SVD-LDA: Topic Modeling for Full-Text Recommender Systems

Sergey Nikolenko[1,2,3](✉)

[1] Steklov Institute of Mathematics at St. Petersburg, St. Petersburg, Russia
sergey@logic.pdmi.ras.ru
[2] Laboratory for Internet Studies, National Research University – Higher School of Economics, St. Petersburg, Russia
[3] Kazan (Volga Region) Federal University, Kazan, Russia

Abstract. In recommender systems, matrix decompositions, in particular singular value decomposition (SVD), represent users and items as vectors of features and allow for additional terms in the decomposition to account for other available information. In text mining, topic modeling, in particular latent Dirichlet allocation (LDA), are designed to extract topical content of a large corpus of documents. In this work, we present a unified SVD-LDA model that aims to improve SVD-based recommendations for items with textual content with topic modeling of this content. We develop a training algorithm for SVD-LDA based on a first order approximation to Gibbs sampling and show significant improvements in recommendation quality.

1 Introduction

Modern recommender systems deal with items of very different nature, including images, videos, tagged items, goods and services, and texts; many of these items have some kind of meaningful content that can be used to improve recommendations. Therefore, one natural direction of research in recommender systems would be to mine the content of the items being recommended. This is especially relevant for the cold start problem: to recommend new content with no history of preferences it would be very useful to make first recommendations to users who prefer this kind of content. In practice, such models usually represent a modification of some classical collaborative filtering model, usually based either on similarity between users or items [1,2] or on matrix decompositions. Over the last decade, collaborative filtering, starting from the Netflix Prize Challenge and beyond, have been dominated by various matrix decomposition techniques, mainly singular value decomposition (SVD) and nonnegative matrix factorization (NMF) [3,4]. On the other hand, again over the last decade, topic modeling, starting from probabilistic latent semantic analysis [5] and continuing with its Bayesian version, latent Dirichlet allocation [6], has become the method of choice for understanding large text corpora. Topic modeling is basically a dimensionality reduction technique, in many aspects very similar to an SVD decomposition,

O. Pichardo Lagunas et al. (Eds.): MICAI 2015, Part II, LNAI 9414, pp. 67–79, 2015.
DOI: 10.1007/978-3-319-27101-9_5

where the word-document matrix is decomposed into the "product" of word-topic and topic-document matrices.

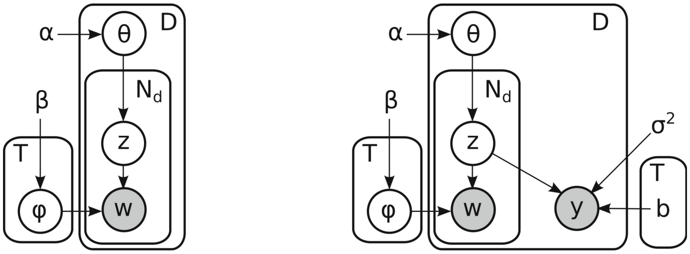
In this work, we combine the SVD and LDA decompositions into a single unified model that optimizes a joint likelihood function and thus infers topics that are especially useful for improving recommendations. We provide an inference algorithm based on Gibbs sampling. However, it turns out that straightforward Gibbs sampling would have prohibitive computational costs, as each iteration of Gibbs sampling would require iterating over all ratings in the recommender dataset. Therefore, we develop an approximate sampling scheme based on a first order approximation to Gibbs sampling. The resulting algorithm has the same complexity as the original LDA Gibbs sampling and provides both meaningful topics and improved recommendations. We also add user metadata (demographic features), showing that the resulting topic factors are meaningful and provide a snapshot of the corresponding demographic group's tastes.

The paper is organized as follows. In Sect. 2, we remind the basic facts about latent Dirichlet allocation and briefly survey relevant extensions; Sect. 3 does the same for singular value decomposition as applied to recommender systems. In Sect. 4, we introduce the novel SVD-LDA model, show the inference for Gibbs sampling in this model, and approximate it to make the sampling tractable. Section 5 shows practical evaluation on a large dataset of full-text recommended items, and Sect. 6 concludes the paper.

2 LDA and sLDA

2.1 Latent Dirichlet Allocation

We begin with the basic latent Dirichlet allocation (LDA) model that we extend in the next section. The graphical model of LDA [6,7] is shown on Fig. 1a. We assume that a corpus of D documents contains T topics expressed by W different words. Each document $d \in D$ is modeled as a discrete distribution $\theta^{(d)}$ on the set of topics: $p(z_w = j) = \theta^{(d)}$, where z is a discrete variable that defines the topic of each word $w \in d$. Each topic, in turn, corresponds to a multinomial distribution on words: $p(w \mid z_w = k) = \phi_w^{(k)}$. The model also introduces prior Dirichlet distributions with parameters α for the topic vectors θ, $\theta \sim \mathrm{Dir}(\alpha)$, and β for the word distributions ϕ, $\phi \sim \mathrm{Dir}(\beta)$. A document is generated word by word: for each word, we (1) sample the topic index k from distribution $\theta^{(d)}$; (2) sample the word w from distribution $\phi_w^{(k)}$. Inference in LDA is usually done via either variational approximations or Gibbs sampling; we use the latter since it is easy to generalize to further extensions. In the basic LDA model, Gibbs sampling reduces to the so-called *collapsed Gibbs sampling*, where θ and ϕ variables are integrated out, and z_w are iteratively resampled according to the following distribution: $p(z_w = t \mid \boldsymbol{z}_{-w}, \boldsymbol{w}, \alpha, \beta) \propto \frac{n_{-w,t}^{(d)}+\alpha}{\sum_{t' \in T}\left(n_{-w,t'}^{(d)}+\alpha\right)} \frac{n_{-w,t}^{(w)}+\beta}{\sum_{w' \in W}\left(n_{-w,t}^{(w')}+\beta\right)}$, where $n_{-w,t}^{(d)}$ is the number of words in document d chosen with topic t and $n_{-w,t}^{(w)}$ is the number of times word w has been generated from topic t apart from the current value z_w;



(a) (b)

Fig. 1. The (a) LDA and (b) sLDA graphical models.

both counters depend on the other variables $\boldsymbol{z}_{-w}$. Samples are then used to estimate model variables: $\theta_{d,t} = \frac{n^{(d)}_{-w,t}+\alpha}{\sum_{t'\in T}\left(n^{(d)}_{-w,t'}+\alpha\right)}$, $\phi_{w,t} = \frac{n^{(w)}_{-w,t}+\beta}{\sum_{w'\in W}\left(n^{(w')}_{-w,t}+\beta\right)}$, where $\phi_{w,t}$ denotes the probability to draw word w in topic t and $\theta_{d,t}$ is the probability to draw topic t for a word in document d.

After it was introduced in [6], the basic LDA model has been subject to many extensions, each presenting either a variational or a Gibbs sampling algorithm for a model that builds upon LDA to incorporate some additional information or additional presumed dependencies. In this work, we will further extend a specific extension of LDA named Supervised LDA.

2.2 Supervised LDA

The SVD-LDA model we present in Sect. 4 is, in a way, an extension of the Supervised LDA (sLDA) model [8]. In fact, in [8] the authors give recommender systems as an example, albeit more in the context of sentiment analysis: the authors predict the rating a user gives a movie based on the text of this user's review. The sLDA graphical model is shown on Fig. 1b. Each document is now augmented with a response variable y; in sLDA, y is drawn from a normal distribution centered around a linear combination of the document's topical distribution ($\bar{\boldsymbol{z}}$, average z variables in this document) with some unknown parameters $\boldsymbol{b}$, a that are also to be trained when learning the model: $y \sim \mathcal{N}(y \mid \boldsymbol{b}^\top \bar{\boldsymbol{z}} + a, \sigma^2)$.

The original work [8] presents an inference algorithm for sLDA based on variational approximations, but in this work we operate with Gibbs sampling which will be easier to extend to SVD-LDA later. Thus, we show an sLDA Gibbs sampling scheme. It differs from the original LDA in that the model likelihood gets another factor corresponding to the y variable: $p(y_d \mid \boldsymbol{z}, \boldsymbol{b}, \sigma^2) = \exp\left(-\frac{1}{2}\left(y_d - \boldsymbol{b}^\top \bar{\boldsymbol{z}} - a\right)^2\right)$, and the total likelihood is now

$$p(\boldsymbol{z} \mid \boldsymbol{w}, \boldsymbol{y}, \boldsymbol{b}, \alpha, \beta, \sigma^2) \propto \prod_d \frac{B(\boldsymbol{n}_d + \alpha)}{B(\alpha)} \prod_t \frac{B(\boldsymbol{n}_t + \beta)}{B(\beta)} \prod_d e^{-\frac{1}{2}\left(y_d - \boldsymbol{b}^\top \bar{\boldsymbol{z}}_d - a\right)^2}.$$

On each iteration of the sampling algorithm, we now first sample $\boldsymbol{z}$ for fixed $\boldsymbol{b}$ and then train $\boldsymbol{b}$ for fixed (sampled) $\boldsymbol{z}$. The sampling distribution of each z variable, according to the equation above, are $p(z_w = t \mid \boldsymbol{z}_{-w}, \boldsymbol{w}, \alpha, \beta) \propto q(z_w, t, \boldsymbol{z}_{-w}, \boldsymbol{w}, \alpha, \beta) e^{-\frac{1}{2}\left(y_d - \boldsymbol{b}^\top \bar{\boldsymbol{z}} - a\right)^2} = \frac{n^{(d)}_{-w,t}+\alpha}{\sum\limits_{t'}\left(n^{(d)}_{-w,t'}+\alpha\right)} \frac{n^{(w)}_{-w,t}+\beta}{\sum\limits_{w'}\left(n^{(w')}_{-w,t}+\beta\right)} e^{-\frac{1}{2}\left(y_d - \boldsymbol{b}^\top \bar{\boldsymbol{z}} - a\right)^2}$. The latter equation can be either used directly or further transformed by separating $\boldsymbol{z}_{-w}$ explicitly.

In what follows, we consider a recommender system based on likes and dislikes, where we will use the logistic sigmoid $\sigma(x) = 1/\left(1 + \exp(-x)\right)$ of a linear function to model the probability of a "like": $p = \sigma\left(\boldsymbol{b}^\top \bar{\boldsymbol{z}} + a\right)$. In this version of sLDA, the graphical model remains the same, only conditional probabilities change. The total likelihood is now $p(\boldsymbol{z} \mid \boldsymbol{w}, \boldsymbol{y}, \boldsymbol{b}, \alpha, \beta, \sigma^2) \propto$

$$\prod_d \frac{B(\boldsymbol{n}_d + \alpha)}{B(\alpha)} \prod_t \frac{B(\boldsymbol{n}_t + \beta)}{B(\beta)} \prod_d \prod_{x \in X_d} \sigma\left(\boldsymbol{b}^\top \bar{\boldsymbol{z}}_d + a\right)^{y_x} \left(1 - \sigma\left(\boldsymbol{b}^\top \bar{\boldsymbol{z}}_d + a\right)\right)^{1-y_x},$$

where X_d is the set of experiments (ratings) for document d, and y_x is the binary result of one such experiment. The sampling procedure also remains the same, except that now we train logistic regression with respect to $\boldsymbol{b}$, a for fixed $\boldsymbol{z}$ instead of linear regression, and the sampling probabilities for each z variable are now $p(z_w = t \mid \boldsymbol{z}_{-w}, \boldsymbol{w}, \alpha, \beta) \propto$

$$q(z_w, t, \boldsymbol{z}_{-w}, \boldsymbol{w}, \alpha, \beta) \prod_{x \in X_d} \left[\sigma\left(\boldsymbol{b}^\top \bar{\boldsymbol{z}}_d + a\right)\right]^{y_x} \left[1 - \sigma\left(\boldsymbol{b}^\top \bar{\boldsymbol{z}}_d + a\right)\right]^{1-y_x}$$

$$= \frac{n^{(d)}_{-w,t} + \alpha}{\sum_{t' \in T}\left(n^{(d)}_{-w,t'} + \alpha\right)} \frac{n^{(w)}_{-w,t} + \beta}{\sum_{w' \in W}\left(n^{(w')}_{-w,t} + \beta\right)} e^{s_d \log p_d + (|X_d| - s_d)\log(1-p_d)},$$

where s_d is the number of successful experiments among X_d, and $p_d = \frac{1}{1+e^{-\boldsymbol{b}^\top \bar{\boldsymbol{z}}_d - a}}$.

3 SVD in Recommender Systems

3.1 Basic SVD Model in Collaborative Filtering

Recommender systems usually rely on collaborative filtering which can be expressed with the "like like like" maxim: users are similar if they like similar items (movies, musical compositions, web pages etc.), objects are similar if similar users have liked them, and similar users will keep liking similar items in the future. Collaborative filtering is usually based either on nearest neighbors [1,9] or, more importantly for this work, on matrix decompositions. Matrix decompositions, such as SVD, are used to extract from the data and estimate latent factors that influence the rating a user assigns to an item. In collaborative filtering based on singular value decomposition (SVD), the rating is represented as a sum of baseline predictors for both user and item and the scalar product of user features and item features: $\hat{r}_{i,a} = \mu + b_i + b_a + q_a^\top p_i$, where μ is a general mean, b_i and b_a are the user and item baseline predictors respectively, p_i and q_a are the user and item feature vectors. In case of a linear scale of ratings $\hat{r}_{i,a}$ can be used directly as an estimate of $r_{i,a}$ with Gaussian noise. However, in our case we have binary ratings, likes and dislikes, so we adopt the logistic SVD model where the probability of a like is modeled as $p(\text{like}_{i,a}) = \sigma(\hat{r}_{i,a})$ for the logistic sigmoid $\sigma(x) = 1/(1+e^{-x})$. Such models can be trained with stochastic gradient descent (SGD) or alternating least squares (ALS). Thus, the rating matrix that has many unknown components (ratings to be predicted) undergoes a low-rank approximation: an $N \times M$ matrix for N users and M items is decomposed into a product of $N \times F$ and $F \times M$ matrices, where F is the number of features which is usually several orders of magnitude smaller than both N and M. There are many different variations of this model; see, e.g., [3] and other works.

3.2 Cold Start, Additional Information, and Content

A key issue of all recommender systems is the cold start problem: how do we recommend content to a new user with no history of rated items and how do we recommend a new item that has no history of being rated by users? Basic collaborative filtering works very well for users and items with sufficient statistics already accumulated, and if we have no additional information except for the matrix of ratings, there is little we can do to cope with cold start; lists of top rated items are usually recommended to new users.

However, real life recommender systems almost always have some additional information about their users and/or items. Such information is often well structured; for instance, a movie may come with its genre, director, release date etc. Structured additional information, where there is a closed and relatively small list of possible values (e.g., movie genres), and they are known for items in the dataset, can be directly incorporated in the SVD model: $\hat{r}_{i,a} = \mu + b_i + b_a + q_a^\top p_i + r_a^\top s_i$, where r_a represents the additional information about the items, and s_i are feature vectors (or perhaps one-dimensional predictors) for the additional information in question; s_i can be trained individually for each user, but this often leads to overfitting, so s_i are often shared among users in a cluster. The clustering can be based either on the ratings themselves or on additional information about the users, such as, for instance, demographic information from the user profile (age, gender, country etc.). In the latter case, the resulting features can be used for cold start recommendations: a new user who has filled out his/her profile can have recommendations that are suitable for his/her demographic group. Note that the training algorithms do not change at all, we simply introduce new additive terms in the model, so both partial derivatives are still easy to compute for SGD and the model still reduces to linear or logistic regression with respect to user or item features in case of ALS.

The situation becomes more complicated, however, when the additional information also has to be trained. In this work, we concentrate on the case when the items being recommended have textual content (in the dataset below they will be web pages), and the additional information that we want to use represents topical content of the items extracted with a topic model as discussed in Sect. 2. One could, of course, first train the LDA model separately and then use the topic distributions θ_a for each document a as additional information, adding new features l_i for each user or cluster of users and each topic: $\hat{r}_{i,a} = \mu + b_i + b_a + q_a^\top p_i + \theta_a^\top l_i$. This approach was formalized and further developed in the fLDA model [10], which is an extension of regression-based latent factor models (RLFM) [11]. We will develop a new unified model that trains LDA topics in such a way as to improve SVD recommendations, similar to how sLDA extracts topics that are relevant for its response variable.

4 SVD-LDA

In this section, we present the new SVD-LDA model that combines logistic SVD for modeling the probability of a like with additional terms based on LDA topics that are trained together with SVD. In Sect. 4.1, we begin with a Gibbs sampling scheme that proves to be too computationally intensive, so in Sect. 4.2 we present an approximation which makes it tractable but still useful.

4.1 SVD-LDA: Exact Sampling

In the SVD-LDA model, for recommendations we use an SVD model with additional predictors corresponding to how much a certain user or group of user likes the topics trained in the LDA model; since our dataset is binary (like-dislike), we use a logistic version of the SVD model:

$$p(\text{success}_{i,a}) = \sigma\left(\hat{r}_{i,a}\right) = \sigma\left(\mu + b_i + b_a + q_a^\top p_i + \theta_a^\top l_i\right),$$

where p_i may be absent in case of cold start, and l_i may be shared among groups (clusters) of users. The total likelihood of the dataset with ratings comprised of triples $D = \{(i, a, r)\}$ (user i rated item a as $r \in \{-1, 1\}$) is a product of the likelihood of each rating (assuming, as usual, that they are independent): $p(D \mid \mu, b_i, b_a, p_i, q_a, l_i, \theta_a) = \prod_D \sigma\left(\hat{r}_{i,a}\right)^{[r=1]} \left(1 - \sigma\left(\hat{r}_{i,a}\right)\right)^{[r=-1]}$, and the logarithm is

$$\ln p(D \mid \mu, b_i, b_a, p_i, q_a, l_i, \theta_a) = \sum_D \ln\left([r=-1] - \sigma\left(\hat{r}_{i,a}\right)\right),$$

where $[r = -1] = 1$ if $r = -1$ and $[r = -1] = 0$ otherwise, and θ_a is the vector of topics trained for document a in the LDA model, $\theta_a = \frac{1}{N_a}\sum_{w \in a} z_w$, where N_a is the length of document a. Sampling probabilities for each z variable now look like $p(z_w = t \mid \boldsymbol{z}_{-w}, \boldsymbol{w}, \alpha, \beta) \propto q(z_w, t, \boldsymbol{z}_{-w}, \boldsymbol{w}, \alpha, \beta) p(D \mid \mu, b_i, b_a, p_i, q_a, l_i, \theta_a^{w \to t})$

$$= \frac{n_{-w,t}^{(d)} + \alpha}{\sum_{t' \in T}\left(n_{-w,t'}^{(d)} + \alpha\right)} \frac{n_{-w,t}^{(w)} + \beta}{\sum_{w' \in W}\left(n_{-w,t}^{(w')} + \beta\right)} p(D \mid \mu, b_i, b_a, p_i, q_a, l_i, \theta_a^{w \to t})$$

$$= \frac{n_{-w,t}^{(d)} + \alpha}{\sum_{t' \in T}\left(n_{-w,t'}^{(d)} + \alpha\right)} \frac{n_{-w,t}^{(w)} + \beta}{\sum_{w' \in W}\left(n_{-w,t}^{(w')} + \beta\right)} e^{\sum_D \ln\left([r=-1] - \sigma\left(\hat{r}_{i,a}^{\text{SVD}} + l_i^\top \theta_a^{w \to t}\right)\right)},$$

where $\hat{r}^{\mathrm{SVD}}_{i,a} = \mu + b_i + b_a + q_a^\top p_i$, and $\theta_a^{w\to t}$ is the vector of topics for document a where topic t is substituted in place of z_w. We see that in the formula above, to compute the sampling distribution for a single z_w variable one has to take a sum over all ratings all users have provided for this document, and due to the presence of the sigmoid function one cannot cancel out terms and reduce the sum to updating counts. It is possible to store precomputed values of $\hat{r}^{\mathrm{SVD}}_{i,a}$ in memory, but it does not help because the z_w variables change during sampling, and when they do all values of $\sigma(\hat{r}^{\mathrm{SVD}}_{i,a} + l_i^\top \theta_a^{w\to t})$ also have to be recomputed for each rating from the database. We have developed an implementation of this sampling algorithm; as expected, it works well on toy examples but cannot run in any reasonable time on a real-world sized dataset.

4.2 SVD-LDA: First Order Approximation

To make the model feasible, we had to develop a simplified SVD-LDA training algorithm that could run reasonably fast on large datasets. For the purposes of this simplification, we used a first order approximation to the value of the log likelihood, decomposing it into a Taylor series at the point where log likelihood is zero. Such an approximation will be very bad far from zero, but note that this is the logarithm of a value proportional to a multinomial probability: large negative values will all be sufficiently close to zero after exponentiation as to not matter, and large positive values will all yield a dominating advantage over alternatives. The only values where we need the approximation to be relatively precise are exactly the values around zero, where there are several alternative topics with comparable and significant probabilities. To construct the approximation, we differentiate $\ln p(D \mid \mu, b_i, b_a, p_i, q_a, l_i, \theta_a)$ with respect to θ_a; it is convenient to use the fact that $\sigma'(x) = \sigma(x)(1 - \sigma(x))$, which means that $\frac{\partial \ln \sigma(x)}{\partial x} = 1 - \sigma(x)$, $\frac{\partial \ln(1-\sigma(x))}{\partial x} = -\sigma(x)$, so

$$\begin{aligned}\frac{\partial \ln p(D \mid l_i, \theta_a, \ldots)}{\partial \theta_a} &= \sum_D \left[[r=1] \left(1 - \sigma(\hat{r}^{\mathrm{SVD}}_{i,a} + \theta_a^\top l_i)\right) l_i \right. \\ &\qquad \left. -[r=-1] \sigma(\hat{r}^{\mathrm{SVD}}_{i,a} + \theta_a^\top l_i) l_i \right] = \sum_D \left[[r=1] - \sigma(\hat{r}^{\mathrm{SVD}}_{i,a} + \theta_a^\top l_i) \right] l_i.\end{aligned}$$

We denote $s_a = \sum_D \left([r=1] - \sigma\left(\hat{r}^{\mathrm{SVD}}_{i,a} + \theta_a^\top l_i\right)\right) l_i$. We can now precompute s_a (it is a vector over topics) for each document right after SVD is trained (this requires additional memory of the same size as to hold the θ matrix) and then use it on the LDA sampling step as follows:

$$\begin{aligned}p(z_w = t \mid \boldsymbol{z}_{-w}, \boldsymbol{w}, \alpha, \beta) &\propto q(z_w, t, \boldsymbol{z}_{-w}, \boldsymbol{w}, \alpha, \beta) p(D \mid \mu, b_i, b_a, p_i, q_a, l_i, \theta_a^{w\to t}) \\ &\approx \frac{n^{(d)}_{-w,t} + \alpha}{\sum_{t' \in T} \left(n^{(d)}_{-w,t'} + \alpha\right)} \frac{n^{(w)}_{-w,t} + \beta}{\sum_{w' \in W} \left(n^{(w')}_{-w,t} + \beta\right)} e^{s_a \theta_a^{w\to t}},\end{aligned}$$

and the latter is proportional to simply $\frac{n^{(d)}_{-w,t}+\alpha}{\sum_{t'\in T}\left(n^{(d)}_{-w,t'}+\alpha\right)}\frac{n^{(w)}_{-w,t}+\beta}{\sum_{w'\in W}\left(n^{(w')}_{-w,t}+\beta\right)}e^{s_t}$ because $s_a\theta_a^{w\rightarrow t} = s_a\theta_a - s_w z_w + s_t z_t$, and the first two terms do not depend on t which is being sampled. Thus, the first order approximation yields a very simple modification of LDA sampling that incurs relatively small computational overhead as compared to the sampling itself. In Sect. 5, we will see that this approximation does indeed work.

4.3 Variations of SVD-LDA

We have outlined a general approximate sampling scheme; however, several different variations are possible depending on which predictors are shared in the basic SVD model, $p(\text{success}_{i,a}) = \sigma\left(\hat{r}_{i,a}\right)$. In general, it hardly makes sense to train a separate set of l_i features for every user, as each user will be represented by far too many independent variables, which will lead to heavy overfitting. We used two variations:

(1) share $l_i = l$ among all users; in this case, we simply want to find out which topics are better received by the user base in general;

(2) share $l_i = l_c$ among certain clusters of users, preferably inferred from some external information; in our experiments below, we used demographic information (age and gender) to divide the users into 20 approximately equal clusters; in this case, we can infer topics that are better or worse received by specific demographic groups of users.

Both variations can be used for cold start with respect to users; for cold start recommendations, we simply substitute $p_i = 0$ in the prediction formula above.

5 Evaluation

5.1 Dataset

For our experimental evaluation, we have used a dataset provided by the recommender system *Surfingbird*[1]; this system recommends web pages that will hopefully be of interest to users (it is similar to *StumbleUpon*), so most items available for recommendation (except pictures and videos) come with full text (usually in Russian) that has already been parsed by *Surfingbird*. The dataset contains 515K users and 1364K items (web pages). We split it into a training set with 29M ratings (users rate items with likes and dislikes) containing ratings provided by the users from December 2011 through April 2014, and a test set with 1.7M ratings which entered the system in May 2014. Note that the dataset is imbalanced, there are about 8 times more likes than dislikes, so the values of all ranking metrics like AUC are close to 1 by default, and small changes are even more important than usually in recommender systems.

[1] http://surfingbird.ru.

For the experiments, we have developed an implementation of the SVD-LDA training algorithm with Gibbs sampling in the C++ programming language. In particular, we have implemented the basic LDA training, supervised LDA, SVD-LDA with exact sampling as shown in Sect. 4.1 (impractical on this scale), SVD-LDA with approximate sampling as in Sect. 4.2, and a natural extension of the algorithm where SVD has both base predictors and LDA topic predictors for demographic user clusters.

5.2 RMSE Improves with LDA Training

In the first series of experiments, we used RMSE (root mean squared error) to support that approximate inference in the SVD-LDA model does indeed work and does make LDA topics gradually more relevant to improving prediction quality in the SVD model. As we have seen in Sect. 4 on each iteration of LDA training the SVD-LDA model training algorithm learns an SVD model with predictors corresponding to current document-topic distributions θ_a. Figure 2 shows a sample graph of how final RMSE (on the test set) after SVD training declines as LDA iterations progress. The graph indicates that better LDA topics do indeed help SVD train better, significantly increasing its predictive power.

5.3 SVD-LDA Recommends Better Than SVD

The second series of experiments uses results of SVD-LDA training to further provide content recommendations to the users, in particular to recommend new web pages and/or make recommendations to new users ("cold start"). We have trained the following models on the training set: (1) SVD model without additional predictors; (2) SVD-LDA model with additional topic predictors; (3) SVD-LDA-DEM model with additional topic predictors for each demographic cluster.

To evaluate the results, we use ranking evaluation metrics: recommendations are represented as an ordered list (the order of recommendations is all that matters for the user), and the perfect ranking would be to have all "likes" in front of the list followed by all "dislikes". We used the following metrics:

- NDCG – Normalized Discounted Commulative Gain [12];
- AUC – Area Under (ROC) Curve, which is in the binary case equivalent to the share of correctly ranked pairs of items [13,14];
- Top-N metrics that show the share of "likes" in the top N recommendations, including Top-1 under the name of WTA (winner takes all) and Top-10 traditionally called MAP (mean average precision).

In the experiments, we varied the number of topics in SVD-LDA training, the number of features in the SVD model, and the regularization coefficient λ used for SVD training. Table 1 shows the best results we obtained for each model. It is clear that SVD-LDA outperforms SVD on all counts, while SVD-LDA-DEM provides an additional improvement over SVD-LDA. Note that in an imbalanced dataset, even small changes in the ranking metrics convert to significant improvements in recommendation quality.

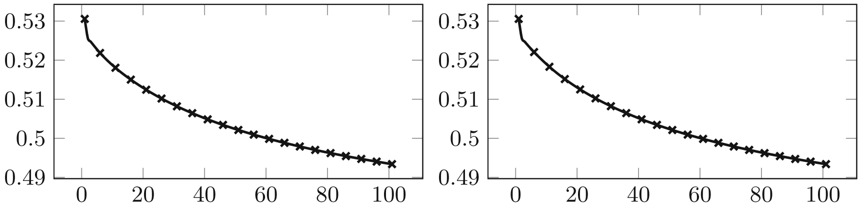


Fig. 2. RMSE at the end of each SVD-LDA training step as it changes through LDA iterations. Left: SVDLDA, 50 topics; right: SVDLDA with demographic user clusters, 200 topics.

Table 1. Ranking metrics on the test set. Only the best results w.r.t. λ and the number of features are shown.

Model	Topics	Features	λ	NDCG	AUC	MAP	WTA	Top3	Top5
SVD		5	0.1	0.9814	0.8794	0.9406	0.9440	0.9434	0.9424
SVD		10	0.15	0.9815	0.8801	0.9405	0.9448	0.9434	0.9425
SVD		15	0.2	0.9815	0.8802	0.9405	0.9453	0.9435	0.9426
SVD		20	0.2	0.9816	0.8803	0.9406	0.9453	0.9437	0.9427
SVD-LDA	50	5	0.025	0.9829	0.8893	0.9418	0.9499	0.9466	0.9445
SVD-LDA	100	10	0.025	0.9829	0.8893	0.9418	0.9500	0.9465	0.9445
SVD-LDA	200	15	0.01	0.9830	0.8895	0.9417	0.9524	0.9470	0.9446
SVD-LDA-DEM	50	10	0.01	**0.9840**	0.8901	**0.9428**	**0.9531**	**0.9481**	**0.9456**
SVD-LDA-DEM	100	5	0.01	**0.9840**	**0.8904**	**0.9428**	0.9528	0.9480	**0.9456**
SVD-LDA-DEM	200	10	0.01	**0.9840**	0.8898	**0.9428**	0.9524	**0.9481**	**0.9456**

It is interesting to note that our results, both in these experiments and in terms of RMSE shown in Sect. 5.2, do not depend much on the number of LDA topics; results from 50 through 200 topics are very similar, although the topics themselves do not deteriorate in quality, and inspection shows that LDA models with more topics do indeed uncover new meaningful distinctions in the content of web pages. This suggests that in reality, while there may be plenty of different topics in recommended items (our dataset is a collection of general interest web pages, so topics are very distinctive), not many topics do indeed have a distinctive effect on the recommendations, and it may suffice to use a small number of topics, e.g., 50 or 100, for similar systems in the future. This is important because LDA training with Gibbs sampling is computationally intensive, and it scales linearly with the number of topics.

5.4 Predictors for Demographic Clusters

One more demonstration of the of our approach comes from inspecting the predictors trained in the SVD-LDA-DEM model. Table 2 shows some of the best and worst topics for a selection of demographic groups characterized by gender and age (we have removed topics consisting of common words that are hard to interpret and topics that are good or bad uniformly across all demographic

Table 2. Topic predictors for some demographic groups (transl. from Russian).

l_{ct}	Topic	l_{ct}	Topic
Male, age ≤ 15		Female, age ≤ 15	
0.016	Movie song music album cinema musical	0.015	Girl star model wedding session singer
0.016	Car auto model engine company driver	0.011	Online free dog animal video cat
0.015	Bird forest animal height rock beach	0.010	Butter dish meat salt egg taste
0.013	Planet space Sun star gas satellite	0.009	Facebook social blog post vkontakte video
...		...	
0.004	Law political Putin deputy society	−0.002	Olympic games Sochi winner medal gold
0.001	Hair skin color shade mask make-up	−0.003	Wall room design style furniture
Male, age 25-29		Female, age 25-29	
0.014	Facebook social blog post vkontakte video	0.024	Music head buy read girl favorite
0.011	Xbox console company playstation world	0.023	Butter dish meat salt egg taste
0.010	Company user google mobile client phone	0.021	Movie song music album cinema musical
...		...	
0.000	Olympic games Sochi winner medal gold	0.006	Airplane tank war machine vessel flight
−0.002	Problem business reply plan company client	0.005	Exhibition museum art curated message

groups in order to emphasize the differences). Topics are characterized by a list of their top words; note that the values of predictors are incomparable across different groups because they can be redistributed with individual baseline predictors b_i (one can add and subtract a constant from all b_i and l_i inside a cluster, and predictions will not change). While sociological conclusions based on this data would not be sufficiently justified, we do believe that these results match sociological expectations, perhaps even stereotypes, very well.

6 Conclusion

In this work, we have presented a probabilistic model that unifies SVD as it is used in recommender systems and LDA for topic modeling into a single SVD-LDA model that attempts to train LDA topics that are useful for further recommendations. We have evaluated the resulting model on a real life full-text recommender system dataset, showing that both RMSE in SVD training and final ranking metrics improve significantly with the new model and that resulting topic predictors do indeed make sense in the context of demographic user clusters. As for further work, recent advances in pLSA regularization [15,16] suggest that it may be an interesting idea to develop regularizers that would lead to supervised pLSA and ultimately an SVD-pLSA model similar to SVD-LDA developed in this work. Another idea for further study might be to extend the SVD-LDA model to matrix decomposition techniques other than LDA (for the content of recommended items) and/or SVD (for the recommender system itself). This would let one process datasets with one-sided recommendations (e.g., with only likes and no dislikes) by switching from SVD to NMF [4] and process items with non-textual context with LDA variations for images [17], music [18], and other content.

Acknowledgements. This work was supported by the Samsung Research Center grant "Recommendation Systems based on Probabilistic Graphical Models", the Government of the Russian Federation grant 14.Z50.31.0030, and the Russian Foundation for Basic Research grant no. 15-29-01173.

References

1. Resnick, P., Iacovou, N., Sushak, M., Bergstrom, P., Riedl, J.T.: Grouplens: an open architecture for collaborative filtering of netnews. In: 1994 ACM Conference on Computer Supported Collaborative Work Conference, pp. 175–186, Chapel Hill, NC, Association of Computing Machinery (1994)
2. Said, A., Jain, B.J., Albayrak, S.: Analyzing weighting schemes in collaborative filtering: cold start, post cold start and power users. In: Proceedings of the 27th Annual ACM Symposium on Applied Computing, SAC 2012, pp. 2035–2040, New York (2012)
3. Koren, Y., Bell, R.M.: Advances in collaborative filtering. In: Ricci, F., Rokach, L., Shapira, B., Kantor, P.B. (eds.) Recommender Systems Handbook, pp. 145–186. Springer, US (2011)
4. Hu, Y., Koren, Y., Volinsky, C.: Collaborative filtering for implicit feedback datasets. In: Proceedings of the 8th IEEE International Conference on Data Mining, pp. 263–272, Pisa, Italy. IEEE Computer Society (2008)
5. Hoffmann, T.: Unsupervised learning by probabilistic latent semantic analysis. Mach. Learn. **42**, 177–196 (2001)
6. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent Dirichlet allocation. J. Mach. Learn. Res. **3**, 993–1022 (2003)
7. Griffiths, T., Steyvers, M.: Finding scientific topics. Proc. Nat. Acad. Sci. **101**(Suppl. 1), 5228–5335 (2004)
8. Blei, D.M., McAuliffe, J.D.: Supervised topic models. In: Advances in Neural Information Processing Systems, vol. 22 (2007)
9. Linden, G., Smith, B., York, J.: Amazon.com recommendations: item-to-item collaborative filtering. IEEE Internet Computing **7**(1), 76–80 (2003)
10. Agarwal, D., Chen, B.C.: fLDA: matrix factorization through latent Dirichlet allocation. In: Proceedings of the 3rd WSDM, pp. 91–100, New York. ACM (2010)

11. Agarwal, D., Chen, B.C.: Regression-based latent factor models. In: Proceedings of the 15th KDD, pp. 19–28 New York. ACM (2009)
12. Jarvelin, K., Kekalainen, J.: Cumulated gain-based evaluation of IR techniques. ACM Trans. Inf. Syst. **20**, 422–446 (2002)
13. Fawcett, T.: An introduction to ROC analysis. Pattern Recogn. Lett. **27**, 861–874 (2006)
14. Ling, C.X., Huang, J., Zhang, H.: AUC: a statistically consistent and more discriminating measure than accuracy. In: Proceedings of the International Joint Conference on Artificial Intelligence 2003, pp. 519–526 (2003)
15. Potapenko, A., Vorontsov, K.: Robust PLSA performs better than LDA. In: Serdyukov, P., Braslavski, P., Kuznetsov, S.O., Kamps, J., Rüger, S., Agichtein, E., Segalovich, I., Yilmaz, E. (eds.) ECIR 2013. LNCS, vol. 7814, pp. 784–787. Springer, Heidelberg (2013)
16. Vorontsov, K.: Additive regularization for topic models of text collections. Doklady Mathematics **89**, 301–304 (2014)
17. Cao, L., Fei-Fei, L.: Spatially coherent latent topic model for concurrent segmentation and classification of objects and scenes. In: IEEE 11th International Conference on Computer Vision, ICCV 2007, pp. 1–8 (2007)
18. Hu, D., Saul, L.K.: A probabilistic topic model for unsupervised learning of musical key-profiles. In: Hirata, K., Tzanetakis, G., Yoshii, K. (eds.) ISMIR, International Society for Music Information Retrieval, pp. 441–446 (2009)