Constraint Reasoning and Kernel Clustering for Pattern Decomposition With Scaling

Ronan LeBras[1], Theodoros Damoulas[1], John M. Gregoire[2]
Ashish Sabharwal[3], Carla P. Gomes[1], and R. Bruce van Dover[4]

[1] Dept. of Computer Science, Cornell University, Ithaca, NY 14853, USA
[2] School of Engr. and Applied Sciences, Harvard University, Cambridge, MA 02138
[3] IBM Watson Research Center, Yorktown Heights, NY 10598, USA
[4] Dept. of Materials Science and Engr., Cornell University, Ithaca, NY 14853, USA

Abstract. Motivated by an important and challenging task encountered in material discovery, we consider the problem of finding K basis patterns of numbers that jointly compose N observed patterns while enforcing additional spatial and scaling constraints. We propose a Constraint Programming (CP) model which captures the exact problem structure yet fails to scale in the presence of noisy data about the patterns. We alleviate this issue by employing Machine Learning (ML) techniques, namely kernel methods and clustering, to decompose the problem into smaller ones based on a global data-driven view, and then stitch the partial solutions together using a global CP model. Combining the complementary strengths of CP and ML techniques yields a more accurate and scalable method than the few found in the literature for this complex problem.

1 Introduction

Consider a setting where our goal is to infer properties of a system by observing patterns of numbers (e.g., discretized waveforms, locations of peak intensities in a signal, etc.) at N sample points. Suppose these N patterns are a combination of K unobserved basis patterns. The *pattern decomposition problem* seeks to identify, given patterns at the N sample points as input, K basis patterns that generate the observed patterns and which of these basis patterns appear at any given sample point. The sample points are often embedded in the Euclidean space, enforcing a constraint that points near each other should generally be explained by a similar subset of patterns (except for a few transition boundaries).

Variants of this problem arise in a number of scenarios. For example, in the well-known *cocktail party problem*, the observed patterns are mixtures of voices of people as recorded by various microphones and the task is to decompose the signal at each microphone into the voices of individuals – the basis patterns – contributing to that signal. The microphones observe a spatial correlation, in the sense that if person's voice is heard at a microphone, it is likely that it is also heard at a neighboring microphone but not at a far away one.

Problems such as these fall under the category of *source separation problems*. Typically, purely data-driven methods are used for these, relying heavily on pattern recognition from a global analysis of the available data. A limitation of this

approach, however, is that it makes it very difficult to enforce *hard constraints*. While one may argue that the spatial and other requirements in problems such as the cocktail party problem are somewhat "soft", the setting we consider in this paper is motivated by a materials science problem that imposes hard constraints dictated by physics. When solving this problem, even slight deviation from the requirements of the underlying physics makes "solutions" meaningless. Moreover, in this setting, observed patterns are allowed to be superpositions of basis patterns *stretched* by a small multiplicative scaling factor, leading to what we call the *Decomposition Problem With Scaling*. This problem generalizes a known NP-complete problem, namely, the Set Basis Problem [19].

Faced with the challenge of handling hard constraints and scaling factors, we propose a Constraint Programming (CP) approach to solve our variant of the pattern decomposition problem. Our CP formulation captures the desired constraints in a detailed and exact fashion. However, as expected, it does not scale well with problem size once we introduce errors and noise in the input data. To alleviate this issue, we turn to Machine Learning (ML) and use kernel-based clustering as a way to guide the CP solver by creating multiple smaller sub-problems within its reach. After solving these smaller sub-problems with CP, we take a step back and combine the multiple partial solutions into a consistent global solution, using the original, global CP model.

Our contributions include bringing this intriguing and challenging problem to the CP community, providing a CP model for it, and enhancing the global scalability of the model while preserving local accuracy by exploiting ML methods for designing a divide-and-conquer approach. Using data from our material discovery application as a testbed, we demonstrate that the proposed hybrid ML-CP approach yields more accurate and meaningful solutions than existing, mostly data-driven approaches.

1.1 Pattern Decomposition for Material Discovery

The particular variant of the pattern decomposition problem considered in this paper is motivated by an important application in the area of material discovery. Specifically, a detailed analysis of libraries of inorganic materials has become an increasingly useful technique in this line of work, as evident from the number and variety of recently published methods for combinatorial materials research [e.g., 2, 16]. These libraries can be screened for a desired property, providing an understanding of the underlying material system. This is an important direction in *computational sustainability* [8], and aims to achieve the best possible use of our available material resources.

A fundamental property of inorganic materials is their crystallographic phase, and thus creating a "phase map" of an inorganic library across various compositions is a key aspect of combinatorial materials science. Often, correlations between the phase map and other material properties provide important insights into the behavior of the material system. For example, a recent study of a Platinum-Tantalum library revealed an important correlation between crystallographic phase and improved catalytic activity for fuel cell applications [10].

The most common technique for creating such a phase map is to first use Xray diffraction to generate diffraction patterns (continuous waveforms) for sample points with different compositions. Inferring the phase map from these diffraction patterns is then done using a laborious manual inspection. Doing this automatically, without human interaction, is a long standing problem in combinatorial crystallography. Several recent algorithms have been proposed which correctly solve the phase map for limited cases [3, 4, 14, 15]. In 2007, Long et al. [15] suggested a *hierarchical agglomerative clustering* (HAC) approach which aims to cluster the observed patterns that involve the same subset of basis patterns, but relies on a manual inspection in order to discover the actual basis patterns. In a follow-up paper, Long et al. [14] applied *non-negative matrix factorization* (NMF), which approximates (through gradient descent) the observed diffraction patterns with a linear combination of positive basis patterns. A main limitation of both approaches lies in the assumption that peaks of a phase will always appear at the same position and with the same relative intensities in any pattern. However, the position and intensity of diffraction peaks typically *scale* as a function of composition due to chemical alloying. Also, these approaches are unable to enforce hard constraints such as connectivity requirements.

Our goal is to take the actual physics behind the crystallographic process (e.g., the nature of scalings in the patterns and connectivity) into account in order to design a robust and scalable method for solving this problem in the presence of experimental noise.

2 Problem Description

From a computational perspective, we are interested in solving the following constraint reasoning (and optimization) problem. We will define this problem over rational numbers, $\mathbb{Q}$, rather than reals as this ensures that the problem is within NP; if there is a solution, using rational numbers will allow us to compactly represent and verify its correctness. We will refer to a set $P \subseteq \mathbb{Q}^+$ of positive rationals as a *pattern* over positive rational. For a scaling factor $s \in \mathbb{Q}$, let us define the *scaled pattern* sP as the pointwise scaled version of the pattern P, namely, $sP = \{sp \mid p \in P\}$.

Informally speaking, the problem is the following. Suppose we are given a graph over N vertices and, associated with each vertex v_i, a pattern P_i consisting of a finite set of numbers. Given $K \leq N$, the goal is to decompose these N patterns into K patterns that form a "basis" in the following sense: each P_i must be the union of at most M scaled basis patterns (i.e., scaled versions of at most M basis patterns must *appear* at each vertex), and the subgraph formed by the vertices where the k-th basis pattern appears must be connected.

The problem, illustrated in Figure 1, is formally defined as follows:

Definition 1 (Problem: Pattern Decomposition With Scaling). *Let*

- $G = (V, E)$ *be an undirected graph with* $V = \{v_1, \ldots, v_N\}$,
- $\mathcal{P} = \{P_1, \ldots, P_N\}$ *be a collection of* N *patterns over a finite set* $S \subseteq \mathbb{Q}^+$,

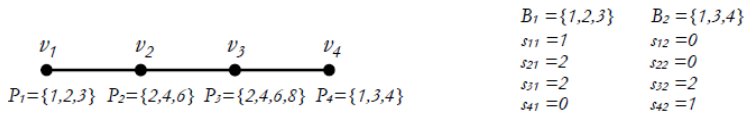

Fig. 1. Left: Toy example illustrating Def. 1. Right: Solution for $M = K = 2$ and $\delta = 2$

.

– $M \leq K \leq N$ *be positive integers, and* $\delta \geq 1$ *be a rational.*

Determine whether there exists a collection $\mathcal{B}$ *of* K basis patterns *over* S *and* scaling factors $s_{ik} \in \{0\} \cup [1/\delta, \delta]$ *for* $1 \leq i \leq N, 1 \leq k \leq K$, *such that:*

(a) $\forall i$*: P_i is the union of scaled basis patterns, i.e.,* $P_i = \bigcup_{k=1}^{K} s_{ik} B_k$*;*
(b) $\forall i$*: the number of basis patterns with a non-zero coefficient at vertex* v_i *is at most* M*, i.e.,* $|\{k \mid s_{ik} > 0\}| \leq M$*; and*
(c) $\forall k$*: the subgraph of* G *induced by* $V_k = \{v_i \in V \mid s_{ik} > 0\}$ *is connected.*

Noisy Data. In practice, one may not have accurate information about the pattern P_i at each vertex v_i. Indeed, in our material discovery application to be discussed shortly, it is very common for several types of *noise* to be present in the patterns provided as input to this problem. For the purposes of this paper, we will assume that *there may be false negatives in the N observed patterns, but no false positives.* In other words, our models will be designed to tolerate *missing elements* in patterns, by relaxing the first condition in the problem definition to $P_i \subseteq \bigcup_k s_{ik} B_k$ rather than requiring a strict equality. Note that this relaxation severely limits the propagation that a constraint enforcing this condition might be able to perform, as we can no longer remove an element from a candidate basis pattern B_k even if that element (appropriately scaled) does not appear in the observed pattern P_i. We will discuss this issue in more detail in Section 3.

Further, we will make the assumption that for every basic pattern, there is *at least one recurrent element* that is not missing in every observed pattern involving this basic pattern. This assumption is often quite realistic in many applications where elements of a pattern are, for example, locations of peaks in a waveform. Indeed, even though the highest peak of a given basic pattern might not be observed as the highest one in each pattern where it appears, it is quite unlikely to completely disappear due to noise.

Other Dimensions to the Elements of a Pattern. Depending on the particular application under consideration, the elements of a pattern may have associated with them other dimensions as well that an algorithm may be able to exploit. E.g., when elements correspond to "locations" of peaks in a waveform, they naturally have *height* and *width* of the corresponding peaks associated with them as well. We will use these additional dimensions, specifically height, in the material discovery application experiments in order to control the amount of tolerable error. The machine learning part of our hybrid method will also exploit height and width indirectly when computing the similarity between patterns.

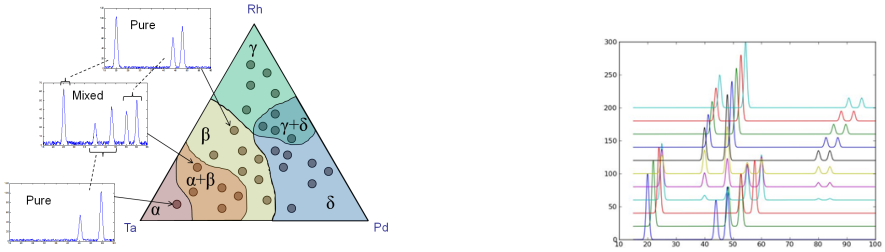

Fig. 2. Left: Pictorial depiction of the problem, showing 29 sample locations each corresponding to a composition and associated with an observed x-ray diffraction pattern. The green, blue, yellow, and red colored regions denote pure phase regions. Also shown are two mixed regions, formed by the overlap of $\alpha + \beta$ and $\gamma + \delta$. Right: Multiplicative *shift* in waveforms as one moves from one point to an adjacent one; waveforms are shown stacked up vertically to highlight the shift.

2.1 Motivating Application: Phase Identification in Materials

A combinatorial method for discovering new materials involves sputtering three metals (or oxides) onto a silicon wafer, resulting in a so-called thin film. The goal is to identify structural regions in thin films. Any location on a thin film corresponds to a crystal with a particular composition of the three sputtered metals (or oxides); a number of such locations are sampled during experimentation, as shown with black dots in Figure 2. The structural information of this crystal lattice is usually characterized by its x-ray diffraction pattern – a continuous waveform obtained by electromagnetic radiation. The resulting diffraction pattern associated with each location represents the intensity of the electromagnetic waves as a function of the scattering angle of radiation (see Figure 2).

The pattern observed at any location is often a superposition of a number of basis patterns, known as *phases*, possibly *stretched* by a small multiplicative scaling factor; the shifts are depicted in the right panel of Figure 2 where adjacent lines correspond to waveforms observed at adjacent locations. In other words, a thin film involves a small number of basic crystallographic phases, and the crystal corresponding to each sampled location lies either in a *pure region* comprising of just one phase, or a *mixed region* made from a superposition of multiple, possibly stretched phases (e.g., the waveform shown in the middle of the left panel of Figure 2 is the superposition of the ones shown above and below it).

Given the diffraction patterns at N sampled locations, the problem is to compute the most likely *phase map*, i.e., the set of phases that are involved at any location of the thin film and in which proportion. A sub-problem, often considered in the literature [e.g., 15], is to cluster the sampled locations such that locations in each cluster are superpositions of the same subset of phases.

When three metals are used for this experiment, the result is referred to as a *ternary diagram*. A physical constraint in a ternary diagram is that independent of the total number of phases present, the number of phases that may appear at any given location is at most 3. Furthermore, if 3 phases do appear at a

location, then it does not leave any degree of freedom for the aforementioned shifts to happen, i.e., only pure regions or mixed regions comprising 2 phases exhibit shifting.

We can cast this problem as the Pattern Decomposition With Scaling problem discussed earlier, with an additional constraint enforcing scaling factors to be precisely 1 when 3 phases appear at a location. The idea is to pre-process these x-ray diffraction patterns by performing *peak detection*, for which reliable techniques are available in the context of materials science. This results in a finite set of scattering angles – a pattern in our earlier notation – at which peaks are observed at a given sampled location. Specifically, N is the number of sampled locations, G is obtained by Delaunay triangulation over the sampled points based on their given x-y coordinates on the thin film, $\mathcal{P}$ is the set of such patterns associated with each location, $M = 3$, δ is typically 1.15 (i.e., allowing shifts by a maximum scaling factor of 15%), K is the number of underlying phases or basis patterns we are interested in discovering. Without loss of generality, we fix S to be the set of all scattering angles (i.e., pattern elements) at which a peak is observed in the sampled locations.

In general, the goal from a material discovery perspective is two-fold: explain the diffraction patterns observed at the N locations with the fewest number K of phases, while also minimizing the error resulting from missing peaks and other noise in the data. We will assume for the purposes of this paper that although we might miss some peaks (i.e., false negatives), the scattering angle where we do observe a peak is accurate (i.e., no false positives). Given the small range of K in reality (typically 5-8), we will take K to be a parameter of the problem and attempt to minimize error introduced due to missing peaks. As a practically relevant *objective function*, we use the *sum of the estimated heights of missing peaks*. Note that "heights" and "peaks" are not part of the formal definition of the satisfaction problem, Pattern Decomposition With Scaling. Nonetheless, this data is readily available for this material discovery application and we use it to enhance the problem with a realistic objective function. In fact, when discussing the machine learning part to boost scalability, we will use for computing "similarity" between locations not only the scattering angles where peaks appear but also a discretized version of the complete waveforms.

2.2 NP-Completeness

In order to prove that the Pattern Decomposition With Scaling problem as defined above is NP-hard, we simplify it in three steps to what is called the Set Basis Problem, which is known to be NP-complete. First, let $M = K$, i.e., allow the K basis patterns to appear at any vertex. Second, let the underlying graph G be a clique, thereby trivially satisfying the third condition in the problem definition (subgraph connectivity). Finally, let $\delta = 1$, thereby forcing all scaling factors s_{ij} to be either 0 or 1. With these three modification steps, our problem simplifies to what is known in the literature as the Set Basis Problem, defined as follows and known to be NP-complete [19]:

Definition 2 (Set Basis Problem [19]). *Given a collection* $\mathcal{P} = \{P_1, \ldots, P_N\}$ *of* N *subsets of* S *and an integer* K *satisfying* $2 \leq K \leq N$*, is there a collection* $\mathcal{B}$ *of* K *subsets of* S *such that for all* $1 \leq i \leq N$ *there exist* $\mathcal{B}_i \subseteq \mathcal{B}$ *such that* $P_i = \cup_{B \in \mathcal{B}_i} B$*?*

To see that the Pattern Decomposition With Scaling problem is within NP, we observe that given a candidate solution to the problem, namely a collection $\mathcal{B}$ of K subsets of S and scaling factors $s_{ik} \in \mathbb{Q}$ for $1 \leq i \leq N, 1 \leq k \leq K$, one can easily verify in polynomial time that all requirements of the problem are satisfied. Note that defining the problem over $\mathbb{Q}$ rather than the reals ensures that if an instance has a solution, then there is also one with all $s_{ik} \in \mathbb{Q}$, allowing succinct representation and efficient verification of a candidate solution.

Together, these imply that this problem is NP-complete.

3 A Constraint Programming Formulation

We first describe a CP formulation of this problem assuming no errors, i.e., no missing elements in patterns nor experimental noise in the element value. A natural way to encode this problem is to have one variable for each element of each of the N patterns indicating which of the K basic patterns "explains" this element. This formulation, however, results in too many variables and also fails to account for overlaps, i.e., that an element of an observed pattern may in fact be explained by *multiple* basic patterns (since we take the union of basic patterns in the problem definition). An alternative formulation can try to analyze the N given patterns to identify which elements are shared between neighboring vertices of G, and use this as a basis for creating basis patterns. This formulation too results in too many variables and constraints. We present below a formulation that proved to be the most successful. This formulation explicitly uses the underlying basis patterns as the central variables, and merges sets of large numbers of constraints into global ones in order to reduce memory consumption.

In a preprocessing step, we compute the set r_{ij} as P_i normalized by its j^{th} element. For example, if P_5 corresponds to $\{1, 2, 4\}$, then $r_{5,2}$ becomes $\{1/2, 1, 2\}$.

Variables. We model whether a basis pattern k is present in a pattern P_i using a *decision variable* p_{ki}. According to the assumption mentioned in Section 2, there is at least one element of any basis pattern that appears in all sample points in which this basis pattern is present. As a result, if we use this element as a normalizing one, the set of elements of this basis pattern becomes the same in all of these sample points. We represent the normalizing element of basis pattern k in sample point P_i as p_{ki}, whose domain is $\{0, 1, ..., |P_i|\}$ and where value 0 denotes that basis pattern k is not present in pattern P_i. Furthermore, *auxiliary Boolean variable* a_{ki} indicates whether basis pattern k appears in P_i while *auxiliary set variable* q_k represents the normalized elements of pattern k and initially ranges over all possible scaled elements. The domain representation used for the q_{ik} variables is the classical subset-bound, yet more advanced representations ([see eg. 7, 11]) might further enhance the model.

Constraints. We first express the relationship between the auxiliary variables a_{ki} and the decision variables p_{ki} as follows:

$$(a_{ki} = 0) \iff (p_{ki} = 0) \qquad \forall\, 1 \leq k \leq K, 1 \leq i \leq n \tag{1}$$

At this point, we can directly enforce that a pattern has to be composed of at least one basis pattern, and at most M:

$$1 \leq \sum_{s=1}^{K} a_{si} \leq M \qquad \forall\, 1 \leq i \leq n \tag{2}$$

Next, anytime a pattern P_i involves a particular basis pattern k, every element of k has to match one of the normalized elements of P_i. Formally:

$$(p_{ki} = j) \Rightarrow (q_k \subseteq r_{ij}) \qquad \forall\, 1 \leq k \leq K, 1 \leq i \leq n, 1 \leq j \leq |P_i| \tag{3}$$

Nonetheless, in order to fully determine q_k from the p_{ki}'s, we require that all elements of a pattern appear in one of the basis patterns that compose this point. First, if a pattern is made of only one basis pattern, their elements should be identical, up to a scaling factor. It means that if p_{ki} is set to be equal to j, then r_{ij} also has to be a subset of q_k. Second, if a pattern P_i is made of two basic patterns k and k', then every element of P_i has to appear in q_k or in q'_k, when normalized by their respective scaling factor. The first case translates into:

$$(p_{ki} = j \wedge \sum_{s=1}^{K} a_{si} = 1) \Rightarrow (r_{ij} \subseteq q_k) \qquad \forall\, 1 \leq k \leq K, 1 \leq i \leq n, 1 \leq j \leq |P_i| \tag{4}$$

while the second one entails the following equation:

$$(p_{ki} = j \wedge p_{k'i} = j' \wedge \sum_{s=1}^{K} a_{si} = 2) \Rightarrow \big(member(r_{ij}[j''], q_k) \vee member(r_{ij'}[j''], q_{k'})\big)$$
$$\forall\, 1 \leq k, k' \leq K, 1 \leq i \leq n, 1 \leq j, j', j'' \leq |P_i| \tag{5}$$

Similarly, we derive constraints for points that are made of g basis patterns, where $3 \leq g \leq M$. Then, we guarantee that the scaling factors of a basis pattern belong to a valid range. For two patterns to be composed of the same basis pattern, these constraints require that the two respective normalizing elements are not too far apart in the pattern. This step relies as well on a preprocessing step of the data, in order to compute the relative distances and to post the required constraints. For a given $\delta \geq 1$, we consider that this preprocessing step yields a set $\Phi = \{(i, j, i', j') \mid \frac{P_i[j]}{P'_i[j']} < 1/\delta \vee \frac{P_i[j]}{P'_i[j']} > \delta, i < i'\}$ of pairs of elements that do not satisfy this property (typically $\delta \leq 1.15$). It yields:

$$(p_{ki} = j) \Rightarrow (p_{ki'} \neq j') \qquad \forall\, 1 \leq k \leq K, (i, j, i', j') \in \Phi \tag{6}$$

Finally, we implement a special-purpose global constraint, called *basisPattern-Connectivity* which maintains the set of basis patterns in every connected component. Formally, if $a_{ki_1} = 1$ and $a_{ki_t} = 1$, then there exists an undirected path $i_1 \rightarrow i_2 \cdots \rightarrow i_t$ such that $a_{ki_u} = 1$ for all $1 \leq u \leq t$. We could perform propagation based on component and bridge information [see 13, 17], however in practice this extra filtering does not seem to justify the added overhead for our

particular problem setting. Instead we simply make sure that the aforementioned statement is not violated. We define this constraint as:

$$basisPatternConnectivity(\{a_{ki} | 1 \leq i \leq n\}) \qquad \forall\ 1 \leq k \leq K \qquad (7)$$

During search, the branching variables are the p_{ki}s. The variable ordering using an arbitrary BFS on G to statically order the vertices v_i, and dynamically select k such that a neighbor of v_i has set its phase k, proved to be the most successful.

Symmetry Breaking. In order to break symmetries, we systematically assign either an already existing basis pattern or the lowest one available. This means that for example, given the three basis patterns q_1, q_2 and q_3, and considering a new pattern P_i, the variables $p_{5,i}, ..., p_{K,i}$ must be assigned value 0. This is reminiscent, for example, of work on the *Steel Mill Slab Design* [12].

3.1 Handling Errors and Noisy Data

In order to handle missing elements, we adapt constraints (3) to allow for elements of q_k not to appear in P_i, even if the sample point P_i involves basis pattern k. Therefore, the propagation of constraints (3) gets weaker, as we can no longer filter out an element of q_k that is anomalously missing from a sample point (see following section). Also, to avoid a trivial solution in which all possible elements belong to q_k, we introduce an optimization objective that aims to minimize either the overall number of missing elements or the overall relative importance of the missing elements. The importance of an element is application specific, and in the case of our motivating application, a natural way to penalize for a missing peak is to consider its inferred height: the higher the missing peak, the worse the solution. Finally, note that handling missing elements does not affect constraints (4) nor (5), as we do not allow for false positives.

Also, in order to account for noise, we introduce a precision value that represents how far off an observed value can be from its true one. Thus, in constraints (3) to (5), when checking whether an element belongs to a set, we use this precision to assess whether the element appears as a slightly different value.

3.2 Limitations of the Pure CP Approach: Scaling

Although this CP model captures the details of the problem very well, it scales very poorly – especially when errors are introduced in the data in terms of missing peaks. In Table 1, we show the running time of the CP model on (small) instances of various sizes from our material discovery application. Experiments were conducted using IBM ILOG CP Solver version 6.5 deployed on 3.8 GHz Intel Xeon machines with 2GB memory running Linux 2.6.9-22.ELsmp. The time limit used was 1,200 seconds. The observed patterns in each of these instances can, in reality, be explained by $K = 6$ basic patterns. We create simpler versions of the problem by *fixing* some of these basic patterns as a partial solution, leaving $K' \in \{0, 1, \ldots, 6\}$ unknown basic patterns, for each of which we have a row in

Table 1. Scaling of the pure CP model, without errors (pure) and with errors. Rows: num. of unknown basic patterns. Cols: num. of observed patterns. Timeout 1,200 sec.

	$N = 10$		$N = 15$		$N = 18$		$N = 28$		$N = 219$	
	pure	errors	pure	errors	pure	errors	pure	errors	pure	errors
$K' = 0$	0.0	0.0	0.0	0.0	0.0	0.1	0.0	0.1	1.1	3.5
$K' = 1$	0.0	0.1	0.0	0.1	0.0	0.3	0.1	0.4	115.3	343.2
$K' = 2$	0.0	0.2	0.0	0.3	1.0	—	1.4	—	—	—
$K' = 3$	0.5	717.3	0.5	—	384.8	—	276.0	—	—	—
$K' = 4$	668.5	—	824.2	—	—	—	—	—	—	—
$K' = 5$	—	—	—	—	—	—	—	—	—	—

the table ($K' = 6$ is omitted as all instances timed out in this case). As we see, for all N, the instances go from being solvable in a fraction of a second to not solvable in 20 minutes extremely fast. Moreover, when errors are introduced in the form of missing peaks, the scaling behavior becomes worse. Finally, even with a very small problem size such a $N = 10$ and the ideal case of no errors, we cannot solve for all 6 (or even 5) basic patterns. It becomes evident that we need a methodology that can allow us to scale to instances of realistic sizes (e.g., over 100 patterns and with $K' = 6$). This will be the subject of the rest of this paper.

4 Boosting Scalability: Exploiting Kernel-Based Clustering to Guide the CP Formulation

The CP approach discussed thus far attempts to accurately solve the full problem under certain assumptions and, as we saw, fails to scale up to instance sizes of interest as soon as errors are introduced. We discuss in this section how we can leverage ideas from machine learning (ML), specifically kernel-based similarity measures and clustering, in order to make the problem solving task easier for the CP formulation. This integration of the two approaches is inspired by their complementary strengths: While CP techniques are excellent at enforcing detailed constraints at a local level, data-driven ML methods are more robust to noise and good at recognizing global patterns of similarity.

The integration uses the following 4-step "divide-and-conquer" process:

i. use kernel methods to analyze the patterns P_i at a global scale in order to compute a robust similarity measure between pairs of patterns;
ii. use clustering with this similarity measure to "over-segment" the N vertices into J clusters and choose a set $V^{(j)}$ of vertices associated with each cluster based on their distance to the cluster centroid; the vertices in these $V^{(j)}$ are expected to be explained by the same subset of basis patterns;
iii. solve the CP formulation, without the connectivity constraint, on the subgraph induced by the vertices in each $V^{(j)}$ to obtain a partial solution defined by a collection of basis patterns $\mathcal{B}^{(j)}$ each of size at most M; and
iv. glue the basis patterns $\mathcal{B}^{(j)}$ found for the J sub-problems together using a global CP formulation in order to obtain the full set $\mathcal{B}$ of K basis patterns.

4.1 Kernels as Robust Similarity Measures

Assuming D is an upper bound on the number of elements in each input pattern, we will think of the N input patterns as the input dataset $\mathbf{X} \in \mathbb{Q}^{N \times D}$ where each of the N patterns is represented by its D "features" in the D-dimensional space. One can model rich, non-linear relationships between the D base features by instead representing the N patterns in a much larger feature space, one of dimension $L \gg D$. Thus, instead of modeling non-linear relationships directly in D dimensions, one achieves the same effect more easily by still modeling linear relationships but in a much higher dimensional space, using an expanded feature representation $\boldsymbol{\phi}(\mathbf{X}) \in \mathbb{Q}^{N \times L}$.

The problem, of course, is that explicitly constructing this L-dimensional space and working in it can be computationally prohibitive. Kernel methods solve this issue by allowing us to directly model the desired inner product, i.e., the "similarity" measure, $\langle \boldsymbol{\phi}(\mathbf{X}), \boldsymbol{\phi}(\mathbf{X}) \rangle$, and reduce the dimensionality we must deal with while leaving open, in principle, the possibility of even an infinite-dimensional underlying feature expansion ($L = \infty$). Note that this inner product computation results in the construction of a symmetric positive semi-definite $N \times N$ matrix, independent of the dimension L of the much expanded feature space. This matrix is known as the *kernel*.

Typically used generic kernel functions include the *linear* or *cosine* kernel $\mathbf{x}_i^\intercal \mathbf{x}_j$, the *polynomial* kernel $(\mathbf{x}_i^\intercal \mathbf{x}_j + 1)^k$ of degree k, and the *Gaussian* or *radial basis function (RBF)* kernel $\exp(-\frac{||\mathbf{x}_i - \mathbf{x}_j||^2}{2\sigma^2})$. Two specific material-discovery characteristics, however, pose a big challenge when computing similarity between x-ray diffraction waveforms – the inherent peak *shifts* (with multiplicative scaling) and varying peak *intensity* or height levels. This is especially true in cases where the presence of small peaks indicates a novel phase and the existence of a new crystal structure. In order to address this we propose to use the *dynamic time warping* technique [18] to construct a global alignment kernel. Such a kernel was recently used successfully in the context of Bayesian classification [6]. The idea is to construct a kernel from *minimum-cost alignment* of two sequences $\mathbf{x}_i, \mathbf{x}_j$ based on DTW: $\mathbf{k}_{\text{DTW}}(\mathbf{x}_i, \mathbf{x}_j) = \exp\big(-\frac{||\mathbf{c}_i - \mathbf{c}_j||^2}{2\sigma^2}\big)$ where $\mathbf{c}_i$ is the i^{th} row of the minimum-cost alignment matrix. We refer the reader to Damoulas et al. [6] for further details.

4.2 Clustering and Sample Selection

Having constructed the kernel matrix capturing similarity between the patterns at the N vertices of our underlying graph G, we now seek to create small subsets $V^{(j)}, 1 \leq j \leq J$, of the vertices such that all vertices within each $V^{(j)}$ are the unions of the *same* subset of basis patterns, scaled appropriately. The sub-problems induced by these small subsets will be passed on to the CP model to be solved exactly to discover the basic patterns appearing in each of these subsets. Therefore, we would ideally like these subsets to be small enough to be solvable

by the CP model, and at the same time large enough so that if there is shifting involved, the corresponding scaling factor can be recovered by the CP model.

To this end, we use *k-means* algorithm [5] with multiple initializations (centroids of clusters) and the Euclidean distance $d(\mathbf{k}_i, \mathbf{k}_j) = \left(\sum_{n=1}^{N} (k_{in} - k_{jn})^2\right)^{1/2}$ as metric. We over-segment the kernel by choosing a large number of clusters when performing k-means. The final proposed vertices, $V^{(j)}$, are chosen from within each cluster based on their proximity to the cluster centroid.

4.3 Scaling CP: Solving Sub-Problems and Fusing Solutions

Assuming the vertices of $V^{(j)}$ are the unions of the same subset of basis patterns, we know by definition of the problem that at most M basis patterns compose all the patterns of these vertices. Therefore, this is in fact a pattern decomposition problem with scaling by itself, where $N = |V^{(j)}|$ and $K = M$. If this subproblem is within the reach of the CP model (cf. Section 3.2), then we will have uncovered M of the initial K basis patterns. Otherwise, or if our previous assumption about the vertices of $V^{(j)}$ turns out to be wrong, the CP model will simply not be able to solve the instance, and will then consider the next cluster of points. Hence, every cluster may provide up to M basis patterns and contribute to a pool of basis patterns. After taking care of redundancy within this pool (which, is in the worst case, exponential in M), the pool is made of at most K basis patterns, and is used to initialize the basis patterns of the global CP model, thus typically becoming a much simpler problem (again, cf. Section 3.2).

5 Empirical Validation

In order to evaluate the performance of the hybrid method described above, we use our material discovery application as the testbed. As discussed in Section 3.2, the pure CP approach suffers from very poor scaling. On the other end, data-driven approaches such as non-negative matrix factorization (NMF) used in the literature [14] for such problems suffer, as we will show, from low accuracy – to the point that "solutions" found by them for material discovery instances can be meaningless. Our hybrid method avoids both of these extreme kinds of failures, in scaling and in accuracy.

Instance Generation. We use the same underlying known phase map for the Al-Li-Fe system [1] that was used for the instances discussed in Section 3.2. Specifically, this is a ternary system composed of 6 phases or basis patterns, $\alpha, \beta, \gamma, \delta, \epsilon$, and ζ; see Figure 3 for a pictorial depiction. These 6 phases appear together at various locations in the "triangle" in different combinations to generate 7 mixed regions, such as $\{\alpha, \delta\}$, $\{\alpha, \gamma, \delta\}$, etc. Recall that each location in the ternary diagram corresponds to a certain composition of the three constituent elements, in this case Al, Li, and Fe and these compositions can be sampled at various granularities. For the rest of this paper, we will focus on a realistic instance size, 219, and a smaller instance size, 91.

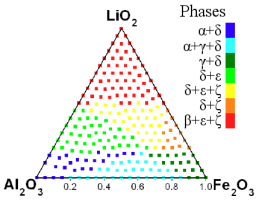

Fig. 3. Ternary system composed of Al, Li, and Fe.

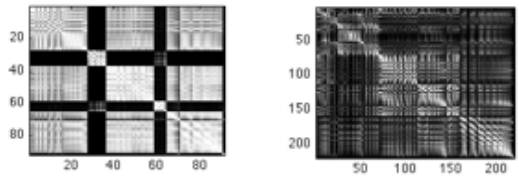

Fig. 4. DTW-Gaussian kernel as a similarity measure. Left: $N = 91$. Right: $N = 219$.

For these instances, we generated synthetic x-ray diffraction data by starting with known diffraction patterns of constituent phases from the JCPDS database [1] with parameters reflecting those of a recently developed combinatorial crystallography technique [9]. This diffraction data was then converted into a set of peaks to generate discrete patterns with typically 20-30 peaks. The effect of experimental noise on the inability to detect low-intensity peaks was simulated by the random removal of Gaussian peaks from the synthetic data with probability proportional to the square of the inverse peak height. The total heights of the peaks removed was provided as a parameter for instance generation. This noise model intends to legitimately reflect not only the true underlying physics (e.g., overlapping peaks), but also experimental imperfections of the thin film on which the metals/oxides are sputtered during experimentation.

Results. All experiments were conducted on the same machine and using the same CP solver as in Section 3.2. We first used the DTW-Gaussian kernel as a measure of similarity between sampled locations. Figure 4 depicts the resulting similarity matrix for $N = 91$ and 219; the latter is admittedly hard to understand visually because of too fine a granularity. A point (x, y) in this symmetric matrix is depicted as white if x and y are deemed to be similar, and 0 otherwise; e.g., the main diagonal, representing (x, x) similarity, is white. A similarity matrix such as this is generally considered to be good if areas within it have clear rectangular boundaries, thus identifying small groups of points that are similar to each other but different from the rest of the points. Compared to other standard kernels, we found this DTW-Gaussian kernel to perform the best.

Starting with this kernel as the similarity measure, we used k-means clustering to obtain 50 clusters and asked for 4 points closest to the resulting cluster centroids to generate 50 very small sub-problems for the CP model. Note that these 50 sub-problems are not necessarily disjoint. We then solved each sub-problem with a corresponding CP model (without the connectivity constraint, as mentioned earlier), each of which was either easily solved (average 0.4 sec) when feasible or discarded after 30 seconds if no solution was found in that time. Note that we need to solve a sub-problem this way first for $M = 1$ and then for $M = 2$, which takes 60 seconds in the worst case. When solved, each of them identified 1 or 2 basic patterns or phases; recall that the sub-problem data is insufficient to distinguish between 1 and 3 basic patterns. In the final

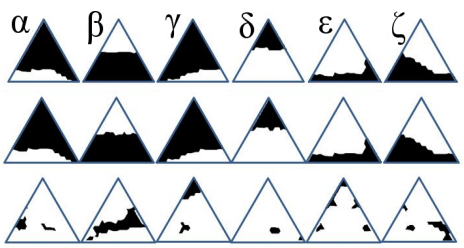

Fig. 5. Results: appearance (white) or not (black) of the 6 phases underlying the Al-Li-Fe system. Top: the true values. Middle: phases found by our hybrid method. Bottom: phases found by the competing NMF approach.

'global' phase, we used these partial solutions to initialize a full CP model of the complete instance as discussed in Section 4.3.

The resulting 6 basic patterns found by the hybrid model are depicted in Figure 5, where the spread of each basis pattern over the composition space appears in white. The top line shows the true answer, which we know from the construction of the instance. The middle row shows the result as produced by our hybrid method. We observe that this solution is extremely close to the true answer in each one of the 6 basic patterns, except for some noise at the boundaries, and it translates into a precision/recall performance across all sampled points, averaged over individual phases of 77.4% / 84.2%.

The bottom row shows the results obtained by the NMF approach recently proposed for this problem. Comparatively, it results in a precision/recall performance of 39.5% / 77.9%. We see that this "solution" is in fact nowhere close to the true answer. Moreover, it violates the hard constraints imposed by physics, such as connectivity (violated for patterns β and ζ) and no more than 3 basis patterns appearing at any location (violated essentially everywhere). This highlights the inability of purely data-driven approaches to effectively deal with hard constraints – a clear strength of CP based approaches.

On the instance with fewer locations (91), we also obtained similar results (and faster) but we omit them here due to lack of space.

6 Conclusion

We explored the use of CP techniques to solve a challenging and interesting problem studied for the most part by researchers in data-driven sub-fields of computer science, or by application domain experts such as physicists in the case of our motivating application — a deeper understanding and discovery of new materials. Our CP model captures the details of the Pattern Decomposition With Scaling problem much better than, say, a matrix factorization or clustering approach, but at the high expense of poor scaling. We therefore introduce a hybrid model that avoids the pitfalls of CP and ML individually, and results in meaningful solutions respecting hard constraints while preserving scalability.

Acknowledgments

Supported by NSF (Expeditions in Computing award for Computational Sustainability, grant 0832782; NSF IIS award 0514429) and IISI, Cornell Univ. (AFOSR grant FA9550-04-1-0151). LeBras was partially funded by a NSERC fellowship. Gregoire and van Dover acknowledge support from the Energy Materials Center at Cornell (USDE award DE-SC0001086). Work done while Gregoire and Sabharwal were at Cornell University.

References

[1] *Powder Diffract. File, JCPDS Internat. Centre Diffract. Data, PA*, 2004.
[2] Z. H. Barber and M. G. Blamire. High throughput thin film materials science. *Mat. Sci. Tech.*, 24(7):757–770, 2008.
[3] G. Barr, W. Dong, and C. J. Gilmore. Polysnap3: a computer program for analysing and visualizing high-throughput data from diffraction and spectroscopic sources. *J. Appl. Cryst.*, 42:965, 2009.
[4] L. A. Baumes, M. Moliner, and A. Corma. Design of a full-profile-matching solution for high-throughput analysis of multiphase samples through powder x-ray diffraction. *Chem. Eur. J.*, 15:4258, 2009.
[5] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, NY, 2006.
[6] T. Damoulas, S. Henry, A. Farnsworth, M. Lanzone, and C. Gomes. Bayesian Classification of Flight Calls with a novel Dynamic Time Warping Kernel. In *ICMLA-2010*, pp. 424–429. IEEE, 2010.
[7] C. Gervet and P. V. Hentenryck. Length-lex ordering for set csps. In *AAAI*, 2006.
[8] C. P. Gomes. Computational Sustainability: Computational methods for a sustainable environment, economy, and society. *The Bridge, NAE*, 39(4), 2009.
[9] J. M. Gregoire, D. Dale, A. Kazimirov, F. J. DiSalvo, and R. B. van Dover. 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.
[10] J. M. Gregoire, M. E. Tague, S. Cahen, S. Khan, H. D. Abruna, F. J. DiSalvo, and R. B. van Dover. Improved fuel cell oxidation catalysis in pt1-xtax. *Chem. Mater.*, 22(3):1080, 2010.
[11] P. Hawkins and P. J. Stuckey. Solving set constraint satisfaction problems using robdds. *Journal of Artificial Intelligence Research*, 24:109–156, 2005.
[12] P. V. Hentenryck and L. Michel. The steel mill slab design problem revisited. In *CPAIOR-08*, pp. 377–381, 2008.
[13] J. Holm, K. de Lichtenberg, and M. Thorup. Poly-logarithmic deterministic fully-dynamic algorithms for connectivity, minimum spanning tree, 2-edge, and biconnectivity. In *STOC-98*, pp. 79–89, New York, NY, USA, 1998.
[14] C. J. Long, D. Bunker, V. L. Karen, X. Li, and I. Takeuchi. Rapid identification of structural phases in combinatorial thin-film libraries using x-ray diffraction and non-negative matrix factorization. *Rev. Sci. Instruments*, 80(103902), 2009.
[15] C. J. Long, J. Hattrick-Simpers, M. Murakami, R. C. Srivastava, I. Takeuchi, V. L. Karen, and X. Li. Rapid structural mapping of ternary metallic alloy systems using the combinatorial approach and cluster analysis. *Rev. Sci. Inst.*, 78, 2007.
[16] R. A. Potyrailo and W. F. Maier, editors. *Combinatorial and High-Throughput Discovery and Optimization of Catalysts and Materials*. CRC Press, 2007.
[17] P. Prosser and C. Unsworth. A connectivity constraint using bridges. In *ECAI-06*, pp. 707–708, The Netherlands, 2006.
[18] H. Sakoe and S. Chiba. Dynamic programming algorithm optimization for spoken word recognition. *Readings in speech recognition*, page 159, 1990.
[19] L. J. Stockmeyer. The set basis problem is np-complete. Technical Report Report No. RC-5431, IBM Watson Research Center, East Lansing, Michigan, 1975.