Learning From Missing Data Using Selection Bias in Movie Recommendation

Claire Vernade[1] and Olivier Cappé[2]

[1,2]LTCI, CNRS, Télécom ParisTech, Université Paris-Saclay

Abstract

Recommending items to users is a challenging task due to the large amount of missing information. In many cases, the data solely consist of ratings or tags voluntarily contributed by each user on a very limited subset of the available items, so that most of the data of potential interest is actually missing. Current approaches to recommendation usually assume that the unobserved data is missing at random.

In this contribution, we provide statistical evidence that existing movie recommendation datasets reveal a significant positive association between the rating of items and the propensity to select these items. We propose a computationally efficient variational approach that makes it possible to exploit this selection bias so as to improve the estimation of ratings from small populations of users. Results obtained with this approach applied to neighborhood-based collaborative filtering illustrate its potential for improving the reliability of the recommendation.

1 Introduction

Since the early 1990's, automated methods for recommending content to users based on historical data has been an active line of research in connection with the widespread deployment of online services [1, 2]. The Netflix prize [3, 4] was a recent highlight that triggered off a lot of attention on movie recommendation. In this contribution, we consider settings that are typical of the collaborative filtering paradigm in which the available information can be summarized by the list of ratings of some "items", contributed voluntarily by the users. The aim is to exploit these ratings originating from the whole population so as to recommend items to a specific user, based on his/her own historical data [5,6]. In practice, these methods are rarely used alone and can be complemented by using item and/or user metadata or features so as to improve the prediction.

A frequent abstraction used in this field consists in viewing the data available at some point in the process as a very large matrix of ratings, whose rows correspond to users and columns to items, that is incompletely observed. In typical datasets available in movie recommendation, the number of missing entries from this matrix is two orders of magnitude larger than the number of entries that are actually observed. The main challenge is thus to extrapolate the observed ratings despite the very large fraction of missing data. To do so, it is a standard practice to consider only the ratings that have been actually observed; the ratings that have not been observed being simply ignored. Doing so means that the sampling distribution, under which the ratings in the matrix are revealed by the data is viewed as a nuisance parameter rather than as an aspect of the data that could be use for estimation —see, e.g., [7] in the context of matrix completion.

Our goal with this paper is to investigate the gain achievable by exploiting the "selection bias" that is present in available movie recommendation datasets. This bias consists in a significant positive association between the rating of items in a given population and the natural propensity of this population to select these items. We will show in Section 2 below that this observation is robust, being present both at the scale of whole datasets but also in ratings corresponding to small subpopulations of users. To leverage this observation in a manner that stays computationally feasible in realistic scenarios, we will use a simple convex variational criterion that captures the main features of the relationship between the ratings and the item popularity. To illustrate the approach, we will specialize it to the case of neighborhood-based collaborative filtering in which the preferences of the user is extrapolated from the population of users that are closest to him/her given his/her historical ratings.

Taking into account the popularity of items in order to improve the recommendation has been considered before by [8] who design a greedy sequential preprocessing procedure aimed at subtracting different explanatory effects that may have an influence on the data. These include standard user and item rating effects as well as popularity —referred to as "support" in [8]— and time-related effects. Interestingly, [8] uses successive Bayesian regressions for each effect to reduce the vari-

ability inherent in using a linear model with many missing observations. However, our finding that the item rating and the popularity effects are strongly positively associated suggests that these should not be considered as uncorrelated effects that can be be successively eliminated in a linear regression model.

A more comprehensive model of an informative selection bias in recommendation has been investigated earlier by [9] who proposed to use generative probabilistic models involving both the observed ratings as well as latent variables. The latent variables correspond to unobservable explanatory variables that can influence both the fact that a particular rating is available as well as its value, allowing to model Not Missing At Random (NMAR) data in the sense of [10]. In [11], the authors report the results of experiments that support the necessity of such a modeling by showing a significant mismatch between the empirical distributions of voluntary ratings of user-selected songs (listened through Yahoo! Music's LaunchCast Radio) and ratings of system-selected songs by users recruited to participate to the experiment.

In contrast to [9], we do not intend to model explicitly the missing data mechanism nor to introduce latent variables representing implicit categories of the population of users. The proposed approach consists in deriving a simple regularization (or penalty) term that incorporates some general knowledge about the selection bias. This regularization term can be used with any recommendation method expressed as the solution of a variational criterion that involves the true population average of the ratings. In the context of this paper, we only consider the case where the regularization term is used to improve the estimation of movie ratings from small samples of the population in neighborhood-based prediction.

In Section 2, we provide a quantification of the selection bias phenomenon on the MovieLens and Netflix datasets. Section 3 describes our variational approach for learning ratings taking into account this selection bias. Sections 4 and 5 provide, respectively, numerical experiments on simulated data and on the MovieLens dataset.

2 Characterizing the Selection Bias

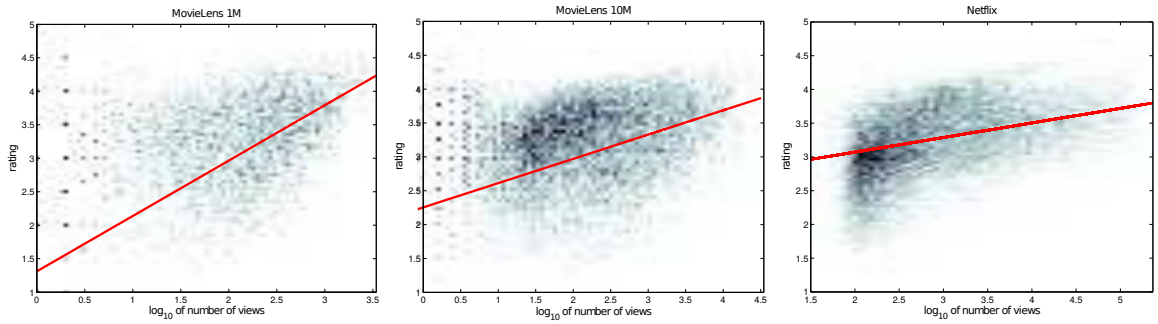
In this Section, we present statistical observations made on well-known movie rating datasets showing significant positive association between the popularity of movies and their ratings.

Figure 1 displays scatterplots of movie average rating as a function of the number of ratings corresponding to, from left to right, the MovieLens 1M (6k users, 3.7k movies, 1M ratings); MovieLens 10M (70k users, 11k movies, 10M ratings); and, Netflix (480k users, 17.8k movies, 100M ratings) datasets. The y-scales of the three subplots of Fig. 1 are directly comparable and correspond to a 1-5 scale where 5 corresponds to the highest possible rating[1]. The number of ratings are plotted on the x-scale using a base ten log scale. In particular, the number of ratings obtained by the most rated movies is observed to be roughly proportional to the overall number of ratings, which increases by a factor ten when going from one plot to the next. It is important to keep in mind that the leftmost part of each plot —especially for the two MovieLens datasets— corresponds to very small samples (movies with less than ten ratings when the x-value is smaller than one) and should not be considered as individually reliable. From Fig. 1, it is observed that despite their differences[2], the three datasets do show the same general pattern that the highest rated movies are also among the most "popular" ones (i.e., those with highest numbers of ratings). Given the relatively small spread of ratings on the y-axis (for MovieLens 10M for instance, half of the movie ratings are between 2.8 and 3.6), this positive association between the popularity of movies and their rating is rather significant; we will refer to this phenomenon as the *selection bias*.

The observed positive association does not imply a direct causality relationship. However, it is to be expected that this observation applies, to some extent, to all situations where rating an item is a voluntary action taken by the user. The important point is that the popularity of an item is in itself a valuable information even when the objective is to estimate the population average of its rating. To illustrate this fact, assume that movies A and B have comparable average ratings but that these ratings were obtained from n_A and n_B users, where n_B is larger than n_A. Standard statistical arguments suggest that the rating of item B is more reliable than that of item A because it has been estimated from more users. The positive association reinforces this observation by making the hypothesis that the actual population rating of B exceeds that of A more likely, as it has been selected more often. This effect will be most likely negligible if n_A is itself large as the statistical error in estimating the population rating of item A is small anyway in this case. On the other hand, when dealing with *small samples* —when n_A is, say, less than fifty—, taking into account the selection bias can be significant. The issue of small samples is inherent to recommendation. For instance, 32% of the 10k movies

[1] MovieLens ratings are half integers from .5 to 5 and Netflix ratings are integer-valued from 1 to 5. Given the variance of the ratings —which is about 1—, the difference between both sorts of ratings is not significant. In the following we treat the ratings as continuous Gaussian random variables with homoscedastic variance.

[2] One difference is that all movies included in the Nextflix dataset have been rated by at least fifty users, as can be observed on the rightmost subplot of Fig. 1.



(a) MovieLens1M (b) MovieLens10M (c) Netflix

Figure 1: Scatterplot of average movie rating as function of the base ten logarithm of the number of views. The red line indicates the fitted regression curve. From left to the right: MovieLens 1M, MovieLens 10M and Netflix.

of MovieLens 10M have been rated by less than 50 users, despite the fact that the most popular movie has been rated by about half of the users. When one wants to go beyond recommending the highest rated items based on the whole population, the explicit or implicit use of subpopulations of users will necessitate a proper handling or small samples, even when the complete database is very large.

In the rest of this section, our objective is twofold. First, we make the previous comments more formal by providing a quantitative measure of the association between the popularity and the rating. Next, our aim is to do so in a way which can be exploited in a computationally efficient manner to improve the estimation of movie rating based on small populations of users. This objective will be addressed by use of linear regression.

Assuming the population can be considered as homogeneous, we define the observations as the pairs $(X_t, Y_t)_{t=1,..,n}$, where $X_t \in \{1, \dots K\}$ is the selected movie and $Y_t \in \mathbb{R}$ denotes the rating of the movie. n and K refer, respectively, to the total number of ratings and to the number of rated items. We further assume that pairs can be considered as independent and identically distributed so that the statistical model is parameterized by

$$\begin{cases} \theta_k = \mathrm{E}(Y_t | X_t = k) & \text{(expected item rating)} \\ \lambda_k = \mathrm{P}(X_t = k) & \text{(item selection probability)} \end{cases} \tag{1}$$

for $k = 1, \dots, K$. Sufficient statistics for this model are given by

$$N_k = \sum_{t=1}^{n} \mathbf{1}_{\{X_t = k\}}$$

and

$$S_k = \sum_{t=1}^{n} Y_t \mathbf{1}_{\{X_t = k\}}$$

which are, respectively, the number of times item k has been rated as well as the cumulated sum of its ratings.

The subplots of Fig. 1 display S_k/N_k as a function of $\log_{10}(N_k)$, for all items $k = 1, \dots, K$. As noted earlier, both the x- and y- values of this scatterplot correspond to statistics computed from data and should thus be considered as noisy, which is clearly visible in the left-hand part of each subplot. To account for this fact we use weighted Total Least-Squares (TLS) – or Deming – regression to fit a symmetrized form of the linear regression curve to the scatterplot. More precisely, denoting by $x_k = \ln(N_k/n)$ and $y_k = S_k/N_k$ for $k = 1, \dots, K$ we fit (a, b) by minimizing

$$\sum_{k=1}^{K} (x_k - \hat{x}_k)^2 / v_k + (y_k - \hat{y}_k)^2 / w_k \tag{2}$$

where $(\hat{x}_k, \hat{y}_k)$ is the orthogonal projection of (x_k, y_k) onto the straight line $y = ax + b$. Standard asymptotic statistical arguments show that, as n tends to infinity,

$$\sqrt{N_k}(x_k - \ln \lambda_k) \Rightarrow \mathcal{N}(0, 1 - \lambda_k)$$
$$\sqrt{N_k}(y_k - \theta_k) \Rightarrow \mathcal{N}(0, \sigma^2)$$

where $\Rightarrow$ corresponds to convergence in distribution, $\mathcal{N}(\mu, v)$ denote the Gaussian distribution with mean μ and variance v, and, σ^2 is the common variance of the ratings, which can be estimated from the data. In light of these results and given the fact that most of the movies have a selection probability λ_k that is much smaller than 1, we used as weights

$$v_k = 1/N_k \quad \text{and} \quad w_k = \sigma^2/N_k$$

The main effect of the weights v_k and w_k in (2) is to focus the estimation on the most popular movies, whose average ratings should be considered as being more precisely estimated. Note that the use of TLS also makes the problem symmetric and one would obtain the same linear fit by permuting the data associated to the x- and y- axis (which is of course not true for standard linear regression which assumes that the data on the x-axis is observed without noise). The weighted TLS regression

Data	MovieLens 1M	MovieLens 10M	Netflix
full dataset	0.36	0.16	0.09
random subsets of size 100	-	0.27	0.15

Table 1: Slope a estimated by weighted TLS on the three datasets MovieLens 1M, MovieLens 10M and Netflix (see Fig. 1) and median slopes estimated on random subsets of MovieLens 10M and Netflix (see Fig. 2).

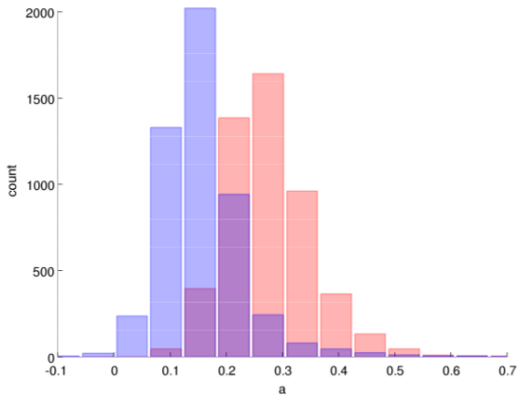


Figure 2: Histogram of slopes a estimated on 5,000 independent random subsets of size 100; for MovieLens 10M (red) and Netflix (blue).

estimate of a and b is obtained using the implementation of [12].

The first row of Table 2 reports the values of a estimated by the above method for the three datasets. Corresponding regression lines are shown in red on the three subplots of Fig. 1 (note that the x-axis is there $\log_{10}(N_k)$ to make the interpretation of the values easier). In all three cases, one obtains a significantly positive slope a. When fitting the regression model on whole datasets it is also observed that the estimated slope decays with the size of the dataset. This observation can be related to the fact that as the size of the dataset increases, the maximal number of ratings obtained by the most popular movies increases proportionally while the rating scale stays unchanged.

This being said, even in very large datasets, recommendation methods will explicitly (neighborhood-based) or implicitly (matrix factorization methods) define sub-populations of reduced size that can be considered as homogeneous. To illustrate the effect of considering sub-populations we randomly drew $5,000$ random sub-populations from both the MovieLens 10M and the Netflix datasets. On each of them we fitted a regression model using weighted TLS. The size of each population was chosen to consist of the ratings of 100 users chosen at random. The selections of these users typically correspond to subsets of 3.5k (MovieLens 10M) or 4.9k (Netflix) movies, with number of ratings in the range 1–50, where about 43% of the movies in the selection have a single rating. The values of a fitted on these sub-populations are displayed as histograms on Figure 2 and the median value of each histogram is reported in the second row of Table 2. This experiment shows that values of a appropriate for sub-populations are higher and usually in the range 0.1–0.35.

To understand the implication of these numbers, consider again our running example: if $a = 0.25$, it means that if we know that movie A has been viewed by n_A users and movie B by n_B users, where n_B is twice as large as n_A, and in the absence of any other data, we should expect the average rating of B to be $0.25 \ln(2) \approx 0.17$ higher than that of B. This information is non-negligible, corresponding to roughly one sixth of the standard deviation σ of the ratings. In Section 5, we will show that knowledge of this selection bias can indeed be leveraged to improve the prediction of ratings in realistic scenarios.

3 Using the Selection Bias as a Regularizer

In this section, we derive an estimation criterion that corresponds to a regularized likelihood estimator, where the regularization accounts for the selection bias. We show that optimizing this criterion, which is both smooth and convex, can be done efficiently using standard numerical optimization tools.

3.1 Variational Model

As indicated before, we model the ratings of movie k by a Gaussian distribution

$$p(Y|X = k; \theta) \sim \mathcal{N}(\theta_k; \sigma^2)$$

where σ^2 is a fixed variance for all movies and $\theta = (\theta_k)_{k=1,\dots,K}$ is the vector of expected movie ratings. This is clearly not the only option and the method could also be applied using the logit link function, as in [13, 14], if we were given binary ("like/dislike") ratings.

For, the selection probabilities $\lambda = (\lambda_k)_{k=1,\dots,K}$ it is important to consider the logistic form of the multinomial distribution parameterized by a vector β:

$$\lambda_k = \frac{e^{\beta_k}}{\sum_{j=1}^{K} e^{\beta_j}}$$

Up to a shift, β_k is homogeneous to $\ln \lambda_k$ and hence to the quantity displayed on the x-axis of Fig. 1. Note that the vector β itself is only identifiable up to a shift as replacing all β_k by $\beta_k + \delta$ would leave the vector λ of probabilities unchanged due to the logistic normalization term. This lack of identifiability is not a problem

as estimating β is not required and we will treat β as a so-called "nuisance parameter", optimizing it over all possible values.

Our model of the selection bias thus relies on the assumption that β —the log-probability of selecting each item (up to a constant)— must be close to $a\theta + b$ where a and b are global parameters that quantify the selection bias, estimated following the method exposed in Section 2. Due to the non-identifiability of β, we can disregard the intercept b and let β be determined in the shift direction by the value of θ.

Thus, the problem of estimating θ boils down to jointly minimizing over θ and β the following cost function :

$$\begin{aligned} f\left((X_t, Y_t)_{t=1}^n; \theta, \beta\right) = & L_1\left((Y_t|X_t)_{t=1}^n; \theta\right) \\ & + L_2\left((X_t)_{t=1}^n; \beta\right) \\ & + r\|\theta - a\beta\|_2^2 \end{aligned} \tag{3}$$

where

1. L_1 is the negative conditional log-likelihood of the Gaussian model of the observed ratings;
2. L_2 is the negative log-likelihood of the marginal distribution of the selections (X_t);
3. the last term is the regularization that constrains the ratings θ to stay close to $a\beta$, the log-probability of selection scaled by a.

The parameter $r > 0$ controls the influence of the regularization term and will typically be set using cross-validation on the training data.

3.2 Inference Algorithm

We first rewrite L_1 and L_2 :

$$L_1 = \sum_{k=1}^{K} \sum_{t=1}^{n} \mathbf{1}_{\{X_t = k\}} \frac{(Y_t - \theta_k)^2}{2\sigma^2}$$

$$L_2 = -\sum_{k=1}^{K} \sum_{t=1}^{n} \mathbf{1}_{\{X_t = j\}} \beta_k + n \log\left(\sum_{j=1}^{K} e^{\beta_j}\right)$$

Using the notations N_k and S_k introduced in Section 2, one obtains

$$L_1 = \sum_{k=1}^{K} \frac{\theta_k^2}{2\sigma^2} N_k - \frac{\theta_k}{\sigma^2} S_k + C$$

and

$$L_2 = n \log\left(\sum_{j=1}^{K} e^{\beta_j}\right) - \sum_{k=1}^{K} N_k \beta_k$$

C being a constant that does not depends on θ.

The gradient of the objective function f is the concatenation of the gradients with respect to θ and β:

$$\nabla f = \begin{bmatrix} \nabla_\theta f \\ \nabla_\beta f \end{bmatrix} \in \mathbb{R}^{2K}$$

where $\nabla_\theta f(k) = \frac{\theta_k}{\sigma^2} N_k - \frac{S_k}{\sigma^2} + 2r(\theta_k - a\beta_k)$ and $\nabla_\beta f(k) = -N_k + n\frac{e^{\beta_k}}{\sum_j e^{\beta_j}} - 2ra(\theta - a\beta)$.

The Hessian has the following block structure

$$Hf = \begin{bmatrix} H_{\theta\theta} & H_{\theta\beta} \\ H_{\beta\theta} & H_{\beta\beta} \end{bmatrix}$$

where $H_{\theta\theta} = diag\left(N_k/\sigma^2 + 2r, k = 1...K\right)$, $H_{\theta\beta} = H_{\beta\theta} = diag(-2ra)$ and

$$[H_{\beta\beta}]_{ij} = \begin{cases} n\lambda_i(1-\lambda_i) + 2ra^2 & \text{if } i = j \\ -n\lambda_i\lambda_j & \text{otherwise} \end{cases}$$

Proposition 1. *Assuming that all counts $(N_k)_{k=1,\dots,K}$ are strictly positive, the criterion*

$$f\left((X_t, Y_t)_{t=1}^N | \theta, \beta\right)$$

is strictly convex with respect to (θ, β).

Proof. Assuming $N_k > 0$, L_1 and L_2 are known to be strictly convex wrt. to, respectively, θ and β up to the already mentioned identifiability issue for β (the Hessian of L_2 has $\beta = (1, \dots, 1)^T$ as null direction).

For the regularization term, the function

$$(\theta, \beta) \in \mathbb{R}^K \times \mathbb{R}^K \mapsto \|\theta - a\beta\|^2$$

being separable in k, it is sufficient to consider the case where $K = 1$, i.e when $(\theta, \beta) \in \mathbb{R}^2$. In that case, the Hessian of the regularization term reduces to the following 2 by 2 matrix:

$$H(\theta, \beta) = \begin{bmatrix} 2 & -2a \\ -2a & 2a^2 \end{bmatrix}$$

The eigenvalues of this matrix are 0 and $2(a^2 + 1)$ and the associated eigenvectors are respectively $(a, 1)$ and $(-a, 1)$. Hence the Hessian of $\|\theta - a\beta\|^2$ is a positive matrix. Its K null directions (vectors of the form $\theta = (0, \dots, 0, a, 0, \dots, 0), \beta = (0, \dots, 0, 1, 0, \dots, 0)$) do not span the null direction of L_2 and hence Hf is positive definite. □

The parameter inference can thus be performed using fast converging algorithms like Newton-Raphson. In most scenarios however, the large size of the Hessian matrix —equal to twice the number of items— makes its storage and inversion cumbersome. For the experiments reported in the following we thus used open source implementations of the Limited-memory BFGS (L-BFGS) approach that yields comparable performance using a very small memory footprint (typically of the order of ten times the number of items in our case).

4 Validation: experiments on simulated data

To illustrate the behavior of the method, we start by considering a small-scale simulated scenario in which the data is generated using a probabilistic model related to the variational criterion proposed in (3). We first show that in this situation, the proposed approach — referred to as SB (for Selection Bias) in the following— is indeed able to recover the true mean rating parameter more efficiently than the least squares (LS) estimator that estimates θ_k directly by the empirical average S_k/N_k[3]. We also discuss the robustness of the approach with respect to the parameters a, r, as well as, the size n of the sample.

For simulating the data, we select an arbitrary vector of mean ratings θ^* of size K and generate the associated vectors of logistic parameters according to $\beta^* \sim \mathcal{N}(\theta^*/a, \sigma_b)$. The data is then simulated according to

$$X_t \sim \mathcal{M}ult(\lambda^*)$$
$$Y_t | X_t = k \sim \mathcal{N}(\theta_k^*; \sigma^2)$$

for $t = 1, \ldots, n$, where $\lambda_k^* = e^{\beta_k^*}/(\sum_{j=1}^K e^{\beta_j^*})$.

The parameters of the simulation are selected to be somewhat comparable to the observations made on real data in Section 2: values of θ_k in the range 1–5, $a = 0.35$, $\sigma = 1$. The value of σ_b is set to 1. For illustration purpose we use a small value of K, $K = 3$, varying the number n of ratings between 20 and 2,000. Note that due to the relatively high values of a and of the ratings spread (see Fig. 3 below) the best rated, and hence most popular, of the three items is typically forty times more frequent than the lowest rated one. Hence, the value of N_k for the lowest rated item is usually rather small, in the range between 1 to 50, depending on the value of n[4].

4.1 Recovery of ratings

Figure 3, compares the results obtained by the Selection Bias (SB) and Least Squares (LS) estimators for $n = 2,000$ using 200 Monte Carlo replications of the data $(X_t, Y_t)_{t=1,\ldots,n}$. It is observed that the value corresponding to the item that is simultaneously worst rated and least selected (corresponding to $k = 1$) is slightly under-estimated by the SB estimator compared to LS, with a standard deviation of the estimator that is also reduced. For the third item that is both highly rated and very frequent the difference between both estimators becomes to be negligible. Taking into account the selection bias in the SB estimator thus produces a downward bias and reduced variability of the estimated mean rating for infrequent items.

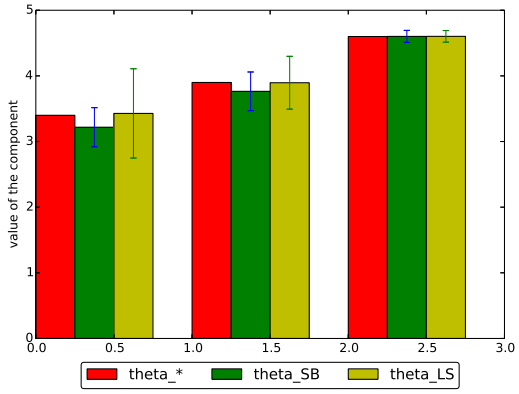


Figure 3: Recovery of the parameter θ: comparison of the Selection Bias (SB) and the Least Squares (LS) estimators. Red: true value θ^* for $k = 1, \ldots, 3$; Dark green: mean value estimated by SB; Light green: mean value estimated by LS. The vertical whiskers represent the standard deviation of the estimates..

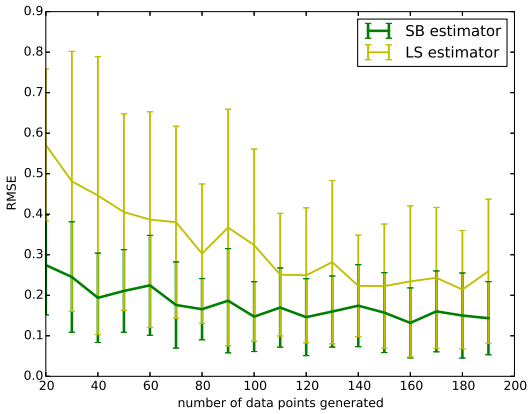


Figure 4: RMSE for SB and LS as a function of n, with corresponding error bars.

4.2 Robustness in small samples

The effect observed on Fig. 3 is all the more pronounced that the sample size n is small. To illustrate this fact, Figure 4 plots the Root Mean Square Error (RMSE) to the true value θ^*, when n increases from 20 to 200, computed from 50 Monte Carlo replications of the data. The RMSE is the square root of the average value of $\|\hat{\theta} - \theta^*\|^2$, where $\hat{\theta}$ denotes the estimated value of θ. Fig 4 confirms that the RMSE of the SB estimator is always smaller than that of LS and that the gap between

[3]Note that the LS estimator may also be interpreted as the solution of (3) when the regularization parameter r is set to zero.

[4]To ensure that the minimizer of (3) is uniquely defined, only the simulations for which all N_k, for $k = 1, \ldots, 3$, are strictly positive are retained.

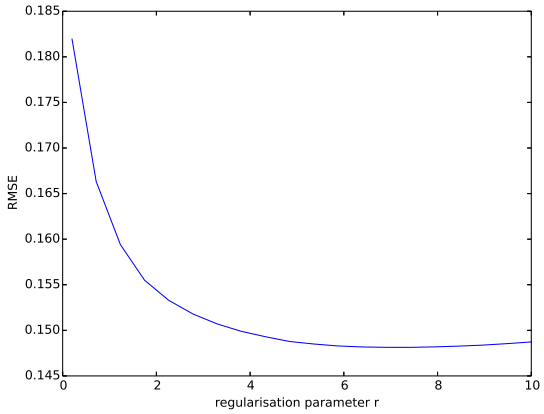


Figure 5: RMSE as a function of the regularization parameter r.

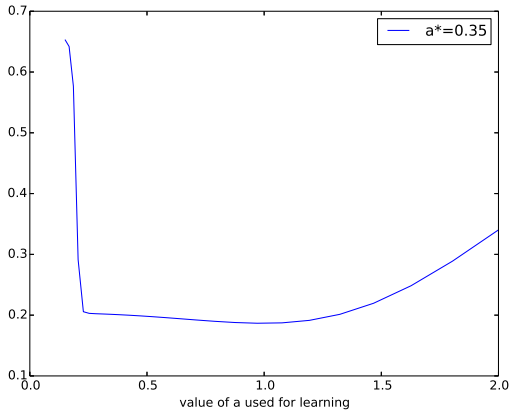


Figure 6: RMSE as a function of the parameter a used in the objective function; the parameter value used to simulate the data is set to $a^* = 0.35$.

them increases for small sample sizes (when n is small). Note that the variability of the SB estimator is also much reduced compared to that of LS.

4.3 Influence of r

The value selected for r may affect the quality of the results. Figure 5, shows the RMSE obtained when $n = 2,000$ for different values of r varying between 0.2 and 10. The curve is averaged over 200 Monte Carlo replications. Here the optimal value of r is $r = 7$, which is rather high as the simulation parameters almost satisfy $\beta^* = a^*\theta^*$, up to the Gaussian perturbation of variance σ_b^2. Most importantly, the curve displayed on Fig. 5 is very smooth around its minimum with a small curvature, meaning that values of r between 5 and 10 yield, in this case, a performance comparable to that corresponding to the optimal choice of r.

In Section 5, r will be set by searching for the minimum value of the RMSE on a validation subset corresponding to a small number of users.

4.4 Influence of a

The previous experiments have been carried out in the idealized setting where the parameter a that controls the generation of the data is known and used for the inference. In realistic scenarios, a will be known approximately only and it is not advisable to try to estimate a together with the other parameters, in light of the variability observed on Fig. 2. Figure 6 shows the RMSE when the parameter a used in the objective function (3) differs from the value a^* used for simulating the data, which is here fixed to $a^* = 0.35$. It is observed that the RMSE barely varies for $a \in (0.23, 1.2)$, indicating that the SB estimator is very robust to the overestimation of the slope a and only requires that it be set high enough to perform well.

In the experiments to be described in the next section, we also observed that the results were very robust to the choice of a, with values in the range 0.25–0.5 yielding unnoticeable changes in overall performance.

5 Experiments On Real Data Using Neighborhood-based Collaborative Filtering

In this section, we describe recommendation experiments carried out on the MovieLens dataset. The inference algorithm described in the previous section is used to estimate the ratings of films selected by subpopulations of users. Each sub-population corresponds to the neighborhood of the user for which one wants to make recommendations. We first discuss the choice of the similarity measure used to define the neighborhoods.

5.1 Neighborhood construction

The penalization scheme introduced in (3) is a generic tool designed to improve the estimation of ratings from small samples by taking into account the selection bias. In this section, we describe the simplest way in which this method can be used in the context of recommendation. We consider a standard neighborhood-based collaborative filtering approach in which (3) will be used only to estimate the movie ratings from the subpopulation of users that belong to each neighborhood. The baseline approach usually considered in the literature consists in using the empirical averages of the ratings in the sub-population, that is, LS (Least Squares) following the terminology of Section 4. To allow for a meaningful comparison between the proposed estima-

tor (termed SB) and LS, we will use the exact same algorithm to define the relevant sub-population of users that belong to the neighborhood of the user for which we want to discover new relevant movies.

For the baseline to be significant we will use a state-of-the-art approach for defining neighborhoods based on data-driven features of reduced dimension. Dimensionality reduction is a popular method in recommendation. Among the numerous algorithms that have been proposed to perform matrix factorization, the Singular Value Decomposition (SVD) was shown quite early to provide good performance (see for example [4]). For the experiments, we used an incremental implementation of SVD for the Julia language that can handle large sparse data matrices [5]. Considering the MovieLens 10M dataset, we first split randomly the data into a training set containing 75 % of the ratings of each user and a test set with the remaining 25 %. A rank-25 SVD of the rating matrix (considering unobserved values as zeroes) corresponding to training data was computed so as to determine a representation of each user as a vector of features in a 25-dimensional space.

To define the neighborhood of a user whose feature vector is denoted by u, we used the cosine similarity defined, for another vector v, by $c = \langle u, v\rangle / \|u\|\|v\|$. The size of the neighborhoods was set to 100 and hence the first 100 vectors v with highest cosine similarity with u are included in the neighborhood of u. The typical size of these neighborhoods is comparable to that of the random subsets considered in Section 2, that is, K (number of movies viewed in the neighborhood) and n (number of ratings) of the order of a few thousands.

Note that it would be very easy to use the similarity c_t corresponding to each item as a weight in (3): simply redefining N_k and S_k as

$$N_k = \sum_{t=1}^{n} c_t \mathbf{1}_{\{X_t=k\}}$$
$$S_k = \sum_{t=1}^{n} c_t Y_t \times \mathbf{1}_{\{X_t=k\}}$$

is equivalent to weighting each observation (X_t, Y_t) in the likelihoods L_1 and L_2 by c_t rather than 1. For $r = 0$, the solution of (3) when using this weighting will be the similarity-weighed average of each item's ratings rather than the simple average. In our case however, the results obtained with these similarity-weighted versions of the SB and LS estimators were not significantly different from those obtained with the basic (unweighted) versions and are not reported here.

[5]This library can be found at https://github.com/aaw/IncrementalSVD.jl

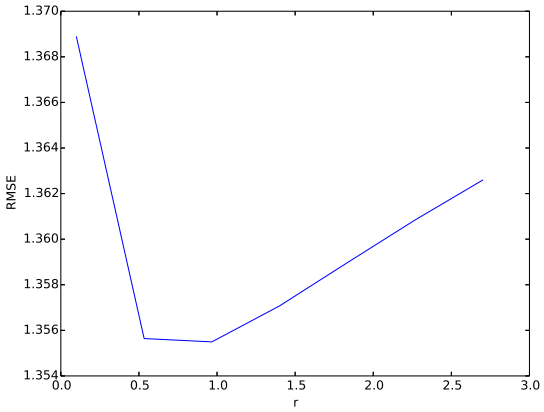


Figure 7: RMSE as a function of r, computed on the first 100 users of the base.

5.2 Choice of the parameters

The slope parameter a was set to the value estimated in Section 2 on random subsets of 100 users and was kept fixed through the experiments.

For the regularization parameter r, some tuning is necessary as observed in Section 4.3. Here again, the value of r is fixed globally, using a common value for all neighborhoods, as tuning r on small populations is definitely not advisable and would require validation data for each user. For that purpose, we conducted a preliminary experiment on a subset of 100 users of the base and computed the averaged RMSE (see below) for different values of r between 0.1 and 2.5. The results are displayed on Figure 7. Similarly to the graph of Fig. 5 computed on simulated data, we observe that the quality of the recommendations improves whenever $r > 0$, with an optimum about $r = 1$ which was used in the following.

5.3 Evaluation Metrics

To evaluate the results, we used various metrics focusing on different aspects of the estimation.

The first criterion, RMSE, is aimed at quantifying the calibration of the rating estimates provided by the algorithms. The RMSE, is the classical metric that was used in the Netflix Challenge. If $\mathcal{T}$ denotes the set of indices featured in the test set for a given user, we define

$$\text{user RMSE} = \sqrt{\frac{1}{|\mathcal{T}|}\sum_{t\in\mathcal{T}}\left(Y_t - \hat{\theta}_{X_t}\right)^2}$$

where $\hat{\theta}$ denotes the rating estimates determined on the training data for this user[6]. The RMSE is the average

[6]For a movie k that was not selected in the user neighborhood but that is present in the user test set, we set by convention $\hat{\theta}_l = 3.5$, which corresponds to the empirical average of all ratings

Table 2: Results of the experiments on the MovieLens 10M dataset.

	RMSE	P@N3	P@N14	P@τ4
$\hat{\theta}(SB)$	**0.923**	**0.183**	**0.125**	**0.0332**
$\hat{\theta}(LS)$	0.952	0.0022	0.0028	0.0211
Popularity Ranking	-	0.239	0.160	-

of the user-level RMSE for all users. It should be noted that, in contrast to the criterion used in Section 4 that weighted all items equally, the RMSE as defined above does give more weights to the error corresponding to movies that appear frequently.

In addition, the RMSE gives as much importance to accuracy in predicting low ratings that it does for high ratings, whereas the latter is arguably more relevant in the perspective of recommendation. For this reason, we also consider the precision associated with the recommendation of a few number of items that are believed to be relevant. Relevant items are defined as the movies actually selected by the user in the test set *and that were rated 4 or above* (that is, 4, 4.5 or 5 for the MovieLens datasets).

The first measure is the standard Precision-at-N (denoted P@N) that assumes that only N items are to be recommended and counts the number of relevant items (also called true positives) in those N recommended items:

$$\text{P@N} = |\text{relevant items}|/N$$

In this case, it is natural to consider that the N movies recommended by a method are those that have the highest estimated ratings. However, P@N is fundamentally a ranking measure and we will see below that due to the selection bias it is optimized by a very simple heuristic that does not even rely on estimating the movie ratings.

To mitigate this observation, we also consider P@τ in which the set of relevant items is determined by considering the movies for which the estimated rating is above the threshold τ, that is,

$$\text{P@}\tau = |\text{relevant items}|/|\text{items with est. rating} \geq \tau|$$

Although, this second way of proceeding does not explicitly control the size of the recommendation set that corresponds to a given value of τ, it is appropriate to measure the accuracy in predicting, in a calibrated way, high values of the ratings.

5.4 Results

We present in this section the results obtained on the MovieLens 10M dataset. We computed RMSE, P@N and P@τ by selecting randomly and averaging results over 1,000 users.

The first column of Table 2 shows that the SB estimator improves the RMSE compared to LS. This improvement is significant, confirming the good behavior of LS for rating estimation. The order of magnitude of the improvement is limited but this is mainly due to the weighting by the popularity of movies inherent to the RMSE computation. For less frequent movies, the improvement brought by SB is indeed major as will be shown below. The remaining columns of Table 2 report the performance in term of the P@N and P@τ metrics defined in the previous section. The values selected for N correspond to two realistic use cases that can be of interest in movie recommendation: suggesting a top-3 short list (N=3) or building a recommendation page on a website containing a human-sized list (N=14). For P@τ, $\tau = 4$ was selected in light of the actual threshold used to determine relevant items in the test set[7]. These results show that, when it comes to identifying highly rated items, the SB estimator significantly outperforms the standard empirical average (or LS) estimator.

Figure 8 gives more details by displaying the results obtained for the P@N metric, for values of N between 3 and 30. It is important to keep in mind that the P@N metric being a ranking criterion it does not measure the accuracy in evaluating the ratings but rather the ability to produce a correct ordering of the movies. The general shape of the performance curve for the SB estimator (blue curve) in Fig. 8 suggests that it succeeds in putting at the top of the list the most relevant items, with a precision that decreases as N increases. In contrast, the red curve that corresponds to the performance of the LS estimator shows that it largely fails when it comes to ranking items. The fact that as N grows, the P@N metric increases (slightly) with N for the LS estimator suggests that the failure of LS comes from the fact that it can attribute high ratings to irrelevant movies. As discussed in Section 2, a significant fraction of movies (almost half of them) rated in each neighborhood has been rated only once. The LS estimation for these "hapax" items is extremely noisy, being based on a single occurrence: it suffices that one of these be rated at 5 (the maximal note) to perturb the highest rank of the recommendation list. This interpretation is confirmed by the green curve of Fig. 8 that corresponds to the performance of the LS estimator, when restricted to the movies that were at least select twice in the neighborhood. By raising the threshold value (above 2) the impact of unreliable ratings could be further reduced and the ranking performance of the LS estimator improved, at the price of a reduced diversity of the recommendations. An alternative would be to use a Bayesian mean estimate, as in [8], to shrink the estimates towards the global mean rating for scarcely

[7]Note however that due to the fact that the actual observed ratings are half integers, the value of the relevance threshold is not precisely defined.

observed items. It is remarkable that the SB estimator does not necessitate any adjustment of this sort: as discussed about Fig. 3, the presence of the regularization term creates a downward bias for ratings based on few observations, making it highly unlikely that these unreliable ratings appear at the top of the ranking list.

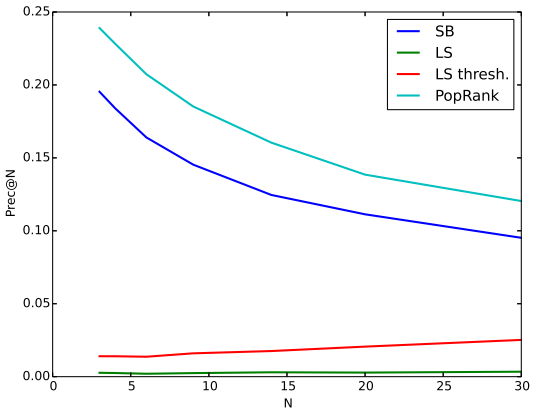


Figure 8: Precision-at-N for N varying in 3-30. The results are averaged over 1,000 users randomly selected among the ML10M database.

Interestingly, in terms of the P@N metric, the SB estimator is dominated by the simple heuristic (light blue curve in Fig. 8) that ranks the movies in the neighborhood according to their popularity and recommend the N most popular ones. The fact that it is possible to recommend the highly rated movies, without even using the rating data (except for the definition of the features used to build the neighborhoods) is a clear illustration of the selection bias phenomenon. This "ranking by popularity" approach is also naturally immune against the variability due to movies with a small number of ratings. This being said, this strategy is not calibrated and recommends items whose value is not clearly defined. It is also likely that, in terms of the diversity of the recommendations, always recommending the most popular items in each neighborhood is not the optimal approach.

The alternative consists in measuring the precision at a given threshold τ, as shown in Figure 9: for thresholds τ between 3 and 4.5, both the LS and the SB estimators were used to create lists of recommended items whose estimated rating exceeded τ. The drawback of this approach is that the size of the recommendation list is not explicitly controlled. In particular, when interpreting the curves on Fig. 9 it is important to keep in mind that the size of the recommendation lists corresponding to the same value of τ may in fact be different for the two methods (SB and LS) under consideration.

It is observed on Figure 9 that up to $\tau = 3.5$, which correspond to the overall average rating, both estimators have comparably low performance as they recommended about half of the items present in the training set for each neighborhood. When the threshold grows up to 4, the lists of recommendations for both SB and LS become more relevant as shown by the increase of the P@τ metric. For the LS estimator however, the precision values are decreasing for thresholds τ above 4, showing that a significant fraction of the highly rated estimates is in fact strongly contaminated by unreliable values. In contrast, the precision of the SB estimator keeps improving as τ increases, showing once again that the highly rated estimates are much more reliable with the SB approach.

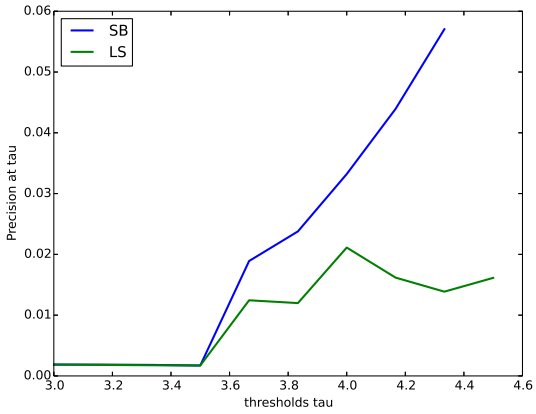


Figure 9: Precision-at-τ for τ varying in 3-4.5. The results are averaged over 1,000 users randomly selected among the ML10M database.

6 Conclusion

In this paper, we introduced a model for the link between the probability of selecting an item and its underlying rating. The corresponding optimization-based estimator of the underlying rating effectively uses the two available pieces of information about each item: the empirical frequencies of selection and the empirical rating averages. The experiments performed on simulated data showed that in the presence of a selection bias the proposed estimator provides more reliable estimate of the underlying rating of highly-rated items. Finally, the approach was used for collaborative filtering on the MovieLens data showing a large improvement in terms of relevance of the recommendation.

We believe that the proposed approach can be incorporated in other types of recommendation algorithms. A first idea would consist in using the selection bias regularization term, or a variant of it, in matrix completion methods based on a variational criterion [15], including the cases where a different noise model is considered [13, 14]. Another idea would be to incorporate

the selection bias in the prior specification for methods based on Bayesian modeling [7, 16].

References

[1] D. Goldberg, D. Nichols, B. M. Oki, and D. Terry, "Using collaborative filtering to weave an information tapestry," *Communications of the ACM*, vol. 35, no. 12, pp. 61–70, 1992.

[2] P. Resnick and H. R. Varian, "Recommender systems," *Communications of the ACM*, vol. 40, no. 3, pp. 56–58, 1997.

[3] J. Bennett and S. Lanning, "The netflix prize," in *Proceedings of KDD cup and workshop*, vol. 2007, 2007, p. 35.

[4] R. M. Bell and Y. Koren, "Lessons from the netflix prize challenge," *ACM SIGKDD Explorations Newsletter*, vol. 9, no. 2, pp. 75–79, 2007.

[5] J. L. Herlocker, J. A. Konstan, A. Borchers, and J. Riedl, "An algorithmic framework for performing collaborative filtering," in *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, 1999, pp. 230–237.

[6] J. Herlocker, J. A. Konstan, and J. Riedl, "An empirical analysis of design choices in neighborhood-based collaborative filtering algorithms," *Information retrieval*, vol. 5, no. 4, pp. 287–310, 2002.

[7] T. T. Mai and P. Alquier, "A bayesian approach for matrix completion: Optimal rate under general sampling distribution," *Electronic Journal of Statistics*, vol. 9, pp. 823–841, 2015.

[8] R. M. Bell and Y. Koren, "Scalable collaborative filtering with jointly derived neighborhood interpolation weights," in *Proceedings of the Seventh IEEE International Conference on Data Mining (ICDM)*, 2007, pp. 43–52.

[9] B. M. Marlin, "Modeling user rating profiles for collaborative filtering," in *Advances in neural information processing systems (NIPS)*, 2003.

[10] R. J. A. Little and D. B. Rubin, *Statistical analysis with missing data*, ser. Wiley Series in Probability and Mathematical Statistics: Applied Probability and Statistics. John Wiley & Sons, Inc., New York, 1987.

[11] B. Marlin, R. S. Zemel, S. Roweis, and M. Slaney, "Collaborative filtering and the missing at random assumption," *arXiv preprint arXiv:1206.5267*, 2012.

[12] M. Krystek and M. Anton, "A weighted total least-squares algorithm for fitting a straight line," *Meas. Sci. Technol.*, vol. 18, no. 3438–3442, 2007.

[13] M. A. Davenport, Y. Plan, E. van den Berg, and M. Wootters, "1-bit matrix completion," *Information and Inference*, vol. 3, no. 3, pp. 189–223, 2014.

[14] J. Lafond, O. Klopp, E. Moulines, and J. Salmon, "Probabilistic low-rank matrix completion on finite alphabets," in *Advances in Neural Information Processing Systems (NIPS)*, 2014, pp. 1727–1735.

[15] E. Candés and Y. Plan, "Matrix completion with noise," *Proc. IEEE*, vol. 98, no. 6, pp. 925–936, 2010.

[16] P. Gopalan, F. J. Ruiz, R. Ranganath, and D. M. Blei, "Bayesian nonparametric poisson factorization for recommendation systems," in *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2014, pp. 275–283.