Relaxation Methods for Constrained Matrix Factorization Problems: Solving the Phase Mapping Problem in Materials Discovery

Junwen Bai[1], Johan Bjorck[1(✉)], Yexiang Xue[1], Santosh K. Suram[2], John Gregoire[2], and Carla Gomes[1]

[1] Department of Computer Science, Cornell University, Ithaca, NY 14850, USA
ujb225@cornell.edu

[2] Joint Center for Artificial Photosynthesis, California Institute of Technology, Pasadena, CA 91125, USA

Abstract. Matrix factorization is a robust and widely adopted technique in data science, in which a given matrix is decomposed as the product of low rank matrices. We study a challenging constrained matrix factorization problem in materials discovery, the so-called phase mapping problem. We introduce a novel "lazy" Iterative Agile Factor Decomposition (IAFD) approach that relaxes and postpones non-convex constraint sets (the lazy constraints), iteratively enforcing them when violations are detected. IAFD interleaves multiplicative gradient-based updates with efficient modular algorithms that detect and repair constraint violations, while still ensuring fast run times. Experimental results show that IAFD is several orders of magnitude faster and its solutions are also in general considerably better than previous approaches. IAFD solves a key problem in materials discovery while also paving the way towards tackling constrained matrix factorization problems in general, with broader implications for data science.

Keywords: Constrained matrix factorization · Relaxation methods · Multiplicative updates · Phase-mapping

1 Introduction

Matrix factorization has become a ubiquitous technique in data analysis, with applications in a variety of domains such as computer vision [10], topic modeling [6], audio signal processing [11], and crystallography [12]. Often the phenomena considered is naturally non-negative. In non-negative matrix-factorization, the goal is to explain a non-negative signal as the product of (typically) two non-negative low rank matrices. Nonnegative matrix factorization is known to be NP-Hard [13], so a general algorithm for matrix factorization most likely scales exponentially in the worst case.

We consider a challenging and central problem in materials discovery, so-called phase-mapping, an inverse problem whose goal is to infer the materials'

D. Salvagnin and M. Lombardi (Eds.): CPAIOR 2017, LNCS 10335, pp. 104–112, 2017.
DOI: 10.1007/978-3-319-59776-8_9

crystal structure based on X-ray sample data, see Fig. 1(Left). Phase-mapping was shown to be NP-Hard [5]. Existing approaches to phase mapping, discussed in the next section, do not satisfy all the problem constraints. Furthermore, approaches that explicitly try to incorporate the main problem constraints have prohibitive run times on typical real-world data, hours or days, while still not producing solutions that are completely physically meaningful.

We propose a novel **Interleaved Agile Factor Decomposition (IAFD)** approach that "lazily" relaxes and postpones non-convex constraint sets (the lazy constraints), iteratively enforcing them when violations are detected, see Fig. 1(Right). IAFD uncovers the main underlying problem structure revealed by the sample data by rapidly performing a large number of lightweight gradient-based moves. In order to incorporate more intricate combinatorial constraints, the algorithm interleaves the multiplicative gradient-based updates with efficient modular algorithms that detect and repair constraint violations, while still ensuring fast run times, scaling up to large scale real-world problems. Our experimental results show that IAFD is several orders of magnitude faster and its solutions are also in general considerably better than previous approaches. Our work provides an efficient approach to solving a central problem in materials discovery, while paving the way towards tackling constrained matrix factorization problems in general, with broader implications for data science.

2 The Phase Mapping Problem

In search of new materials a common experimental method is to deposit several elements onto a sample wafer at different angles. The sample locations on the wafer receive different concentrations of the elements. As a result, distinct and potentially undiscovered materials are formed at different locations. All materials can be characterized by a one-dimensional X-ray diffraction pattern $F(q)$, which can be measured at high energy accelerators. However, several phases might be present at one sample location and the X-ray diffraction pattern at that location then becomes a linear combination of a set of basis patterns, each corresponding to the pattern of one pure phase. Figure 1(Left) illustrates this phenomenon.

In the mathematical model of the problem, a matrix A representing a set of X-ray measurements on a sample wafer is obtained. Each column of A is a vector representing the pattern $F(q)$ obtained at one sample location, sampled for Q fixed values of q. The phase mapping problem entails factorizing A into the product of W and H such that $A \approx WH$.

The matrix W encodes the characteristic patterns of pure phases while H represents how much of the different phases are present at individual sample location. A complicating factor of the phase-mapping problem is that the laws of thermodynamics induce a set of physical constraints on the possible underlying low rank representation. The solutions must satisfy these constraints, defined below, and must additionally be nonnegative as the physical quantities described by the matrices cannot be negative. Efficient methods of solving this problem accelerates materials science and enables automatic experimentation in search of tomorrow's semiconductor and photvoltaic materials.

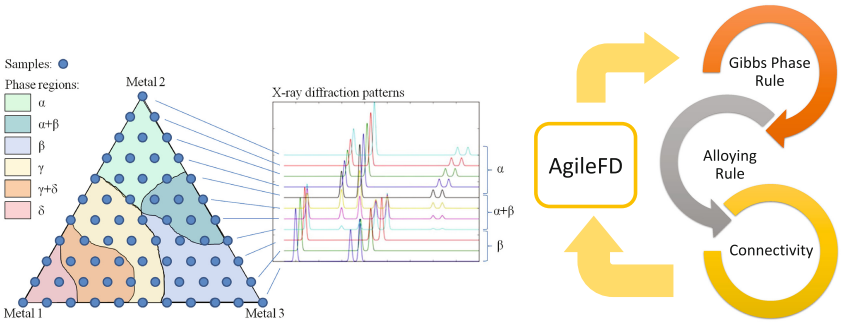

Fig. 1. (Left) The goal of the phase mapping problem is to explain observed X-ray diffraction patterns at multiple sample locations in terms of the underlying phases or crystal structures of the materials. Here the X-ray diffraction patterns of sample locations on the right edge of the triangle are shown in the middle plot. The top four sample locations only have phase α, the bottom three only have phase β, while the middle four sample locations have both α and β. In addition, the X-ray diffraction patterns of both phase α and β are shifting to the right. **(Right)** At a high level, our Interleaved Agile Factor Decomposition (IAFD) algorithm starts with solving a relaxed problem using the multiplicative update rules of AgileFD [14], without enforcing combinatorial constraints. Violations of the Gibbs' phase rule, the alloying rule, and the connectivity constraint in the relaxed solutions are then addressed by efficient modular algorithms, in an interleaving manner. This procedure is iterated, creating a closed loop involving AgileFD and the three modules.

Shifting. A phenomenon that complicates the matrix factorization is "shifting", where the X-ray patterns are changed in the sense $F(q) \rightarrow F(\lambda_k q)$, for some real number λ_k that is fixed for each phase k and column in A. For example, the X-ray patterns in Fig. 1 are shifting to the right. The problem can be circumvented by resampling the signal uniformly on a logarithmic scale, where multiplicative shifts becomes additive. For fixed m and k, the vector $(0, \ldots, 0, W_{1,k}, \ldots, W_{Q-m,k})^T$ formed by shifting the k-th column of W down by m entries (and filling 0 for remaining entries) describes basis pattern of phase k shifted by an amount controlled by m. We can then allow λ_k to attain M different discrete values by letting $m \in 0, 1...M-1$. By characterizing the H matrix with three indices, one per phase k, sample point n, and allowed discrete value of λ_k m, we can now express a linear combination of shifted basis patterns as $A_{qn} \approx \sum_{km} W_{q-m,k} H_{kmn}$. Since this specific formulation will be used, the constraints of the phase mapping problem will be given in terms of W_{qk} and H_{kmn}, however other formulations of the rules are possible [3].

Gibbs' Phase Rule. In a setting with three elements deposited, such as in Fig. 1, Gibbs' phase rule [1] states that the number of phases present at each sample location is at most three. Mathematically, it is equivalent to constraining the number of non-zero elements in vector $(\sum_m H_{1mn}, \sum_m H_{2mn}, \ldots)$ for any phase k to be no more than three. Thus, for fixed n we have $\| \sum_m H_{kmn} \|_0 \leq 3$.

Connectivity. The connectivity rule requires that the sample points where a specific phase is present form a continuous domain on the sample wafer. For example, in Fig. 1, each pattern occupies a continuous region. Mathematically, since we have a discrete set of measurements we describe the constraint via a graph G where sample points are nodes and nearby sample points are connected with an edge. This graph is obtained through Delauney triangulation [7] of the sample points. A continuous domain then corresponds to a connected component on this graph, and we require that all sample points n with phase k present, i.e. $\sum_m H_{kmn} > 0$, form a connected component on G.

Alloying Rule. The shifting parameter λ_k for phase k may shift continuously across the sample points as a result of so called alloying. The alloying rule states that for points where λ_k is changing, Gibbs' phase rule becomes even stricter and requires $\|\sum_m H_{kmn}\|_0 \leq 2$. In this discrete setting we interpret λ_k of a point n as $\sum_m H_{nkm} m / \sum_m H_{nkm}$, which can be thought of as the expectation of m when we normalize H_{kmn} to a probability distribution. Two neighboring sample points n and n' with phase k present, which means $\sum_m H_{kmn} > 0$ and $\sum_m H_{kmn'} > 0$, are considered shifting if

$$\left\| \frac{\sum_m H_{kmn} m}{\sum_m H_{kmn}} - \frac{\sum_m H_{kmn'} m}{\sum_m H_{kmn'}} \right\| > \epsilon, \tag{1}$$

The alloying rule states that if Eq. 1 is satisfied for any phase k and neighbouring sample points n' and n, then we must have $\|\sum_m H_{kmn}\|_0 \leq 2$.

2.1 Previous Approaches

Many algorithms have been proposed for solving the phase mapping problem, for example [5,8,9]. Recently an efficient algorithm called AgileFD [14], based on coordinate descent using multiplicative updates, has been proposed. If we let the matrix R represent the product of H and W, i.e. $R_{qn} = \sum_m W_{q-m,k} H_{kmn}$ these updates are

$$H_{kmn} \leftarrow H_{kmn} \frac{\sum_q W_{q-m,k}(A_{qn}/R_{qn})}{\sum_q W_{q-m,k} + \gamma}, \tag{2}$$

$$W_{qk} \leftarrow W_{qk} \frac{\sum_{mn} \frac{A_{q+m,n}}{R_{q+m,n}} H_{kmn} + W_{qk} \sum_{q'nm} H_{kmn} W_{q'k}}{\sum_{nk} H_{kmn} + W_{qk} \sum_{q'nm} \frac{A_{q'+m,n}}{R_{q'+m,n}} H_{kmn} W_{q'k}}. \tag{3}$$

The algorithm relies on manual refinement by domain experts to enforce combinatorial constraints, which makes it problematic to use in a scalable fashion.

Another approach called combiFD, able to express all constraints, has been proposed [2]. It relies on a combinatorial factor decomposition formulation, where iteratively H or W are frozen while the other is updated by solving a MIP. This formulation allows all constraints to be expressed upfront, however solving the complete MIP programs is infeasible in practice.

3 Interleaved Agile Factor Decomposition

Given a non-negative Q-by-N measurement matrix A and the dimensions K and M of the factorization, the phase mapping problem entails explaining A as a generalized product of two low rank non-negative matrices W, H. The entire mathematical formulation becomes:

$$\begin{aligned} \min \quad & \sum_{qn} |A_{qn} - \sum_{mk} W_{q-m,k} H_{kmn}|, \quad s.t. \;\; H \in \mathbb{R}_+^{K \times M \times N}, \;\; W \in \mathbb{R}_+^{Q \times K}, \\ & H, W \text{ satisfies Gibbs' phase rule, Connectivity, Alloying rule.} \end{aligned} \tag{4}$$

Representing the combinatorial rules as integer constraints has previously been tried [2], however the resulting large MIP formulations are not feasible to solve in practice. Instead, we propose a novel iterative framework that interleaves efficient multiplicative updates with compact subroutines able to address specific constraints, called Interleaved Agile Factor Decomposition (IAFD). The algorithm is illustrated, at a high level, in Fig. 1 (Right). The central insight is that our constraints are too expensive to explicitly encode and maintain, however finding and rectifying individual violations can be done efficiently. This motivates a lazy approach that relaxes and postpones non-convex constraint sets (the lazy constraints), iteratively enforcing them only as violations are detected. For each constraint we provide an efficient method to detect violations and repair them through much smaller optimization problems.

The IAFD algorithm starts with solving the relaxed problem, with only the convex non-negativity constraint, using the multiplicative updating rules (2) and (3) of AgileFD [14]. This relaxed solution is then slightly refined by three subroutines which sample and rectify violations of Gibbs' phase rule, the alloying rule, and the connectivity constraint respectively, by solving small scale optimization problems. The refined solution is then relaxed again and improved through the multiplicative updates. This process is repeated in an interleaving manner which creates a closed loop involving AgileFD and the three refining modules. A reason why this interleaving can be expected to not produce much duplicate effort is due to the following observation:

Proposition 1. *The number of non-zero entries in* H *:* $\|\{(n,k)|\sum_m H_{nkm} > 0\}\|$ *is nonincreasing under updates* (2) *of AgileFD.*

This comes from the fact that every component is updated through multiplication with itself in (2), which ensures that zero-components stay zero. Thus, if Gibbs' phase rule is satisfied before the multiplicative updates, it will still be satisfied after. We now describe the subroutines handling the constraints.

Gibbs' Phase Rule Refinement. After obtaining the matrix W and H, we find violations of Gibbs' phase rule by scanning sample points and noting which ones have more than three phases present. One key insight is that the problem of enforcing Gibbs' phase rule decouples between sample points once the matrix W is fixed. In order to represent the constraint that no more than three phases are

present, we introduce a binary variable δ_{kn} denoting whether phase k is present at sample location n (i.e., $\sum_m H_{kmn}$ is nonzero). The constraint is now enforced by solving the following mixed integer program with W fixed for each violated sample point, which results in a very light-weight refinement:

$$\min_{\delta, H_{kmn} \forall k,m} \sum_q |A_{qn} - \sum_{mk} W_{q-m,k} H_{kmn}|,$$
$$\text{s.t. } \forall k, m \; H_{kmn} \leq M \delta_{nk}, \; \sum_k \delta_{nk} \leq 3. \tag{5}$$

Here, $H_{kmn} \leq M\delta_{nk}$ is a big-M constraint, which enforces that phase k is zero if δ_{nk} is zero. We use $\sum_k \delta_{nk} \leq 3$ to enforce that only three phases are allowed. These compact programs typically contains two orders of magnitude fewer variables then the complete program, and can be quickly solved in parallel.

Alloying Rule Refinement. Violations of the alloying rule can be found by comparing the shift parameter λ_k of some sample point n, here interpreted as $\sum_m H_{kmn} m / \sum_m H_{kmn}$, to that of its neighbors in graph G. This simply amounts to a linear scan through all sample points. It is again possible to decouple the constraint by taking W and n as fixed, which allows for a compact mixed integer program formulation. We fix the violating sample point n, denote the set of its neighbors as $N(n)$, and then calculate $\lambda_{kn'} = \sum_m m H_{kmn'} / \sum_m H_{kmn'}$ for all neighbors $n' \in N(n)$ where phase k is present. In the MIP the binary variable δ_{kn} is used to denote whether phase k is present at sample point n, another binary variable τ_n is then introduced to denote whether the sample point undergoes shift. By using a large M-constraint as in Gibbs' phase rule module we can encode that unless the sample point is shifting or doesn't contain the phase k, the λ_k has to be close to that of it's neighbors as follows:

$$\min_{\tau, \delta, H_{kmn} \forall k,m} \; \sum_q |A_{qn} - \sum_{mk} W_{q-m,k} H_{kmn}|,$$
$$\text{s.t. } |\sum_m H_{kmn} m - \lambda_{kn'} \sum_m H_{kmn}| \leq \epsilon \sum_m H_{kmn} + M\tau_n + M(1 - \delta_{kn}),$$
$$\forall k, m, n' \in N(n), \qquad H_{kmn} \leq M\delta_{nk}, \qquad \sum_k \delta_{nk} + \tau_n \leq 3. \tag{6}$$

Connectivity Refinement. While explicitly encoding the constraint is computationally expensive, finding violations can be done in a lightweight manner. For each phase k we find all continuous regions containing phase k by simply finding the connected components of our graph G where phase k is present. To rectify the constraint, every connected component C is then weighted by the total amount of present phase, which amounts to calculating the quantity $\sum_{n \in C, m} H_{kmn}$. This weight corresponds to the amount of present signal. We then zero out components in H corresponding to phase k and sample points in the least weighted connected components. This procedure ensures that all the phases correspond to a single contiguous regions, without deteriorating much (if at all) the objective function in general.

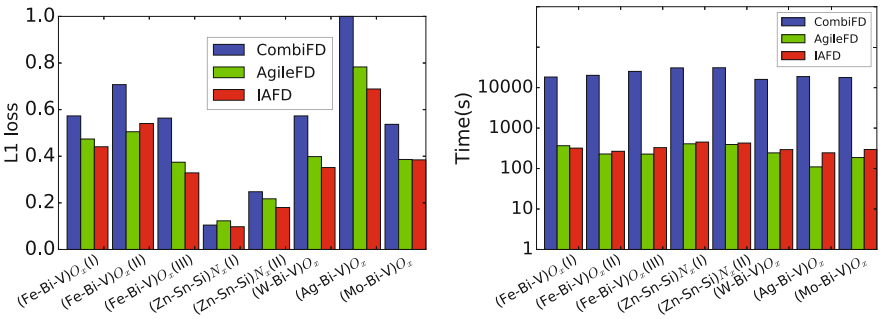

Fig. 2. (Left) Normalized L1 Loss of the difference between ground-truth and reconstructed X-ray patterns for the algorithms on 8 real world systems. IAFD performs best, with combiFD lagging behind the two other methods. (Right) Runtime for CombiFD, AgileFD, IAFD to solve 8 real systems, note the logarithmic time scale. We can clearly see that the heavy duty MIP formulation of combiFD results running times of hours, while the two lightweight methods runs in a matter of minutes.

4 Experimental Results

IAFD is evaluated on several real world instances of the phase mapping problem, available at [4]. We randomly initialize the matrices, and as the interleaving with the connectivity-subroutine and the alloying-subroutine assumes structured data, the whole algorithm starts with several rounds of AgileFD interleaving with Gibbs' rule followed by the other two subroutines. The diffraction patterns are probed at around 200 locations of the respective wafers with approximately 1700 values of q sampled, we set $K = 6$ and $M = 8$ which gives us around two million variables per problem. More rounds of interleaving lead to better results but of course it takes more time. We chose to do three rounds of AgileFD interleaving with Gibbs' rule followed by enforcing the other two constraints to balance these tradeoffs. Our method is compared against CombiFD [2], with a mipgap of 0.1 and 15 iterations. Due to its poor scaling properties only the Gibbs' phase rule is enforced for CombiFD. We also compare IAFD against AgileFD [14], with termination constant set to 10^{-5}.

The most important metric when comparing different methods is the solution quality, measured by L1 loss. Results shown in Fig. 2 (Left). It is evident that CombiFD in [2] has subpar performance, while IAFD wins by a slight margin over AgileFD. This suggests that enforcing the constraints actually improves the reconstruction error. The area where we expect IAFD to perform the best is in terms of enforcing the physical constraints, which is illustrated in Table 1. Here IAFD consistently performs the best with zero violations, which results in physically meaningful solutions to the phase mapping problem.

The smaller subroutines are evidently able to handle all constraints and additionally provide a low loss, which might lead one to suspect that IAFD has long run times. That is not the case. The run times can be viewed in Fig. 2 (Right). While AgileFD is slightly faster than IAFD, the difference is very small.

CombiFD, which explicitly enforces the constraints [2], has prohibitive long run times in practice, which suggests that a complete MIP encoding is both inefficient and unnecessary. These results show that IAFD can enforce all physical rules, without sacrificing much in either reconstruction error or running time.

Table 1. To the left we see the fraction of sample points violating the alloying rule for different algorithms, where IAFD consistently has no violations. The right side gives the average number of connected components per phase, and here only IAFD always contain a single continuous region as required by the connectivity constraint.

System	Alloying constraint			Connectivity constraint		
	CombiFD	AgileFD	IAFD	CombiFD	AgileFD	IAFD
(Fe-Bi-V)O_x(I)	0.57	0.15	**0.00**	**1.00**	1.65	**1.00**
(Fe-Bi-V)O_x(II)	0.55	0.30	**0.00**	2.40	1.65	**1.00**
(Fe-Bi-V)O_x(III)	0.18	0.03	**0.00**	2.50	2.18	**1.00**
(Zn-Sn-Si)N_x(I)	0.06	0.01	**0.00**	**1.00**	2.38	**1.00**
(Zn-Sn-Si)N_x(II)	0.05	0.02	**0.00**	2.00	1.38	**1.00**
(W-Bi-V)O_x	0.54	0.08	**0.00**	1.67	2.31	**1.00**
(Ag-Bi-V)O_x	0.84	0.16	**0.00**	3.60	1.96	**1.00**
(Mo-Bi-V)O_x	0.46	0.08	**0.00**	1.60	1.72	**1.00**

5 Conclusions

We propose a novel Interleaved Agile Factor Decomposition (IAFD) framework for solving the phase mapping problem, a challenging constrained matrix factorization problem in materials discovery. IAFD is a lightweight iterative approach that lazily enforces non-convex constraints. The algorithm is evaluated on several real world instances and outperforms previous solvers both in terms of run time and solution quality. IAFD's approach, based on efficient multiplicative updates from unconstrained nonnegative matrix factorization and lazily enforced constraints, performs much better compared to approaches that enforce all constraints upfront, using a large mathematical program. This approach opens up a new angle for efficiently solving more general constrained factorization problems. We anticipate deploying IAFD at the Stanford Synchrotron Radiation Lightsource in the near future to the benefit of the materials science community.

Acknowledgements. We thank Ronan Le Bras and Rich Bernstein for fruitful discussion. This material is supported by NSF awards CCF-1522054, CNS-0832782, CNS-1059284, IIS-1344201 and W911-NF-14-1-0498. Experiments were supported through the Office of Science of the U.S. Department of Energy under Award No. DE-SC0004993. Use of the Stanford Synchrotron Radiation Lightsource, SLAC National Accelerator Laboratory, is supported by the U.S. Department of Energy, Office of Science, Office of Basic Energy Sciences under Contract No. DE-AC02-76SF00515.

References

1. Atkins, P., De Paula, J.: Atkins' Physical Chemistry, p. 77. Oxford University Press, New York (2006)
2. Ermon, S., Bras, R.L., Suram, S.K., Gregoire, J.M., Gomes, C., Selman, B., Van Dover, R.B.: Pattern decomposition with complex combinatorial constraints: application to materials discovery. arXiv preprint arXiv:1411.7441 (2014)
3. Ermon, S., Bras, R., Gomes, C.P., Selman, B., Dover, R.B.: SMT-aided combinatorial materials discovery. In: Cimatti, A., Sebastiani, R. (eds.) SAT 2012. LNCS, vol. 7317, pp. 172–185. Springer, Heidelberg (2012). doi:10.1007/978-3-642-31612-8_14
4. Le Bras, R., Bernstein, R., Suram, S.K., Gregoire, J.M., Selman, B., Gomes, C.P., van Dover, R.B.: A computational challenge problem in materials discovery: synthetic problem generator and real-world datasets (2014)
5. LeBras, R., Damoulas, T., Gregoire, J.M., Sabharwal, A., Gomes, C.P., Dover, R.B.: Constraint reasoning and kernel clustering for pattern decomposition with scaling. In: Lee, J. (ed.) CP 2011. LNCS, vol. 6876, pp. 508–522. Springer, Heidelberg (2011). doi:10.1007/978-3-642-23786-7_39
6. Lee, D.D., Seung, H.S.: Learning the parts of objects by non-negative matrix factorization. Nature **401**(6755), 788–791 (1999)
7. Lee, D.T., Schachter, B.J.: Two algorithms for constructing a delaunay triangulation. Int. J. Comput. Inf. Sci. **9**(3), 219–242 (1980)
8. Long, C., Bunker, D., Li, X., Karen, V., Takeuchi, I.: Rapid identification of structural phases in combinatorial thin-film libraries using X-ray diffraction and non-negative matrix factorization. Rev. Sci. Instrum. **80**(10), 103902 (2009)
9. Long, C., Hattrick-Simpers, J., Murakami, M., Srivastava, R., Takeuchi, I., Karen, V.L., Li, X.: Rapid structural mapping of ternary metallic alloy systems using the combinatorial approach and cluster analysis. Rev. Sci. Instrum. **78**(7), 072217 (2007)
10. Shashua, A., Hazan, T.: Non-negative tensor factorization with applications to statistics and computer vision. In: Proceedings of the 22nd International Conference on Machine Learning, pp. 792–799. ACM (2005)
11. Smaragdis, P.: Non-negative matrix factor deconvolution; extraction of multiple sound sources from monophonic inputs. In: Puntonet, C.G., Prieto, A. (eds.) ICA 2004. LNCS, vol. 3195, pp. 494–499. Springer, Heidelberg (2004). doi:10.1007/978-3-540-30110-3_63
12. Suram, S.K., Xue, Y., Bai, J., Bras, R.L., Rappazzo, B., Bernstein, R., Bjorck, J., Zhou, L., van Dover, R.B., Gomes, C.P., et al.: Automated phase mapping with agilefd and its application to light absorber discovery in the V-Mn-Nb oxide system. arXiv preprint arXiv:1610.02005 (2016)
13. Vavasis, S.A.: On the complexity of nonnegative matrix factorization. SIAM J. Optim. **20**(3), 1364–1377 (2009)
14. Xue, Y., Bai, J., Le Bras, R., Rappazzo, B., Bernstein, R., Bjorck, J., Longpre, L., Suram, S., van Dover, B., Gregoire, J., Gomes, C.: Phase mapper: an AI platform to accelerate high throughput materials discovery. In: Twenty-Ninth International Conference on Innovative Applications of Artificial Intelligence (2016)