Collective Factorization for Relational Data: An Evaluation on the Yelp Datasets

Nitish Gupta
Indian Institute of Technology
Kanpur, India
nitishgupta.iitk@gmail.com

Sameer Singh
University of Washington
Seattle, WA
sameer@cs.washington.edu

Abstract

Matrix factorization has found incredible success and widespread application as a collaborative filtering based approach to recommendations. Unfortunately, incorporating additional sources of incomplete and noisy evidence is quite difficult to achieve in such models, however this information is often crucial for obtaining further gains in accuracy. For example, in the Yelp datasets, additional information about businesses from reviews, categories, and attributes should be leveraged for predicting ratings, even though these are often inaccurate and partially-observed. Instead of creating customized solutions that are specific to the types of evidences, in this paper we present a generic approach to factorization of relational data that collectively models all the relations in the database. By learning a set of factors that are shared across all the relations, the model is able to incorporate observed information from all the relations, while also predicting all the relations of interest. Our evaluation on four Yelp datasets demonstrates effective utilization of additional information for held-out user preference and attribute prediction, but further, we present accurate models even for *cold-start* businesses for which we do not observe *any* ratings or attributes. We also present joint visualizations of word, category, and attribute factors, demonstrating learned dependencies between them that are not directly observed in the data.

1 Introduction

Predicting user preferences, for items such as for commercial products, movies, and businesses, is an important and well-studied problem in recommendation systems. Collaborative filtering using matrix factorization [Koren et al., 2009], in particular, has found widespread adoption as the tool of choice for this problem. By relying on co-occurrences in the ratings, however, these methods do not perform well on users or items that do not have ample observed ratings, i.e. that are rare or new to the system.

Fortunately, since users and items are part of a larger database, extra relational information about such users and items can often be utilized for predicting preferences. It is, for example, often not difficult to obtain information such as product categories, album genres, review text, and attributes/features of the items, however this external evidence is rarely complete or noise-free. A number of existing approaches have thus been proposed to use these sources of information for improving user preferences. Koren [2008], for example, combines the factorization model with an encoding of the external information as fully-observed features. Several studies have also investigated algorithms for incorporating specific sources of information, for example modeling user reviews [McAuley et al., 2012, Ling et al., 2014, Ganu et al., 2009], integrating context information [Karatzoglou et al., 2010, Hariri et al., 2014], exploiting item taxonomy [Koenigstein et al., 2011, Weng et al., 2008], and learning changes in user preferences and expertise over time [Koren, 2010, McAuley and Leskovec, 2013] to improve rating prediction. However, these approaches face a number of disadvantages when applied to heterogeneous, incomplete, multi-relational schema common in practice. First, these approaches are designed for certain *types* of relations and are restricted to relations of that type. Thus, it is not clear how additional sources of information can be incorporated, for example, how partially observed product categories can be used to improve rating prediction in McAuley et al. [2012]. Further, by training the model to predict entries of only one or two relations, these approaches ignore the dependencies between other relations and entities in the database, such as simultaneously predicting the cuisine of a restaurant, and the users that will like it, from the user reviews of the restaurant. There's a need for a generic machine learning approach that is able to leverage the dependencies between users, items, and additional data for estimating user preferences more accurately.

In this paper, we present a collective factorization model for incorporating heterogeneous relational data for user preference prediction in a domain-independent manner. Collective factorization assigns a latent low-dimensional vector (an *embedding* or *factor*) for every entity in the database that is used to predict all of the observed relations between pairs of entities. The collective model thus extends the intuition behind matrix factorization based recommendation systems that include embeddings for every user and business/product, and is a generalization of McAuley et al. [2012] that assign factors to every user, business/product, and review words. Since the latent embeddings in collective factorization are used to model all of the observed entries in the database, it is capable of predicting *any* type of relation between entities. Training the embeddings to capture all of the dependencies also makes it easy to integrate multiple evidences for the same relation; incorporating another source of information is as simple as including an additional relation/table in the database. Further, since the embeddings for all the entities are defined over the same low-dimensional space, we can compute similarity between any pair of entities, even if they are not directly observed in the same relation. The collective factorization model provides further benefits for practical deployment: the training algorithm is efficient and scalable, and the model complexity can be controlled by varying the embedding dimensionality.

We present a four-way evaluation of the collective factorization model (§2), as applied to the Yelp[1] datasets. (1) We demonstrate that the collective factorization model is effective in incorporating additional sources of information in §5.1, in particular provides significant accuracy gains for predicting user preferences. (2) In §5.2, we show that the proposed model is especially useful for *cold-start* estimation, e.g. for estimating preferences for *new* businesses and products for which no reviews or ratings have been observed. (3) An advantage of the model is that it can be used to impute missing values in the external data; we present an evaluation of this capability in §5.1 and §5.2. (4) We explore the implicit relations learned by the model that were not observed in the data by visualizing categories, business attributes, and the review words in the same two-dimensional plot in §5.3.

2 Probabilistic Collective Factorization

In this section, we present the probabilistic collective matrix factorization that jointly models the relations between entities, by leveraging data from all the other relations the entities participate in.

2.1 Relational Data

We represent relational data as a set of entities ($\mathcal{E}$) and relations between them ($\mathcal{R}$). Formally, the observed database, denoted by $\mathcal{D}$, consists of tuples of the form $\{r_t, e_{t1}, e_{t2}, y_t\}_{t=1}^{T}$, where $r_t \in \mathcal{R}$ is a relation, $e_{t1}, e_{t2} \in \mathcal{E}$ are a pair of entities, and $y_t \in \{0, 1\}$ denotes whether $r_t(e_{t1}, e_{t2})$ holds (or not). For example, a simple database that consists only of the user preferences from Yelp would contain businesses and users as the entities, and only a single relation r, such that $r(e_{t1}, e_{t2}) = 1$ if user e_{t1} liked the business e_{t2}. As is clear from this example, many databases in real-life are only sparsely observed, in that only a very small set of possible relations are observed, and the goal of modeling such datasets is to be able to *complete* this database. Specifically, given any query $r_q(e_{q1}, e_{q2})$ that is absent from the observed database, we would like to predict whether the relation holds.

2.2 Collective Factorization Model

Collective matrix factorization model [Singh and Gordon, 2008] extends the matrix factorization model to multiple matrices by assigning each entity a low-dimensional latent vector that is shared across all the relations the entity appears in. Formally, we assign each entity e in our database a k-dimensional latent vector $\phi_e \in \mathbb{R}^k$ (the set of these vectors for all the entities in the database is Φ). We model the probability that $r(e_1, e_2)$ holds by:

$$P_\Phi \left[r(e_1, e_2) = 1 \right] = \sigma(\phi_{e1} \cdot \phi_{e1}) \quad (1)$$

where σ is the *sigmoid* function, $\sigma(s) = \frac{1}{1+e^{-s}}$[2].

Therefore the probability that $r(e_1, e_2) = y$ is,

$$P_\Phi \left[r(e_1, e_2) = y \right] = \sigma(\phi_{e1} \cdot \phi_{e1})^y (1 - \sigma(\phi_{e1} \cdot \phi_{e1}))^{(1-y)} \quad (2)$$

The collective factorization model presents a number of advantages for our task. By sharing the entity factors amongst all the relations, they are able to capture all the sources of evidence in a joint manner, for example the factors used predict user ratings will leverage information from other ratings in a collaborative filtering fashion, but also from business attributes, categories, and words that appear in the reviews (the details of the model as applied to the Yelp data are described in §3). The sharing of factors also allows them to be used to predict any of the relations in the database, i.e. along with predicting user preferences, we can also predict business categories, attributes, and the text of the reviews. A further advantage of learning collective factors is that all the entities are effectively *embedded* in the same k-dimensional space, and thus similarities and distances can be computed and analyzed for any set

[1] www.yelp.com/dataset_challenge

[2] Along with producing numbers between 0 and 1, using a sigmoid provides an additional benefit of being more expressive than regular linear factorization, as shown in Bouchard et al. [2015].

of entities (we explore such visualizations in § 5.3). Finally, test-time inference takes constant time and thus is incredibly efficient: we only need a dot-product between low-dimensional vectors for estimating the probability of a relation to hold between a pair of entities.

2.3 Estimating Entity Factors

To estimate the parameters i.e. latent vectors Φ, we maximize the regularized log likelihood of the observed training instances (observed entries in the database, $\mathcal{D}$). Specifically, we maximize:

$$\hat{\Phi} = \arg\max_{\Phi} \; l(\mathcal{D}, \Phi) \tag{3}$$

$$l(\mathcal{D}, \Phi) = \sum_{t=1}^{T} \log P_{\Phi}\left[r_t(e_{t1}, e_{t2}) = y_t\right] - \lambda \left(\|\Phi\|_2^2\right) \tag{4}$$

To optimize this objective and estimate the latent factors, we use stochastic gradient descent (SGD) by cycling over the entries of the database multiple times, updating the latent factors in the direction of stochastic gradient for each entry. In particular, the i^{th} update that uses t^{th} database entry is given by,

$$\phi_{e_{t1}}^{(i+1)} \leftarrow \phi_{e_{t1}}^{(i)} + \gamma\left(e_t * \phi_{e_{t2}}^{(i)} - \lambda\phi_{e_{t1}}^{(i)}\right) \tag{5}$$

$$\phi_{e_{t2}}^{(i+1)} \leftarrow \phi_{e_{t2}}^{(i)} + \gamma\left(e_t * \phi_{e_{t1}}^{(i)} - \lambda\phi_{e_{t2}}^{(i)}\right) \tag{6}$$

where $e_t = y_t - \sigma\left(\phi_{et1}^{(i)} \cdot \phi_{et2}^{(i)}\right)$ and γ is the learning rate. Along with strong theoretical properties and widespread empirical success, the algorithm is also memory and time efficient since it runs on a single entry at a time, and additionally, provides further potential for scalability via parallelism [Niu et al., 2011].

3 Collective Factorization for Yelp

Yelp contains rich relational data for businesses and users in the form of business attributes, categories, and user reviews and ratings. Much of this relational data has inherent dependencies amongst the entities that pose exciting potential for integration, such as predicting user preferences and completing the missing information in the Yelp database by leveraging available information about the various entities. For example, predicting business attributes can be aided by the business categories and user reviews. Similarly, incorporating information about the businesses and learning user preferences from their past reviews can significantly improve user preference prediction.

We use the collective factorization model to learn universal latent factors for entities in Yelp by incorporating multiple relations that the entity participates in. We show how these factors can be used to predict relations and to estimate similarity between entities. The entities present in the Yelp database are *businesses*, *categories*, *attributes*, *users* and *review words*. We denote the set of these entities by S_B, S_C, S_A, S_U and S_W respectively and represent each entity by a k-dimensional factor, as shown in the top part of Figure 1. In the sections below we describe in detail the various relations we use from Yelp and show how we represent them as binary relational matrices.

3.1 Business Categories

Each business in Yelp is categorized into a set of nearly 700 types according to the nature of the business. The categories available in Yelp include broad-level classes such as *Doctors*, *Education*, and *Restaurant*, but also fine-grained descriptions such *Italian*, *Hookah Bars*, and *Orthodontists*. The business category data can be viewed as a binary relation between businesses and categories, and is represented as matrix C.

3.2 Business Attributes

Apart from categorization, Yelp also describes various attributes for each business. Such attributes include *type of parking*, *delivers* (or not), *noise level* and so on. We represent this relation between businesses and attributes as a binary matrix denoted by A. We transform attributes that are multi-valued into multiple binary valued attributes, for example the attribute "*Smoking*" in the dataset has "*Yes*", "*No*" and "*Outdoor*" as possible values. To represent this with binary values, we unfold it into three separate attributes, namely "*Smoking(Yes)*", "*Smoking(No)*" and "*Smoking(Outdoor)*", each of which is expressed as a binary value.

3.3 Ratings and Reviews

A complex relationship between users and businesses exists in the form of ratings and text reviews given by users. We represent this user-business relation in various forms. The ratings given by users on a 5-scale are converted to a binary-valued *preference* relation between businesses and users with high ratings (4 and 5) as *true(1)* and low ratings (3 and below) as *false(0)*. We denote the binary matrix representing this user preference relation by R, i.e. the likes and dislikes of users towards businesses.

The relationship between a business and words that appear in its reviews is represented by the relational matrix BW in which a *true(1)* value for a (business, word) tuple denotes the usage of the word for the business in at least one review. Similarly the relation between users and the words used by them in reviews is represented as a binary matrix denoted by UW.

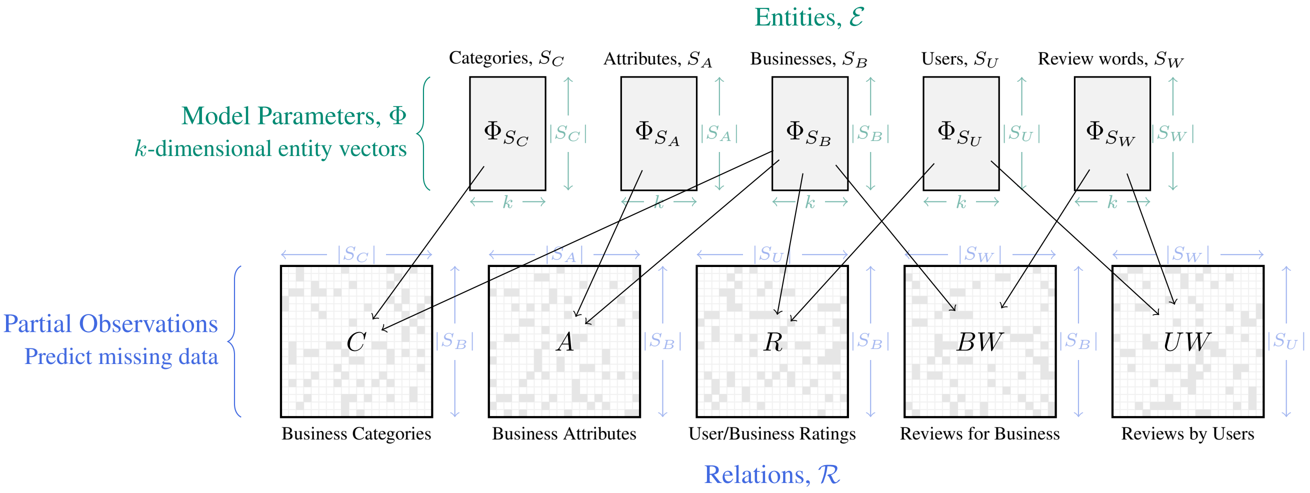

Figure 1: **Collective Factorization for the Yelp Dataset:** Overview of the entities and the relations, with the latter represented by sparsely-observed matrices. The collective factorization model contains low-dimensional dense factors for all the entities which are used to model the respective relations the entities appear in (denoted by arrows).

	$\mathbf{\|S_A\|}$	$\mathbf{\|S_C\|}$	$\mathbf{\|S_B\|}$	$\mathbf{\|S_W\|}$	$\mathbf{\|S_U\|}$
Phoenix	92	472	22 180	25 277	102 576
Las Vegas	92	416	14 583	28 551	147 774
Madison	77	176	2 118	6 811	9 737
Edinburgh	74	160	2 840	6 830	2 484

Table 1: Number of entities of each type

	$\mathbf{\|A\|}$	$\mathbf{\|C\|}$	$\mathbf{\|R\|}$	$\mathbf{\|BW\|}$	$\mathbf{\|UW\|}$
Phoenix	354 068	10 468 960	475 116	8 533 231	12 339 706
Las Vegas	235 735	6 066 528	556 326	7 246 237	16 598 396
Madison	41 105	372 768	35 661	706 026	987 735
Edinburgh	41 218	454 400	20 306	730 871	435 801

Table 2: Number of observed entries for each relation

3.4 Datasets and Sizes

Yelp provides data from five cities namely, *Phoenix*, *Las Vegas*, *Madison*, *Edinburgh* and *Waterloo*, but we focus on the first four datasets due to the small size of the *Waterloo* dataset. Each of the datasets follows the same schema, allowing evaluation of our model. For each of the datasets we create the relational matrices, A, C, R, BW and UW. Table 1 shows the number of entities participating in each relation of the various databases, while in Table 2 we present the number of observed entries for each relation.

Figure 1 gives an overview of the various relations and entities present in the Yelp database we create. It shows how different entities participate in multiple relations, which leads to their latent factors being shared among different relations. For example, the latent factors for *businesses* (S_B) participate in modeling relations A, C, R and BW.

4 Experiment Setup

In this section, we describe some of the details of our evaluation setup. To create the BW and UW matrices, we tokenize the reviews, remove the punctuations, numbers, and stop words, and stem the words using Porter [1980]. For evaluation purposes, we only consider words that appear in at least 10 reviews. Since BW and UW matrices only contain observed words (all positives), we sample negative data entries in each epoch by randomly selecting a set of words that were not observed to be true for the business/user. The number of negative samples chosen for each relation is same as the number of observed entries for the relation. We found the categories C matrix to be fairly comprehensive, and thus explicitly treat all unobserved entries to be negative (thus effectively C is fully-observed and complete). For our experiments, we only consider categories that are associated with at least 5 businesses.

The primary benchmarks for evaluating our models will be on predicting user ratings and business attributes, in particular study how incorporating additional information into the factorization model provides significant improvement in predictions. The baseline models that perform standard matrix factorization of R and A independent of other relations are denoted by **R** and **A** respectively. We evaluate the effect of integration of different relations by factorizing combinations of different matrices collectively with the relation we want to predict. An example of the model that predicts ratings by incorporating business categories is denoted by **R + C**. To predict whether a relation holds between entities, we primarily use the default logistic threshold of 0.5 for the predicted probability. We measure the performance of our relation prediction in terms of the *F1 score* defined as the harmonic mean of the precision and recall, which is a much more accurate measure than accuracy for imbalanced label distributions. To present a combined score for all the datasets, we aggregate all the predictions of the datasets, and compute a single F1 score over them in the micro-averaged fashion. The value of the regularization constant, $\lambda = 0.001$, latent-factor dimensions $k = 30$ and learning rate, $\gamma = 0.01$ is used, based on the performance on validation data.

5 Results

In this section, we evaluate the effect of incorporating relational information on the collective factorization model in predicting user preferences and business attributes. First, we present the accuracy of predicting user preferences and attributes on a *held-out* test set in §5.1 to test the performance on entities with observed ratings and attributes. We also investigate the performance of different models on *cold-start* estimation for businesses in §5.2, where, for example, we predict user preferences for businesses for which *no* past ratings or reviews have been observed. Finally, utilizing the fact that the model embeds all entity types in the same k-dimensional space, we present visualizations in §5.3 that explore similarities between entities for which explicit relations are not observed.

5.1 Held-Out Evaluation

The most important problem in database completion is to be able to predict unobserved relations for entities that already exist in the database. For example, predicting user preferences for existing users in Yelp is important to recommend businesses to users by learning preferences from their past rating and review data. Similarly, attribute prediction for existing businesses is essential to complete the Yelp database, which could help users make more informed decisions when choosing between businesses.

To show how our model improves significantly on predicting relations by leveraging additional information for existing entities, we carry out evaluations on a held-out test set from the observed data. To split the data for the evaluation into training, validation and test sets, we randomly choose 70% of the observed cells of the relation to be tested for training and equally divide the remaining data into validation and test sets. We vary the business and user relations available during training for both rating and attribute prediction.

User Preference Prediction: We expect that incorporation of additional information about *businesses* and *users* such as reviews, categories, and attributes should improve prediction of the user preferences. Results for collective factorization of combinations of various relational matrices with the R matrix are shown in Table 3. Our baseline model achieves an *F1 score* of 71.3%, with an increase of 1.4% when incorporating information about the businesses in terms of its attributes (A) or review words used for them (BW). Incorporating business categories (C) improves upon the baseline model by 3.22%. Significant improvement of 11.07% from the baseline is obtained by incorporating relationship between the users and their review words (UW). From this, it is clear that the user reviews are quite indicative of their likes and dislikes for various aspects of a business, which further helps to predict user preferences. Further increase of 1.5% obtained by incorporating *business category* (C) information on top of *user-words* (UW) relation, suggests that the addition of categories helps the model learn user biases towards categories along with their other preferences. When adding information about *business attributes* along with *business categories* and *user-words* relation, we find that the prediction accuracy falls only slightly by 0.62%. A reason for this may be a lack of dependence between user user preferences and business attributes, and further, the model has to predict both user preferences and attributes simultaneously. The precision/recall curves in Figure 2 show how incorporating different kinds of information about users and businesses affect user preference prediction. It is clear that *user-word* relation provides higher gains than incorporating information about businesses, but more importantly, integrating information about both businesses and users achieves the best performance.

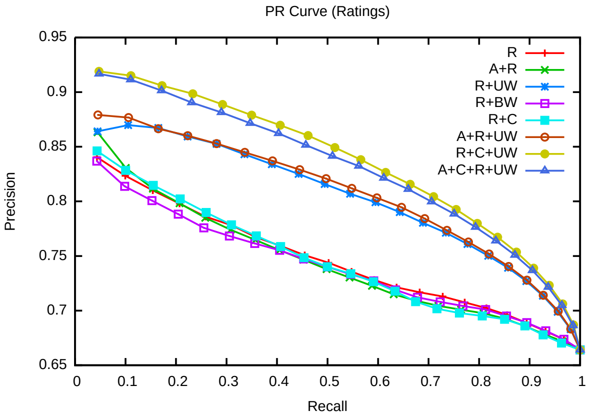

Figure 2: Precision/Recall for Held-out User Preferences

Attribute Prediction: Since our model is factorizing all relations collectively, it is capable of predicting missing entries for any of the relations. Table 4 shows how we can improve attribute prediction by integrating additional information about businesses, such as categories and review words written for them. Integrating the *business-word*

	Phoenix			Las Vegas			Madison			Edinburgh			**Combined**		
	P	R	**F1**	P	R	**F1**	P	R	**F1**	P	R	**F1**	P	R	**F1**
R	72.3	72.0	72.2	70.6	70.5	70.5	71.3	65.9	68.5	75.1	75.2	75.2	71.5	71.1	71.3
R+A	71.3	76.1	73.7	69.1	73.2	71.1	70.2	70.7	70.5	73.1	79.8	76.3	70.2	74.5	72.3
R+BW	72.1	76.4	74.2	69.4	72.2	70.8	70.1	72.7	71.4	68.8	84.4	75.8	70.6	74.3	72.4
R+C	70.8	80.0	75.1	68.4	76.3	72.2	70.0	74.4	72.2	73.5	79.6	76.4	69.7	78.0	73.6
R+UW	73.9	87.1	80.0	74.3	83.2	78.5	73.9	83.8	78.5	75.7	81.5	78.5	74.2	84.9	79.2
R+A+C	71.6	77.0	74.2	69.0	73.2	71.1	70.1	71.2	70.7	72.8	76.9	74.8	70.3	74.9	72.5
R+A+BW	72.1	76.1	74.1	69.4	72.0	70.7	71.1	70.6	70.8	74.3	78.9	76.5	70.8	73.9	72.3
R+C+BW	71.9	76.9	74.3	69.3	72.5	70.9	71.1	72.3	71.7	74.6	79.9	77.2	70.6	74.6	72.6
R+A+UW	74.7	86.1	80.0	74.2	83.4	78.5	73.3	85.6	78.9	75.9	82.7	79.1	74.4	84.7	79.2
R+C+UW	76.8	85.5	80.9	75.8	84.3	79.9	76.5	85.5	80.8	76.4	81.8	79.0	76.3	84.9	80.4
R+A+C+BW	72.1	75.8	73.9	69.3	72.3	70.8	71.4	70.8	71.1	74.5	78.3	76.4	70.7	74.0	72.3
R+A+C+UW	76.5	85.5	80.8	75.6	83.4	79.3	77.0	78.9	77.9	76.4	80.7	78.5	76.1	84.1	79.9

Table 3: **Held-out Evaluation of User Preference Prediction:** Precision/Recall/F1 on the different datasets on predicting held-out user preferences from R . The models being evaluated vary in the number of relations modeled when learning the factors, with additional relations often resulting in more accurate models across datasets.

	Phoenix			Las Vegas			Madison			Edinburgh			**Combined**		
	P	R	**F1**	P	R	**F1**	P	R	**F1**	P	R	**F1**	P	R	**F1**
A	83.3	82.3	82.8	82.7	78.4	80.5	80.2	76.4	78.3	80.9	76.8	78.8	82.8	80.3	81.5
A+R	82.7	81.3	82.0	81.7	78.1	79.9	79.5	75.8	77.6	79.0	77.4	78.2	82.0	79.7	80.8
A+BW	85.7	84.0	84.8	83.9	82.0	82.9	81.2	78.5	79.8	79.7	77.2	78.4	84.5	82.6	83.5
A+C	85.3	84.7	85.0	84.5	81.5	83.0	82.9	79.4	81.1	82.6	80.0	81.3	84.7	83.0	83.9
A+R+C	85.0	84.5	84.7	84.1	80.3	82.1	82.9	79.0	80.9	82.3	79.2	80.7	84.4	82.4	83.4
A+R+BW	85.4	84.2	84.8	83.9	81.5	82.7	81.7	78.8	80.2	80.4	77.3	78.8	84.4	82.6	83.5
A+C+BW	85.1	84.1	84.6	84.4	82.2	83.3	81.2	78.6	79.9	80.2	79.2	79.7	84.4	82.8	83.6
A+R+C+BW	85.3	84.3	84.8	84.0	82.2	83.1	81.3	80.5	80.9	79.7	79.1	79.4	84.3	83.1	83.7

Table 4: **Held-out Evaluation of Business Attributes:** Accuracy of predicting held out attributes from A on the different datasets. The learned factors for businesses are ore accurate when additional relations about the businesses is included. We exclude relation UW since it does not share any entities with A .

(BW) and *business category* (C) relation improves the attribute prediction baseline of 81.5% by 2.45% and 2.94%, respectively. This behaviour is expected as categories often dictate the presence (and absence) of certain attributes almost surely. Further, attributes mentioned in the reviews also leads to better attribute prediction ($+\mathbf{BW}$). Incorporating the *ratings* relation performs slightly worse by 0.8% which confirms that attributes are not the most vital aspects for ratings, as we saw in user preference prediction. Thus, simultaneously predicting ratings affects model performance, however only by a small amount.

5.2 Business Cold-Start Evaluation

One of the major challenges faced by recommendation systems is to predict user preferences for new businesses and users for which no reviews or ratings have been observed. This problem is not just specific to recommendation systems, but common to all relation prediction frameworks. Most of the factorization models for relation prediction fail to incorporate information about entities from relations, apart from the relation to be predicted, and thus provide poor cold-start performance.

Collective factorization benefits substantially since it leverages all the sources of information about the entity. Hence, in the absence of observed data for a particular relation, factors learnt from other relations can still be used to predict the relation. Specifically, we show that our model can learn factors for businesses for which no reviews or ratings were observed from its categories and attributes, and use them to predict user preferences. We also show that categories, reviews, and ratings information about businesses can be leveraged to predict attributes for businesses without any observed attributes. For evaluation, we withhold *all* observed cells of the relation being predicted (R or A) for a random 15% of the businesses. We use 80% of the remaining data for training and the rest for validation. Apart from the variety of collective models, we also include the uninformative straw man that has the same prediction for all cold-start businesses in both user preference and attribute prediction, evaluated

	Phoenix			Las Vegas			Madison			Edinburgh			**Combined**		
	P	R	**F1**	P	R	**F1**	P	R	**F1**	P	R	**F1**	P	R	**F1**
R	68.4	48.9	57.0	65.2	50.4	56.9	68.8	50.2	58.0	67.2	48.5	56.3	66.8	49.7	57.0
R+UW	68.4	50.1	57.8	65.3	50.6	57.0	67.3	48.5	56.4	70.9	51.5	59.7	66.8	50.3	57.4
R+A	71.2	74.5	72.8	67.8	71.1	69.4	71.5	69.6	70.6	72.4	76.2	74.2	69.6	72.7	71.1
R+C	71.6	79.0	75.1	67.8	77.0	72.1	70.8	73.9	72.3	72.9	78.3	75.5	69.7	77.8	73.5
R+A+C	71.9	78.2	74.9	68.2	74.6	71.2	70.6	72.8	71.7	71.2	85.3	77.6	70.0	76.4	73.0
R+A+UW	69.7	88.8	78.1	67.5	83.0	74.5	69.9	89.2	78.4	72.2	78.2	75.1	68.7	85.8	76.3
R+C+UW	72.4	88.3	79.6	69.8	83.8	76.2	76.2	72.3	74.2	75.2	65.5	70.0	71.3	85.1	77.6
R+A+C+UW	73.4	86.6	79.5	70.3	85.9	77.4	73.9	82.2	77.8	73.2	71.9	72.6	71.9	85.9	78.2

Table 5: **Cold-Start Evaluation of User Preference Prediction:** We present the precision/recall/F1 of different collective factorization models in predicting user preferences for businesses for which *no* ratings or reviews are observed. Conventional matrix factorization, **R**, is a trivial straw-man in that it does not have any way to differentiate amongst cold-start businesses.

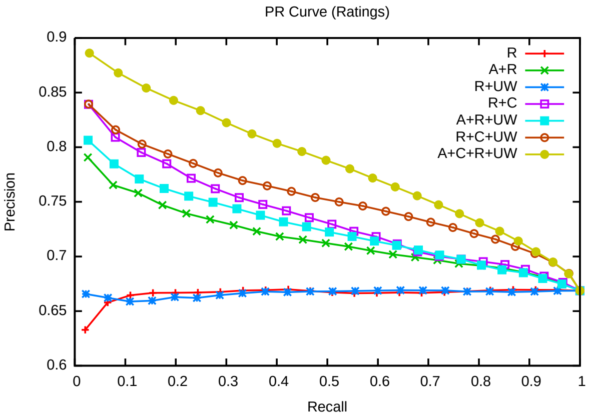

Figure 3: Precision/Recall for Cold-Start User Preferences

by computing the *F1 score* when the the factors for new businesses are randomly initialized to small values.

Cold-Start User Preference Prediction: Predicting user preferences for new businesses *a priori* is an exciting problem since it can help owners or Yelp quickly identify the target audiences that would like the business. Users that have biases towards certain categories and attributes prefer businesses that cater to their needs. Using our collective factorization model, we can integrate categories and attributes for new businesses, along with user reviews and ratings of existing businesses, to predict user preferences for new ones.

Table 5 shows the performance of different models that vary in the information being used to predict user preferences. The results corroborate the fact that learning good factors just for users (via UW) is not enough to predict user preferences. We find that incorporating business information in terms of attributes (A) and categories (C) obtains an accuracy as high as 73%, and further, integrating of user-word relation (UW) leads to attainment of prediction accuracy as high as 78.2% F1. By obtaining results surprisingly close to those in § 5.1, we demonstrate that the collective factorization model is able to almost completely overcome the lack of existing user preferences by utilizing other relations. Figure 3 also shows how UW does not provide improvements on user preference prediction for new businesses, but incorporating additional data about new businesses leads to greater improvements in user preference prediction.

Cold-Start Attribute Prediction: Quite surprisingly, 15.91% of the total $42\,151$ businesses in the Yelp database do not contain any information about their attributes. Here we show how attributes of such businesses can be learned effectively from their category, review and rating data. In Table 6 we show the accuracy of our model on attribute prediction for businesses with no attribute data observed during training. We find that the user preference history of a business helps the least in predicting attributes, which is consistent with our findings in held-out evaluations. As expected, incorporating business-category (C) and business-word (BW) relation helps the most in predicting attributes for new businesses. *F1 score* as high as 81.1% is achieved on incorporating both the C and BW relations. Integrating ratings data on top, doesn't affect the model a lot and obtains an *F1 score* of 80.9%. Here, as well as in the user preferences, the collective factorization model obtains performance quite close to the held-out evaluation, demonstrating that it compensates for missing data by effectively incorporating additional relations.

5.3 Qualitative Evaluation

Finding relationships and similarity between entities that do not participate in the same relation in the database schema is a challenging problem in relation learning. Similarity between entities has important applications in data visualization and developing intuitive user interfaces,

	Phoenix			Las Vegas			Madison			Edinburgh			**Combined**		
	P	R	**F1**	P	R	**F1**	P	R	**F1**	P	R	**F1**	P	R	**F1**
A	32.1	49.3	38.9	30.6	48.8	37.6	31.5	48.5	38.2	26.3	49.2	34.3	31.2	49.1	38.1
A+R	52.4	63.6	57.5	48.1	60.4	53.5	46.1	57.7	51.3	46.9	61.8	53.4	50.2	62.1	55.5
A+C	81.3	77.9	79.6	81.5	73.8	77.4	77.1	71.3	74.0	79.5	73.0	76.2	81.0	75.8	78.3
A+BW	83.0	80.6	81.8	83.1	80.6	81.8	79.8	75.4	77.5	73.2	72.1	72.7	82.3	79.8	81.0
A+R+C	80.4	77.1	78.7	79.4	72.6	75.9	74.6	71.3	72.9	77.1	72.5	74.7	79.6	75.0	77.2
A+R+BW	83.3	80.9	82.1	82.0	79.2	80.6	78.9	75.5	77.2	74.5	74.3	74.4	82.1	79.6	80.9
A+C+BW	83.2	81.0	82.1	82.8	79.3	81.0	79.2	75.2	77.1	78.5	74.9	76.6	82.6	79.7	81.1
A+R+C+BW	83.2	81.0	82.1	81.9	79.3	80.6	79.4	75.3	77.3	77.4	74.1	75.7	82.2	79.7	80.9

Table 6: **Cold-Start Evaluation of Attributes:** Performance of different collective factorization models in predicting attributes for business without any observed attributes. As for ratings, matrix factorization **A** is an uninformative baseline that has the same predictions for all business attributes.

amongst others. Since our model defines latent factors for all entities over the same k-dimensional space, we can compute similarity and distances between any pairs of entities even if they do not appear in the same relation. For example, reviews, along with indicating user preferences, also contain information about the business categories and attributes. In this section, we show how our model is able to learn factors for review words, categories and attributes that reveal similarity between them. We project the factors of a select subset of categories, attributes and a subset of similar review words onto a 2-dimensional plot using the *t-Distributed Stochastic Neighbor Embedding (t-SNE)* [van der Maaten and Hinton, 2008] technique for dimensionality reduction. This is a randomized, approximate technique that attempts to maintain the distances between entities in k dimensions when projecting them to two dimensions. The vectors used here for categories, attributes and review words are obtained by collectively factorizing A, R, C and BW relations on *Phoenix* data.

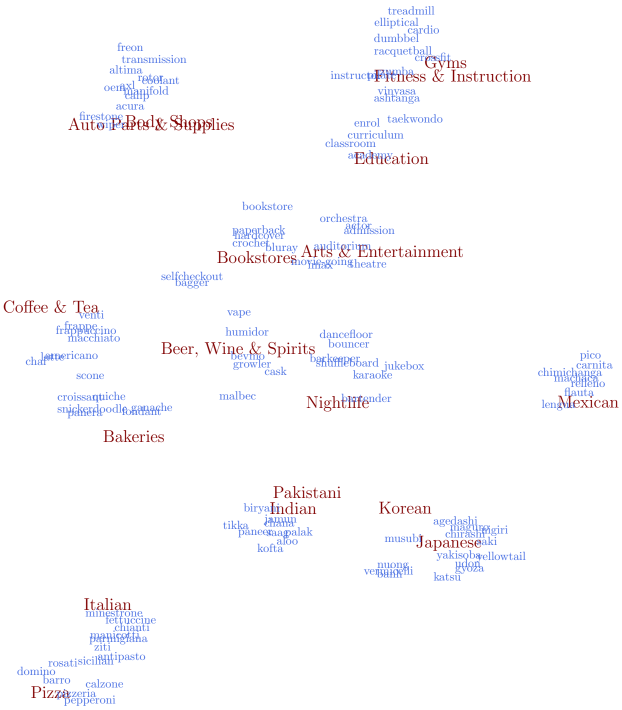

Figure 4: Visualization of Category and Review Word factors, showing similarity of factors for semantically-similar words and categories. Best viewed in color.

Visualizing Categories and Words: The efficacy of our model in learning inter-category similarity and similarity between categories and review words is shown in Figure 4. For example, our model is able to learn that *Indian* and *Pakistani* cuisines and *Korean* and *Japanese* cuisines are similar to each other, *Gyms* and *Fitness & Instruction* categories for businesses are similar and *Beer, Wine & Spirits* is related to *Nightlife*. Our model also learns the semantic similarity between categories and review words. For example, words closest to *Auto Parts & Supplies* are *rotor*, *coolant*, *wiper*, *transmission* and closest to *Arts & Entertainment* are *auditorium*, *theatre*, *imax*, *movie-going* and *orchestra*. For categories related to food, our model learns the names of dishes as being closest to categories, suggesting that the users mostly talk about the dishes when reviewing restaurants. For example, the words closest to *Mexican* are *carnita*, *flauta*, *chimichanga*, and *relleno*, *Coffee & Tea* are *frappe*, *chai*, *macchiato*, and *frappuccino*, and *Bakeries* are *scone*, *croissant*, and *quiche*. Our model is also able to learn word factors in such a manner that words used in reviews for dissimilar businesses are approximately between both the categories. For example, words like *enrol*, *taekwondo* and *curriculum*, which may belong to reviews of both education related businesses and fitness centres, lie in between the *Education* and *Fitness & Instruction* category. Similar observations are made in words that are close to *Bakeries* and *Coffee & Tea* categories.

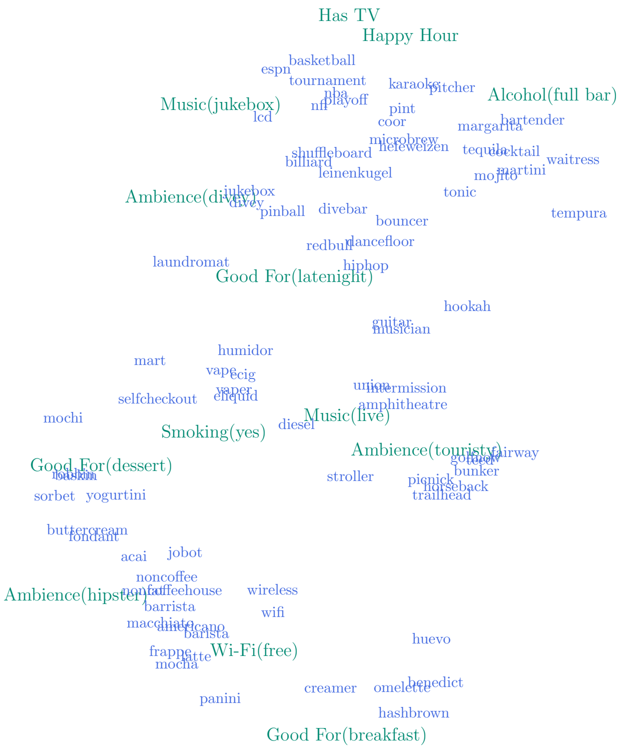

Figure 5: Visualization of Attribute and Review Word factors, showing similarity of factors for semantically-similar words and attributes. Best viewed in color.

Visualizing Attributes and Words: The similarities learned between business attributes and review words by our model is shown in Figure 5. From the figure, we see that our model learns similar factors for attributes that co-exist for certain types of businesses. For example, *Has TV*, *Happy Hour* and *Alcohol(full bar)* lie close to each in the figure suggesting that a business that serves alcohol, often has happy hours and TV screens. Further, places that have a divey ambience are good for late nights is suggested by the proximity of *Ambience(divey)* and *Good For(latenight)* attributes. The word factors also give interesting insights into attributes that are not evident from their categories. For example, word like *barista*, *frappe*, *mocha*, *americano* lie close to the attribute *Wifi(free)* along with the words *wifi* and *wireless* that endorses with the fact that coffee shops and cafes mostly have free Wi-Fi. Finally, words such as *omelette*, *hashbrown*, *creamer* lying close to *Good For(breakfast)* indicates that reviews often mention breakfast dishes for businesses that are good at it.

Visualizing Categories, Attributes, and Words: In Figure 6, we plot a subset of categories and the attributes and review words that are close to them. The proximity of the categories *Fast Food* and *Coffee & Tea* to the attributes *Drive-Thru*, *Wi-fi(free)* and *Alcohol(none)*, category *Doctors* to attribute *By Appointment Only*, and the category *Arts & Entertainment* to attribute *Music(live)*, all demonstrate that the model is able to learn how certain categories of businesses are most likely to have certain attributes. We also see from the figure that the even though the reviews do not explicitly talk about the attributes and categories, our model is able to capture the similarity between them simultaneously. For example, words like *jukebox*, *karaoke*, *bartender*, *chianti* lie close to attributes *Good For(latenight)*, *Happy Hour* and the category *Nightlife*.

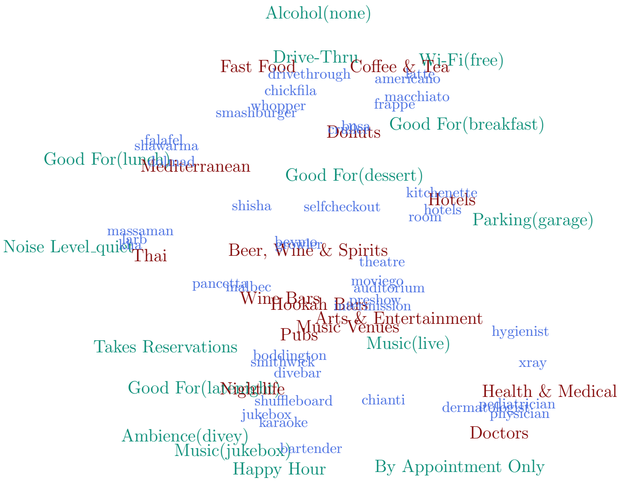

Figure 6: Visualization of Categories, Attributes, & Words

6 Related Work

This work builds upon a large and growing area of machine learning applied to recommendation systems and modeling of structured datasets. We describe a subset of these approaches that are directly related to, and inspired, our proposed work.

The idea of using low-dimensional vectors as latent factors has found widespread use in recommendation systems. The task of suggesting products/items to users is traditionally viewed as matrix completion where the sparse rating matrix with users as rows and items as columns is to be completed with predicted ratings. Sarwar et al. [2000] show how Singular Value Decomposition (SVD) can be used to decompose the rating matrix into low rank feature matrices to reduce dimensions of the rating matrix. This gave rise to the widely used matrix factorization techniques for predicting ratings [Koren et al., 2009] in which the user and item factors capture the similarities amongst them. Conventional matrix factorization techniques predict ratings directly as the dot product of the factors of the user and the item, and use regularized least-squares as the loss function to optimize. Our model here however uses the *probabilistic* interpretation of matrix factorization [Salakhutdinov and Mnih, 2008] and uses the sigmoid function with log-likelihood as it is a generalization of PCA to binary matrices [Collins et al., 2001].

Since many collaborative filtering applications often have auxiliary information available for users and products/businesses, a number of approaches have studied how this information can be combined with matrix factorization for better rating prediction. If the auxiliary observation can be treated as fully-observed and noise-free, it can be used in conjunction with the neighborhood model to augment the matrix factorization objective [Koren, 2008]. In practice, however, the auxiliary data is commonly noisy and incomplete, and thus has to be explicitly modeled for adequately leveraging it. McAuley et al. [2012] combines matrix factorization with review text by modeling the words using a LDA topic model, and aligning the item/user latent vector with the review text topic vectors to learn better factors for rating prediction. Ling et al. [2014] similarly combine review text, but use mixture of Gaussian instead of matrix factorization, avoiding any transformation of the factors and thus retaining the interpretability of latent topics. Ganu et al. [2009] predict ratings for restaurants from the review text alone, but require additional manually labeled data for classifying the sentiment and aspects of sentences. External information in the form of item taxonomies have also been investigated, for example, Weng et al. [2008] combine users' preferences with the item types to learn type-level preferences, while also addressing the cold-start problem for items with only taxonomic information, but do not employ any factorization model. Koenigstein et al. [2011] use global item biases in the Yahoo! Music dataset by using shared parameters amongst items with a common ancestor in the taxonomy hierarchy. Koren [2010] incorporates temporal dynamics into matrix factorization to learn changes in user movie preferences that occur over time, whereas, McAuley and Leskovec [2013] argue that, to enjoy certain kinds of products such as *beer* and *gourmet foods*, one requires a certain level of *expertise*, hence their model tries to combine temporal ratings data to make better personalized recommendations according to the *experience* of each user. Methods described above propose models that are specific to their domains, and thus the generalization capabilities of these models is unclear.

An alternative approach is to combine all the data and represent it using tensors, allowing the use of tensor factorization, an extension of matrix factorization to tensors. For example, the approach by De Lathauwer et al. [2000] is used to predict tags for a user-item pairs [Symeonidis et al., 2008, Rendle and Schmidt-Thieme, 2010] and to predict user ratings by integrating context information as a tensor [Karatzoglou et al., 2010]. The main shortcoming of such approaches, however, is that they model only a single additional source of information, and further, focus on predicting only the relation of interest.

To model multiple relations in a joint manner, collective matrix factorization [Singh and Gordon, 2008] extend the idea of matrix factorization to multiple matrices. The rows and columns of the matrices have corresponding latent factors, with shared latent factors for entities that appear as rows or columns in multiple matrices. These approaches learn parameters for entities by jointly factorizing all of these matrices, and thus learn factors that predict multiple matrices. The empirical evaluation on relatively small databases with only two relations did not show considerable improvements; this is expected since collective factorization requires, and would benefit from, larger datasets. We use this model with the logistic/sigmoid formulation in this paper, combined with stochastic gradient descent (SGD) for optimization, and evaluate on 4 large-scale, multi-relation real-world datasets from Yelp.

Our formulation of relational data, and the collective factorization model, can be easily extended. For example, the current formulation assumes at most a single relation exists between any specific pair of entities (since P_Φ is independent of relation r in Eq 1). Although this assumption holds for many applications, we can extend this model to multiple relations between the same pair of entities by introducing latent factors for the relations, similar to CP-decomposition (or PARAFAC) and recently proposed RESCAL [Nickel et al., 2011]. Our work is also related to work in statistical relational learning (SRL) and probabilistic databases (PDBs) that propose probabilistic modeling of relational data [Taskar et al., 2002, Heckerman et al., 2004, Wang et al., 2008, Dalvi et al., 2009, Singh and Graepel, 2013], in particular, Krompaß et al. [2014] obtain highly compressed representations of large triple stores by using RESCAL to represent them as PDBs, and present methods to efficiently answer complex queries on PDBs by breaking them into sub-queries.

Our model also assumes binary absence/presence relations, however non-Boolean binary relations can be modeled either by treating them as multiple relations or by using a different function than the sigmoid, while n-ary relations can be modeled as tensors with CP decomposition. It is worth mentioning that the Yelp dataset does contain such deviations from our assumptions: the business attributes are discrete valued (Wi-Fi: Free, Paid, No) which is converted to multiple Boolean yes/no entities (Wi-Fi:Free, Wi-Fi:Paid, Wi-Fi:No), while the reviews in Yelp are 3-way relation between users, businesses, and words which we split into two binary relations.

7 Conclusions and Future Work

In this paper, we presented the application of the collective relational factorization model to four of the Yelp datasets. By learning entity factors that are shared between all the relations the entity participates in, the model is able to combine multiple sources of evidence, predicts relations of multiple types, and further, allows computation of similarity between entities that do not share any direct relations. We presented empirical evaluation of user preference and business attribute prediction that demonstrates that the collective model achieves higher accuracy with access to additional evidence. We also investigated *cold-start* evaluation for businesses, and showed that the collective model is accurate in predicting user preferences (and attributes) even when none of the ratings (and attributes, respectively) of the business have been observed. We additionally explore joint visualization of categories, business attributes, and review words, facilitated by the collective factors. The code for the algorithm, along with data processing and evaluation, is available for download[3].

We would like to explore a number of avenues for future work. As we described in §6, we will extend our collective factorization representation of relational data to support n-ary relations (by using tensor factorization) and to non-binary, multi-valued relations (for example, by introducing additional factors for relations). These extensions will enable us to support a wider variety of relations and databases; we will be able to model the complete Yelp schema, including attributes such as tips, locations, temporal information, and review tags, with a single collective factorization model. We will also investigate applications of this model on relational databases from other domains.

Acknowledgements

This work was supported in part by the TerraSwarm Research Center, one of six centers supported by the STARnet phase of the Focus Center Research Program (FCRP) a Semiconductor Research Corporation program sponsored by MARCO and DARPA, and in part by ARO grant number W911NF-13-1-0246. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the view of the sponsors.

References

Guillaume Bouchard, Sameer Singh, and Theo Trouillon. On approximate reasoning capabilities of low-rank vector spaces. In *AAAI Spring Syposium on Knowledge Representation and Reasoning (KRR): Integrating Symbolic and Neural Approaches*, 2015.

Michael Collins, Sanjoy Dasgupta, and Robert E Schapire. A generalization of principal components analysis to the exponential family. In *Advances in neural information processing systems*, pages 617–624, 2001.

Nilesh N. Dalvi, Christopher Ré, and Dan Suciu. Probabilistic databases: diamonds in the dirt. *Comm. of the ACM*, 52(7):86–94, 2009.

Lieven De Lathauwer, Bart De Moor, and Joos Vandewalle. A multilinear singular value decomposition. *SIAM journal on Matrix Analysis and Applications*, 21(4):1253–1278, 2000.

Gayatree Ganu, Noemie Elhadad, and Amélie Marian. Beyond the stars: Improving rating predictions using review text content. In *WebDB*, volume 9, pages 1–6, 2009.

Negar Hariri, Bamshad Mobasher, and Robin Burke. Context adaptation in interactive recommender systems. In *ACM Conference on Recommender systems*, 2014.

David Heckerman, Christopher Meek, and Daphne Koller. Probabilistic models for relational data. Technical Report MSR-TR-2004-30, Microsoft Research, 2004.

Alexandros Karatzoglou, Xavier Amatriain, Linas Baltrunas, and Nuria Oliver. Multiverse recommendation: n-dimensional tensor factorization for context-aware collaborative filtering. In *Proceedings of the fourth ACM conference on Recommender systems*, pages 79–86. ACM, 2010.

Noam Koenigstein, Gideon Dror, and Yehuda Koren. Yahoo! music recommendations: modeling music ratings with temporal dynamics and item taxonomy. In *ACM conference on Recommender systems*, 2011.

Yehuda Koren. Factorization meets the neighborhood: A multifaceted collaborative filtering model. In *SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD, pages 426–434, 2008.

Yehuda Koren. Collaborative filtering with temporal dynamics. *Communications of the ACM*, 53(4):89–97, 2010.

Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, 2009.

Denis Krompaß, Maximilian Nickel, and Volker Tresp. Querying factorized probabilistic triple databases. In *The Semantic Web–ISWC*, pages 114–129. 2014.

Guang Ling, Michael R Lyu, and Irwin King. Ratings meet reviews, a combined approach to recommend. In *Proceedings of the 8th ACM Conference on Recommender systems*, 2014.

Julian McAuley, Jure Leskovec, and Dan Jurafsky. Learning attitudes and attributes from multi-aspect reviews. In *Data Mining (ICDM), 2012 IEEE 12th International Conference on*, pages 1020–1025. IEEE, 2012.

Julian John McAuley and Jure Leskovec. From amateurs to connoisseurs: modeling the evolution of user expertise through online reviews. In *International conference on World Wide Web (WWW)*, 2013.

[3] `http://nitishgupta.github.io/factorDB/`

Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. A three-way model for collective learning on multi-relational data. In *International Conference on Machine Learning (ICML)*, pages 809–816, 2011.

Feng Niu, Benjamin Recht, Christopher R, and Stephen J. Wright. Hogwild: A lock-free approach to parallelizing stochastic gradient descent. In *Neural Information Processing Systems (NIPS)*, 2011.

Martin F. Porter. An algorithm for suffix stripping. *Program*, 14(3), 1980.

Steffen Rendle and Lars Schmidt-Thieme. Pairwise interaction tensor factorization for personalized tag recommendation. In *Proceedings of the third ACM international conference on Web search and data mining*, pages 81–90. ACM, 2010.

Ruslan Salakhutdinov and Andriy Mnih. Probabilistic matrix factorization. In *Advances in Neural Information Processing Systems (NIPS)*, volume 20, 2008.

Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. Application of dimensionality reduction in recommender system-a case study. Technical report, DTIC Document, 2000.

Ajit P Singh and Geoffrey J Gordon. Relational learning via collective matrix factorization. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 650–658. ACM, 2008.

Sameer Singh and Thore Graepel. Automated probabilistic modeling for relational data. In *ACM Conference of Information and Knowledge Management (CIKM)*, 2013.

Panagiotis Symeonidis, Alexandros Nanopoulos, and Yannis Manolopoulos. Tag recommendations based on tensor dimensionality reduction. In *Proceedings of the 2008 ACM conference on Recommender systems*, pages 43–50. ACM, 2008.

Ben Taskar, Abbeel Pieter, and Daphne Koller. Discriminative probabilistic models for relational data. In *Uncertainty in Artificial Intelligence (UAI)*, 2002.

Laurens van der Maaten and Geoffrey E. Hinton. Visualizing high-dimensional data using t-sne. *Journal of Machine Learning Research*, 9:2579–2605, 2008.

Daisy Zhe Wang, Eirinaios Michelakis, Minos Garofalakis, and Joseph M. Hellerstein. Bayesstore: Managing large, uncertain data repositories with probabilistic graphical models. In *Conference on Very Large Data Bases (VLDB)*, 2008.

Li-Tung Weng, Yue Xu, Yuefeng Li, and Richi Nayak. Exploiting item taxonomy for solving cold-start problem in recommendation making. In *International Conference on Tools with Artificial Intelligence (ICTAI)*, 2008.