Data-driven imputation of miscibility of aqueous solutions via graph-regularized logistic matrix factorization

Diba Behnoudfar,[†] Cory M. Simon,[*,†] and Joshua Schrier[*,‡]

[†]School of Chemical, Biological, and Environmental Engineering. Oregon State University, Corvallis, OR, USA.

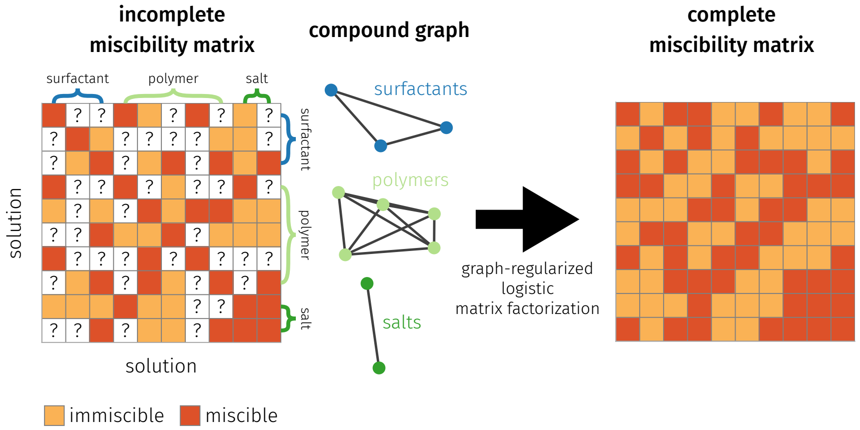
[‡]Department of Chemistry, Fordham University, The Bronx, New York 10458, USA.

E-mail: cory.simon@oregonstate.edu; jschrier@fordham.edu

Abstract

Aqueous, two-phase systems (ATPSs) may form upon mixing two solutions of independently water-soluble compounds. Many separation, purification, and extraction processes rely on ATPSs. Predicting the miscibility of solutions can accelerate and reduce the cost of the discovery of new ATPSs for these applications. Whereas previous machine learning approaches to ATPS prediction used physicochemical properties of each solute as a descriptor, in this work, we show how we can impute missing miscibility outcomes directly from an incomplete collection of pairwise miscibility experiments. We use graph-regularized logistic matrix factorization to learn a latent vector of each solution from (i) the observed entries in the pairwise miscibility matrix and (ii) a graph (nodes: solutes, edges: shared relationships) indicating the general category of the solute (i.e., polymer, surfactant, salt, protein). Using an experimental dataset of the pairwise miscibility of 68 solutions from Peacock et al. [*ACS Appl. Mater. Interfaces* 2021, 13, 9], we show that graph-regularized logistic matrix factorization more accurately predicts missing (im)miscibility outcomes of pairs of solutions than ordinary logistic matrix factorization and random forest classifiers using physicochemical features of the compounds.

TOC Graphic



Introduction

Aqueous, two-phase systems (ATPS)

Aqueous, two-phase systems (ATPS)[1–3]—also known as aqueous biphasic systems (ABS)—may form upon mixing the solutions of two independently water-soluble compounds (e.g., polymer, surfactants, or salts), resulting in a phase separation owing to incompatibility of the compounds.[4] Fig. 1 displays an example ATPS. Beijerinck observed the first ATPS, an incompatible mixture of aqueous starch and gelatin solutions, in 1896.[5] Today, ATPSs are widely used for separation, purification, and extraction of biomolecules[6] for biotechnology applications,[7] metals,[8,9] and environmental contaminants.[10] In general, ATPSs can be used as a green alternative to organic-aqueous solvent extraction.[11]

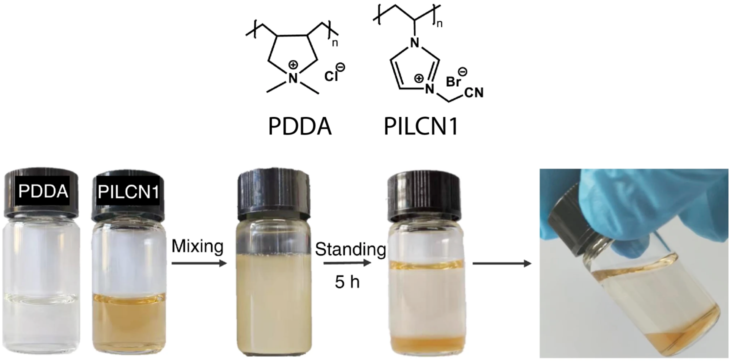


Figure 1: **An example of an aqueous, two-phase system (ATPS).** Two aqueous solutions of two polymers, PILCN1 and PDDA, are prepared, mixed, then allowed to settle for five hours at 20 °C. The mixture forms an ATPS apparent from the naked eye. PDDA = poly(dimethyl diallyl ammonium chloride) at 60 mg/mL. PILCN1 = poly(1-cyanomethyl-3-vinylimidazolium bromide) at 100 mg/mL. *Reproduced and modified from Fig. 1 of Zhang, Liu, Gong, and Zhao[12] under a Creative Commons Attribution 4.0 International License* `creativecommons.org/licenses/by/4.0/`.

Although ATPS-based separations are sufficiently simple to perform in an undergraduate teaching laboratory,[13] predicting whether two solutions will form an ATPS is challenging. The fundamental thermodynamics of ATPS are reasonably-described by Flory-Huggins theory.[4,14,15] However, accurately obtaining/predicting the solute-solute and solute-solvent interaction parameters is challenging for both experiment and atomistic simulation.[16,17] Direct molecular dynamics simulations of liquid-liquid separations are computationally expensive, predicated on an accurate force field, and challenging because there is no obvious microscopic order parameter.[18] Alternatively, machine learning can be used to predict miscibility from experimental data. One approach is to machine-learn the Flory-Huggins parameters,[19] but this assumes that Flory-Huggins captures all of the relevant thermodynamics for miscibility. Instead, one could predict miscibility directly, i.e., whether two solutions give rise to an ATPS. Peacock et al. recently published a dataset of ATPS outcomes for 2278 pairwise mixtures of 68 water-soluble compounds,[20] using a microscopy-based high-throughput assay.[21] By training a random forest model on this data, using physicochemical features of the compounds drawn from PubChem,[22] Peacock et al. achieved ∼74% accuracy in predicting the (im)miscibility of the pairs of solutions. (Some closely related, but distinct problems in machine-learning for aqueous phase thermodynamics include predicting partition coefficient of biomolecules in the two different phases of an ATPS[23–25] and predicting the phase behavior of one surfactant in water as a function of temperature and surfactant concentration.[26])

Matrix Factorization

The pairwise structure of ATPSs suggests the possibility of imputing missing or unobserved mixture miscibilities using matrix factorization. A classic example of matrix factorization for data imputation is in recommending movies to users based on a limited set of movie ratings by users.[27,28] The collection of ratings are organized into a matrix, whose rows and columns pertain to the users and movies, respectively. This matrix is only sparsely populated, and the missing entries are the ratings we wish to predict for making movie recommendations. By assuming that the movies and users can be represent by low-dimensional latent vectors that can be learned from the observed ratings, one can factorize the ratings matrix. Conceptually, the underlying latent vectors representing the movies could include dimensions indicating seriousness vs. escapist, historical vs. futuristic, romantic vs. unromantic, etc., and the latent vectors of the users represent their affinities towards those types of movies.[27,28] To learn the low-rank latent representation of the movies and users, MF needs a few example ratings; this requirement for a *new* movie or user is known as the "cold-start problem".[27]

Matrix factorization approaches have been applied to a variety of *chemical systems*, e.g., to predict gas adsorption in nanoporous materials,[29,30] diffusion coefficients[31] and activity coefficients[32–34] of binary liquids, Henry's Law coefficients,[35] the synthesis of metal oxides[36,37] and halide perovskites,[38] gas permeabilities in polymers,[39] and antiviral activities of molecules.[40] In these examples, the rows and columns represent different entities (e.g., gases and MOFs, elements and reaction conditions) like the movie setting.

Our contribution

In this paper, we demonstrate matrix factorization for ATPS predictions: imputing unobserved outcomes in a solution miscibility matrix via logistic matrix factorization (LMF).[27,41,42] An $n \times n$ *incomplete miscibility matrix* organizes pairwise miscibility outcomes of n distinct aqueous solutions; row i and column i pertain to solution i (at some concentration), and entry (i, j) contains the miscibility of solution i and j (0: forms ATPS, 1: miscible, `missing`: not observed). In contrast to the movie rating matrix, the (square) miscibility matrix is symmetric; its rows and columns pertain to the same entities (solutions). The objective is to leverage the observed values in the miscibility matrix to predict the unobserved (`missing`) ones. Rather than engineering a set of features of each solution (e.g., physicochemical features of its solute molecule), then training a machine learning model to use those features to predict the miscibility of the two solutions, LMF instead *learns* a low-dimensional latent vector representation of each solution from the observed miscibility outcomes. Then, LMF uses the learned latent representations of the solutions to make predictions (miscible or immiscible (ATPS)) for the unobserved entries. The underlying assumption is that the compound miscibility matrix exhibits a low-rank structure,[43] owing to underlying chemical/thermodynamic principles governing miscibility of the solutions and chemical similarities (e.g., functional groups in common) between the compounds.

For the miscibility matrix, the rows and columns both pertain to solutions, but the different types of solutes involved—polymer, proteins, surfactants, and salts—might admit a chemically-meaningful grouping. This suggests a modification to LMF to exploit this type of limited information about the categories of solutes by using *graph-regularized* matrix factorization (GR-LMF)[44]—also known as "manifold-regularized"[45,46] or "neighborhood-regularized"[47] MF. In this graph, the nodes represent solutes and edges joining pairs of nodes represent pairwise relationships (e.g., "both of these solutes are polymers"). The basic idea behind the graph regularization is to incorporate prior knowledge into LMF learning—that similar categories of solutes exhibit similar miscibility behavior—by promoting vector representations of solutions of similar solutes to be near each other in the latent space. (GR-MF has been applied to single-cell RNA-seq clustering[48] and predicting drug-drug, drug-target, and metabolite-disease interactions[47,49–52] and side effects of drugs.[53]) Starting from only (1) incomplete experimental observations of (im)miscibility of pairs of 68 solutions of distinct compounds from Peacock et al.[20] (2) and the rough grouping of the compounds into the categories of polymer, protein, surfactant, or salt, we show that GR-LMF learns latent representations of the solutions that give ATPS predictions on missing entries outperforming (i) ordinary LMF and (ii) a standard supervised machine learning approach using a random forest classifiers taking as input physicochemical features of the compounds in the solutions. Like all MF methods, GR-LMF requires a few initial miscibility observations of a new solution to learn an appropriate representation before a prediction can be made.

Theory: Graph-regularized matrix factorization for imputing solution miscibility

Problem setup: imputing missing entries in a compound miscibility matrix

We are interested in the pairwise miscibility of a set of n aqueous solutions at some constant temperature. Each solution consists of a distinct (water-soluble) compound dissolved in water at some compound-specific concentration. (This simplification precludes our model from needing to capture the concentration-dependence of the miscibility.) Each compound belongs to a category (e.g., polymer, salt, etc.), defined based on chemical intuition.

The miscibility matrix. Let $m_{ij} \in \{0, 1\}$ (0: immiscible and ATPS-forming; 1: miscible) be the miscibility of solution i and j. The $n \times n$ *miscibility matrix* M contains all solution miscibilities. Entry (i, j) of M is m_{ij}. Row/column i pertains to the solution of compound i. The matrix M is symmetric because $m_{ij} = m_{ji}$.

The data. Suppose we have conducted experiments where we have mixed a set of pairs of solutions $\Omega_{\text{obs}} \subset \{1, ..n\} \times \{1, ..., n\} =: \Omega_{\text{all}}$ and observed their miscibilities $\{m_{ij} : (i, j) \in \Omega_{\text{obs}}\}$. Importantly, many of the pairwise solution miscibilities have not been observed, i.e., the miscibility matrix M is *incomplete*; the entries $\Omega_{\text{all}} \setminus \Omega_{\text{obs}}$ are missing. Let $\theta := 1 - |\Omega_{\text{obs}}|/|\Omega_{\text{all}}|$ be the fraction of entries that are missing. Further, suppose each compound belongs to a known category (e.g., polymer, salt, etc.). We represent this information as a simple graph $G = (\mathcal{V}, \mathcal{E})$. The nodes $\mathcal{V} = \{v_1, ..., v_n\}$ in the graph represent the compounds. The presence of an edge $\{v_i, v_j\} \in \mathcal{E}$ joining compounds v_i and v_j indicates that these two compounds belong to the same category; the absence of an edge indicates the two compounds do not belong to the same category. In other words, the edges in the compound graph G express the relationships between the types of compounds.

Objective. Our objective is to impute the missing entries of the miscibility matrix M—i.e., to predict the values of the unobserved solution miscibilities $\{m_{ij} : (i,j) \in \Omega_{\text{all}} \setminus \Omega_{\text{obs}}\}$. To do so, we take a data-driven approach and leverage (i) the observed solution miscibilities $\{m_{ij} : (i,j) \in \Omega_{\text{obs}}\}$ and (ii) the graph G indicating the categories to which the compounds belong. Specifically, we will seek a logistic-transformed low-rank factorization of the miscibility matrix M regularized by the compound graph G that specifies relationships between its rows/columns.

The solution miscibility model

We propose a probabilistic model for the miscibility of any pair of solutions. The model assumes that each solution i may be represented as a low-dimensional *latent vector* $c_i \in \mathbb{R}^k$. Conceptually, the vector c_i encodes the chemical features of compound i relevant to its compatibility with the other compounds, and it contains information about the concentration of the compound in the specific solution under consideration. (In principle, solutions of the same compound with different concentrations would occupy different rows and columns of M and correspond to different c_i vectors, although one would expect them to point in the same direction. In the Peacock et al. dataset, each compound is present only at a single concentration, and so in the present analysis there is a bijection between solution vectors and compounds.)

Given the latent vectors $\{c_1, ..., c_n\}$ and a bias $b \in \mathbb{R}$, our model for the probability that solution i and solution j are miscible is a conditional probability:

$$\pi(M_{ij} = 1 \mid c_1, ..., c_n, b) = \sigma(c_i \cdot c_j + b). \tag{1}$$

The monotonic sigmoid function $\sigma(x) = (1 + e^{-x})^{-1}$ squashes its input x to its range $(0, 1)$

for interpretation as a probability.

The model in eqn. 1 admits a geometric interpretation. Note, $c_i \cdot c_j = \|c_i\| \|c_j\| \cos\theta_{ij}$, where θ_{ij} is the angle between solution vectors c_i and c_j in the latent space. Consequently, pairs of solutions whose vectors c_i and c_j point in roughly the same (opposite) direction tend to be miscible (immiscible), i.e., the location of the solution vectors $\{c_i\}_{i=1}^n$ in latent space relative to each other dictate our predictions about their pairwise miscibilities. Changing the magnitude, but not the direction, of a c_i vector loosely corresponds to increasing the likelihood of the miscibility of solution i with the others, as it increases the magnitude of the input to $\sigma(x)$, and in turn makes the output of $\sigma(x)$ closer to zero or one. A positive (negative) bias b shifts the predictions towards miscible (immiscible).

The matrix perspective. The model in eqn. 1 approximates the miscibility matrix as:

$$M \approx \sigma.\left(\begin{bmatrix} — & c_1^\intercal & — \\ & \vdots & \\ — & c_n^\intercal & — \end{bmatrix} \begin{bmatrix} | & & | \\ c_1 & \dots & c_n \\ | & & | \end{bmatrix} + \begin{bmatrix} b & \cdots & b \\ \vdots & \ddots & \vdots \\ b & \cdots & b \end{bmatrix} \right) \tag{2}$$

where $\sigma.(x)$ denotes element-wise operation of the sigmoid function. That is, we approximate the miscibibility matrix as a sigmoid-transformed low-rank (rank $\leq k$) matrix factorized as $C^\intercal C$, where the $k \times n$ matrix C contains the latent solution vector i in its column i, plus a constant bias matrix. Thus, our miscibility model falls in the "matrix factorization"/"low-rank matrix model" category of machine learning algorithms,[27,41] more specifically, *logistic matrix factorization*.[42]

Training vs. imputation stages. In the *training* stage, we *learn* the latent vectors of the solutions $\{c_i\}_{i=1}^n$ and the bias b from the *observed* miscibilities $\{m_{ij} : (i, j) \in \Omega_{\text{obs}}\}$ and the compound graph G. The dimension k of the latent solution vector space is a hyperparameter specified before training.

In the *imputation* stage, we use the learned $\{c_i\}_{i=1}^n$ and b and the model in eqn. 1 to predict the *unobserved* solution miscibilities $\{m_{ij} : (i,j) \in \Omega_{\text{all}} \setminus \Omega_{\text{obs}}\}$. We employ a threshold t (a hyperparameter) for the classification rule, i.e., the model prediction for the miscibility of compound solutions i and j is:

$$\hat{m}_{ij} := \mathcal{I}\left(\sigma(c_i \cdot c_j + b) > t\right) \tag{3}$$

with $\mathcal{I} : \mathbb{R} \rightarrow \{0,1\}$ the indicator function.

Training

To train the model in eqn. 1, we tune the latent vectors $\{c_i\}_{i=1}^n$ and bias b to minimize a loss function posed over the observed entries of the miscibility matrix, $\{m_{ij} : (i,j) \in \Omega_{\text{obs}}\}$, and the compound graph G:

$$\begin{aligned}\ell(c_1, ..., c_n, b) = & -\frac{1}{2} \sum_{(i,j) \in \Omega_{\text{obs}}} \left[m_{ij} \log\left(\sigma(c_i \cdot c_j + b)\right) + (1 - m_{ij}) \log\left(1 - \sigma(c_i \cdot c_j + b)\right)\right] \\ & + \lambda \sum_{i=1}^{n} ||c_i||^2 + \gamma \sum_{\{i,j\} : \{v_i, v_j\} \in \mathcal{E}} ||c_i - c_j||^2 \end{aligned} \tag{4}$$

The loss function comprises three terms:

- cross-entropy loss. This term expresses our desire for the model predictions via eqn. 1 to match the observed miscibility outcomes in the training data $\{m_{ij} : (i,j) \in \Omega_{\text{obs}}\}$. If a pair of solutions i and j are observed to be miscible, i.e. $m_{ij} = 1$, (immiscible, i.e. $m_{ij} = 0$) then this term penalizes output of the model in eqn. 1 straying far below 1.0 (above 0.0). From a probabilistic perspective, the cross-entropy loss follows from minimizing the negative log-likelihood of the outcomes in the training data under the probabilistic (likelihood) model in eqn. 1. The factor of 1/2 corrects for overcounting if one considers (i, j) as ordered pairs, but in a computational implementation, it can be eliminated by considering the set of (unordered) sets $\{i, j\}$, i.e., all unique pairs.

- 2-norm regularization. To prevent overfitting, this term penalizes latent compound solution vectors with a large magnitude. The hyperparameter $\lambda > 0$ modulates the strength of the 2-norm regularization.

- graph-based regularization. To promote latent compound solution vectors pertaining to compounds of the same category to lie closeby in latent space, this term penalizes pairs of compound solution vectors *that belong to the same category* (hence, an edge exists in the compound graph G) from being distal in latent space. This term is a means of building our prior knowledge "compounds of the same category are more likely to have similar miscibility behavior" into the model. The hyperparameter $\gamma > 0$ modulates the strength of the graph-based regularization.

Note, the loss function does not concern the miscibility outcome of a solution with itself (a solution is always miscible with itself, so $m_{ii} = 1$).

The loss function ℓ in eqn. 4 can be minimized by gradient descent. One epoch of gradient descent comprises using the gradients to update b and $\{c_i\}_{i=1}^{n}$ in shuffled order. The relevant gradients are, similar to those in logistic regression:[54]

$$\nabla_{c_i}\ell = \frac{1}{2}\sum_{j:(i,j)\in\Omega_{\text{obs}}}[\sigma(c_i \cdot c_j + b) - m_{ij}]c_j + 2\lambda c_i + 2\gamma \sum_{j:\{v_i,v_j\}\in\mathcal{E}}(c_i - c_j) \tag{5}$$

$$\nabla_b\ell = \frac{1}{2}\sum_{(i,j)\in\Omega_{\text{obs}}}(\sigma(c_i \cdot c_j + b) - m_{ij}) \tag{6}$$

The model in eqn. 1 trained with the loss function in eqn. 4 comprise graph-regularized matrix factorization (GR-LMF). Note, the compound graph G is only used in the training stage, not the imputation stage. See Ref.[44] for more details on GR-LMF.

Hyperparameters

Our GR-LMF model contains four hyperparameters: (1) k: the dimension of the latent solution space, (2) λ: the 2-norm regularization parameter, (3) γ: the graph-based regularization parameter, and (4) t: the classification threshold that balances false positives and false negatives. We optimize k, λ, and γ via a random search paired with 3-folds class-stratified cross-validation of the observed entries $\{m_{ij} : (i, j) \in \Omega_{\text{obs}}\}$ and select the values that give the largest mean balanced accuracy for imputing the validation entries. We optimize t via grid search based on balanced accuracy over the training entries.

Imputation performance metric: balanced accuracy

As a primary metric of the imputation performance of a GR-LMF model appropriate for balanced classes, we employ the *balanced accuracy*: the average of (i) the recall of immiscible solution pairs: among those pairs of solutions that are truly immiscible, the fraction that the model correctly predicts to be immiscible and (ii) the recall of miscible solution pairs: among those pairs of solutions that are truly miscible, the fraction that the model correctly predicts to be miscible.

Computational Methods

Data. To test our approach, we used the experimental data from Peacock et al.[20] This data set provides (i) the complete miscibility matrix M_{complete} containing all pairwise mixing outcomes of $n = 68$ aqueous solutions of distinct compounds at compound-specific concentrations (1559 miscible and 719 immiscible pairs) near room temperature and (ii) for the compound graph G, the category to which each compound belongs: 46 polymers, 11 surfactants, 8 proteins, and 3 salts. See Fig. S1 for the complete miscibility matrix M_{complete} for all $n(n-1) = 2278$ pairs of solutions. For the purpose of demonstrating missing value imputation, we generate, from M_{complete}, a simulated incomplete miscibility matrix $M^{(\theta)}$; decoration with θ indicates the fraction of missing entries in the matrix.

Simulating incomplete experiments to construct an incomplete miscibility matrix. To simulate hypothetical incomplete experimentation, we construct an $n \times n$, symmetric incomplete miscibility matrix $M^{(\theta)}$, where θ is the fraction of missing entries in the matrix, as follows. First, we randomly partition the complete set of $n(n-1)$ pairs of miscibility observations $\{(i, j, m_{ij})\}$ of Peacock et al.[20] into an *observed* set $(i, j) \in \Omega_{\text{obs}}$ (fraction: $1 - \theta$) and *unobserved* set $(i, j) \in \Omega_{\text{all}} \setminus \Omega_{\text{obs}}$ (fraction: θ). We stratify the split according to the miscibility outcome to preserve the class distribution. We then construct the incomplete miscibility matrix $M^{(\theta)}$ by placing the miscibility outcomes belonging to the observed set in the appropriate entries; the remaining entries are set as `missing`. The unobserved set serves as hold-out test data—not used for training the GR-LMF model, nor for tuning its hyperparameters.

Note, the compound graph G is static. Regardless of θ and which entries of the miscibility matrix $M^{(\theta)}$ are missing, G contains n nodes, one for each compound, and any two compounds are joined by an edge iff they are of the same category (polymer, surfactant, protein, or salt).

GR-LMF training and hyperparameter tuning. We examined both ordinary gradient descent and Adam[55] to minimize the loss function. Preliminary investigations showed that ordinary gradient descent was sensitive to the learning rate, resulted in oscillations in the loss for large learning rates, and required ∼350 epochs to converge. In contrast, Adam was insensitive to the choice of learning rate and required many fewer than 250 epochs for convergence, and therefore was a superior choice. (See Fig. S2).

We determined optimal GR-LMF hyperparameters k, γ, and λ through random search over 25 hyperparameter sets independently drawn from a uniform distribution over $\{2, 3\}$, $[0, 0.1]$ and $[0, 1]$, respectively. (Our choice to limit the number of latent vector dimensions to two or three was motivated by: (i) the ability to visualize the vectors and (ii) precedent from Hansen solubility theory,[56] which describes solute-solvent interactions in terms of a 3-dimensional representation.) We make a 3-fold split of the observed entries in $M^{(\theta)}$ for cross-validation. Then, we select the hyperparameter set among the 25 that, when a GR-LMF model with these hyperparameters is trained on two of the folds of the observed entries, it yields the largest (mean over three models) balanced accuracy for imputing miscibilities in the remaining fold serving as the validation set. The classification threshold t is tuned based on grid search, using balanced accuracy over the train set.

Finally, once the optimal hyperparameters $k^\star$, $\gamma^\star$, and $\lambda^\star$ are found, we train a new GR-LMF model—the *deployment* model—based on hyperparameters $k^\star$, $\gamma^\star$, and $\lambda^\star$ on all of the observed entries in the miscibility matrix $M^{(\theta)}$.

Baseline models.

Compound-category-informed guessing. For each ordered pair of compound categories $(x, x') \in \mathcal{C} \times \mathcal{C}$ with $\mathcal{C} = \{\text{protein}, \text{polymer}, \text{surfactant}, \text{salt}\}$, we compute the fraction of the observed entries in the training data that are miscible, $\theta_{\text{miscible}}(x, x')$. Then, given an ordered pair of solution compounds belonging to categories $(x_{\text{new}}, x'_{\text{new}})$, whose miscibility is `missing`, we assign a miscible label with probability $\theta_{\text{miscible}}(x_{\text{new}}, x'_{\text{new}})$ and immiscible label with probability $1 - \theta_{\text{miscible}}(x_{\text{new}}, x'_{\text{new}})$.

Ordinary matrix factorization (MF). To develop an ordinary LMF deployment model, we employ the same procedure as for GR-LMF but with $\gamma := 0$ to turn off the graph-regularization term in the loss function in eqn. 4.

Random forest (RF). Conceptually, the input to the random forest (RF) classifier is a vector representing a pair of solutions of compounds, and the output is a prediction of either miscible (1) or immiscible (0). As a vector representation of a given solution of a compound, we use (i) the physicochemical features compiled from PubChem[22] by Peacock et al.:[20] monomer and polymer molecular weight, the log of the predicted octanol-water partition coefficient, hydrogen bond donor and acceptor counts, and complexity, (ii) the concentration of the compound in the solution, and (iii) a one-hot encoding of the category (polymer, protein, surfactant, salt). Note, since the compounds could not be fully annotated with group (i) features, we used the imputed features based on an 8-nearest-neighbors algorithm from Peacock et al.[20] This gives a length-11 feature vector for each solution. To represent a *pair* of solutions for input to the RF, then, we concatenate the vector representations of the two compounds, giving a 22-dimensional input. We encourage the predictions of the RF to be invariant to the order in which the pair of solutions are presented through data augmentation: we repeat each training data point twice, reordering the solution vectors in the input. For a fair comparison with GR-LMF, we tune the classification threshold t for the RF (used to map the set of binary votes by the trees to a single binary output) to maximize balanced accuracy on the observed miscibilities. We used the RF implementation in `scikit-learn` version 1.2.2 with default settings (100 trees grown to the maximum depth, Gini impurity for splits, five randomly-selected features as candidates for each split).[57]

Code and data. All `Julia`[58] code to reproduce our plots in `Makie.jl`[59] is available at `github.com/SimonEnsemble/miscibility_matrix_factorization` *Note to reviewers: we will also archive this to Zenodo after the paper is accepted.* The raw miscibility data is also on our Github repo in comma-separated-value (`.csv`) format, which we reformatted from the supplementary information of Peacock et al.[20]

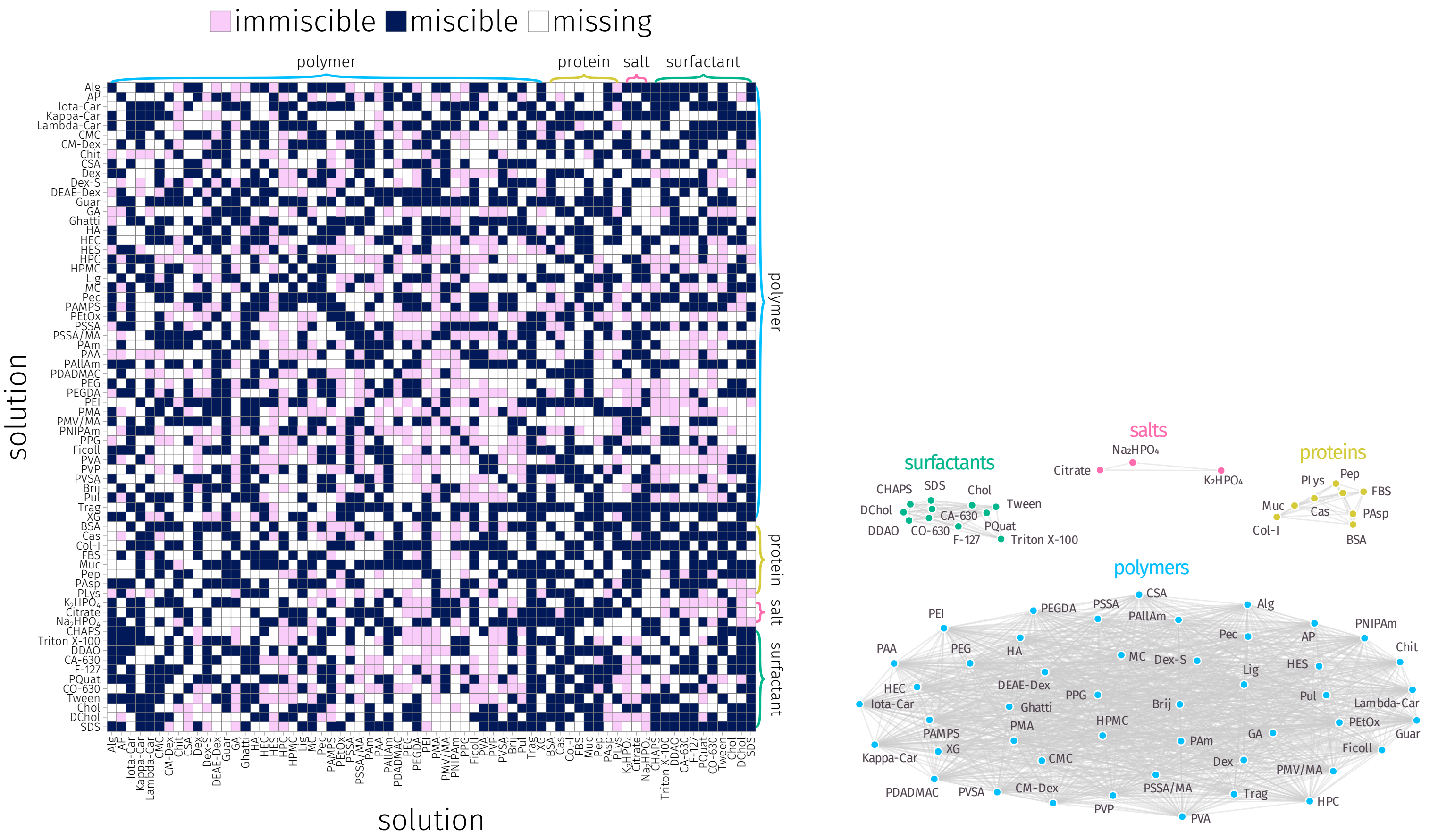
Results and Discussion

A GR-LMF case study: $\theta = 0.4$

For a demonstration and to gather insights, we present a case study where we train, test, and analyze a GR-LMF model on one instance of an incomplete miscibility matrix $M^{(\theta)}$ with 40% missing values ($\theta = 0.4$).

The incomplete miscibility matrix $M^{(0.4)}$ is shown in Fig. 2a. The rows and columns are labeled by the compounds they represent and sorted by compound category (polymer, protein, salt, surfactant). Each entry is colored according to the miscibility outcome of that pair of compound solutions: immiscible (0), miscible (1), or missing. Note, the matrix is symmetric, and miscibility is more common than immiscibility.

The compound graph G is shown in Fig. 2b. Each node is labeled by the compound it represents. Any pair of nodes (compounds) of the same category (polymer, protein, salt, surfactant) share a color and are joined by an edge. The induced subgraph of nodes belonging to a certain category is complete. The edges in the compound graph G indicate relationships between the compounds and thus between the rows/columns in the miscibility matrix $M^{(\theta)}$.



(a)

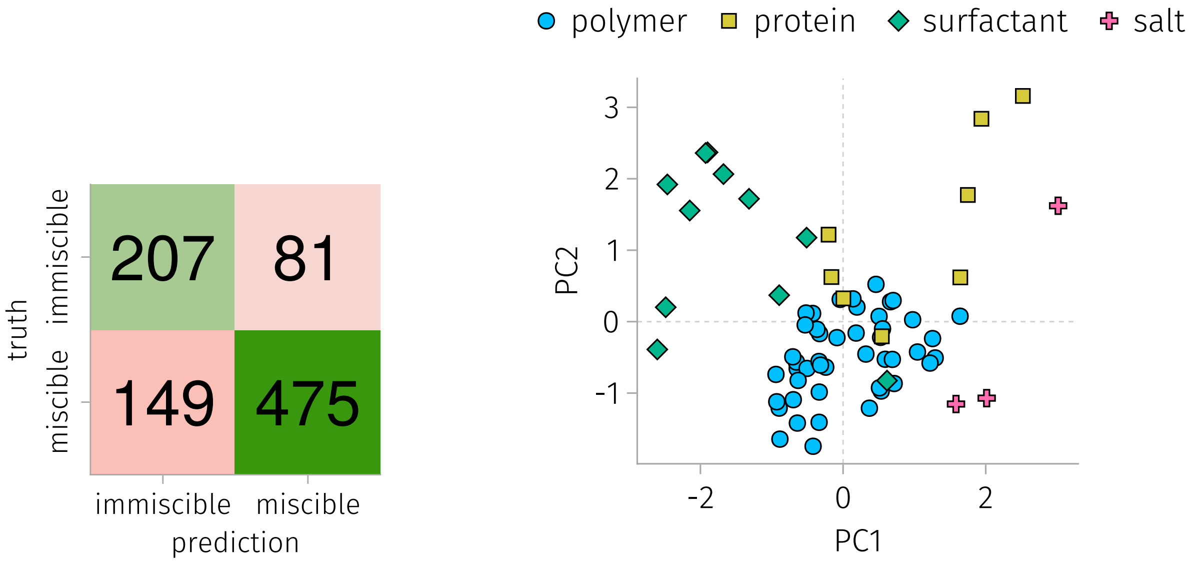
(b)

Figure 2: **Data for the miscibility matrix completion problem.** (a) An incomplete miscibility matrix $M^{(\theta)}$ with a fraction $\theta = 0.4$ missing entries. Entry (i, j) indicates the miscibility of a pair of solutions of compounds i and j. (b) The compound graph G indicating relationships (edges) between the compounds (nodes).

We trained and tuned the hyperparameters of a GR-LMF on the observed entries in the incomplete miscibility matrix $M^{(0.4)}$ and compound graph G. The optimal hyperparameters were $k^\star = 3$, $\gamma^\star = 0.03$, and $\lambda^\star = 0.15$.

We evaluated the imputation performance of the deployment GR-LMF model by comparing the true miscibility outcomes, which were held out as missing during training and hyperparameter tuning, to the imputed values by the deployment GR-LMF model via eqn. 1. The model does not perfectly discriminate between the two outcomes (Fig. S3), so the choice of threshold will depend on the evaluation metric. The choice of balanced accuracy is a compromise between the recall of the two classes, which is appropriate because of the class imbalance in this dataset. The confusion matrix over the `missing` entries is shown in Fig. 3a. The GR-LMF achieved a balanced accuracy of 74%.

The relative location of the learned vectors in latent space is indicative of the predicted miscibility of the solutions they represent, so visualizing them can help us interpret the GR-LMF model. Fig. 3b shows the first two principal components of the latent solution space, where each point represents a compound. (Alternatively, Fig. S5 shows the latent vectors in 3D, but PCA is more effective as a static illustration.) The shapes/colors of the points indicate the categories to which each compound belong. The latent vectors are clustered by category, in part due to the graph-regularization ($\gamma > 0$). Without the graph-regularization term ($\gamma := 0$), the latent vectors of the solutions still display some clustering according to category (see Fig. S6), but it is weaker than if the prior information about solute categories is included through G. The relative orientation of the latent vectors also describes general trends in miscibility. For example, in Fig. 3b, the latent vectors of the salts (red) tend to point in the opposite direction of those of the surfactants (green). Correspondingly, surfactant-salt mixtures in the data set are the most likely to be immiscible compared to other pairs of distinct compound categories (see Fig. S7). Similarly, most protein vectors point in a similar direction, and thus tend to be miscible, at least within this dataset. These trends are also illustrated in the visualization of the matrix C (eqn. 2) whose columns contain the c_i vectors (see Fig. S4).



(a) (b)

Figure 3: **GR-LMF results for $M^{(\theta=0.4)}$.** (a) Confusion matrix for imputation of unobserved test entries of the miscibility matrix. (b) The first two principal components of the learned latent solution vectors, $\{c_i\}$. This GR-LMF model was trained on the observed data in Fig. 2 with optimized hyperparameters $k^\star = 3$, $\lambda^\star = 0.15$, $\gamma^\star = 0.03$.

Imputation performance comparisons

We investigate how the performance of GR-LMF for miscibility matrix completion compares against the baseline LMF and RF methods, as a function of experimental incompleteness, θ. Fig. 4 displays the distribution of balanced accuracy of GR-LMF, LMF, RF, and compound-category-informed guessing for imputing missing miscibilities over 10 realizations of incomplete miscibility matrices $M^{(\theta)}$'s for different missing fractions of miscibility experiments $\theta \in \{0.2, 0.5, 0.8\}$.

How does performance vary over the random process of incomplete experimentation, i.e., over different `missing` value patterns in the incomplete miscibility matrix $M^{(\theta)}$? We conducted 10 simulations of incomplete experimentation to generate 10 realizations of incomplete miscibility matrices $M^{(\theta)}$, then trained and hyperparameter-tuned the models on each $M^{(\theta)}$ (and G, which is static). The box plots in Fig. 4 show the distribution of balanced accuracy of each model over these 10 realizations. For each θ, the standard deviation of the balanced accuracy of the GR-LMF is less than 2.5%.

How does performance depend on experimental incompleteness, i.e., on the fraction θ of entries `missing` in the miscibility matrix $M^{(\theta)}$? We trained and hyperparameter-tuned the models for three different levels of matrix incompleteness, $\theta \in \{0.2, 0.5, 0.8\}$ (10 runs each). Intuitively, the imputation accuracy for each model diminishes as the missing fraction θ increases. GR-LMF improves upon LMF ($\gamma := 0$) dramatically when the matrix is very incomplete ($\theta = 0.8$). This shows that the information provided by the compound graph G is beneficial for the imputation task.

How does GR-LMF compare to the traditional supervised learning approach of predicting miscibility of a pair of solutions from hand-engineered features of the compounds? In all cases, GR-LMF outperforms the random forest classifier, regardless of θ; the gap in balanced accuracy is more prominent at the largest fraction of missing entries $\theta = 0.8$. This seems unexpected, because one would think that the input features to the RF encode meaningful physicochemical aspects of the solutes pertinent to miscibility, and especially relevant when data is scarce. The RF has 22 input features, but even in the case of $\theta = 0.8$, has access to 900 examples, so its poor performance is not likely to be due to overfitting. This implies that GR-LMF learns latent representations of the compound solutions that better describe the pairwise miscibility than the physicochemical features provided to the RF. That even ordinary LMF outperforms RF when the matrix is more complete underscores this.

How do the different models perform for miscibility imputation according to different metrics? Using the hyperparameters determined by cross-validation on balanced accuracy, we assessed each model type on accuracy, precision, recall, and F1 score (harmonic mean of precision and recall). (See Fig. S8.) For each experimental incompleteness, GR-LMF still outperforms or is comparable to LMF according to all metrics. GR-LMF also outperforms RF on precision for each value of experimental incompleteness. However, RF outperforms GR-LMF according to the F1 score and accuracy. This illustrates the general quality of GR-LMF across other performance metrics, despite not explicitly tuning for those metrics.

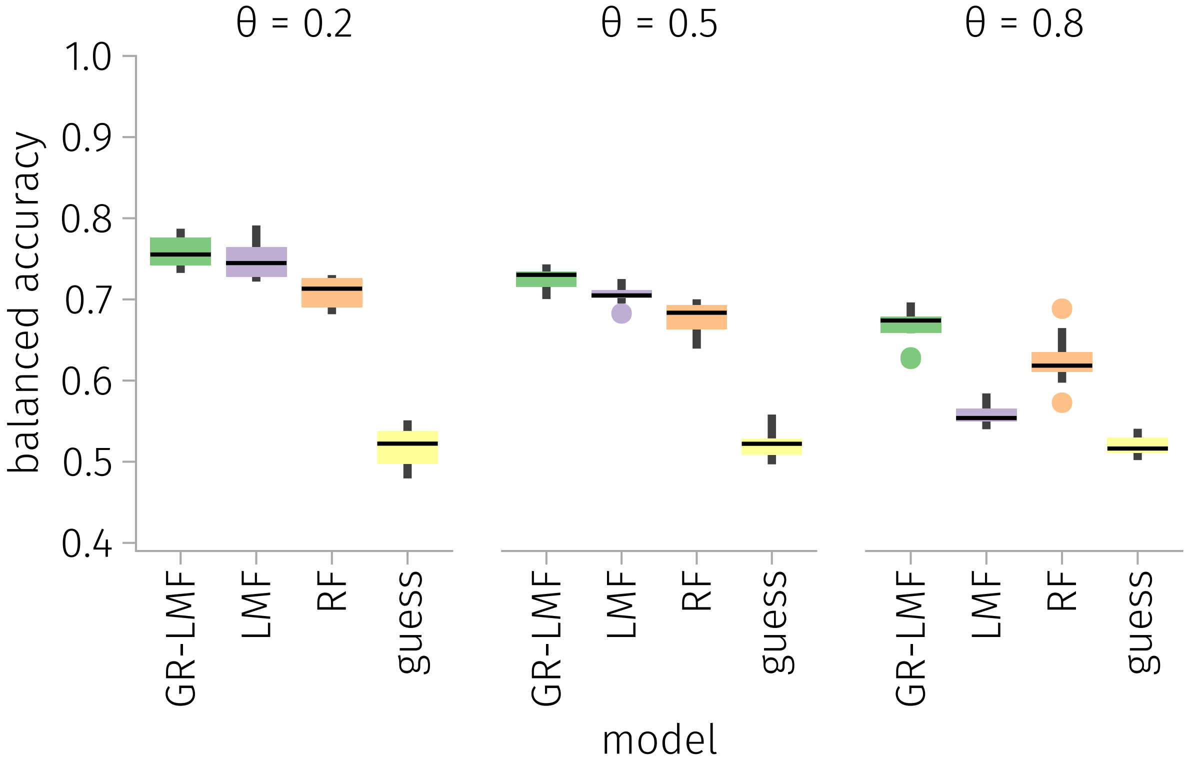


Figure 4: **Performance of graph-regularized matrix factorization (GR-LMF), ordinary matrix factorization (LMF), random forest (RF) classifier, and compound-category-informed guessing.** Box plots show the distribution of balanced accuracy for missing miscibility imputation (the hold-out test set) over 10 realizations of the incomplete miscibility matrix (i.e., different `missing` value patterns). The three panels correspond to different fractions of `missing` values, θ, in the miscibility matrix.

Strengths and Weaknesses of GR-LMF

Beneficially, GR-LMF does not require any feature engineering of the solutes, except for assigning them to general categories. This is advantageous because feature engineering of the solutes is difficult: it requires prior physical insight about solute features that are related to miscibility and the measurement or computation of those features for a diverse set of solutes. As we show above, the latent vector representation of the solution learned by GR-LMF directly from incomplete experimental miscibility observations is superior at capturing the underlying physicochemical features of the solutes that determine pairwise miscibility, even when most of the pairs of mixtures have not been observed.

However, *learning* as opposed to engineering features of the solutes can also be a weakness of MF methods, known as the "cold start" problem. Because the latent solution vector is learned from miscibility experiments involving that solution, the model cannot make predictions on a new compound solution whose miscibility with other solutions in the data set has never been observed before. However, as we showed in the previous section, GR-LMF in particular can perform well at large θ, so not many examples are needed.

Another weakness of our GR-LMF model is that it does not trivially generalize to capture the temperature- or concentration-dependence of miscibility. (In contrast, with RF one would simply add an additional concentration or temperature feature to the input and retrain the model.) In principle, GR-LMF can be modified to capture temperature- and concentration-dependence of miscibility by (1) endowing solutions of the same solute at different concentration and temperature with distinct solution vectors and (2) modifying the graph-regularization term in the loss to keep solution vectors that only vary in concentration or temperature nearby in latent space. Alternatively, matrix-factorization could be used to determine parameters in a physically-motivated equation which captures the concentration and temperature dependence, similar to Damay et al.'s approach for predicting temperature-dependent activity coefficients.[33]

Conclusion

In this work, we described and demonstrated a data-driven method, graph-regularized matrix factorization (GR-LMF), to impute missing entries in a pairwise miscibility matrix (rows/columns = solutions of a compound, entries = miscibility). As opposed to hand-engineering feature vectors of the solutions (which may be difficult), GR-LMF learns latent vector representations of each solution from the observed entries in the miscibility matrix and a compound graph indicating the categories to which each compound belongs. These latent vectors then may be used to impute the missing entries in the incomplete miscibility matrix, which in principle eliminates the need for a chemist to perform all of the pairwise experiments. GR-LMF outperforms both ordinary LMF and a random forest classifier that predicts miscibility based on physicochemical features of the compounds in the solutions. Unsurprisingly, the performance of GR-LMF diminishes with the fraction of missing entries in the miscibility matrix, but even when 80% of the data is missing it outperforms the competing imputation methods.

Our machine learning study was enabled by the open miscibility data set of Peacock et al.[20] To both improve and expand the capabilities of data-driven models for ATPS prediction, we encourage the generation and curation of large, FAIR (findable, accessible, interoperable, and reusable[60]) miscibility data sets. Though subject to error, natural language processing could also be used to mine existing miscibility data from the scientific literature[61] (to our knowledge, only one other sizeable miscibility data set of 34 polymers and surfactants has been generated[62]). On the other hand, the advantage of using a single dataset is that it avoids inter-lab variations, such as variations in purity or protocol, that might make the data less useful for machine learning.[63] The relative simple nature of the experimental process (mixing aqueous solutions) and characterization (via microscopic image analysis[21]) lends itself to laboratory automation for generating a large dataset.

Our ideas for extensions and future directions for miscibility matrix completion: (1) Extend our model in eqn. 1 to include the dependence of miscibility on temperature and concentration of the solute; this effort is predicated on the availability of miscibility data at different temperatures and containing solutions of each compound at different concentrations. (2) Leverage Bayesian probabilistic matrix factorization[32,64] to develop a GR-LMF model that quantifies uncertainty in its predictions. (3) Use the uncertainty quantification to guide the selection of miscibility experiments that provide the most information for improving imputation.[65] (4) Impose graph-regularization through a graph with weighted edges, whose weights are determined by a kernel that continuously quantifies the similarity between the compounds. (5) Extend to ternary and higher-component mixtures—also predicated upon the availability of higher-component miscibility data; this corresponds to a known problem of tensor decomposition.[66] (6) Investigate the effect that experimental selection bias[67,68] may have on the performance of miscibility matrix factorization.

Acknowledgement

The authors acknowledge support by the National Science Foundation under Grants No. CBET-1920945, PHY-2226511, and CNS-2018427.

Data Availability

All of the data and code needed for this work is available on Github at `github.com/SimonEnsemble/miscibility_matrix_factorization` and archived on Zenodo at DOI: *ASSIGNED AFTER ACCEPTANCE*

Supporting Information Available

The Supporting Information is available free of charge at `pubs.acs.org/PENDING`

- Complete miscibility matrix, example loss function optimizations, distribution of predictions, visualization and 3D plots of the learned latent vectors, visualization of the latent space with $\gamma = 0$, fraction of immiscible solutions by category, F1, accuracy, precision, and recall performance metrics for the models. (PDF)

Author Information

Corresponding Authors

Cory M. Simon

Email: Cory.Simon@oregonstate.edu

ORCID: 0000-0002-8181-9178

Joshua Schrier

Email: jschrier@fordham.edu

ORCID: 0000-0002-2071-1657

Authors

Diba Behnoudfar

Notes

The authors have no competing financial interests to declare.

References

(1) Iqbal, M.; Tao, Y.; Xie, S.; Zhu, Y.; Chen, D.; Wang, X.; Huang, L.; Peng, D.; Sattar, A.; Shabbir, M. A. B. et al. Aqueous two-phase system (ATPS): An overview and advances in its applications. *Biol. Proced. Online* **2016**, *18*, 18, DOI: `10.1186/s12575-016-0048-8`.

(2) Chao, Y.; Shum, H. C. Emerging aqueous two-phase systems: From fundamentals of interfaces to biomedical applications. *Chem. Soc. Rev.* **2020**, *49*, 114–142, DOI: `10.1039/C9CS00466A`.

(3) Hatti-Kaul, R. Aqueous two-phase systems: A general overview. *Mol. Biotechnol.* **2001**, *19*, 269–277, DOI: `10.1385/MB:19:3:269`.

(4) Gustafsson, Å.; Wennerström, H.; Tjerneld, F. The nature of phase separation in aqueous two-polymer systems. *Polymer* **1986**, *27*, 1768–1770, DOI: `10.1016/0032-3861(86)90274-0`.

(5) Albertsson, P.-Å. *Partitioning in Aqueous Two-Phase System*; Elsevier, 1985; pp 1–10, DOI: `10.1016/b978-0-12-733860-6.50008-6`.

(6) Grilo, A. L.; Aires-Barros, M. R.; Azevedo, A. M. Partitioning in Aqueous two-phase systems: Fundamentals, applications and trends. *Sep. Purif. Rev.* **2014**, *45*, 68–80, DOI: `10.1080/15422119.2014.983128`.

(7) Teixeira, A. G.; Agarwal, R.; Ko, K. R.; Grant-Burt, J.; Leung, B. M.; Frampton, J. P. Emerging biotechnology applications of aqueous two-phase systems. *Adv. Healthcare Mater.* **2017**, *7*, 1701036, DOI: `10.1002/adhm.201701036`.

(8) Sun, P.; Huang, K.; Song, W.; Gao, Z.; Liu, H. Separation of rare earths from the transition metals using a novel ionic-liquid-based aqueous two-phase system: Toward

green and efficient recycling of rare earths from the NdFeB magnets. *Ind. Eng. Chem. Res.* **2018**, *57*, 16934–16943, DOI: `10.1021/acs.iecr.8b04549`.

(9) Sunder, G. S. S.; Adhikari, S.; Rohanifar, A.; Poudel, A.; Kirchhoff, J. R. Evolution of environmentally friendly strategies for metal extraction. *Separations* **2020**, *7*, 4, DOI: `10.3390/separations7010004`.

(10) Willauer, H. D.; Huddleston, J. G.; Griffin, S. T.; Rogers, R. D. Partitioning of aromatic molecules in aqueous biphasic systems. *Sep. Sci. Technol.* **1999**, *34*, 1069–1090, DOI: `10.1080/01496399908951081`.

(11) Assis, R. C.; Mageste, A. B.; de Lemos, L. R.; Orlando, R. M.; Rodrigues, G. D. Application of aqueous two-phase system for selective extraction and clean-up of emerging contaminants from aqueous matrices. *Talanta* **2021**, *223*, 121697, DOI: `10.1016/j.talanta.2020.121697`.

(12) Zhang, C.; Liu, X.; Gong, J.; Zhao, Q. Liquid sculpture and curing of bio-inspired polyelectrolyte aqueous two-phase systems. *Nature Commun.* **2023**, *14*, 2456, DOI: `10.1038/s41467-023-38236-8`.

(13) Giuliano, K. A. Aqueous two-phase protein affinity partitioning: A laboratory demonstration of a biotechnological process. *J. Chem. Educ.* **1994**, *71*, 590, DOI: `10.1021/ed071p590`.

(14) Scott, R. L. The thermodynamics of high polymer solutions. V. Phase equilibria in the ternary system: Polymer 1—polymer 2—solvent. *J. Chem. Phys.* **1949**, *17*, 279–284, DOI: `10.1063/1.1747239`.

(15) Johansson, H.-O.; Karlström, G.; Tjerneld, F.; Haynes, C. A. Driving forces for phase separation and partitioning in aqueous two-phase systems. *J. Chromatog. B: Biomed. Sci. Appl.* **1998**, *711*, 3–17, DOI: `10.1016/S0378-4347(97)00585-9`.

(16) Shetty, S.; Gomez, E. D.; Milner, S. T. Predicting χ of polymer blends using atomistic morphing simulations. *Macromolecules* **2021**, *54*, 10447–10455, DOI: `10.1021/acs.macromol.1c01550`.

(17) Zhang, W.; Gomez, E. D.; Milner, S. T. Predicting Flory-Huggins χ from simulations. *Phys. Rev. Lett.* **2017**, *119*, 017801, DOI: `10.1103/PhysRevLett.119.017801`.

(18) Farshad, M.; DelloStritto, M. J.; Suma, A.; Carnevale, V. Detecting liquid–liquid phase separations using molecular dynamics simulations and spectral clustering. *J. Phys. Chem. B* **2023**, *127*, 3682–3689, DOI: `10.1021/acs.jpcb.3c00805`.

(19) Masullo, F.; Beldengrün, Y.; Miras, J.; Mackie, A. D.; Esquena, J.; Avalos, J. B. Phase behavior of gelatin/maltodextrin aqueous mixtures studied from a combined experimental and theoretical approach. *Fluid Phase Equilibria* **2020**, *524*, 112675, DOI: `10.1016/j.fluid.2020.112675`.

(20) Peacock, C. J.; Lamont, C.; Sheen, D. A.; Shen, V. K.; Kreplak, L.; Frampton, J. P. Predicting the mixing behavior of aqueous solutions using a machine learning framework. *ACS Appl. Mater. Interf.* **2021**, *13*, 11449–11460, DOI: `10.1021/acsami.0c21036`.

(21) Ruthven, M.; Ko, K. R.; Agarwal, R.; Frampton, J. P. Microscopic evaluation of aqueous two-phase system emulsion characteristics enables rapid determination of critical polymer concentrations for solution micropatterning. *The Analyst* **2017**, *142*, 1938–1945, DOI: `10.1039/c7an00255f`.

(22) Kim, S.; Chen, J.; Cheng, T.; Gindulyte, A.; He, J.; He, S.; Li, Q.; Shoemaker, B. A.; Thiessen, P. A.; Yu, B. et al. PubChem 2023 update. *Nucleic Acids Res.* **2022**, *51*, D1373–D1380, DOI: `10.1093/nar/gkac956`.

(23) Gautam, S.; Simon, L. Prediction of equilibrium phase compositions and β-glucosidase

partition coefficient in aqueous two-phase systems. *Chem. Eng. Commun.* **2007**, *194*, 117–128, DOI: `10.1080/00986440600715896`.

(24) Pazuki, G. R.; Taghikhani, V.; Vossoughi, M. Prediction of the partition coefficients of biomolecules in polymer–polymer aqueous two-phase systems using the artificial neural network model. *Part. Sci. Technol.* **2010**, *28*, 67–73, DOI: `10.1080/02726350903408175`.

(25) Chen, Y.; Liang, X.; Kontogeorgis, G. M. Artificial neural network modeling on the polymer-electrolyte aqueous two-phase systems involving biomolecules. *Sep. Purif. Technol.* **2023**, *306*, 122624, DOI: `10.1016/j.seppur.2022.122624`.

(26) Thacker, J. C. R.; Bray, D. J.; Warren, P. B.; Anderson, R. L. Can machine learning predict the phase behavior of surfactants? *J. Phys. Chem. B* **2023**, *127*, 3711–3727, DOI: `10.1021/acs.jpcb.2c08232`.

(27) Koren, Y.; Bell, R.; Volinsky, C. Matrix factorization techniques for recommender systems. *Computer* **2009**, *42*, 30–37, DOI: `10.1109/mc.2009.263`.

(28) Aggarwal, C. C. *Recommender systems*; Springer, 2016.

(29) Sturluson, A.; Raza, A.; McConachie, G. D.; Siderius, D. W.; Fern, X. Z.; Simon, C. M. Recommendation system to predict missing adsorption properties of nanoporous materials. *Chem. Mater.* **2021**, *33*, 7203–7216, DOI: `10.1021/acs.chemmater.1c01201`.

(30) Feng, M.; Cheng, M.; Ji, X.; Zhou, L.; Dang, Y.; Bi, K.; Dai, Z.; Dai, Y. Finding the optimal CO2 adsorption material: Prediction of multi-properties of metal-organic frameworks (MOFs) based on DeepFM. *Sep. Purif. Technol.* **2022**, *302*, 122111, DOI: `10.1016/j.seppur.2022.122111`.

(31) Großmann, O.; Bellaire, D.; Hayer, N.; Jirasek, F.; Hasse, H. Database for liquid phase

diffusion coefficients at infinite dilution at 298 K and matrix completion methods for their prediction. *Digital Discovery* **2022**, *1*, 886–897, DOI: `10.1039/d2dd00073c`.

(32) Jirasek, F.; Alves, R. A. S.; Damay, J.; Vandermeulen, R. A.; Bamler, R.; Bortz, M.; Mandt, S.; Kloft, M.; Hasse, H. Machine Learning in Thermodynamics: Prediction of Activity Coefficients by Matrix Completion. *J. Phys. Chem. Lett.* **2020**, *11*, 981–985, DOI: `10.1021/acs.jpclett.9b03657`.

(33) Damay, J.; Jirasek, F.; Kloft, M.; Bortz, M.; Hasse, H. Predicting Activity Coefficients at Infinite Dilution for Varying Temperatures by Matrix Completion. *Ind. Eng. Chem. Res.* **2021**, *60*, 14564–14578, DOI: `10.1021/acs.iecr.1c02039`.

(34) Jirasek, F.; Hayer, N.; Abbas, R.; Schmid, B.; Hasse, H. Prediction of parameters of group contribution models of mixtures by matrix completion. *Phys. Chem. Chem. Phys.* **2023**, *25*, 1054–1062, DOI: `10.1039/d2cp04478a`.

(35) Hayer, N.; Jirasek, F.; Hasse, H. Prediction of Henry's law constants by matrix completion. *AIChE J.* **2022**, *68*, e17753, DOI: `10.1002/aic.17753`.

(36) Seko, A.; Hayashi, H.; Kashima, H.; Tanaka, I. Matrix- and tensor-based recommender systems for the discovery of currently unknown inorganic compounds. *Phys. Rev. Materials* **2018**, *2*, 013805, DOI: `10.1103/PhysRevMaterials.2.013805`.

(37) Hayashi, H.; Hayashi, K.; Kouzai, K.; Seko, A.; Tanaka, I. Recommender system of Successful Processing Conditions for New Compounds Based on a Parallel Experimental Data Set. *Chem. Mater.* **2019**, *31*, 9984–9992, DOI: `10.1021/acs.chemmater.9b01799`.

(38) Tynes, M. F. Tensor Factorizations for Recommending Perovskite Crystallization Trials. M.Sc. thesis, Fordham University, 2020.

(39) Yuana, Q.; Longob, M.; Thorntonc, A.; McKeownd, N. B.; Comesaña-Gándarad, B.; Jansenb, J. C.; Jelfsa, K. E. Imputation of Missing Gas Permeability Data for Polymer Membranes using Machine Learning. *J. Membrane Sci.* **2021**, *627*, 119207, DOI: `10.1016/j.memsci.2021.119207`.

(40) Sosnina, E. A.; Sosnin, S.; Nikitina, A. A.; Nazarov, I.; Osolodkin, D. I.; Fedorov, M. V. Recommender systems in antiviral drug discovery. *ACS Omega* **2020**, *5*, 15039–15051, DOI: `10.1021/acsomega.0c00857`.

(41) Udell, M.; Horn, C.; Zadeh, R.; Boyd, S. Generalized low rank models. *Foundations and Trends® in Machine Learning* **2016**, *9*, 1–118, DOI: `10.1561/2200000055`.

(42) Johnson, C. C. Logistic matrix factorization for implicit feedback data. *Advances in Neural Information Processing Systems* **2014**, *27*, 1–9.

(43) Udell, M. Big data is low rank. *SIAG/OPT Views and News* **2019**, *27*, 7–12.

(44) Paradkar, M.; Udell, M. Graph-Regularized Generalized Low-Rank Models. 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW). 2017; pp 7–12, DOI: `10.1109/cvprw.2017.240`.

(45) Cai, D.; He, X.; Han, J.; Huang, T. S. Graph regularized nonnegative matrix factorization for data representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **2010**, *33*, 1548–1560, DOI: `10.1109/TPAMI.2010.231`.

(46) Zhang, Z.; Zhao, K. Low-rank matrix approximation with manifold regularization. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **2012**, *35*, 1717–1729, DOI: `10.1109/TPAMI.2012.274`.

(47) Liu, Y.; Wu, M.; Miao, C.; Zhao, P.; Li, X.-L. Neighborhood regularized logistic matrix factorization for drug-target interaction prediction. *PLoS Comput. Biol.* **2016**, *12*, e1004760, DOI: `10.1371/journal.pcbi.1004760`.

(48) Shu, Z.; Long, Q.; Zhang, L.; Yu, Z.; Wu, X.-J. Robust graph regularized NMF with dissimilarity and similarity constraints for ScRNA-seq data clustering. *J. Chem. Inf. Model.* **2022**, *62*, 6271–6286, DOI: `10.1021/acs.jcim.2c01305`.

(49) Zhang, W.; Chen, Y.; Li, D.; Yue, X. Manifold regularized matrix factorization for drug-drug interaction prediction. *J. Biomed. Informatics* **2018**, *88*, 90–97, DOI: `10.1016/j.jbi.2018.11.005`.

(50) Li, J.; Yang, X.; Guan, Y.; Pan, Z. Prediction of Drug–Target Interaction Using Dual-Network Integrated Logistic Matrix Factorization and Knowledge Graph Embedding. *Molecules* **2022**, *27*, 5131, DOI: `10.3390/molecules27165131`.

(51) Jain, S.; Chouzenoux, E.; Kumar, K.; Majumdar, A. Graph regularized probabilistic matrix factorization for drug-drug interactions prediction. *IEEE J. Biomed. Health Inform.* **2023**, *27*, 2565–2574, DOI: `0.1109/JBHI.2023.3246225`.

(52) Ma, Y.; He, T.; Jiang, X. Multi-network logistic matrix factorization for metabolite–disease interaction prediction. *FEBS Lett.* **2020**, *594*, 1675–1684, DOI: `10.1002/1873-3468.13782`.

(53) Azuma, I.; Mizuno, T.; Kusuhara, H. NRBdMF: A recommendation algorithm for predicting drug effects considering directionality. *J. Chem. Inf. Model.* **2023**, *63*, 474–483, DOI: `10.1021/acs.jcim.2c01210`.

(54) Murphy, K. P. *Probabilistic machine learning: An introduction*; MIT press, 2022.

(55) Kingma, D. P.; Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* **2014**,

(56) Hansen, C. M. *Hansen Solubility Parameters: A User's Handbook*, 2nd ed.; CRC Press: Boca Raton, 2007; OCLC: ocm71312998.

(57) Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V. et al. Scikit-learn: Machine learning in Python. *J. Mach. Learn. Res.* **2011**, *12*, 2825–2830.

(58) Bezanson, J.; Edelman, A.; Karpinski, S.; Shah, V. B. Julia: A fresh approach to numerical computing. *SIAM Review* **2017**, *59*, 65–98, DOI: `10.1137/141000671`.

(59) Danisch, S.; Krumbiegel, J. Makie.jl: Flexible high-performance data visualization for Julia. *J. Open Source Software* **2021**, *6*, 3349, DOI: `10.21105/joss.03349`.

(60) Scheffler, M.; Aeschlimann, M.; Albrecht, M.; Bereau, T.; Bungartz, H.-J.; Felser, C.; Greiner, M.; Groß, A.; Koch, C. T.; Kremer, K. et al. FAIR data enabling new horizons for materials research. *Nature* **2022**, *604*, 635–642, DOI: `10.1038/s41586-022-04501-x`.

(61) Olivetti, E. A.; Cole, J. M.; Kim, E.; Kononova, O.; Ceder, G.; Han, T. Y.-J.; Hiszpanski, A. M. Data-driven materials research enabled by natural language processing and information extraction. *Appl. Phys. Rev.* **2020**, *7*, 041317, DOI: `10.1063/5.0021106`.

(62) Mace, C. R.; Akbulut, O.; Kumar, A. A.; Shapiro, N. D.; Derda, R.; Patton, M. R.; Whitesides, G. M. Aqueous multiphase systems of polymers and surfactants provide self-assembling step-gradients in density. *J. Am. Chem. Soc.* **2012**, *134*, 9094–9097, DOI: `10.1021/ja303183z`.

(63) David, N.; Sun, W.; Coley, C. W. The promise and pitfalls of AI for molecular and materials synthesis. *Nature Comput. Sci.* **2023**, *3*, 362–364, DOI: `10.1038/s43588-023-00446-x`.

(64) Salakhutdinov, R.; Mnih, A. Bayesian probabilistic matrix factorization using Markov chain Monte Carlo. Proc. 25th International Conference on Machine Learning. 2008; pp 880–887, DOI: `10.1145/1390156.1390267`.

(65) Agnihotri, A.; Batra, N. Exploring Bayesian optimization. *Distill* **2020**, *5*, e26, DOI: `10.23915/distill.00026`.

(66) Kolda, T. G.; Bader, B. W. Tensor Decompositions and Applications. *SIAM Review* **2009**, *51*, 455–500, DOI: `10.1137/07070111X`.

(67) Jia, X.; Lynch, A.; Huang, Y.; Danielson, M.; Lang'at, I.; Milder, A.; Ruby, A. E.; Wang, H.; Friedler, S. A.; Norquist, A. J. et al. Anthropogenic biases in chemical reaction data hinder exploratory inorganic synthesis. *Nature* **2019**, *573*, 251–255, DOI: `10.1038/s41586-019-1540-5`.

(68) Beker, W.; Roszak, R.; Wołos, A.; Angello, N. H.; Rathore, V.; Burke, M. D.; Grzybowski, B. A. Machine learning may sometimes simply capture literature popularity trends: A case study of heterocyclic Suzuki–Miyaura coupling. *J. Am. Chem. Soc.* **2022**, *144*, 4819–4827, DOI: `10.1021/jacs.1c12005`.