Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17)

Non-Negative Inductive Matrix Completion for Discrete Dyadic Data

Piyush Rai
Dept. of Computer Science and Engineering
IIT Kanpur, India
piyush@cse.iitk.ac.in

Abstract

We present a non-negative inductive latent factor model for binary- and count-valued matrices containing dyadic data, with side information along the rows and/or the columns of the matrix. The side information is incorporated by conditioning the row and column latent factors on the available side information via a regression model. Our model can not only perform matrix factorization and completion with side-information, but also infers interpretable latent topics that explain/summarize the data. An appealing aspect of our model is in the full local conjugacy of all parts of the model, including the main latent factor model, as well as for the regression model that leverages the side information. This enables us to design scalable and simple to implement Gibbs sampling and Expectation Maximization algorithms for doing inference in the model. Inference cost in our model scales in the number of nonzeros in the data matrix, which makes it particularly attractive for massive, sparse matrices. We demonstrate the effectiveness of our model on several real-world data sets, comparing it with state-of-the-art baselines.

Introduction

Matrix factorization of partially observed matrices having binary- or count-valued observations is ubiquitous in many applications involving dyadic data, such as recommender systems (Gopalan, Hofman, and Blei 2013), text analysis (Wang and Blei 2011; Gopalan, Charlin, and Blei 2014), social/biological network analysis (Zhu 2012; Zhou 2015), and so on. Often, in these applications, additional side information may be available along the rows and/or the columns of the data matrix (Wang and Blei 2011; Kim, Hughes, and Sudderth 2012; Gopalan, Charlin, and Blei 2014). Leveraging this information can be especially useful in cases when the data matrix is highly sparse, and in handling the cold-start problem where the data matrix may not have any observations along some of the rows/columns, a critical problem in modern recommender systems (Wang and Blei 2011; Gopalan, Charlin, and Blei 2014).

Most of the existing methods assume the data and/or the latent factors to be real-valued (Agarwal and Chen 2009; Park, Kim, and Choi 2013; Kim and Choi 2014). Many matrix factorization problems, however, involve discrete-valued data (e.g., binary or counts). Moreover, often we want interpretability in the learned latent factors and real-valued latent factors may not be useful. Although some recent methods (Wang and Blei 2011; Gopalan, Charlin, and Blei 2014) have proposed methods to incorporate meta-data in specific types of discrete data, e.g., ratings with text information about products in recommender systems, a framework for incorporating general types of side information for matrix factorization of data is currently lacking.

We present a fully Bayesian framework for non-negative matrix factorization of discrete data, where we also have additional side information, in form of *arbitrary* types of features, along the rows and/or the columns of the data matrix. Our model is based on conditioning the row/column latent factors on these observed features via a flexible regression model. In addition to leveraging the side information, our model has the following properties that distinguish it from other existing methods for matrix factorization with side information: (1) both count as well as binary matrices can be handled under a unified approach which models count data via a Poisson latent factor model and binary data via a *truncated* Poisson latent factor model; (2) our model learns non-negative latent factors which are easily interpretable (e.g., can be thought of as corresponding to genres or topics, when modeling ratings or text data); (3) our model enjoys full local conjugacy which allows designing an efficient Gibbs sampler as well as Expectation Maximization algorithm for doing inference; and (4) computational cost for both count as well as binary data case scales in the number of nonzeros in the data matrix, which makes it very efficient for sparse matrices.

Our framework is also sufficiently general and can be easily adapted to solve a number of other specialized problems, such as link prediction with node features (Kim, Hughes, and Sudderth 2012), topic modeling with document meta-data (Mimno and McCallum 2008), or a generalized setting of the well-studied multi-label learning problem (Yu et al. 2014) where in addition to the example features, we may also have label features. At the same time, our model also generalizes recent line on work on Bayesian models for count and binary matrices (Gopalan, Hofman, and Blei 2013; Zhou 2015), which either cannot leverage side information, or can do so only for very specific settings, e.g., count-valued matrix with text based meta-data along the rows/columns (Gopalan, Charlin, and Blei 2014).

The Model

We first briefly describe the basic setup of our framework, which is based on a Poisson latent factor model (for count matrices) and a truncated Poisson latent factor model (for binary matrices), and then describe how to design the *inductive* counterpart of these latent factor models, in order to be able to incorporate side information.

Latent Factor Models for Count/Binary Data

We assume that we are given a partially observed count/binary matrix $\mathbf{X}$ of size $N \times M$. We first discuss the case of count-valued $\mathbf{X}$ (binary case discussed subsequently). For count-valued $\mathbf{X}$, we assume each entry of $\mathbf{X}$ to be drawn from a Poisson latent factor model (Zhou et al. 2012; Gopalan, Charlin, and Blei 2014) as $X_{nm} \sim \text{Poisson}(\sum_{k=1}^{K} \lambda_k u_{nk} v_{mk}) = \text{Poisson}(\boldsymbol{u}_n^\top \Lambda \boldsymbol{v}_m)$,

where $\boldsymbol{u}_n$ and $\boldsymbol{v}_m$ are K-dimensional *non-negative* latent factors for row n and column m of $\mathbf{X}$, respectively, and Λ is a $K \times K$ non-negative diagonal matrix representing the weights of each of the K latent factors. The same can be written in a matrix notation as $\mathbf{X} \sim \text{Poisson}(\mathbf{U}\Lambda\mathbf{V}^\top)$, where $\mathbf{U} = [\boldsymbol{u}_1^\top, \ldots, \boldsymbol{u}_N^\top]^\top \in \mathbb{R}_+^{N \times K}$, $\mathbf{V} = [\boldsymbol{v}_1^\top, \ldots, \boldsymbol{v}_N^\top]^\top \in \mathbb{R}_+^{M \times K}$, and $\Lambda = \text{diag}(\lambda_1, \ldots, \lambda_K)$, with $\lambda_k \in \mathbb{R}_+$, $\forall k = 1, \ldots, K$. Gamma priors can be placed on $\boldsymbol{u}_n$, $\boldsymbol{v}_m$, and Λ to yield a non-negative factorization.

When $\mathbf{X}$ is a *binary* matrix, we model each binary observation X_{nm} using a *truncated* Poisson latent factor model (Zhou 2015; Hu, Rai, and Carin 2015), which first draws a *latent* count $Z_{nm} \sim \text{Poisson}(\sum_{k=1}^{K} \lambda_k u_{nk} v_{mk})$ and then thresholds it to generate X_{nm} as $X_{nm} = \mathbf{1}(Z_{nm} \geq 1)$. These two steps can be equivalently expressed as $X_{nm} \sim \text{Bernoulli}\left(1 - \exp(-\sum_{k=1}^{K} \lambda_k u_{nk} v_{mk})\right)$. This model is referred to as the Bernoulli-Poisson link (Zhou 2015; Hu, Rai, and Carin 2015). The same can be written in a matrix notation as $\mathbf{X} \sim \text{Bernoulli}(1 - \exp(-\mathbf{U}\Lambda\mathbf{V}^\top))$.

A particularly appealing aspect of the Bernoulli-Poisson link function for binary data is that the conditional posterior Z_{nm} can be written as $Z_{nm} \sim X_{nm} \cdot \text{Poisson}_+(\sum_{k=1}^{K} \lambda_k u_{nk} v_{mk})$, where $\text{Poisson}_+()$ denotes the zero-truncated Poisson distribution. If $X_{nm} = 0$ then the latent count Z_{nm} is also zero with probability one (Zhou 2015), and therefore, during model inference, the latent counts Z_{nm} only need to be estimated for the nonzero observations in $\mathbf{X}$. This can lead to huge computational savings for massive but sparse matrices with very few nonzero observations. This makes the Bernoulli-Poisson model especially attractive for modeling binary data, as opposed to logistic/probit models in which inference scales in the number of both zeros and nonzeros in the data.

Inductive Non-negative Latent Factor Model

Although the Poisson and truncated Poisson latent factor models provide a rich and flexible framework for learning non-negative low-rank factorizations of count/binary matrices, these models are unable to incorporate side information that may be available along the rows and/or columns of the matrix $\mathbf{X}$. The ability of leveraging the side information may be especially desirable in cases where the matrix $\mathbf{X}$ has a significantly large fraction of entries as missing, or in cold-start settings (Wang and Blei 2011; Gopalan, Charlin, and Blei 2014) where some rows/columns in $\mathbf{X}$ may not have any observations. Although some existing Poisson latent factor models can imcorporate specific types of side information (e.g., text based side information (Wang and Blei 2011; Gopalan, Charlin, and Blei 2014)), these models cannot incorporate more general types of features as side information (the Related Work section discusses other models that can incorporate side information in matrix factorization).

With this desideratum, we propose a generalization of the models described in the previous section, which allows us to incorporatw more general forms of side information given as observed *features* along the rows and/or columns of $\mathbf{X}$. We accomplish this by augmenting the Poisson/truncated-Poisson latent factor model with a regression model that connects the observed features to the latent factors $\boldsymbol{u}_n$, $\boldsymbol{v}_m$ in a statistically clean manner, while keeping inference simple and computationally efficient.

We assume that the side information is given in form of feature matrices (Fig. 1-left). Along the rows of $\mathbf{X}$, we are given an $N \times D_u$ feature matrix $\mathbf{\Phi} = [\boldsymbol{\phi}_1, \ldots, \boldsymbol{\phi}_N]^\top$, where $\boldsymbol{\phi}_n \in \mathbb{R}^{D_u}$ denotes the features given along row n in $\mathbf{X}$. Along the columns of $\mathbf{X}$, we are given an $M \times D_v$ feature matrix $\mathbf{\Psi} = [\boldsymbol{\psi}_1, \ldots, \boldsymbol{\psi}_M]^\top$, where $\boldsymbol{\psi}_m \in \mathbb{R}^{D_v}$ denotes the features given along column m in $\mathbf{X}$.

To incorporate the side information, we parameterize the row and column latent factors $\boldsymbol{u}_n$ and $\boldsymbol{v}_m$ (each of which is given a gamma prior) by conditioning $\boldsymbol{u}_n$ and $\boldsymbol{v}_m$ on the row and column features $\boldsymbol{\phi}_n$ and $\boldsymbol{\psi}_m$, respectively (Fig. 1-right). To this end, we parameterize the scale parameter of the gamma priors on these row/column features. Specifically, we model each row latent factor $\boldsymbol{u}_n$ as

$$\begin{aligned} u_{nk}|\boldsymbol{\phi}_n &\sim \text{Gamma}(r_n^{(u)}, p_{nk}^{(u)}/(1 - p_{nk}^{(u)})) \\ p_{nk}^{(u)} &= \sigma({\boldsymbol{w}_k^{(u)}}^\top \boldsymbol{\phi}_n + b_k^{(u)}) \end{aligned}$$

where the regression weight vector $w_k^{(u)} \in \mathbb{R}^{D_u}$, bias $b_k^{(u)} \in \mathbb{R}$, and $\sigma()$ denotes the logistic function. Note that the above can also be written as

$$u_{nk}|\boldsymbol{\phi}_n \sim \text{Gamma}(r_n^{(u)}, \exp({\boldsymbol{w}_k^{(u)}}^\top \boldsymbol{\phi}_n + b_k^{(u)}))$$

Likewise, the column latent factors $\boldsymbol{v}_m$

$$\begin{aligned} v_{mk}|\boldsymbol{\psi}_m &\sim \text{Gamma}(r_m^{(v)}, p_{mk}^{(v)}/(1 - p_{mk}^{(v)})) \\ p_{mk}^{(v)} &= \sigma({\boldsymbol{w}_k^{(v)}}^\top \boldsymbol{\psi}_m + b_k^{(v)}) \end{aligned}$$

We collectively denote the regression weight vectors for the row and the column latent factors by matrices $\mathbf{W}^{(u)} = \{\boldsymbol{w}_k^{(u)}\}_{k=1}^K$ and $\mathbf{W}^{(v)} = \{\boldsymbol{w}_k^{(v)}\}_{k=1}^K$, respectively. These are given zero mean, sparsity inducing ARD priors (Tipping 2001). Moreover, note that we also have separate shape parameters $\{r_n^{(u)}\}_{n=1}^N$ and $\{r_m^{(v)}\}_{m=1}^M$ in the gamma priors of each of the row latent factors and each of the column latent factors, to model the specificity of each row and each column latent factor. The full generative model described in the next section. Finally, although not our focus in this work, our

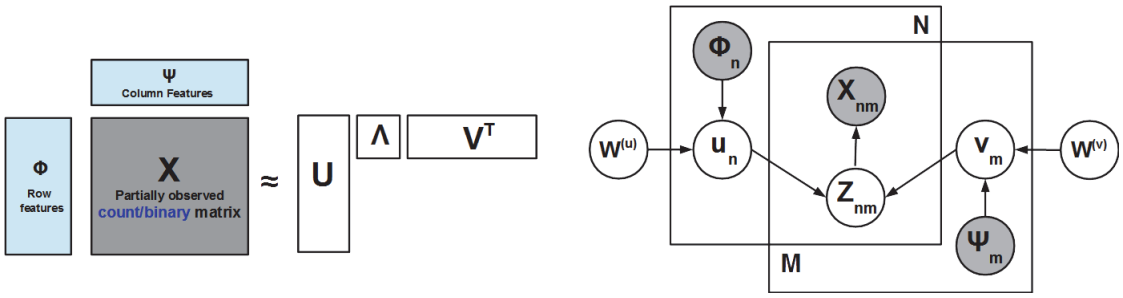

Figure 1: (Left) A high-level illustration of count/binary matrix factorization with side information given as features along rows and columns (the $\approx$ is only meant to denote the underlying process). (Right) The graphical model for the proposed inductive non-negative latent factor model (hyperparameters not shown for brevity).

framework can be readily extended into a deep architecture by modeling the gamma-distributed shape parameters using a gamma belief-net (Zhou, Cong, and Chen 2015).

The Full Generative Model

When $\mathbf{X}$ is a count matrix, we model it as

$$X_{nm} \sim \text{Poisson}(\sum_{k=1}^{K} \lambda_k u_{nk} v_{mk}) \quad (1)$$

When $\mathbf{X}$ is a binary matrix, we model it as

$$X_{nm} \sim \text{Bernoulli}(1 - \exp(-\sum_{k=1}^{K} \lambda_k u_{nk} v_{mk})) \quad (2)$$

The generative model for the rest of the parameters is

$$\begin{aligned}
u_{nk} &\sim \text{Gamma}(r_n^{(u)}, p_{nk}^{(u)}/(1-p_{nk}^{(u)})) && (3)\\
v_{mk} &\sim \text{Gamma}(r_m^{(v)}, p_{mk}^{(v)}/(1-p_{mk}^{(v)})) && (4)\\
r_n^{(u)} &\sim \text{Gamma}(a_0, 1/b_0) && (5)\\
r_m^{(v)} &\sim \text{Gamma}(c_0, 1/d_0) && (6)\\
p_{nk}^{(u)} &= \sigma({\boldsymbol{w}_k^{(u)}}^\top \boldsymbol{\phi}_n + b_k^{(u)}) && (7)\\
p_{mk}^{(v)} &= \sigma({\boldsymbol{w}_k^{(v)}}^\top \boldsymbol{\psi}_m + b_k^{(v)}) && (8)\\
\boldsymbol{w}_k^{(u)} &\sim \mathcal{N}(0, \Gamma_k^{(u)}), \quad \boldsymbol{w}_k^{(v)} \sim \mathcal{N}(0, \Gamma_k^{(v)}) && (9)\\
\lambda_k &\sim \text{Gamma}(e_0/K, 1/f_0) && (10)
\end{aligned}$$

In the above, $\Gamma_k^{(u)}$ and $\Gamma_k^{(v)}$ denote the diagonal covariance matrices of the Gaussian priors on the regression weight vectors of row and column latent factors, respectively. Each diagonal entry of $\Gamma_k^{(u)}$ and $\Gamma_k^{(v)}$ is given an inverse-gamma prior, resulting in an ARD-like prior (Tipping 2001) on the regression weights, which gives the model robustness against irrelevant/noisy features in the side information. The biases $b_k^{(u)}$ and $b_k^{(u)}$ are also given Gaussian priors and their respective variances are given inverse-gamma priors and are learned from data. The gamma shape/scale hyperparameters in Eq. (5), (6), and (10) are set to yield uninformative gamma priors.

Note that, unlike existing inductive matrix completion methods (Natarajan and Dhillon 2014), our model does not solely rely on the side-information; the parametrization of the scale parameters $p_{nk}^{(u)}$ and $p_{mk}^{(v)}$ also includes the bias terms which allows capturing the structural properties (e.g., low-rank) of $\mathbf{X}$ even when there is no side information. In this case, the model reduces to a standard gamma-Poisson latent factor model (Zhou et al. 2012).

Inference

Inference in our model require inferring the latent factors $\{\boldsymbol{u}_n\}_{n=1}^N$, $\{\boldsymbol{v}_m\}_{m=1}^M$, the regression weight vectors and biases $\{\boldsymbol{w}_k^{(u)}, b_k^{(u)}\}_{k=1}^K$ and $\{\boldsymbol{w}_k^{(v)}, b_k^{(v)}\}_{k=1}^K$, and the other latent variables and hyperparameters $\{\lambda_k\}_{k=1}^K$, $\{r_n^{(u)}\}_{n=1}^N, \{r_m^{(v)}\}_{m=1}^M$, $\{\Gamma_k^{(u)}, \Gamma_k^{(v)}\}_{k=1}^K$.

An appealing aspect of our framework is simplicity of the model inference despite the richness of the model. In particular, as we will show, using data augmentation techniques allows us to derive closed form Gibbs sampling updates for all the model parameters. Also, as we will show, inference in our model scales in the number of nonzeros in $\mathbf{X}$, for both count as well as binary $\mathbf{X}$, which leads to excellent scalability even on large but sparse matrices which usually have very few nonzero entries. In contrast, other likelihood models for binary matrices (e.g., logistic/probit) scale in the total number of observations, which includes both zeros and nonzeros.

In the rest of this section, when giving the inference update equations for various model parameters, we will assume $\mathbf{X}$ to be count-valued. If $\mathbf{X}$ is binary then we will maintain another latent count Z_{nm} associated with the corresponding entry X_{nm} in $\mathbf{X}$ (cf, Fig. 1, right). This latent count Z_{nm} will be drawn from a truncated Poisson distribution and it only needs to be done for the nonzero entries in $\mathbf{X}$. For the binary case, quantities with notation X's in equations below will be replaced by corresponding Z's.

Gibbs Sampling

Sampling Latent Factors. Inferring the latent factors $\{\boldsymbol{u}_n\}_{n=1}^N$ and $\{\boldsymbol{v}_m\}_{m=1}^M$ is straightforward in our model due to the Poisson-gamma conjugacy. In particular, using Poisson additivity, each count-valued X_{nm} can be written as a sum of K smaller counts; e.g., $X_{nm} = \sum_{k=1}^K X_{nmk}$ which in turn can be thought of as draws from another Poisson (Dunson and Herring 2005)

$$X_{nmk} \sim \text{Poisson}(\lambda_k u_{nk} v_{mk}) \quad (11)$$

Moreover, using the Poisson-multinomial equivalence (Zhou et al. 2012), given X_{nm}, the latent counts $\{X_{nmk}\}_{k=1}^K$ can be also drawn from a multinomial as

$$[X_{nm1}, \ldots, X_{nmK}] \sim \text{Mult}(X_{nm}; \zeta_{nm1}, \ldots, \zeta_{nmK})$$

where $\zeta_{nmk} = \frac{\lambda_k u_{nk} v_{mk}}{\sum_{k=1}^K \lambda_k u_{nk} v_{mk}}$. Here X_{nm} (cf, Z_{nm} when X_{nm} is binary) itself is drawn from a Poisson (cf., truncated Poisson) distribution with rate $\sum_{k=1}^K \lambda_k u_{nk} v_{mk}$. Again note that this needs to be done only for the nonzeros observations in $\mathbf{X}$, which makes inference very efficient.

Using Eq. 11, we define $X_{n.k} = \sum_{m=1}^M X_{nmk} \sim \text{Poisson}(\lambda_k u_{nk} \sum_{m=1}^M v_{km})$. Using gamma-Poisson conjugacy, we draw each row latent factors u_{nk} as $u_{nk} \sim \text{Gamma}\left(r_n^{(u)} + X_{n.k}, \frac{p_{nk}^{(u)}}{(1-p_{nk}^{(u)}) + p_{nk}^{(u)} \sum_{m=1}^M \lambda_k v_{mk})}\right)$, where $p_{nk}^{(u)} = \sigma({w_k^{(u)}}^\top \boldsymbol{\phi}_n)$. The posterior distribution over the column latent factors v_{mk} has the same form. We provide the detailed equations in the Supplementary Material.

Sampling Regression Weights. The Gaussian priors on the regression weight vectors $\{\boldsymbol{w}_k^{(u)}, \boldsymbol{w}_k^{(v)}\}_{k=1}^K$ are not readily conjugate to the Poisson likelihood. However, note that we have $X_{n.k} \sim \text{Poisson}(\lambda_k u_{nk} \sum_{m=1}^M v_{km})$, and if we integrate out u_{nk} which, in turn, is drawn from a gamma, we get a negative Binomial marginal distribution for $X_{n.k}$ (note: this negative Binomial distribution will be on $Z_{n.k}$ when $\mathbf{X}$ is binary). For models with Gaussian priors on the latent variables, and non-conjugate negative Binomial likelihoods that have the log-odds expressible as a linear function of the latent variables, we can use the Pólya-Gamma (PG) strategy (Polson, Scott, and Windle 2013) to transform the non-conjugate likelihood into a conjugate Gaussian likelihood using a set of Pólya-Gamma auxiliary variables, and get closed form Gaussian posterior over the latent variables (more details in the Supplementary Material).

The other model parameters can also be sampled easily due to conjugacy (or conjugacy obtained via appropriate data-augmentation techniques (Zhou and Carin 2015)). We skip the sampling equations here for brevity and provide these in the Supplementary Material.

An Expectation Maximization Algorithm

The Gibbs sampler described above is easy to implement and is efficient to run. However, for very large data sets, Gibbs sampling can still be slow and it may take a long time for the Markov chain to mix. Moreover, note that the Gibbs sampler, for sampling the regression weights, requires sampling the Pólya-Gamma auxiliary variables which can be slow for large data sets. A more efficient alternative can be an Expectation Maximization (EM) algorithm, which is particularly more efficient for our model bacause the Pólya-Gamma random variables have expectations that can be efficiently and analytically computed (Scott and Sun 2013).

The expectations of other variables are also easy to compute in closed form. In the M step, we can estimate the regression weights $\boldsymbol{w}_k^{(u)}$ by solving a simple linear system or using conjugate gradients. More details are provided in the Supplementary Material.

Dyad Predictions along *New* Rows/Columns in X

An appealing aspect of our model is that, given $\mathbf{V}$, and given a new row in $\mathbf{X}$ with features $\boldsymbol{\phi}_*$, we can directly predict the dyads along this row without inferring the corresponding row latent factors $\boldsymbol{u}_*$, by marginalizing out $\boldsymbol{u}_*$. This property makes prediction efficient at test time. For example, for the binary case, the dyad prediction for the m-th entry in this row

$$p(X_{*m} = 1 | \boldsymbol{\phi}_*, \mathbf{V}) = 1 - \prod_{k=1}^K \left[v_{mk} \exp({\boldsymbol{w}_k^{(u)}}^\top \boldsymbol{\phi}_*) + 1 \right]^{-r_k^{(u)}}$$

The above equation can be easily obtained by integrating out the embedding $\boldsymbol{u}_*$ from the Bernoulli-Poisson likelihood. Predicting the dyads along a new columns given its features $\boldsymbol{\psi}_*$ can be done in a similar manner.

Related Work

Matrix factorization in the presence of side information has also been investigated in other prior works such as regression based latent factor models (Agarwal and Chen 2009), Inductive matrix completion (Natarajan and Dhillon 2014; Chiang, Hsieh, and Dhillon 2015), and other extensions of latent factor models (Park, Kim, and Choi 2013; Kim and Choi 2014; Adams, Dahl, and Murray 2010). However, most of these methods usually have one or more of the following limitations: (1) the data is assumed to be real-valued; (2) inference cost depends on the size of the matrix rather than just on the number of nonzeros; and (3) the factors do not have nice "topic like" interpretability unlike our model.

Relatively much less work exists on matrix factorization of discrete-valued data with side information. Some of the existing methods include Poisson matrix factorization with text-based meta-data (Gopalan, Charlin, and Blei 2014) or specialized extensions of topic models (Wang and Blei 2011). However, these methods can either be only applied to very specific settings (e.g., ratings data with specific types of side information such as text (Gopalan, Charlin, and Blei 2014; Wang and Blei 2011)), and/or require computationally intensive inference (Kim, Hughes, and Sudderth 2012).

Side information in the context of matrix factorization/completion problems can also be incorporated using methods based on collective matrix factorization (CMF) methods (Bouchard, Yin, and Guo 2013) or multiview matrix factorization with the side information representing another view of the data. However, these methods tend to work well when the side information is very informative and the main matrix as well as the side information be explained by a set of common latent factors. This may not be the case when the side information is usually a limited and noisy set of features.

Our approach of conditioning gamma distributed latent factors using features was also considered recently in (Rai et al. 2015) in the context multi-label learning, whereas our focus here is on the more general problem setting of inductive non-negative latent factor models and matrix completion.

In contrast to the methods described above, our framework is significantly more general (handles both count and binary matrices in a unified manner via Poisson/truncated Poisson factor models), handles missing data easily, allows incorporating arbitrary types of row/column features via a regression model in a robust manner, and has computational cost that scales only in the number of nonzeros in the data matrix. Moreover, our framework is fully Bayesian with a simple to implement Gibbs sampling as well as EM algorithm, and the inference method can be easily extended to online Gibbs sampling (Guhaniyogi, Qamar, and Dunson 2014) or online EM (Scott and Sun 2013) for better efficiency.

Experiments

We evaluate our model by performing experiments on a wide variety of data sets, on both quantitative tasks (matrix completion with side information) as well as qualitative analyses (interpretability of the inferred latent factors). We compare our model with three baselines: (1) gamma-Poisson latent factor model (GPLFM) (Zhou et al. 2012) which is similar in construction to our model but cannot leverage side information; (2) Regression-based Latent Factor Model (RLFM) (Agarwal and Chen 2009); and (3) inductive matrix completion (Chiang, Hsieh, and Dhillon 2015), a state-of-the-art model, which is similar in spirit to our model and can leverage side information. We denote our model by NILFM (for **N**on-negative **I**nductive **L**atent **F**actor **M**odel). The data sets used in our experiments include:

Drug-Target: The drug-target interaction network [1] represents binary-valued interactions between 200 drug molecules and 150 target proteins. As the side information, we have drug and target features representing information from chemical structure similarity and amino acid sequence, respectively.

Lazega-Lawyers: We consider the "advising" relation in the Lazega lawyers dataset [2] consisting of 71 partners and associates. In addition to advising relation, each entity in the network is described by 7 features such as gender, office-location, age, years employed, etc.

Movielens [3]: We use two versions of this data: Movielens-100K and Movielens-1M. Movielens-100K is a 943×1682 matrix containing 100K ratings by 943 users on 1682 movies. Movielens-1M is a 6040×3942 matrix containing 1 million ratings by 6040 users on 3942 movies. In addition, we have 29 features for each user and 18 features for each movie. This data contains ratings on a scale of 1-5 which we convert to 0-1 based on whether it is greater than 2.

NIPS-1-17: This data set contains a list of all the papers and authors from the collection NIPS 1-17, along with the list of keywords in each paper. We have the document-author binary matrix of size 2484×2865. In addition, we also have the list of keywords associated with each paper and with each author. We perform an SVD on papers-words and authors-words matrices to construct 100 features that we use as side information for the papers and the authors.

Cora [4]: This data set is a citation network consisting of a total of 2708 research papers. In addition, for each paper, a label denoting its research area is also given as side information. We convert the label of each paper to a one-hot encoding and use it as the side information associated with that paper.

Experimental Settings. For NILFM, we use Gibbs sampling for model inference. For NILFM as well as GPLFM, the Gibbs sampler is run for 2000 iterations with 1000 post-burnin collection samples. Our EM based inference algorithm also yields almost similar results as Gibbs sampling based inference (while being much faster); however, since the Gibbs sampler for our model is fast enough, we only used this in our experiments. In all our experiments, K was set to 20 which worked well in practice for all the data sets. Note that the shrinkage prior on λ_k effectively prunes out the uncessarily components by shrinking λ_k to close to zero (Zhou et al. 2012). All the model parameters for GPLFM as well as for our model NILFM are initialized randomly.

Purely Inductive Matrix Completion

We first perform an experiment with the purely inductive setting of matrix completion (or the so-called "cold-start" setting) where we train the model using 20% of the rows and 20% of the columns in $\mathbf{X}$, and predict the dyads corresponding to the rows and columns that remain unseen during training time, using only the side information available for these rows and columns. For Cora data, however, we used 50% as the performance of other baselines was unstable when using only 20% training data. Each experiment is repeated 5 times with different training/test splits and we report the averaged area under the ROC curve (AUC) for all the data sets. Table 1 shows the results on this task.

As Table 1 shows, NILFM is the best performing method on almost all the data sets. Also note that NIFLM performs consistently better than GPLFM (which cannot incorporate side information) by a large margin in most cases, especially Cora data where the class-id of each paper seems to turn out to be a strong source of side information. NILFM also outperforms IMC on most of the cases, as well as RLFM. Note that IMC strongly relies on features being strongly informative, which may not always be the case (Chiang, Hsieh, and Dhillon 2015). This can be the reason why IMC does not perform so well as compared to our method which can leverage both the structural information in the matrix (low-rank) as well as the side information, to yield considerably better matrix completion accuracies than IMC.

Standard Matrix Completion with Side-Info

We next evaluate the performance of our model in the standard binary matrix completion setting where we do have data from all rows and all columns but each row/column may have a significant amount of data is missing and needs to be predicted. For this experiment, we specifically compare our model with GPLFM (which cannot leverage the side information) to assess how much our model additionally benefits using the side information as compared to an otherwise similar model GPLFM which cannot handle the side information.

[1] http://www.genome.jp/tools/dinies/help.html

[2] https://www.stats.ox.ac.uk/~snijders/siena/Lazega_lawyers_data.htm

[3] http://grouplens.org/datasets/movielens/

[4] https://relational.fit.cvut.cz/dataset/CORA

	Lawyers	Drug	ML-100K	ML-1M	NIPS	Cora
GPLFM	0.6879	0.8002	0.8564	0.8252	0.8131	0.6034
RLFM	0.6892	0.7644	0.9082	0.7424	0.8292	0.8602
IMC	0.6301	0.7040	0.9023	0.6685	**0.8312**	0.8606
NILFM	**0.7752**	**0.8249**	**0.9202**	**0.8382**	0.8260	**0.8652**

Table 1: AUC scores obtained by our method NILFM and the two baselines

Authors (Kernels and Learning Theory)	**Papers (Kernels and Learning Theory)**
J. Weston, J. Shawe-Taylor, T. Darrell, R. Herbrich,	- Kernel Dependency Estimation
O. Chapelle, B. Scholkopf T. Graepel, G. Cottrell,	- Learning with Local and Global Consistency
G. Ratsch, A.J. Smola, N. Cristianini, A. Elisseeff,	- Learning to Find Pre-Images
T. Poggio, M.E. Tipping, B. Moghaddam, V. Vapnik,	- Sampling Techniques for Kernel Methods
A. Pentland, P. Vincent, P. Niyogi, O Bousquet	- On the Complexity of Learning the Kernel Matrix

Table 2: Top few authors and papers in the topic "kernels and learning theory"

Authors (Probabilistic Models)	**Papers (Probabilistic Models)**
G. Hinton, J. Tenenbaum, R. Zemel, T. Jaakkola,	- Discovering Hidden Variables..
L. Hansen, T. Anastasio, P. Smyth, S. Hanson, D.	- Learning Representations by Recirculation
Dong, J. McClelland, O. Winther, A.J. Storkey,	- Unsupervised learning of distributions on binary..
D. Haussler, S. Schreiner, M. Gluck, D. Lee, H.	- Spherical Units as Dynamic Consequential Regions
Steck	- On the Dirichlet Prior and Bayesian Regularization

Table 3: Top few authors and papers in the topic "probabilistic models"

We use 50% data to train and test on the rest 50%. As Fig. 2 shows, our model NILFM achieves better AUC scores than GPLFM in all the cases. As in the case of purely inductive matrix completion. In some cases where the AUC improvements are marginal, this can be attributed to the fact that the side information is not very strong.

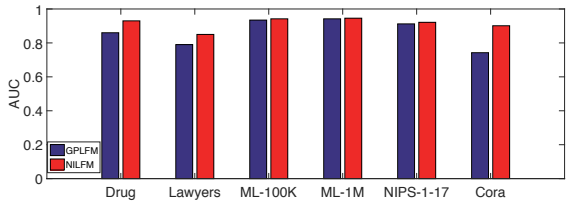

Figure 2: Comparison of GPLFM and NILFM on standard matrix completion with side information

Qualitative Results: Latent Factors as Topics

We next perform an experiment to evaluate our model in terms of the interpretability of the inferred latent factors (which are non-negative in our model). The interpretability may be useful for tasks such as identifying clusters/topics in the data based on the inferred latent factors.

For this experiment, we run NILFM (with $K = 10$) on NIPS-1-17 data and take the output of the model and look at each latent factor (which can be treated as a topic). For each row factor (rows correspond to papers) and the corresponding column factor (columns correspond to authors), both of which can be thought of as a "topic", we rank the papers and the authors based on the (non-negative) factor scores. This gives us a ranked list of papers and authors in each topic.

Table 2 and 3 show two of the topics inferred by the model and shows the list of the top few authors and papers within each topic for the NIPS-1-17 corpus. The result shows that the latent factors inferred by our model not only have good predictive power (as shown by the matrix completion experiments) but are also interpretable with clear semantics.

Gibbs Sampling vs EM

Finally, we also perform another experiment to compare the running times of our model using the Gibbs sampler vs using the EM variant. We run our model with each of these inference methods on Cora and Movielens-1M data sets. Fig. 3 shows the per-iteration running times. Note that the numbers reported are obtained using unoptimized MATLAB implementations of these algorithms, and further implementation based optimizations would speed up either variant equally. As Fig. 3 shows, EM is much faster, while giving similar AUC scores on all the data sets (for example, Gibbs sampling on Cora gave an AUC = 0.9012 while EM on Cora gave an AUC = 0.8956).

Conclusion

We have presented a probabilistic, non-negative latent factor model for count/binary matrices, where additional side information may be available along the rows and/or columns of the matrix. Our model scales in the number of non-zeros in the data, and can not only perform matrix factorization and completion with side-information, but also infers interpretable latent topics that explain/summarize the data. It is broadly applicable to a large class of problems such as recommender

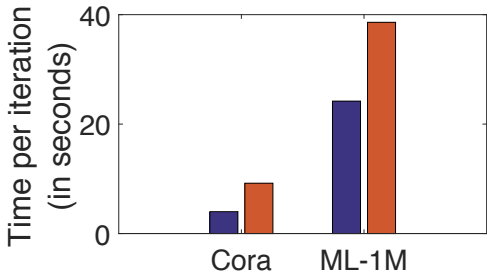

Figure 3: Timings of Gibbs sampler (orange) and EM (blue)

systems and link prediction in networks, while leveraging side information in a simple and clean way, without complicating model inference. Our model can be easily generalized for non-negative tensor decomposition where additional side information may be available in one or more tensor dimensions. Moreover, it can also be applied to other problems such zero-shot, multilabel learning problems where previously unseen classes can be predicted based on the feature descriptions available for such unseen classes.

Acknowledgement: The author thanks support from IBM Faculty Award, Deep Singh and Daljeet Kaur Faculty Fellowship, and the Research I Foundation, IIT Kanpur.

References

Adams, R. P.; Dahl, G. E.; and Murray, I. 2010. Incorporating side information in probabilistic matrix factorization with gaussian processes. *arXiv preprint arXiv:1003.4944*.

Agarwal, D., and Chen, B.-C. 2009. Regression-based latent factor models. In *KDD*, 19–28. ACM.

Bouchard, G.; Yin, D.; and Guo, S. 2013. Convex collective matrix factorization. In *AISTATS*, 144–152.

Chiang, K.-Y.; Hsieh, C.-J.; and Dhillon, I. S. 2015. Matrix completion with noisy side information. In *NIPS*.

Dunson, D. B., and Herring, A. H. 2005. Bayesian latent variable models for mixed discrete outcomes. *Biostatistics*.

Gopalan, P. K.; Charlin, L.; and Blei, D. 2014. Content-based recommendations with poisson factorization. In *NIPS*, 3176–3184.

Gopalan, P.; Hofman, J. M.; and Blei, D. M. 2013. Scalable recommendation with poisson factorization. *arXiv preprint arXiv:1311.1704*.

Guhaniyogi, R.; Qamar, S.; and Dunson, D. B. 2014. Bayesian conditional density filtering. *arXiv preprint arXiv:1401.3632*.

Hu, C.; Rai, P.; and Carin, L. 2015. Zero-truncated poisson tensor factorization for massive binary tensors. In *UAI*.

Kim, Y.-D., and Choi, S. 2014. Scalable variational bayesian matrix factorization with side information. In *AISTATS*, 493–502.

Kim, D. I.; Hughes, M.; and Sudderth, E. B. 2012. The nonparametric metadata dependent relational model. In *ICML*.

Mimno, D., and McCallum, A. 2008. Topic models conditioned on arbitrary features with dirichlet-multinomial regression. In *UAI*.

Natarajan, N., and Dhillon, I. S. 2014. Inductive matrix completion for predicting gene–disease associations. *Bioinformatics* 30(12):i60–i68.

Park, S.; Kim, Y.-D.; and Choi, S. 2013. Hierarchical bayesian matrix factorization with side information. In *IJCAI*.

Polson, N. G.; Scott, J. G.; and Windle, J. 2013. Bayesian inference for logistic models using pólya–gamma latent variables. *Journal of the American Statistical Association* 108(504):1339–1349.

Rai, P.; Hu, C.; Henao, R.; and Carin, L. 2015. Large-scale bayesian multi-label learning via topic-based label embeddings. In *NIPS*.

Scott, J. G., and Sun, L. 2013. Expectation-maximization for logistic regression. *arXiv preprint arXiv:1306.0040*.

Tipping, M. E. 2001. Sparse bayesian learning and the relevance vector machine. *JMLR* 1:211–244.

Wang, C., and Blei, D. M. 2011. Collaborative topic modeling for recommending scientific articles. In *KDD*, 448–456. ACM.

Yu, H.-F.; Jain, P.; Kar, P.; and Dhillon, I. S. 2014. Large-scale multi-label learning with missing labels. In *ICML*.

Zhou, M., and Carin, L. 2015. Negative binomial process count and mixture modeling. *PAMI* 37(2).

Zhou, M.; Hannah, L.; Dunson, D.; and Carin, L. 2012. Infinite edge partition models for overlapping community detection and link prediction. In *AISTATS*.

Zhou, M.; Cong, Y.; and Chen, B. 2015. Gamma belief networks. *arXiv preprint arXiv:1512.03081*.

Zhou, M. 2015. Infinite edge partition models for overlapping community detection and link prediction. In *AISTATS*.

Zhu, J. 2012. Max-margin nonparametric latent feature models for link prediction. In *ICML*.