

---

# Collaborative Filtering Techniques for Document Modeling

---

**Massimo Caccia\***

MILA, Université de Montréal  
massimo.p.caccia@gmail.com

**Lucas Caccia\***

MILA, McGill University  
lucas.page-caccia@mail.mcgill.ca

**Laurent Charlin**

MILA, HEC Montréal

\* Indicates first authors

## Abstract

Collaborative filtering (CF) makes predictions about user preferences for items (e.g., movies, products, restaurants) by exploiting the similarity patterns across users. CF represent users as the set of items they consumed. Similarly, documents are often represented as their set of words. Document modeling methods and CF have however evolved mostly separately. In this work, we apply older and more recent CF methods to the document modeling field. Interestingly, we find those approaches learn more transferable latent representation of documents over the popular Neural Variational Document Model [Miao et al., 2016], as highlighted by the important gains on a document classification transfer task. Lastly, we provided a qualitative analysis of the latent variables in exposure matrix factorization [Liang et al., 2016] when applied to documents.

## 1 Introduction

Unsupervised document models or topic models refers to a type of statistical algorithms for discovering latent representations or *abstract* topics in a collection of documents. They allow for clustering of large number of (unlabeled) documents. Although recurrent neural networks (RNN) have enjoyed tremendous success in NLP, bag-of-words (BoW) are still widely used to represent large documents because of memory and training constraints. The recommendation system (RecSYS) domain also employs BoW representations. Precisely, users are represented as the set of items they consumed (because the models employed in this field are often invertible, one can also think of an item as the set of users that consumed them). Exploiting similarity patterns across users to infer their preferences is often referred to as collaborative filtering (CF). Albeit those two fields share lots of commonalities, they both have evolved somewhat separately. In this work, we bridge the gap between both domains by applying both standard and more recent CF models to the task of document modeling. Concretely, we apply matrix factorization, weighted matrix factorization [Hu et al., 2008] and exposure matrix factorization [Liang et al., 2016] to BoW representations of documents. To evaluate the relevance of such approaches for topic modeling, we compare their performance with the Neural Variational Document Model (NVDM) [Miao et al., 2016]. To the best of our knowledge, NVDM achieves state-of-the-out log-likelihoods on BoW document representations. Because matrix factorization models don't share the same probabilistic space as standard language models (see Section 4 for more details) we measure the models against each other on a downstream document classification task. Specifically, a linear classifier is trained on the documents' representations in order to predict their classes. The popular *20news group* dataset is used.

Remarkably, we find that recently proposed matrix factorization managed to outperform deepers models, i.e. NVDM. Thus, downweighting (un)observations is an extremely powerful tool for document modeling. Additional to the quantitative results, we provide a qualitative analysis of the latent variables in exposure matrix factorization [Liang et al., 2016]. In Section 2, we go over old and more recent developments in CF. Section 3 reviews standard document modeling approaches. Section 4 compares and contrast both set of techniques. We present quantitative and qualitative results in Section 5.

## 2 Collaborative Filtering Techniques

In a typical recommendation system setting, each user  $u = 1, \dots, U$  consumes items  $i = 1, \dots, I$  and the task of interest is to suggest new items that a user would be interested in. The observations can be encoded in a sparse  $\mathbf{Y} = \{y_{ui}\}$  matrix, often called *consumption matrix*. In this work, we focus on the implicit data case as it is easily applicable to document compression. The most common way to frame this task is to model the users and items in such a way that we can predict the consumption (or not) of held-out user-items pairs. Importantly, all of the users preferences/representations are learned at training time. This setting is often referred to as *weak* generalization. In a strong generalization setting, the model would learn new users' representation at test time while keeping the items' attribute fixed [Marlin, 2004].

To test the generalization capabilities of the learned models, information retrieval metrics (for example, precision and recall) are standard. In this work, we will use recall (Recall@ $k$ ), mean average precision (MAP@ $k$ ) and normalized discounted cumulative gains (NDCG@ $k$ ). For details on the metrics, We refer the reader to Liang et al. [2016]. In the context of document modeling, these metrics will assess document completion abilities of a model.

Next, we describe models that have been successfully applied to collaborative filtering problems and that will be pertinent to our analysis. All models are part of the *matrix factorization* family. This list is non-exhaustive.

### 2.1 Matrix Factorization

Matrix factorization (MF) is the standard approach to collaborative filtering [Koren et al., 2009]. Given the observed entries in the *consumption matrix*  $Y$ , matrix factorization models infer latent user preferences and item attributes by factorizing  $Y$ . They assume that the observations are drawn from a specific distribution e.g. a Gaussian or a Poisson when  $y_{ui} \in \mathbb{N}$  or a Bernoulli when  $y_{ui} \in \{0, 1\}$ . For the rest of the paper, we will focus on the Gaussian case. Formally, Gaussian matrix factorization [Mnih and Salakhutdinov, 2007] is:

$$\begin{aligned}\boldsymbol{\theta}_u &\sim \mathcal{N}(\mathbf{0}, \lambda_\theta^{-1} I_K) \\ \boldsymbol{\beta}_i &\sim \mathcal{N}(\mathbf{0}, \lambda_\beta^{-1} I_K) \\ y_{ui} &\sim \mathcal{N}(\boldsymbol{\theta}_u^\top \boldsymbol{\beta}_i, \lambda_y^{-1}),\end{aligned}$$

where  $\theta_u$  and  $\beta_i$  represent the user and item latent representations, or users' preference and item's attributes in recommendation system parlance. The variance of the Gaussian distributions are parametrized by the  $\lambda$ s and are treated as hyperparameters controlling the regularization.  $I_K$  stands for the identity matrix of dimension  $K$ . These models can easily be understood from a generative model perspective. First, users preference  $\theta_u$  and item attributes  $\beta_i$  are drawn. Next, the observations  $y_{ui}$  are drawn from the previously chosen distribution i.e. Gaussian in our case. A graphical representation of this process is shown in Figure 1a.

When applied to documents,  $\theta_u$  represents the document latent representations and  $\beta_i$  the word embeddings. In MF, both documents and words are encoded in the same  $\mathbb{R}^K$  latent space.

### 2.2 Weighted Matrix Factorization

In implicit data a missing observation can indicate a negative preference for an item, but it could also indicate that a user does not know about a particular item. To model this ambiguity, weighted matrix factorization (WMF) [Hu et al., 2008] proposes to downweight all unobserved user-item interactions:

$$y_{ui} \sim \mathcal{N}(\boldsymbol{\theta}_u^\top \boldsymbol{\beta}_i, c_{y_{ui}}^{-1}),$$

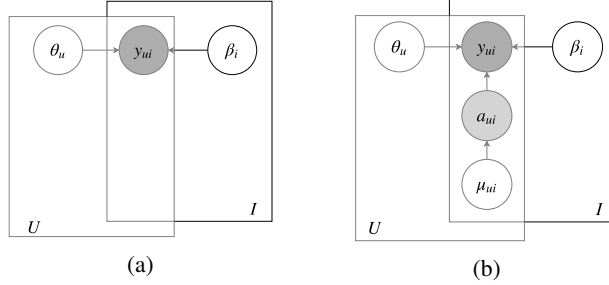


Figure 1: **a)** Plate notation of the matrix factorization model. **b)** Plate notation of the exposure matrix factorization model Liang et al. [2016].

where the "confidence"  $c$  is set such that  $c_1 > c_0$ . Because of the dependency between a user-item interaction and itself, WMF is not a valid generative model. In section 2.3, we will explain how to modify it such that it becomes one.

Going back to the document modeling analogy, WMF factorizes the document-word matrix while underweighting the omitted (unobserved) words in the loss function.

### 2.3 Exposure Matrix Factorization

Inferring (latent) users' preferences and items' attributes from implicit data has its downside. For example, what if a user lives too far away from a restaurant that he would otherwise like, or what if a user is not aware of a movie that he would have otherwise enjoyed. MF and WMF would interpret this (un)observation as a *dislike* and would learn latent representations accordingly. Exposure matrix factorization (Expo-MF) [Liang et al., 2016] addresses this problem. The model infers if a user was *exposed* or not to the an unconsumed item. E.g., if we go back to the restaurant example the model could underweight the (un)observation of the restaurant for which the user's preferences points to liking. Concretely,  $A = \{a_{ui}\}$  represents the exposure matrix, which is part observable (when  $y_{ui} > 1$ ) part latent (when  $y_{ui} = 0$ ). Formally, Expo-MF is :

$$\begin{aligned}
 \theta_u &\sim \mathcal{N}(\mathbf{0}, \lambda_\theta^{-1} I_K) \\
 \beta_i &\sim \mathcal{N}(\mathbf{0}, \lambda_\beta^{-1} I_K) \\
 a_{ui} &\sim \text{Bernoulli}(\mu_{ui}) \\
 y_{ui} | a_{ui} = 1 &\sim \mathcal{N}(\theta_u^\top \beta_i, \lambda_y^{-1}) \\
 y_{ui} | a_{ui} = 0 &\sim \delta_0,
 \end{aligned} \tag{1}$$

where  $\delta_0$  denotes that  $p(y_{ui} = 0 | a_{ui} = 0) = 1$ , and a set of hyperparameters denoting the inverse variance ( $\lambda_\theta, \lambda_\beta, \lambda_y$ ) is introduced.  $\mu_{ui}$  is the prior probability of exposure, which we will set to be proportional to item popularity (or word occurrence in our case).

From a generative modeling perspective, whether a user is exposed to an item is modeled as Bernoulli. Conditional on being exposed, users preferences comes from a matrix factorization model. Similarly to MF and WMF, the conditional distribution is factorized to  $K$  user preferences  $\theta_{i,1:K}$  and  $K$  item attributes  $\beta_{u,1:K}$ . A graphical representation of the model in Equation (1) is given in 1b.

ExpoMF applied to document modeling can be understood easily with the following example. Let "I love dogs" be a document in the dataset. Because "dogs" can often be substituted with "puppies", one could hope that ExpoMF would decrease the documents exposure to the word "puppies", thus reducing the need for the MF to create document and word embeddings reflecting that "puppies" is **not** in the document. A qualitative analysis of this phenomenon is provided in Section 5.4.

## 3 NLP Techniques

Unsupervised document models or topic models refer to a type of statistical models for discovering latent representations or *abstract* topics in a collection of documents. They can be used to cluster (unlabeled) documents. Latent Dirichlet Allocation (LDA) Blei et al. [2003] is the original and

standard topic model. Under this model, documents are encoded into semantic vectors where each dimension can be thought of as a topic.

Undirected topic models using restricted Boltzmann machines (RBMs) have also been proposed as well as a neural topic-model based on autoregressive modelling [Larochelle and Lauly, 2012] called DocNADE. A topic model based on Sigmoid Belief Networks and Deep AutoRegressive Neural Networks structures can be found in Mnih and Gregor [2014] where an MLP is employed to build a Monte-Carlo control variate estimator for stochastic estimation.

Since, recurrent neural networks (RNNs) have had tremendous success at modelling sequential data, e.g. speech, language modeling, machine translation, etc. However, they remain difficult to train on long sequences like documents in great part because of GPU memory constraints. Moreover, when RNNs model (unlabelled) documents, it is unclear where (or even if) the document representations live. For this reason, BoW representations is still standard for modelling whole documents.

In most unsupervised learning cases, state-of-the-art methods use the variational autoencoder framework [Kingma and Welling, 2013, Rezende et al., 2014] (VAEs). VAEs are often used to train deep generative models and are described in Section 3.1. Applying VAE on BoW representations [Miao et al., 2016] outperforms all aforementioned models thus achieving the state-of-the-art for document modeling. For this reason, we compare against this methodology in our empirical study (Section 5). The specific approach we compare to is Neural Variational Document Model (NVDM) [Miao et al., 2016] and is described in Section 3.2.

### 3.1 Variational Autoencoder

The VAE [Kingma and Welling, 2013, Rezende et al., 2014] is a regularized version of the traditional autoencoder (AE). It consists of two parts : an inference network  $q_\phi(z|x)$  that maps an input  $x$  to a posterior distribution of latent codes  $z$ , and a generative network  $p_\theta(x|z)$  that aims to reconstruct the original input conditioned on the latent encoding. In practice, both models are represented as neural networks.

By imposing a prior distribution  $p(z)$  on latent codes, this model enforces the distribution over  $z$  to be regular and well-behaved. This property enables proper sampling from the model via ancestral sampling from latent to input space. Without it,  $q_\phi(z|x)$  could encode the inputs as single points without the need to generalize, i.e., learning a look-up table<sup>1</sup>.

The full objective of the VAE is then:

$$\mathcal{L}(\theta; x) = \mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x|z)] - KL(q_\phi(z|x)||p(z)) \leq \log p(x), \quad (2)$$

which is often call evidence lower bound (ELBO). The ELBO is a valid lower bound of the log-likelihood, thereby making VAE proper generative models. For a more in depth analysis of VAE we refer the reader to Doersch [2016].

An optimization problem plagues VAE: the KL term can over-regularize the latent code, i.e. the posterior  $q_\phi(z|x)$  collapse towards the prior  $p(z)$ , thus not encoding anything about  $x$ . This issue is often referred to as *latent code collapse*. To relax some pressure on the latent code,  $\beta$ -VAE [Higgins et al., 2016] slightly modifies the VAE objective:

$$\mathcal{L}_\beta(\theta; x) = \mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x|z)] - \beta \cdot KL(q_\phi(z|x)||p(z)) \leq \log p(x),$$

The  $\beta$  hyperparameter now controls the latent code regularization. Reducing it will allow the model to encode more information in the posterior, although the representations might not be as parsimonious as the VAE's. Setting it to 0 collapse the model into a standard AE.

A particular advantage of VAE over CF methods leveraging Expectation-Maximisation (EM) is that the encoder  $q_\phi(z|x)$  can be used to encode new data, whereas EM requires retraining. This process is referred to as *amortized* inference. Arguably, amortized inference is less sample-efficient than EM because  $q_\phi(z|x)$  needs to learn how to generalize to new observations. However, we will go over some of its advantages in the next Section 4.

### 3.2 Neural Variational Document Model

The NVDM is a simple instance of unsupervised learning where a continuous hidden variable  $z \in \mathbb{R}^K$ , which generates all the words in a document independently, is introduced to represent its

<sup>1</sup>In this case the model collapses to a regular AE.

semantic content. Let  $\mathbf{X} \in \mathbb{R}^{|V|}$  be the BoW representation of a document with vocabulary  $V$ . Then, straightforwardly Equation 2 can be applied to documents and thus the likelihood can be maximized. Finally,  $\mathbb{E}[q_\phi(z|x)]$  can be used as a transferable representations of documents.

Finally, the model assumes that the observations were drawn from a multinomial distribution. In this work, we treated the output distribution as an hyperparameter sampled from {Bernoulli, Gaussian, multinomial}.

## 4 Contrasting Both Set of Techniques

In this section, we contrast collaborative filtering techniques with the NVDM. The methodologies can be applied to both user-item interactions or document-words occurrence. Thus, for the remaining of this section we use the terms user and item (instead of document and word).

First, an important difference comes from the inference method or dataset split. In CF, the training set is composed of all users and the held-out set contains unseen (often randomly sampled from the observed set) user-item observations. The models learn to predict a user-item interaction given both latent codes (users and items) learned at training time.

In contrast, in the NVDM case the training dataset is composed of randomly sampled users and the remaining users are assigned to the held-out set. At test time, NVDM encodes new users with its inference network  $q_\phi(z|x)$ . The posterior can then be used to compute the log-likelihood of said users with annealed importance sampling. Moreover, the latent codes, e.g.  $\mathbb{E}[q_\phi(z|x)]$ , can be used as a document representation in downstream tasks, in our case document classification.

Considering the performance of both approaches on held-out likelihood, CF methods have the advantage of modeling all users (or documents) at training time whereas NVDM has to generalize to new users. However, NVDM has access to more data per user. Thus, it is unclear if a methodology has a clear advantage over the other. From a practitioner’s perspective, in some cases it might be more important to find the most accurate representations for the users, i.e. learn all representations at training time with standard CF. Although, the NVDM framework might become handy once new users’ representations are needed because the encoder  $q_\phi(z|x)$  can be used in lieu of retraining the model.

Another considerable difference between both approaches is the choice of distribution for the observations. In standard CF, observations are modeled with Bernoulli or Gaussian distributions whereas standard language modeling (including NVDM) uses Multinomials. The dynamics are quite different. E.g. in CF, the model can increase the probability of seeing a particular item without decreasing the probability of observing the others. In the multinomial setting, this is not possible: because of the normalization over all items, if the model wants to increase the probability mass of an item it needs to remove it from elsewhere. Note, Liang et al. [2018] is a recent exception which models CF data using a multinomial.

Unfortunately, we can’t compare the methodologies on held-out likelihood because the models’ outputs live in different probability spaces. However, we can compare them by means of performance on a downstream task, in our case document classification (see Section 5.3). Precisely, the accuracy obtained by a document classifier trained on latent representations will serve as a proxy for the quality of the representations.

## 5 Empirical Study

In this section, we compare recommender systems models, i.e. MF, WMF and ExpoMF, with NLP models, i.e. NVDM, on the downstream task of document classification. The downstream task will shine light on the transferability of the learned representations, which is of the utmost importance in unsupervised learning. We find that the the MF techniques that downweights the (un)observations dramatically outperform the shallow NDVM and marginally outperforms the deeper version. Next, we provide a qualitative analysis of the latent exposures  $a_{ui}$  inferred by ExpoMF. MORE.

	Recall@20	Recall@50	NDCG@100	MAP@100
MF	0.33	0.45	0.33	0.19
WMF	<b>0.36</b>	<b>0.48</b>	<b>0.47</b>	<b>0.23</b>
ExpoMF	0.34	0.46	0.44	0.21

Table 1: Comparison between MF, WMF [Hu et al., 2008] and Expo-MF [Liang et al., 2016] in the low data regime.

## 5.1 Dataset

To test the quality of the learned document representation we choose a document dataset with labels, the classic news corpora *20NewsGroup*.<sup>2</sup> Every document in the dataset falls into one of 20 categories. We created two datasets in order to test the models in a low- and big-data regime. The vocabulary sizes are 1.5k and 10k respectively. The number of documents are 9k and 17k respectively.

## 5.2 Experimental Procedure

For both set of techniques, we proceeded in the same manner. First, we learn document and word embeddings in an unsupervised fashion with the methods described in Section 2 and 3. Early-stopping is performed on the validation set. Next, a linear classifier learns to map the latent representations of each document to their respective class. Again, early-stopping is performed on a new validation set. Finally, we pick the best performing models on the validation set and report their test error.

In order to fairly compare both methodologies we benchmark the CF models against 1-layer NVDM, because MF models are linear. For contextualization we also report the results for deeper VAE and for a supervised multi-layer perceptron (MLP).

Finally, because of the latent code collapse that hinders VAE training, we also report the results of  $\beta$ -VAE, i.e.  $0 < \beta < 1$ , and standard AE, ie.  $\beta = 0$ .

## 5.3 Quantitative Analysis

Before reporting the results on the downstream task analysis, we first present the standard CF metrics which in our case can be interpreted as document completion performance. Results are shown in Table 1 Over all metrics, we find a gain to underweighting the (un)observations, as highlighted by the WMF and ExpoMF outperforming MF, their more primitive version. This suggests that letting the models *focus* on observed words at the expense of the omitted ones facilitates the learning of good latent representations.

Next, we present the downstream task results in Table 2. We found the shallow VAE to perform poorly on the transfer task, their best performance being 10% which is not far from random (5%). We hypothesize that latent code collapse is to blame in this case, i.e. a linear model has low capacity thus it’s easy to over-regularize it with the KL term in the objective. The shallow AE (where the pressure is removed) achieved 20%. In both data regimes, the deeper versions of the AE and VAE performed considerably better, which is to be expected. Remarkably, WMF managed to outperform the all deepers models and ExpoMF tied the best performing deep model in the big data regimes. The increase in performance from MF to WMF highlight the fact that downweighting (un)observations is an extremely powerful tool for document modeling. Future work explores deeper versions of MF, WMF and ExpoMF.

## 5.4 Qualitative Analysis of Exposure

In Figure 2, we compare ExpoMF’s inferred posteriors of the exposure matrix for a document against the prior probability for exposure. The document class is ‘computer.windows’. Interestingly, ExpoMF learns to downweight the word ‘computer’ which was omitted from the document. This is evidence that ExpoMF can sensibly be applied to documents.

<sup>2</sup><http://qwone.com/~jason/20NewsGroups>

	Low Data	Big Data
1-layer AE	0.20	0.05
1-layer $\beta$ -VAE	0.10	0.07
1-layer VAE	0.08	0.08
MF	0.25	0.27
WMF	<b>0.46</b>	<b>0.64</b>
ExpoMF	0.36	0.49
AE	0.42	0.12
$\beta$ -VAE	0.37	0.49
VAE	0.30	0.40
Supervised MLP	0.60	0.75

Table 2: Test accuracy of the document classifier trained on the latent representations of each models. The MF models managed to outperform both shallow and deeper versions of the NVDM in both data regimes. Remarkably, the classifier trained on WMF representations’ performance is not so far from the supervised MLP model.

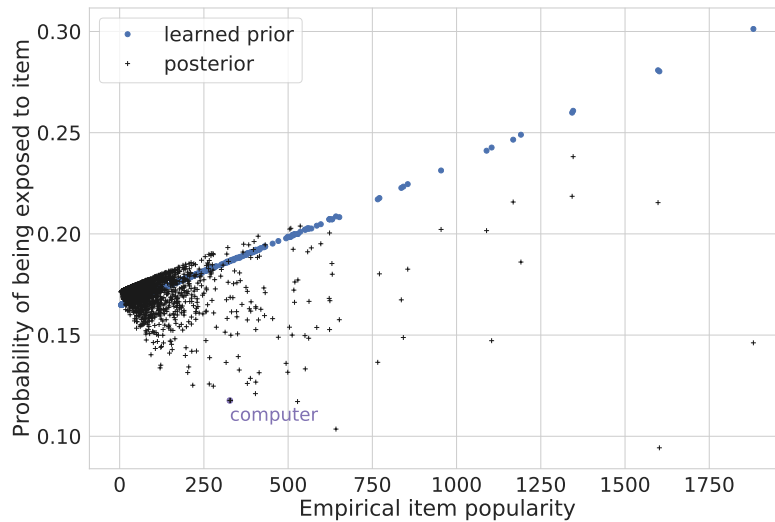


Figure 2: We compare the inferred posteriors of the exposure matrix for a document (denoted by black dots) against the prior probability for exposure (blue dots). The document class is 'computer.windows'. The word 'computer' was omitted from the document and the model efficiently learned to downweight this (un)observation.

## 6 Discussion

In this work, we apply older and more recent CF methods to the document modeling field. Interestingly, we find those methods result in more transferable representation of documents over the popular Neural Variational Document Model [Miao et al., 2016], as highlighted by the important gains on document classification transfer task. Future works explores deeper version of the matrix factorization models.

## References

- D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent Dirichlet allocation. *the Journal of Machine Learning Research*, 3:993–1022, 2003.
- C. Doersch. Tutorial on variational autoencoders. *arXiv preprint arXiv:1606.05908*, 2016.
- I. Higgins, L. Matthey, A. Pal, C. Burgess, X. Glorot, M. Botvinick, S. Mohamed, and A. Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. 2016.
- Y. Hu, Y. Koren, and C. Volinsky. Collaborative filtering for implicit feedback datasets. In *Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on*, pages 263–272. IEEE, 2008.
- D. P. Kingma and M. Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.
- Y. Koren, R. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, Aug. 2009. ISSN 0018-9162.
- H. Larochelle and S. Lauly. A neural autoregressive topic model. In *Advances in Neural Information Processing Systems*, pages 2708–2716, 2012.
- D. Liang, L. Charlin, J. McInerney, and D. M. Blei. Modeling user exposure in recommendation. In *Proceedings of the 25th International Conference on World Wide Web*, pages 951–961. International World Wide Web Conferences Steering Committee, 2016.
- D. Liang, R. G. Krishnan, M. D. Hoffman, and T. Jebara. Variational autoencoders for collaborative filtering. In P.-A. Champin, F. L. Gandon, M. Lalmas, and P. G. Ipeirotis, editors, *WWW*, pages 689–698. ACM, 2018. URL <http://dblp.uni-trier.de/db/conf/www/www2018.html#LiangKHJ18>.
- B. M. Marlin. Modeling user rating profiles for collaborative filtering. In S. Thrun, L. Saul, and B. Schölkopf, editors, *Advances in Neural Information Processing Systems 16*, pages 627–634. MIT Press, 2004. URL <http://papers.nips.cc/paper/2377-modeling-user-rating-profiles-for-collaborative-filtering.pdf>.
- Y. Miao, L. Yu, and P. Blunsom. Neural variational inference for text processing. In *International Conference on Machine Learning*, pages 1727–1736, 2016.
- A. Mnih and K. Gregor. Neural variational inference and learning in belief networks. *arXiv preprint arXiv:1402.0030*, 2014.
- A. Mnih and R. Salakhutdinov. Probabilistic matrix factorization. In *Advances in Neural Information Processing Systems*, pages 1257–1264, 2007.
- D. J. Rezende, S. Mohamed, and D. Wierstra. Stochastic backpropagation and approximate inference in deep generative models. *arXiv preprint arXiv:1401.4082*, 2014.