Leveraging aggregate ratings for improving predictive performance of recommender systems. Working paper. CeDER-08-03.

AKHMED UMYAROV
New York University
ALEXANDER TUZHILIN
New York University

This paper describes an approach for incorporating externally specified aggregate ratings information into certain types of recommender systems, including two types of collaborating filtering and a hierarchical linear regression model. First, we present a framework for incorporating aggregate rating information and apply this framework to the aforementioned individual rating models. Then we formally show that this additional aggregate rating information provides more accurate recommendations of individual items to individual users. Further, we experimentally confirm this theoretical finding by demonstrating on several datasets that the aggregate rating information indeed leads to better predictions of unknown ratings. We also propose scalable methods for incorporating this aggregate information and test our approaches on large datasets. Finally, we demonstrate that the aggregate rating information can also be used as a solution to the cold start problem of recommender systems.

Categories and Subject Descriptors: H.1.2 [**Information Systems**]: Models and Principles—*User/Machine Systems, Human information processing*; H.3.3 [**Information Systems**]: Information Storage and Retrieval—*Information Search and Retrieval, Information filtering*

Additional Key Words and Phrases: Recommender systems, collaborative filtering, hierarchical linear models, predictive models, aggregate ratings, cold-start problem

1. INTRODUCTION

Consider a Netflix recommender system [Bennett and Lanning 2007] and assume that it is augmented with the aggregate ratings from the IMDB database [IMDB 2006], 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 determining individual ratings for individual users can be supplemented with an externally provided OLAP-based [Adomavicius et al. 2005] set 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, that are specified for various cells

...

Permission to make digital/hard copy of all or part of this material without fee for personal or classroom use provided that the copies are not made or distributed for profit or commercial advantage, the ACM copyright/server notice, the title of the publication, and its date appear, and notice is given that copying is by permission of the ACM, Inc. To copy otherwise, to republish, to post on servers, or to redistribute to lists requires prior specific permission and/or a fee.
© 2009 ACM 1529-3785/2009/0700-0001 $5.00

of the OLAP-based hierarchy.

The main research question addressed in this paper is how these *external aggregate* ratings can be used for providing better recommendations of *individual* items to *individual* users. Since the answer to this question depends critically on the type of a recommender system used, we first need to select a particular recommender system and then augment it with aggregate ratings. In this paper, we consider three different types of recommender system models, two of them based on collaborative filtering and the third belonging to a class of hierarchical linear regression models (HLM) widely used in statistics and marketing [Raudenbush and Bryk 2001]. For each of these recommendation models we propose a method of incorporating the external aggregate rating information into the overall recommendation process and show, both theoretically and experimentally, that this additional information indeed helps to improve estimations of unknown individual ratings.

The proposed approach is important for the following reasons. First, it shows that the externally provided aggregate rating data does matter and indeed leads to improved recommendations. Since such data is often available or can be obtained in practical business settings, such as in the IMDB and Netflix example, the proposed method is useful in many "real-life" applications. Second, being aggregated, such data is often publicly available, as in the case of IMDB, and does not carry privacy implications. Therefore, it can be *freely* and *widely* used in many recommender systems. Finally, we also show in the paper that this aggregate rating information is especially useful in those cases when items have only few ratings, i.e., they belong to the Long Tail of recommender systems [Park and Tuzhilin 2008]. More specifically, we demonstrate that the aggregate ratings information better improves rating estimations of the items having only few than those having many ratings. Therefore, our proposed method can be considered as one of the possible solutions to the *cold start* problem [Schein et al. 2002]. Intuitively, this makes sense since our results show that the *aggregate* rating information should be relied upon more heavily when estimating unknown ratings of the items having only few *individual* ratings.

In all this work, we make one fundamental assumption that the external aggregate ratings are generated from the unknown individual ratings that have the *same distribution* as the individual internal ratings. For example, in the IMDB/Netflix example, we assume that the females from the IMDB database in the age category of 18 to 29 who saw the movie "Madagascar" have the same statistical properties as the same generation of females who saw "Madagascar" in the Netflix database (and therefore, the Netflix population would also give an average rating of 6.9 (out of 10), as in the IMDB case). Although not always the case, nevertheless this assumption holds well for the three "real-life" datasets used in our studies and, therefore, the experiments confirm our theoretical result that the aggregate external information improves rating estimations.

Although the proposed methods theoretically improve predictions of unknown ratings, in practice some of them work quite slowly on large datasets. Therefore, one of the contributions of the paper lies in developing more advanced algorithms that scale well to larger datasets. We present such advanced methods in the paper and show how well they work on such datasets.

In summary, we made the following contributions in this paper:

(1) presented a framework for incorporating aggregate information into various recommender system models

(2) showed for these models how to apply the aggregate rating information to estimating individual ratings for individual users

(3) theoretically and experimentally demonstrated that the aggregate ratings information indeed helps to provide better recommendations for these models

(4) showed how to scale the proposed methods to larger datasets

(5) proposed our methods as a solution to the cold-start problem and demonstrated that our results indeed support our claim.

The rest of the paper is organized as follows. In Section 2, we present the related work. In Section 3, we state the general problem of incorporating aggregate rating information into individual-rating-based recommender systems and present the Aggregate Rating Recommendation Model (ARRM). In Section 4.1, we describe a model-based approach to collaborative filtering, present a way to introduce aggregate information into the model and describe how to make the estimation procedure scalable. In Section 4.2 we use the theoretical insights obtained from the model-based collaborative filtering approach presented in Section 4.1 in order to introduce the aggregate information into the classical heuristic item-based collaborative filtering. In Section 5, we describe how to add aggregate ratings into a recommendation system based on the hierarchical linear regression model from [Ansari et al. 2000] and present a scalable solution to estimating the parameters of the model. In Section 6, we prove theoretically that the aggregate information indeed improves the average MSE performance of the aforementioned models. In Section 7 and 8, we show empirically on several datasets that the significant improvement is achieved for all the presented models when the aggregate information is introduced. In Section 9, we describe how the external aggregate rating information can be used for solving the cold-start problem. In Section 10, we summarize our results and describe some directions for future research.

2. RELATED WORK

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 [Adomavicius and Tuzhilin 2001]. This approach was subsequently extended and explored further in [Adomavicius et al. 2005]. Also, [O'Connor et al. 2001] presents a method for providing recommendations to a group of users. [Jameson and Smyth 2006] discusses the new issues that arise when one considers web-based personalization involving groups for a certain subclass of group recommender systems. [McCarthy et al. 2006] report a recommender system that uses an iterative process to achieve good group recommendations from individual recommendations by collecting individual user's critiques. These methods concentrate on the bottom-up approach to recommendations that use aggregate ratings as a basis for recommendations to groups of users. In contrast to this, [Bollen 2000] 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. [Bell et al. 2007] 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, [Bell et al. 2007] 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. [Agarwal et al. 2007] 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. This paper differs from all this prior work in that it presents a general framework for incorporating the aggregate rating information to improve estimations of individual unknown ratings in a top-down fashion, considers three specific recommendation models and shows, both theoretically and experimentally, that the proposed aggregate rating methods indeed improve individual recommendations for these models.

Some preliminary work described in this article is reported in two conference papers [Umyarov and Tuzhilin 2007] and [Umyarov and Tuzhilin 2008] previously written by the authors. In particular, [Umyarov and Tuzhilin 2007] presents the basic idea of incorporating aggregate rating information into the statistical model of a recommender system described in [Ansari et al. 2000]. Furthermore, [Umyarov and Tuzhilin 2007] theoretically demonstrates that these incorporated aggregate ratings indeed provide for better estimation of unknown ratings. However, [Umyarov and Tuzhilin 2007] focused only on the recommender system from [Ansari et al. 2000] (no collaborative filtering models were studied), the proposed method was highly unscalable, and no empirical validation of the aforementioned theoretical result was reported in that paper.

In contrast, [Umyarov and Tuzhilin 2008] studied how to incorporate aggregate rating information into the collaborative filtering models, reported experimental results on small datasets and suggested the ways to make the method more scalable. However, [Umyarov and Tuzhilin 2008] studied scalability only on a theoretical level and did not experimentally test the considered solutions on large datasets. Also, [Umyarov and Tuzhilin 2008] did not study how to use the aggregate rating information as a solution to the cold-start problem, and did not compare the influence of aggregate information on items with different density of observed ratings.

3. AGGREGATE RATING RECOMMENDATION MODEL (ARRM)

The idea of using aggregate ratings to improve estimations of individual ratings can be operationalized by formulating a class of models containing the following two components:

—*Individual-Rating Model:* a basic recommendation model estimating individual ratings, such as unknown ratings of individual movies provided by individual users and

—*Aggregate-Rating Model:* provides a way to introduce externally observed aggregate ratings into the Individual-Rating Model, where rating aggregation is done

over *sets* of users and items. For example, knowledge of an average rating of 7.7 (out of 10) from the IMDB database [IMDB 2006] given by males in the age category of 30 to 44 for the movie "Last King of Scotland" can be used to improve predictions of individual movie ratings in the Netflix database using the Individual-Rating Model described in the previous point. Although we refer to this approach as an Aggregate-Rating Model, we should note that the Aggregate Rating Model is *tightly coupled* to the Individual-Rating Model, as explained below. More specifically, we demonstrate below that each aggregate rating can be interpreted as a certain type of a constraint that we call an *aggregate constraint*.

Note that we may assume a whole range of different types of Individual-Rating Models. For example, an Individual-Rating Model can be a classical collaborative filtering [Sarwar et al. 2001], or a hierarchical linear regression model [Raudenbush and Bryk 2001], or any other method estimating an unknown individual rating r_{ij} given by user i for item j. When we assumed a particular Individual-Rating Model, the Aggregate-Rating Model is required to be consistent with the underlying Individual-Rating model. For example, if individual ratings are assumed to be from a multivariate normal distribution, then this implies that the true average aggregate rating also has a normal distribution.

In practice however, we do not observe the true average rating from external sources for various reasons ranging from aggregation over small sample to difference in populations that the aggregate ratings and individual ratings come from. In order to accommodate this issue, we need to also specify an Aggregate-Rating Model that incorporates them in one way or another. For instance, following the previous example, we may assume that we observe not the true average rating, but the true average rating with *some additive noise*.

When both the Individual- and the Aggregate-Rating models are specified, the combined model is called the *Aggregate Recommendation Ratings Model (ARRM)*. In the next three sections, we present three different types of such ARRM models to illustrate how aggregate ratings can be used to improve individual recommendations. We then show, both theoretically and empirically, that the aggregate rating information indeed improves individual recommendations for these models.

4. COLLABORATIVE FILTERING

First, we study how the aggregate ratings information can be used in the *collaborative filtering (CF)* systems that constitute the "bread-and-butter" of recommender systems. In particular, we use the standard item-based collaborative filtering [Sarwar et al. 2001] and show how aggregate information can help in predicting unknown ratings. However, since the item-based method constitutes a heuristic-based approach and we would like to explore the problem not only empirically but also theoretically, we first study how aggregate ratings information improves *model-based* CF methods. To this extent, we first present a model-based CF method from [Schwaighofer et al. 2004] in Section 4.1, that is grounded in the fundamentals of statistical theory, and analyze it theoretically. Then in Section 4.2, we describe a related approach for the item-based case and also show how the insights from the theoretical model-based method are applied to handle the item-based approach.

4.1 Model-based Collaborative Filtering

Following the ideas from Section 3, we first describe the Individual-Rating model in Section 4.1.1 and then the Aggregate-Rating model in Section 4.1.2.

4.1.1 *Individual-Rating Model.* In this section, we follow [Schwaighofer et al. 2004] when describing a model-based approach. 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[1] $\boldsymbol{r}_j = (r_{1j}, r_{2j}, \ldots, r_{Nj})'$ 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:

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

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 [Bishop and Nasrabadi 2007], 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:

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

As it is shown in [Flury 1997], the conditional expected value has the following form:

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

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.

4.1.2 *Aggregate-Ratings Model.* In this section we describe a method of adding aggregate information to the unconstrained (URE) collaborative filtering model presented in Section 4.1.1. 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

[1] We typed vectors in bold font as opposed to matrices and scalars that are typed in regular font.

particular item j.[2] In particular, assume that:

$$r^a = \frac{1}{N}\sum_{i=1}^{N} r_{ij} = a_j = \alpha_j + \varepsilon_j, \quad \varepsilon \sim N(0, \sigma_j^2) \tag{4}$$

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 [Bennett and Lanning 2007] to predict the rating of "Madagascar" for a particular user and we also know from IMDB [IMDB 2006] 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, as mentioned in Section 3, the external aggregate ratings *may come from a sample that is different from the sample of the given individual ratings.* 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:

$$\begin{pmatrix} \boldsymbol{r}_{Uj} \\ \boldsymbol{r}_{Kj} \\ r^a \end{pmatrix} \sim N\left(\begin{pmatrix} \boldsymbol{\mu}_1 \\ \boldsymbol{\mu}_2 \\ \mu^a \end{pmatrix}, \begin{pmatrix} \Sigma_{11} & \Sigma_{12} & \Sigma_{13} \\ \Sigma_{21} & \Sigma_{22} & \Sigma_{23} \\ \Sigma_{31} & \Sigma_{32} & \Sigma_{33} \end{pmatrix} \right) \tag{5}$$

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

$$\operatorname{cov}(r_{ij}, r^a) = \operatorname{cov}\left(r_{ij}, \frac{1}{N}\sum_{k=1}^{N} r_{kj} + \varepsilon \right) = \frac{1}{N}\sum_{k=1}^{N} \operatorname{cov}\left(r_{ij}, r_{kj}\right) \tag{6}$$

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

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

$$= \frac{1}{N^2}\sum_{i=1}^{N}\sum_{k=1}^{N} \operatorname{cov}\left(r_{ij}, r_{kj}\right) + \sigma^2 \tag{7}$$

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.

[2]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.

Therefore, following the ideas from [Flury 1997], as in the case of equation (3), the least mean squared error unbiased estimator that takes into account the observed aggregate information (4) is

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

$$+ \begin{pmatrix} \Sigma_{12} & \Sigma_{13} \end{pmatrix} \begin{pmatrix} \Sigma_{22} & \Sigma_{23} \\ \Sigma_{32} & \Sigma_{33} \end{pmatrix}^{-1} \begin{pmatrix} \boldsymbol{y} - \boldsymbol{\mu}_2 \\ a - \mu^a \end{pmatrix} \tag{8}$$

We call the estimator (8) the *constrained rating estimator (CRE)*, since the estimator is conditional not only on the observed ratings $\boldsymbol{r}_{Kj}$, but also on the additional constraint of the type (4). Therefore, we call equation (4) an *aggregate constraint*.

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

$$\hat{r}^*_{ij} = \hat{r}_{ij} + T_{ij} \tag{9}$$

where $\hat{r}_{ij}$ is the unconstrained estimator from (3) and $\hat{r}^*_{ij}$ is the constrained estimator from (8). Subtracting (3) from (8), we get the following expression for the vector of correction terms $\boldsymbol{T}_{Uj}$:

$$\boldsymbol{T}_{Uj} = \begin{pmatrix} \Sigma_{12} & \Sigma_{13} \end{pmatrix} \times \left[\begin{pmatrix} \Sigma_{22} & \Sigma_{23} \\ \Sigma_{32} & \Sigma_{33} \end{pmatrix}^{-1} - \begin{pmatrix} \Sigma_{22}^{-1} & 0 \\ 0 & 0 \end{pmatrix} \right] \begin{pmatrix} \boldsymbol{y} - \boldsymbol{\mu}_2 \\ a - \mu^a \end{pmatrix} \tag{10}$$

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 6.1 shows below, adding the correction term to the URE estimator (3) should improve the performance of the individual model. Moreover, as we show in Section 8, this result holds not only in theory, but it is also confirmed on the real-life rating data.

Also note that the case of observing multiple aggregate ratings $r^{a1},\ldots,r^{al}$ for a particular item j is a simple generalization from the described case of observing just a single rating r^a for item j.

4.1.3 *Basic Solution.* 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 model (1) are unobserved as well. Following [Schwaighofer et al. 2004] and [Gelman 2004], 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 [Gelman 2004] 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 hyperparameter ν_0 describes the degrees of freedom and the matrix Λ_0 describes the scale of inverse-Wishart distribution. The vector hyperparameter $\boldsymbol{\mu}_0$ 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 $P(\boldsymbol{\mu}, \Sigma|$ 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 [Schwaighofer et al. 2004], 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 [Gelman 2004] 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):

$$-\log P(\boldsymbol{\mu}, \Sigma| \text{ observed}\{r_{kl}\}) = \left(\frac{\nu_0 + N}{2} + 1\right) \log|\Sigma| + \tag{11}$$

$$+\frac{1}{2}\text{tr}\left(\Lambda_0 \Sigma^{-1}\right) + \frac{k_0}{2}(\boldsymbol{\mu} - \boldsymbol{\mu}_0)'\Sigma^{-1}(\boldsymbol{\mu} - \boldsymbol{\mu}_0) +$$

$$+\sum_{j=1}^{M} \frac{(\boldsymbol{r}_{Ki} - \boldsymbol{\mu}_K)'\Sigma_K^{-1}(\boldsymbol{r}_{Kj} - \boldsymbol{\mu}_K)}{2} + \frac{1}{2}\sum_{j=1}^{M} \log|\Sigma_K|$$

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 (11) 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 (11) 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 (11) 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 (11) w.r.t. parameter Σ is

$$\frac{\partial(-\log P)}{\partial \Sigma_{ij}} = \left(\frac{\nu_0 + N}{2} + 1\right) \mathrm{tr}\left[\Sigma^{-1}\frac{\partial \Sigma}{\partial \Sigma_{ij}}\right] + \frac{1}{2}\mathrm{tr}\left[-\Lambda_0 \Sigma^{-1}\frac{\partial \Sigma}{\partial \Sigma_{ij}}\Sigma^{-1}\right] -$$

$$-\frac{k_0}{2}(\boldsymbol{\mu}-\boldsymbol{\mu}_0)'\Sigma^{-1}\frac{\partial \Sigma}{\partial \Sigma_{ij}}\Sigma^{-1}(\boldsymbol{\mu}-\boldsymbol{\mu}_0) - \sum_{j=1}^{M}\frac{1}{2}\left[(\boldsymbol{r}_{Ki} - \boldsymbol{\mu}_K)'\,(L_K'\Sigma L_K)^{-1}L_K'\frac{\partial \Sigma}{\partial \Sigma_{ij}}L_K \times\right.$$

$$\left.\times (L_K'\Sigma L_K)^{-1}(\boldsymbol{r}_{Kj} - \boldsymbol{\mu}_K)\right] + \frac{1}{2}\sum_{j=1}^{M}\mathrm{tr}\left[(L_K'\Sigma L_K)^{-1}L_K\frac{\partial \Sigma}{\partial \Sigma_{ij}}L_K\right]$$

Second, after defining this gradient, we estimate parameters $\boldsymbol{\mu}$ and Σ using the line search gradient descent procedure [Fletcher 1980] 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.

In summary, the basic solution to the ARRM rating estimation problem can be described as the following procedure:

(1) Estimate the complete parameters $\hat{\boldsymbol{\mu}}$ and $\hat{\Sigma}$ by minimizing the objective function described in (11).

(2) Once the parameters are known, estimate unknown ratings using equation (8).

Although the computational performance of the basic solution reported in Section 8 for a particular smaller dataset was reasonable, more scalable methods are needed for larger datasets. The method described so far requires estimation of the $N \times N$ covariance matrix Σ in (1) where N is the number of users. It works well with certain optimizations only for small- to medium-size datasets, such as the Movie Rating dataset from [Adomavicius et al. 2005] as described in Section 7. To address the scalability question, we present an alternative scalable solution in the next subsection.

4.1.4 *Scalable Solution.* In this section, we present enhancements to our basic solution that make it more scalable and allow it to work on larger datasets. In particular, we propose the following estimation method of unknown ratings. Consider the ratings estimator from (8). 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 (8) 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 (8) to estimate these ratings. More specifically, as it is written in equation (8), we would only need to estimate the corresponding covariance subma-

trix

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

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, as we have already mentioned, not only are the items with small number of given ratings the easiest to estimate from the computational point of view, but also the external aggregate information is meant to be especially *helpful for them*. 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. Indeed, the items with fewer ratings achieve better predictive performance improvements when aggregate rating information is applied to them, as is shown in Section 9 describing our solution to the Long Tail problem.

Therefore, we propose the following scalable estimation procedure:

(1) For every *item* with a *small* number of given ratings (as will be shown in Section 8, computational complexity is reasonable for the items with a "small" number of ratings, as reported in Figure 6) we estimate the vector of parameters $\begin{pmatrix} \boldsymbol{\mu}_2' & \mu_3 \end{pmatrix}$ and the matrices Σ_{22}, Σ_{23}, Σ_{32} and Σ_{33} by minimizing the objective function described in (11).
(2) For every *rating* for such items, we estimate vector $\begin{pmatrix} \Sigma_{11} & \Sigma_{12} & \Sigma_{13} \end{pmatrix}$ by minimizing the objective function described in (11).
(3) We use equations (8) to calculate the prediction for the rating based on the estimated parameters.

We should note that, for this method, we estimate matrices Σ_{31}, Σ_{32} and Σ_{33} directly from the data. More specifically, 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 (11).

Computational complexity of the estimation procedure. Note that, although the dimension of the parameter space for this optimization procedure in Step 1 grows quadratically with the number of observed ratings $|K_j|$ for item j, we need to do it *only once per item*. 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. Once estimated, these parameters are used *unchanged* for computing every unknown rating for that particular item, and this makes our approach scalable for large datasets, as will be experimentally shown in Section 8.

Furthermore, note that, although Step 2 in the above procedure is performed for *every rating*, the optimization procedure is performed over a number of parameters that is linear in $|K_j|$. For example, if an item has 10 known ratings, then the size of vector $\begin{pmatrix} \Sigma_{11} & \Sigma_{12} & \Sigma_{13} \end{pmatrix}$ is 12. Therefore, these parameters can be estimated efficiently.

Finally, after the required parameters of Σ are estimated in Steps 1 and 2, the complexity of estimating a single rating in Step 3 using equation (8) is $O(|K_j|^3)$, where $|K_j|$ is the number of observed rating for item j. Further, note that 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 $O(n|K_j|^3)$.

Note that it is hard to determine the exact computational complexity of the described estimation procedure because Steps 1 and 2 involve iterative optimization method converging to a local optimum. Therefore, we tested experimentally the performance of our procedure and report the results in Section 8.

In conclusion, all the three steps in the estimation procedure are scalable for the items with a small number of ratings. Further, this procedure is also *highly parallelizable* because Steps 1, 2 and 3 can be performed independently and in parallel for *each item*.

In summary, in this section we presented the model-based approach to CF and showed that the external aggregate information can be incorporated into the individual rating model by introducing the correction term T_{ij} in equation (9). In Section 4.2, we apply this correction term approach when we incorporate the aggregate rating information into the classical item-based collaborative filtering.

4.2 Item-based Collaborative Filtering

We follow the standard approach of [Sarwar et al. 2001] 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 4.1.2.

Item-based collaborative filtering is one of the most popular recommendations techniques used widely in industry [Schafer et al. 2001] including such companies as Amazon [Linden et al. 2003]. 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 corre-

lation coefficient:

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

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 according to [Adomavicius and Tuzhilin 2005], unlike the model-based URE and CRE estimators described in Sections 4.1.1 and 4.1.2. Therefore, formal statistical analysis to improving rating estimations presented in Section 4.1.2 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 4.1.2 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 (9) 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

$$\hat{r}^*_{ij} = \frac{\sum_{k \in I(i)} s_{jk} r_{ik}}{\sum_{k \in I(i)} s_{jk}} + T_{ij} \tag{12}$$

where T_{ij} is calculated as in (10). In Section 8, we empirically show that this new rating estimator indeed improves performance of the item-based CF.

In summary, we propose the following estimation method for predicting unknown rating r_{ij} for the item-based CF:

(1) Estimate the individual rating $\hat{r}_{ij}$ from item-based CF model.
(2) Estimate the correction term T_{ij}.
(3) Apply the correction term to the individual rating.

The scalability of the proposed method consists of two parts: 1) scalability of item-based CF itself, 2) scalability of computation of correction term T_{ij}. As we demonstrated for the model-based CF in Section 4.1.2 the correction term T_{ij} can be calculated in a scalable manner.

After presenting the CF-based Aggregate-Rating models in this section, we next present alternative Hierarchical Linear models.

5. HIERARCHICAL LINEAR MODEL

We have selected an instance of a Hierarchical Linear Regression Model (HLM) for the Individual-Rating Model for the following reasons. First, HLM is a popular type of a statistical model used by statisticians in modeling many "real-life" phenomena [Hox 2002]. Second, it is also grounded in a sound statistical theory and has several nice statistical properties [Raudenbush and Bryk 2001]. Therefore, an HLM model can not only be studied experimentally, as the item-based CF model,

but can also be analyzed theoretically, as is done in this paper. Third, HLMs have already been independently proposed for modeling recommender systems both by statisticians [Condliff et al. 1999] and marketers [Ansari et al. 2000]. Moreover, it was demonstrated in [Ansari et al. 2000] that this type of the recommender system outperformed the classical collaborative filtering model [Ansari et al. 2000].

In the next subsection, we present the basic Individual-Rating and in Section 5.2 the Aggregate-Rating Model.

5.1 The Individual-Rating Model

Another approach to Individual-Ratings Model that we consider in this paper is a frequentist probabilistic version of the Bayesian recommendation models described in [Ansari et al. 2000] and [Condliff et al. 1999]. Although less known in the field of recommender systems and more computationally complex, these models constitute hybrid recommender systems that use both item and user attributes to make recommendations, and as [Ansari et al. 2000] demonstrated, these models tend to outperform classical CF. We decided to use the frequentist rather than the Bayesian approach in this paper because of the scalability and performance issues.

Assume there are N_U users, where each user i is defined by $\boldsymbol{z}_i$ of attributes of user i, such as age, gender, income etc. Also assume that there are N_I items, where each item j is defined by vector $\boldsymbol{w}_j$ of attributes[3] of the item, such as price, weight, size etc.

Let r_{ij} be a rating assigned to item j by user i, where r_{ij} is a real-valued number. Moreover, ratings r_{ij} are only known for some subset of all possible (user, item) pairs. Assume we also observe a vector of user attributes $\boldsymbol{z}_i$ for each user i, a vector of item attributes $\boldsymbol{w}_j$ for each item j, a vector $\boldsymbol{x}_{ij} = \boldsymbol{z}_i \otimes \boldsymbol{w}_j$, where $\otimes$ is the Kronecker product. Algebraically, $\boldsymbol{x}_{ij}$ is a long vector containing all possible cross-products between individual elements of $\boldsymbol{z}_i$ and $\boldsymbol{w}_j$.

Then the rating generation model is defined by [Ansari et al. 2000] as

$$r_{ij} = \boldsymbol{x}'_{ij}\boldsymbol{\mu} + \boldsymbol{z}'_i\boldsymbol{\gamma}_j + \boldsymbol{w}'_j\boldsymbol{\lambda}_i + \varepsilon_{ij}, \tag{13}$$

$$\begin{cases} E\left[\varepsilon_{ij}\right] = 0, & \mathrm{Var}\left[\varepsilon_{ij}\right] = \sigma^2, \quad \forall i, j \\ E\left[\boldsymbol{\gamma}_j\right] = \mathbf{0}, & \mathrm{Var}\left[\boldsymbol{\gamma}_j\right] = \Gamma, \quad \forall j \\ E\left[\boldsymbol{\lambda}_i\right] = \mathbf{0}, & \mathrm{Var}\left[\boldsymbol{\lambda}_i\right] = \Lambda, \quad \forall i \end{cases} \tag{14}$$

where vector $\boldsymbol{\mu}$, set of vectors $\{\boldsymbol{\gamma}_j\}$, set of vectors $\{\boldsymbol{\lambda}_i\}$, matrices Λ and Γ, and a scalar σ constitute unobserved parameters of the model. The detailed derivation of the functional form for this model from the simple underlying assumptions is presented in Appendix B.

Intuitively, equation (13) presents a regression model specifying unknown ratings r_{ij} in terms of the characteristics $\boldsymbol{z}_i$ of user i, the characteristics $\boldsymbol{w}_j$ of item j and the interaction effects $\boldsymbol{x}_{ij}$ between them. Interaction effects arise from the hierarchical structure of the model and are intended to capture effects such as how the age of a user changes his or her preferences for certain genres of movies.

Vector $\boldsymbol{\mu}$ in (13) represents unobserved (unknown), slope of the regression line. Intuitively, each element of $\boldsymbol{\mu}$ here represents a general "effect" of some user char-

[3]We also include constant term both in $\boldsymbol{z}_i$ as a user attribute and in $\boldsymbol{w}_j$ as an item attribute.

acteristic on his or her "appreciation" of some item characteristic. For example, if l-th item characteristic is its release year, k-th user characteristic is user age and size of vector $\boldsymbol{z}_i$ is $|\boldsymbol{z}|$. Then the element $\boldsymbol{\mu}_{(l-1)|\boldsymbol{z}|+k}$ can be interpreted as the general effect of user age on his or her attitude towards item release year. This interpretation is very similar to the interpretation of linear regressions with included *interaction terms* that are widely used in social research.

Vector $\boldsymbol{\gamma}_j$ represents weight coefficients *specific* to item j that determine idiosyncracy of item j, i.e., the unobserved heterogeneity of item j. Similarly, vector $\boldsymbol{\lambda}_i$ represents weight coefficients *specific* to user i that determine idiosyncracy of that user, i.e., the unobserved heterogeneity of user i. In (14) we make assumptions about the moments of distributions for these heterogeneities. More specifically, we assume they are independent from each other and have the same covariance matrix, i.e.,

$$\begin{cases} \mathrm{Var}\left[\boldsymbol{\lambda}_i\right] = \Lambda & \forall i \in [1, N_U] \\ \mathrm{Var}\left[\boldsymbol{\gamma}_j\right] = \Gamma & \forall j \in [1, N_I] \end{cases}$$

where Γ and Λ are unobserved covariance matrices.

We would like to stress that the parameters $\boldsymbol{\gamma}_j$ and $\boldsymbol{\lambda}_i$ of the regression model (13) are *unique* for each individual item and each individual user respectively, thus making rating estimations r_{ij} in (13) targeted to particular items and users. This implies, among other things, that *even if two items have exactly same attributes, the model will produce different ratings for them even for the same user,* because each item has its own item-specific individual heterogeneity vector $\boldsymbol{\gamma}_j$, which is estimated based on each item's own historical ratings.

We also assume that each observation r_{ij} has independent disturbances ε_{ij} with the same variance, that is,

$$\mathrm{Var}\left[\varepsilon_{ij}\right] = \sigma^2, \quad \forall i, j$$

where σ is also an unobserved parameter.

5.2 Aggregate-Ratings Model

In addition to the individual ratings r_{ij} modelled by (13), we assume that we also know several aggregate ratings provided *externally* for the model. We model these aggregate ratings as expected values[4] of an average rating across some segment S of user-item pairs. That is, if there are k total possible user-item pairs in the segment S, then we also observe the value a such as

$$E_\varepsilon \left[\frac{\sum_{i,j} r_{ij}}{k} \right] = a, \tag{15}$$

where the sum is taken over *all* the user-item pairs $(i, j) \in S$, and r_{ij} are all the possible observed and unobserved ratings in segment S. For example, assume that external statistical studies have shown that the expected average rating of some 100 action movies provided by 20 graduate CS students is $a = 7.8$ based on $k = 2000$ possible ratings of these movies by these users.

[4]Here we take expected value only over ε, not $\boldsymbol{\gamma}_j$ and $\boldsymbol{\lambda}_i$

Moreover, we may not be exactly sure about this aggregate information. Thus, we assume that the aggregate ratings are "noisy," which can be formally represented as:

$$\begin{cases} E_\varepsilon \left[\frac{\sum_{i,j} r_{ij}}{k} \right] = \alpha, \\ a = \alpha + \xi, \quad E\xi = 0,\ \mathrm{Var}(\xi) = \sigma_\xi^2, \end{cases} \tag{16}$$

where ξ is an unknown noise component, α is an unknown true value for the aggregate rating, a is the observed noisy value for the aggregate rating and σ_ξ^2 is a known parameter. For instance, we may assume in the previous example that the observed expected average rating of 100 action movies provided by 20 graduate CS students is a random variable with mean $a = 7.8$ and the standard deviation of $\sigma_\xi = 0.1$.

Note that in practice, as mentioned in Section 3, the external aggregate ratings may not be known exactly or may come from a sample that is different from the sample of the given individual ratings. However, *the noise term ξ 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 summary, the overall Aggregate Rating Recommendation Model model specifies individual ratings r_{ij} using (13) and (14) and the aggregate ratings modeled as (16). Note that as in the case of model-based CF, each observed aggregate rating can be interpreted as an additional constraint on the parameters of the model that we call an *aggregate constraint*. Then our task is to estimate parameters of this model, given the data.

Given the model (13) with additional aggregate information of type (16), our problem is to find the feasible way to estimate unknown parameters of the model: vectors $\boldsymbol{\mu}$, $\{\boldsymbol{\gamma}_j\}$, $\{\boldsymbol{\lambda}_i\}$, matrices Λ, Γ and the scalar σ.

In Section 5.3 we present the "natural" statistical approach, previously reported in our preliminary short paper [Umyarov and Tuzhilin 2007][5]. Unfortunately, this "natural" method itself does not scale well to the medium- and large-size problems, as is demonstrated in Section 5.3. To address this problem, we present a new and a more scalable method in Section 5.4.

5.3 Basic Solution

The "natural" solution of the model (13) and (16) presented in Section 5.2 is based on the straightforward use of the Generalized Least Squares (GLS) estimator [Greene 2002]. To describe it, we first introduce the notion of a *compound disturbance* η_{ij} by grouping together all the random effects in (13) as follows

$$r_{ij} = \boldsymbol{x}'_{ij}\boldsymbol{\mu} + \underbrace{\boldsymbol{z}'_i\boldsymbol{\gamma}_j + \boldsymbol{w}'_j\boldsymbol{\lambda}_i + \varepsilon_{ij}}_{\eta_{ij}}, \tag{17}$$

where we define $\eta_{ij} = \boldsymbol{z}'_i\boldsymbol{\gamma}_j + \boldsymbol{w}'_j\boldsymbol{\lambda}_i + \varepsilon_{ij}$.

[5]We presented only the theoretical part of this approach in [Umyarov and Tuzhilin 2007] and did not do any experimental analysis that is reported in this paper.

By considering the compound disturbance term, we make the model a Generalized Least Squares linear regression model (GLS) [Greene 2002].

The covariance structure of residuals η_{ij} can be determined from equations (13) and (17) using simple algebraic computations that result in the following expressions:

$$\begin{cases} E\eta_{ij} = 0, \\ E\eta_{ij}\eta_{kl} = 0, \quad \text{if } i \neq k \text{ and } j \neq l, \\ E\eta_{ij}\eta_{ik} = \boldsymbol{w}_j'\Lambda\boldsymbol{w}_k, \quad \text{if } j \neq k, \\ E\eta_{ij}\eta_{kj} = \boldsymbol{z}_i'\Gamma\boldsymbol{z}_k, \quad \text{if } i \neq k, \\ E\eta_{ij}^2 = \sigma^2 + \boldsymbol{z}_i'\Gamma\boldsymbol{z}_i + \boldsymbol{w}_j'\Lambda\boldsymbol{w}_j, \end{cases} \tag{18}$$

where expected value $E(\cdot)$ is taken over ε_{ij}, $\boldsymbol{\lambda}_i$ and $\boldsymbol{\gamma}_j$, and parameters Γ, Λ and σ are defined in (14).

Let $\boldsymbol{\eta}$ be a *vector* consisting of all the residuals η_{ij} corresponding to observed ratings r_{ij}, and let $\Omega = \mathrm{Var}(\boldsymbol{\eta})$ be its covariance matrix.

From (18), we conclude that Ω depends just on a few unknown parameters σ, Γ and Λ. If we consistently estimate these parameters, then parameter $\boldsymbol{\mu}$ of model (13) can be estimated asymptotically efficiently using the *Feasible GLS (FGLS)* estimator approach as shown in [Greene 2002]:

$$\hat{\boldsymbol{\mu}} = \left(X'\hat{\Omega}^{-1}X\right)^{-1} X'\hat{\Omega}^{-1}\boldsymbol{r}, \tag{19}$$

where $\boldsymbol{r}$ is a column-vector of observed scalars r_{ij} stacked on top of each other, so the first element of the vector is scalar $r_{i_1j_1}$, the second element is $r_{i_2j_2}$ and so on. X is a matrix of row-vectors $\boldsymbol{x}_{ij}'$ stacked on top of each other one-by-one; thus the first row of matrix X is the row-vector $\boldsymbol{x}_{i_1j_1}'$ corresponding to observation $r_{i_1j_1}$, the second row of matrix X is the row-vector $\boldsymbol{x}_{i_2j_2}'$ and so on. $\hat{\Omega}$ is an estimate of Ω.

Next, we show how to utilize aggregate ratings to improve estimation of unknown ratings r_{ij}. First, observe that the external aggregate rating (15) can be expressed as:

$$E\left[\frac{\sum r_{ij}}{k}\right] = E\left[\frac{\sum\left(\boldsymbol{x}_{ij}'\boldsymbol{\mu} + \boldsymbol{z}_i'\boldsymbol{\gamma}_j + \boldsymbol{w}_j\boldsymbol{\lambda}_i + \varepsilon_{ij}\right)}{k}\right] = \tag{20}$$

$$= \frac{\sum \boldsymbol{x}_{ij}'}{k}\boldsymbol{\mu} + \frac{\sum \boldsymbol{z}_i'\boldsymbol{\gamma}_j}{k} + \frac{\sum \boldsymbol{w}_j\boldsymbol{\lambda}_i}{k} = a. \tag{21}$$

We can see that the new information from equation (21) about the expected average rating can be interpreted as an additional observation. To see this, denote $\tilde{\boldsymbol{x}} = \frac{\sum \boldsymbol{x}_{ij}}{k}$ and $\tilde{\eta} = \frac{\sum \boldsymbol{z}_i'\boldsymbol{\gamma}_j}{k} + \frac{\sum \boldsymbol{w}_j\boldsymbol{\lambda}_i}{k}$. Then equation (21) is equivalent to having the following additional observation in the model:

$$a = \tilde{\boldsymbol{x}}'\boldsymbol{\mu} + \tilde{\eta}, \tag{22}$$

where the residual $\tilde{\eta}$ has a known covariation structure with other residuals η_{ij} defined in (17):

$$E[\tilde{\eta}\eta_{ij}] = \sum_{\substack{t: \\ (i,t)\in S}} \frac{\boldsymbol{w}_j'\Lambda\boldsymbol{w}_t}{k} + \sum_{\substack{t: \\ (t,j)\in S}} \frac{\boldsymbol{z}_i'\Gamma\boldsymbol{z}_t}{k}, \tag{23}$$

$$E[\tilde{\eta}^2] = \sum_{\substack{i,j,t: \\ (i,j)\in S, \\ (i,t)\in S}} \frac{\boldsymbol{w}_j' \Lambda \boldsymbol{w}_t}{k^2} + \sum_{\substack{i,j,t: \\ (i,j)\in S, \\ (t,j)\in S}} \frac{\boldsymbol{z}_i' \Gamma \boldsymbol{z}_t}{k^2}. \tag{24}$$

where S is the set of user-item pairs as defined in (15).

Therefore, the constrained model still fits the GLS paradigm. Note that for the GLS estimator, equations (23) and (24) amount to introducing an additional row and a column to matrix Ω corresponding to covariances (23) and (24). That is,

$$\tilde{\Omega} = \left(\begin{array}{c|c} \Omega & * \\ \hline * & * \end{array} \right),$$

where $*$ denotes these additional column and row. Thus, by including this additional observation we create the extended covariance matrix $\tilde{\Omega}$ from the matrix Ω.

As we explained before, parameter $\boldsymbol{\mu}$ in model (13) can be estimated asymptotically efficiently using the *Feasible GLS (FGLS)* estimator approach (19).

Although the basic solution works well with certain optimizations for small datasets, such as the Movie Rating dataset from [Adomavicius et al. 2005] as described in Section 7, more scalable methods are needed for larger datasets. As expression (19) demonstrates, straightforward estimation of $\boldsymbol{\mu}$ requires inverting matrix $\hat{\Omega}$ that is of size $N \times N$, where N is the total number of ratings, which is usually very large, even for the medium-size problems[6]. Thus, the GLS estimator solution described in this subsection requires the inversion of a large matrix $\hat{\Omega}$, which is very hard to do because it does not have a "nice" structure (e.g., it is not block-diagonal, as matrix Θ described in Section 5.4 is).

Therefore, the "natural" basic solution presented in this subsection is not computationally feasible, even for the medium-sized problems. To address this scalability issue, we propose an alternative more advanced method in the next section that scales well to significantly larger problems.

5.4 Scalable solution

In order to overcome the difficulty of inverting matrix $\hat{\Omega}$ while preserving the given covariance structure (18), our model (13) can be rewritten as follows:

$$r_{ij} = \boldsymbol{x}_{ij}'\boldsymbol{\mu} + \underbrace{\begin{pmatrix} \mathbf{0} & \cdots & \boldsymbol{w}_j' & \mathbf{0} & \cdots & \boldsymbol{z}_i' & \cdots & \mathbf{0} \end{pmatrix}}_{\boldsymbol{y}_{ij}} \underbrace{\begin{pmatrix} \boldsymbol{\lambda}_1 \\ \vdots \\ \boldsymbol{\lambda}_i \\ \vdots \\ \gamma_1 \\ \vdots \\ \gamma_j \\ \vdots \end{pmatrix}}_{\boldsymbol{u}} + \varepsilon_{ij} \tag{25}$$

[6]For example, the Netflix Prize dataset has on the order of 100,000,000 ratings [Bennett and Lanning 2007].

where $\boldsymbol{u} = \begin{pmatrix} \boldsymbol{\lambda}_1' & \cdots & \boldsymbol{\lambda}_i' & \cdots & \boldsymbol{\gamma}_1' & \cdots & \boldsymbol{\gamma}_j' & \cdots \end{pmatrix}'$ and $\boldsymbol{y}_{ij} = \begin{pmatrix} \mathbf{0} & \cdots & \boldsymbol{w}_j' & \mathbf{0} & \cdots & \boldsymbol{z}_i' & \cdots & \mathbf{0} \end{pmatrix}$
Therefore expression (25) becomes:

$$r_{ij} = \boldsymbol{x}_{ij}'\boldsymbol{\mu} + \boldsymbol{y}_{ij}'\boldsymbol{u} + \varepsilon_{ij} \tag{26}$$

[Neumaier and Groeneveld 1995] show that the models of type (26) are equivalent to the following augmented model:

$$\begin{pmatrix} \boldsymbol{r} \\ \mathbf{0} \end{pmatrix} = \underbrace{\begin{pmatrix} X & Y \\ 0 & I \end{pmatrix}}_{A} \begin{pmatrix} \boldsymbol{\mu} \\ \boldsymbol{u} \end{pmatrix} + \underbrace{\begin{pmatrix} \boldsymbol{\varepsilon} \\ -\boldsymbol{u} \end{pmatrix}}_{\boldsymbol{\nu}} \tag{27}$$

where $\boldsymbol{r}$, X and Y are observations r_{ij}, $\boldsymbol{x}_{ij}'$ and $\boldsymbol{y}_{ij}'$ stacked on top of each other respectively, as explained in (19), and $\boldsymbol{\mu}$, $\boldsymbol{u}$ are unobserved parameters. Denote

$$\begin{cases} \boldsymbol{\nu} = \begin{pmatrix} \boldsymbol{\varepsilon} \\ -\boldsymbol{u} \end{pmatrix}, \quad A = \begin{pmatrix} X & Y \\ 0 & I \end{pmatrix} \\ \mathrm{Var}[\boldsymbol{\nu}] = \Theta \end{cases} \tag{28}$$

As in the case of the FGLS solution (19), the FGLS estimator for the augmented model (27) is

$$\begin{pmatrix} \hat{\boldsymbol{\mu}} \\ \hat{\boldsymbol{u}} \end{pmatrix} = \left(A'\Theta^{-1}A\right)^{-1} A'\Theta^{-1} \begin{pmatrix} \boldsymbol{r} \\ \mathbf{0} \end{pmatrix} \tag{29}$$

The trick of analyzing model (27) instead of (17) is that the disturbance term in this model is $\boldsymbol{\nu}$, not $\boldsymbol{\eta}$. Unlike the case of $\mathrm{Var}[\boldsymbol{\eta}] = \Omega$ (as discussed in Section 5.3), the covariance matrix $\mathrm{Var}[\boldsymbol{\nu}] = \Theta$ has the following *block-diagonal* structure[7]:

$$\begin{pmatrix} \boxed{\sigma^2} & 0 & \cdots & & & 0 \\ 0 & \ddots & & & & \\ & & \boxed{\sigma^2} & & & \\ & & & \boxed{\Lambda} & & \vdots \\ & & & & \ddots & \\ \vdots & & & & \boxed{\Lambda} & \\ & & & & & \boxed{\Gamma} \\ & & & & & \ddots \quad 0 \\ 0 & & \cdots & & & 0 \quad \boxed{\Gamma} \end{pmatrix} \tag{30}$$

For this reason, it is easy and scalable to invert Θ, unlike the case of the more complex matrix Ω described in Section 5.3. Note also, that Θ is a block-diagonal matrix filled with σ, Λs and Γs on the main diagonal as shown in (30). That is, the entire large covariance matrix Θ is completely determined from parameters σ, Λ and Γ.

[7]This follows immediately from the definition of $\boldsymbol{\nu}$.

In order to estimate the unknown variance components σ, Λ and Γ, we employ traditional procedure of Restricted Maximum Likelihood (REML) estimator [Geert Verbeke 2000]. According to the theory presented in [Geert Verbeke 2000], the values $\hat{\sigma}, \hat{\Lambda}, \hat{\Gamma}$ that minimize the following expression are consistent estimators of σ, Λ, Γ:

$$\{\hat{\sigma}, \hat{\Lambda}, \hat{\Gamma}\} = \arg\min \left[\hat{\boldsymbol{\nu}}' \Theta^{-1} \hat{\boldsymbol{\nu}} + \log\det\Theta + \log\det(A'\Theta^{-1}A)\right] \tag{31}$$

where $\hat{\boldsymbol{\nu}}$ is a vector of ordinary least squares (OLS) residuals of regression model (27) and A is defined in (27) and (28).

Parameters $\hat{\sigma}$, $\hat{\Lambda}$ and $\hat{\Gamma}$ can be computed based on (31) using the BFGS Quasi-Newton method with a mixed quadratic and cubic line search gradient descent procedure [Fletcher 1980] that guaranteed to converge to a local minimum.

In summary, our proposed scalable solution can be described as follows:

(1) Estimate variance components σ, Λ and Γ using REML estimator (31)
(2) Estimate regression components $\boldsymbol{\mu}$, $\boldsymbol{u}$ (and therefore, $\{\boldsymbol{\lambda}_i\}$, $\{\boldsymbol{\gamma}_j\}$) using equation (29).

To show scalability of the proposed methods, we need to show that the REML estimator of variance components and the estimator of $\boldsymbol{\mu}$, $\{\boldsymbol{\lambda}_i\}$, $\{\boldsymbol{\gamma}_j\}$, as described at the end of Section 5.4, can be computed efficiently even for the large-scale problems.

The REML estimator, as defined by (31), requires an iterative optimization procedure, for which it is difficult to derive good theoretical estimates of computing time. However, what we should note is that REML estimator estimates only a few unknown parameters σ, Λ and Γ. The number of these parameters is fixed and *does not* depend on the sample size nor on the number of users and items. Therefore, we do not need to use the whole sample of observations, users and items to estimate these parameters, and need only a smaller random subsample to generate good estimates of σ, Λ and Γ. Thus, in practice, the computational complexity of this estimator is $O(1)$ in terms of the sample size and the number of users and items.

For the scalable estimator of $\boldsymbol{\mu}$ and $\boldsymbol{u}$, the computational complexity stems from the complexity of solving the system of linear equations[8] (29) and matrix multiplication in (29). The number of unknowns in (29) is $O(N_U + N_I)$, where N_U is the number of users and N_I is the number of items. Thus, the complexity of solving the system of linear equations in (29) is $O\left((N_U + N_I)^3\right)$. The complexity of matrix multiplication in (29) is $O(N(N_U + N_I))$, because of the very sparse and block diagonal structure of Θ as shown in (30), where N is the total number of known ratings. Thus, the overall complexity of the advanced estimator is $O\left((N_U + N_I)^3 + N(N_U + N_I)\right)$.

We should note however, that in practice due to specific structure of matrices A and Θ, multiple matrix multiplications $A'\Theta^{-1}A$ with different Θs can be done very efficiently. More specifically, the "dense" part of A, that is (XY) in equation (28),

[8]When estimating $\boldsymbol{\mu}$ and $\boldsymbol{u}$ in (29) in practice, we do not need to actually invert the matrix $A'\Theta^{-1}A$. Instead, we need to solve the system of linear equations, e.g. using the Gaussian elimination technique.

is multiplied only by a diagonal matrix of Θ consisting of σ^{-2}, and only the sparse part of A is multiplied by Λ^{-1} and Γ^{-1}.

Note that this is a "reasonable" complexity, because the number of users N_U and items N_I in practical applications is not as big and does not grow as fast as the number of actual ratings N. For instance, the Netflix prize dataset, which constitutes a very large-scale problem, has less than 18 thousand items, less than 500 thousand users, but over 100 million ratings. Fortunately, the complexity of the presented method in terms of N alone is linear.

In our experiments, reported in Section 8, we used the advanced solution described in Section 5.4 that we applied to medium-size datasets. As we describe in Section 8, computational performance was "manageable" on these datasets, which confirms the conclusions of this subsection that the computational performance of the REML estimator and the regression components are "reasonable."

6. THEORETICAL RESULTS

We have presented two theory-based models incorporating aggregate ratings, i.e., model-based CF in Section 4.1 and HLM in Section 5. In this section, we theoretically demonstrate that these two models provide for better rating estimations than the standard individual rating models without aggregate rating information. We start with the model-based CF by formulating and proving the following theorem.

THEOREM 6.1. *The expected prediction mean squared error of the constrained model-based CF rating estimator (CRE) is smaller than the expected mean squared error of the unconstrained model-based CF rating estimator (URE).*

The proof of this theorem is in Appendix A, and it is based on the following idea. If we compare the variance $\mathrm{Var}[\boldsymbol{r}_{Uj}|\boldsymbol{r}_{Kj} = \boldsymbol{y}]$ of the unconstrained estimator (3) and the variance $\mathrm{Var}[\boldsymbol{r}_{Uj}|\boldsymbol{r}_{Kj} = \boldsymbol{y}, r^a = k]$ of the constrained estimator (8), then algebraically the following relationship holds:

$$\underbrace{\mathrm{Var}[\boldsymbol{r}_{Uj}|\boldsymbol{r}_{Kj} = \boldsymbol{y}, r^a = k]}_{\text{constrained}} = \underbrace{\mathrm{Var}[\boldsymbol{r}_{Uj}|\boldsymbol{r}_{Kj} = \boldsymbol{y}]}_{\text{unconstrained}} - V$$

where V is some non-negative definite matrix. Since both estimators are unbiased, then the lower standard error of the estimator implies the lower mean squared error of predictions [Bishop and Nasrabadi 2007]. Therefore, the expected mean squared error of the predicted ratings can only decrease from the additional aggregate rating information.

The next theorem pertains to the HLM model from Section 5.

THEOREM 6.2. *The expected prediction mean squared error of the HLM estimator with aggregate information is smaller than the expected mean squared error of the HLM estimator without aggregate information.*

The proof of this theorem is in Appendix A, and it is based on a similar idea that is behind the proof of Theorem 6.1: if we compare algebraically the variance $\mathrm{Var}[\hat{r}_{ij}]$ of the HLM estimator without aggregate information and the variance $\mathrm{Var}[\hat{r}^*_{ij}]$ of the HLM estimator with aggregate information, then it is possible to algebraically

derive the following relationship:

$$\underbrace{\mathrm{Var}[\hat{r}^*_{ij}]}_{\text{with aggr.}} = \underbrace{\mathrm{Var}[\hat{r}_{ij}]}_{\text{w/o aggr.}} - v$$

where v is some non-negative number. These two estimators are also unbiased. Therefore, lower variance leads to lower expected mean squared error when using aggregate ratings.

Although the proofs of these two theorems are technically quite different, both of them are based on the same fundamental idea that the extra aggregate ratings information can be interpreted as *some sort of an additional observation* ("pseudo-observation") in the sample. Moreover, the sample size usually matters in the sense that an estimator trained on a bigger sample will, on average, outperform the estimator trained on a smaller sample. Then we mathematically show (using different techniques) that these additional "pseudo-observations" obtained from the aggregate ratings data indeed help to decrease the variance of the rating estimators in both cases.

Furthermore, we believe that this fundamental idea is not limited just to the model-based CF and the HLM models and that Theorems 6.1 and 6.2 can be generalized to a broader class of recommendation models. Determination of this broader class of models constitutes a topic of our future research, as will be explained further in Section 10.

In this section, we theoretically demonstrated that using aggregate ratings in the model-based CF and the HLM models indeed improves estimations of unknown individual ratings. In the next section we present the results of our empirical study in which we experimentally confirm these theoretical findings.

7. EMPIRICAL SETTINGS

In this section, we describe the data used in our experiments, partitioning of the data into the training and the testing sets, and the performance measures used in our experiments.

7.1 Individual rating datasets

We used the following "real-life" datasets for learning individual ratings and empirically validating our methods.

7.1.1 *MovieLens Dataset.* We used the full MovieLens dataset [MovieLens 2006] consisting of more than 1 million ratings of 3900 movies provided by 6040 users. The user attributes included age and gender. Movie attributes included movie release year and genres represented by 7 dummy variables taken from the IMDB record corresponding for each movie.

7.1.2 *Movie Rating Dataset.* We used the data from the study [Adomavicius et al. 2005] on 61 users that provided 1110 ratings for 62 movies. More specifically, the dataset from [Adomavicius et al. 2005] 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 the movies, the dataset from [Adomavicius et al. 2005] pro-

vides the movie title and the 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.

7.1.3 *Subsample #1 of the Netflix Prize Dataset.* We also used a random subsample of the Netflix Prize dataset [Bennett and Lanning 2007] consisting of 10,000 users and 10,000 movies with 200,000 ratings. The subsample was produced using the following procedure:

(1) Select 10,000 random users from the set of all the Netflix users ranked between #10,000 and #300,000 based on the total number of ratings they gave.
(2) Select 10,000 random movies out of the movies that these 10,000 users watched.
(3) Select 200,000 random ratings out of the ratings that these 10,000 users provided for these 10,000 movies.

This dataset contains the release year and the genre (represented by 7 dummy variables) for the movies and no attributes for the users since the Netflix Prize dataset [Bennett and Lanning 2007] contains *no data at all* about their customers beyond the customer ID number. For movies, the Netflix Prize dataset [Bennett and Lanning 2007] provides the movie title and the release year. For users' ratings, the dataset contains the timestamp when the rating appeared on the website.

Finally, each of these 3 datasets was split into ten subsets for the 10-fold cross validation. All rating data were normalized to the [0,1] interval.

7.1.4 *Subsample #2 of the Netflix Prize Dataset.* Another random subsample of the Netflix Prize dataset [Bennett and Lanning 2007] consisting of 1,000 users and 1,000 movies with 5,000 ratings. The subsample was produced using the same procedure as described in Section 7.1.3.

7.2 Aggregate rating dataset

In order to introduce aggregate rating information from the external sources into the Individual Rating datasets described above, we extracted from the IMDB database [IMDB 2006] the average ratings of the movies used in those datasets, i.e. for each movie in the Individual Rating datasets, we attempted to find a corresponding average movie rating from IMDB. The results of this matching process are presented in the following table:

Name of dataset	Total Number of Movies	Movies Matched
MovieLens	3,952	2,162
Movie Rating Dataset	62	62
Netflix #1	10,000	9,949
Netflix #2	1,000	998

We also extracted the information from IMDB on the number of votes used to compute the aggregate rating.

7.3 Training and testing strategies

In order to empirically validate our approach, we split each dataset into 10 subsets for the 10-fold cross validation. The Netflix and the Movie dataset from [Adomavi-

cius et al. 2005] were splitted randomly into 10 subsets with 90% of the data going into training set and 10% of the data going into the test set.

For the MovieLens dataset, we used a different procedure. MovieLens dataset is an example of a fairly dense dataset, and it does not contain enough seldomly rated items that are used in our studies described in Section 9. Moreover, a significant number of these movies are not represented in the IMDB database. Therefore, we used the following special sampling procedure to increase the number of seldomly rated movies with the available aggregate rating information that we needed in our experiments:

(1) Find all items having from 2 to 5 observed ratings in the dataset and put 1 random rating of these into the training set and the rest into the test set
(2) Find all items having from 6 to 7 observed ratings in the dataset and put 2 random ratings of these into the training set and the rest into the test set
(3) Find all items having 8 observed ratings in the dataset and put 3 random ratings of these into the training set and the rest into the test set
(4) Find all items having 9 observed ratings in the dataset and put 4 random ratings of these into the training set and the rest into the test set
(5) Find all items having 10 observed ratings in the dataset and put 5 random ratings of these into the training set and the rest into the test set

This procedure is repeated 10 times for MovieLens dataset in order to obtain 10 different random training and test sets.

7.4 Performance measures

We compute the performance measure as follows. First, we select k aggregate observations in some predefined random order for $k = 0, 1, 2, 3, \ldots$, and incorporate them into the individual Rating Datasets, as described in the ARRM models above. Then, for each k, we calculate the average mean squared error (MSE) of predictions of the models that use exactly k aggregate ratings across all the aforementioned test sets. Finally, we plot the graph of these MSEs for each value of $k = 0, 1, 2, 3, \ldots$, as is shown, for example, in Figure 3 for the case of the MovieLens dataset. Note that $k = 0$ means that *no* aggregate rating information is used at all, and we are dealing with the basic individual rating prediction model in this case.

Based on the theoretical results from Section 6 (Theorems 6.1 and 6.2), we conjecture that the MSEs in these graphs should decrease with the number k of available aggregate ratings, and we report the results of these experiments in Section 8.

8. EMPIRICAL RESULTS

For the Netflix #1 dataset, we run the models only on items having no more than 4 observed ratings for the following reasons. First, as argued in Section 9, these items are expected to benefit the most from the aggregate information. Second, more than 4,000 items out of 10,000 qualify for this criteria as reported in Figure 1, and therefore we have a large sample of items for our experiment. Third, as explained in Section 4.1.4, our method is scalable for this setting. For the MovieLens dataset, we also run the model only on items having no more than 4 observed ratings for the same reasons as for the Netflix #1 dataset. Furthermore, for the Netflix #2

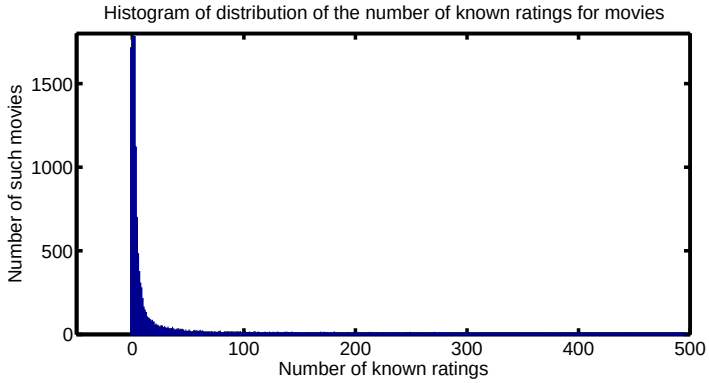


Fig. 1. Histogram of the distribution of the number of known ratings in Netflix #1 dataset

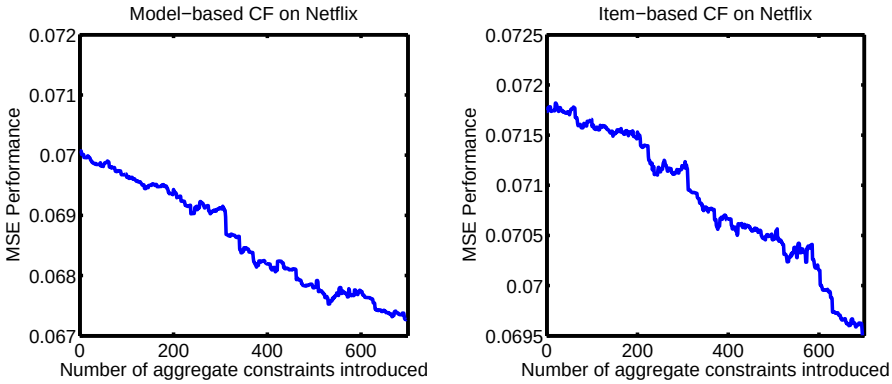


Fig. 2. MSE drifts down on a subset #1 of Netflix as more aggregate information is introduced

and the Movie Rating dataset from [Adomavicius et al. 2005], we run the models on the full sample of data.

The graphs in Figures 2, 3, 4 represent the MSE performance of each corresponding model as a function of the number of additional aggregate ratings introduced. Each figure plots 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 the basic recommendation model with only a single aggregate rating being introduced into the model. The 2nd tick adds one more aggregate rating to the aggregate rating of the 1st tick, and so on. On the y-axis we plot the average MSE performance of the model based on the 10-fold cross-validation. As Figures 2, 3, 4 show, the MSE measure drifts down in all the cases and we explain this phenomenon further below.

We next show for each model and each dataset by how much the MSE performance is improved for the case of no aggregate information vs. when all the available aggregate information is used. For example, in Figure 3, we compare the MSE of ≈ 0.05 when $k = 0$ which corresponds to no aggregate information vs. the

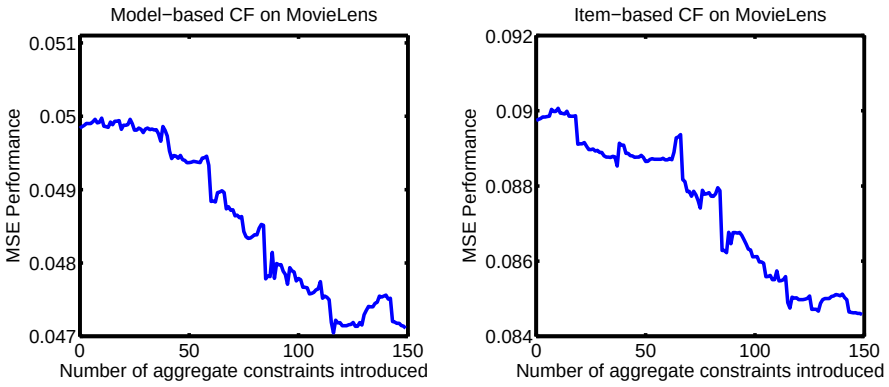


Fig. 3. MSE drifts down on a MovieLens dataset as more aggregate information is introduced

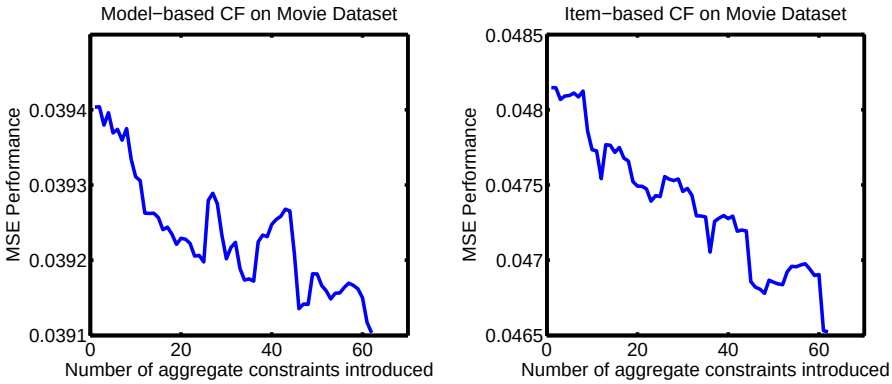


Fig. 4. MSE drifts down on a Movie dataset as more aggregate information is introduced

MSE of ≈ 0.047 when $k = 149$ for the case of all the available aggregate information being used. The results of such MSE improvements across all the 6 graphs in Figures 2, 3, 4 are presented in the following table:

Dataset	# of aggr.ratings	Model-based CF	Item-based CF
MovieLens	149	5.7%	6.1%
Movie Dataset	62	0.77%	3.4%
Netflix #1	696	4.1%	3.2%

We would like to emphasize that these results constitute solid performance improvements, given the number of additional aggregate ratings they are based on. For comparison, the $1,000,000 Grand Prize of the Netflix Prize Competition required the total performance improvement of 10% for the RMSE. Moreover, if the leading competitor of the Netflix Competition could achieve an RMSE performance improvement of 0.85% today (as of September 14, 2008), that competitor would have won the $1,000,000 Netflix Grand Prize.

Similarly, the graphs in Figure 5 represent the MSE performance of the HLM

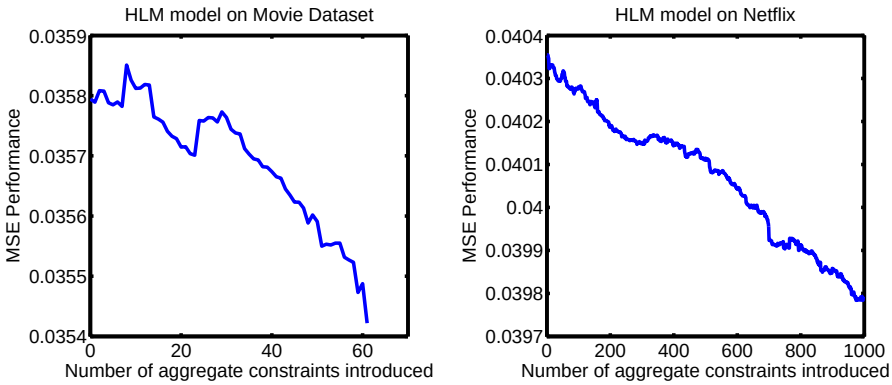


Fig. 5. MSE drifts down for HLM model as more aggregate information is introduced

model presented in Section 5 as a function of the number of additional aggregate ratings introduced.

For the HLM model, we also show in the following table by how much the MSE performance is improved for the case of no aggregate information vs. when all the available aggregate information is used:

Dataset	# of aggr.ratings	HLM model
Movie Dataset	62	1.0%
Netflix #2	998	1.4%

These results also constitute significant improvements over the basic non-aggregated model, as was argued before.

Figures 2, 3, 4 and 5 clearly show that MSE drifts down as the number of aggregate ratings increases, but not in a purely monotonic fashion. More generally, the graphs of MSE on Figures 2, 3, 4 and 5 represent an underlying stochastic process [Karatzas and Shreve 1991]. Theorems 6.1 and 6.2 state that the *expected* MSE of the respective models can only decrease as more and more aggregate ratings are introduced. In other words, these theorems state that *MSE process is a stochastic process with the drift down*. Therefore, occasional non-monotonic jumps can happen because adding one particular realization of an aggregate rating to the 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 that happen to give the ratings in MovieLens dataset, and this aggregate rating may not fit well with the particular individual ratings given to movie "Madagascar" by the users in our dataset. However, as Theorems 6.1 and 6.2 state, the stochastic process should have a drift down and this is *exactly* what we observe for all the datasets and for all the models in Figures 2, 3, 4 and 5 .

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 in Figures 2, 3, 4 and 5 are the results of a stochastic process with zero drift with MSE improvements jumping arbitrarily up and down as additional aggregate ratings are

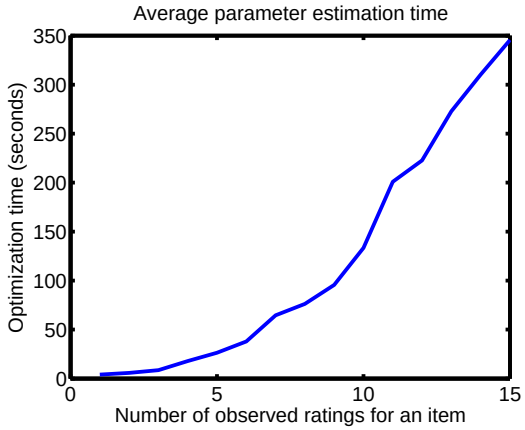


Fig. 6. Parameter estimation time for 1 item in $10,000 \times 10,000$ dataset

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. There is no single case of the MSE of the constrained estimator with 50-100 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 very unlikely under the assumption that the performance improvement is just a random white noise, which is in line with the result of Theorems 6.1 and 6.2.

Computational Performance and Scalability. We tested the basic solution method for CF described in Section 4 on the Movie Rating dataset from [Adomavicius et al. 2005] that was implemented in MATLAB and run on Intel Xeon CPU 3.73GHz. In particular, Step 1 of the estimation procedure from Section 4.1.2, i.e. the estimation of the unknown parameters $\boldsymbol{\mu}$ and of the complete matrix Σ using the iterative gradient descent algorithm, took on the order of 6 hours. It follows from this that the basic solution is applicable only to small datasets such as the Movie Rating dataset.

To test the computational performance of the scalable CF solution presented in Section 4, we ran the algorithm on a much larger datasets, such as Netflix #1 and MovieLens. As described in this Section, we estimated the ratings for the items with no more than 4 known ratings and the estimation procedure implemented in MATLAB on Intel Xeon CPU 3.73GHz took 19.5 hours for MovieLens and 22.1 hours for the Netflix #1 dataset. More generally, computational performance of the scalable CF solution depends on the number of already observed ratings for an item, as presented in Figure 6 and therefore is reasonable for the items with few ratings. More specifically, Figure 6 depicts the average CPU time spent on Intel Xeon CPU 3.73GHz by the optimization procedure minimizing the value function (11) for a single item as a function of the number of already observed ratings for this item.

The computational time for the scalable estimator of HLM model was reasonable on all the datasets. In particular, the solution of the REML optimization problem (31) implemented in MATLAB on Intel Xeon CPU 3.73GHz took 6.5 hours for

Netflix #2 dataset. As pointed out in Section 5.4, this computational time does not grow in the number of ratings, users and items, and therefore is to be considered as a constant. After the parameters were estimated, the solution of (29) took less than 1 second for the whole dataset used.

9. USING AGGREGATE RATINGS FOR SOLVING THE COLD START PROBLEM

In this section, we explore the relationship between the proposed aggregate rating methods described in the paper and the *cold start* problem of recommender systems [Schein et al. 2002]. In particular, we show how our proposed methods can be used as a solution to the cold start problem.

9.1 The Cold Start Problem and the Long Tail of Recommender Systems

One of the key problems of recommender systems is the *cold start* problem: how to recommend items that have no ratings at all or only few ratings. This is a serious problem because a significant majority of the items fall into this category by belonging to the Long Tail of recommender systems [Park and Tuzhilin 2008]. This observation is also confirmed, for example, in Figure 1 for the Netflix #1 dataset, where the vast majority of the movies have only few available ratings. Furthermore, as [Park and Tuzhilin 2008] demonstrates empirically, the prediction error rates of individual rating recommender systems increase significantly for the items belonging to the Long Tail due to the lack of the available ratings data. This problem has been studied before, and various solutions, such as the ones described in [Schein et al. 2002], have been proposed to solve the cold start problem.

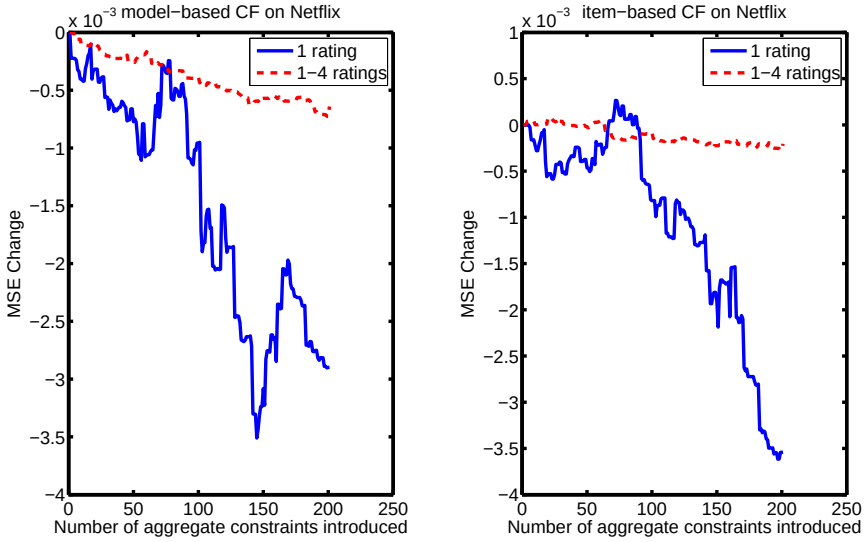
In this section we demonstrate that our methods for incorporating aggregate ratings information into recommender systems can be used as one of the possible solutions to the cold start problem. The intuition behind our idea is quite simple: when only few ratings are specified for an item, we do not have enough observations to make good estimations of unknown ratings for the individual-level models and, therefore, should rely more heavily on the aggregate ratings corresponding to this item. In contrast, if many ratings are available for an item, these ratings carry enough observations to better estimate unknown ratings for the item, and the aggregate rating information is better "revealed" in the dataset itself and therefore the *external* aggregate rating does not provide as much improvement.

The empirical evidence for this intuition is provided in the next subsection.

9.2 Empirical Results

In Sections 7 and 8, we theoretically and experimentally demonstrated that the aggregate ratings information improves overall estimations of unknown ratings. In this section we demonstrate that these improvements are even more significant for the items having only few ratings (and thus belonging to the Long Tail of recommender systems) and decrease as the number of ratings grows.

To demonstrate this effect, for the Netflix #1 and the MovieLens datasets, we applied our aggregate rating estimation methods *separately* to the items having a certain number of known ratings, starting with items having only 1 rating, only 2 ratings, etc. and built rating estimation models separately for each of these cases. Then we recorded the MSE performance improvements across these models for both datasets. The results of these experiments are recorded in Figures 7 and 8.



(a) Model-based CF on Netflix (b) Item-based CF on Netflix

Fig. 7. MSE performance improvement is better for the items having only 1 observed rating than average performance for the items having between 1 and 4 observed ratings on the Netflix #1 dataset.

Figures 7 and 8 plot the MSE improvement results for the model-based CF and the item-based CF for the Netflix #1 and MovieLens datasets respectively, as a function of the number of additional aggregate constraints introduced. On the x-axis each figure plots the cumulative number of additional aggregate ratings introduced into the model, and on the y-axis each figure plots the *change* in MSE from the case of no aggregate information to the case of having the specified amount of aggregate ratings introduced.

As Figures 7 and 8 demonstrate, the drop in the MSE error rates when additional aggregate information is added to the individual rating models is more significant when we run the model only on the items having very small number of ratings than when we add items with more observed ratings.

These results confirm our hypothesis, based on the aforementioned intuition, that when only few ratings are specified for an item, more significant predictive performance improvements can be achieved by using the aggregate rating information.

10. CONCLUSIONS

In this paper we present an approach to incorporating externally specified aggregate ratings information into certain types of recommender systems, including model- and item-based collaborating filtering and a hierarchical linear regression (HLM) models.

For the model-based CF and the HLM methods, we formally showed that this additional aggregated information provides more accurate recommendations of individual items to individual users. Furthermore, theoretical insights gained from the

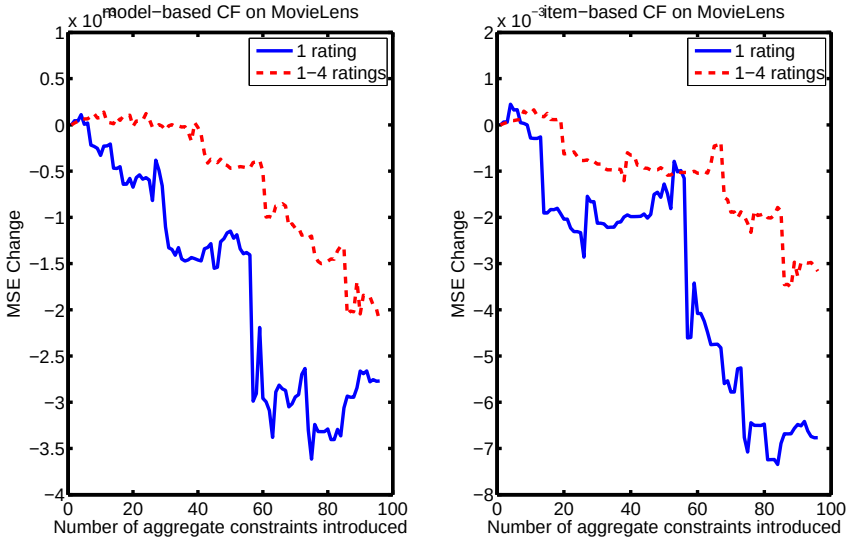


Fig. 8. MSE performance improvement is better for the items having only 1 observed rating than average performance for the items having between 1 and 4 observed ratings on the MovieLens dataset.

analysis of the model-based CF method suggested a way to incorporate aggregate information into the heuristic item-based CF method. We empirically tested all the three approaches on several datasets, and our experiments uniformly confirmed that the aggregate rating information significantly improved the basic non-aggregated recommendation models. Furthermore, we developed advanced scalable versions of the three basic aggregate recommendation models and showed that they scale well to larger recommendation problems.

Finally, we demonstrated that when only few ratings are specified for an item, more significant predictive performance improvements can be achieved by using the aggregate rating information corresponding to this item than in the case when more ratings are provided. This means, among other things, that our methods for incorporating aggregate ratings information into recommender systems can be used as a solution to the cold start problem.

The results reported in this paper are important because they demonstrate that the externally specified aggregated data is useful for providing better recommendations. Moreover, being aggregated, such data is often publicly available and does not carry privacy implications. Therefore, it can be freely and widely used in many recommender systems. Finally, our results are also important because they contribute to the solution of the difficult cold-start problem.

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. Finally, another direction for future research is the development of even more scalable and more advanced methods for faster estimation of various parameters of the aggregate rating models.

REFERENCES

Adomavicius, G., Sankaranarayanan, R., Sen, S., and Tuzhilin, A. 2005. Incorporating contextual information in recommender systems using a multidimensional approach. *ACM Transactions on Information Systems (TOIS) 23,* 1, 103–145.

Adomavicius, G. and Tuzhilin, A. 2001. Multidimensional recommender systems: a data warehousing approach. In *2nd Intl. Workshop on Electronic Commerce. LNCS 2232*.

Adomavicius, G. and Tuzhilin, A. 2005. Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions.

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

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

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

Bennett, J. and Lanning, S. 2007. The Netflix Prize. *Proceedings of KDD Cup and Workshop 2007*.

Bhatia, R. 2007. *Positive definite matrices*. Princeton University Press.

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

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

Condliff, M., Lewis, D., Madigan, D., and Posse, C. 1999. Bayesian mixed-effects models for recommender systems. *ACM SIGIR'99 Workshop on Recommender Systems: Algorithms and Evaluation 15,* 5.

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

Flury, B. 1997. *A First Course in Multivariate Statistics*.

Geert Verbeke, G. M. 2000. *Linear Mixed Models for Longitudinal Data*. Springer.

Gelman, A. 2004. *Bayesian Data Analysis*. CRC Press.

Greene, W. 2002. *Econometric Analysis*. Prentice Hall.

Hardle., W. 2004. *Nonparametric and Semiparametric Models*. Springer.

Hox, J. 2002. *Multilevel Analysis: Techniques and Applications*. Lawrence Erlbaum.

IMDB. 2006. http://www.imdb.com.

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

Karatzas, I. and Shreve, S. 1991. *Brownian Motion and Stochastic Calculus*. Springer.

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

McCarthy, K., Salamó, M., Coyle, L., McGinty, L., Smyth, B., and Nixon, P. 2006. Group recommender systems: a critiquing based approach. *Proceedings of the 11th international conference on Intelligent user interfaces*, 267–269.

MovieLens. 2006. available at http://www.grouplens.org/node/73 (as provided in 2006).

Neumaier, A. and Groeneveld, E. 1995. Restricted maximum likelihood estimation of covariances in sparse linear models. Unpublished manuscript.

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

Park, Y. and Tuzhilin, A. 2008. The Long Tail of Recommender Systems and How to Leverage It. *Proceedings of the ACM Conference on Recommender Systems*.

Raudenbush, S. W. and Bryk, A. S. 2001. *Hierarchical Linear Models: Applications and Data Analysis Methods*. Sage Publications, Inc.

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

Schafer, J., Konstan, J., and Riedl, J. 2001. E-Commerce Recommendation Applications. *DMKD 5,* 1.

Schein, A., Popescul, A., Ungar, L., and Pennock, D. 2002. Methods and metrics for cold-start recommendations. *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, 253–260.

Schwaighofer, A., Tresp, V., and Yu, K. 2004. Learning Gaussian Process Kernels via Hierarchical Bayes. *Advances in Neural Information Processing Systems 17*, 1209–1216.

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

Umyarov, A. and Tuzhilin, A. 2008. Improving Collaborative Filtering Recommendations Using External Data. *Proceedings of the IEEE ICDM 2008 Conference*.

A. APPENDIX: PROOFS OF THEOREMS 6.1 AND 6.2

A.1 Proof of Theorem 6.1

Proof. From (5), (8) 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:

$$\mathrm{Var}\left[\left.\begin{pmatrix}\boldsymbol{r}_{Uj}\\ r^a\end{pmatrix}\right| \boldsymbol{r}_{Kj} = \boldsymbol{y}\right] = \begin{pmatrix} S_{11} & S_{12}\\ S_{21} & S_{22}\end{pmatrix} \tag{32}$$

Therefore, the variance of the estimator without aggregate ratings is

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

However, as it follows from (32) and the properties of multivariate normal distribution [Flury 1997], the variance of the estimator with aggregate information is

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

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

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

That is, in terms of comparison of non-negative definite matrices, the covariance matrix of CRE is "smaller" than the 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 [Bishop and Nasrabadi 2007]. □

A.2 Proof of Theorem 6.2

PROOF. Intuitively, the proof is based on the idea that specifying an aggregate rating is equivalent to adding a new observation and on the idea that the sample size matters, i.e., the expected MSE on the test set of the estimator trained on the bigger sample size will be smaller than the expected MSE on the test set of the estimator trained on the subset of the sample.

More formally, consider the model as we have it in (17)

$$\boldsymbol{y} = X\boldsymbol{\mu} + \boldsymbol{\eta}, \quad E\boldsymbol{\eta}\boldsymbol{\eta}' = \Omega \tag{33}$$

Denote $\boldsymbol{m}$ — the GLS estimator of $\boldsymbol{\mu}$. This model doesn't take into account additional information, so we call $\boldsymbol{m}$ unrestricted estimator.

Consider also the following model

$$\boldsymbol{y}_* = X_*\boldsymbol{\mu} + \boldsymbol{\eta}_*, \quad E\boldsymbol{\eta}_*\boldsymbol{\eta}_*' = \Omega_*$$

where we just added one observation to equation (33). So X_* is just X with one additional row corresponding to the observation and Ω_* is just Ω with additional row and column corresponding to covariances of the additional observation with all other observations. That is,

$$\boldsymbol{y}_* = \left(\frac{\boldsymbol{y}}{*}\right) \quad X_* = \left(\frac{X}{*}\right)$$

and

$$\Omega_* = \left(\begin{array}{c|c} \Omega & * \\ \hline * & * \end{array}\right)$$

Denote $\boldsymbol{m}_*$ — the estimator for this model. The model takes into account the additional observation, so we call it restricted estimator.

Denote $V = \mathrm{Var}[\boldsymbol{m}]$ and $V_* = \mathrm{Var}[\boldsymbol{m}_*]$.

As we know from [Greene 2002]

$$\mathrm{Var}[\boldsymbol{m}] = \left(X'\Omega^{-1}X\right)^{-1}$$

and Cholesky decomposition of Ω:

$$\Omega = C'C$$

Thus

$$\Omega^{-1} = C^{-1}\left(C^{-1}\right)'$$

Now do the same thing for Ω_*:

$$\Omega_* = C_*'C_*$$

Actually, C_* is equal to C with an additional column (and an additional row of zeros). That is,

$$C_* = \left(\begin{array}{c|c} C & * \\ \hline 0 & * \end{array}\right)$$

It is a trivial fact since Ω_* differs from Ω just by existence of additional column and additional row. It is also a trivial fact that C_*^{-1} is equal to C^{-1} with an additional

column. That is,

$$C_*^{-1} = \left(\begin{array}{c|c} C^{-1} & * \\ \hline 0 & * \end{array}\right)$$

Consider

$$(\mathrm{Var}[\boldsymbol{m}])^{-1} = X'\Omega^{-1}X = X'C^{-1}\left(C^{-1}\right)'X$$

Consider also

$$(\mathrm{Var}[\boldsymbol{m}_*])^{-1} = X_*'\Omega_*^{-1}X_* = X_*'C_*^{-1}\left(C_*^{-1}\right)'X_*$$

As we noted, C_*^{-1} is equal to C^{-1} with an additional column, thus $\left(C_*^{-1}\right)'$ is equal to $\left(C^{-1}\right)'$ with an additional row. It is also easy to notice that $\left(C_*^{-1}\right)'X_*$ differs from $\left(C^{-1}\right)'X$ only by the addition of the last row. Denote this last row as row-vector $\tilde{\boldsymbol{x}}'$. Then,

$$\left(C_*^{-1}\right)'X_* = \left(\begin{array}{c} \left(C^{-1}\right)'X \\ \tilde{\boldsymbol{x}}' \end{array}\right)$$

It means that

$$\overbrace{\left(\left(C_*^{-1}\right)'X_*\right)'\left(C_*^{-1}\right)'X_*}^{(\mathrm{Var}[\boldsymbol{m}_*])^{-1}} = \tag{34}$$

$$= \underbrace{\left(\left(C^{-1}\right)'X\right)'\left(C^{-1}\right)'X}_{(\mathrm{Var}[\boldsymbol{m}])^{-1}} + \underbrace{\tilde{\boldsymbol{x}}\tilde{\boldsymbol{x}}'}_{\text{positive semidefinite}} \tag{35}$$

For positive-semidefinite matrices A and B, we write that $A \succeq B$ if $\exists$ positive-semidefinite matrix C such as

$$A = B + C$$

In terms of these positive-semidefinite inequalities, we can rewrite equation (34) as follows

$$(\mathrm{Var}[\boldsymbol{m}_*])^{-1} \succeq (\mathrm{Var}[\boldsymbol{m}])^{-1}$$

As we know from theory of positive-semidefinite inequalities [Bhatia 2007], it means that

$$\underbrace{\mathrm{Var}[\boldsymbol{m}_*]}_{V^*} \preceq \underbrace{\mathrm{Var}[\boldsymbol{m}]}_{V}$$

So there is a precise sense in which we can say that the covariance matrix of the restricted estimator V^* is actually smaller than the covariance matrix V of the unrestricted one.

Now consider predictions that we make from these two models for some vector of regressors $\boldsymbol{x}$:

$$\begin{cases} \hat{y} = \boldsymbol{x}'\boldsymbol{m}, \quad E[\boldsymbol{x}'\boldsymbol{m}] = \boldsymbol{x}'\boldsymbol{\mu}, \mathrm{Var}[\boldsymbol{x}'\boldsymbol{m}] = \boldsymbol{x}'V\boldsymbol{x} \\ \hat{y}^* = \boldsymbol{x}'\boldsymbol{m}^*, \quad E[\boldsymbol{x}'\boldsymbol{m}] = \boldsymbol{x}'\boldsymbol{\mu}, \mathrm{Var}[\boldsymbol{x}'\boldsymbol{m}] = \boldsymbol{x}'V^*\boldsymbol{x} \end{cases}$$

We know that $V^* \preceq V$. We also assume that $\tilde{\boldsymbol{x}} \not\equiv \mathbf{0}$ in equation (34), that is the constraint is informative. Algebraically, it means that

$$\begin{cases} \forall \boldsymbol{x}: \quad \boldsymbol{x}'V^*\boldsymbol{x} \leq \boldsymbol{x}'V\boldsymbol{x} \\ \exists \boldsymbol{x} \text{ such that } \boldsymbol{x}'V^*\boldsymbol{x} < \boldsymbol{x}'V\boldsymbol{x} \end{cases} \tag{36}$$

Denote y a **true value** at test data point. That is, the test data point itself is going to be a noisy measurement of this true value:

$$y_t = y + \eta$$

Denote $\boldsymbol{x}$, $\boldsymbol{z}$, $\boldsymbol{w}$ corresponding observables. According to the famous equation for expected MSE [Hardle. 2004], the MSE between the true value and predicted value for the unrestricted estimator is

$$E[\text{MSE}_U|\boldsymbol{x}] = E[\hat{y} - y]^2 = \text{bias}^2 + \text{Var}[\hat{y}] = \boldsymbol{x}'V\boldsymbol{x}$$

since given our assumption about independence of residuals and regressors, the GLS estimator is unbiased, so bias = 0.

Similarly, expected MSE of the restricted estimator is

$$E[\text{MSE}_R|\boldsymbol{x}] = \boldsymbol{x}'V^*\boldsymbol{x}$$

Taking into account equation (36), we get that

$$\begin{cases} \forall \boldsymbol{x}: E[\text{MSE}_R|\boldsymbol{x}] \leq E[\text{MSE}_U|\boldsymbol{x}] \\ \exists \boldsymbol{x} \text{ such that } E[\text{MSE}_R|\boldsymbol{x}] < E[\text{MSE}_U|\boldsymbol{x}] \end{cases}$$

So assuming not pathological data generation mechanism for $\boldsymbol{x}$, that is it can possibly generate $\boldsymbol{x}$ such as inequality holds in eq.(36), then it is clear that

$$\underbrace{E_{\boldsymbol{x}}\left[E[\text{MSE}_R|\boldsymbol{x}]\right]}_{E[\text{MSE}_R]} < \underbrace{E_{\boldsymbol{x}}\left[E[\text{MSE}_U|\boldsymbol{x}]\right]}_{E[\text{MSE}_U]}$$

Thus,

$$E[\text{MSE}_R] < E[\text{MSE}_U]$$

So we proved that a single additional observation reduces $E[\text{MSE}]$. We apply this idea inductively and conclude that adding an additional observation can only reduce $E[\text{MSE}]$. Thus, the incorporation of multiple information on aggregate ratings can only reduce $E[\text{MSE}]$. □

B. APPENDIX: DERIVATION OF THE HLM MODEL

Consider the following steps in deriving this model:

(1) Assume first we run linear regression of movie ratings r_{ij} solely on movie attributes $\boldsymbol{w}_j$:

$$r_{ij} = \boldsymbol{w}_j'\boldsymbol{\beta}_i + \varepsilon_{ij}, \quad \varepsilon_{ij} \sim N(0, \sigma^2) \tag{37}$$

That is, we run separate regressions for each user i and therefore we get the user-specific vector of coefficients $\boldsymbol{\beta}_i$. Intuitively, j-th element of each vector $\boldsymbol{\beta}_i$ is a (user-specific) "appreciation" to the j-th characteristic of movies. For

example, if j-th characteristic of a movie is movie release year, then j-th element of $\boldsymbol{\beta}_i$ will represent average "attitude" of user i towards newer or older movies.

(2) Now we say that since the vector of coefficients $\boldsymbol{\beta}_i$ is user-specific, we can try to explain **each element** of it from known user attributes $\boldsymbol{z}_i$.

$$\boldsymbol{\beta}_i = Z_i\boldsymbol{\mu} + \boldsymbol{\lambda}_i, \quad \boldsymbol{\lambda}_i \sim N(\mathbf{0}, \Lambda) \tag{38}$$

where matrix Z_i is constructed from the vector-column of the user attributes $\boldsymbol{z}_i$ as follows:

$$Z_i = \begin{pmatrix} \boldsymbol{z}_i' & 0 & \cdots & 0 \\ 0 & \boldsymbol{z}_i' & \cdots & 0 \\ & & & \\ 0 & 0 & \ddots & 0 \\ 0 & \cdots & 0 & \boldsymbol{z}_i' \end{pmatrix}$$

Intuitively, each element of $\boldsymbol{\mu}$ here represents a general "effect" of some user characteristic on his "appreciation" of some movie characteristic. For example, if j-th movie characteristic is movie release year, k-th user characteristic is user age and size of vector $\boldsymbol{z}_i$ is $|\boldsymbol{z}|$. Then the element $\boldsymbol{\mu}_{(j-1)|\boldsymbol{z}|+k}$ can be interpreted as the general effect of user age on his attitude towards movie release year. This interpretation is very similar to the interpretation of regressions with included *interaction terms* that are widely used in social research.

(3) Now we substitute eq.(38) into eq.(37) and get

$$r_{ij} = \boldsymbol{w}_j'\boldsymbol{\beta}_i + \varepsilon_{ij} = \boldsymbol{w}_j'(Z_i\boldsymbol{\mu} + \boldsymbol{\lambda}_i) + \varepsilon_{ij} =$$

$$= \underbrace{\boldsymbol{w}_j' Z_i}_{\boldsymbol{x}_{ij}'}\boldsymbol{\mu} + \boldsymbol{w}_j\boldsymbol{\lambda}_i + \varepsilon_{ij}$$

This is how we define the vector $\boldsymbol{x}_{ij}$ and if we examine the vector in detail this vector contains all "interactions" (cross-products) between elements of vectors $\boldsymbol{z}_i$ and $\boldsymbol{w}_j$.
Therefore, we achieved the following functional form for the model

$$r_{ij} = \boldsymbol{x}_{ij}'\boldsymbol{\mu} + \boldsymbol{w}_j'\boldsymbol{\lambda}_i + \varepsilon_{ij} \tag{39}$$

Now if we repeat the same procedure but at the step 1) we will regress r_{ij} on user attributes $\boldsymbol{z}_i$, instead of movie attributes $\boldsymbol{w}_j$, we will get the model

$$r_{ij} = \boldsymbol{x}_{ij}'\boldsymbol{\mu} + \boldsymbol{z}_i'\boldsymbol{\gamma}_j + \varepsilon_{ij} \tag{40}$$

(since this task is purely symmetrical of movie attributes and user attributes)

(4) Next we sum the two models from equation (39) and equation (40) to incorporate properties of both and the model has the functional form that is suggested by [Ansari et al. 2000]:

$$r_{ij} = \boldsymbol{x}_{ij}'\boldsymbol{\mu} + \boldsymbol{z}_i'\boldsymbol{\gamma}_j + \boldsymbol{w}_j'\boldsymbol{\lambda}_i + \varepsilon_{ij}$$

C. THEOREM ON COMBINATION OF ESTIMATORS

THEOREM C.1. *Assume $\hat{x}_1$ and $\hat{x}_2$ are two biased estimators of unknown quantity x with the following properties:*

$$\begin{cases} E\hat{x}_1 = a_1 x + b_1, & Var(\hat{x}_1) = v_1 \\ E\hat{x}_2 = a_2 x + b_2, & Var(\hat{x}_2) = v_2 \\ cov(\hat{x}_1, \hat{x}_2) = c_{12} \end{cases}$$

where a_1, a_2, b_1, b_2, v_1, v_2, c_{12} are known values.

Assume that we create a new estimator $\hat{x}$ as a linear combination of $\hat{x}_1$ and $\hat{x}_2$, that is

$$\hat{x} = \alpha + \beta\hat{x}_1 + \gamma\hat{x}_2$$

Then, the estimator $\hat{x}$ is unbiased and achieves the lowest variance if

$$\begin{cases} \beta = \dfrac{a_1 v_2 - c_{12} a_2}{a_1^2 v_2 - 2c_{12} a_1 a_2 + a_2^2 v_1} \\ \gamma = \dfrac{c_{12} a_1 - a_2 v_1}{a_1^2 v_2 - 2c_{12} a_1 a_2 + a_2^2 v_1} \\ \alpha = -\beta b_1 - \gamma b_2 \end{cases}$$

PROOF. First of all, we show what the expected value and variance of this new estimator are in terms of observed values.

$$E[\hat{x}] = \alpha + \beta E[\hat{x}_1] + \gamma E[\hat{x}_2] = \alpha + \beta(a_1 x + b_1) + \gamma(a_2 x + b_2) =$$

$$= \alpha + \beta b_1 + \gamma b_2 + x(\beta a_1 + \gamma a_2)$$

$$\mathrm{Var}[\hat{x}] = \beta^2 \mathrm{Var}[\hat{x}_1] + \gamma^2 \mathrm{Var}[\hat{x}_2] + 2\beta\gamma \mathrm{cov}(\hat{x}_1, \hat{x}_2) =$$

$$= \beta^2 v_1 + \gamma^2 v_2 + 2\beta\gamma c_{12}$$

Note that we restrict our attention only to unbiased estimators, therefore

$$E[\hat{x}] = \alpha + \beta b_1 + \gamma b_2 + x(\beta a_1 + \gamma a_2) = x \tag{41}$$

Among those α, β and γ that satisfy the equation (41), we would like to choose α, β and γ such that this is the estimator has the minimal variance, therefore we have to solve the constrained optimization problem to find the optimal α, β and γ

$$\min_{\alpha,\beta,\gamma} \beta^2 v_1 + \gamma^2 v_2 + 2\beta\gamma c_{12}$$

subject to

$$\begin{cases} \beta a_1 + \gamma a_2 = 1 \\ \alpha + \beta b_1 + \gamma b_2 = 0 \end{cases}$$

First of all, note that the objective function does not depend on α and the structure of the restrictions is such that α can be unambiguosly deduced once we know β and γ. That is, the optimization task is equivalent to the following

$$\min_{\beta,\gamma} \beta^2 v_1 + \gamma^2 v_2 + 2\beta\gamma c_{12}$$

subject to $\beta a_1 + \gamma a_2 = 1$

Then α can be deduced as $\alpha = -\beta b_1 - \gamma b_2$.

This is a standard quadratic programming program with linear equality constraint. Denote

$$V = \begin{pmatrix} v_1 & c_{12} \\ c_{12} & v_2 \end{pmatrix};\ A = \begin{pmatrix} a_1 & a_2 \end{pmatrix};\ \rho = \begin{pmatrix} \beta \\ \gamma \end{pmatrix};\ b = (1)$$

then the original optimization problem can be rewritten as

$$\min_{\rho} \frac{1}{2}\rho' V \rho$$

subject to $A\rho = b$.

In order to solve this optimization problem, we apply the method of Lagrange multipliers by specifying a Lagrangian function and looking for its saddle points

$$L = \frac{1}{2}\rho' V \rho - \lambda'(A\rho - b)$$

Then the saddle point is characterized by the following first-order conditions

$$L'_\rho = V\rho - A'\lambda = 0$$

$$A'\lambda = V\rho, \quad \text{so } \lambda = (AA')^{-1}AV\rho$$

Substituting it back into the Lagrangian we get

$$L = \frac{1}{2}\rho' V \rho - \rho' V A'(AA')^{-1}(A\rho - b)$$

Then,

$$L'_\rho = V\rho - (VA'(AA')^{-1}A + A'(AA')^{-1}AV)\rho + V(AA')^{-1}b = 0$$

That is,

$$\rho = (VA'(AA')^{-1}A + A'(AA')^{-1}AV - V)^{-1}VA'(AA')^{-1}b$$

$$\rho = \left(A'(AA')^{-1}A + V^{-1}A'(AA')^{-1}AV - I\right)^{-1} A'(AA')^{-1}b$$

By substituting our original expressions for V, A and b back and by simplifying the expression we get the expressions for β and γ. □