbol{\mu} - \boldsymbol{\mu}_0)'\Sigma^{-1}(\boldsymbol{\mu} - \boldsymbol{\mu}_0) + \\ &+ \sum_{j=1}^{M} \frac{(\boldsymbol{r}_{Ki} - \boldsymbol{\mu}_K)'\Sigma_K^{-1}(\boldsymbol{r}_{Kj} - \boldsymbol{\mu}_K)}{2} + \frac{1}{2}\sum_{j=1}^{M}\log|\Sigma_K| \end{aligned} \tag{4}$$

where $K = K(j)$ is the ordered set of users whose ratings we observed for item j, $\boldsymbol{\mu}_K$ is a subvector of $\boldsymbol{\mu}$ corresponding to mean ratings of the users from the set $K(j)$, Σ_K is a submatrix of Σ corresponding to rating covariance matrix of the users from the set $K(j)$.

Therefore, the point estimates for unobserved parameters $\boldsymbol{\mu}$ and Σ that are required for our analysis can be found by minimizing the expression (4) with respect to $\boldsymbol{\mu}$ and Σ, thus maximizing their posterior probability.

The minimization can be done using the following gradient descent iterative procedure. First, we compute the gradient of the negative log posterior (4) as follows. Let us denote elements of some index set K as $(k_1, \ldots, k_{|K|})$. Then we introduce the matrix L_K of size $N \times |K|$ as follows:

$$\begin{cases} L_{k_i,i} = 1 & \forall i \in [1, \ldots, |K|] \\ L_{i,j} = 0 & \text{for all other elements} \end{cases}$$

Intuitively, if we multiply any matrix A by the matrix L_K, then we just swap and arrange columns of A according to ordered set K and remove from A the columns corresponding to numbers that are not in K. For example,

$$\Sigma_K = L_K'\Sigma L_K$$

Then the gradient of (4) w.r.t. parameter $\boldsymbol{\mu}$ is

$$\frac{\partial(-\log P)}{\partial\boldsymbol{\mu}} = k_0\Sigma^{-1}(\boldsymbol{\mu} - \boldsymbol{\mu}_0) + \sum_{j=1}^{M} L_K\Sigma_K^{-1}(\boldsymbol{\mu}_K - \boldsymbol{r}_{Kj})$$

The gradient of (4) w.r.t. parameter Σ is

$$\begin{aligned} \frac{\partial(-\log P)}{\partial\Sigma_{ij}} &= \left(\frac{\nu_0 + N}{2} + 1\right)\text{tr}\left[\Sigma^{-1}\frac{\partial\Sigma}{\partial\Sigma_{ij}}\right] + \\ &+ \frac{1}{2}\text{tr}\left[-\Lambda_0\Sigma^{-1}\frac{\partial\Sigma}{\partial\Sigma_{ij}}\Sigma^{-1}\right] - \\ &- \frac{k_0}{2}(\boldsymbol{\mu} - \boldsymbol{\mu}_0)'\Sigma^{-1}\frac{\partial\Sigma}{\partial\Sigma_{ij}}\Sigma^{-1}(\boldsymbol{\mu} - \boldsymbol{\mu}_0) + \\ &- \sum_{j=1}^{M}\frac{1}{2}\left[(\boldsymbol{r}_{Ki} - \boldsymbol{\mu}_K)'\,(L_K'\Sigma L_K)^{-1}L_K'\frac{\partial\Sigma}{\partial\Sigma_{ij}}L_K \times \right. \\ &\left. \times (L_K'\Sigma L_K)^{-1}(\boldsymbol{r}_{Kj} - \boldsymbol{\mu}_K)\right] + \\ &+ \frac{1}{2}\sum_{j=1}^{M}\text{tr}\left[(L_K'\Sigma L_K)^{-1}L_K\frac{\partial\Sigma}{\partial\Sigma_{ij}}L_K\right] \end{aligned}$$

Second, after defining this gradient, we estimate parameters $\boldsymbol{\mu}$ and Σ using the line search gradient descent procedure [10] that guaranteed to converge to a local minimum.

The estimates $\hat{\boldsymbol{\mu}}$ and $\hat{\Sigma}$ that are obtained after convergence of this algorithm are substituted into equation (3) for computing ratings predictions.

3 Aggregated Ratings Model

In this section we describe a method of adding aggregate information to the unconstrained (URE) collaborative filtering model presented in Section 2. We assume that we have some external source of information from which we also observe an aggregate rating $r^a = \frac{1}{N}\sum_{i=1}^{N} r_{ij}$ for a particular item j.[1] In particular, assume that:

$$r^a = \frac{1}{N}\sum_{i=1}^{N} r_{ij} = a_j = \alpha_j + \varepsilon_j, \quad \varepsilon \sim N(0, \sigma_j^2) \tag{5}$$

where a_j is the observed noisy average rating for item j, α_j is the unobserved true value of the average rating, ε_j is a noise component, σ_j is a known item-specific parameter of the noise.

For example, assume we are using the Netflix Prize movie rating dataset [7] to predict the rating of "Madagascar" for a particular user and we also know from IMDB [13] that the average rating r^a for "Madagascar" is $a_j = 6.5$ with unobserved noise ε_j having the aggregate noise uncertainty $\sigma_j = 0.15$ for this movie. Note that, in general, the external aggregate ratings *may come from a sample that is different from the sample of the given individual ratings*, including having different distribution properties. However, *the noise term ε allows us to handle such occasions* by choosing σ accordingly. More specifically, if the sample for the aggregate information is quite different in their characteristics from the sample of individual information, then specifying high σ will allow to accommodate for the inappropriateness of the aggregate rating for the particular sample and force the estimation procedure not to treat this information as precise.

In this model, the joint distribution of the observed, the unobserved and the aggregate ratings is a multivariate normal:

$$\begin{pmatrix} \boldsymbol{r}_{Uj} \\ \boldsymbol{r}_{Kj} \\ r^a \end{pmatrix} \sim N\left(\begin{pmatrix} \boldsymbol{\mu}_1 \\ \boldsymbol{\mu}_2 \\ \mu^a \end{pmatrix}, \begin{pmatrix} \Sigma_{11} & \Sigma_{12} & \Sigma_{13} \\ \Sigma_{21} & \Sigma_{22} & \Sigma_{23} \\ \Sigma_{31} & \Sigma_{32} & \Sigma_{33} \end{pmatrix}\right) \tag{6}$$

where Σ_{11}, Σ_{12}, Σ_{21} and Σ_{22} are as in (2). Since

$$\text{cov}(r_{ij}, r^a) = \text{cov}\left(r_{ij}, \frac{1}{N}\sum_{k=1}^{N} r_{kj} + \varepsilon\right) =$$

$$= \frac{1}{N}\sum_{k=1}^{N} \text{cov}\left(r_{ij}, r_{kj}\right) \tag{7}$$

the matrix Σ_{31} is just an average of all rows of Σ_{11} and Σ_{21}. Similar analysis applies to Σ_{32} which is an average of all rows of Σ_{12} and Σ_{22}. To compute Σ_{33}, we use the following

$$\text{cov}(r^a, r^a) = \text{cov}\left(\frac{1}{N}\sum_{i=1}^{N} r_{ij} + \varepsilon, \frac{1}{N}\sum_{k=1}^{N} r_{kj} + \varepsilon\right) =$$

$$= \frac{1}{N^2}\sum_{i=1}^{N}\sum_{k=1}^{N} \text{cov}\left(r_{ij}, r_{kj}\right) + \sigma^2 \tag{8}$$

That is, Σ_{33} is the average of all the elements of Σ_{11}, Σ_{12}, Σ_{21} and Σ_{22} and the variance of the aggregate noise σ^2.

Therefore, following the ideas from [11], as in the case of equation (3), the least mean squared error unbiased estimator that takes into account the observed aggregate information (5) is

$$\hat{\boldsymbol{r}}^*_{Uj} = E[\boldsymbol{r}_{Uj} | \boldsymbol{r}_{Kj} = \boldsymbol{y}, r^a = a] = \boldsymbol{\mu}_1 +$$

$$+ \begin{pmatrix} \Sigma_{12} & \Sigma_{13} \end{pmatrix} \begin{pmatrix} \Sigma_{22} & \Sigma_{23} \\ \Sigma_{32} & \Sigma_{33} \end{pmatrix}^{-1} \begin{pmatrix} \boldsymbol{y} - \boldsymbol{\mu}_2 \\ a - \mu^a \end{pmatrix} \tag{9}$$

We call the estimator (9) the *constrained rating estimator (CRE)*, since besides using the observed ratings $\boldsymbol{r}_{Kj}$, it also incorporates the additional constraint of the type (5).

Theorem 1. *The expected mean squared error of the constrained rating estimator (CRE) is smaller than the expected mean squared error of the unconstrained rating estimator (URE).*

Proof. From (6), (9) and the properties of multivariate normal distribution, we conclude that the conditional distribution of $(\boldsymbol{r}_{Uj}, r^a)'$, given that $\boldsymbol{r}_{Kj} = \boldsymbol{a}$, is also a multivariate normal distribution with the covariance matrix:

$$\text{Var}\left[\begin{pmatrix} \boldsymbol{r}_{Uj} \\ r^a \end{pmatrix} \middle| \boldsymbol{r}_{Kj} = \boldsymbol{y}\right] = \begin{pmatrix} S_{11} & S_{12} \\ S_{21} & S_{22} \end{pmatrix} \tag{10}$$

Therefore, the variance of the estimator without aggregate ratings is

$$\text{Var}[\boldsymbol{r}_{Uj} | \boldsymbol{r}_{Kj} = \boldsymbol{y}] = S_{11}$$

However, as it follows from (10) and the properties of multivariate normal distribution [11], the variance of the estimator with aggregate information is

$$\text{Var}[\boldsymbol{r}_{Uj} | \boldsymbol{r}_{Kj} = \boldsymbol{y}, r^a = k] = S_{11} - S_{12}S_{22}^{-1}S_{21}$$

Since S_{22} is a non-negative definite matrix, $S_{12}S_{22}^{-1}S_{21}$ is also a non-negative definite matrix. Therefore,

$$\text{Var}[\boldsymbol{r}_{Uj} | \boldsymbol{r}_{Kj} = \boldsymbol{y}, r^a = k] \preceq \text{Var}[\boldsymbol{r}_{Uj} | \boldsymbol{r}_{Kj} = \boldsymbol{y}]$$

That is, in terms of comparison of non-negative definite matrices, the covariance matrix of CRE is "smaller" than the

[1] The assumption that the average rating is computed only over item j is for algebraic convenience only. The theoretical results can be easily generalized to the case of the average rating across any arbitrary segment of users and items.

covariance matrix of URE. This implies that the standard errors of CRE are also smaller.

Since both estimators $\hat{r}^*_{ij}$ and $\hat{r}_{ij}$ are unbiased, then lower standard error of the estimator implies lower mean squared error of predictions [8]. □

Define an *aggregate correction term* T_{ij} using expression

$$\hat{r}^*_{ij} = \hat{r}_{ij} + T_{ij} \qquad (11)$$

where $\hat{r}_{ij}$ is the unconstrained estimator from (3) and $\hat{r}^*_{ij}$ is the constrained estimator from (9). Subtracting (3) from (9), we get the following expression for the vector of correction terms $\boldsymbol{T}_{Uj}$:

$$\boldsymbol{T}_{Uj} = \begin{pmatrix} \Sigma_{12} & \Sigma_{13} \end{pmatrix} \times \qquad (12)$$

$$\times \left[\begin{pmatrix} \Sigma_{22} & \Sigma_{23} \\ \Sigma_{32} & \Sigma_{33} \end{pmatrix}^{-1} - \begin{pmatrix} \Sigma_{22}^{-1} & 0 \\ 0 & 0 \end{pmatrix} \right] \begin{pmatrix} \boldsymbol{y} - \boldsymbol{\mu}_2 \\ a - \mu^a \end{pmatrix}$$

where $\boldsymbol{T}_{Uj}$ is a vector of individual correction terms for all unobserved ratings $U = U(j)$ for item j.

From this definition of T_{ij}, we may consider the process of introducing external aggregate information as an addition of aggregate correction term T_{ij} to the standard model (3). As Theorem 1 shows, adding the correction term to the URE estimator (3) must improve the performance of the model. Moreover, this result holds not only in theory, but it is also confirmed on the real-life rating data as shown in Section 6.

The case of observing multiple aggregate ratings r^{a1},. . .,r^{al} for a particular item j is a simple generalization from the described case of observing just a single rating r^a for item j. Moreover, these aggregate ratings can be specified over specific segments of users and groups of items, such as an aggregate rating taken over females living in New York and over Woody Allen's comedies, thus, implicitly introducing user segmentations and item groupings into the model.

In the next section, we apply this correction term approach (11) when we incorporate the aggregate rating information into the classical item-based collaborative filtering.

4 Item-based collaborative filtering

We follow the standard approach of [17] to the item-based CF in this section and show how it can be improved using the aggregate information and some of the ideas from Section 3.

Item-based collaborative filtering is one of the most popular recommendations techniques used widely in industry by such companies as Amazon [15, 18]. The item-based approach attempts to determine a user's rating for an item based on the ratings of similar items this user rated in the past, where the similarity between two items is established based on the correlation between the ratings for these two items. More specifically, for user i and item j, item-based CF estimates the rating r_{ij} as:

$$\hat{r}_{ij} = \frac{\sum_{k \in I(i)} s_{jk} r_{ik}}{\sum_{k \in I(i)} s_{jk}}$$

where $I(i)$ is the set of the items for which ratings by user i are observed and s_{jk} is a measure of "similarity" between item j and item k.

A common measure of similarity between two items j and k is a Pearson correlation coefficient:

$$\hat{s}_{jk} = \frac{\displaystyle\sum_{i \in K(j) \cap K(k)} (r_{ij} - \overline{r_{\cdot j}})(r_{ik} - \overline{r_{\cdot k}})}{\sqrt{\displaystyle\sum_{i \in K(j)} (r_{ij} - \overline{r_{\cdot j}})^2 \sum_{i \in K(j)} (r_{ik} - \overline{r_{\cdot k}})^2}}$$

where $\overline{r_{\cdot j}}$ is a sample average rating for j-th item.

The item-based approach described above falls into the heuristic-based category [3], unlike the model-based URE and CRE estimators described in Sections 2 and 3. Therefore, formal statistical analysis to improving rating estimations presented in Section 3 cannot be directly applied to the item-based approach.

However, we decided to use the theoretical insights obtained from the model-based approach from Section 3 and applied them to the item-based approach as follows. Since the rating estimator $\hat{r}^*_{ij}$ that uses aggregate information has the form defined by equation (11) having the additive correction term T_{ij}, we conjecture that the same correction term may help to improve the item-based CF. In particular, we defined a new item-based rating estimator for the item-based CF method as

$$\hat{r}^*_{ij} = \frac{\sum_{k \in I(i)} s_{jk} r_{ik}}{\sum_{k \in I(i)} s_{jk}} + T_{ij} \qquad (13)$$

where T_{ij} is calculated as in (12). In Section 6, we empirically show that this new rating estimator indeed improves performance of the item-based CF.

5 Empirical settings

To empirically validate the theoretical results from Section 3 and to see if estimator (13) indeed improves the item-based collaborative filtering, we conducted the experiments on the following "real-world" datasets. The first one was the Netflix Competition dataset [7], from which we took three subsamples for validation purposes. The other one pertains to recommending movies and was used in [1]. We also took the external aggregate ratings from the IMDB dataset and applied them to both datasets. In the rest of Section 5 we describe these datasets, and in Section 6 we present the experimental results.

5.1 Samples of the Netflix Prize Dataset

For our experiments we used three random subsamples of the Netflix Prize dataset [7]. Specifically, the first subsample of the Netflix dataset was produced by selecting 100 users from the set of all the Netflix users ranked between 2000 to 2100 based on the total number of ratings they gave. Then we selected 100 random movies that these selected users watched and 3000 random ratings that they provided for those 100 movies[2]. The second and third subsamples of the Netflix dataset were produced in a similar manner, except that we selected 100 random users for each subsample from the group of users ranked between 10,000 and 300,000 based on the total number of ratings they gave. Finally, each dataset was randomly split into ten subsets for the 10-fold cross validation.

For each of the three aforementioned datasets, we introduced external aggregate information from the IMDB database exactly as for the movie database in Section 5.2 with the aggregate rating variance of $\sigma^2 = 0.02$ for movies that received more than 1000 votes. Both IMDB ratings and Netflix Prize dataset ratings were normalized to the $[0, 1]$ interval.

The original Netflix Prize dataset [7] contains no demographic or other information about users except their id number. For movies, Netflix Prize dataset [7] provides the movie title and movie release year. For users' ratings, the time when the rating appeared on the website is also provided by the original dataset.

5.2 Movie Rating Dataset

In order to diversify our sample, we used the data from the study [1] on 61 users that provided 1110 ratings for 62 movies. The dataset was also randomly split into 10 subsets for 10-fold cross validation.

In order to introduce external aggregate information into the data, the average rating from the IMDB database [13] for each movie was used. For each movie in the dataset from [1], we found the average movie rating contained in the IMDB database and used this average rating as the external aggregate rating for that movie.

The dataset from [1] contains demographic information about users such as user's age, gender, home ZIP code and preferences about the context behind the movie watching experience, such as the preferred time and venues for watching movies. For movies, the dataset from [1] provides the movie title and movie release year. For users' ratings, this dataset provides complete description of the context of the movie watching experience, such as when, where and with whom the movie was seen.

6 Results

6.1 Samples of the Netflix Prize Dataset

The graphs in Figures 1, 2 and 3 represent the mean squared error (MSE) performance of the model-based CF and item-based CF as a function of the number of additional aggregate ratings introduced for the movies dataset from several subsets of Netflix Prize Dataset [7]. More specifically, Figures 1, 2 and 3 plot on the x-axis the cumulative number of additional aggregate ratings introduced into the model. The 0-th tick corresponds to the plain basic recommendation model without any aggregate ratings. The 1st tick corresponds to just one aggregate rating of type (5) for the first item as described in Section 3. The 2nd tick adds one more aggregate rating of type (5) for the second item, and so on. On the y-axis we plot the MSE performance of the model based on 10-fold cross-validations described in Section 5.2.

Although the MSE error rates are not monotone, as Figures 1, 2 and 3 demonstrate, the overall drift taken across multiple average ratings is definitely downward, as Figures 1, 2 and 3 clearly show. This experimental finding is in line with Theorem 1 and empirically supports the theory.

Therefore, all the three graphs confirm the theoretical result from Theorem 1 that the predictive rating errors decrease on average, as the number of aggregate ratings increases. The MSE in Figure 1 decreased by 1.25% for model-based CF and by 2% for item-based CF from the case of no aggregate to 96-99 additional aggregate ratings. The MSE in Figure 2 decreased by 1.1% for model-based CF and by 2.14% for item-based CF. The MSE in Figure 3 decreased by 1.5% for model-based CF and by 1.5% for item-based CF.

Despite relatively small numbers, we would like to emphasize that 1% and 2% constitute solid performance improvements, given that they are based on less than 100 additional aggregate ratings. For comparison, the $1,000,000 Grand Prize of the Netflix Prize Competition required performance improvement of 10% for the RMSE. Moreover, if the leading competitor of the Netflix Competition could achieve an MSE performance improvement of 1.7% today (as of July 06, 2008), that competitor would have won the $1,000,000 Netflix Grand Prize.

Furthermore, all these performance improvements do not happen by chance alone. To see this, assume that the reported performance improvements simply constitute random white noise. This assumption would imply that the MSE graphs on Figures 1, 2 and 3 are the results of a random process with zero drift with MSE improvements jump-

[2] Very recently we also used sample datasets of $10,000$ users and $10,000$ movies to empirically test scalability of our methods reported in Section 7. The results of these experiments are presented in our recent technical report [21].

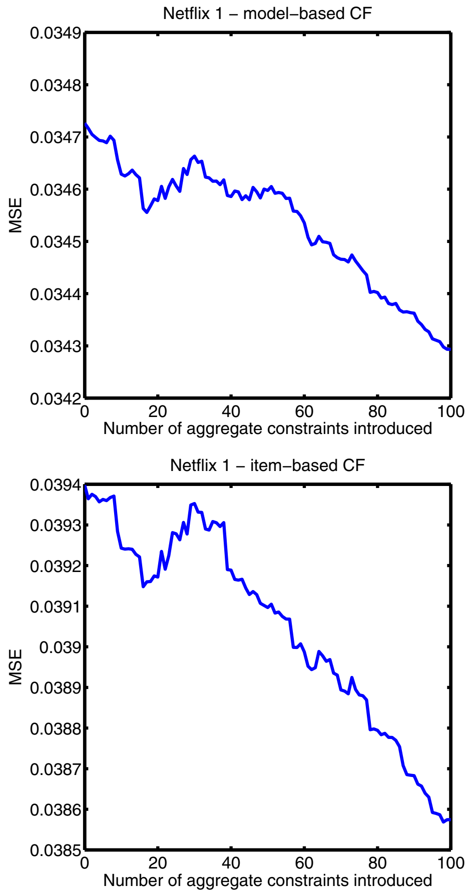

Figure 1. MSE decreases both for model-based CF (top) and item-based CF (bottom) on a Netflix-1 data as more aggregate information is introduced.

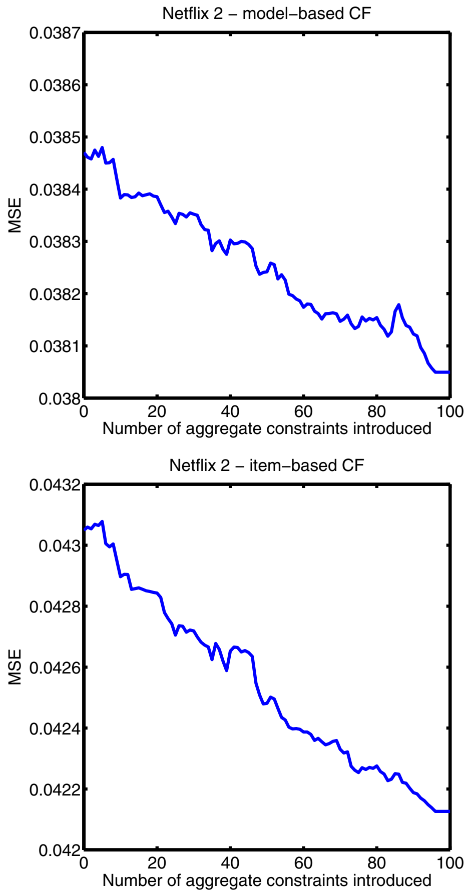

Figure 2. MSE decreases both for model-based CF (top) and item-based CF (bottom) on a Netflix-2 data as more aggregate information is introduced.

ing arbitrarily up and down as additional aggregate ratings are introduced (and plotted along the x-axis). However, it is clear from the graphs that the process has a drift down for all the datasets that we used. The same drift down is observed for other datasets that are not reported in this paper because of the space limitations. There is no single case of the MSE of the constrained estimator with 15-20 aggregate ratings or more being bigger or equal to the MSE of the unconstrained estimator. Therefore, we conclude from these observations that this downward drift would have been unlikely under the assumption that the performance improvement is a random white noise, which is in line with the result of Theorem 1.

6.2 Movie Rating Dataset

The graph in Figure 4 represents the mean squared error (MSE) performance of the model-based CF and item-based CF as a function of the number of additional aggregate ratings introduced for the movies dataset from [1].

As before, since the graph in Figure 4 is not averaged across multiple datasets, there are occasional non-monotonic jumps in it. This happens because adding one aggregate rating to a training sample does not always improve MSE on the test set. For example, the aggregate rating of 6.5 given to movie “Madagascar” may not reflect biases of the particular segment of users who gave the ratings, and this aggregate rating may not fit well with the particular individual ratings given to movie “Madagascar” by the users in our dataset. This explains the big jumps observed in Figure 4 for the model-based CF, where inaccurate aggregate rating information was used for some of the movies (such as movie #25). This happened for this particular movie because all the users who gave good rating for the movie happened to be excluded from the subsample since they provided only few overall ratings.

Furthermore, note that the MSE decreased in Figure 4

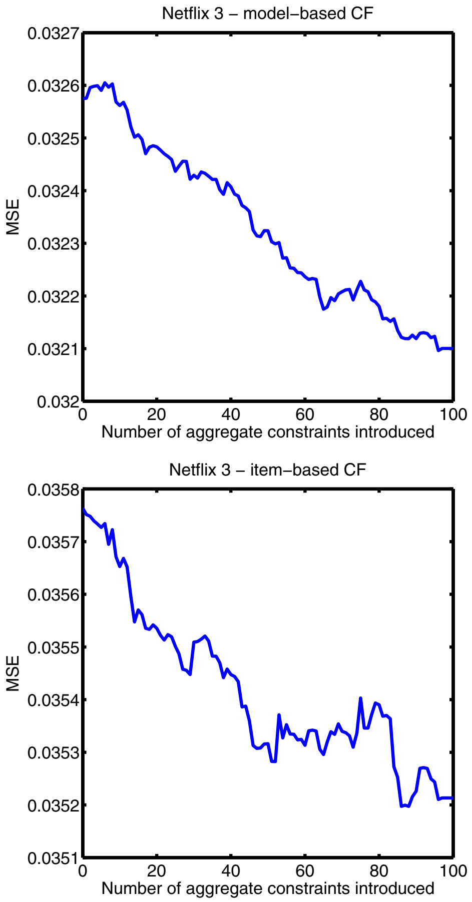

Figure 3. MSE decreases both for model-based CF (top) and item-based CF (bottom) on a Netflix-3 data as more aggregate information is introduced.

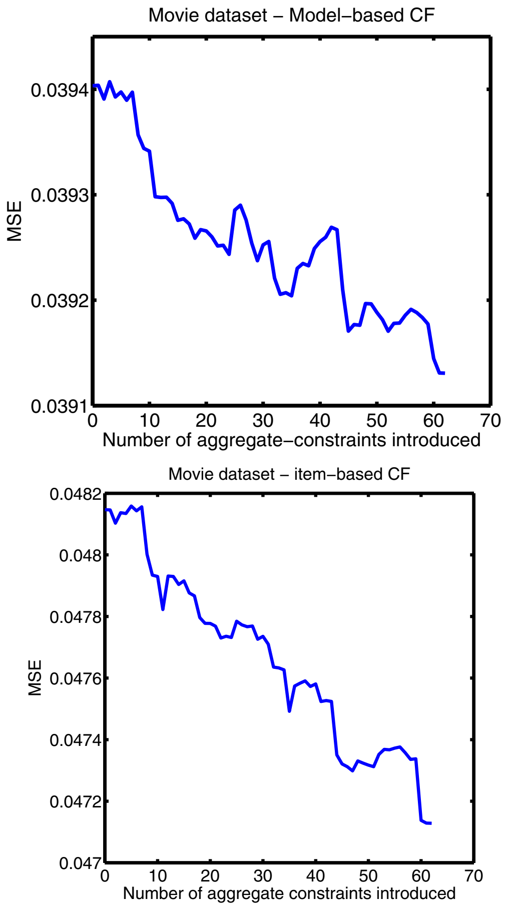

Figure 4. MSE decreases both for model-based CF (top) and item-based CF (bottom) on a movie rating dataset as more aggregate information is introduced.

by about 0.7% from the case of no aggregate to 62 additional aggregate ratings for the model-based CF and by about 2.1% for the item-based CF.

As in the case of the samples from Netflix dataset from Section 6.1, this case constitutes a solid performance improvement, given that they are based on less than 62 additional aggregate ratings. Again, for comparison, the Netflix Prize competitors try to achieve performance improvements of 10% over the Netflix baseline, and the key competitors would have won the $1,000,000 Grand Prize today if they could achieve such performance improvements now.

7 Performance and Scalability

The computational performance of our method was reasonable on the datasets used in our experiments. In particular, the estimation of unknown parameters $\boldsymbol{\mu}$ and complete matrix Σ using the iterative gradient descent algorithm presented in Section 3 and implemented in MATLAB on Intel Xeon CPU 3.73GHz took about 6 hours, depending on the particular dataset used in the experiments. Once the parameters were estimated, the solution of (9) took on the order of 1 second for each of the datasets.

However, the method described so far requires estimation of the $N \times N$ covariance matrix Σ in (1) where N is the number of users. Therefore, it works well with certain optimizations only for small- to medium-size datasets, such as the ones described in Section 5. In this section, we present enhancements to our basic method that make it more scalable and allow it to work on large datasets.

In particular, we propose the following estimation method of unknown ratings. Consider the ratings estimator from (9). If we estimate all the required unknown ratings $\boldsymbol{r}_{Uj}$ *simultaneously* using this equation, then the estimation of the full $N \times N$ covariance matrix Σ is required. However, note that if we could estimate the unobserved ratings *one-by-one* using the same equation (9) and if we have only $|K_j|$ observed ratings for item j, then we would require approximately a $|K_j| \times |K_j|$ submatrix of matrix Σ in (9) to estimate these ratings. More specifically, as it is written in equation (9), we would only need to estimate the corresponding covariance submatrix

$$\mathrm{Var}\begin{pmatrix} r_{uj} \\ \boldsymbol{r}_{Kj} \\ r^a \end{pmatrix} = \begin{pmatrix} \Sigma_{11} & \Sigma_{12} & \Sigma_{13} \\ \Sigma_{21} & \Sigma_{22} & \Sigma_{23} \\ \Sigma_{31} & \Sigma_{32} & \Sigma_{33} \end{pmatrix}$$

instead of the whole matrix Σ, where r_{uj} is the unknown scalar rating for item j that we are attempting to estimate.

If the number of observed ratings $|K_j|$ for item j is small, then this method is clearly better than the original one. For example, if there are 10,000 users and every item has only 5 ratings, then it is clearly much faster to estimate 9,995 times the matrix of size 7×7 than to do it once but on a matrix of size $10,002 \times 10,002$.

Moreover, the external aggregate information is meant to be especially *helpful for items with small amount of known ratings*. The items with very sparse known ratings are the ones targeted by us for prediction improvement, since for heavily rated items the external aggregate rating information is almost revealed in the dataset itself.

Therefore, we suggest the following scalable estimation procedure:

1. Choose items *with small number of given ratings*.

2. For every such item, estimate the matrices Σ_{22}, Σ_{23}, Σ_{32} and Σ_{33} by minimizing (4). For example, for the item with 10 observed ratings, the size of the matrix $\begin{pmatrix} \Sigma_{22} & \Sigma_{23} \\ \Sigma_{32} & \Sigma_{33} \end{pmatrix}$ is 11×11. This estimation only needs to be done *once per item*.

3. Then, for *every rating* that is to be predicted for such items, estimate the vector $\begin{pmatrix} \Sigma_{11} & \Sigma_{12} & \Sigma_{13} \end{pmatrix}$. If the item has only 10 known ratings, then the size of this vector is 12.

4. For *every rating*, use equation (9) to calculate the prediction for the rating based on these estimated parameters.

Assuming that the required parameters of Σ are already estimated, the complexity of the estimator of one rating (9) is $O(|K_j|^3)$, where $|K_j|$ is the number of observed rating for item j. Note that, as explained above, the unknown ratings can be estimated independently from each other. Therefore, the computational time for estimating n unknown ratings is *linear* in the number of unknown ratings n, i.e., is $O(n|K_j|^3)$.

The estimation step involves iterative optimization procedure, which makes it difficult to determine the theoretical complexity of the method. In order to examine scalability on practice, we selected $10,000$ users and $10,000$ items subsample from the Netflix Prize dataset[3]. In Figure 5 we present the average time it takes to complete Step 1 for one arbitrary item as a function of the number of already observed ratings for the item. The Figure 5 clearly demonstrates that, even without special optimization, the MATLAB code solves the problem in reasonable time, since the most time-consuming optimization in Step 2 needs to be done only once for an item. Step 3 of the estimation procedure takes on the order of seconds for estimating one rating for the example provided above.

Speeding up the estimation procedure is one of the directions for the future research. Fortunately, there is a large number of Bayesian estimation methods substituting or complementing the optimization methods with sampling techniques.

We should also note that, for the scalable method, we estimate the matrices Σ_{31}, Σ_{32} and Σ_{33} directly from the data as if we treat the aggregate rating r^a as a rating provided by some pseudo-user and estimate the covariance matrix from the observed vectors $\begin{pmatrix} r_{uj} & \boldsymbol{r}_{Kj} & r^a \end{pmatrix}'$ using the same Bayesian approach as we described above in Equation (4).

Note also that the rating prediction task is *highly parallelizable*. If we have 10,000 items to predict ratings for, we may predict ratings for each item as separate independent tasks. For example, the estimation of parameters for one item with 10 known ratings will take 150 seconds as shown in the Figure 5. However, if we have 10,000 items with 10 known ratings for each item and if we have 100 CPUs, then the estimation of parameters of all items will take $\frac{10,000 \times 150}{100} = 15,000$ seconds, which is about 4 hours.

In addition, the results of running the described methods for several datasets containing $10,000$ users and $10,000$ items are presented in our technical report [21]. Therefore, we conclude from all these discussions that the proposed method is scalable when used on modern computational architectures. We should also note that not only are the items with small number of given ratings the easiest to estimate from the computational point of view, but also that they are the ones targeted for the best improvement from the aggregate information.

8 Conclusions

In this paper we presented an approach for incorporating externally specified aggregate ratings information into certain types of collaborative filtering (CF) methods. This was

[3] This constitutes a considerable amount of the items from complete Netflix Prize dataset [7] that has a total of 17 thousand items.

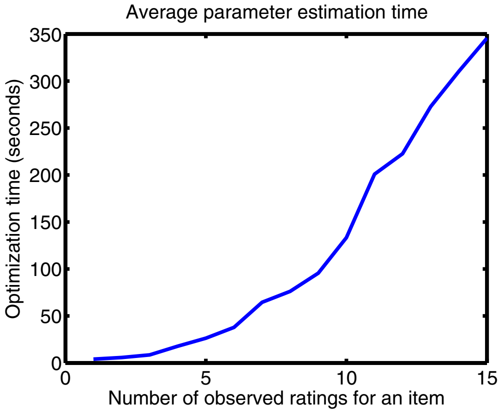

Figure 5. Parameter estimation time for 1 item in $10,000 \times 10,000$ dataset

done by introducing a statistical model of aggregate ratings and incorporating it into the CF methods. For a statistical model-based CF method, we formally showed that this additional aggregated information provides more accurate recommendations of individual items to individual users. Furthermore, theoretical insights gained from the analysis of this model-based method suggested a way to incorporate aggregate information into the heuristic item-based CF method. We empirically tested the model-based and the heuristic item-based CF methods on several datasets; the experiments uniformly confirmed that the aggregate rating information significantly improves CF recommendations.

Among other things, this work shows the power of theory by demonstrating how the insights gained from theoretical developments can shed light on proper selection of good heuristic methods.

We also demonstrated that the method can be scaled to large datasets and is inherently easy to parallelize. Moreover, scalability is achieved easiest for the items that have only few known ratings. As explained in the paper, these items stand the most to benefit from the aggregate information, thus making scalability and sparsity "work" together.

As a future research, we plan to combine the top-down aggregate rating method presented in this paper with the bottom-up method of computing aggregate ratings for the groups of users. The solution to this problem will help us to fill-in the entire OLAP-based hierarchy of aggregate ratings and provide for even better predictions of individual ratings as well as group ratings. We also plan to develop a wider theoretical framework by identifying a class of recommendation methods for which we can formally show that the aggregate ratings can help to provide better recommendations. Another direction for future research constitutes the study of other methods of introducing this type of aggregate information into heuristic-based recommendation models.

References

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

[2] G. Adomavicius and A. Tuzhilin. Multidimensional recommender systems: a data warehousing approach. In *2nd Intl. Workshop on Electronic Commerce. LNCS 2232*, 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. 2005.

[4] D. Agarwal, A. Broder, D. Chakrabarti, D. Diklic, V. Josifovski, and M. Sayyadian. Estimating rates of rare events at multiple resolutions. *Procs of the 13th ACM SIGKDD*, pages 16–25, 2007.

[5] A. Ansari, S. Essegaier, and R. Kohli. Internet Recommendation Systems. *J. of Marketing Research*, 37(3), 2000.

[6] R. Bell, Y. Koren, and C. Volinsky. Modeling relationships at multiple scales to improve accuracy of large recommender systems. *Procs of the 13th ACM SIGKDD*, 2007.

[7] J. Bennett and S. Lanning. The Netflix Prize. *Procs of KDD Cup and Workshop 2007*, 2007.

[8] C. Bishop and N. Nasrabadi. Pattern Recognition and Machine Learning. *J. of Electronic Imaging*, 16:049901, 2007.

[9] J. Bollen. Group user models for personalized hyperlink recommendations. *Procs of the Int. Conference on Adaptive Hypermedia and Adaptive Web-Based Systems*, 2000.

[10] R. Fletcher. *Practical Methods of Optimization, Vol. 1, Unconstrained Optimization*. Wiley-Interscience, 1980.

[11] B. Flury. *A First Course in Multivariate Statistics*. 1997.

[12] A. Gelman. *Bayesian Data Analysis*. CRC Press, 2004.

[13] IMDB. http://www.imdb.com.

[14] A. Jameson and B. Smyth. *Recommendation to Groups*, chapter The Adaptive Web: Methods and strategies of web personalization. Springer, 2006.

[15] G. Linden, B. Smith, and J. York. Amazon. com Recommendations: Item-to-Item Collaborative Filtering. 2003.

[16] M. O'Connor, D. Cosley, J. A. Konstan, and J. Riedl. Polylens: A recommender system for groups of users. *Procs of ECSCW*, 2001.

[17] B. Sarwar, G. Karypis, J. Konstan, and J. Reidl. Item-based collaborative filtering recommendation algorithms. *Procs of the 10th int. conference on World Wide Web*, 2001.

[18] J. Schafer, J. Konstan, and J. Riedl. E-Commerce Recommendation Applications. *DMKD*, 5(1), 2001.

[19] A. Schwaighofer, V. Tresp, and K. Yu. Learning Gaussian Process Kernels via Hierarchical Bayes. *Advances in Neural Information Processing Systems*, 17:1209–1216.

[20] A. Umyarov and A. Tuzhilin. Leveraging aggregate ratings for better recommendations. *Procs of the 2007 ACM conference on Recommender systems*, pages 161–164, 2007.

[21] A. Umyarov and A. Tuzhilin. Improving predictive performance of recommender systems using aggregate ratings. Working paper. Stern School of Business. New York University. CeDER-08-03, October, 2008.