Fast Variational Bayesian Inference for Non-Conjugate Matrix Factorization Models

Matthias Seeger
Probabilistic Machine Learning Laboratory
Ecole Polytechnique Fédérale de Lausanne
INR 112, Station 14, CH-1015 Lausanne
matthias.seeger@epfl.ch

Guillaume Bouchard
Xerox Research Centre Europe
6, Chemin de Maupertuis, 38240 Meylan, France
guillaume.bouchard@xerox.com

February 15, 2012

Abstract

Probabilistic matrix factorization methods aim to extract meaningful correlation structure from an incomplete data matrix by postulating low rank constraints. Recently, variational Bayesian (VB) inference techniques have successfully been applied to such large scale bilinear models. However, current algorithms are of the alternate updating or stochastic gradient descent type, slow to converge and prone to getting stuck in shallow local minima. While for MAP or maximum margin estimation, singular value shrinkage algorithms have been proposed which can far outperform alternate updating, this methodological avenue remains unexplored for Bayesian techniques. In this paper, we show how to combine a recent singular value shrinkage characterization of fully observed spherical Gaussian VB matrix factorization with local variational bounding in order to obtain efficient VB inference for general MF models with non-conjugate likelihood potentials. In particular, we show how to handle Poisson and Bernoulli potentials, far more suited for most MF applications than Gaussian likelihoods. Our algorithm can be run even for very large models and is easily implemented in *Matlab*. It exhibits better prediction performance than MAP estimation on several real-world datasets.

1 Introduction

Matrix factorization (MF) models are widely used in machine learning and applications, for reduced rank regression [13], sparse principal components analysis [12], partial least squares [14], multi-task learning [3], latent semantic indexing [4], or collaborative filtering [7]. Machine learning approaches include maximum margin MF [17], maximum a posteriori (MAP) estimation [17, 15] and, more recently, variational Bayesian inference [11, 8, 12, 9].

Whether maximum margin, MAP or variational Bayes, methods for learning MF models from data roughly fall into two different classes. First, alternate minimization or stochastic gradient descent methods are based on efficient and simple updates, which directly exploit both the sparsity in the likelihood and the bilinear model structure. For example, alternate minimization solves for one matrix at a time, keeping the other fixed, then iterates this process in a round-robin fashion. In VB, the posterior over matrices is assumed to factorize, which at least in conjugate likelihood cases leads to simple updates of posterior factors. These are iterated over in the same fashion. While easily implemented and scaled up to large problems, these techniques tend to converge exceedingly slowly and are prone to getting stuck in poor local minima, necessitating multiple restarts and additional heuristics. The second class of algorithms is based on recent fundamental results which establish *analytical solutions* for certain basic MF models [19, 17, 2, 9], essentially by computing a singular value decomposition (SVD) and shrinking or thresholding the singular values. Can these exact characterizations be used in order to learn *general* MF models, as subroutines of iterative algorithms, much like Newton-Raphson optimization

Appearing in Proceedings of the 15^{th} International Conference on Artificial Intelligence and Statistics (AISTATS) 2012, La Palma, Canary Islands. Volume XX of JMLR: W&CP XX. Copyright 2012 by the authors.

is based on quadratic minimization? If so, we should end up with algorithms which are much faster to converge and behave more robustly than methods based on simple gradient descent or alternate updating. Moreover, due to its vast impact on diverse applications, very efficient, highly parallelizable code for approximate SVD is available to drive MF learning algorithms. This idea has been realized both for maximum margin and MAP estimation [17, 20, 18], while all previous variational Bayesian algorithms for general models belong to the first class [11, 8].

Our main contribution in this paper is a novel SVD-based algorithm for variational Bayesian inference in matrix factorization models with general likelihoods. The main primitive driving our method is a recent SVD shrinkage characterization of VB matrix factorization for complete data spherical Gaussian likelihood [9]. We show how to solve VB inference problems for realistic MF models with highly incomplete Poisson or Bernoulli likelihoods at the expense of a few calls to approximate SVD code. Our method can be formulated entirely in terms of sparse or low rank matrices, and it is easily scaled up to very large problems. Its computational cost per iteration is equivalent to SVD-based algorithms for MF MAP estimation [20, 18]. The structure of this paper is as follows. We discuss related work in Section 2. In Section 3, we introduce the variational Bayesian matrix factorization setup and review an analytically solvable special case [9]. Our algorithm is derived in Section 4. We comment on non-conjugate likelihood potentials in Section 4.1 and describe our large-scale implementation in Section 4.2. We present experimental results on a range of real-world datasets in Section 5, and close with conclusions in Section 6.

2 Related Work

Matrix factorization (MF) models are core components of collaborative filtering or recommender systems and have attracted a large amount of research to date. We restrict our focus to probabilistic approaches. Maximum a posteriori (MAP) estimation for MF models or closely related "maximum margin" variants was pioneered in [17], where the link to SVD with shrinkage on the spectrum was observed. Previously, an analytical solution based on eigendecomposition was established for probabilistic principal components analysis in [19]. The MAP interpretation of the method of [17], as well as the link to nuclear-norm regularization, was made explicit in [15].

More recently, variational Bayesian inference methods have been proposed [11, 8], where factors are integrated out approximately rather than estimated. While easy to implement and scalable to large problems, previous VBMF algorithms are alternate updating or stochastic gradient descent variants, which converge slowly and are prone to getting stuck in poor local optima. A key result for VBMF is the global analytical solution for the Gaussian complete likelihood case established in [9], of pivotal importance for our work here.

3 Variational Bayesian Matrix Factorization

Suppose our goal is to predict missing entries in a $m \times n$ matrix $\boldsymbol{Y}$ from observations y_{ij}, $(ij) \in \mathcal{O}$. For example, the m rows may index users of an e-commerce website, the n columns items on sale, and the goal is to make recommendations. If $|\mathcal{O}| \ll mn$, it is essential to share statistical strength by learning latent correlations between the groups, which in bilinear *matrix factorization models* are represented via a joint r-dimensional space, $r \ll \min\{n, m\}$. We specify latent factors $\boldsymbol{U} \in \mathbb{R}^{m\times r}$, $\boldsymbol{V} \in \mathbb{R}^{n\times r}$, and set $\boldsymbol{X} = \boldsymbol{U}\boldsymbol{V}^T \in \mathbb{R}^{m\times n}$. Here, $\boldsymbol{U} = [\boldsymbol{u}_k]$, $\boldsymbol{V} = [\boldsymbol{v}_k]$, $k = 1, \dots, r$, where $\boldsymbol{u}_k \in \mathbb{R}^m$, $\boldsymbol{v}_k \in \mathbb{R}^n$ are the columns which span the latent low-dimensional representation.

For the likelihood, we assume that $P(\boldsymbol{Y}|\boldsymbol{U}, \boldsymbol{V}) = P(\boldsymbol{Y}|\boldsymbol{X})$, where $P(\boldsymbol{Y}|\boldsymbol{X})$ factorizes w.r.t. matrix entries:

$$P(\boldsymbol{Y}|\boldsymbol{X}) = \prod_{(ij)\in\mathcal{O}} P(y_{ij}|x_{ij}), \quad x_{ij} = \sum_{k=1}^{r} u_{ik}v_{jk}.$$

In this paper, we are interested in general likelihood potentials $P(y_{ij}|x_{ij})$ which do not admit conjugacy properties. For the prior, we assume the following factorized form:

$$P(\boldsymbol{U}, \boldsymbol{V}) = \prod_{k=1}^{r} P(\boldsymbol{u}_k)P(\boldsymbol{v}_k).$$

Bayesian inference is generally intractable in matrix factorization models, even in the case of conjugate Gaussian likelihood potentials. The problem arises from product couplings $u_{ik}v_{jk}$ in the likelihood, which can give rise to a complex posterior distribution $P(\boldsymbol{U}, \boldsymbol{V}|\boldsymbol{Y})$. In the *variational Bayesian* (VB) approach, we approximate the posterior with a distribution $Q(\boldsymbol{U}, \boldsymbol{V})$ from a tractable class, by minimizing

$$\begin{aligned}\mathcal{F} &= \mathrm{E}_{Q(\boldsymbol{U},\boldsymbol{V})}\left[\log \frac{Q(\boldsymbol{U}, \boldsymbol{V})}{P(\boldsymbol{Y}|\boldsymbol{U}, \boldsymbol{V})P(\boldsymbol{U}, \boldsymbol{V})}\right] \\ &= \mathrm{D}[Q(\boldsymbol{U}, \boldsymbol{V}) \,\|\, P(\boldsymbol{U}, \boldsymbol{V}|\boldsymbol{Y})] - \log P(\boldsymbol{Y}).\end{aligned} \tag{1}$$

We focus on variational distributions of the factorized form

$$Q(\boldsymbol{U}, \boldsymbol{V}) = \prod_{k=1}^{r} Q(\boldsymbol{u}_k)Q(\boldsymbol{v}_k). \tag{2}$$

In other words, $Q(\boldsymbol{U}, \boldsymbol{V})$ admits the same factorization between pairs of columns of $[\boldsymbol{U}\,\boldsymbol{V}]$ as the prior.

Even with these simplifying assumptions, the minimization of (1) is not a simple problem. The most frequently chosen approach is to update $Q(\boldsymbol{U})$ and $Q(\boldsymbol{V})$ in turn [8].

However, such alternate updating algorithm are notoriously prone to getting stuck in poor local minima. Our main contribution is an algorithm which fares much better in general, by exploiting recent results about a special case of (1) which can be solved analytically.

3.1 Fully Gaussian Complete Likelihood: G-VBMF

Maybe the simplest instance of VBMF is the spherical Gaussian complete likelihood case, called G-VBMF in the sequel. Here, all entries of $\boldsymbol{Y} \in \mathbb{R}^{m\times n}$ are observed ($\mathcal{O} = \{1,\dots,m\} \times \{1,\dots,n\}$), and $P(y_{ij}|x_{ij}) = N(y_{ij}|x_{ij},\sigma^2)$ is Gaussian. The noise variance σ^2 has to be the same for all likelihood potentials. Moreover, the priors are Gaussian as well:

$$P(\boldsymbol{U}) = \prod_k N(\boldsymbol{u}_k|\boldsymbol{0}, c_{u,k}^2\boldsymbol{I}),\; P(\boldsymbol{V}) = \prod_k N(\boldsymbol{v}_k|\boldsymbol{0}, c_{v,k}^2\boldsymbol{I}).$$

It is straightforward to apply alternating minimization to (1) in this case [8], yet the problem remains non-convex. Importantly, Nakajima *et.al.* [9] showed that its global minimum points, unique up to obvious orthonormal symmetries, can be solved for analytically. If λ_k, $\tilde{\boldsymbol{u}}_k$, $\tilde{\boldsymbol{v}}_k$ are the r largest singular values, left and right singular vectors of $\boldsymbol{Y}$, so that $\tilde{\boldsymbol{U}}\boldsymbol{\Lambda}\tilde{\boldsymbol{V}}^T$ is closest to $\boldsymbol{Y}$ in Frobenius[1] norm among all rank r matrices, then for a global minimum point Q^* of (1):

$$\mathrm{E}_{Q^*}\left[\boldsymbol{U}\boldsymbol{V}^T\right] = \mathrm{E}_{Q^*}[\boldsymbol{U}]\mathrm{E}_{Q^*}[\boldsymbol{V}]^T = \tilde{\boldsymbol{U}}(\operatorname{diag}\boldsymbol{\gamma})\tilde{\boldsymbol{V}}^T,$$

where the elements γ_k of $\boldsymbol{\gamma}$ are closed form functions depending on λ_k, σ^2, $c_{u,k}^2$, $c_{v,k}^2$, m and n. Essentially, $\mathrm{E}_{Q^*}[\boldsymbol{U}\boldsymbol{V}^T]$ is obtained from $\tilde{\boldsymbol{U}}\boldsymbol{\Lambda}\tilde{\boldsymbol{V}}^T$ by leaving the matrices in place, but shrinking $\lambda_k \to \gamma_k$ in a specific way [9]. This characterization allows us to circumvent error-prone alternating minimization entirely.

In practice, the G-VBMF setup is severely restricted. It requires a complete likelihood function with potentials on each entry of $\boldsymbol{Y}$, which have to be Gaussian and must share the same variance. These restrictions do not make much sense for our recommender system example, a prototypical application of matrix factorization models. The likelihood is incomplete by default. Moreover, observed entries of $\boldsymbol{Y}$ are binary or natural numbers, which are poorly represented by a spherical Gaussian likelihood. In the next section, we show how to combine the G-VMBF result with variational bounding techniques in order to overcome these restrictions.

4 An Algorithm for General Potentials

Our aim is to solve the VBMF problem (1) in the general case described in Section 3, yet to make use of the analytical G-VBMF solution of Section 3.1 to drive our algorithm. Technically speaking, we have to approximate the likelihood $P(\boldsymbol{Y}|\boldsymbol{X})$ by a Gaussian with spherical covariance (all variances equal). While this is certainly a bad one-off approximation, we will introduce variational parameters which determine the (pseudo-)observations $\boldsymbol{Y}$ in G-VBMF, then iteratively improve the fit to the posterior distribution. For simplicity, we assume that the prior $P(\boldsymbol{U},\boldsymbol{V})$ already has the form required by G-VBMF, so we only have to deal with the likelihood. An extension to more general Gaussian or non-Gaussian priors can be obtained in much the same way.

We can write $\min_Q \mathcal{F}$ from (1) as

$$\min_{Q(\boldsymbol{U},\boldsymbol{V})} \mathrm{E}_Q\left[-\log P(\boldsymbol{Y}|\boldsymbol{U}\boldsymbol{V}^T)\right] + \mathrm{D}[Q\,\|\,P],$$

where $\mathrm{D}[Q\,\|\,P] = \mathrm{D}[Q(\boldsymbol{U},\boldsymbol{V})\,\|\,P(\boldsymbol{U},\boldsymbol{V})]$. Introducing $\boldsymbol{X} = \boldsymbol{U}\boldsymbol{V}^T = [x_{ij}]$, we can write

$$-\log P(\boldsymbol{Y}|\boldsymbol{U}\boldsymbol{V}^T) = \sum_{i,j} f_{ij}(x_{ij}), \quad x_{ij} = \sum_{k=1}^{r} u_{ik}v_{jk},$$

where $f_{ij}(x_{ij}) = -\log P(y_{ij}|x_{ij})$ for $(ij) \in \mathcal{O}$, $f_{ij}(x_{ij}) = 0$ otherwise. We require that the $f_{ij}(x_{ij})$ are twice differentiable and

$$f_{ij}''(x_{ij}) \le \kappa \quad \forall x_{ij}\; \forall i,j.$$

We demonstrate below how to choose κ for Bernoulli and Poisson likelihoods. Now, by Taylor's theorem:

$$f_{ij}(x_{ij}) \le \underbrace{\frac{\kappa}{2}(x_{ij}-\xi_{ij})^2 + f'(\xi_{ij})(x_{ij}-\xi_{ij}) + f_{ij}(\xi_{ij})}_{=:q_{ij}(x_{ij};\xi_{ij})}.$$

Plugging these tight bounds into the criterion and interchanging $\mathrm{E}_Q[\dots]$ and $\min_{[\xi_{ij}]}$ (which weakens the bound), we obtain our final variational optimization problem:

$$\min_{Q(\boldsymbol{U},\boldsymbol{V}),[\xi_{ij}]} \sum_{ij} \mathrm{E}_Q\left[q_{ij}(x_{ij};\xi_{ij})\right] + \mathrm{D}[Q\,\|\,P].$$

Our algorithm alternates between updates of $[\xi_{ij}]$ and of $Q(\boldsymbol{U},\boldsymbol{V})$. For the former, the criterion decouples additively, so each ξ_{ij} can be updated independently. Since q_{ij} is a quadratic, we have that $\mathrm{E}_Q[q_{ij}(x_{ij};\xi_{ij})] = q_{ij}(\mathrm{E}_Q[x_{ij}];\xi_{ij}) + C_{ij}$, where C_{ij} does not depend on ξ_{ij}. Now,

$$q_{ij}\left(\mathrm{E}_Q[x_{ij}];\xi_{ij}\right) \ge f_{ij}\left(\mathrm{E}_Q[x_{ij}]\right) = q_{ij}\left(\mathrm{E}_Q[x_{ij}];\mathrm{E}_Q[x_{ij}]\right),$$

so that the update is $\xi_{ij} \leftarrow \mathrm{E}_Q[x_{ij}]$, or compactly

$$[\xi_{ij}] \leftarrow \mathrm{E}[\boldsymbol{U}]\mathrm{E}[\boldsymbol{V}]^T.$$

For the update of $Q(\boldsymbol{U})Q(\boldsymbol{V})$, note that

$$q_{ij}(x_{ij};\xi_{ij}) \doteq \frac{\kappa}{2}\left(x_{ij} - (\xi_{ij} - f'(\xi_{ij})/\kappa)\right)^2$$
$$\doteq -\log N(\tilde{y}_{ij}|x_{ij}, 1/\kappa), \quad \tilde{y}_{ij} := \xi_{ij} - f'(\xi_{ij})/\kappa.$$

[1] The Frobenius norm is defined as $\|\boldsymbol{A}\|_2 = (\mathrm{tr}\,\boldsymbol{A}^T\boldsymbol{A})^{1/2}$.

where "$\doteq$" denotes equality up to a constant independent of x_{ij}. Therefore, for fixed $[\xi_{ij}]$, the update of $Q(\boldsymbol{U})Q(\boldsymbol{V})$ is equivalent to G-VBMF with pseudo-data $\tilde{\boldsymbol{Y}} = [\tilde{y}_{ij}]$ and variance $\sigma^2 = 1/\kappa$. We can solve for $\mathrm{E}[\boldsymbol{U}\boldsymbol{V}^T]$ analytically, as noted in Section 3.1. Our algorithm iterates between updates of $[\xi_{ij}]$, of $\tilde{\boldsymbol{Y}}$, and calls of G-VBMF. It is summarized in Algorithm 1.

Algorithm 1 General VBMF Algorithm

```
$\mathrm{E}[\boldsymbol{U}] \leftarrow \mathbf{0}$, $\mathrm{E}[\boldsymbol{V}] \leftarrow \mathbf{0}$.
while not converged do
    Update $[\xi_{ij}] \leftarrow \mathrm{E}[\boldsymbol{U}]\mathrm{E}[\boldsymbol{V}]^T$.
    Update $\mathrm{E}[\boldsymbol{U}]\mathrm{E}[\boldsymbol{V}]^T$ by G-VBMF analytical solution,
    based on pseudo-data $\tilde{\boldsymbol{Y}} = [\xi_{ij} - f'(\xi_{ij})/\kappa]$.
end while
```

4.1 Bounds for Likelihood Potentials

In this section, we establish quadratic upper bounds on $-\log P(y|x)$ both for binary classification (Bernoulli) and Poisson likelihood potentials. In both cases, $x \mapsto -\log P(y|x)$ is a convex function (the potentials are log-concave). While the posterior distribution $P(\boldsymbol{U}, \boldsymbol{V}|\boldsymbol{Y})$ is complicated due to the product coupling $\boldsymbol{U}\boldsymbol{V}^T$ in the likelihood, log-concave potentials at least do not add additional complexities. Moreover, since $f''_{ij}(x_{ij}) \geq 0$, our quadratic bounds are tighter in this case. Essentially, $f_{ij}(x_{ij})$ is sandwiched between the upper bound $q_{ij}(x_{ij};\xi_{ij})$ and the linear lower bound $f'(\xi_{ij})(x_{ij} - \xi_{ij}) + f_{ij}(\xi_{ij})$.

If reponses are binary, $y \in \{0, 1\}$, a Bernoulli likelihood is appropriate:

$$P(y|x) = \frac{e^{yx}}{1+e^x}, \quad f(x) = \log\left(1 + e^x\right) - yx.$$

Clearly, $f(x)$ is convex. Moreover, $f'(x) = \pi(x) - y$, $\pi(x) = 1/(1+e^{-x})$ and $f''(x) = \pi(x)\pi(-x) \leq 1/4$. For Bernoulli likelihood potentials, we can use a quadratic bound with $\kappa = 1/4$. Moreover, we update

$$\tilde{y}_{ij} = \xi_{ij} - f'_{ij}(\xi_{ij})/\kappa = \xi_{ij} - 4(\pi(\xi_{ij}) - y_{ij}).$$

Note that this bound tends to be looser than Jaakkola's bound [5] (see [6] for numerical experiments comparing the two). The latter allows for the curvature term κ to depend on ξ_{ij} as well. While Jaakkola's bound still leads to Gaussian updates for $Q(\boldsymbol{U})$ and $Q(\boldsymbol{V})$, its heteroscedasticity would not allow for the reduction to G-VBMF, since the latter calls for a homoscedastic likelihood.

If $y \in \mathbb{N} = \{0, 1, \dots\}$, we can use a Poisson likelihood with rate function $\lambda(x) > 0$:

$$P(y|x) \propto \lambda(x)^y e^{-\lambda(x)}, \quad f(x) = \lambda(x) - y\log\lambda(x). \quad (3)$$

It has been shown in [10] that $P(y|x)$ is log-concave ($x \mapsto -\log P(y|x)$ is convex) if $\lambda(x)$ is both convex and log-concave. A simple choice satisfying this property is $\lambda(x) = e^x$. The problem with this rate function is its exponential growth for large x, which implies non-robust behaviour in the presence of outliers. Moreover, e^x does not have bounded curvature, so our reduction to G-VBMF would not work.

In order to remedy these problems, we propose a novel link function not previously used in this context, $\lambda(x) = \log(1+e^x)$. It is well known that $\lambda(x)$ is convex, and we prove log-concavity at the end of this section. It shares with e^x the exponential decay as $x \to -\infty$, yet grows only linearly (with slope approaching 1) for large x (see Figure 1). Finally, we need to bound $f''(x)$. First, $\lambda''(x) \leq 1/4$, as seen above. Second, it is confirmed by inspection that the second derivative of $-\log\log(1+e^x)$ is upper bounded by 0.17, so that

$$f''_{ij}(x_{ij}) \leq \kappa = 1/4 + 0.17 y_{\max}, \quad y_{\max} = \max_{ij} y_{ij}.$$

Not surprisingly, our bound degrades with the presence of entries with large y_{ij}. In our experiments, we follow common practice and clip overly large counts. We update

$$\tilde{y}_{ij} = \xi_{ij} - \frac{f'_{ij}(\xi_{ij})}{\kappa} = \xi_{ij} - \frac{\pi(\xi_{ij})(1 - y_{ij}/\lambda(\xi_{ij}))}{\kappa}.$$

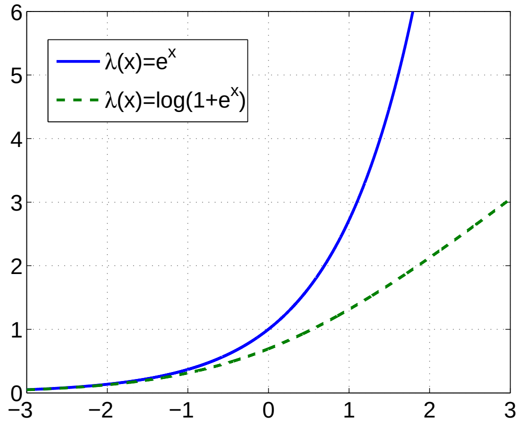
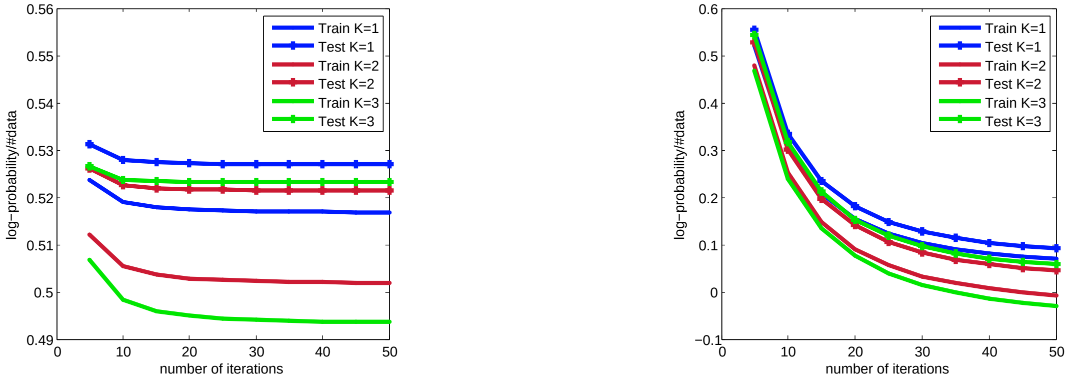


Figure 1: Rate functions $\lambda(x)$ for Poisson potentials used in this paper.

Finally, we establish the log-concavity of $\lambda(x) = \log(1+e^x)$. First, $\lambda'(x) = \pi(x) = (1+e^{-x})^{-1}$, $\pi'(x) = \pi(x)\pi(-x) = \pi(x)^2 e^{-x}$. If $g(x) = \log\lambda(x)$, then $g'(x) = \pi(x)/\lambda(x)$,

$$g''(x) = \frac{1}{\lambda(x)}\left(\pi'(x) - \frac{\pi(x)^2}{\lambda(x)}\right) = \frac{\pi(x)^2}{\lambda(x)}\left(e^{-x} - \frac{1}{\lambda(x)}\right).$$

Now, $\log(1+x) \leq x$, so that $-1/\lambda(x) \leq -e^{-x}$, therefore $g''(x) \leq 0$, so that $g(x)$ is concave.



Binary potentials

Poisson potentials

Figure 2: Learning curves for binary and Poisson potential on the Bus Stops dataset for several values of the latent feature dimension. The vertical axis is the normalised log-likelihood computed on train/test data.

4.2 Large-Scale Implementation

Most real-world applications of MF models feature very large m and n along with sparse observations: $|\mathcal{O}| \ll mn$. Both m and n can be in the tens or hundreds of thousands, and $|\mathcal{O}|$ can be many millions. Viewed in this light, the analytical solution of G-VBMF in terms of an SVD of a $m \times n$ matrix (Section 3.1) seems much less attractive. Storing or even building a dense matrix of this size is far beyond tractability, let alone performing an SVD at the cost of $O(\max\{m,n\} \min\{m,n\}^2)$. For reasons such as these, SVD-based algorithms are not used much in machine learning, where simpler alternate updating or stochastic gradient descent algorithms are preferred. In this section, we show that our VBMF algorithm, while based on SVD, can nevertheless be run at very large scales, using code which is publicly available and in fact part of *Matlab*. Our observations, which are related to what is proposed in [20], should be useful beyond the VBMF algorithm of interest here.

Recall the VBMF updates from Algorithm 1. We will show that beyond $\mathrm{E}[\boldsymbol{U}] \in \mathbb{R}^{m\times r}$ and $\mathrm{E}[\boldsymbol{V}] \in \mathbb{R}^{n\times r}$, only sparse matrices with $|\mathcal{O}|$ non-zeros have to be maintained. First, $[\xi_{ij}] = \mathrm{E}[\boldsymbol{U}]\mathrm{E}[\boldsymbol{V}]^T$. Second, $\tilde{\boldsymbol{Y}} = [\xi_{ij} - f'_{ij}(\xi_{ij})/\kappa]$. However, $f'_{ij}(\xi_{ij}) = 0$ for $(ij) \notin \mathcal{O}$, so that $\tilde{\boldsymbol{Y}}$ is $\mathrm{E}[\boldsymbol{U}]\mathrm{E}[\boldsymbol{V}]^T$ plus a $|\mathcal{O}|$-sparse matrix. We can compute matrix-vector multiplications with $\tilde{\boldsymbol{Y}}$ at the cost $O(|\mathcal{O}| + r(m+n))$, by multiplying with $\mathrm{E}[\boldsymbol{U}]$, $\mathrm{E}[\boldsymbol{V}]^T$ and the sparse matrix separately. This means that the analytical solution of G-VBMF can be obtained by an approximate SVD package such as `ARPACK`, the code behind *Matlab's* `SVDS`. For not too large rank r, `SVDS` scales about as $O(r)$ matrix-vector multiplications. Once $\mathrm{E}[\boldsymbol{U}]$, $\mathrm{E}[\boldsymbol{V}]$ are updated, we compute $(\mathrm{E}[\boldsymbol{U}]\mathrm{E}[\boldsymbol{V}]^T)_{\mathcal{O}}$ in $O(|\mathcal{O}|r)$, which is the basis for computing the next $\tilde{y}_{ij}$, $(ij) \in \mathcal{O}$. To conclude, even though our algorithm is based on the analytical SVD solution of G-VBMF, it requires no more memory than an alternate minimization algorithm, and it comes at about the same cost per iteration (if we count iterations of `SVDS` in our case).

5 Experiments

We consider the following datasets:

- *Bus Stops*: In this public transport analysis scenario, commuters validate a ticket upon entering a bus. These validations are stored in a database, to be mined in order to identify potential problems or bottlenecks occuring at certain times during the day. We used a full day of validation tickets of a public transport system from a large city (about 2 million inhabitants), resulting in a matrix with 500 rows (we selected the 500 most used bus stops) and 300 columns (corresponding to 5 rush hours of the day), containing a total of 249 500 validations. Our objective is data imputation in order to compensate for missing information due to machine failures, deficiency of ticketing, frauds or other error sources.

- *Antenna*: This dataset consist of (anonymized) mobile phone connectivity data of foreign tourists traveling within a large city. A typical day of the week was split in five-minute intervals, and the number of connections to each antenna (or base station) in this time period was computed, indicating which locations are preferred by tourists like at what time of the day. However, spatial information was not used in the current experiment, where the data was organized in a 1229×244 (number of antennas $\times$ number of time intervals) matrix. At a total of 988715 connections, 79% of the matrix entries are zeros. The busiest interval corresponds to 49 connections on a single antenna.

- *Print Jobs*: We obtained five months of print logs data from an office environment, indicating which

		90% missing		80% missing		50% missing	
Dataset	Algo	Binary loss	time(sec)	Binary loss	time(sec)	Binary loss	time(sec)
busstops	MAP	34515.2(558.9)	17.9	20292.9(91.8)	10.6	4497.0(201.8)	12.3
busstops	VB	34404.4(489.4)	18.8	19781.9(388.9)	10.5	4500.8(225.4)	13.2
antenna	MAP	-101995.0(3807.3)	9.5	-85221.9(1184.2)	10.7	-81724.1(1069.1)	11.0
antenna	VB	-108155.7(4634.9)	14.8	-93535.7(1231.9)	11.2	-81724.2(1068.6)	10.4
printJobs	MAP	2895.1(274.5)	1.3	2473.6(464.3)	0.9	466.6(143.9)	0.9
printJobs	VB	2773.8(127.7)	1.6	2427.8(880.2)	1.8	460.8(166.9)	1.1

Table 1: Test log-likelihood of the MAP estimator and the proposed VB algorithm for binary data on various data sets with different proportion of missing data. Results are averaged over 10 runs (standard deviation in parenthesis).

user prints on which printer at what time. The analysis of these logs is useful for the print infrastructure manager to detect malfunctionning devices, e.g. by identifying where some printers tend to be under userd compared to their historical data. The data sample that we obtained contains 27249 unique print logs which where agregated into 11447 print events (multiple prints within one minutes were considered as a single print event). There are 124 users printing on 22 printers. Even if it is small, this dataset is relatively hard to model with a low-rank parameterization because each user tend to print only on one or two printers.

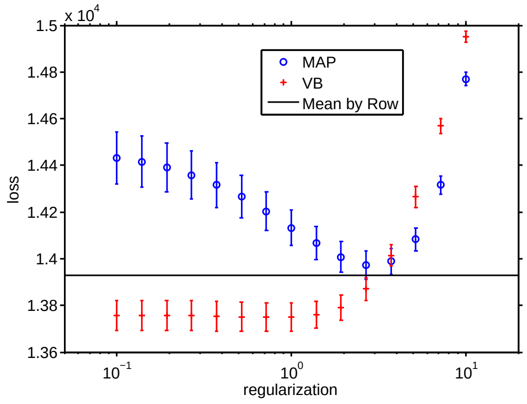


Figure 3: Predictive loss computed on the binarized BusStop dataset to illustrate the automatic smoothing of the VB the algorithm compared to MAP estimation. The horizontal axis is the amount of regularization, i.e. $c_{u,k}^{-2} = c_{v,k}^{-2}$.

For each dataset, we generated a binary version of the data by setting to one any strictly positive count. For the count data, we removed 5% of the highest values to obtain a reasonable value for $y_{\max}$. We used logistic regression (i.e. sigmoid inverse link function) to model binary observations and Poisson potentials (3) where based on the $\lambda(x) = \log(1 + e^x)$ inverse link function. Our goal is data imputation: predicting the values of missing entries in a count matrix. We evaluate our predictions using the likelihood of the test data, i.e. the logistic loss

$$\ell(y, \hat{x}) = y \log(\hat{x}) + (1 - y) \log(1 - \hat{x})$$

for binary data and the asymmetric Poisson loss:

$$\ell(y, \hat{x}) = \lambda(\hat{x}) - y \log\left(\frac{x}{\lambda(\hat{x})}\right),$$

for count data, where y is the true value and $\hat{x}$ is the prediction.

Implementation details For binary data, we used the logistic regression inverse link function, and for count data, we used the novel rate function $\lambda(x) = \log(1 + e^x)$ presented earlier. The prior variances $c_{u,k}^2$, $c_{v,k}^2$ of the latent factors where set to be equal and their value was selected in the range $[0.1, 10]$ and the dimension K of the latent factors varied from 1 to 5. For each dataset, we selected these hyperparameters by optimizing the performance on a validation data matrix, which was not used in the subsequent experiments. The training times quoted below do not include these validation runs. Our implementation is written in *Matlab*. Computations were run on a standard 8-core Linux server.

Baseline algorithm We compare against MAP estimation. As noted there, the MAP algorithm has a similar structure to our VB method, where the G-VBMF subroutine simply shrinks and threshold at 0 the eigenvalues, by a amount directly proportional to $c_{u,k}^{-2}$ and $c_{v,k}^{-2}$. Note that the singular value thresholding step in the MAP algorithm corresponds to the solution of a nuclear-norm regularized problem with spherical Gaussian likelihood.

In all our experiments, convergence speed is similar to EM-like algorithms: the train likelihood quickly increases with the first iterations, but the algorithm is extremely slow to converge to the exact maximum values. In any case, after approximately 100 iterations, the test likelihood does not increase significantly. In all the experiments, we stopped after a maximum of 200 iterations. One can see expample of learning curves for various values of the latent dimension

		90% missing		80% missing		50% missing	
Dataset	Algo	Poisson loss	time(sec)	Poisson loss	time(sec)	Poisson loss	time(sec)
busstops	MAP	72987.86(235.8)	25.9	63394.70(71.4)	21.8	38641.02(233.6)	17.8
busstops	VB	72873.65(247.1)	27.3	63392.47(78.4)	28.1	38639.73(232.5)	18.5
antenna	MAP	65694.61(23.7)	27.7	56708.71(67.3)	26.2	34155.29(73.2)	18.4
antenna	VB	65541.63(22.0)	34.8	56701.88(64.9)	31.2	34160.26(74.6)	19.1
printJobs	MAP	1793.74(2.0)	2.9	1391.99(18.5)	2.6	486.67(0.3)	2.6
printJobs	VB	1701.68(2.3)	3.1	1288.45(20.1)	2.7	486.85(0.1)	3.0

Table 2: Test log-likelihood of the MAP estimator and the proposed VB algorithm for count data on various data sets with different proportion of missing data. Results are averaged over 10 runs (standard deviation in parenthesis).

in Figure 2. Results on binary data are shown in Table 1, while results on count data based on Poisson potentials are shown in Table 2. VB predictions generally outperform those of MAP estimation, indicating that Bayesian averaging over uncertainties helps in our data imputation tasks, especially when the fraction of missing data is large. When the matrix is nearly full (i.e. with 50% of missing data), the MAP solution and the VB solution tend to become equivalent. Note also that the runtime of our VB method is close to that of MAP, showing that the additional computation involved in the G-VBMF subroutine is negligible.

6 Conclusions

We presented a novel efficient algorithm for variational Bayesian inference in general matrix factorization models with non-conjugate Poisson and Bernoulli likelihoods. Based on the analytical solution for fully observed spherical Gaussian likelihood Matrix Factorization models derived in [9], our method is driven by few calls to approximate SVD solvers for "sparse plus low rank" matrices, in contrast to previous VB alternate minimization algorithms which typically require many iterations and restarts to avoid poor local optima. We employ Poisson potentials for count or discrete score data, proposing a novel inverse link function with better properties than the canonical exponential choice. Our method can be scaled to very large problems using standard software available in *Matlab*. It outperforms MAP estimation on a range of real-world problems, while running in time comparable to a highly efficient MAP solver which employs SVD shrinkage in a similar manner.

In future work, we aim to use more sophisticated local variational bounds, while maintaining the reduction to G-VBMF. Specifically, we are not yet using the degrees of freedom offered by the $c_{u,k}$, $c_{v,k}$ parameters in G-VBMF. Also, we are working on a improvement on the bound by relaxing the constraint of having a constant variance κ accross all the observations, as done for heteroscedastic matrix factorization [1]: by using local variational bounds valid for every row rather than for the whole matrix, one can obtain tighter bound. A simple rescaling of the latent matrix rows $\boldsymbol{X} = \boldsymbol{U}\boldsymbol{V}^T$, would then lead to the same subproblem solved by the G-VBMF. This idea could equivalently be applied to the columns of the matrix, but it is not clear how to do it jointly in rows and columns, in order to be efficient on large and nearly squared matrices.

References

[1] Lakshimanarayan B., G. Bouchard, and C. Archambeau. Robust Bayesian matrix factorisation. In G. Gordon and D. Dunson, editors, *Workshop on Artificial Intelligence and Statistics 14*, 2011.

[2] J. Cai, E. Candes, and Z. Shen. A singular value thresholding algorithm for matrix completion. *SIAM J. Optim.*, 20:1956–1982, 2008.

[3] O. Chapelle and Z. Harchaoui. A machine-learning approach to conjoint analysis. In Saul et al. [16], pages 257–264.

[4] S. Deerwester, S. Dumais, G. Furnas, R. Landauer, and R. Harshman. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6):391–407, 1990.

[5] T. Jaakkola. *Variational Methods for Inference and Estimation in Graphical Models*. PhD thesis, Massachusetts Institute of Technology, 1997.

[6] E. Khan, B. Marlin, G. Bouchard, and K. Murphy. Variational bounds for mixed-data factor analysis. In Y. Bengio, D. Schuurmans, J. Lafferty, C. K. I. Williams, and A. Culotta, editors, *Advances in Neural Information Processing Systems 22*. Curran Associates, 2009.

[7] J. Konstan, B. Miller, D. Maltz, J. Herlocker, L. Gordon, and J. Riedl. Grouplens: Applying collaborative filtering to usenet groups. *Communications of the ACM*, 40(3):77–87, 1997.

[8] Y. Lim and Y.-W. Teh. Variational Bayesian approach to movie rating prediction. In *Proceedings of KDD Cup and Workshop*, 2007.

[9] S. Nakajima and Sugiyama. Theoretical analysis of Bayesian matrix factorization. *Journal of Machine Learning Research*, 12:2579–2644, 2011.

[10] L. Paninski, J. Pillow, and E. Simoncelli. Maximum likelihood estimation of a stochastic integrate-and-fire neural encoding model. *Neural Computation*, 16:2533–2561, 2004.

[11] T. Raiko, A. Ilin, and J. Karhunen. Principal component analysis for large scale problems with lots of missing values. In J. Kok, J. Koronacki, R. Lopez, S. Matwin, D. Mladenic, and A. Skowron, editors, *European Conference on Machine Learning 18*, pages 691–698. Springer, 2007.

[12] M. Rattray, O. Stegle, K. Sharp, and J. Winn. Inference algorithms and learning theory for Bayesian sparse factor analysis. *Journal of Physics: Conference Series*, 197(012002), 2009.

[13] G. Reinsel and R. Velu. *Multivariate Reduced-Rank Regression: Theory and Applications*. Springer, 1st edition, 1998.

[14] R. Rosipal and N. Krämer. Overview and recent advances in partial least squares. In *Subspace, Latent Structure and Feature Selection Techniques*. Springer, 2006.

[15] R. Salakhutdinov and A. Mnih. Probabilistic matrix factorization. In J. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems 20*, pages 1257–1264. Curran Associates, 2008.

[16] L. Saul, Y. Weiss, and L. Bottou, editors. *Advances in Neural Information Processing Systems 17*. MIT Press, 2005.

[17] N. Srebro, J. Rennie, and T. Jaakkola. Maximum margin matrix factorization. In Saul et al. [16], pages 1329–1336.

[18] M. Tao and X. Yuan. Recovering low-rank and sparse components of matrices from incomplete and noisy observations. *SIAM J. Optim.*, 21(1):57–81, 2011.

[19] M. Tipping and C. Bishop. Probabilistic principal component analysis. *Journal of Roy. Stat. Soc. B*, 61(3):611–622, 1999.

[20] R. Tomioka, T. Suzuki, M. Sugiyama, and H. Kashima. An efficient and general augmented Lagrangian algorithm for learning low-rank matrices. In J. Fürnkranz and T. Joachims, editors, *International Conference on Machine Learning 27*, pages 1087–1094. Omni Press, 2010.