An Efficient Relaxed Projection Method for Constrained Non-negative Matrix Factorization with Application to the Phase-Mapping Problem in Materials Science

Junwen Bai[1](✉), Sebastian Ament[1], Guillaume Perez[1], John Gregoire[2], and Carla Gomes[1]

[1] Department of Computer Science, Cornell University, Ithaca, NY 14853, USA
jb2467@cornell.edu
[2] Joint Center for Artificial Photosynthesis, California Institute of Technology, Pasadena, CA 91125, USA

Abstract. In recent years, a number of methods for solving the constrained non-negative matrix factorization problem have been proposed. In this paper, we propose an efficient method for tackling the ever increasing size of real-world problems. To this end, we propose a general relaxation and several algorithms for enforcing constraints in a challenging application: the phase-mapping problem in materials science. Using experimental data we show that the proposed method significantly outperforms previous methods in terms of ℓ_2-norm error and speed.

1 Introduction

Matrix factorization is a well-known method used for data compression and information extraction. Given a matrix A, matrix factorization is the problem of finding two matrices W and H such that $A \approx WH$. As W and H are assumed to be low-rank, the sum of their sizes is usually much smaller than the size of A. Further, the columns of W can be interpreted as basis components, which are linearly combined by columns of H to reconstruct A. The matrix factorization problem occurs in numerous fields, for example topic modeling [1], audio signal processing [2], and crystallography [3]. While successful algorithms for classical matrix factorization have been found, some variants of this problem are challenging. For example, merely restricting W and H to be element-wise non-negative is known to lead to an NP-Hard problem [4], called non-negative matrix factorization (NMF) [1,4]. These variants are important for many practical problems.

Moreover, many real-world problems which can be modeled by NMF also involve domain-specific constraints. A good example is the phase-mapping problem in the field of materials discovery [5]. This problem arises when materials scientists generate potentially novel materials by applying a physical transformation to mixtures of known materials. Individual materials are commonly characterized by a variety of spectrographic techniques, like x-ray diffraction [6] and

W.-J. van Hoeve (Ed.): CPAIOR 2018, LNCS 10848, pp. 52–62, 2018.
https://doi.org/10.1007/978-3-319-93031-2_4

Raman spectroscopy [7]. Each spectrogram produced by these experiments typically corresponds to a mixture of potentially novel materials or phases, whose individual spectrograms are unkown. The phase-mapping problem is then to uncover these unknown phases, from which materials scientists can understand the phase behavior of the associated materials and its relationship to other measured properties. At a high level, the problem can be framed as an NMF problem, where the columns of A are the measured spectrograms Importantly, the matrices W and H of the resulting factorization problem have to respect hard physical constraints for the solution to be meaningful to scientists. This makes the phase-mapping problem particularly challenging from a computational perspective.

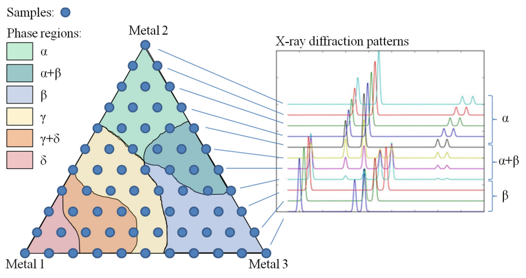
In order to incorporate these hard constraints, recent work used mixed-integer programming (MIP) to project the matrices W and H onto the constraint space, after an unconstrained optimization [8] of the NMF problem [9]. Subsequently, W and H were re-optimized without leaving the constraint space. While the results of this work are promising, the method still requires solving time-consuming, hard combinatorial problems for enforcing the constraints. Other previous approaches to this problem were purely based on combinatorial search algorithms applied to the constraint space [5,10,11]. However, these methods are often intractable and deteriorate with the presence of noise in the data. The goal of this paper is to obviate the need for combinatorial optimization techniques for the solution of this problem.

In this paper, we propose a heuristic approach that uses a polynomial-time algorithm **Projected Interleaved Agile Factor Decomposition (PIAFD)** interleaving the enforcement of essential hard-constraints, combined with a relaxation which allows the algorithm to move the optimization variables into the constraint space continuously while optimizing the objective function for the phase-mapping problem. This projection-based algorithm is designed to enforce the three constraints which are most essential in the phase-mapping problem: the Gibbs, Alloying and connectivity constraints. This new algorithm works well in practice, and allows to extract accurate, constraint-satisfying solutions in a very short time. We study the impact of the application of these algorithms on the solution quality, as measured by different objective functions.

This paper is organized as follows. The phase-mapping problem is first described and modeled as an associated matrix factorization problem. After an overview of the state-of-the-art methods for this problem, we present our new method *PIAFD* for enforcing the relevant constraints. Finally, *PIAFD* is applied to several real-world data sets of the phase-mapping problem to demonstrate its performance.

2 Preliminaries

The phase-mapping problem has recently drawn attention because of its great importance to the discovery of new materials [12,13]. In their search for new materials, scientists try to characterize novel materials with spectrographic techniques, like x-ray diffraction (XRD) spectroscopy. For each sample point on an



$$\boxed{\begin{aligned} H^m &= H^m \circ \frac{W^{\downarrow m^T}(\frac{A}{R})}{W^{\downarrow m^T}\mathbb{1}_{L\times N}} \\ W &= W \circ \frac{\sum\limits_m (\frac{A^{\uparrow m}}{R^{\uparrow m}})H^{m T}}{\sum\limits_m \mathbb{1}_{L\times K}H^{m T}} \end{aligned}}$$

Fig. 1. (Left) the schematic on the left is known as a phase diagram. The sides of the triangle correspond to proportions of the metals which are deposited on the wafer. The corners correspond to regions where only one metal has been deposited, while the center of the triangle corresponds to a location where all metals are deposited in equal proportions. The legend indicates that four phases are present: $\alpha, \beta, \gamma, \delta$. Each colors represents a region in which a unique phase or a unique combination of phases is present. The graph on the right shows how the spectrograms can vary as the proportions of the initial metals are changed. (Right) multiplicative gradient update rules. (Color figure online)

experimental wafer, we denote by $F(q)$ the vector of diffraction intensities as a function of q, the scattering vector. That is, $F(q)$ corresponds to the spectrogram observed at a location on the wafer, and we will refer to it as an XRD pattern. Importantly, each location on the wafer is likely to contain mixtures of new materials. Therefore, $F(q)$ will be a combination of the spectrograms of several unkown materials (i.e. phases), which are generally not observed directly. This makes the phase-mapping problem non-trivial.

This problem is naturally formulated using a matrix A, each of whose columns consists of a XRD pattern $F(q)$ of length Q. If there are N sample points on the wafer, A is of size $Q \times N$. The algorithms of this paper are based on factorizing A using two matrices W and H such that $A \approx WH$. W stores the different phases, while H contains the quantity of each basis pattern at each sample point.

While the non-negativity of the spectrograms puts one constraint on W and H, several other constraints are also present. We briefly describe these additional constraints, and current methods of enforcing them. The *IAFD* method [9] alternates multiplicative gradient updates (Fig. 1) and constraint refinement. The generalized Kullback-Leibler (KL) divergence between A and WH is taken to be the objective function. The update rules above are proven to be non-decreasing, but do not necessarily converge to a stationary point of the objective function. Notably, [14] introduced modifications to these updates which do provably converge to a stationary point.

***Shifting*.** A common phenomenon that occurs in the XRD spectroscopy of certain materials is *shifting*. A basis pattern $F_b(q)$ is shifting with a multiplicative factor λ if the pattern $F_b(\lambda q)$ is present at a sample point, instead of $F_b(q)$. Crucially, a shifted pattern should still be recognized as the original pattern. In

Fig. 1, the signals on the bottom right show the shifting of phase β. In order to model this behavior, one can resample the signal $F(\lambda q)$ on a logarithmic scale, so that the multiplicative shifting becomes additive. Then, a convolutive NMF framework can be readily applied to this problem [9]. This framework allows for multiple shifted copies of the phases. In particular, the columns of W are allowed to shift down with a certain shifting amount m, $W^{\downarrow m}$. Then the factorization problem can be defined as $A \approx \sum_m W^{\downarrow m} H^m$ where H^m is of size $K \times N$ consists of the activation coefficients of corresponding shifted phases. The exact shifting of phase k at sample j is defined by a weighted average of different shifting amounts $\lambda_{kj} = \sum_m m H^m_{kj} / \sum_m H^m_{kj}$.

Gibbs. The Gibbs phase rule puts a limit on the number of phases which can be observed at a sample point in a thermodynamic equilibrium. In particular, the maximum number of different phases at a single location is equal to the number of elements, G, which were deposited on the wafer initially. For example, on the wafer where three different elements were deposited, the maximum number of phases at each sample point is three. This implies that the number of non-zero elements in each column of H should not exceed G. The *IAFD* algorithm projects solutions onto the constraint space by solving a MIP for each column of H containing a bounded maximum number of non-zero entries and minimizing the ℓ_1 distance between the reconstructed sample point and real sample point. However, this method introduced a time-consuming combinatorial problem to solve, and the ℓ_1 distance used in MIP is different from the KL divergence used in the multiplicative gradient updates when factorizing A.

Alloying. The alloying rule states that if shifting is detected at a sample point, the Gibbs phase rule loses a degree of freedom. The number of possible phases for this sample point is then bounded by $G-1$. Furthermore, the shifting parameters λ change continuously across the wafer in the presence of alloying. Similar to the Gibbs constraint, the *IAFD* algorithm repairs solutions by solving a set of MIPs. These MIPs embed the alloying constraint and minimize the absolute distance between the current solution and the measurement data A. Once again, a hard combinatorial problem has to be solved and the objective function is not the same as the one used for optimizing W and H.

Connectivity. The last constraint of the phase-mapping problem is the connectivity constraint. This constraint states that the sample locations where a given phase is present are members of a connected region. For this constraint, the *IAFD* algorithm first defines a graph, using the sample point locations, and a Delaunay triangulation [15] of these points. The triangulation gives a graph in which neighborhood relationships are defined. For a sample j, its neighbors constitute a set ω_j. Then a search is performed to find the connected components for each phase. Only the largest component of every phase is kept, and its complement is zeroed out. This procedure is only applied at the end of the algorithm.

3 Constraint Projection for the Phase-Mapping Problem

Multiplicative updates. *PIAFD* is based on the multiplicative update rules for convolutive NMF [16]. These update rules are non-decreasing, though they might not converge to a stationary point of the objective function. Future work will be based on the modified update rules introduced in [14] to eliminate this possibility. In addition, a ℓ_1-penalty term on elements of H is included, to suggest a sparse solution. Note that the scaling indeterminacy between W and H, as $(\alpha W)(\frac{1}{\alpha}H)$ leads to the same reconstruction error for all non-zero α. If no further adjustments were made, this property would lead the algorithm to make the elements of H arbitrarily small to minimize the ℓ_1-penalty. In order to avoid this behavior, a normalized version of W is used to derive the multiplicative update rules. See [17] for details. *PIAFD* method alternates between multiplicative update and projection onto the constraint space. The rest of this section shows how to efficiently project onto the phase-mapping constraints.

Constraint Projection. The projection of a vector y onto a constraint space C is often defined by the following equation:

$$P_C(y) = \arg\min_{x \in C} \|x - y\|_2 \tag{1}$$

Finding the projection of a given point onto a constraint space is often a hard task, but there are some constraints for which efficient algorithms can be defined [18–20]. In this section, we propose to define the projection of each constraint.

Gibbs. First, the projection $P_{Gibbs}(y)$ of the current solution y is the closest point to y which satisfies the Gibbs constraint. This constraint states that no more than G entries in each column of H can have a non-zero value. Each column of H therefore can be projected independently. The closest point satisfying this constraint is the closest point having less or equal to G non-zero values. This point is composed of the G largest elements of the column. Note that the worst-case complexity of finding the G largest component of a vector of size n can be bounded by $O(n + G\log(n))$. Let S_G^j be the vector containing the indexes of the rows, of column j, larger or equal to the G-th largest element of the column. Let $\alpha \in [0, 1]$ be a real value; let the matrix vp be defined by:

$$vp_{ij} = \begin{cases} 1 & \text{if } i \in S_G^j, \\ 1 - \alpha & \text{otherwise} \end{cases} \tag{2}$$

Property 1. *If $\alpha = 1$, then the result of an element-wise multiplication of H and vp respects the Gibbs constraint.*

Property 2. *If $\alpha \in [0, 1]$, then the result of an element-wise multiplication of H and vp is closer to the solution space of the Gibbs constraint than H.*

The parameter α is a relaxation parameter of the constraint. We can set and modify it during the search for a solution. One advantage of having such a parameter is to not drastically modify the current solution, while performing the gradient-based optimization. As shown in the experimental solution, this gives us a great flexibility in practice.

Alloying. The alloying constraint is a conditional constraint. That is, it has to be enforced only when alloying occurs in the data. Alloying occurs if at least one of the phases at a given sample point is shifting. The following equation determines this:

$$Y^j = \sum_{k\in[1,K],n\in\omega_j} H_{kj} \times \max(0, |\lambda_{kj} - \lambda_{kn'}| - \epsilon) \tag{3}$$

If the alloying variable Y^j is bigger than 0, then the jth column loses a degree of freedom regarding the Gibbs constraint. That is, instead of G entries, only $G-1$ are allowed to be non-zero. This behaviour can be incorporated into the vp matrix, which was previously used for enforcing the Gibbs constraint, by modifying it as follow:

$$vp_{ij} = \begin{cases} 1 & \text{if } Y^j = 0 \wedge i \in S^j_G, \\ 1 & \text{if } Y^j > 0 \wedge i \in S^j_{G-1}, \\ 1-\alpha & \text{otherwise} \end{cases} \tag{4}$$

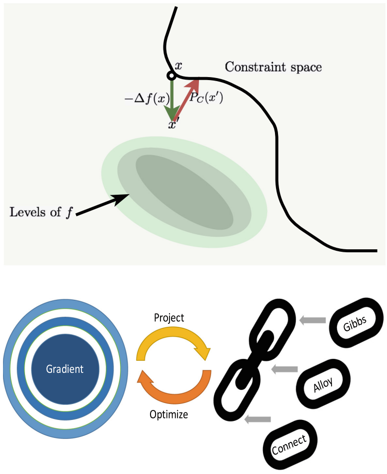
The matrix vp can be used to enforce or relax both Gibbs and alloying constraints.

Connectivity. The existing algorithm for enforcing the connectivity constraint set entries of H that do not belong to the largest connected component to zero. As for the two previous constraints, we can relax the connectivity constraint by multiplying these values by $1-\alpha$ instead of 0. When $\alpha = 1$, the exact constraint is enforced. But when $\alpha \in (0, 1)$, the new point is only closer to, but not equal to the exact projection. Thus, the constraint is not enforced, but the solution is moved closer to the constraint space.

Let C_k be the set of column indices indicating the largest connected component of basis k. As for the two previous constraints, we can use a similar vp matrix:

$$vp_{ij} = \begin{cases} 1 & \text{if } i \in C_i \wedge Y^j = 0 \wedge i \in S^j_G, \\ 1 & \text{if } i \in C_i \wedge Y^j > 0 \wedge i \in S^j_{G-1}, \\ 1-\alpha & \text{otherwise} \end{cases} \tag{5}$$

The matrix vp is finally used to enforce or relax the Gibbs, alloying, and connectivity constraints. This leads to a simple two-step update for solving the constrained NMF problem. Figure 2 shows the projection method and a high-level schematic of our method. *PIAFD* starts with unconstrained gradient updates and then interleaves enforcing relaxed constraints and multiplicative updates of the matrices. In each iteration, W and H are first alternatively updated till convergence or for a certain amount of times, whichever happens, and then the relaxed constraints are enforced subsequently. Hard constraints are enforced at the end of the algorithm (Fig. 2).



Algorithm 1: The algorithm of *PIAFD*

```
Input  : $A$, $max\_iters$
Output : $W$, $H$
Initialization;
while not converge do
    alternatively update $W$ and $H$ ($\alpha = 0$);
$n\_iters \leftarrow 0$;
while not converge and $n\_iters < max\_iters$ do
    alternatively update $W$ and $H$ and then enforce relaxed Gibbs, Alloying, Connectivity constraints ($0 < \alpha < 1$);
    $n\_iters \leftarrow n\_iters + 1$;
while not converge do
    alternatively update $W$ and $H$ and then enforce exact constraints ($\alpha = 1$);
```

Fig. 2. (Upper left) gradient step (green arrow) followed by a projection (red arrow). Solutions updated by the gradient step might leave the constraint space. They are dragged back to the constraint space through a projection. (Lower left) workflow of the *PIAFD* method. It interleaves gradient updates and interleaves projections. (Right) the pseudocode of *PIAFD*. (Color figure online)

4 Experiments

In this section, we compare our method, *PIAFD*, against other methods for solving the phase-mapping problem. Namely, we compare the proposed method against *IAFD* [9], *CombiFD* [5], and *AgileFD* [16]. *CombiFD* is one of the early, purely combinatorial methods for solving the phase-mapping problem. It uses MIP to encode all constraints, and updates solutions in an iterative fashion. However, it is not very efficient, especially compared to more recent methods. *AgileFD* is based on a matrix factorization framework to acquire solutions more efficiently, but it does not encode all the constraints. *IAFD* refines *AgileFD* by alternating multiplicative updates of the matrices and constraint projections using a MIP.

All experiments were run on a server containing 24 nodes, where each node contained an Intel x5650 2.67 GHZ, 48 GB of memory, and was running CentOS 7. The data sets are from real-world experiments from materials science.

***Runtime*.** All methods were tested on 8 different datasets [21] to compare their runtime. In these instances, the inner dimension K of W and H is 6, and $G = 3$. At the beginning of the optimization procedure, α is set to 0 and the optimization is then run to convergence. Subsequently, α is increased to 0.15, a value heuristically found to provide the best reconstruction error, and finally to 1 after a preset number (1000) of iterations is reached. Once α is equal to 1, the algorithm is run to convergence yielding a solution to the constrained optimization

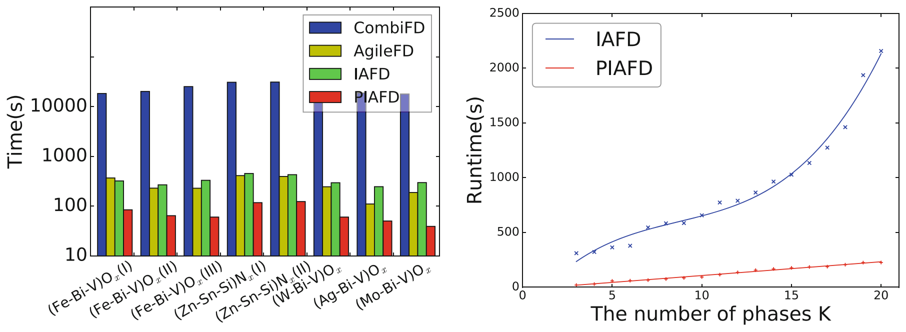


Fig. 3. Runtime comparison of the different methods. The bar chart on the left shows the runtime of *CombiFD*, *AgileFD*, *IAFD*, and *PIAFD* in seconds. The plot on the right compares the runtime of *IAFD*, and *PIAFD* as a function of K, the inner dimension of the matrices W and H.

problem. Figure 3 (left) shows that our new method improves on the runtime of previous methods by at least an order of magnitude.

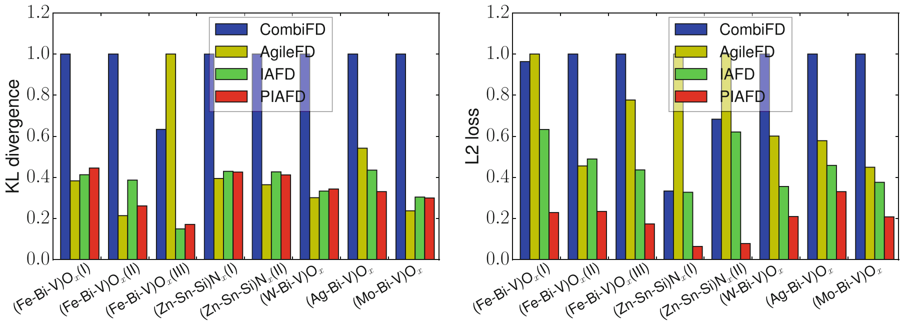


Fig. 4. Accuracy comparison of the different methods (*CombiFD*, *AgileFD*, *IAFD*, and *PIAFD*). The left bar chart shows the minimal error attained for each method using the KL-divergence as the objective function. The right bar chart is similar, just that the ℓ_2-norm was used as the objective function.

Figure 3 (right) shows the runtime of *IAFD* and *PIAFD* as a function of K. Since *IAFD* depends on MIP, it has a worst-case time complexity of $O(\binom{K}{G})$ for each column of H. In other words, if $G = 3$, the worst-case runtime behavior scales proportionally to K^3. Therefore, *IAFD* does not scale well with K. In contrast, *PIAFD* scales linearly in K, as demonstrated by Fig. 3. This improvement in the asymptotic scaling of the constraint projection is crucial for advancing materials science, as the cutting edge of the field deals with datasets of ever increasing size and complexity.

Table 1. Constraint satisfaction comparison of the different methods for the alloying and connectivity constraints. The system column denotes the particular dataset used for each row. The entries in the alloying constraint column are percentages of the columns of H which satisfy the constraint after the respective method has terminated. The entries in the connectivity constraint column correspond the percentages of the phases in W which satisfy the connectivity constraint.

System	Alloying constraint				Connectivity constraint			
	CombiFD	AgileFD	IAFD	PIAFD	CombiFD	AgileFD	IAFD	PIAFD
(Fe-Bi-V)O_x(I)	0.44	0.90	**1.00**	**1.00**	**1.00**	0.17	**1.00**	**1.00**
(Fe-Bi-V)O_x(II)	0.47	0.76	**1.00**	**1.00**	**1.00**	0.17	**1.00**	**1.00**
(Fe-Bi-V)O_x(III)	0.87	0.98	**1.00**	**1.00**	0.83	0.00	**1.00**	**1.00**
(Zn-Sn-Si)N_x(I)	0.98	**1.00**	**1.00**	**1.00**	**1.00**	0.17	**1.00**	**1.00**
(Zn-Sn-Si)N_x(II)	0.95	0.98	**1.00**	**1.00**	**1.00**	0.00	**1.00**	**1.00**
(W-Bi-V)O_x	0.51	0.95	**1.00**	**1.00**	**1.00**	0.00	**1.00**	**1.00**
(Ag-Bi-V)O_x	0.19	0.90	**1.00**	**1.00**	0.67	0.00	**1.00**	**1.00**
(Mo-Bi-V)O_x	0.55	0.94	**1.00**	**1.00**	0.83	0.00	**1.00**	**1.00**

Accuracy. To compare the different methods in terms of accuracy, we benchmarked all methods using two popular objective functions: the KL-divergence and the ℓ_2-norm. Figure 4 (left) shows that *PIAFD* consistently finds solutions with a KL-divergence comparable to the next best method (*IAFD*). If the ℓ_2-norm is used, *PIAFD* significantly outperforms all other methods in terms of accuracy, as evidenced by Fig. 4 (right).

Constraint Satisfaction. Our goal is to have solutions which satisfy the constraints of the phase-mapping problem. Table 1 demonstrates the ability of the different methods to yield solutions in the constraint space. All the methods respect the Gibbs constraint, so it is not shown in the table. Notably, both *IAFD* and *PIAFD* satisfy all constraints. *CombiFD* encodes the constraints using MIP. As the complexity of the constraints increases, it takes more time to find a satisfactory solution. Within the maximum wall time of the server (4 h) *CombiFD* fails to find a solution satisfying all the constraints but still respects Gibbs and connectivity constraints. The solution generated by *AgileFD* neither satisfy the alloying nor the connectivity constraint, due to the model's limited expressiveness.

5 Conclusion

This paper proposed *PIAFD*, a new method for solving the phase-mapping problem. using a novel algorithm for projecting solutions onto the constraint space. Crucially, the method tends to continuously move the optimization variables towards the constraint space, while minimizing the objective function with a gradient-based optimization procedure. The experimental section shows that this new method is orders of magnitude faster than existing methods and, depending on the choice of objective function, gives comparable or more accurate solutions. Because of this improvement in runtime, problems of previously intractable size become feasible. Consequently, the method has the potential of contributing to accelerating discoveries in materials science.

Acknowledgments. Work supported by an NSF Expedition award for Computational Sustainability (CCF-1522054), NSF Computing Research Infrastructure (CNS-1059284), NSF Inspire (1344201), a MURI/AFOSR grant (FA9550), and a grant from the Toyota Research Institute.

References

1. Lee, D.D., Seung, H.S.: Learning the parts of objects by non-negative matrix factorization. Nature **401**(6755), 788 (1999)
2. 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). https://doi.org/10.1007/978-3-540-30110-3_63
3. Suram, S.K., Newhouse, P.F., Zhou, L., Van Campen, D.G., Mehta, A., Gregoire, J.M.: High throughput light absorber discovery, part 2: establishing structure-band gap energy relationships. ACS Comb. Sci. **18**(11), 682–688 (2016)
4. Vavasis, S.A.: On the complexity of nonnegative matrix factorization. SIAM J. Optim. **20**(3), 1364–1377 (2009)
5. LeBras, R., Damoulas, T., Gregoire, J.M., Sabharwal, A., Gomes, C.P., van 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). https://doi.org/10.1007/978-3-642-23786-7_39
6. Gregoire, J.M., Dale, D., Kazimirov, A., DiSalvo, F.J., van Dover, R.B.: High energy x-ray diffraction/x-ray fluorescence spectroscopy for high-throughput analysis of composition spread thin films. Rev. Sci. Instrum. **80**(12), 123905 (2009)
7. Colthup, N.: Introduction to Infrared and Raman Spectroscopy. Elsevier, Amsterdam (2012)
8. Mørup, M., Schmidt, M.N.: Sparse non-negative matrix factor 2-D deconvolution. Technical report (2006)
9. Bai, J., Bjorck, J., Xue, Y., Suram, S.K., Gregoire, J., Gomes, C.: Relaxation methods for constrained matrix factorization problems: solving the phase mapping problem in materials discovery. In: Salvagnin, D., Lombardi, M. (eds.) CPAIOR 2017. LNCS, vol. 10335, pp. 104–112. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-59776-8_9
10. Ermon, S., Le Bras, R., Suram, S.K., Gregoire, J.M., Gomes, C.P., Selman, B., van Dover, R.B.: Pattern decomposition with complex combinatorial constraints: application to materials discovery. In: AAAI, pp. 636–643 (2015)
11. Ermon, S., Le Bras, R., Gomes, C.P., Selman, B., van 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). https://doi.org/10.1007/978-3-642-31612-8_14
12. Gregoire, J.M., Van Campen, D.G., Miller, C.E., Jones, R.J.R., Suram, S.K., Mehta, A.: High-throughput synchrotron X-ray diffraction for combinatorial phase mapping. J. Synchrotron Radiat. **21**, 1262–1268 (2014)

13. Hattrick-Simpers, J.R., Gregoire, J.M., Kusne, A.G.: Perspective: composition-structure-property mapping in high-throughput experiments: turning data into knowledge. APL Mater. **4**, 053211 (2016)
14. Lin, C.-J.: On the convergence of multiplicative update algorithms for nonnegative matrix factorization. IEEE Trans. Neural Netw. **18**(6), 1589–1596 (2007)
15. Lee, D.-T., Schachter, B.J.: Two algorithms for constructing a Delaunay triangulation. Int. J. Comput. Inf. Sci. **9**(3), 219–242 (1980)
16. Xue, Y., Bai, J., Le Bras, R., Rappazzo, B., Bernstein, R., Bjorck, J., Longpre, L., Suram, S.K., van Dover, R.B., Gregoire, J.M., et al.: Phase-Mapper: an AI platform to accelerate high throughput materials discovery. In: AAAI, pp. 4635–4643 (2017)
17. Le Roux, J., Weninger, F.J., Hershey, J.R.: Sparse NMF-half-baked or well done? Mitsubishi Electric Research Labs (MERL), Cambridge, MA, USA, Technical report no. TR2015-023 (2015)
18. Duchi, J., Shalev-Shwartz, S., Singer, Y., Chandra, T.: Efficient projections onto the ℓ_1-ball for learning in high dimensions. In Proceedings of the 25th International Conference on Machine Learning, pp. 272–279. ACM (2008)
19. Condat, L.: Fast projection onto the simplex and the l_1 ball. Math. Program. **158**(1–2), 575–585 (2016)
20. Perez, G., Barlaud, M., Fillatre, L., Régin, J.-C.: A filtered bucket-clustering method for projection onto the simplex and the ℓ_1 ball. In: Colloque GRETSI, Juan-les-Pins, France (2017)
21. 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)