Data Fusion by Matrix Factorization

Marinka Žitnik and Blaž Zupan

Abstract—For most problems in science and engineering we can obtain data sets that describe the observed system from various perspectives and record the behavior of its individual components. Heterogeneous data sets can be collectively mined by data fusion. Fusion can focus on a specific target relation and exploit directly associated data together with contextual data and data about system's constraints. In the paper we describe a data fusion approach with penalized matrix tri-factorization (DFMF) that simultaneously factorizes data matrices to reveal hidden associations. The approach can directly consider any data that can be expressed in a matrix, including those from feature-based representations, ontologies, associations and networks. We demonstrate the utility of DFMF for gene function prediction task with eleven different data sources and for prediction of pharmacologic actions by fusing six data sources. Our data fusion algorithm compares favorably to alternative data integration approaches and achieves higher accuracy than can be obtained from any single data source alone.

Index Terms—data fusion, intermediate data integration, matrix factorization, data mining, bioinformatics, cheminformatics

1 Introduction

DATA abound in all areas of human endeavour. We may gather various data sets that are directly related to the problem, or data sets that are loosely related to our study but could be useful when combined with other data sets. Consider, for example, the exposome [1] that encompasses the totality of human endeavour in the study of disease. Let us say that we examine susceptibility to a particular disease and have access to the patients' clinical data together with data on their demographics, habits, living environments, friends, relatives, movie-watching habits, and movie genre ontology. Mining such a diverse data collection may reveal interesting patterns that would remain hidden if we would analyze only directly related, clinical data. What if the disease was less common in living areas with more open spaces or in environments where people need to walk instead of drive to the nearest grocery? Is the disease less common among those that watch comedies and ignore politics and news?

Methods for data fusion can collectively treat data sets and combine diverse data sources even when they differ in their conceptual, contextual and typographical representation [2], [3]. Individual data sets may be incomplete, yet because of their diversity and complementarity, fusion can improve the robustness and predictive performance of the resulting models [4], [5].

According to Pavlidis *et al.* (2002) [6], data fusion approaches can be classified into three main categories depending on the modeling stage at which fusion takes place. *Early (or full) integration* transforms all data sources into a single feature-based table and treats this as a single data set that can be explored by any of the well-established feature-based machine learning algorithms. The inferred models can in principle include any type of relationships between the features from within and between the data sources. Early integration relies on procedures for feature construction. For our exposome example, patient-specific data would need to include both clinical data and information from the movie genre ontologies. The former may be trivial as this data is already related to each specific patient, while the latter requires more complex feature engineering. Early integration also neglects the modular structure of the data.

In *late (decision) integration*, each data source gives rise to a separate model. Predictions of these models are fused by model weighting. Again, prior to model inference, it is necessary to transform each data set to encode relations to the target concept. For our example, information on the movie preferences of friends and relatives would need to be mapped to disease associations. Such transformations may not be trivial and would need to be crafted independently for every data source.

The youngest branch of data fusion algorithms is *intermediate (partial) integration*. Algorithms in this category explicitly address the multiplicity of data and fuse them through inference of a single joint model. Intermediate integration does not merge the input data, nor does it develop separate models for each data source. It instead retains the structure of the data sources by incorporating it within the structure of predictive model. This particular approach is often preferred because of its superior predictive accuracy [6], [5], [7], [8], [9], but for a given model

- *M. Žitnik is with Faculty of Computer and Information Science, University of Ljubljana, Tržaška 25, SI-1000 Ljubljana, Slovenia.*
- *B. Zupan is with Faculty of Computer and Information Science, University of Ljubljana, Tržaška 25, SI-1000 Ljubljana, Slovenia, and with Department of Molecular and Human Genetics, Baylor College of Medicine, Houston, TX-77030, USA.*
E-mail: blaz.zupan@fri.uni-lj.si

type, it requires the development of a new inference algorithm.

We here report on the development of a new method for intermediate data fusion based on constrained matrix factorization. Our aim was to construct an algorithm that requires no or only minimal transformation of input data and can fuse feature-based representations, ontologies, associations and networks. We focus on the challenge of dealing with collections of heterogeneous data sources, and while showing that our method can be used on sizable problems from current research, scaling is not the focus of the present paper. We first present our data fusion algorithm, henceforth DFMF (Sec. 2), and then place it within the related work of relational learning approaches (Sec. 3). We also refer to related data integration approaches, specifically to methods of kernel-based data fusion (Sec. 3). We then examine the utility of DFMF and experimentally compare it with intermediate integration by multiple kernel learning, early integration with random forests, and tri-SPMF [10], previously proposed matrix tri-factorization approach (Sec 4).

2 Data Fusion Algorithm

The DFMF considers r object types $\mathcal{E}_1, \ldots, \mathcal{E}_r$ and a collection of data sources, each relating a pair of object types $(\mathcal{E}_i, \mathcal{E}_j)$. In our introductory example of the exposome, object types could be a patient, a disease or a living environment, among others. If there are n_i objects of type $\mathcal{E}_i$ (o^i_p is p-th object of type $\mathcal{E}_i$) and n_j objects of type $\mathcal{E}_j$, we represent the observations from the data source that relates $(\mathcal{E}_i, \mathcal{E}_j)$ for $i \neq j$ in a sparse matrix $\mathbf{R}_{ij} \in \mathbb{R}^{n_i \times n_j}$. An example of such a matrix would relate patients and drugs by reporting on patient's current drug prescriptions. Notice that matrices $\mathbf{R}_{ij}$ and $\mathbf{R}_{ji}$ are in general asymmetric. A data source that provides relations between objects of the same type $\mathcal{E}_i$ is represented by a constraint matrix $\mathbf{\Theta}_i \in \mathbb{R}^{n_i \times n_i}$. Examples of such constraints are social networks and drug interactions.

In real-world scenarios we might not have access to relations between all pairs of object types. Our data fusion algorithm still integrates all available data if the underlying graph of relations between object types is connected. In that case, low-dimensional representations of objects of certain type borrow information from related objects of the different type. Fig. 1 shows an example of an underlying graph of relations and a block configuration of the fusion system with four object types.

To retain the block structure of our fusion system and hence model distinct relations between object types, we propose the simultaneous factorization of all relation matrices $\mathbf{R}_{ij}$ constrained by $\mathbf{\Theta}_i$. The resulting system contains factors that are specific to each data source and factors that are specific to each object type. Through factor sharing we fuse the data but also identify source-specific patterns.

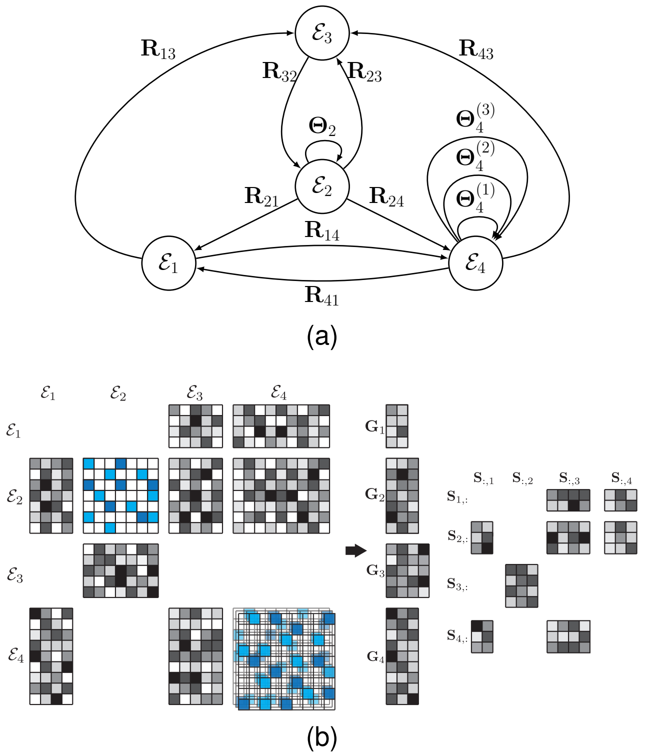


Fig. 1. Conceptual fusion configuration for four object types, $\mathcal{E}_1, \mathcal{E}_2, \mathcal{E}_3$ and $\mathcal{E}_4$, equivalently represented by the graph of relations between object types (a) and the block-based matrix structure (b). Every data source relates a pair of object types, denoted by arcs in a graph (a) or matrices with shades of gray in block matrix (b). For example, data matrix $\mathbf{R}_{23}$ relates object types $\mathcal{E}_2$ and $\mathcal{E}_3$. Some relations are entirely missing. For instance, there is no data source relating objects from $\mathcal{E}_3$ and $\mathcal{E}_1$, as there is no arc linking nodes $\mathcal{E}_3$ and $\mathcal{E}_1$ in (a) or equivalently, a matrix $\mathbf{R}_{31}$ is missing in (b). Relations can be asymmetric, such that $\mathbf{R}_{23} \neq \mathbf{R}_{32}^T$. Constraints denoted by loops in (a) or matrices with blue entries in (b) relate objects of the same type. In our example configuration, constraints are provided for object types $\mathcal{E}_2$ (one constraint matrix) and $\mathcal{E}_4$ (three constraint matrices).

We have developed a variant of three-factor penalized matrix factorization that simultaneously decomposes all available relation matrices $\mathbf{R}_{ij}$ into $\mathbf{G}_i \in \mathbb{R}^{n_i \times k_i}$, $\mathbf{G}_j \in \mathbb{R}^{n_j \times k_j}$ and $\mathbf{S} \in \mathbb{R}^{k_i \times k_j}$, and regularizes their approximation through constraint matrices $\mathbf{\Theta}_i$ and $\mathbf{\Theta}_j$ such that $\mathbf{R}_{ij} \approx \mathbf{G}_i \mathbf{S}_{ij} \mathbf{G}_j^T$. Approximation can be rewritten such that entry $\mathbf{R}_{ij}(p,q)$ is approximated by an inner product of the p-th row of matrix $\mathbf{G}_i$ and a linear combination of the columns of matrix $\mathbf{S}_{ij}$, weighted by the q-th column of $\mathbf{G}_j$. The matrix $\mathbf{S}_{ij}$, which has relatively few vectors compared to $\mathbf{R}_{ij}$ ($k_i \ll n_i$, $k_j \ll n_j$), is used to represent many data vectors, and a good approximation can only be achieved in the presence of the latent structure in the original data.

The proposed fusion approach is different from treating an entire system (e.g., from Fig. 1) as a large single matrix. Factorization of such a matrix would yield factors that are not object type-specific and would thus disregard the structure of the system. We also show (Sec. 5.5) that such an approach is inferior in terms of predictive performance.

In comparison with existing multi-type relational data factorization approaches (see Sec. 3) the following characterizes our DFMF data fusion method:

i DFMF can model *multiple* relations between *multiple* object types.
ii Relations between some object types can be completely missing (see Fig. 1).
iii Every object type can be associated with multiple constraint matrices.
iv The algorithm makes no assumptions about structural properties of relations (*e.g.* symmetry of relations).

In order to be applicable to general real-world fusion problems, data fusion algorithm would need to jointly address all of these characteristics. Besides DFMF proposed in this manuscript, we are not aware of any other approach that would do so. Most real-world data integration problems would usually consider a larger number of object types, but with growing number of object types, it is likely that data relating a pair of object types is either not available nor meaningful. On the other side, there may be various data sources available on interactions between objects of the same type that also require appropriate treatment. For example of this type of data, consider abundance of data bases on drug or disease interactions.

In the case study presented in this paper we apply data fusion to infer relations between two target object types, $\mathcal{E}_i$ and $\mathcal{E}_j$ (Sec. 2.6 and Sec. 2.7). This relation, encoded in a target matrix $\mathbf{R}_{ij}$, will be observed in the context of all other data sources (Sec. 2.1). We assume that our target $\mathbf{R}_{ij}$ is a $[0,1]$-matrix that is only partially observed. Its entries indicate a degree of relation, 0 denoting no relation and 1 denoting the strongest relation. We aim to predict unobserved entries in $\mathbf{R}_{ij}$ by reconstructing them through matrix factorization. Such treatment in general applies to multi-class or multi-label classification tasks, which are conveniently addressed by multiple kernel fusion [11], with which we compare our performance in this paper.

In the following, we present the factorization model, objective function, derive the updating rules for optimization, and describe the procedure for prediction of relations from matrix factors. In the optimization part, we closely follow [10] in notation, mathematical derivation and proof technique.

2.1 Factorization Model for Multi-Relational and Multi-Object Type Data

An input to DFMF is a relation block matrix $\mathbf{R}$ that conceptually represents all relation matrices:

$$\mathbf{R} = \begin{bmatrix} * & \mathbf{R}_{12} & \cdots & \mathbf{R}_{1r} \\ \mathbf{R}_{21} & * & \cdots & \mathbf{R}_{2r} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{R}_{r1} & \mathbf{R}_{r2} & \cdots & * \end{bmatrix}. \tag{1}$$

Here, an asterisk ("*") denotes the relation between the same type of objects that DMFM does not model. Notice that our method does not require the presence of all relation matrices in Eq. (1). Depending on a particular data setup, any subset of relation matrices might be missing and thus, unmodeled. A block in the i-th row and j-th column ($\mathbf{R}_{ij}$) of matrix $\mathbf{R}$ represents the relationship between object type $\mathcal{E}_i$ and $\mathcal{E}_j$. The p-th object of type $\mathcal{E}_i$ (i.e. o_p^i) and q-th object of type $\mathcal{E}_j$ (i.e. o_q^j) are related by $\mathbf{R}_{ij}(p,q)$. An important aspect of Eq. (1) for data fusion and what distinguishes DMFM from other conceptually related matrix factorization models such as S-NMTF [12] or even tri-SPMF [10] is that it is designed for multi-object type and multi-relational data where the relations can be asymmetric, $\mathbf{R}_{ji} \neq \mathbf{R}_{ij}^T$, and some can be completely missing (unknown $\mathbf{R}_{ij}$) (Sec. 2.3).

We additionally consider constraints relating objects of the same type. Several data sources of this kind may be available for each object type. For instance, personal relations may be observed from a social network or a family tree. Assume there are $t_i \geq 0$ data sources for object type $\mathcal{E}_i$ represented by a set of constraint matrices $\mathbf{\Theta}_i^{(t)}$ for $t \in \{1, 2, \ldots, t_i\}$. Constraints are collectively encoded in a set of constraint block diagonal matrices $\mathbf{\Theta}^{(t)}$ for $t \in \{1, 2, \ldots, \max_i t_i\}$:

$$\mathbf{\Theta}^{(t)} = \mathrm{Diag}(\mathbf{\Theta}_1^{(t)}, \mathbf{\Theta}_2^{(t)}, \ldots, \mathbf{\Theta}_r^{(t)}) \tag{2}$$

The i-th block along the main diagonal of $\mathbf{\Theta}^{(t)}$ is zero if $t > t_i$. Entries in constraint matrices are positive for objects that are not similar and negative for objects that are similar. The former are known as *cannot-link constraints* because they impose penalties on the current approximation of the matrix factors, and the latter are *must-link constraints*, which are rewards that reduce the value of the cost function during optimization. Must-link constraint expresses the notion that a pair of objects of the same type should be close in their latent component space. An example of must-link constraints are, for instance, drug-drug interactions, and example of cannot-link constraints the matrix of adversaries. Typically, data sources with must-link constraints are more abundant.

The block matrix $\mathbf{R}$ is tri-factorized into block matrix factors $\mathbf{G}$ and $\mathbf{S}$:

$$\mathbf{G} = \mathrm{Diag}(\mathbf{G}_1^{n_1 \times k_1}, \mathbf{G}_2^{n_2 \times k_2}, \ldots, \mathbf{G}_r^{n_r \times k_r}),$$

$$\mathbf{S} = \begin{bmatrix} * & \mathbf{S}_{12}^{k_1 \times k_2} & \cdots & \mathbf{S}_{1r}^{k_1 \times k_r} \\ \mathbf{S}_{21}^{k_2 \times k_1} & * & \cdots & \mathbf{S}_{2r}^{k_2 \times k_r} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{S}_{r1}^{k_r \times k_1} & \mathbf{S}_{r2}^{k_r \times k_2} & \cdots & * \end{bmatrix}. \quad (3)$$

Matrix $\mathbf{S}$ in Eq. (3) has the same block structure as $\mathbf{R}$ in Eq. (1). It is in general asymmetric (*i.e.* $\mathbf{S}_{ji} \neq \mathbf{S}_{ij}^T$) and if a relation matrix is missing in $\mathbf{R}$ then also its corresponding matrix factor in $\mathbf{S}$ will be missing. These two properties of $\mathbf{S}$ stem from our decision to model relation matrices without assuming their structural properties or their availability for every possible combination of object types.

A factorization rank k_i is assigned to $\mathcal{E}_i$ during inference of the factorized system. Factor $\mathbf{S}_{ij}$ defines the latent relation between object types $\mathcal{E}_i$ and $\mathcal{E}_j$, while factor $\mathbf{G}_i$ is specific to objects of type $\mathcal{E}_i$ and is used in the reconstruction of every relation with this object type. In this way, each relation matrix $\mathbf{R}_{ij}$ obtains its own factorization $\mathbf{G}_i\mathbf{S}_{ij}\mathbf{G}_j^T$ with factor $\mathbf{G}_i$ ($\mathbf{G}_j$) that is shared across all relations which involve object types $\mathcal{E}_i$ ($\mathcal{E}_j$). This can also be observed from the block structure of the reconstructed system $\mathbf{G}\mathbf{S}\mathbf{G}^T$:

$$\begin{bmatrix} * & \mathbf{G}_1\mathbf{S}_{12}\mathbf{G}_2^T & \cdots & \mathbf{G}_1\mathbf{S}_{1r}\mathbf{G}_r^T \\ \mathbf{G}_2\mathbf{S}_{21}\mathbf{G}_1^T & * & \cdots & \mathbf{G}_2\mathbf{S}_{2r}\mathbf{G}_r^T \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{G}_r\mathbf{S}_{r1}\mathbf{G}_1^T & \mathbf{G}_r\mathbf{S}_{r2}\mathbf{G}_2^T & \cdots & * \end{bmatrix}. \quad (4)$$

Here, the p-th row in factor $\mathbf{G}_i$ holds the latent component representation of object o_p^i. By holding $\mathbf{G}_j$ and $\mathbf{S}_{ij}$ fixed, it is clear that latent component representation of o_p^i depends on $\mathbf{G}_j$ as well as on the existence of relation $\mathbf{R}_{ij}$. Consequently, all direct and indirect relations have a determining influence on the calculation of o_p^i-th latent representation. Just as the objects of type $\mathcal{E}_i$ are represented by $\mathbf{G}_i$, each relation is represented by factor $\mathbf{S}_{ij}$, which models how the latent components interact in the respective relation. The asymmetry of $\mathbf{S}_{ij}$ takes into account whether a latent component occurs as a subject or an object of corresponding relation $\mathbf{R}_{ij}$.

2.2 Objective Function

The objective function minimized by DFMF aims at good approximation of the input data and adherence to must-link and cannot-link constraints:

$$\begin{aligned} \min_{\mathbf{G} \geq 0} J(\mathbf{G}; \mathbf{S}) \quad &= \sum_{\mathbf{R}_{ij} \in \mathcal{R}} ||\mathbf{R}_{ij} - \mathbf{G}_i\mathbf{S}_{ij}\mathbf{G}_j^T||^2 + \\ &+ \sum_{t=1}^{\max_i t_i} \mathrm{tr}(\mathbf{G}^T\mathbf{\Theta}^{(t)}\mathbf{G}), \end{aligned} \quad (5)$$

Here, $||\cdot||$ and $\mathrm{tr}(\cdot)$ denote the Frobenius norm and trace, respectively, and $\mathcal{R}$ is the set of all relations included in our model. Our objective function explicitly allows that relations between some object types are entirely missing.

Notice that in Eq. (5) we do not approximate input data by $||\mathbf{R} - \mathbf{G}\mathbf{S}\mathbf{G}^T||^2$ as was proposed in related approaches of S-NMTF [12] and tri-SPMF [10]. To model the data system such as that from Fig. 1, one could be tempted to replace the missing relation matrices with zero matrices. This would enable the optimization to further reduce the value of objective function, but would also introduce relations in factorized system that were intentionally not present in the input data. Their inclusion in the model would distort inferred relations between other object types (see Sec. 5.1).

2.3 Computing the Factorization

The DFMF algorithm for solving the minimization problem specified in Eq. (5) is shown in Fig. 2. The algorithm first initializes matrix factors (Sec. 2.8) and then iteratively refines them by alternating between fixing $\mathbf{G}$ and updating $\mathbf{S}$, and then fixing $\mathbf{S}$ and updating $\mathbf{G}$, until convergence. Successive updates of $\mathbf{G}_i$ and $\mathbf{S}_{ij}$ converge to a local minimum of the optimization problem.

We derive multiplicative updating rules for regularized decomposition of relation matrices by fixing one matrix factor (e.g., $\mathbf{G}$) and considering the roots of the partial derivative with respect to the other matrix factor (e.g., $\mathbf{S}$, and vice-versa) of the Lagrangian function. The latter is constructed from the objective function (Eq. 5):

$$\begin{aligned} J(\mathbf{G}; \mathbf{S}) \quad &= \sum_{\mathbf{R}_{ij} \in \mathcal{R}} \mathrm{tr}(\mathbf{R}_{ij}^T\mathbf{R}_{ij} - 2\mathbf{G}_j^T\mathbf{R}_{ij}^T\mathbf{G}_i\mathbf{S}_{ij} + \\ &+ \mathbf{G}_i^T\mathbf{G}_i\mathbf{S}_{ij}\mathbf{G}_j^T\mathbf{G}_j\mathbf{S}_{ij}^T) + \\ &+ \sum_{t=1}^{\max_i t_i} \sum_{i=1}^{r} \mathrm{tr}(\mathbf{G}_i^T\mathbf{\Theta}_i^{(t)}\mathbf{G}_i). \end{aligned} \quad (6)$$

Regarding the correctness and convergence of the algorithm in Fig. 2 we have the following two theorems.

Theorem 1 (Correctness of DFMF algorithm): If the update rules for matrix factors $\mathbf{G}$ and $\mathbf{S}$ from Fig. 2 converge, then the final solution satisfies the KKT conditions of optimality.

Proof: We introduce the Lagrangian multipliers $\lambda_1, \lambda_2, \ldots, \lambda_r$ and construct the Lagrange function:

$$L = J(\mathbf{G}; \mathbf{S}) - \sum_{i=1}^{r} \mathrm{tr}(\lambda_i\mathbf{G}_i^T). \quad (7)$$

Then for i, j, such that $\mathbf{R}_{ij} \in \mathcal{R}$:

$$\frac{\partial L}{\partial \mathbf{S}_{ij}} \quad = \quad -2\mathbf{G}_i^T\mathbf{R}_{ij}\mathbf{G}_j + 2\mathbf{G}_i\mathbf{G}_i^T\mathbf{S}_{ij}\mathbf{G}_j^T\mathbf{G}_j,$$

and for $i = 1, 2, \ldots, r$:

$$\begin{aligned}\frac{\partial L}{\partial \mathbf{G}_i} \quad = \quad & \sum_{j:\mathbf{R}_{ij}\in\mathcal{R}} (-2\mathbf{R}_{ij}\mathbf{G}_j\mathbf{S}_{ij}^T + 2\mathbf{G}_i\mathbf{S}_{ij}\mathbf{G}_j^T\mathbf{G}_j\mathbf{S}_{ij}^T) + \\ & + \sum_{j:\mathbf{R}_{ji}\in\mathcal{R}} (-2\mathbf{R}_{ji}^T\mathbf{G}_j\mathbf{S}_{ji} + 2\mathbf{G}_i\mathbf{S}_{ji}^T\mathbf{G}_j^T\mathbf{G}_j\mathbf{S}_{ji}) + \\ & + \sum_{t=1}^{\max_i t_i} 2\boldsymbol{\Theta}_i^{(t)}\mathbf{G}_i - \lambda_i. \end{aligned} \tag{8}$$

Fixing $\mathbf{G}_1, \mathbf{G}_2, \ldots, \mathbf{G}_r$ and letting $\frac{\partial L}{\partial \mathbf{S}_{ij}} = 0$ for all $i, j = 1, 2, \ldots, r$, we obtain:

$$\mathbf{S} = (\mathbf{G}^T\mathbf{G})^{-1}\mathbf{G}^T\mathbf{R}\mathbf{G}(\mathbf{G}^T\mathbf{G})^{-1}.$$

We then fix $\mathbf{S}$ and let $\frac{\partial L}{\partial \mathbf{G}_i} = 0$ for $i = 1, 2, \ldots, r$. We get an expression for the KKT multiplier λ_i from Eq. (8). Then the KKT complementary condition for the nonnegativity of $\mathbf{G}_i$ is:

$$\begin{aligned}\mathbf{0} \quad = \quad & \lambda_i \circ \mathbf{G}_i = \\ = \quad & \Bigg[\sum_{j:\mathbf{R}_{ij}\in\mathcal{R}} (-2\mathbf{R}_{ij}\mathbf{G}_j\mathbf{S}_{ij}^T + 2\mathbf{G}_i\mathbf{S}_{ij}\mathbf{G}_j^T\mathbf{G}_j\mathbf{S}_{ij}^T) + \\ & + \sum_{j:\mathbf{R}_{ji}\in\mathcal{R}} (-2\mathbf{R}_{ji}^T\mathbf{G}_j\mathbf{S}_{ji} + 2\mathbf{G}_i\mathbf{S}_{ji}^T\mathbf{G}_j^T\mathbf{G}_j\mathbf{S}_{ji}) + \\ & + \sum_{t=1}^{\max_i t_i} 2\boldsymbol{\Theta}_i^{(t)}\mathbf{G}_i \Bigg] \circ \mathbf{G}_i. \end{aligned} \tag{9}$$

Let us here introduce variables $\boldsymbol{\Gamma}_i$ to denote $\boldsymbol{\Gamma}_i = \lambda_i \circ \mathbf{G}_i$. Eq. (9) is a fixed point equation and the solution must satisfy it at convergence. We let:

$$\begin{aligned}\boldsymbol{\Theta}_i^{(t)} &= [\boldsymbol{\Theta}_i^{(t)}]^+ - [\boldsymbol{\Theta}_i^{(t)}]^- \\ \mathbf{R}_{ij}\mathbf{G}_j\mathbf{S}_{ij}^T &= (\mathbf{R}_{ij}\mathbf{G}_j\mathbf{S}_{ij}^T)^+ - (\mathbf{R}_{ij}\mathbf{G}_j\mathbf{S}_{ij}^T)^- \\ \mathbf{S}_{ij}\mathbf{G}_j^T\mathbf{G}_j\mathbf{S}_{ij}^T &= (\mathbf{S}_{ij}\mathbf{G}_j^T\mathbf{G}_j\mathbf{S}_{ij}^T)^+ - (\mathbf{S}_{ij}\mathbf{G}_j^T\mathbf{G}_j\mathbf{S}_{ij}^T)^- \\ \mathbf{R}_{ji}^T\mathbf{G}_j\mathbf{S}_{ji} &= (\mathbf{R}_{ji}^T\mathbf{G}_j\mathbf{S}_{ji})^+ - (\mathbf{R}_{ji}^T\mathbf{G}_j\mathbf{S}_{ji})^- \\ \mathbf{S}_{ji}^T\mathbf{G}_j^T\mathbf{G}_j\mathbf{S}_{ji} &= (\mathbf{S}_{ji}^T\mathbf{G}_j^T\mathbf{G}_j\mathbf{S}_{ji})^+ - (\mathbf{S}_{ji}^T\mathbf{G}_j^T\mathbf{G}_j\mathbf{S}_{ji})^- \end{aligned}$$

where all matrices on right-hand sides are nonnegative. Then, given an initial guess of $\mathbf{G}_i$, the successive updates of $\mathbf{G}_i$ using Eq. (10)–(12) converge to a local minimum of the problem in Eq. (5). It can be easily seen that using such a rule, at convergence, $\mathbf{G}_i$ satisfies $\boldsymbol{\Gamma}_i \circ \mathbf{G}_i = \mathbf{0}$, which is equivalent to $\boldsymbol{\Gamma}_i = 0$ (Eq. (9)) due to nonnegativity of $\mathbf{G}_i$. □

Theorem 2 (Convergence of DFMF algorithm): The objective function $J(\mathbf{G}; \mathbf{S})$ given by Eq. (5) is nonincreasing under the updating rules for matrix factors $\mathbf{G}$ and $\mathbf{S}$ in Fig. 2.

Please see the Appendix for a detailed proof of the above theorem. Our proof essentially follows the idea of *auxiliary functions* often used in the convergence proofs of approximate matrix factorization algorithms [13].

```
Input: A set $\mathcal{R}$ of relation matrices $\mathbf{R}_{ij}$; constraint matrices $\boldsymbol{\Theta}^{(t)}$
for $t \in \{1, 2, \ldots, \max_i t_i\}$; ranks $k_1, k_2, \ldots, k_r$ $(i, j \in [r])$.
Output: Matrix factors $\mathbf{S}$ and $\mathbf{G}$.
 1) Initialize $\mathbf{G}_i$ for $i = 1, 2, \ldots, r$.
 2) Repeat until convergence:
  • Construct $\mathbf{R}$ and $\mathbf{G}$ using their definitions in Eq. (1) and Eq. (3).
  • Update $\mathbf{S}$ using:
        $\mathbf{S} \leftarrow (\mathbf{G}^T\mathbf{G})^{-1}\mathbf{G}^T\mathbf{R}\mathbf{G}(\mathbf{G}^T\mathbf{G})^{-1}$.
  • Set $\mathbf{G}_i^{(e)} \leftarrow \mathbf{0}$ for $i = 1, 2, \ldots, r$.
  • Set $\mathbf{G}_i^{(d)} \leftarrow \mathbf{0}$ for $i = 1, 2, \ldots, r$.
  • For $\mathbf{R}_{ij} \in \mathcal{R}$:
        $\mathbf{G}_i^{(e)}$ += $(\mathbf{R}_{ij}\mathbf{G}_j\mathbf{S}_{ij}^T)^+ + \mathbf{G}_i(\mathbf{S}_{ij}\mathbf{G}_j^T\mathbf{G}_j\mathbf{S}_{ij}^T)^-$
        $\mathbf{G}_i^{(d)}$ += $(\mathbf{R}_{ij}\mathbf{G}_j\mathbf{S}_{ij}^T)^- + \mathbf{G}_i(\mathbf{S}_{ij}\mathbf{G}_j^T\mathbf{G}_j\mathbf{S}_{ij}^T)^+$
        $\mathbf{G}_j^{(e)}$ += $(\mathbf{R}_{ij}^T\mathbf{G}_i\mathbf{S}_{ij})^+ + \mathbf{G}_j(\mathbf{S}_{ij}^T\mathbf{G}_i^T\mathbf{G}_i\mathbf{S}_{ij})^-$
        $\mathbf{G}_j^{(d)}$ += $(\mathbf{R}_{ij}^T\mathbf{G}_i\mathbf{S}_{ij})^- + \mathbf{G}_j(\mathbf{S}_{ij}^T\mathbf{G}_i^T\mathbf{G}_i\mathbf{S}_{ij})^+$   (10)
  • For $t = 1, 2, \ldots, \max_i t_i$:
        $\mathbf{G}_i^{(e)}$ += $[\boldsymbol{\Theta}_i^{(t)}]^-\mathbf{G}_i$   for $i = 1, 2, \ldots, r$
        $\mathbf{G}_i^{(d)}$ += $[\boldsymbol{\Theta}_i^{(t)}]^+\mathbf{G}_i$   for $i = 1, 2, \ldots, r$   (11)
  • Construct $\mathbf{G}$ as:
        $\mathbf{G} \leftarrow \mathbf{G} \circ \mathrm{Diag}(\sqrt{\frac{\mathbf{G}_1^{(e)}}{\mathbf{G}_1^{(d)}}}, \sqrt{\frac{\mathbf{G}_2^{(e)}}{\mathbf{G}_2^{(d)}}}, \ldots, \sqrt{\frac{\mathbf{G}_r^{(e)}}{\mathbf{G}_r^{(d)}}})$,   (12)
    where $\circ$ denotes the Hadamard product. The $\sqrt{\cdot}$ and $\frac{\cdot}{\cdot}$ are
    entry-wise operations.
```

Fig. 2. Factorization algorithm of proposed data fusion approach (DFMF).

2.4 Stopping Criteria

In this paper we apply data fusion to infer relations between two target object types, $\mathcal{E}_i$ and $\mathcal{E}_j$. We hence define the stopping criteria that observes convergence in approximation of only the target matrix $\mathbf{R}_{ij}$. Our convergence criteria is $||\mathbf{R}_{ij} - \mathbf{G}_i\mathbf{S}_{ij}\mathbf{G}_j^T||^2 < \epsilon$, where ϵ is a user-defined parameter, possibly refined through observing log entries of the target matrix approximation error for several runs of the factorization algorithm. In our experiments ϵ was set to 10^{-5}. To reduce the computational load, the convergence criteria was assessed only every fifth iteration.

2.5 Parameter Estimation

Parameters to DFMF algorithm are factorization ranks, $k_1, k_2, \ldots, k_r$. These are chosen from a predefined interval of possible rank values such that their choice maximizes the estimated quality of the model. To reduce the number of required factorization runs we mimic the bisection method by first testing rank values at the midpoint and borders of specified ranges and then for each rank value selecting the subinterval for which the resulting model was of higher quality. We evaluate the models through the explained variance, the residual sum of squares (RSS) and a measure based on the cophenetic correlation coefficient ρ [14]. We compute these measures for the target relation matrix. The RSS is computed over observed associations (o_p^i, o_q^j) in $\mathbf{R}_{ij}$ as $\text{RSS}(\mathbf{R}_{ij}) = \sum \left[(\mathbf{R}_{ij} - \mathbf{G}_i\mathbf{S}_{ij}\mathbf{G}_j^T)(p,q)\right]^2$. Similarly, explained variance is $R^2(\mathbf{R}_{ij}) = 1 - \text{RSS}(\mathbf{R}_{ij}) / \sum[\mathbf{R}_{ij}(p,q)]^2$.

We assess the three quality scores through internal cross-validation and observe how $R^2(\mathbf{R}_{ij})$, $\text{RSS}(\mathbf{R}_{ij})$ and $\rho(\mathbf{R}_{ij})$ vary with changes of factorization ranks. We select ranks $k_1, k_2, \ldots, k_r$ where the cophenetic coefficient begins to fall, the explained variance is high and the RSS curve shows an inflection point [15].

2.6 Prediction from Matrix Factors

The approximate relation matrix $\widehat{\mathbf{R}}_{ij}$ for the target pair of object types $\mathcal{E}_i$ and $\mathcal{E}_j$ is reconstructed as $\widehat{\mathbf{R}}_{ij} = \mathbf{G}_i\mathbf{S}_{ij}\mathbf{G}_j^T$. When the model is requested to propose relations for a new object $o_{n_i+1}^i$ of type $\mathcal{E}_i$ that was not included in the training data, we need to estimate its factorized representation and use the resulting factors for prediction. We formulate a non-negative linear least-squares and solve it with an efficient interior point Newton-like method [16] for $\min_{\mathbf{x}_l \geq 0} ||(\mathbf{G}_l\mathbf{S}_{li} + \mathbf{G}_l\mathbf{S}_{il}^T)\mathbf{x}_l - \mathbf{o}_{n_i+1}^{i,l}||_2^2$, where $\mathbf{o}_{n_i+1}^{i,l} \in \mathbb{R}^{n_l}$ is the original description of object $o_{n_i+1}^i$ (if available) and $\mathbf{x}_l \in \mathbb{R}^{k_i}$ is its factorized representation (for $l = 1, 2, \ldots, r$ and $l \neq i$). A solution vector given by $\sum_l \mathbf{x}_l^{*T}$ is added to $\mathbf{G}_i$ and a new $\widehat{\mathbf{R}}_{ij} \in \mathbb{R}^{(n_i+1)\times n_j}$ is computed.

We would like to identify object pairs (o_p^i, o_q^j) for which the predicted degree of relation $\widehat{\mathbf{R}}_{ij}(p,q)$ is unusually high. We are interested in candidate pairs (o_p^i, o_q^j) for which the estimated association score $\widehat{\mathbf{R}}_{ij}(p,q)$ is greater than the mean estimated score of all known relations of o_p^i:

$$\widehat{\mathbf{R}}_{ij}(p,q) > \frac{1}{|\mathcal{A}(o_p^i, \mathcal{E}_j)|} \sum_{o_m^j \in \mathcal{A}(o_p^i, \mathcal{E}_j)} \widehat{\mathbf{R}}_{ij}(p,m), \qquad (13)$$

where $\mathcal{A}(o_p^i, \mathcal{E}_j)$ is the set of all objects of $\mathcal{E}_j$ related to o_p^i. Notice that this rule is row-centric, that is, given an object of type $\mathcal{E}_i$, it searches for objects of the other type ($\mathcal{E}_j$) that it could be related to. We can modify the rule to become column-centric, or even combine the two rules.

For example, let us consider that we are studying disease predispositions for a set of patients. Let the patients be objects of type $\mathcal{E}_i$ and diseases objects of type $\mathcal{E}_j$. A patient-centric rule would consider a patient and his medical history and through Eq. (13) propose a set of new disease associations. A disease-centric rule would instead consider all patients already associated with a specific disease and identify other patients with a sufficiently high association score.

We can combine row-centric and column-centric approaches. For example, we can first apply a row-centric approach to identify candidates of type $\mathcal{E}_i$ and then estimate the strength of association to a specific object o_q^j by reporting an inverse percentile of association score in the distribution of scores for all true associations of o_q^j, that is, by considering the scores in the q-ed column of $\widehat{\mathbf{R}}_{ij}$. In our gene function prediction study, we use row-centric approach for candidate identification and column-centric approach for association scoring, and in the experiment from cheminformatics we apply row-centric approach to both tasks.

2.7 An Ensemble Approach to Prediction

Different initializations of $\mathbf{G}_i$ may in practice give rise to different factorizations of the fusion system. To leverage this effect we construct an ensemble of factorization models. The resulting matrix factors in each model may also be different due to small random perturbations of selected factorization ranks. We use each factorization system for inference of associations (Sec. 2.6) and then select the candidate pair through a majority vote. That is, the rule from Eq. (13) must apply in more than one half of factorized systems of the ensemble. Ensembles improved the predictive accuracy and stability of the factorized system and the robustness of the results. In our experiments the ensembles combined 15 factorization models.

2.8 Matrix Factor Initialization

The inference of the factorized system in Sec. 2.1 is sensitive to the initialization of factor $\mathbf{G}$. Proper initialization sidesteps the issue of local convergence and reduces the number of iterations needed to obtain matrix factors of equal quality. We initialize $\mathbf{G}$ by separately initializing each $\mathbf{G}_i$, using algorithms for single-matrix factorization. Factors $\mathbf{S}$ are computed from $\mathbf{G}$ (Fig. 2) and do not require initialization.

Wang *et al.* (2008) [10] and several other authors [13] use simple random initialization. Other more informed initialization algorithms include random C [17], random Acol [17], non-negative double SVD and its variants [18], and k-means clustering or relaxed SVD-centroid initialization [17]. We show that the latter approaches are indeed better over a random initialization (Sec. 5.4). We use random Acol in our case study. Random Acol computes each column of $\mathbf{G}_i$ as an element-wise average of a random subset of columns in $\mathbf{R}_{ij}$.

3 Related Work

Approximate matrix factorization estimates a data matrix $\mathbf{R}$ as a product of low-rank matrix factors that are found by solving an optimization problem. In two-factor decomposition, $\mathbf{R} \in \mathbb{R}^{n\times m}$ is decomposed to a product $\mathbf{WH}$, where $\mathbf{W} \in \mathbb{R}^{n\times k}$, $\mathbf{H} \in \mathbb{R}^{k\times m}$ and $k \ll \min(n,m)$. A large class of matrix factorization algorithms minimize discrepancy between the observed matrix and its low-rank approximation, such that $\mathbf{R} \approx \mathbf{WH}$. For instance, SVD, non-negative matrix factorization and exponential family PCA all minimize Bregman divergence [19].

Although often used in data analysis for dimensionality reduction, clustering or low-rank approximation, there have been only a few applications of matrix factorization in data fusion. Lange *et al.* (2005) [20] proposed an integration by non-negative matrix factorization of a target matrix, which was a convex combination of similarity matrices obtained from multiple information sources. Their work is similar to that of Wang *et al.* (2012) [21], who applied non-negative matrix tri-factorization with input matrix completion. Note that both approaches implement early integration and can model only multiple dyadic relations. Their approaches cannot be used to model relations between more than two object types, which is a major distinction with the algorithm proposed in this paper.

Zhang *et al.* (2012) [22] proposed a joint matrix factorization to decompose a number of data matrices $\mathbf{R}_i$ into a common basis matrix $\mathbf{W}$ and different coefficient matrices $\mathbf{H}_i$, such that $\mathbf{R}_i \approx \mathbf{WH}_i$ by minimizing $\sum_i ||\mathbf{R}_i - \mathbf{WH}_i||^2$. This is an intermediate integration approach with different data sources but it can describe only relations whose objects (*i.e.* rows in $\mathbf{R}_i$) are fixed across relation matrices. Similar approaches but with various regularization types were also proposed, such as network- or relation-regularized constraints [23], [24] and hierarchical priors [25], [26]. Our work generalizes these approaches by simultaneously dealing with objects of different types, where we can vary object types along both dimensions of relation matrices, $\mathbf{R}_{ij}$) and can constrain objects of every type.

There is an abundance of work on matrix factorization models that consider a single dyadic relation matrix or multiple relation matrices between the same two types of objects [10], [27], [28], [26], [24], [21] that are subsumed in our approach. For instance, Nickel *et al.* (2011) [29] proposed a tri-factorization model for multiple dyadic relations that factorized every $\mathbf{R}_i$ as $\mathbf{R}_i \approx \mathbf{AS}_i\mathbf{A}^T$. Although their model is appropriate for certain tasks of collective learning, all $\mathbf{R}_i$ describe relations between the same two sets of objects, whereas our approach models multi-relational and multi-object type data.

Rettinger *et al.* (2012) [30] proposed context-aware tensor decomposition for relation prediction in social networks, CARTD. They decompose a tensor into additive factorized matrices using two-factor decomposition. They assume that input data is provided together with the contextual information that describes one specific relation, the recommendation. The drawback of their and similar approaches [31], [27], [32] for r-ary tensors is that in higher dimensions ($r > 3$) the tensors become increasingly sparse and the computational requirements become infeasible. Notice that here r corresponds to number of different object types in DFMF. In comparison, the approach proposed in this paper can handle tens of different object types.

Wang *et al.* (2008) [10] and Wang *et al.* (2011) [12] proposed tri-SPMF and S-NMTF, respectively, a simultaneous clustering of multi-type relational data via symmetric nonnegative matrix tri-factorization. These two methods are conceptually similar to our approach and use both inter-type and intra-type relations, but they require a full set of symmetric relation matrices, $\mathbf{R}_{ij} = \mathbf{R}_{ji}^T$. These assumptions of tri-SPMF and S-NMTF are rarely met in real-world fusion scenarios (see, for example, a fusion configuration from Fig. 3, which is not a 6-clique), where we do not have access to relation matrices between all possible pairs of object types (*i.e.* $\mathbf{R}_{ij}$ for $1 \leq i < j \leq r$). The tri-SPMF and S-NMTF algorithms do not converge to a local minimum if described relations are asymmetric ($\mathbf{R}_{ij} \neq \mathbf{R}_{ji}^T$).

We are currently witnessing increasing interest in the joint treatment of heterogeneous data sets and the emergence of approaches specifically designed for data fusion. Besides matrix factorization-based methods as reviewed above, these approaches include canonical correlation analysis [33], combining many interaction networks into a composite network [34], multiple graph clustering with linked matrix factorization [8], a mixture of Markov chains associated with different graphs [35], dependency-seeking clustering algorithms with variational Bayes [36], latent factor analysis [37], [38], nonparametric Bayes ensemble learning [39], approaches based on Bayesian theory [40], [41], [42], neural networks [43], and module guided random forests [44].

Data integration approaches from the previous paragraph either fuse input data (early integration) or predictions (late integration) and do not directly combine heterogeneous representation of objects of different types. A state-of-the-art approach that can address such data through intermediate integration is kernel-based learning. Multiple kernel learning (MKL) has been pioneered by Lanckriet *et al.* (2004) [45] and Bach *et al.* (2004) [46] and is an additive extension of single kernel SVM to incorporate multiple kernels in classification, regression and clustering. The MKL assumes that $\mathcal{E}_1, \ldots, \mathcal{E}_r$ are r different representations of the same set of n objects. Extension from single to multiple data sources is achieved by additive combination of kernel matrices, given by $\Omega = \left\{\sum_{i=1}^r \theta_i \mathbf{K}_i \middle| \forall i : \theta_i \geq 0, \sum_{i=1}^r \theta_i^\delta = 1, \mathbf{K}_i \succeq 0\right\}$, where θ_i are weights of the kernel matrices, δ is a parameter determining the norm of constraint posed on coefficients (for L_2, L_p-norm MKL, see [47], [48], [11], [49]) and $\mathbf{K}_i$ are normalized kernel matrices centered in the Hilbert space. Among other improvements, Yu *et al.* (2010) extended the framework of the MKL in Lanckriet *et al.* (2004) [45] by optimizing various norms in the dual problem of SVMs that allows non-sparse optimal kernel coefficients θ_i^*. Gönen *et al.*

(2011) [50] recently reviewed several MKL algorithms and concluded that, in general, using multiple kernels instead of a single one is useful. The heterogeneity of data sources in the MKL is resolved by transforming different object types and data structures (e.g., strings, vectors, graphs) into kernel matrices. These transformations depend on the choice of the kernels, which in turn affects the method's performance [51].

4 Experiments

We present two case studies from bioinformatics and cheminformatics, where recent technological advancements have allowed researchers to collect large and diverse experimental data sets [52], [53], [54], [39]. From bioinformatics, we study prediction of gene function, where the target relation is given by a binary matrix representing relationships between genes of the amoeba *Dictyostelium discoideum* and their associated functions or processes (Gene Ontology (GO) terms, $\mathbf{R}_{12}$). In the cheminformatics study, the binary target matrix encodes the pharmacologic actions of a subset of chemicals from PubChem database. We apply DFMF to fuse eleven data matrices for gene function prediction and six data matrices for the prediction of pharmacologic actions. During testing, we estimate the relation for a previously-unseen pair (Gene, GO Term) or (Chemical, Pharmacologic Action).

We compare DFMF to an early integration by random forests [55], [56], intermediate integration by multiple kernel learning (MKL) [11] and relational learning by matrix factorization (tri-SPMF) [10]. Kernel-based fusion used a multi-class L_2 norm MKL with Vapnik's SVM [57]. The MKL was formulated as a second order cone program (SOCP) and its dual problem was solved by the conic optimization solver SeDuMi. Random forests from the Orange data mining suite were used with default parameters. Relational learning by tri-SPMF used the matrix factorization algorithm from Wang *et al.* [10] and a procedure described in Sec. 2.6 for predicting associations.

4.1 Setup for Gene Function Prediction Task

Various classification schemes were developed to standardize the association of genes to its function. Of these, Gene Ontology (GO) [58] is adopted widely and is thus suitable for computational studies [34], [59]. In our study, given a gene, we aimed to predict a set of its associated GO terms along with the confidence of the association.

4.1.1 Data

We observed six object types (Fig. 3): genes (type 1), ontology terms (type 2), experimental conditions (type 3), publications from the PubMed database (PMID) (type 4), Medical Subject Headings (MeSH) descriptors (type 5), and KEGG pathways [60] (type 6). The data included gene expression measured during different time-points of a 24-hour development cycle [61] ($\mathbf{R}_{13}$, 14 experimental conditions), gene annotations with experimental evidence code to 148 generic slim terms from the GO ($\mathbf{R}_{12}$), PMIDs and their associated *D. discoideum* genes from dictyBase ($\mathbf{R}_{14}$), genes participating in KEGG pathways ($\mathbf{R}_{16}$), assignments of MeSH descriptors to publications from PubMed ($\mathbf{R}_{45}$), references to published work on associations between a specific GO term and gene product ($\mathbf{R}_{42}$), and associations of enzymes involved in KEGG pathways and related to GO terms ($\mathbf{R}_{62}$).

To balance $\mathbf{R}_{12}$, our target relation matrix, we added an equal number of non-associations for which there is no evidence of any type in the GO. We constrained our system by considering gene interaction scores from STRING v9.0 ($\mathbf{\Theta}_1$) and slim term similarity scores ($\mathbf{\Theta}_2$) computed as -0.2^{hops}, where hops was the length of the shortest path between two terms in the GO graph. Similarly, MeSH descriptors were constrained with the average number of hops in the MeSH hierarchy between each pair of descriptors ($\mathbf{\Theta}_5$). Constraints between KEGG pathways corresponded to the number of common ortholog groups ($\mathbf{\Theta}_6$). The slim subset of GO terms was used to limit the optimization complexity of the MKL and the number of variables in the SOCP, and to ease the computational burden of early integration by random forests, which inferred a separate model for each of the terms.

We conducted three experiments in which we considered either 100 or 1000 most GO-annotated genes or the whole *D. discoideum* genome ($\sim$12,000 genes). We also examined the predictions of gene associations with any of nine GO terms that are of specific relevance to the current research in the *Dictyostelium* community (upon consultations with Gad Shaulsky, Baylor College of Medicine, Houston, TX; see Table 2). Instead of using a generic slim subset of terms, we examined the predictions in the context of a complete set of GO terms. This resulted in a data set with $\sim$2.000 terms, each term having $\sim$10 direct gene annotations.

4.1.2 Preprocessing for Kernel-Based Fusion

We generated an RBF kernel for gene expression measurements from $\mathbf{R}_{13}$ with the RBF function $\kappa(\mathbf{x}_i, \mathbf{x}_j) = \exp(-||\mathbf{x}_i - \mathbf{x}_j||^2/2\sigma^2)$, and a linear kernel for $[0,1]$-protein-interaction matrix from $\mathbf{\Theta}_1$. This particular choice of kernels was motivated by the experimental study and kernel comparison in [5]. Kernels were applied to data matrices. We used a linear kernel to generate a kernel matrix from *D. discoideum* specific genes that participate in pathways ($\mathbf{R}_{16}$), and a kernel matrix from PMIDs and their associated genes ($\mathbf{R}_{14}$). Several data sources describe relations between object types other than genes. For kernel-based fusion we had to transform them to explicitly relate to genes. For instance, to relate genes and MeSH descriptors,

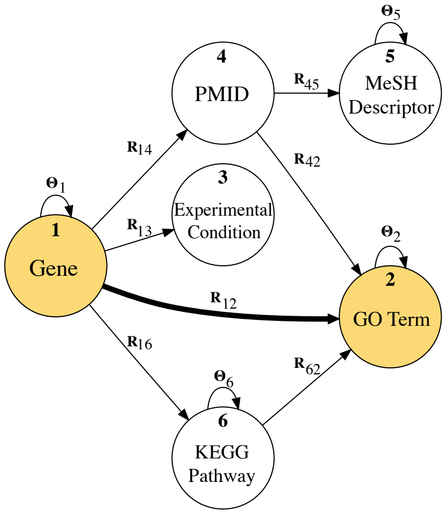


Fig. 3. The fusion configuration for gene function prediction task in *D. discoideum*. Some relations are entirely missing, for instance $\mathbf{R}_{23}$. Nodes represent object types used in our study. Edges correspond to relation and constraint matrices. The arc that represents the target matrix $\mathbf{R}_{12}$ and its object types are highlighted.

we counted the number of publications that were associated with a specific gene ($\mathbf{R}_{14}$) and were assigned a specific MeSH descriptor ($\mathbf{R}_{45}$, see also Fig. 3). A linear kernel was applied to the resulting matrix. Kernel matrices that incorporated relations between KEGG pathways and GO terms ($\mathbf{R}_{62}$), and publications and GO terms were obtained in similar fashion.

To represent the hierarchical structure of MeSH descriptors ($\mathbf{\Theta}_5$), the semantic structure of the GO graph ($\mathbf{\Theta}_2$) and ortholog groups that correspond to KEGG pathways ($\mathbf{\Theta}_6$), we considered the genes as nodes in three distinct large weighted graphs. In the graph for $\mathbf{\Theta}_5$, the link between two genes was weighted by the similarity of their associated sets of MeSH descriptors using information from $\mathbf{R}_{14}$ and $\mathbf{R}_{45}$. We considered the MeSH hierarchy to measure these similarities. Similarly, for the graph for $\mathbf{\Theta}_2$ we considered the GO semantic structure in computing similarities of sets of GO terms associated with genes. In the graph for $\mathbf{\Theta}_6$, the gene edges were weighted by the number of common KEGG ortholog groups. Kernel matrices were constructed with a diffusion kernel [62].

The resulting kernel matrices $\mathbf{K} \in \mathbb{R}^{n\times n}$ were centered as $\mathbf{K}^c(i,j) = \mathbf{K}(i,j) - 1/n\sum_i \mathbf{K}(i,j) - 1/n\sum_j \mathbf{K}(i,j) + 1/n^2\sum_{ij}\mathbf{K}(i,j)$ and normalized as $\mathbf{K}^n(i,j) = \mathbf{K}^c(i,j)/\sqrt{\mathbf{K}^c(i,i)\mathbf{K}^c(j,j)}$. The parameters for all kernels were selected through internal cross-validation. In cross-validation, only the training part of the matrices was optimized for learning, while centering and normalization were performed on the entire data set. The prediction task was defined through the classification matrix of genes and their associated GO slim terms from $\mathbf{R}_{12}$.

4.1.3 Preprocessing for Early Integration

The gene-related data matrices prepared for kernel-based fusion were also used for early integration and were concatenated into a single data table. Each row in the table represented a gene profile obtained from all available data sources. For our case study, each gene was characterized by a fixed 9,362-dimensional feature vector. For each GO slim term, we then separately developed a classifier with a random forest of classification trees and reported cross-validated results.

4.1.4 Preprocessing for tri-SPMF Learning

Relation and constraint matrices prepared for DFMF were also used for tri-SPMF factorization algorithm. Tri-SPMF requires a full set of relation matrices between all pairs of object types. Thus, we used zero matrices for non-existing relations from Fig. 3. For instance, $\mathbf{R}_{63}$ and $\mathbf{\Theta}_4$ were represented by zero matrices of proper dimensions. Because tri-SPMF requires that relations are symmetric, we set $\mathbf{R}_{ji} = \mathbf{R}_{ij}^T$ for all available relation matrices.

4.2 Setup for Pharmacologic Action Prediction Task

Identification of the mechanisms of action of chemical compounds is a crucial task in drug discovery [63], [64]. Here, our aim was to computationally predict pharmacologic actions of chemical compounds as defined in the PubChem database [65].

4.2.1 Data

We considered six object types (Fig. 4): chemicals (type 1), PubChem's [65] pharmacologic actions (type 2), publications from the PubMed database (PMID) (type 3), depositors of chemical substances (type 4) and their categorization (type 6), and PubChem substructure fingerprints (type 5).

The data included 1,260 chemicals extracted from the complete DrugBank [66] database (accessed in Feb. 2014) that were identified with at least one pharmacologic action in the PubChem Compound database. In that way, every chemical (drug) was assigned one or more MeSH headings that described its pharmacologic actions and corresponded to D27.505 tree of the 2014 MeSH Tree Structure (target relation $\mathbf{R}_{12}$). For example, established pharmacologic actions for Aspirin include "Anti-Inflammatory Agents, Non-Steroidal", "Fibrinolytic Agents" and "Antipyretics." To increase the number of chemicals assigned to a particular pharmacologic action, the actions of the chemical also included those from its direct parents in the D27.505 tree.

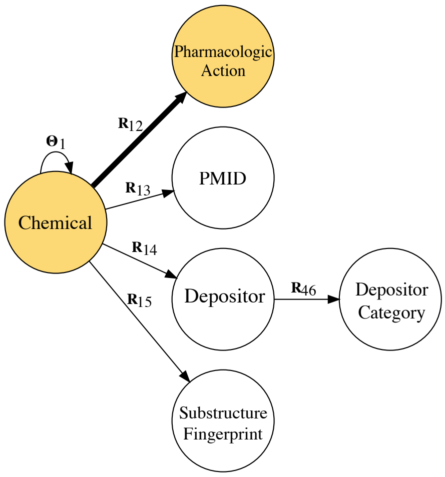


Fig. 4. The fusion configuration for the prediction of pharmacologic actions of chemicals, with object types denoted with nodes and relations between them with edges. The edge representing the target relation and its corresponding data matrix $\mathbf{R}_{12}$ is highlighted.

Other data considered were publications from the PubMed database ($\mathbf{R}_{13}$), data on depositors who submitted substances of the chemicals present in PubChem Compound records ($\mathbf{R}_{14}$), categories of data depositors ($\mathbf{R}_{46}$) and PubChem substructure fingerprints ($\mathbf{R}_{15}$). These fingerprints consist of a series of 881 binary indicators, each denoting the presence or absence of a particular substructure in a molecule. Collectively, these binary keys provide a “fingerprint” of a particular chemical structure form. Chemicals are constrained by a matrix of substructure-based Tanimoto 2D similarity ($\mathbf{\Theta}_1$) obtained through PubChem Score Matrix Service.

4.2.2 Preprocessing for Alternative Learning Methods

For the kernel-based fusion, we generated the kernel matrices for chemicals from $\mathbf{R}_{13}$, $\mathbf{R}_{14}$, $\mathbf{R}_{15}$ and $\mathbf{\Theta}_1$ (Fig. 4) using the polynomial kernel of degree 2. We included data on depositors ($\mathbf{R}_{46}$) by applying a polynomial kernel to $\mathbf{R}_{14}\mathbf{R}_{46}$. The resulting kernel matrices were centered and normalized, and the kernel parameters were selected in internal cross-validation (see Sec. 4.1.2 for details). Preprocessing for early integration by random forests and tri-SPMF learning followed the same procedures as described in Sec. 4.1.3 and Sec. 4.1.4, respectively. The prediction task was defined by the associations of chemicals to pharmacologic actions given by $\mathbf{R}_{12}$ (Fig. 4).

4.3 Scoring

We estimated the quality of inferred models by ten-fold cross-validation. In each iteration, we split the set of genes (chemicals) to a train and test set. The corresponding data on genes (chemicals) from the test set was entirely omitted from the training data. We developed prediction models from the training data and tested them on the genes (chemicals) from the test set. The performance was evaluated using an F_1 score, a harmonic mean of precision and recall, and area under ROC curve (AUC). Both scores were averaged across cross-validation runs.

5 Results and Discussion

5.1 Predictive Performance

Table 1 presents the cross-validated F_1 and AUC scores for both gene function prediction (data set of slim GO terms) and prediction of pharmacologic actions. The accuracy of DFMF is at least comparable to MKL and substantially higher than that of early integration by random forests and relational learning by tri-SPMF. When more genes and hence more data were considered for the gene function prediction the performance of all four fusion approaches improved.

Poorer performance of tri-SPMF was most probably due to required introduction of relations into factorized system that were not present in the input data. Consequently, the ability of tri-SPMF to infer relations of interest between other object types deteriorated considerably. Notice also that tri-SPMF could not be applied if fusion schemes in Figs. 3 or 4 would contain asymmetric or one-way relations, such as those from the analysis of signed networks [67] and computational biology [68], among others. We also observed numerical instability with tri-SPMF, which was exhibited as an increase in the value of objective function between successive iterations. In contrast, DFMF exhibited numerical stability in all experiments (results not shown).

The accuracy for nine GO terms selected by domain expert is given in Table 2. The DFMF performs consistently better than the other three approaches. Again, the early integration by random forests is inferior to all three intermediate integration methods. Notice that, with only a few exceptions, both F_1 and AUC scores of DFMF are high. This is important, as all nine gene processes and functions observed are relevant for current research of *D. discoideum* where the methods for data fusion can yield new candidate genes for focused experimental studies.

Our fusion approach is faster than multiple kernel learning. DFMF required 18 minutes of runtime on a standard desktop computer compared to 77 minutes for MKL to finish one iteration of cross-validation of the whole-genome variant of gene function prediction task. The factorization algorithm of DFMF also took less time to execute than tri-SPMF due to redundant representation of fusion system required by tri-SPMF.

5.2 Sensitivity to Inclusion of Data Sources

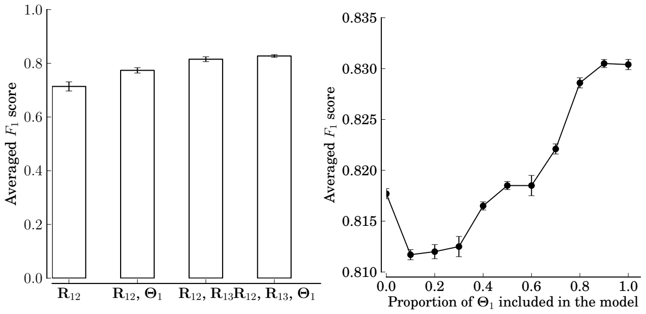
Inclusion of additional data sources improves the accuracy of prediction models. We illustrate this for gene

TABLE 1
Cross-validated F_1 and AUC accuracy scores for fusion by matrix factorization (DFMF), kernel-based method (MKL), random forests (RF) and relational learning-based matrix factorization (tri-SPMF).

Prediction task	DFMF		MKL		RF		tri-SPMF	
	F_1	AUC	F_1	AUC	F_1	AUC	F_1	AUC
100 *D. discoideum* genes	0.799	0.801	0.781	0.788	0.761	0.785	0.731	0.724
1000 *D. discoideum* genes	0.826	0.823	0.787	0.798	0.767	0.788	0.756	0.741
Whole *D. discoideum* genome	0.831	0.849	0.800	0.821	0.782	0.801	0.778	0.787
Pharmacologic actions	0.663	0.834	0.639	0.811	0.643	0.819	0.641	0.810

TABLE 2
Gene Ontology term-specific cross-validated F_1 and AUC accuracy scores for fusion by matrix factorization (DFMF), kernel-based method (MKL), random forests (RF) and relational learning-based matrix factorization (tri-SPMF). Terms in Gene Ontology belong to one of three namespaces, biological process (BP), molecular function (MF) or cellular component.

GO term name	Term identifier	Namespace	Size	DFMF		MKL		RF		tri-SPMF	
				F_1	AUC	F_1	AUC	F_1	AUC	F_1	AUC
Activation of adeny. cyc. act.	0007190	BP	11	0.834	0.844	0.770	0.781	0.758	0.601	0.729	0.731
Chemotaxis	0006935	BP	58	0.981	0.980	0.794	0.786	0.538	0.724	0.804	0.810
Chemotaxis to cAM	0043327	BP	21	0.922	0.910	0.835	0.862	0.798	0.767	0.838	0.815
Phagocytosis	0006909	BP	33	0.956	0.932	0.892	0.901	0.789	0.619	0.836	0.810
Response to bacterium	0009617	BP	51	0.899	0.870	0.788	0.761	0.785	0.761	0.817	0.831
Cell-cell adhesion	0016337	BP	14	0.883	0.861	0.867	0.856	0.728	0.725	0.799	0.834
Actin binding	0003779	MF	43	0.676	0.781	0.664	0.658	0.642	0.737	0.671	0.682
Lysozyme activity	0003796	MF	4	0.782	0.750	0.774	0.750	0.754	0.625	0.747	0.625
Seq.-spec. DNA bind. t. f. a.	0003700	MF	79	0.956	0.948	0.894	0.901	0.732	0.759	0.892	0.852



(a) (b)

Fig. 5. Adding new data sources (a) or incorporating more object-type-specific constraints in $\mathbf{\Theta}_1$ (b) both increase the accuracy of matrix factorization-based models for gene function prediction task.

function prediction in Fig. 5a, where we started with only the target data source $\mathbf{R}_{12}$ and then added either $\mathbf{R}_{13}$ or $\mathbf{\Theta}_1$ or both. Similar effects were observed when we studied other combinations of data sources (not shown here for brevity). Notice also that due to ensembling the cross-validated variance of F_1 is small.

5.3 Sensitivity to Inclusion of Constraints

We varied the sparseness of gene constraint matrix $\mathbf{\Theta}_1$ by holding out a random subset of protein-protein interactions. We set the entries of $\mathbf{\Theta}_1$ that corresponded to held-out constraints to zero so that they did not affect the cost function during optimization. Fig. 5b shows that including additional information on genes in the form of constraints improves the predictive performance of DFMF for gene function prediction.

5.4 Matrix Factor Initialization Study

We studied the effect of matrix factor initialization on DFMF by observing the reconstruction error after one and after twenty iterations of optimization procedure, the latter being about one fourth of the iterations required for the optimization algorithm to converge when predicting gene functions. We estimated the error relative to the optimal $(k_1, k_2, \dots, k_6)$-rank approximation given by the SVD. For iteration v and matrix $\mathbf{R}_{ij}$ the error was computed by:

$$\mathrm{Err}_{ij}(v) = \frac{||\mathbf{R}_{ij} - \mathbf{G}_i^{(v)}\mathbf{S}_{ij}^{(v)}(\mathbf{G}_j^T)^{(v)}||^2 - d_F(\mathbf{R}_{ij}, [\mathbf{R}_{ij}]_k)}{d_F(\mathbf{R}_{ij}, [\mathbf{R}_{ij}]_k)}, \quad (14)$$

where $\mathbf{G}_i^{(v)}$, $\mathbf{G}_j^{(v)}$ and $\mathbf{S}_{ij}^{(v)}$ were matrix factors obtained after v iterations of factorization algorithm. In Eq. (14), $d_F(\mathbf{R}_{ij}, [\mathbf{R}_{ij}]_k) = ||\mathbf{R}_{ij} - \mathbf{U}_k\mathbf{\Sigma}_k\mathbf{V}_k^T||^2$ denotes the Frobenius distance between $\mathbf{R}_{ij}$ and its k-rank approximation given by the SVD, where $k = \max(k_i, k_j)$ is the approximation rank. $\mathrm{Err}_{ij}(v)$ is a pessimistic measure of quantitative accuracy because of the choice of k. This error measure is similar

TABLE 3
Effect of initialization algorithm on reconstruction error of DFMF's factorization model.

Method	Time $\mathbf{G}^{(0)}$	Storage $\mathbf{G}^{(0)}$	$\text{Err}_{12}(1)$	$\text{Err}_{12}(20)$
Rand.	0.0011 s	618K	5.11	3.61
Rand. C	0.1027 s	553K	2.97	1.67
Rand. Acol	0.0654 s	505K	1.59	1.30
K-means	0.4029 s	562K	2.47	2.20
NNDSVDa	0.1193 s	562K	3.50	2.01

to the error of the two-factor non-negative matrix factorization from [17].

Table 3 shows the results for the experiment with 1000 most GO-annotated *D. discoideum* genes and selected factorization ranks $k_i < 65$, $i \in [6]$. The informed initialization algorithms surpass the random initialization. Of these, the random Acol algorithm performs best in terms of accuracy and is also one of the simplest.

5.5 Early Integration by Matrix Factorization

Our data fusion approach simultaneously factorizes individual blocks of data in $\mathbf{R}$. Alternatively, we could also disregard the data structure, and treat $\mathbf{R}$ as a single data matrix. Such data treatment would transform our data fusion approach to that of early integration and lose the benefits of structured system and source-specific factorization. To prove this experimentally, we considered the $1{,}000$ most GO-annotated *D. discoideum* genes. The resulting cross-validated F_1 score for factorization-based early integration was 0.576, compared to 0.826 obtained with our proposed data fusion algorithm. This result is not surprising as neglecting the structure of the system also causes the loss of the structure in matrix factors and the loss of zero blocks in factors $\mathbf{S}$ and $\mathbf{G}$ from Eq. (3). Clearly, data structure carries substantial information and should be retained in the model.

6 Conclusion

We have proposed a new matrix factorization-based data fusion algorithm called DFMF. The approach is flexible and, in contrast to state-of-the-art kernel-based methods, requires minimal, if any, preprocessing of input data. This latter feature, the ability to model multi-relational and multi-object type data, and DFMF's excellent accuracy and time response, are the principal advantages of our new algorithm.

DFMF can model any collection of data sets, each of which can be expressed as a matrix. Tasks from bioinformatics and cheminformatics considered here that were traditionally regarded as classification problems exemplify just one type of data mining problems that can be addressed with our method. We anticipate the utility of factorization-based data fusion in multi-task learning, association mining, clustering, link prediction or structured output prediction.

Appendix
Proof of Convergence (Theorem 2)

Our proof follows the concept of *auxiliary functions* often used in convergence proofs of approximate matrix factorization algorithms [13]. The proof is performed by introducing an appropriate function $F(\mathbf{G}, \mathbf{G}')$, which is an auxiliary function of the objective $J(\mathbf{G}; \mathbf{S})$ that satisfies:

$$F(\mathbf{G}', \mathbf{G}') = J(\mathbf{G}'; \mathbf{S}),$$
$$F(\mathbf{G}, \mathbf{G}') \geq J(\mathbf{G}; \mathbf{S}).$$

If such an auxiliary function F can be found and if $\mathbf{G}$ is updated in $(m+1)$-th iteration as:

$$\mathbf{G}^{(m+1)} = \arg\min_{\mathbf{G}} F(\mathbf{G}, \mathbf{G}^{(m)}), \tag{15}$$

then the following holds:

$$\begin{aligned} J(\mathbf{G}^{(m+1)}; \mathbf{S}) &\leq F(\mathbf{G}^{(m+1)}, \mathbf{G}^{(m)}) \leq \\ &\leq F(\mathbf{G}^{(m)}, \mathbf{G}^{(m)}) = \\ &= J(\mathbf{G}^{(m)}; \mathbf{S}). \end{aligned} \tag{16}$$

That is, if F is an auxiliary function of $J(\mathbf{G}; \mathbf{S})$, then $J(\mathbf{G}; \mathbf{S})$ is nonincreasing under the update Eq. (15). In the proof we show the the update step for $\mathbf{G}$ in Eq. (12) is exactly the update in Eq. (15) with a proper auxiliary function. For that we make use of an auxiliary function specified by Wang *et al.* (2008) (Appendix II in [10]). Wang *et al.* (2008) constructed a function $F_{\text{Wang}}(\mathbf{A}, \mathbf{A}'; \mathbf{B}, \mathbf{C}, \mathbf{D})$ and showed that it satisfied the conditions of auxiliary functions for functions of the form $J(\mathbf{A}; \mathbf{B}, \mathbf{C}, \mathbf{D}) = \text{tr}(-2\mathbf{A}^T\mathbf{B} + \mathbf{A}\mathbf{D}\mathbf{A}^T) + \text{tr}(\mathbf{A}^T\mathbf{C}\mathbf{A})$, where $\mathbf{C}$ and $\mathbf{D}$ are symmetric, and $\mathbf{A}$ is nonnegative. To prove the convergence of our algorithm, we show that the objective function from Eq. (5) is a special case of $J(\mathbf{A}; \mathbf{B}, \mathbf{C}, \mathbf{D})$.

Proof of Theorem 2: First, we view $J(\mathbf{G}; \mathbf{S})$ in Eq. (6) as a function of $\mathbf{G}_1$ and construct the auxiliary function $F_{\text{Wang}}(\mathbf{A}, \mathbf{A}'; \mathbf{B}, \mathbf{C}, \mathbf{D})$ such that:

$$\begin{aligned} \mathbf{A} &= \mathbf{G}_1, \\ \mathbf{B} &= \sum_{j:\mathbf{R}_{1j} \in \mathcal{R}} \mathbf{R}_{1j}\mathbf{G}_j\mathbf{S}_{1j}^T + \sum_{i:\mathbf{R}_{i1} \in \mathcal{R}} \mathbf{R}_{i1}^T\mathbf{G}_i\mathbf{S}_{i1}, \\ \mathbf{C} &= \sum_{t=1}^{\max_i t_i} \mathbf{\Theta}_1^{(t)}, \\ \mathbf{D} &= \sum_{j:\mathbf{R}_{1j} \in \mathcal{R}} \mathbf{S}_{1j}\mathbf{G}_j^T\mathbf{G}_j\mathbf{S}_{1j}^T + \sum_{i:\mathbf{R}_{i1} \in \mathcal{R}} \mathbf{S}_{i1}^T\mathbf{G}_i^T\mathbf{G}_i\mathbf{S}_{i1}. \end{aligned} \tag{17}$$

With these values for $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$ and $\mathbf{D}$, the auxilary function F_{Wang} is convex in $\mathbf{G}_1$. Notice that each of the two summation terms in the right-hand side expression for $\mathbf{D}$ represents the sum of the symmetric matrices of the form

$(\mathbf{G}_j\mathbf{S}_{1j}^T)^T(\mathbf{G}_j\mathbf{S}_{1j}^T)$ and $(\mathbf{G}_i\mathbf{S}_{i1})^T(\mathbf{G}_i\mathbf{S}_{i1})$, respectively. Thus, $\mathbf{D}$ is symmetric. The global minimum (Eq. (15)) of $F_{\text{Wang}}(\mathbf{A},\mathbf{A}';\mathbf{B},\mathbf{C},\mathbf{D})$ is exactly the update rule for $\mathbf{G}_1$ in Eq. (10)–(12).

We repeat this process by constructing the remaining $r-1$ auxiliary functions by separately considering $J(\mathbf{G};\mathbf{S})$ as a function of matrix factors $\mathbf{G}_2\ldots,\mathbf{G}_r$. From the theory of auxiliary functions it then follows that J is nonincreasing under the update rules for each of $\mathbf{G}_1,\mathbf{G}_2,\ldots,\mathbf{G}_r$. Letting $J(\mathbf{G}_1,\mathbf{G}_2,\ldots,\mathbf{G}_r,\mathbf{S}) = J(\mathbf{G};\mathbf{S})$, we have:

$$\begin{aligned} J(\mathbf{G}_1^0,\mathbf{G}_2^0,\ldots,\mathbf{G}_r^0,\mathbf{S}) &\geq J(\mathbf{G}_1^1,\mathbf{G}_2^0,\ldots,\mathbf{G}_r^0,\mathbf{S}) \geq \\ &\geq \cdots \\ &\geq J(\mathbf{G}_1^1,\mathbf{G}_2^1,\ldots,\mathbf{G}_r^1,\mathbf{S}). \end{aligned}$$

Since $J(\mathbf{G};\mathbf{S})$ is certainly bounded from below by zero, we proved the theorem. □

Acknowledgments

We thank Janez Demšar and Gad Shaulsky for their comments on the early version of this manuscript. We acknowledge the support for our work from the ARRS (P2-0209, J2-9699, L2-1112, J2-5480), NIH (P01-HD39691) and EU (Health-F5-2010-242038).

References

[1] S. M. Rappaport and M. T. Smith, "Environment and disease risks," *Science*, vol. 330, no. 6003, pp. 460–461, 2010.

[2] S. Aerts, D. Lambrechts, S. Maity, P. Van Loo, B. Coessens, F. De Smet, L.-C. Tranchevent, B. De Moor, P. Marynen, B. Hassan, P. Carmeliet, and Y. Moreau, "Gene prioritization through genomic data fusion," *Nature Biotechnology*, vol. 24, no. 5, pp. 537–544, 2006.

[3] H. Boström, S. F. Andler, M. Brohede, R. Johansson, A. Karlsson, J. vanLaere, L. Niklasson, M. Nilsson, A. Persson, and T. Ziemke, "On the definition of information fusion as a field of research," University of Skovde, School of Humanities and Informatics, Skovde, Sweden, Tech. Rep., 2007.

[4] D. Greene and P. Cunningham, "A matrix factorization approach for integrating multiple data views," in *Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases*. Berlin, Heidelberg: Springer-Verlag, 2009, pp. 423–438.

[5] G. R. G. Lanckriet, T. De Bie, N. Cristianini, M. I. Jordan, and W. S. Noble, "A statistical framework for genomic data fusion," *Bioinformatics*, vol. 20, no. 16, pp. 2626–2635, 2004.

[6] P. Pavlidis, J. Cai, J. Weston, and W. S. Noble, "Learning gene functional classifications from multiple data types," *Journal of Computational Biology*, vol. 9, pp. 401–411, 2002.

[7] O. Gevaert, F. De Smet, D. Timmerman, Y. Moreau, and B. De Moor, "Predicting the prognosis of breast cancer by integrating clinical and microarray data with Bayesian networks," *Bioinformatics*, vol. 22, no. 14, pp. e184–90, 2006.

[8] W. Tang, Z. Lu, and I. S. Dhillon, "Clustering with multiple graphs," in *Proceedings of the 9th IEEE International Conference on Data Mining*, 2009, pp. 1016–1021.

[9] M. H. van Vliet, H. M. Horlings, M. J. van de Vijver, M. J. T. Reinders, and L. F. A. Wessels, "Integration of clinical and gene expression data has a synergetic effect on predicting breast cancer outcome," *PLoS One*, vol. 7, no. 7, p. e40358, 2012.

[10] F. Wang, T. Li, and C. Zhang, "Semi-supervised clustering via matrix factorization," in *Proceedings of the SIAM International Conference on Data Mining*, 2008, pp. 1–12.

[11] S. Yu, T. Falck, A. Daemen, L.-C. Tranchevent, J. A. Suykens, B. De Moor, and Y. Moreau, "L_2-norm multiple kernel learning and its application to biomedical data fusion," *BMC Bioinformatics*, vol. 11, p. 309, 2010.

[12] H. Wang, H. Huang, and C. H. Q. Ding, "Simultaneous clustering of multi-type relational data via symmetric nonnegative matrix tri-factorization," in *Proceedings of the 20th ACM CIKM International Conference on Information and Knowledge Management*, 2011, pp. 279–284.

[13] D. D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization," in *Advances in Neural Information Processing Systems*, T. K. Leen, T. G. Dietterich, and V. Tresp, Eds. MIT Press, 2000, pp. 556–562.

[14] J.-P. Brunet, P. Tamayo, T. R. Golub, and J. P. Mesirov, "Metagenes and molecular pattern discovery using matrix factorization," *PNAS*, vol. 101, no. 12, pp. 4164–4169, 2004.

[15] L. N. Hutchins, S. M. Murphy, P. Singh, and J. H. Graber, "Position-dependent motif characterization using nonnegative matrix factorization," *Bioinformatics*, vol. 24, no. 23, pp. 2684–2690, 2008.

[16] M. H. Van Benthem and M. R. Keenan, "Fast algorithm for the solution of large-scale non-negativity-constrained least squares problems," *Journal of Chemometrics*, vol. 18, no. 10, pp. 441–450, 2004.

[17] R. Albright, J. Cox, D. Duling, A. N. Langville, and C. D. Meyer, "Algorithms, initializations, and convergence for the nonnegative matrix factorization," Department of Mathematics, North Carolina State University, Tech. Rep., 2006.

[18] C. Boutsidis and E. Gallopoulos, "SVD based initialization: A head start for nonnegative matrix factorization," *Pattern Recognition*, vol. 41, no. 4, pp. 1350–1362, 2008.

[19] A. P. Singh and G. J. Gordon, "A unified view of matrix factorization models," in *Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases*, 2008, pp. 358–373.

[20] T. Lange and J. M. Buhmann, "Fusion of similarity data in clustering," in *Advances in Neural Information Processing Systems*, 2005, pp. 723–730.

[21] H. Wang, H. Huang, C. H. Q. Ding, and F. Nie, "Predicting protein-protein interactions from multimodal biological data sources via nonnegative matrix tri-factorization," in *Research in Computational Molecular Biology*, vol. 7262. Springer, 2012, pp. 314–325.

[22] S. Zhang, C.-C. Liu, W. Li, H. Shen, P. W. Laird, and X. J. Zhou, "Discovery of multi-dimensional modules by integrative analysis of cancer genomic data," *Nucleic Acids Research*, vol. 40, no. 19, pp. 9379–9391, 2012.

[23] W.-j. Li and D.-y. Yeung, "Relation regularized matrix factorization," in *Proceedings of the 20th International Joint Conference on Artificial Intelligence*, 2007, pp. 1126–1131.

[24] S.-H. Zhang, Q. Li, J. Liu, and X. J. Zhou, "A novel computational framework for simultaneous integration of multiple types of genomic data to identify microRNA-gene regulatory modules," *Bioinformatics*, vol. 27, no. 13, pp. 401–409, 2011.

[25] A. P. Singh and G. J. Gordon, "Relational learning via collective matrix factorization," in *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2008, pp. 650–658.

[26] ——, "A Bayesian matrix factorization model for relational data," in *Proceedings of the 26th Conference on Uncertainty in Artificial Intelligence*, 2010, pp. 556–563.

[27] I. Sutskever, "Modelling relational data using bayesian clustered tensor factorization," in *Advances in Neural Information Processing Systems*, 2009, pp. 1821–1828.

[28] T. Li, Y. Zhang, and V. Sindhwani, "A non-negative matrix tri-factorization approach to sentiment classification with lexical prior knowledge," in *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, 2009, pp. 244–252.

[29] M. Nickel, "A three-way model for collective learning on multi-relational data," in *Proceedings of the 28th International Conference on Machine Learning*, 2011, pp. 809–816.

[30] A. Rettinger, H. Wermser, Y. Huang, and V. Tresp, "Context-aware tensor decomposition for relation prediction in social networks," *Social Network Analysis and Mining*, vol. 2, no. 4, pp. 373–385, 2012.

[31] T. G. Kolda and B. W. Bader, "Tensor decompositions and applications," *SIAM Review*, vol. 51, no. 3, pp. 455–500, 2009.

[32] S. Rendle, Z. Gantner, C. Freudenthaler, and L. Schmidt-Thieme, "Fast context-aware recommendations with factoriza-

tion machines," *Proceedings of the 34th ACM SIGIR International Conference on Research and Development in Information*, pp. 635–644, 2011.

[33] K. Chaudhuri, S. M. Kakade, K. Livescu, and K. Sridharan, "Multi-view clustering via canonical correlation analysis," in *Proceedings of the 26th International Conference on Machine Learning*, 2009, pp. 129–136.

[34] S. Mostafavi and Q. Morris, "Combining many interaction networks to predict gene function and analyze gene lists," *Proteomics*, vol. 12, no. 10, pp. 1687–96, 2012.

[35] D. Zhou and C. J. C. Burges, "Spectral clustering and transductive learning with multiple views," in *Proceedings of the 24th International Conference on Machine Learning*, 2007, pp. 1159–1166.

[36] A. Klami and S. Kaski, "Probabilistic approach to detecting dependencies between data sets," *Neurocomput.*, vol. 72, no. 1-3, pp. 39–46, 2008.

[37] H. F. Lopes, D. Gamerman, and E. Salazar, "Generalized spatial dynamic factor models," *Computational Statistics & Data Analysis*, vol. 55, no. 3, pp. 1319–1330, 2011.

[38] J. Luttinen and A. Ilin, "Variational Gaussian-process factor analysis for modeling spatio-temporal data," in *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2009, pp. 1177–1185.

[39] C. Xing and D. B. Dunson, "Bayesian inference for genomic data integration reduces misclassification rate in predicting protein-protein interactions," *PLoS Computational Biology*, vol. 7, no. 7, p. e1002110, 2011.

[40] Y. Zhang and Q. Ji, "Active and dynamic information fusion for multisensor systems with dynamic Bayesian networks," *Trans. Sys. Man Cyber. Part B*, vol. 36, no. 2, pp. 467–472, 2006.

[41] A. Alexeyenko and E. L. L. Sonnhammer, "Global networks of functional coupling in eukaryotes from comprehensive data integration," *Genome Research*, vol. 19, no. 6, pp. 1107–16, 2009.

[42] C. Huttenhower, K. T. Mutungu, N. Indik, W. Yang, M. Schroeder, J. J. Forman, O. G. Troyanskaya, and H. A. Coller, "Detailing regulatory networks through large scale data integration," *Bioinformatics*, vol. 25, no. 24, pp. 3267–3274, 2009.

[43] G. A. Carpenter, S. Martens, and O. J. Ogas, "Self-organizing information fusion and hierarchical knowledge discovery: a new framework using ARTMAP neural networks," *Neural Netw.*, vol. 18, no. 3, pp. 287–295, 2005.

[44] Z. Chen and W. Zhang, "Integrative analysis using module-guided random forests reveals correlated genetic factors related to mouse weight," *PLoS Computational Biology*, vol. 9, no. 3, p. e1002956, 2013.

[45] G. R. G. Lanckriet, N. Cristianini, P. Bartlett, L. E. Ghaoui, and M. I. Jordan, "Learning the kernel matrix with semidefinite programming," *J. Mach. Learn. Res.*, vol. 5, pp. 27–72, 2004.

[46] F. R. Bach, G. R. G. Lanckriet, and M. I. Jordan, "Multiple kernel learning, conic duality, and the smo algorithm," in *Proceedings of the 21st International Conference on Machine Learning*, 2004, pp. 6–14.

[47] M. Kloft, U. Brefeld, S. Sonnenburg, P. Laskov, K.-R. Muller, and A. Zien, "Efficient and accurate L_p-norm multiple kernel learning," *Advances in Neural Information Processing Systems*, vol. 21, pp. 997–1005, 2009.

[48] M. Kloft, U. Brefeld, S. Sonnenburg, and A. Zien, "L_p-norm multiple kernel learning," *J. Mach. Learn. Res.*, vol. 12, pp. 953–997, 2011.

[49] S. Yu, L. Tranchevent, X. Liu, W. Glanzel, J. A. K. Suykens, B. De Moor, and Y. Moreau, "Optimized data fusion for kernel k-means clustering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 5, pp. 1031–1039, 2012.

[50] M. Gönen and E. Alpaydın, "Multiple kernel learning algorithms," *J. Mach. Learn. Res.*, vol. 12, pp. 2211–2268, 2011.

[51] R. Debnath and H. Takahashi, "Kernel selection for the support vector machine," *IEICE Transactions*, vol. 87-D, no. 12, pp. 2903–2904, 2004.

[52] D. Parikh and R. Polikar, "An ensemble-based incremental learning approach to data fusion," *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, vol. 37, no. 2, pp. 437–450, 2007.

[53] G. Pandey, B. Zhang, A. N. Chang, C. L. Myers, J. Zhu, V. Kumar, and E. E. Schadt, "An integrative multi-network and multi-classifier approach to predict genetic interactions," *PLoS Computational Biology*, vol. 6, no. 9, 2010.

[54] R. S. Savage, Z. Ghahramani, J. E. Griffin, B. J. de la Cruz, and D. L. Wild, "Discovering transcriptional modules by Bayesian data integration," *Bioinformatics*, vol. 26, no. 12, pp. i158–i167, 2010.

[55] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[56] A.-L. Boulesteix, C. Porzelius, and M. Daumer, "Microarray-based classification and clinical predictors: on combined classifiers and additional predictive value," *Bioinformatics*, vol. 24, no. 15, pp. 1698–1706, 2008.

[57] J. Ye, S. Ji, and J. Chen, "Multi-class discriminant kernel learning via convex programming," *J. Mach. Learn. Res.*, vol. 9, pp. 719–758, 2008.

[58] M. Ashburner, C. A. Ball, J. A. Blake, D. Botstein, H. Butler, J. M. Cherry, A. P. Davis, K. Dolinski, S. S. Dwight, J. T. Eppig, M. A. Harris, D. P. Hill, L. Issel-Tarver, A. Kasarskis, S. Lewis, J. C. Matese, J. E. Richardson, M. Ringwald, G. M. Rubin, and G. Sherlock, "Gene Ontology: Tool for the unification of biology," *Nature Genetics*, vol. 25, no. 1, pp. 25–29, 2000.

[59] P. Radivojac, W. T. Clark, T. R. Oron, A. M. Schnoes, T. Wittkop, A. Sokolov, K. Graim, C. Funk, K. Verspoor, A. Ben-Hur *et al.*, "A large-scale evaluation of computational protein function prediction," *Nature Methods*, vol. 10, pp. 221–227, 2013.

[60] M. Kanehisa, S. Goto, Y. Sato, M. Kawashima, M. Furumichi, and M. Tanabe, "Data, information, knowledge and principle: back to metabolism in kegg," *Nucleic Acids Research*, vol. 42, no. D1, pp. D199–D205, 2014.

[61] A. Parikh, E. R. Miranda, M. Katoh-Kurasawa, D. Fuller, G. Rot, L. Zagar, T. Curk, R. Sucgang, R. Chen, B. Zupan, W. F. Loomis, A. Kuspa, and G. Shaulsky, "Conserved developmental transcriptomes in evolutionarily divergent species," *Genome Biology*, vol. 11, no. 3, p. R35, 2010.

[62] R. I. Kondor and J. D. Lafferty, "Diffusion kernels on graphs and other discrete input spaces," in *Proceedings of the 19th International Conference on Machine Learning*, 2002, pp. 315–322.

[63] G. V. Paolini, R. H. Shapland, W. P. van Hoorn, J. S. Mason, and A. L. Hopkins, "Global mapping of pharmacological space," *Nature Biotechnology*, vol. 24, no. 7, pp. 805–815, 2006.

[64] F. Iorio, R. Bosotti, E. Scacheri, V. Belcastro, P. Mithbaokar, R. Ferriero, L. Murino, R. Tagliaferri, N. Brunetti-Pierri, A. Isacchi *et al.*, "Discovery of drug mode of action and drug repositioning from transcriptional responses," *Proceedings of the National Academy of Sciences*, vol. 107, no. 33, pp. 14 621–14 626, 2010.

[65] Y. Wang, J. Xiao, T. O. Suzek, J. Zhang, J. Wang, and S. H. Bryant, "Pubchem: a public information system for analyzing bioactivities of small molecules," *Nucleic Acids Research*, vol. 37, no. suppl 2, pp. W623–W633, 2009.

[66] V. Law, C. Knox, Y. Djoumbou, T. Jewison, A. C. Guo, Y. Liu, A. Maciejewski, D. Arndt, M. Wilson, V. Neveu *et al.*, "DrugBank 4.0: shedding new light on drug metabolism," *Nucleic Acids Research*, vol. 42, no. D1, pp. D1091–D1097, 2014.

[67] J. Leskovec, D. Huttenlocher, and J. Kleinberg, "Signed networks in social media," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2010, pp. 1361–1370.

[68] R. A. Notebaart, P. R. Kensche, M. A. Huynen, B. E. Dutilh *et al.*, "Asymmetric relationships between proteins shape genome evolution," *Genome Biology*, vol. 10, no. 2, p. R19, 2009.

Marinka Žitnik received the BS degree in computer science and mathematics from University of Ljubljana, Slovenia, in 2012, where she is currently a PhD student in computer science. Her research interests include machine learning, optimization, mathematical modelling, matrix analysis and probabilistic numerics.

Blaž Zupan studied computer science at University of Ljubljana, Slovenia, and University of Houston, Texas, USA. He is Professor at University of Ljubljana, and a visiting professor at Baylor College of Medicine in Huston. His research in data mining focuses on methods for data mining and applications in bioinformatics and systems biology. He is a co-author of Orange, a Python-based and visual programming data mining suite, and several bioinformatics web applications, such as dictyExpress for gene expression analytics and GenePath for epistasis analysis.