Group-sparse Embeddings in Collective Matrix Factorization

Arto Klami ARTO.KLAMI@CS.HELSINKI.FI
Helsinki Institute for Information Technology HIIT, Department of Information and Computer Science, University of Helsinki

Guillaume Bouchard GUILLAUME.BOUCHARD@XRCE.XEROX.COM
Xerox Research Centre Europe

Abhishek Tripathi ABISHEK.TRIPATHI3@XEROX.COM
Xerox Research Centre India

Abstract

CMF is a technique for simultaneously learning low-rank representations based on a collection of matrices with shared entities. A typical example is the joint modeling of user-item, item-property, and user-feature matrices in a recommender system. The key idea in CMF is that the embeddings are shared across the matrices, which enables transferring information between them. The existing solutions, however, break down when the individual matrices have low-rank structure not shared with others. In this work we present a novel CMF solution that allows each of the matrices to have a separate low-rank structure that is independent of the other matrices, as well as structures that are shared only by a subset of them. We compare MAP and variational Bayesian solutions based on alternating optimization algorithms and show that the model automatically infers the nature of each factor using group-wise sparsity. Our approach supports in a principled way continuous, binary and count observations and is efficient for sparse matrices involving missing data. We illustrate the solution on a number of examples, focusing in particular on an interesting use-case of augmented multi-view learning.

1. INTRODUCTION

Matrix factorization techniques provide low-rank vectorial representations by approximating a matrix $\mathbf{X} \in \mathbb{R}^{n \times d}$ as the outer product of two rank-k matrices $\mathbf{U}_1 \in \mathbb{R}^{n \times k}$ and $\mathbf{U}_2 \in \mathbb{R}^{d \times k}$ (Fig. 1-I). This formulation encompasses a multitude of standard data analysis models from PCA and factor analysis to more recent models such as NMF (Paatero and Tapper, 1994; Lee and Seung, 2001) and various sophisticated factorization models proposed for recommender system applications (Mnih and Salakhutdinov, 2007; Koren et al., 2009; Sarwar et al., 2000).

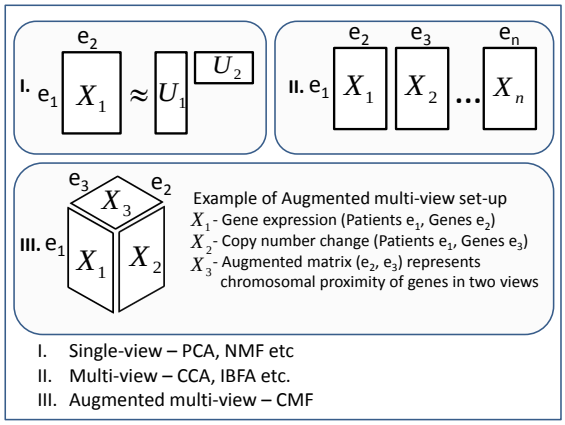

Figure 1. Examples of matrix factorization setups.

Many data analysis tasks call for more complex setups. Multi-view learning (Fig. 1-II) considers scenarios with multiple matrices $\mathbf{X}_m$ that share the same row entities but differ in the column entities; for example, $\mathbf{X}_1$ might contain ratings given for d_1 different movies by n different users, whereas $\mathbf{X}_2$ represents the same n users with d_2 profile features. For such setups the appropriate approach is to factorize the set of matrices $\{\mathbf{X}_m\}$ simultaneously so that (at least some of) the factors in $\mathbf{U}_1$ are shared across the matrices. Models that share all of the factors are fundamentally equivalent to simple factorizations of a concatenated matrix $\mathbf{X} = [\mathbf{X}_1, ..., \mathbf{X}_m]$. To reach a richer class of models one needs to allow each matrix to have also private factors, i.e. factors independent of the other matrices (Jia et al., 2010; Virtanen et al., 2012). For the case

of $M = 2$ the distinction is crystallized by the inter-battery factor analysis (IBFA) formulation of Klami et al. (2013).

Even more general setups with arbitrary collections of matrices that share some sets of entities have been proposed several times by different authors, under names such as co-factorization or multi-relational matrix factorization, and most end up being either a variant of tensor factorization of knowledge bases (Nickel et al., 2011; Chen et al., 2013) or a special case of *Collective Matrix Factorization* (CMF; Singh and Gordon, 2008). In this paper, we concentrate on the CMF model, i.e. on bilinear forms, but the ideas can be easily extended to three-way interactions, i.e. tensors. A prototypical example of CMF, illustrated by Bouchard et al. (2013), would be a recommender system setup where the target matrix X_1 is complemented with two other matrices associating the users and items with their own features. If the users and items are described with the same features, for example by proximities to geographical locations, the setup becomes circular. Another interesting use case for such circular setups is found in augmenting multi-view learning, in scenarios where additional information is provided on relationships between the features of two (or more) views. Figure 1-III depicts an example where the two views $\mathbf{X}_1$ and $\mathbf{X}_2$ represent expression and copy number alteration of the same patients. Classical multi-view solutions to this problem would ignore the fact that the column features for both views correspond to genes. With CMF, however, we can encode this information as a third matrix $\mathbf{X}_3$ that provides chromosomal promixity of the probes used for measuring the two views. Even though this kind of setup is very common in practical multi-view learning, the problem of handling such relationships has not attracted much attention.

Several solutions for the CMF problem have been presented. Singh and Gordon (2008) provided a maximum likelihood solution, Singh and Gordon (2010) and Yin et al. (2013) used Gibbs sampling to approximate the posterior, and Bouchard et al. (2013) presented a convex formulation of the problem. While all of these earlier solutions to the CMF problem provide meaningful factorizations, they share the same problem as the simplest solutions to the multi-view setup; they assume that all of the matrices are directly related to each other and that every factor describes variation in all matrices. Such strong assumptions are unlikely to hold in practical applications, and consequently the methods break down for scenarios where the individual matrices have strong view-specific noise or, more generally, any subset of the matrices has structure independent of the others. In this work we remove the shortcoming by introducing a novel CMF solution that allows also factors private to arbitrary subsets of the matrices, by adding a group-wise sparsity constraint for the factors.

We use group-wise sparse regularization of factors, where the groups corresponds to all the entities with the same type. In the Bayesian setting, this group-regularization is obtained by using automatic relevance determination (ARD) for controlling factor actity (Virtanen et al., 2012). This regularization enables us to automatically learn the nature of each factor, resulting in a solution free of tuning parameters. The model supports arbitrary schemas for the collection of matrices, as well as multiple likelihood potentials for various types of data (binary, count and continous), using the quadratic lower bounds provided by Seeger and Bouchard (2012) for non-Gaussian likelihoods.

To illustrate the flexibility of the CMF setup we discuss interesting modeling tasks in Section 6. We pay particular attention to the augmented multi-view learning setup of Figure 1-III, showing that CMF provides a natural way to improve on standard multi-view learning when the different views lay in related observation spaces. We also show experimentally the key advantage of ARD used for complexity control, compared to computationally intensive cross-validation of regularization parameters.

2. COLLECTIVE MATRIX FACTORIZATION

Given a set of M matrices $\mathbf{X}_m = [x_{ij}^{(m)}]$ describing relationships between E sets of entities (with cardinalities d_e), the goal of CMF is to jointly approximate the matrices with low-rank factorizations. We denote by r_m and c_m the entity sets corresponding to the rows and columns, respectively, of the m-th matrix. For a simple matrix factorization we have $M = 1$, $E = 2$, $r_m = 1$, and $c_m = 2$ (Fig. 1-I). Multi-view setups, in turn, have $E = M + 1$, $r_m = 1\ \forall m$, and $c_m \in \{2, ..., M + 1\}$ (Fig. 1-II). Some non-trivial CMF setups are depicted in Figures 1-III and 2.

2.1. Model

We approximate each matrix with a rank-K product plus additional row and column bias terms. For linear models, the element corresponding to the row i and column j of the m-th matrix is given by:

$$x_{ij}^{(m)} = \sum_{k=1}^{K} u_{ik}^{(r_m)} u_{jk}^{(c_m)} + b_i^{(m,r)} + b_j^{(m,c)} + \varepsilon_{ij}^{(m)} , \quad (1)$$

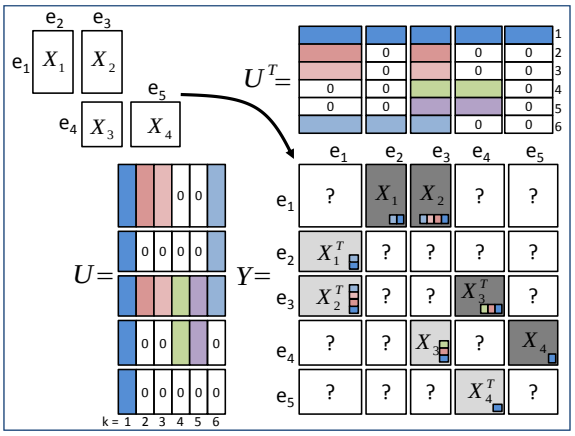

Figure 2. CMF setup encoded as a symmetric matrix factorization, with factors identified by colors. The zero patterns in the $\mathbf{U}$ matrix induce private factors in the resulting $\mathbf{Y}$ matrix. Contribution of factors are identified by small color patches next to the $\mathbf{X}$ matrices, and the question marks (?) represent missing data.

where $\mathbf{U}_e = [u_{ik}^{(e)}] \in \mathbb{R}^{d_e \times K}$ is the low-rank matrix related to the entity set e, $b_i^{(m,r)}$ and $b_j^{(m,c)}$ are the bias terms for the mth matrix, and $\varepsilon_{ij}^{(m)}$ is element-wise independent noise. We immediately see that any two matrices sharing the same entity set use the same low-rank matrix as part of their approximation, which enables sharing information.

The same model can also be expressed in a simpler form by crafting a single large symmetric observation matrix $\mathbf{Y}$ that contains all $\mathbf{X}_m$, following the representation introduced by Bouchard et al. (2013). We will use this representation because it allows implementing the private factors via group-wise sparsity. We create one large entity set with $d = \sum_{e=1}^{E} d_e$ entities and then arrange the observed matrices $\mathbf{X}_m$ into $\mathbf{Y}$ such that the blocks not corresponding to any $\mathbf{X}_m$ are left unobserved. The resulting $\mathbf{Y}$ is of size $d \times d$ but has only (at most) $\sum_{m=1}^{M} d_{r_m} d_{c_m}$ unique observed elements. In particular, the blocks relating the entities of one type to themselves are not observed.

The CMF model can then be formulated as a symmetric matrix factorization (see Figure 2)

$$\mathbf{Y} = \mathbf{U}\mathbf{U}^T + \boldsymbol{\varepsilon}, \tag{2}$$

where $\mathbf{U} \in \mathbb{R}^{d \times K}$ is a column-wise concatenation of all of the different $\mathbf{U}_e$ matrices, and the bias terms are dropped for notational simplicity. The noise $\boldsymbol{\varepsilon}$ is now symmetric but still independent over the upper-diagonal elements, and the variance depends on the block the element belongs to. Given this reformulation, any symmetric matrix factorization technique capable of handling missing data can be used to solve the CMF problem; the fact that the blocks along the diagonal are unobserved will usually be crucial here, since it means that no quadratic terms will be involved in the optimization. In Section 4.2 a variational Bayesian approximation is introduced to learn the model, but before we explain how the basic formulation needs to be extended to allow matrix-specific low-rank variations.

3. Group-wise sparse CMF

3.1. Private factors in CMF

Without further restrictions the solutions to (2) tie all matrices to each other; for each factor k the corresponding column of $\mathbf{U}$ has non-zero values for entities in every set e. This is undesirable for many practical CMF applications where the individual matrices are likely to have structured noise independent of other matrices. Since the structured noise cannot be captured by the element-wise independent noise terms $\boldsymbol{\varepsilon}$, the model will need to introduce new factors for modeling the variation specific to one matrix alone.

We use the following property of the basic CMF model: if the k-th columns of the factor matrices $\mathbf{U}_e$ are null for all but two entity types r_m and c_m, it implies that the k-th factor impacts only the matrix $\mathbf{X}_m$, i.e. the factor k is a *private* factor for relation m. To allow the automatic creation of these private factors, we put group-sparse priors on the columns of the matrices $\mathbf{U}_e$. Using the symmetric representation, this approach creates group-sparse factorial representations similar to the one represented in Figure 2. Note that if more than two groups of variables are non-zero for a given factor k, it means that it is private for a group of matrices rather than a single matrix, and the standard CMF is obtained if no groups equal to zero. In Figure 2 the first factor is a global factor as used in the standard CMF, since it is non-zero everywhere, and the rest are private to some matrices. Note that the last factor represented in light-blue in ($k = 6$) is interesting because it is a private factor overlapping multiple matrices ($\mathbf{X}_1$ and $\mathbf{X}_2$) rather than a single one for the other private factors (matrix $\mathbf{X}_1$ for factors 2 and 3, matrix $\mathbf{X}_3$ for factors 4 and 5).

To emphasize the group-wise sparsity structure in implementing the private factors, we use the abbreviation gCMF for group-wise sparse CMF i.e. a CMF model with this ability to learn separate private factors.

3.2. Probabilistic model for gCMF

We instantiate the general model by specifying Gaussian likelihood and normal-gamma priors for the pro-

jections, so that in (1) we have

$$\varepsilon_{ij}^{(m)} \sim \mathcal{N}(0, \tau_m^{-1}), \qquad \tau_m \sim \mathcal{G}(p_0, q_0),$$
$$u_{ik}^{(e)} \sim \mathcal{N}(0, \alpha_{ek}^{-1}), \qquad \alpha_{ek} \sim \mathcal{G}(a_0, b_0).$$

where e is the entity set that contains the entity i. The crucial element here is the prior for $\mathbf{U}$. Its purpose is to automatically select for each factor a set of matrices for which it is active, which it does by learning large precision α_{ek} for factors k that are not needed for modeling variation for entity set e. In particular, the prior takes care of matrix-specific low-rank structure, by learning factors for which α_{ek} is small for only two entity sets corresponding to one particular matrix.

For the bias terms we use a hierarchical prior

$$b_i^{(m,r)} \sim \mathcal{N}(\mu_{rm}, \sigma_{rm}^2), \quad b_j^{(m,c)} \sim \mathcal{N}(\mu_{cm}, \sigma_{cm}^2),$$
$$\mu_{\cdot m} \sim \mathcal{N}(0, 1), \qquad \sigma_{\cdot m}^2 \sim \mathcal{U}[0, \infty].$$

The hierarchy helps especially in modeling rows (and equivalently columns) with lots of missing data, and in particular provides reasonable values also for rows with no observations (the cold-start problem of new users in recommender systems) through μ_{rm}.

4. LEARNING

4.1. MAP solution

Providing a MAP estimate for the model is straighforward, but results in a practical challenge of needing to choose the hyper-parameters $\{a_0, b_0, p_0, q_0\}$, usually through cross-validation. This is particularly difficult for setups with several heterogeneous data matrices on arbitrary scales. Then large hyper-priors are needed for preventing overfitting, which in turn makes it difficult to push α_{ek} to sufficiently large values to make the factors private to subsets of the matrices. Hence, we proceed to explain more reasonable variational approximation that avoids these problems.

4.2. Variational Bayesian inference

It has been noticed that Bayesian approaches which take into account the uncertainty about the values of the latent variables lead to increased predictive performance (Singh and Gordon, 2010). Another important advantage of Bayesian learning is the ability to automatically select regularization parameters by maximizing the data evidence. While existing Bayesian approaches for CMF used MCMC techniques for learning, we propose here to use variational Bayesian learning (VB) by minimizing the KL divergence between a tractable approximation and the true observation probability. We use a fully factorized approximation similar to what Ilin and Raiko (2010) prsented for Bayesian PCA with missing data, and implement non-Gaussian likelihoods using the quadratic bounds by Seeger and Bouchard (2012). In the following we will summarize the main elements of the algorithm, leaving some of the technical details to these original sources.

Gaussian observations For Gaussian data we approximate the posterior with

$$Q(\Theta) = \left[\prod_{e=1}^{E}\prod_{k=1}^{K}\left(q(\alpha_{ek})\prod_{i=1}^{d_e} q(u_{ik}^{(e)})\right)\right] \tag{3}$$
$$\left[\prod_{m=1}^{M} q(\tau_m)q(\mu_{rm})q(\mu_{cm})\prod_{i=1}^{d_{rm}} q(b_i^{(m,r)})\prod_{j=1}^{d_{cm}} q(b_j^{(m,c)})\right].$$

Here $q(\alpha)$ and $q(\tau)$ are Gamma distributions, whereas the others are normal distributions. For all other parameters we use closed-form updates, but $\bar{\mathbf{U}}_e$, the mean parameters of $q(\mathbf{U}_e)$, are updated with Newton's method for each factor at a time. The gradient-based updates are used because for observation matrices with missing entries closed-form updates would be available only for each element $\bar{u}_{ik}^{(e)}$ separately, which would result in very slow convergence (Ilin and Raiko, 2010). The update rules for $Q(\Theta)$ are in the supplementary material.

Non-Gaussian observations For non-Gaussian data we use the approximation schema presented by Seeger and Bouchard (2012), adaptively approximating non-Gaussian likelihoods with spherical-variance Gaussians. This allows an optimization scheme that alternates between two steps: (i) updating $Q(\Theta)$ given pseudo-data $\mathbf{Z}$ (which is assumed Gaussian), and (ii) updating the pseudo-data $\mathbf{Z}$ by optimizing a quadratic term lower-bounding the desired likelihood potential. The full derivation of the approach is provided by Seeger and Bouchard (2012), but the resulting equations as applied to gCMF are summarized below. We update the pseudodata with

$$\boldsymbol{\xi}_m = E[\mathbf{U}_{r_m}]E[\mathbf{U}_{c_m}]^T,$$
$$\mathbf{Z}_m = (\boldsymbol{\xi}_m - f_m'(\boldsymbol{\xi}_m)/\kappa_m),$$

where the updates are element-wise and independent for each matrix. Here $f_m'(\boldsymbol{\xi}_m)$ is the derivative of the m-th link function $-\log p(\mathbf{X}_m|\mathbf{U}_{r_m}\mathbf{U}_{c_m}^T)$ and κ_m is the maximum value of the second derivative of the same function. Given the pseudo-data $\mathbf{Z}$, the approximation $Q(\Theta)$ can be updated as in the Gaussian case, using $\tau_m = \kappa_m$ as the precision. Note that the link

functions can be different for different observation matrices, which adds support for heterogeneous data; in Section 7 we illustrate binary and count data.

5. RELATED WORK

For $M = 1$ the model is equivalent to Bayesian (exponential family) PCA. In particular, it reduces to gradient-based optimization for the model by Seeger and Bouchard (2012). For this special case it is typically advisable to use their SVD-based algorithm, since it provides closed-form solution for the Gaussian case.

For multi-view setups where every matrix shares the same row-entities the model equals Bayesian inter-battery factor analysis (when $M = 2$) (Klami et al., 2013) and its extension group-factor analysis (when $M > 2$) (Virtanen et al., 2012). However, our inference solution has a number of advantages. In particular, our solution supports wider range of likelihood potentials and provides efficient inference for missing data. These improvements suggests that the proposed algorithm should be preferred over the earlier solutions.

The most closely related methods are the earlier CMF solutions, in particular the ones presented in the probabilistic framework. The early solutions by Lippert et al. (2008) and Singh and Gordon (2008) provide only maximum-likelihood solutions, whereas Singh and Gordon (2010) provided fully Bayesian solution by formulating CMF as a hierarchical model. They use normal-Inverse-Wishart priors for the factors, with spherical hyper-prior for the Inverse-Wishart distribution. This implies each factor is assumed to be roughly equally important in describing each of the matrices, and that their model will not provide matrix-specific factors as our model does. For inference they use computationally heavy Metropolis-Hastings. Their model also supports arbitrary likelihood potentials and arbitrary CMF schemas, though their experiments are limited to cases with $M = 2$.

6. USE CASES

Even though CMF is widely applicable to factorization of arbitrary matrix collections, it is worth describing some typical setups to illustrate common use cases where data analysis practitioners might find it useful.

Augmenting multi-view learning In multi-view learning (Fig. 1-II) the row entities are shared, but the column entities in different views are arbitrary. In many practical applications, however, the column entities share some obvious relationships that are ignored by the multi-view matrix factorization models. A common example considers computing CCA between two different high-throughput systems biology measurements of the same patients, so that both matrices are patients times genes (see, e.g., Witten and Tibshirani, 2009). In natural language processing, in turn, we have setups with different languages as row entities and words as column entities (Tripathi et al., 2010). In both cases there are obvious relationships between the column features. In the first example it is an identity relation, whereas in the latter lexigographic or dictionary-based information provides proximity relations for the column entities. Yet another example can be imagined in joint analysis of multiple brain imaging modalities; the column entities correspond to brain regions that have spatial relationships even though the level of representation might be very different when, e.g., analyzing fMRI and EEG data jointly (Correa et al., 2010).

Such relationships between the column entities can easily be taken into account with CMF using the cyclical relational schema of Figure 1-III. We call this approach *augmented multi-view learning*. We can encode any kind of similarity between the features as long as the resulting matrix can reasonably be modeled as low-rank. In the experimental section we will demonstrate setups where the features live in a continuous space (genes along the chromosome, pixels in a two-dimensional space) and hence we can measure distances between them. We then convert these distances into binary promixity relationships, to illustrate that already that is sufficient for augmenting the learning.

Recommender systems The simplest recommender systems seek to predict missing entries in a matrix of ratings or binary relevance indicators (Koren et al., 2009). The extensive literature on recommender systems indicates that incorporating additional information on the entities helps making such predictions (Stern et al., 2009; Fang and Si, 2011). CMF is a natural way of encoding such information, in form of additional matrices between the entities of interest and some features describing them.

While many other techniques can also be used for incorporating additional information about the entities, the CMF formulation opens up two additional types of extra information not easily implemented by the alternative means. The first is a circular setup where both the row and column entities of the matrix of interest are described by the same features (Bouchard et al., 2013). This is typically the case for example in social interaction recommenders where both rows and columns correspond to human individuals. The

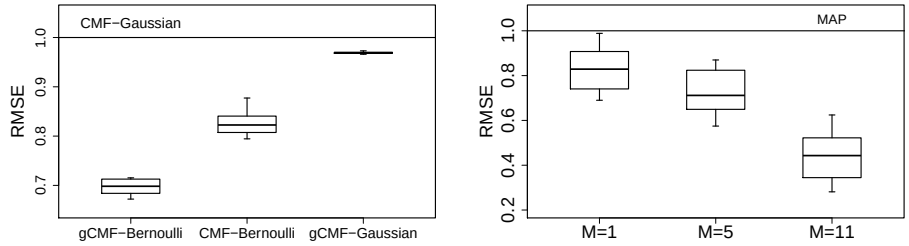

Figure 3. **Left:** Relative error for a circular setup of $M = 5$ binary matrices (see text for details), scaled so that CMF with Gaussian likelihood has error of one. The correct likelihood helps for both gCMF and CMF and modeling the private factors helps for both likelihoods, the combined gain of both aspects being 30%. The results are similar for other values of $M > 1$. **Right:** Relative error of VB vs MAP, scaled so that zero corresponds to the ground truth and one to the error of the MAP solution. For small M MAP can still compete (though it is worse than VB already for $M = 1$), but for large M it becomes worthless; for $M = 11$ VB reduces the error to roughly half. Furthermore, VB requires no tuning parameters, whereas for the MAP solution we needed to perform cross-validation over two regularization parameters.

other interesting formulation uses higher-order auxiliary data. For example, the movies in a classical recommender system can be represented by presence of actors, whereas the actors themselves are then represented by some set of features. This leads to a chain of matrices providing more indirect information on the relationships between the entities.

7. EXPERIMENTS

We start with technical validations showing the importance of choosing the correct likelihood potential and incorporating private factors in the model, as well as the advantages variational approximation provides over MAP estimation. We then proceed to show how CMF outperforms classical multi-view learning methods in scenarios where we can augment the setup with between-feature relationships.

Since the main goal is to demonstrate the conceptual importance of solving the CMF task with private factors, we use special cases of gCMF as comparison methods. This helps to show that the difference is really due to the underlying idea instead of the inference procedure; for example, when comparing against Singh and Gordon (2010) the effects could be masked by differences between Metropolis-Hastings and variational approximation that are here of secondary importance.

The closest comparison method, denoted by CMF, is obtained by forcing α_{ek} to be a constant α_k for every entity type e. It corresponds to the VB solution of the earlier CMF models and hence does not support private factors. For the augmented multi-view setup we will also compare against the special cases of gCMF and CMF that use only two matrices over the three entity sets, denoting them by CCA and PCA, respectively. Finally, in one experiment we will also compare against gCMF without the bias terms, to illustrate their importance in recommender systems. For all methods we use sufficiently large K, letting ARD prune out unnecessary components, and run the algorithms until the variational lower bound converges. We measure the error by root mean square error (RMSE), relative to one of the methods in each experiment.

7.1. Technical illustration

We start by demonstrating the difference between the proposed model and classical CMF approaches on an artificial data. We sample M binary matrices that form a cycle over M entity sets (of sizes $100 - 150$), so that the first matrix is between the entity sets 1 and 2, the second between the entity sets 2 and 3, and finally the last one is between the M-th and first entity set. We generate datasets that have 5 factors shared by all matrices plus two factors of low-rank noise specific to each matrix. This results in $5 + 2M$ true factors, and we learn the models with $10+2M$ factors, letting ARD prune out the extra ones.

Figure 3 (left) shows the accuracy in predicting the missing entries (40% of all) for gCMF as well as a standard CMF model. For both models we show the results for both (incorrect) Gaussian and Bernoulli likelihoods. The experiment verifies the expected results: Using the correct likelihood improves the accuracy, as does correctly modeling private noise factors.

We use the same setup to illustrate the importance of using variational approximation for inference, this time with Gaussian noise and entity set sizes between $40 - 80$. For MAP we validate the strength of the Gamma hyper-priors for τ and α over a grid of 11×11 values for $a_0 = b_0$ and $p_0 = q_0$, using two-fold cross-

validation within the observed data. In total we hence need to run the MAP variant more than 200 times to get the result, in contrast to the single run of the VB algorithm with vague priors using 10^{-10} for every parameter. Figure 3 (right) shows that despite heavy cross-validation the MAP setup is always worse and the gap gets bigger for more complex setups. This illustrates how the VB solution with no tunable hyperparameters is even more crucial for CMF than it would be for simpler matrix factorizations. For MAP using the same hyper-priors for all matrices necessarily becomes a compromise for matrices of different scales, whereas validating separate scales for each matrix would be completely infeasible (requiring validation over $2M$ parameters).

7.2. Augmented multi-view learning

We start with a multi-view setup in computational biology, using data from Pollack et al. (2002) and the setup studied by Klami et al. (2013). The samples are 40 patients with breast cancer, and the two views correspond to high-throughput measurements of expression and copy number alteration for 4287 genes. We compare the models in the task of predicting random missing entries in both views, as a function of the proportion of missing data.

The multi-view methods use the data as such, whereas the CMF variants also use a third $d_2 \times d_3$ matrix that encodes the proximity of the genes in the two views. It is a binary matrix such that $x^{(3)}_{i,j}$ is one with probability $\exp(-|l_i - l_j|)$, where l_i is the chromosomal location measured in 10^7 basepairs. This encodes the reasonable assumption that copy number alterations are more likely to influence the expression of nearby genes. Figure 4 shows how this information helps in making the predictions. For reasonable amounts of missing data, gCMF is consistently the best method, outperforming both CMF as well as the standard multi-view methods. For extreme cases with at least 80% missing data the advantage is finally lost. The importance of the private factors is seen also in CCA outperforming PCA, whereas CMF and CCA are roughly as accurate; both include one of the strenghts of gCMF.

In another example we model images of faces taken in two alternative lighting conditions, but from the same viewing angle. We observe the raw grayscale pixels values of 50×50 images, and for the CMF methods we use a third matrix (size 2500×2500, of which random 10% is observed) to encode proximity of pixels in the two views, using Gaussian kernel to provide the probability of one for a binary relation. We train the model so that we have observed 6 images in both views

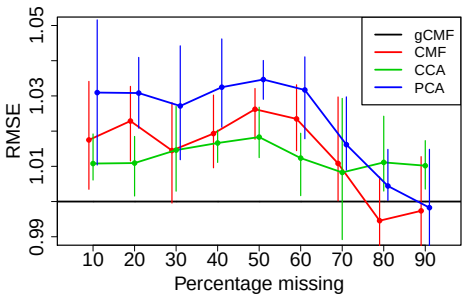

Figure 4. Relative prediction error for augmented multi-view gene experiment, scaled so that gCMF has error one and is represented by the horizontal black line. For reasonable amounts of missing data (x-axis) the methods with private factors (gCMF and CCA) outperform the ones without, and modeling the proximity relationship between the genes (gCMF and CMF) improves the accuracy. The confidence intervals correspond to 10% and 90% quantiles over random choices of missing data.

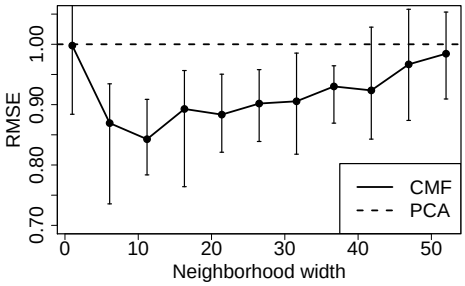

Figure 5. Prediction error for a multi-view image reconstruction task as a function of the neighborhood width in constructing the proximity augmentation view. The augmentation helps for a wide range of promixity relationships, and the solution reverts back to the non-augmented accuracy for very narrow and wide neighborhoods.

and then 7 images for each view alone, for a total of 20 images. The task is to predict the missing views for these images, without any observations.

Figure 5 plots the prediction errors as a function of the neighborhood σ used in constructing the promity relationships. We see that for very narrow and very wide neighborhoods the CMF approach reverts back to the classical multi-view model, since the extra view consists almost completely of zeros or ones, respectively. For proper neighborhood relationships the accuracy in predicting the missing view is considerably improved.

7.3. Recommender systems

Next we consider classical recommender systems, using MovieLens and Flickr data as used in earlier CMF experiments by Bouchard et al. (2013). We compare

Table 1. RMSE for two recommender system setups, with boldface indicating the best results. The results for convex CMF (CCMF) are taken from Bouchard et al. (2013) for the best regularization parameter values. Our model provides comparable result without the bias terms, without needing any tuning for the parameters, and the bias terms helps considerably with the cold-start problem especially in MovieLens. Without bias terms gCMF also outperforms CMF for all cases, but with the bias terms the methods are practically identical for these data sets. This suggests these data sets do not have strong private structure that could not be modeled with the bias terms alone. It is important to note that allowing for the private factors never hurts; gCMF is always at least as good as CMF.

Data	MovieLens	Flickr				
Relation	$\mathbf{X}_1$-Count	$\mathbf{X}_1$-Binary	$\mathbf{X}_2$-Binary	$\mathbf{X}_3$-Binary	$\mathbf{X}_4$-Binary	$\mathbf{X}_5$-Gaussian
CCMF (reg=10)	1.0588	0.7071	0.2473	0.3661	0.2384	**1.0033**
CMF without bias	1.0569	0.5120	0.2324	0.5093	0.2176	1.0092
CMF with bias	0.9475	**0.5000**	0.2369	**0.2789**	**0.2109**	**1.0033**
gCMF without bias	1.0418	0.5003	**0.2291**	0.5014	0.2167	1.0039
gCMF with bias	**0.9474**	**0.5000**	0.2369	**0.2789**	**0.2109**	**1.0033**

gCMF with the convex CMF solution presented in that paper, showing that it finds the same solution when the bias terms are turned off (Table 1). We also illustrate that modeling the bias terms explicitly is as useful for CMF as it has been shown to be for other types of recommender systems. To our knowledge gCMF is the first CMF solution with such bias terms.

Both data sets have roughly 1 million observed entries, and our solutions were computed in a few minutes on a laptop. The total computation time is hence roughly comparable to the times Bouchard et al. (2013) reported for CCMF using one choice of regularization parameters. Full CCMF solution is considerably slower since it has to validate over them.

8. DISCUSSION

Collective matrix factorization is a very general technique for revealing low-rank representations for arbitrary matrix collections. However, the practical applicability of earlier solutions has been limited since they implicitly assume all factors to be relevant for all matrices. Here we presented a general technique for avoiding this problem, by learning the CMF solution as symmetric factorization of a large square matrix while enforcing group-wise sparse factors.

While any algorithm aiming at such sparsity structure will provide shared and private factors for a CMF, the variational Bayesian solution presented in this work has some notable advantages. It is more straighforward than the sampling-based alternative by Singh and Gordon (2010) (which could be modified to incorporate private factors) while being free of tunable regularization parameters required by the convex solution of Bouchard et al. (2013). The model also subsumes some earlier models and provides extensions for them. In particular, it can be used to efficiently learn Bayesian CCA solution for missing data and non-conjugate likelihoods, providing the first efficient Bayesian CCA between binary observations.

One drawback of CMF is its inability to handle multiple relations accross two entity type. Tensor factorization methods alleviate this problem, as illustrated in the recent work on multi-relational data (Glorot et al., 2013; Chen et al., 2013).

Acknowledgments

We acknowledge support from the University Affairs Committee of the Xerox Foundation. AK was also supported by Academy of Finland (grants 251170 and 266969) and Digile SHOK project D2I.

References

Guillaume Bouchard, Shengbo Guo, and Dawei Yin. Convex collective matrix factorization. In *Proceedings of the 16th International Conference on Artificial Intelligence and Statistics*, volume 31 of *JMLR W&CP*, pages 144–152. JMLR, 2013.

Danqi Chen, Richard Socher, Christopher D. Manning, and Andrew Y. Ng. Learning new facts from knowledge bases with neural tensor networks and semantic word vectors. In *International Conference on Learning Representations*, 2013.

Nicolle M. Correa, Tom Eichele, Tülay Adali, Yi-Ou Li, and Vince D. Calhoun. Multi-set canonical correlation analysis for the fusion of concurrent single trial ERP and functional MRI. *Neuroimage*, 50(4): 1438–1445, 2010.

Yi Fang and Luo Si. Matrix co-factorization for recommendation with rich side information and implicit

feedback. In *Proceedings of the 2nd International Workshop on Information Heterogeneity and Fusion in Recommender Systems*, pages 65–69. ACM, 2011.

Xavier Glorot, Antoine Bordes, Jason Weston, and Yoshua Bengio. A semantic matching energy function for learning with multi-relational data. In *International Conference on Learning Representations*, volume abs/1301.3485, 2013.

Alexander Ilin and Tapani Raiko. Practical approaches to principal component analysis in the presence of missing data. *Journal of Machine Learning Research*, 11:1957–2000, 2010.

Yangqing Jia, Mathieu Salzmann, and Trevor Darrell. Factorized latent spaces with structured sparsity. In *Advances in Neural Information Processing Systems 23*, pages 982–990, 2010.

Arto Klami, Seppo Virtanen, and Samuel Kaski. Bayesian canonical correlation analysis. *Journal of Machine Learning Research*, 14:965–1003, 2013.

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

Daniel D. Lee and H. Sebastian Seung. Algorithms for non-negative matrix factorization. In *Advances in neural information processing systems 13*, pages 556–562, 2001.

Christoph Lippert, Stefan-Hagen Weber, Yi Huang, Volker Tresp, Matthias Schubert, and Hans-Peter Kriegel. Relation-prediction in multi-relational domains using matrix-factorization. In *NIPS Workshop: Structured Input - Structured Output*. 2008.

Andriy Mnih and Ruslan Salakhutdinov. Probabilistic matrix factorization. In *Advances in neural information processing systems 20*, pages 1257–1264, 2007.

Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. A three-way model for collective learning on multi-relational data. In *ICML*, page 809816, 2011.

Pentti Paatero and Unto Tapper. Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values. *Environmetrics*, 5(2):111–126, 1994.

Jonathan R. Pollack, Therese Sorlie, and Charles M. Perou et al. Microarray analysis reveals a major direct role of DNA copy number alteration in the transcriptional program of human breast tumors. *Proceedings of the National Academy of Sciences of the United States of America*, 99(20):12963–12968, 2002.

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

Matthias Seeger and Guillaume Bouchard. Fast variational Bayesian inference for non-conjugate matrix factorization models. In *Proceedings of the 15th International Conference on Artificial Intelligence and Statistics*, volume 22, pages 1012–1018. JMLR, 2012.

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

Ajit P. Singh and Geoffrey J. Gordon. A Bayesian matrix factorization model for relational data. In *Uncertainty in Artificial Intelligence*, 2010.

David H Stern, Ralf Herbrich, and Thore Graepel. Matchbox: large scale online Bayesian recommendations. In *Proceedings of the 18th international conference on World wide web*, pages 111–120. ACM, 2009.

Abhishek Tripathi, Arto Klami, and Sami Virpioja. Bilingual sentence matching using kernel cca. In *Machine Learning for Signal Processing (MLSP), 2010 IEEE International Workshop on*, pages 130–135. IEEE, 2010.

Seppo Virtanen, Arto Klami, Suleiman A. Khan, and Samuel Kaski. Bayesian group factor analysis. In *Proceedings of the 15th International Conference on Artificial Intelligence and Statistics*, volume 22 of *JMLR W&CP*, pages 1269–1277. JMLR, 2012.

Daniela M. Witten and Robert J. Tibshirani. Extensions of sparse canonical correlation analysis with applications to genomic data. *Statistical applications in genetics and molecular biology*, 8(1):1–27, 2009.

Dawei Yin, Shengbo Guo, Boris Chidlovskii, Brian D Davison, Cedric Archambeau, and Guillaume Bouchard. Connecting comments and tags: improved modeling of social tagging systems. In *Proceedings of the sixth ACM international conference on Web search and data mining*, pages 547–556. ACM, 2013.

Group-sparse Embeddings in Collective Matrix Factorization

Supplementary material

This supplementary material for the manuscript "Group-sparse Embeddings in Collective Matrix Factorization" provides more details on the variational approximation described in the paper.

Notation

The factors in (3) are

$$
\begin{aligned}
q(u_{ik}^{(e)}) &= \mathcal{N}(\bar{u}_{ik}^{(e)}, \tilde{u}_{ik}^{(e)}),\\
q(\alpha_{ek}) &= \mathcal{G}(a_{ek}, b_{ek}),\ \ q(b_i^{(m,r)}) = \mathcal{N}(\bar{b}_i^{(m,r)}, \tilde{b}_i^{(m,r)}),\\
q(\tau_m) &= \mathcal{G}(p_m, q_m),\ \ q(b_j^{(m,c)}) = \mathcal{N}(\bar{b}_j^{(m,c)}, \tilde{b}_j^{(m,c)}).
\end{aligned}
$$

and we denote by $\bar{\alpha}$ and $\bar{\tau}$ the expectations of α and τ. The observed entries in $\mathbf{X}_m$ are given by $\mathbf{O}_m \in [0,1]^{d_{r_m} \times d_{c_m}}$, with $n_m = \sum_{ij} o_{ij}^{(m)}$ indicating their total number. Finally, we denote $\hat{x}_{ij}^{(m)} = \left(x_{ij}^{(m)} - \sum_{k=1}^{K} \bar{u}_{ik}^{(r_m)} \bar{u}_{jk}^{(c_m)} - \bar{b}_i^{(m,r)} - \bar{b}_j^{(m,c)}\right)$.

Algorithm

The full algorithm repeats the following steps until convergence.

1. For each entity set e, compute the gradient of $\bar{\mathbf{U}}_e$ using (4) and compute the variance parameter $\tilde{\mathbf{U}}_e$ using (5).

2. Update $\bar{\mathbf{U}}_e$ with under-relaxed Newton's step. The element-wise update is $\bar{u}_{ik}^{(e)} \leftarrow u_{ik}^{(e)} + \lambda(\tilde{u}_{ik}^{(e)})^{-1} g_{ik}^{(e)}$ with $0 < \lambda < 1$ as the regularization parameter.

3. Update the bias terms using (6).

4. Update the automatic relevance determination parameters using (7).

5. For all matrices $\mathbf{X}_m$ with Gaussian likelihood, update the noise precision parameters using (8). For all matrices $\mathbf{X}_m$ with non-Gaussian likelihood, update the pseudo-data using (9).

Details

Updates for the factors: The gradient with respect to the mean parameters of the factors is computed as

$$
\begin{aligned}
g_{ik}^{(e)} = \bar{\alpha}_{ek} \bar{u}_{ik}^{e} + &\\
\sum_{m; r_m = e} \bar{\tau}_m \sum_j \left[-\hat{x}_{ij}^{(m)} \bar{u}_{jk}^{(c_m)} + \bar{u}_{ik}^{(e)} \tilde{u}_{jk}^{(c_m)} \right] &\\
\sum_{m; c_m = e} \bar{\tau}_m \sum_j \left[-\hat{x}_{ij}^{(m)} \bar{u}_{ik}^{(r_m)} + \bar{u}_{jk}^{(e)} \tilde{u}_{ik}^{(r_m)} \right].&
\end{aligned} \tag{4}
$$

For $\tilde{\mathbf{U}}_e$ we have closed-form updates

$$
\begin{aligned}
\tilde{u}_{ik}^{(e)} = \Bigg[\bar{\alpha}_{ek} + \sum_{m; c_m = e} \bar{\tau}_m \sum_j \left((\bar{u}_{jk}^{(r_m)})^2 + \tilde{u}_{jk}^{(r_m)} \right) &\\
+ \sum_{m; r_m = e} \bar{\tau}_m \sum_j \left((\bar{u}_{jk}^{(c_m)})^2 + \tilde{u}_{jk}^{(c_m)} \right) \Bigg]^{-1}.&
\end{aligned} \tag{5}
$$

Updates for the bias terms: The row bias terms are updated as

$$
\begin{aligned}
\tilde{b}_i^{(m,r)} &= \left(\bar{\tau}_m \sum_j o_{ij}^{(m)} + \sigma^{-2} \right)^{-1},\\
\hat{b}_i^{(m,r)} &= \tilde{b}_i^{(m,r)} \left(\bar{\tau}_m \mu_i + \mu_{rm} / \sigma_{rm}^2 \right),
\end{aligned} \tag{6}
$$

where μ_i is a shorthand notation for the mean of $x_{ij}^{(m)} - \sum_k \bar{u}_{ik}^{(m)} \bar{u}_{jk}^{(m)} - \bar{b}_j^{(m,c)}$ over the observed entries. We additionally update $q(\mu_{rm})$ using standard variational update for Gaussian likelihood and prior, and use point estimate for σ_{rm}^2. The updates for the column bias terms follow naturally.

Updates for the ARD terms: The ARD variance parameters are updated as

$$
\begin{aligned}
a_{ek} &= a_0^{\alpha} + d_e/2,\\
b_{ek} &= b_0 + 0.5 \sum_{i=1}^{d_s} \sum_{k=1}^{K} \left((\bar{u}_{ik}^{(e)})^2 + \tilde{u}_{ik}^{(e)} \right).
\end{aligned} \tag{7}
$$

Updates for the precision terms: For each matrix with Gaussian likelihood the precision is updated as

$$p_m = p_0 + n_m/2, \tag{8}$$
$$q_m = q_0 + \frac{1}{2n_m}\sum_{ij}\left[(\hat{x}_{ij}^{(m)})^2 + \tilde{b}_i^{(m,r)} + \tilde{b}_j^{(m,c)} + \sum_{k=1}^{K}\left((\bar{u}_{ik}^{(r_m)})^2\tilde{u}_{jk}^{(c_m)} + (\bar{u}_{jk}^{(c_m)})^2\tilde{u}_{ik}^{(r_m)} + \tilde{u}_{jk}^{(c_m)}\tilde{u}_{ik}^{(r_m)}\right)\right],$$

where the sum for q_m is over all observed entries.

Updated for the pseudo-data: For each matrix with non-Gaussian data we update the pseudo-data $\mathbf{Z}_m$ using

$$\begin{aligned}\boldsymbol{\xi}_m &= E[\mathbf{U}_{r_m}]E[\mathbf{U}_{c_m}]^T, \\ \mathbf{Z}_m &= (\boldsymbol{\xi}_m - f_m'(\boldsymbol{\xi}_m)/\kappa_m),\end{aligned} \tag{9}$$

where the updates are element-wise and independent for each matrix. Here $f_m'(\boldsymbol{\xi}_m)$ is the derivative of the m-th link function $-\log p(\mathbf{X}_m|\mathbf{U}_{r_m}\mathbf{U}_{c_m}^T)$ and κ_m is the maximum value of the second derivative of the same function.

MAP estimation

These update rules can be easily modified to provide the MAP estimate instead; the modifications mostly consist of dropping the variance terms and the resulting updates are not repeated here. Similarly, the updates are easy to modify for learning CMF models without private factors, by coercing α_{ek} into α_k.