
Bayesian Hybrid Matrix Factorisation for Data Integration

Thomas Brouwer
University of Cambridge

Pietro Lio
University of Cambridge

Abstract

We introduce a novel Bayesian hybrid matrix factorisation model (HMF) for data integration, based on combining multiple matrix factorisation methods, that can be used for in- and out-of-matrix prediction of missing values. The model is very general and can be used to integrate many datasets across different entity types, including repeated experiments, similarity matrices, and very sparse datasets. We apply our method on two biological applications, and extensively compare it to state-of-the-art machine learning and matrix factorisation models. For in-matrix predictions on drug sensitivity datasets we obtain consistently better performances than existing methods. This is especially the case when we increase the sparsity of the datasets. Furthermore, we perform out-of-matrix predictions on methylation and gene expression datasets, and obtain the best results on two of the three datasets, especially when the predictivity of datasets is high.

1 INTRODUCTION

Matrix factorisation methods offer an elegant way of analysing datasets. Here, a matrix relating two entity types is decomposed into two smaller matrices (so-called latent factors) so that their product approximates the original one. This extracts hidden structure in the data, and allows the prediction of missing values. Non-negativity constraints are often imposed on the matrices (Lee and Seung [1999]) as this makes the results easier to interpret, and it is often inherent to the problem – such as in image processing (Lee and Seung [1999]) or bioinformatics (Brunet et al. [2004]).

Non-negative matrix tri-factorisation is an extension of these methods, first introduced by Ding et al. [2006], where the matrix is decomposed into three smaller matrices, which again are constrained to be non-negative. Both methods are shown in Figure 1.

A key question is how to best predict missing values in these datasets. There are two different settings for this problem. Firstly, **in-matrix predictions**, where if we are trying to predict an unknown value for a pair of drug D1 and cancer type C1, we will have at least one known value for D1 with another cancer type C2, and for C1 with another drug D1. The other setting is **out-of-matrix predictions**, where we predict values for entirely unseen rows or columns, such as a new drug for which we have no observed values inside the matrix. This is illustrated in Figure 1.

In practice we often have many different datasets, relating different entity types. Matrix factorisation methods can be effectively used for data integration, by jointly decomposing multiple datasets and sharing the latent matrices (Zhang et al. [2005]). This can improve our matrix factorisations, and hence our in-matrix predictions, and also allows us to do out-of-matrix predictions. Another approach, based on multiple matrix tri-factorisation, was introduced by Wang et al. [2008], where they shared two of the three latent matrices. By sharing more factors than the multiple matrix factorisation method, and hence having a much smaller dataset-specific matrix in the middle, we can more effectively integrate similar datasets. This is particularly interesting for integrating repeated experiments, where different biological labs perform similar experiments between the same two entity types, such as gene expression profiles and methylation levels. Both approaches are illustrated in Figure 2.

We propose a novel Bayesian model for data integration, which combines multiple matrix factorisation and tri-factorisation. Our method can integrate many datasets across different entity types, including repeated experiments, similarity matrices, and very sparse datasets. In our method, the user can specify for each dataset whether it should be decomposed into two matrices, in which case only the row factor

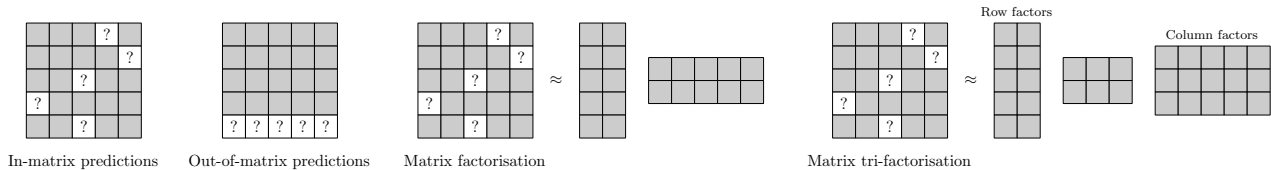


Figure 1: Difference between in- and out-of-matrix predictions for missing values in matrices; and the difference between matrix factorisation and matrix tri-factorisation.

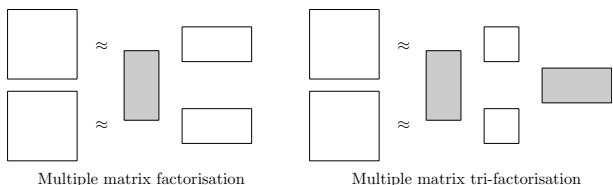


Figure 2: Difference between multiple matrix factorisation and multiple matrix tri-factorisation. The shared factor matrices are highlighted in grey.

matrices are shared, or into three, in which case the row and column matrices are shared. This gives a hybrid between matrix factorisation and tri-factorisation. Additionally, the user can also specify for each of the latent matrices whether the factors should be nonnegative or real-valued, giving a hybrid between nonnegative, semi-nonnegative, and real-valued factorisations. By using a probabilistic approach, our method can effectively handle missing values and predict them, both for in- and out-of-matrix predictions, and the Bayesian approach is much less prone to overfitting than non-probabilistic models. Furthermore, the rank of each matrix is automatically chosen using Automatic Relevance Determination, eliminating the need to perform model selection. Related work is discussed in Section 4.

To demonstrate our method, we apply it to two different settings. Firstly, we consider four drug sensitivity datasets, where the matrices are similar and hence have high predictivity. We measure the in-matrix predictive performance of our method, as well as Bayesian and non-probabilistic matrix factorisation methods, and several state-of-the-art machine learning methods. Our model consistently outperforms all other methods, especially when the sparsity of the data increases. Secondly, we integrate gene expression, promoter region methylation, and gene body methylation profiles for breast cancer patients. These datasets are much more dissimilar, hence predicting one dataset given the others is much harder. However, in out-of-matrix prediction experiments our method achieves better performance than state-of-the-art machine learning methods on two of the three combinations.

2 MATRIX FACTORISATION

The problem of non-negative matrix factorisation (NMF) can be formulated as decomposing a matrix $\mathbf{R} \in \mathbb{R}^{I \times J}$ into two latent (unobserved) factor matrices $\mathbf{F} \in \mathbb{R}_+^{I \times K}$, $\mathbf{G} \in \mathbb{R}_+^{J \times K}$. In other words, solving $\mathbf{R} = \mathbf{F}\mathbf{G}^T + \mathbf{E}$, where noise is captured by matrix $\mathbf{E} \in \mathbb{R}^{I \times J}$. Some entries in the dataset \mathbf{R} may not be known – we represent the indices of observed entries by the set $\Omega = \{(i, j) \mid R_{ij} \text{ observed}\}$. Similarly, non-negative matrix tri-factorisation (NMTF) can be formulated as finding three latent factor matrices $\mathbf{F} \in \mathbb{R}_+^{I \times K}$, $\mathbf{S} \in \mathbb{R}_+^{K \times L}$, $\mathbf{G} \in \mathbb{R}_+^{J \times K}$, such that $\mathbf{R} = \mathbf{F}\mathbf{S}\mathbf{G}^T + \mathbf{E}$.

Some NMF methods such as Lee and Seung [2001] rely on optimisation-based techniques, where a cost function between the observed matrix \mathbf{R} and the predicted matrix $\mathbf{F}\mathbf{G}^T$ is minimised, like the mean squared error or I -divergence, using multiplicative updates. Alternatively, probabilistic models formulate the problem of NMF by treating the entries in \mathbf{F} , \mathbf{G} as unobserved or latent variables, and the entries in \mathbf{R} as observed datapoints. Bayesian approaches furthermore place prior distributions over the latent variables. The problem then involves finding the distribution over \mathbf{F} , \mathbf{G} after observing \mathbf{R} , $p(\mathbf{F}, \mathbf{G} \mid \mathbf{R})$. This Bayesian approach has several benefits: it is less prone to overfitting, especially on small or sparse datasets; a distribution over the factors is obtained, rather than just a point estimate; it allows for flexible and elegant models (such as automatic model selection using Automatic Relevance Determination); and missing entries are easily handled (we simply do not include them in the observed data, through the Ω set introduced earlier). However, finding this posterior distribution can be very inefficient.

Schmidt et al. [2009] introduced a Bayesian model for non-negative matrix factorisation, by using Exponential priors and a Gaussian likelihood. For the precision τ of the likelihood they used a Gamma distribution with shape $\alpha > 0$ and rate $\beta > 0$. The full set of parameters for this model is denoted $\boldsymbol{\theta} = \{\mathbf{F}, \mathbf{G}, \tau\}$.

$$R_{ij} \sim \mathcal{N}(R_{ij} \mid \mathbf{F}_i \cdot \mathbf{G}_j, \tau^{-1})$$

$$F_{ik} \sim \mathcal{E}(F_{ik} \mid \lambda_F) \quad G_{jk} \sim \mathcal{E}(G_{jk} \mid \lambda_G) \quad \tau \sim \mathcal{G}(\tau \mid \alpha, \beta)$$

Inference to find the posterior $p(\mathbf{F}, \mathbf{G} | \mathbf{R})$ can be efficiently performed using Gibbs sampling. This method works by sampling new values for each parameter θ_i from its marginal $p(\theta_i | \boldsymbol{\theta}_{-i}, D)$ given the current values of the other parameters $\boldsymbol{\theta}_{-i}$, and the observed data D . If we sample new values in turn for each parameter θ_i from $p(\theta_i | \boldsymbol{\theta}_{-i}, D)$, we will eventually converge to draws from the posterior $p(\boldsymbol{\theta} | D)$, which can be used to approximate it. When doing so we have to discard the first n draws because it takes a while to converge (*burn-in*), and since consecutive draws are correlated we only use every i th value (*thinning*).

For this model we draw from the following distributions:

$$p(F_{ik} | \tau, \mathbf{F}_{-ik}, \mathbf{G}, D) \quad p(G_{jk} | \tau, \mathbf{F}, \mathbf{G}_{-jk}, D) \\ p(\tau | \mathbf{F}, \mathbf{G}, D)$$

where \mathbf{F}_{-ik} denotes all elements in \mathbf{F} except F_{ik} , and similarly for \mathbf{G}_{-jl} . Using Bayes' theorem we obtain the following posterior distributions:

$$p(\tau | \mathbf{F}, \mathbf{G}, D) = \mathcal{G}(\tau | \alpha^*, \beta^*) \\ p(F_{ik} | \tau, \mathbf{F}_{-ik}, \mathbf{G}, D) = \mathcal{TN}(F_{ik} | \mu_{ik}^F, \tau_{ik}^F) \\ p(G_{jk} | \tau, \mathbf{F}, \mathbf{G}_{-jk}, D) = \mathcal{TN}(G_{jk} | \mu_{jk}^G, \tau_{jk}^G),$$

where

$$\mathcal{TN}(x | \mu, \tau) = \begin{cases} \frac{\sqrt{\frac{\tau}{2\pi}} \exp\{-\frac{\tau}{2}(x - \mu)^2\}}{1 - \Phi(-\mu\sqrt{\tau})} & \text{if } x \geq 0 \\ 0 & \text{if } x < 0 \end{cases}$$

is a truncated normal: a normal distribution with zero density below $x = 0$ and renormalised to integrate to one. $\Phi(\cdot)$ is the cumulative distribution function of $\mathcal{N}(0, 1)$.

The extension of this model to non-negative matrix tri-factorisation is straightforward. We can also choose to remove the nonnegativity constraint, by instead using a Gaussian prior for the factor matrices. This results in a Gaussian posterior in the Gibbs sampling algorithm, with slightly different parameters. A semi-nonnegative model, with only one real-valued matrix (\mathbf{G} for MF, and \mathbf{S} for MTF), is illustrated below. Gibbs samplers for all mentioned models are given in the Supplementary Materials (Section 1).

$$\begin{array}{ll} \text{Prior:} & \text{Posterior:} \\ F_{ik} \sim \mathcal{E}(F_{ik} | \lambda_F) & F_{ik} \sim \mathcal{TN}(F_{ik} | \mu_{ik}^F, \tau_{ik}^F) \\ G_{jk} \sim \mathcal{N}(G_{jk} | 0, \lambda_G^{-1}) & G_{jk} \sim \mathcal{N}(G_{jk} | \mu_{jk}^G, (\tau_{jk}^G)^{-1}) \end{array}$$

3 HYBRID MATRIX FACTORISATION

The idea behind Hybrid Matrix Factorisation (HMF) is to integrate multiple datasets by jointly decomposing them, and sharing their latent factors. Formally,

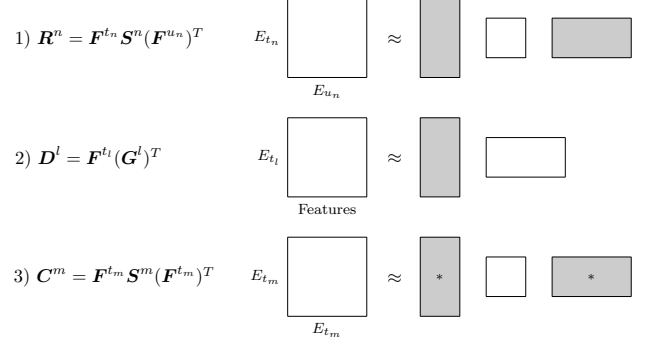


Figure 3: The three different types of datasets and factorisations. Shared factor matrices are grey, and dataset-specific ones are white. The two grey matrices for the third factorisation type are the same (*).

we are given a number of datasets spanning T different entity types E_1, \dots, E_T . Each entity type E_t has I_t instances, K_t factors, and a factor matrix $\mathbf{F}^t \in \mathbb{R}^{I_t \times K_t}$, which is shared across the matrix factorisations of datasets that relate this entity type. We consider three dataset types, which we decompose in different ways (see Figure 3):

1. Main datasets $\mathbf{R} = \{\mathbf{R}^1, \dots, \mathbf{R}^N\}$, relating two entity types, both of which we have other datasets for (such as features or repeated experiments). Each dataset $\mathbf{R}^n \in \mathbb{R}^{I_{t_n} \times I_{u_n}}$ relates entity types E_{t_n}, E_{u_n} . We use matrix tri-factorisation to decompose it into two entity type factor matrices $\mathbf{F}^{t_n}, \mathbf{F}^{u_n}$, and a dataset-specific matrix $\mathbf{S}^n \in \mathbb{R}^{K_{t_n} \times K_{u_n}}$.
2. Feature datasets $\mathbf{D} = \{\mathbf{D}^1, \dots, \mathbf{D}^L\}$, giving feature values for an entity type. Each dataset $\mathbf{D}^l \in \mathbb{R}^{I_{t_l} \times J_l}$ relates an entity type E_{t_l} to J_l features. We use matrix factorisation to decompose it into one entity type factor matrix \mathbf{F}^{t_l} , and a dataset-specific matrix $\mathbf{G}^l \in \mathbb{R}^{J_l \times K_{t_l}}$.
3. Similarity datasets $\mathbf{C} = \{\mathbf{C}^1, \dots, \mathbf{C}^M\}$, giving similarities between entities of the same entity type (such as Jaccard kernels). Each dataset $\mathbf{C}^m \in \mathbb{R}^{I_{t_m} \times I_{t_m}}$ relates an entity type E_{t_m} to itself. We use matrix tri-factorisation to decompose it into an entity type factor matrix \mathbf{F}^{t_m} , a dataset-specific matrix $\mathbf{S}^m \in \mathbb{R}^{K_{t_m} \times K_{t_m}}$, and \mathbf{F}^{t_m} again.

$$\begin{aligned} \mathbf{R}^n &= \mathbf{F}^{t_n} \mathbf{S}^n (\mathbf{F}^{u_n})^T + \mathbf{E}^n \\ \mathbf{D}^l &= \mathbf{F}^{t_l} (\mathbf{G}^l)^T + \mathbf{E}^l \\ \mathbf{C}^m &= \mathbf{F}^{t_m} \mathbf{S}^m (\mathbf{F}^{t_m})^T + \mathbf{E}^m \end{aligned}$$

The above formulation allows the user to very easily choose the kind of joint factorisation. By passing a set

of matrices as D_1, \dots, D_L , multiple matrix factorisation is performed. Instead, passing them as R_1, \dots, R_N gives multiple matrix tri-factorisation. A hybrid combination is also possible, as illustrated in Figure 4. Furthermore, each of the factor matrices can either be non-negative (using an exponential prior), or real-valued (using a Gaussian prior), additionally giving a hybrid of nonnegative, semi-nonnegative and real-valued matrix factorisation. The model likelihood functions are

$$\begin{aligned} R_{ij}^n &\sim \mathcal{N}(R_{ij}^n | \mathbf{F}_i^{t_n} \cdot \mathbf{S}^n \cdot \mathbf{F}_j^{u_n}, (\tau^n)^{-1}) \\ D_{ij}^m &\sim \mathcal{N}(D_{ij}^m | \mathbf{F}_i^{t_l} \cdot \mathbf{G}_j^l, (\tau^l)^{-1}) \\ C_{ij}^m &\sim \mathcal{N}(C_{ij}^m | \mathbf{F}_i^{t_l} \cdot \mathbf{S}^m \cdot \mathbf{F}_j^{t_m}, (\tau^m)^{-1}), \end{aligned}$$

with Bayesian priors

$$\begin{aligned} \tau^n, \tau^l, \tau^m &\sim \mathcal{G}(\tau^* | \alpha_\tau, \beta_\tau) \\ F_{ik}^t &\sim \mathcal{E}(F_{ik}^t | \lambda_k^t) \quad \text{or} \quad F_{ik}^t \sim \mathcal{N}(F_{ik}^t | 0, (\lambda_k^t)^{-1}) \\ G_{jk}^l &\sim \mathcal{E}(G_{jk}^l | \lambda_k^l) \quad \text{or} \quad G_{jk}^l \sim \mathcal{N}(G_{jk}^l | 0, (\lambda_k^l)^{-1}) \\ S_{kl}^n &\sim \mathcal{E}(S_{kl}^n | \lambda_S^n) \quad \text{or} \quad S_{kl}^n \sim \mathcal{N}(S_{kl}^n | 0, (\lambda_S^n)^{-1}) \\ S_{kl}^m &\sim \mathcal{E}(S_{kl}^m | \lambda_S^m) \quad \text{or} \quad S_{kl}^m \sim \mathcal{N}(S_{kl}^m | 0, (\lambda_S^m)^{-1}). \end{aligned}$$

Automatic Relevance Determination (ARD)

We employ a Bayesian ARD prior, which helps perform automatic model selection. Note the λ_k^t parameters in the prior of F_{ik}^t and G_{jk}^l . This parameter is shared by all entities of type E_t , and hence the entire factor k is either activated (if λ_k^t has a low value) or ‘‘turned off’’ (if λ_k^t has a high value). The ARD works by placing a Gamma prior over each of these variables,

$$\lambda_k^t \sim \mathcal{G}(\lambda_k^t | \alpha_0, \beta_0).$$

Through this construction, factors that are active for only a few entities will be pushed further to zero, turning the factor off. This prior has been used extensively for model selection in Virtanen et al. [2011, 2012] for real-valued matrix factorisation, and Tan and Févotte [2013] for nonnegative matrix factorisation. Instead of having to choose the correct values for the K_t , we can give an upper bound and our model will automatically determine the number of factors to use.

Dataset importance One challenge with multiple matrix factorisation is that it relies on finding common patterns in multiple datasets. If two datasets are very different, the methods may end up finding a solution that fits one dataset much better, resulting in poor predictions for the other one. To address this, we add an importance value for each of the $\mathbf{R}^n, \mathbf{D}^l, \mathbf{C}^m$ datasets, respectively $\alpha_n, \alpha_l, \alpha_m$, to ensure that the method will converge to a solution that better fits datasets with higher importance values. We modify the likelihood

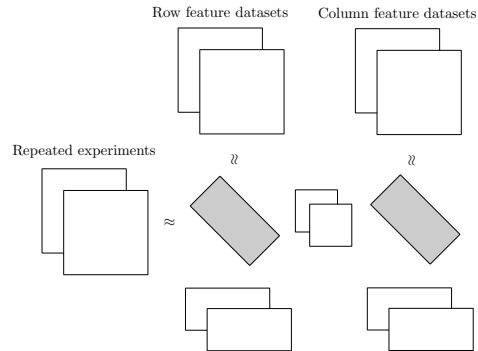


Figure 4: Overview of HMF, combining the multiple matrix tri-factorisation of two repeated experiments with multiple matrix factorisations of row and column feature datasets. Shared factor matrices are grey.

of the model by using these importance values,

$$\begin{aligned} p(\boldsymbol{\theta} | \mathbf{R}, \mathbf{D}, \mathbf{C}) &\propto p(\boldsymbol{\theta}) \times \prod_{n=1}^N p(\mathbf{R}^n | \mathbf{F}^{t_n}, \mathbf{S}^n, \mathbf{F}^{u_n}, \tau^n)^{\alpha^n} \\ &\times \prod_{l=1}^L p(\mathbf{D}^l | \mathbf{F}^{t_l}, \mathbf{G}^l, \tau^l)^{\alpha^l} \times \prod_{m=1}^M p(\mathbf{C}^m | \mathbf{F}^{t_m}, \mathbf{S}^m, \tau^m)^{\alpha^m} \end{aligned}$$

where $\boldsymbol{\theta}$ is the set of model parameters. This technique was used by Remes et al. [2015] to ensure their model fits the binary training labels. This technique can be interpreted as repeating each of the values in the dataset \mathbf{D}^l α^l times, hence forcing the model to fit better to that dataset.

Inference An efficient Gibbs sampling algorithm can be used for inference due to the model’s conjugacy. For details see Supplementary Materials (Section 1).

4 RELATED WORK

The idea of using matrix factorisation and tri-factorisation to integrate multiple datasets can be traced back to the CANDECOMP/PARAFAC (CP) and PARAFAC2 tensor decompositions (Harshman [1970, 1972]). These models are in fact a less general version of multiple matrix tri-factorisation. If we are given multiple datasets for the same two entity types and concatenate them to form a tensor, the CP method will perform multiple matrix tri-factorisation, where the dataset-specific middle matrices \mathbf{S} are restricted to being diagonal.

Multiple matrix factorisation models for integrating datasets between two entity types (such as multiple gene expression profiles), by sharing one of the two factor matrices, can be found amongst others in Zhang et al. [2005] and Lee et al. [2012], with Bayesian models given by Virtanen et al. [2012] and Chatzis [2014].

Some approaches focus on jointly decomposing two datasets spanning three entity types and sharing two latent matrices (Shi et al. [2010]), sometimes using supervised labels for learning (Zhu et al. [2007]). Others do not explicitly share the latent matrices but instead add a penalisation term based on the consensus between the matrices (Seichepine et al. [2013]).

More general matrix factorisation methods are presented by Lippert et al. [2008] and Singh and Gordon [2008], where each entity type has its own latent matrix, with a Bayesian version given in Klami et al. [2014]. However, these approaches cannot integrate multiple datasets between the same two entity types, since all matrices are shared. We would require a third, dataset-specific matrix to solve this problem – which is exactly what matrix tri-factorisation allows us to do. Models for multiple non-negative matrix tri-factorisation are given by Wang et al. [2008] and Žitnik and Zupan [2015], which can also handle constraint matrices, but require all given datasets to be fully observed. As a result, missing values inside each matrix need to be imputed. For binary datasets a missing association can easily be imputed as a zero, but for real-valued datasets this is not a viable option.

Overall, our method is novel in several aspects. Firstly, it is the first general hybrid model between matrix factorisation and tri-factorisation. A non-probabilistic version can be found in Zhu et al. [2007], but this model only combined a single matrix tri-factorisation with a single matrix factorisation. Secondly, our model is a hybrid between nonnegative and real-valued factors. If multiple datasets are jointly decomposed, one can be a nonnegative matrix factorisation, where another can be semi-nonnegative, and another can be real-valued. Finally, through formulating the method as a Bayesian probabilistic model, it can deal with missing values, perform automatic model selection, and is much less prone to overfitting (especially for sparse datasets).

In this paper we are demonstrating our method on two specific biological datasets. However, it can be widely applied to other biological applications such as predicting drug-target interactions (Gönen [2012]) or gene functions (Lippert et al. [2008]), as well as other fields like collaborative filtering (Salakhutdinov and Mnih [2008]) and image analysis (Zhang et al. [2005]).

5 DATASETS

To demonstrate the advantages of our approach for missing values prediction, we consider two different applications. Firstly, integrating four drug sensitivity datasets, where the datasets are similar and hence predictivity of the datasets is high. Here we perform

in-matrix predictions of missing values. Secondly, integrating gene expression and methylation level datasets for breast cancer patients and cancer driver genes, where the datasets are much more dissimilar. We perform out-of-matrix predictions, using the methylation levels of patients to predict gene expression values, and vice versa. We briefly introduce the datasets below; a more thorough description of the datasets can be found in the Supplementary Materials (Section 3).

5.1 Drug Sensitivity Data

We consider four different drug sensitivity datasets, containing 650 unique drugs and 1209 cell lines. Each of these datasets shows the response (sensitivity) of a given cell line (cancer type in a tissue) to a given drug, either measuring the drug concentration needed to inhibit undesired cell line activity by half (IC_{50}), or the drug concentration that achieves half the maximal desired effect on the cell line (EC_{50}).

- Genomics of Drug Sensitivity in Cancer (GDSC v5.0, Yang et al. [2013]). Natural log of IC_{50} values for 139 drugs across 707 cell lines, with 80% observed entries.
- Cancer Therapeutics Response Portal (CTRP v2, Seashore-Ludlow et al. [2015]). EC_{50} values for 545 drugs across 887 cell lines, with 80% observed entries.
- Cancer Cell Line Encyclopedia (CCLE, Barretina et al. [2012]). Both IC_{50} and EC_{50} values for 24 drugs across 504 cell lines, with 96% and 63% observed entries, respectively.

We selected the drugs and cell lines that are present in at least two of the four datasets, and for which we had side information like gene expression profiles available. This resulted in a lot of drugs and cell lines being filtered. For the GDSC dataset we undid the log transform. We rescaled the values per cell line to the range [0,1] in each dataset. We used the cell line features provided by the GDSC dataset (gene expression levels, copy number variations, and mutation information), and for the drugs we extracted 1D and 2D descriptors and structural fingerprints. We obtained primary protein targets from GDSC for 48 of the 52 drugs.

After preprocessing and filtering, the four datasets span 52 unique drugs and 399 cell lines, with 95.1% of the entries having at least one observed value, and 62.9% of the entries having at least two observed values. The information on the four datasets is summarised in Table 1, along with the fraction of over-lapping observed entries.

Table 1: Overview of the four drug sensitivity dataset after preprocessing and filtering.

Dataset	Number cell lines	Number drugs	Fraction observed	Overlap with other datasets			
				GDSC IC_{50}	CTRP EC_{50}	CCLE IC_{50}	CCLE EC_{50}
GDSC IC_{50}	399	48	73.57%	-	52.25%	9.34%	6.00%
CTRP EC_{50}	379	46	86.03%	57.39%	-	11.96%	7.37%
CCLE IC_{50}	253	16	96.42%	44.19%	51.51%	-	55.06%
CCLE EC_{50}	252	16	58.88%	28.52%	31.87%	55.28%	-

5.2 Methylation and Gene Expression Data

Our second application is that of integrating promoter-region methylation (PM) and gene body methylation (GM) datasets with a gene expression (GE) profile for breast cancer patients, coming from the The Cancer Genome Atlas (TCGA, Koboldt et al. [2012]). There are 254 different samples (both healthy and tumor tissues), across 13966 genes. We focus on 160 breast cancer driver genes, from the IntOGen database (Gonzalez-Perez et al. [2013]). We standardise the datasets to have zero mean and unit standard deviation per gene. Note that this dataset is not nonnegative. In our experiments we predict values in one of the three datasets, given the values of the other two.

6 IN-MATRIX PREDICTIONS

We performed 10-fold cross-validation on each of the four drug sensitivity datasets to predict missing values. We tested two variants of our HMF model: multiple matrix tri-factorisation using all four drug sensitivity datasets (HMF D-MTF, \mathbf{R}_n), and multiple matrix factorisation on all four drug sensitivity datasets, sharing the cell line factors (HMF D-MF, \mathbf{D}_l).

We compared our model to several state of the art methods. Since the four datasets are all nonnegative, we can use nonnegative matrix factorisation (NMF) and tri-factorisation (NMTF) models. We compare with non-probabilistic NMF by Lee and Seung [2001] (NP-NMF), Bayesian NMF by Schmidt et al. [2009] (BNMF), non-probabilistic NMTF by Yoo and Choi [2009] (NP-NMTF), Bayesian NMTF (BNMNTF), and Multiple NMF (sharing the cell line factors). We also applied several state-of-the-art machine learning models using the scikit-learn Python package, particularly: Linear Regression (LR), Random Forests (RF, 100 trees), and Support Vector Regression (SVR, *rbf* kernel). These methods were given the drug and cell line features for training. Finally, we used a method called Kernelised Bayesian Matrix Factorisation (KBMF, Gönen and Kaski [2014]), which was used by Ammad-ud din et al. [2014] to predict drug sensitivity values for the GDSC dataset. This method lever-

ages similarity kernels of the drugs and cell lines, which we reconstructed for the feature datasets (Jaccard kernel for binary features, Gaussian for real-valued features after standardising each feature).

We performed nested cross-validation to select the dimensionality K for the matrix factorisation models and KBMF. In contrast, our model simply used $K_t = 10$ for each entity type E_t , and let the ARD choose the correct number of factors. We used nonnegative factors for the entity type factor matrices (\mathbf{F}_t), and real-valued for all other factors. We used K -means and least squares initialisation, and set all importance values to one.

The results for cross-validation are given in Table 2. We see that our HMF models outperform all other methods, giving predictive gains of up to 30%. The multiple matrix tri-factorisation approach (HMF D-MTF) achieves the best performance on three of the datasets, and is a close second on the fourth. We also see that the Bayesian matrix factorisation models outperform both the non-probabilistic approaches, and the state-of-the-art machine learning methods, demonstrating that Bayesian matrix factorisation is a powerful paradigm for in-matrix predictions, with our proposed HMF model giving significant gains in predictive performance.

7 SPARSE DATA PREDICTIONS

A very important use case is when there are few observed entries, leading to a sparse matrix. We measured the performances of in-matrix predictions on sparse matrices, focusing on the GDSC and CTRP drug sensitivity datasets as these are the largest. We vary the fraction of missing values and predict those entries, taking the average of twenty random training-test data splits per fraction. We compared our model’s multiple matrix factorisation and tri-factorisation models (HMF D-MF and HMF D-MTF) with the other matrix factorisation models (NMF, NMTF, BNMF, BNMNTF). For the dimensionality of HMF we use $K_t = 10$ as before, and for the matrix factorisation models we use the most common dimen-

Table 2: Mean squared error (MSE) of 10-fold in-matrix cross-validation results on the drug sensitivity datasets. We also give the relative improvement (% impr.) compared to NMF. The best performances are highlighted in bold.

Method	GDSC IC_{50}		CTRP EC_{50}		CCLE IC_{50}		CCLE EC_{50}	
	MSE	% impr.	MSE	% impr.	MSE	% impr.	MSE	% impr.
NMF	0.0896	-	0.0959	-	0.0746	-	0.1535	-
NMTF	0.0879	1.91%	0.0954	0.44%	0.0747	-0.18%	0.1506	1.91%
Multiple NMF	0.0859	4.10%	0.0928	3.18%	0.0666	10.64%	0.1157	24.66%
BNMF	0.0805	10.20%	0.0919	4.05%	0.0594	20.29%	0.1318	14.19%
BNMTF	0.0799	10.81%	0.0920	4.03%	0.0593	20.52%	0.1292	15.84%
KBMF	0.0819	8.60%	0.0919	4.13%	0.0618	17.13%	0.1303	15.13%
LR	0.0886	1.10%	0.0949	1.00%	0.0719	3.62%	0.1342	12.60%
RF	0.0876	2.21%	0.0989	-3.15%	0.0668	10.47%	0.1219	20.62%
SVR	0.1091	-21.72%	0.1091	-13.80%	0.0916	-22.76%	0.1230	19.92%
HMF D-MF	0.0775	13.54%	0.0919	4.11%	0.0592	20.65%	0.1062	30.81%
HMF D-MTF	0.0768	14.25%	0.0908	5.28%	0.0558	25.17%	0.1073	30.12%

sionality used in the cross-validation from Section 6.¹

Figure 5 shows that the non-probabilistic models start overfitting very quickly as the sparsity levels of two datasets increase, on both the GDSC (5a) and CTRP (5b) datasets. The Bayesian versions perform lot better, but our HMF models consistently outperform all other models, even when only 10% of the values are observed. The multiple matrix tri-factorisation model (HMF D-MTF) performs particularly well.

8 OUT-OF-MATRIX PREDICTIONS

We did three out-of-matrix prediction experiments on the methylation and gene expression data. We performed ten-fold cross-validation, splitting the 254 samples into ten folds. We predicted the gene expression values for new samples, given the gene expression values of the other samples and both of the methylation datasets (PM, GM to GE). We also did this for the other two combinations (GE, GM to PM; GE, PM to GM). Methylation data is known to be correlated with gene expression values (Kundaje et al. [2015]), although this correlation is generally weak. We therefore expected a weak predictive performance, but it is interesting to see which methods perform best.

We used the HMF D-MF and HMF D-MTF models described earlier. We also considered the similarity dataset part of our model (\mathbf{C}_m) by constructing a similarity kernel for the samples using each of the datasets (see Supplementary Materials, Section 3.4). We give the model the dataset we are trying to predict (e.g.

GE), decomposing it using matrix factorisation, and also give it the similarity kernels for the other two (e.g. GM and PM). We call this approach HMF S-MF. We could have also used matrix tri-factorisation, but since the third matrix is not shared this is effectively the same model.

For the HMF D-MF models we used $K_t = 40$, 0.5 as the importance value for the dataset we are trying to predict, and 1.5 for the other two. For HMF D-MTF we used $K_t = 40$, and 0.5 as importance for all three datasets. Finally, for HMF S-MF we used $K_t = 30$, and 1.0 as importance for all three datasets. For all three, we used nonnegative factors for shared matrices (K -means initialisation), and real-valued ones for private matrices (least squares initialisation).

We compared with the LR, RF, and SVR algorithms, giving two datasets as features, and the third as regression values. We used the gene average as a baseline. Since the datasets are real-valued, we cannot compare with any nonnegative matrix factorisation models.

The results for this out-of-matrix cross-validation are given in Table 3. The HMF D-MF model outperforms all state-of-the-art machine learning methods on two of the three datasets, and is only beaten by SVR on the first one. Our model performs especially well on the third case (GE, PM to GM), implying our method works best when the predictivity of values is high (lower MSE). The HMF D-MTF and HMF S-MF methods perform slightly worse, but are still competitive with the other machine learning methods.

Many of the model choices in the experiments (such as model selection, initialisation, factorisation and nega-

¹GDSC: $K = 2$, $(K, L) = (4, 4)$, $K = 4$, $(K, L) = (7, 7)$. CTRP: $K = 2$, $(K, L) = (2, 4)$, $K = 3$, $(K, L) = (3, 3)$.

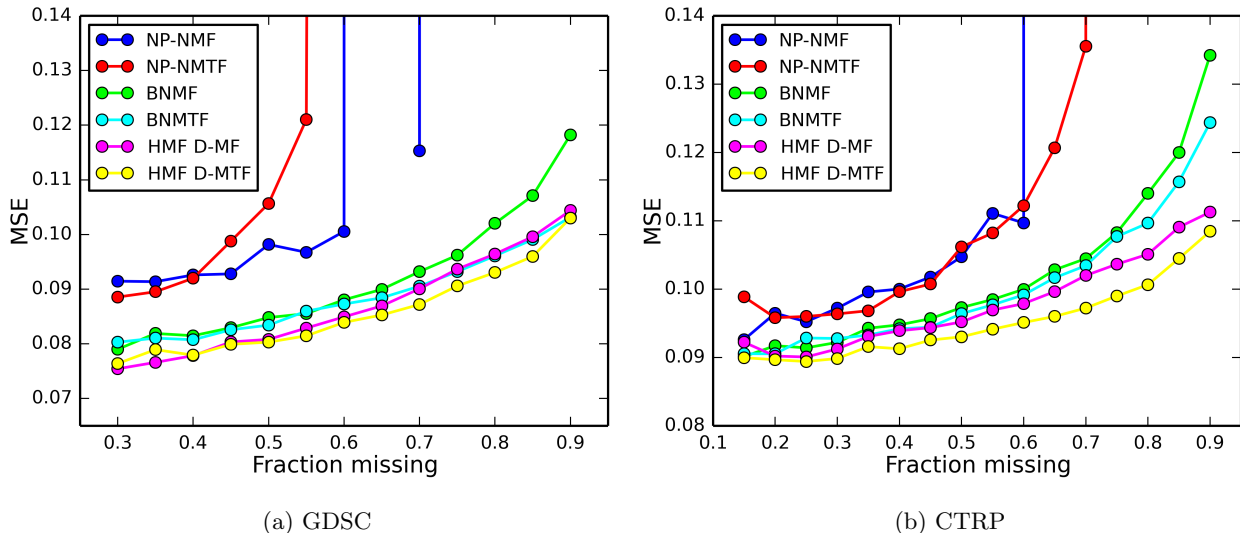


Figure 5: Graphs showing average mean squared error (MSE) and standard deviation of in-matrix predictions on the GDSC (left) and CTRP (right) drug sensitivity datasets. We vary the fraction of missing entries, averaging performance across 20 random splits between train and test data, and compare our HMF models (HMF D-MF, HMF D-MTF) with several matrix factorisation models (NMF, NMTF, BNMF, BNMTF).

Table 3: Mean squared error (MSE) of 10-fold out-of-matrix cross-validation results on the promoter-region methylation (PM), gene body methylation (GM), and gene expression (GE) datasets. We use two datasets as features, and predict values for new samples in the third dataset. The best results are highlighted in bold.

Method	GM, PM to GE	GE, GM to PM	GE, PM to GM
Gene average	1.009	1.008	1.009
LR	2.847	2.036	1.478
RF	0.811	0.799	0.714
SVR	0.767	0.749	0.657
HMF D-MF	0.788	0.735	0.602
HMF D-MTF	0.850	0.798	0.640
HMF S-MF	0.820	0.794	0.672

tivity choices, and importance values) are explored extensively in Section 4 of the Supplementary Materials.

9 CONCLUSION

We have presented a fully Bayesian model for data integration, based on a hybrid of nonnegative, semi-nonnegative, and real-valued matrix factorisation and tri-factorisation models. The general nature of this model allows it to easily integrate many datasets across different entity types, including repeated experiments, similarity matrices, and very sparse datasets.

We demonstrated the model on two different biological

applications. On four drug sensitivity datasets we obtained significant in-matrix prediction improvements compared to state-of-the-art matrix factorisation and machine learning methods. Our data fusion approach based on multiple matrix tri-factorisation (HMF D-MTF) is particularly powerful, achieving the best performance on three of the four datasets. We also show that our proposed model can provide consistently better predictions on very sparse datasets, outperforming all other matrix factorisation models. Finally, we integrated methylation and gene expression data in an out-of-matrix prediction setting, and here the approach based on multiple matrix factorisation (HMF D-MF) proved to be very powerful, beating all state-of-the-art machine learning methods on two of the three datasets. The approaches using multiple matrix tri-factorisation and similarity datasets are also promising.

We showcased our model on different biological datasets, but we believe that this is a powerful and general framework that can also be applied to many other fields.

Acknowledgements

This work was supported by the UK Engineering and Physical Sciences Research Council (EPSRC), grant reference EP/M506485/1; and Methods for Integrated analysis of Multiple Omics datasets (MI-MOmics, 305280).

References

- M. Ammad-ud din, E. Georgii, M. Gönen, T. Laitinen, O. Kallioniemi, K. Wennerberg, A. Poso, and S. Kaski. Integrative and personalized QSAR analysis in cancer by kernelized Bayesian matrix factorization. *Journal of chemical information and modeling*, 54(8):2347–59, Aug. 2014.
- J. Barretina, G. Caponigro, N. Stransky, K. Venkatesan, A. A. Margolin, S. Kim, C. J. Wilson, et al. The Cancer Cell Line Encyclopedia enables predictive modelling of anticancer drug sensitivity. *Nature*, 483(7391):603–7, Mar. 2012.
- J. P. Brunet, T. R. Golub, P. Tamayo, and J. P. Mesirov. Metagenes and molecular pattern discovery using matrix factorization. *Proceedings of the National Academy of Sciences*, 101(12):4164–4169, Mar. 2004.
- S. P. Chatzis. Dynamic Bayesian Probabilistic Matrix Factorization. In *Proceedings of the 28th AAAI Conference on Artificial Intelligence*, pages 1731–1737, 2014.
- C. Ding, T. Li, W. Peng, and H. Park. Orthogonal nonnegative matrix t-factorizations for clustering. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, page 126, New York, New York, USA, Aug. 2006. ACM Press.
- M. Gönen. Predicting drug-target interactions from chemical and genomic kernels using Bayesian matrix factorization. *Bioinformatics*, 28(18):2304–10, Sept. 2012.
- M. Gönen and S. Kaski. Kernelized Bayesian Matrix Factorization. *IEEE transactions on pattern analysis and machine intelligence*, 36(10):2047–60, Oct. 2014.
- A. Gonzalez-Perez, C. Perez-Llamas, J. Deu-Pons, D. Tamborero, M. P. Schroeder, A. Jene-Sanz, A. Santos, and N. Lopez-Bigas. IntOGen-mutations identifies cancer drivers across tumor types. *Nature Methods*, 10(11):1081–1082, Sept. 2013. ISSN 1548-7091. doi: 10.1038/nmeth.2642.
- R. A. Harshman. Foundations of the PARAFAC procedure: Models and conditions for an explanatory multimodal factor analysis. *UCLA Working Papers in Phonetics*, 16(10):1–84, 1970.
- R. A. Harshman. PARAFAC2: Mathematical and technical notes. *UCLA working papers in phonetics*, 22(10):30–44, 1972.
- A. Klami, G. Bouchard, and A. Tripathi. Group-sparse Embeddings in Collective Matrix Factorization. In *Proceedings of the 2nd International Conference on Learning Representations*, Jan. 2014.
- D. C. Koboldt, R. S. Fulton, M. D. McLellan, H. Schmidt, J. Kalicki-Veizer, J. F. McMichael, L. L. Fulton, et al. Comprehensive molecular portraits of human breast tumours. *Nature*, 490(7418):61–70, Sept. 2012.
- A. Kundaje, W. Meuleman, J. Ernst, M. Bilenky, A. Yen, A. Heravi-Moussavi, P. Kheradpour, et al. Integrative analysis of 111 reference human epigenomes. *Nature*, 518(7539):317–330, Feb. 2015.
- C. M. Lee, M. a. V. Mudaliar, D. R. Haggart, C. R. Wolf, G. Miele, J. K. Vass, D. J. Higham, and D. Crowther. Simultaneous non-negative matrix factorization for multiple large scale gene expression datasets in toxicology. *PloS one*, 7(12):e48238, Jan. 2012.
- D. Lee and H. Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–91, Oct. 1999.
- D. Lee and H. Seung. Algorithms for non-negative matrix factorization. In *Advances in Neural Information Processing Systems (NIPS)*, number 1, pages 556–562, 2001.
- C. Lippert, S. Weber, and Y. Huang. Relation prediction in multi-relational domains using matrix factorization. In *NIPS Workshop on Structured Input, Structured Output*, 2008.
- S. Remes, T. Mononen, and S. Kaski. Classification of weak multi-view signals by sharing factors in a mixture of Bayesian group factor analyzers. *NIPS Workshop on Machine Learning and Interpretation in Neuroimaging (MLINI)*, Dec. 2015.
- R. Salakhutdinov and A. Mnih. Probabilistic Matrix Factorization. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1257–1264, 2008.
- M. N. Schmidt, O. Winther, and L. K. Hansen. Bayesian non-negative matrix factorization. *Independent Component Analysis and Signal Separation*, pages 540–547, 2009.
- B. Seashore-Ludlow, M. G. Rees, J. H. Cheah, M. Cokol, E. V. Price, M. E. Coletti, V. Jones, et al. Harnessing Connectivity in a Large-Scale Small-Molecule Sensitivity Dataset. *Cancer discovery*, 5(11):1210–23, Nov. 2015.
- N. Seichepine, S. Essid, C. Févotte, and O. Cappé. Soft nonnegative matrix co-factorization with application to multimodal speaker diarization. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 3537–3541. IEEE, May 2013.
- Y. Shi, M. Larson, and A. Hanjalic. Mining mood-specific movie similarity with matrix factorization for context-aware recommendation. In *Proceedings*

- of the *Workshop on Context-Aware Movie Recommendation (CAMRa)*, pages 34–40, New York, New York, USA, Sept. 2010. ACM Press.
- A. P. Singh and G. J. Gordon. Relational learning via collective matrix factorization. In *Proceeding of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, page 650, New York, New York, USA, Aug. 2008. ACM Press.
- V. Y. F. Tan and C. Févotte. Automatic relevance determination in nonnegative matrix factorization with the (β) -divergence. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(7):1592–1605, July 2013.
- S. Virtanen, A. Klami, and S. Kaski. Bayesian CCA via Group Sparsity. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 457–464, 2011.
- S. Virtanen, A. Klami, S. Khan, and S. Kaski. Bayesian group factor analysis. In *Proceedings of the 15th International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2012.
- M. Žitnik and B. Zupan. Data Fusion by Matrix Factorization. *IEEE transactions on pattern analysis and machine intelligence*, 37(1):41–53, Jan. 2015.
- F. Wang, T. Li, and C. Zhang. Semi-supervised clustering via matrix factorization. In *Proceedings of the 2008 SIAM International Conference on Data Mining (SDM)*, pages 1–12, 2008.
- W. Yang, J. Soares, P. Greninger, E. J. Edelman, H. Lightfoot, S. Forbes, N. Bindal, et al. Genomics of Drug Sensitivity in Cancer (GDSC): a resource for therapeutic biomarker discovery in cancer cells. *Nucleic Acids Research*, 41(Database issue):D955–61, Jan. 2013.
- J. Yoo and S. Choi. Probabilistic matrix trifactorization. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, number 3, pages 1553–1556. IEEE, Apr. 2009.
- D. Q. Zhang, S. C. Chen, and Z. H. Zhou. Two-dimensional non-negative matrix factorization for face representation and recognition. In *Analysis and Modelling of Faces and Gestures*, volume 3723, pages 350–363, 2005.
- S. Zhu, K. Yu, Y. Chi, and Y. Gong. Combining content and link for classification using matrix factorization. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, page 487, New York, New York, USA, July 2007. ACM Press.