University of Maryland, College Park

AMSC 663/664

Pattern Decomposition and Basis Phase Recognition of Inorganic Materials

Author:
Graham Antoszewski
ganto@math.umd.edu

Supervisor:
Dr. Hector Corrada-Bravo
Center for Bioinformatics and Computational Biology
University of Maryland, Department of Computer Science
hcorrada@umiacs.umd.edu

May 18, 2017

Abstract

Phase pattern decomposition of inorganic materials' crystalline structure is extremely important for the unearthing of new properties such as superconductivity. Previously, this process had meticulously been done by hand, so computer algorithms have been developed to try and uncover these phases. They, however, have yet to combine efficiency and accuracy together. The goal of this project is to do just that by extending the Graph-based Endmember Extraction and Labeling algorithm (GRENDEL). We will implement algorithms to address physical constraints needed to increase the accuracy of our phase composition results.

1 Background Information

Inorganic materials are compounds or mixtures of elements which do not contain any carbon. Of particular interest are combinations of metal alloys called ternary systems. To make these ternary systems, three different metallic compounds are heated up and combined into one, also referred to as *alloying* the compounds together. Because of the heating and cooling process, the crystalline structure of each individual metal has been altered, similar to how an ice cube that is melted and refrozen will not be identical to the initial configuration. This means the phase of the metal has changed, as the phase is defined as a region within a material or compound where the crystal structure and composition is uniform [1]. This means these phases have distinct properties, such as density and index of refraction. Within different areas of the ternary alloy, there can be different phases of each metal as well due to how the atoms restructured and the proportions of each compound at the given point. Each point of this material is made of a different composition of the three input metals, meaning there can be three phases present and at different proportions based on the mixing process of the alloy. An example of a typical thin film sample of a ternary system is seen in Figure 1.

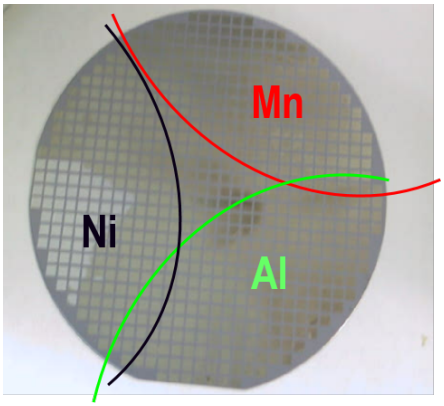

Figure 1: Image of a thin film of an Al-Mn-Ni ternary system, with the regions specified by each color being the predominant areas of each of the composite metals; that is, where each metal was initially introduced into the alloy and then mixed. This highlights how the mixing throughout the material is not uniform, as we want to find all possible combinations of constituent phases [2].

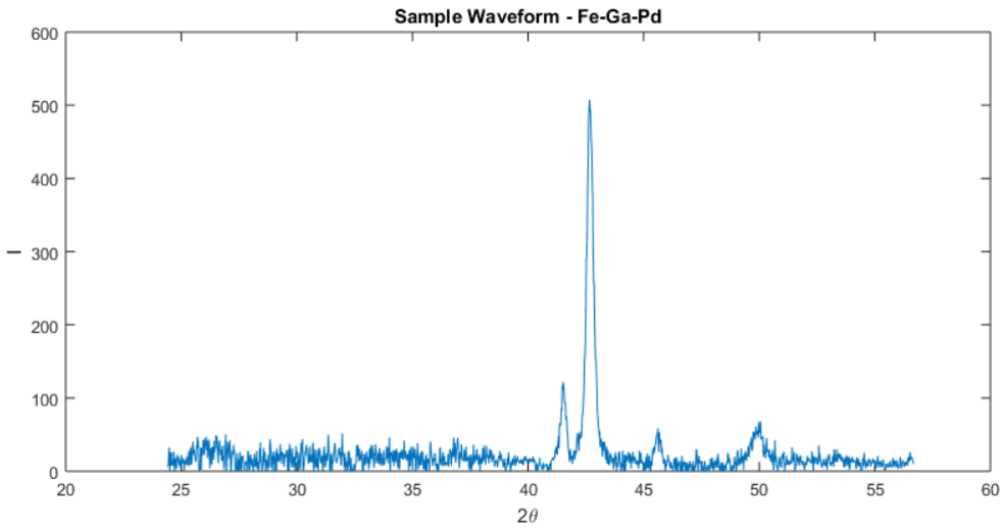

Figure 2: A sample x-ray spectrum from the Fe-Ga-Pd ternary system, with the x-axis being 2θ = the scattering angle observed and the y-axis being the intensity of light detected. Peaks on this plot represent material detection corresponding to given phases of our metals.

A given phase of a metal, as previously stated, has distinct properties, one of these being a unique diffraction pattern. X-ray diffraction is used to probe a given material, sending in beams of electrons and observing the outgoing spectra [1]. X-ray light has a wavelength that is approximately the same as the distance between atoms in a crystal lattice, giving it a better chance to hit the atoms within the structure. The light will hit an electron in the metal, absorb energy, and bounce back at a given angle. Note that this energy exchange only happens at certain incident angles, which is dictated by the Bragg equation,

$$2dsin\theta = n\lambda, \tag{1}$$

where d is the distance between atoms in the lattice, θ is the incoming scattering angle, λ is the wavelength of the x-ray, and n is an integer. The absorbed energy is seen as diffraction peaks at the given angles which satisfy equation (1). Note that since λ is fixed, the only variable which determined the angle θ is the lattice spacing d.

But for an unknown phase, we do not know the distance d. Thus, both the source of x-ray light and the detector rotate in order to record data over all possible scattering angles $2\theta \in [0°, 90°]$ (2θ is defined as the angle between the detector and the incident beam rather than the plane of the material, and will be twice that of θ according to Equation (1)). Figure 2 shows an example of an x-ray spectrum for a single sampled point in a material. The given pattern is called a waveform, where detection of certain phases is indicated by the peaks in the diffraction waveform.

The are three aspects of a given diffraction peak. The scattering angle 2θ is the most important one, as it is the primary marker that tells us about the particular phase and the metal associated with it. The true scattering angle of a given basis phase is unique, and thus is the criterion that tells us about the chemical properties which we are interested in uncovering. The height and width of a given peak can tell us information regarding the phase associated with that peak as well, yet also varies slightly based on the intensity of

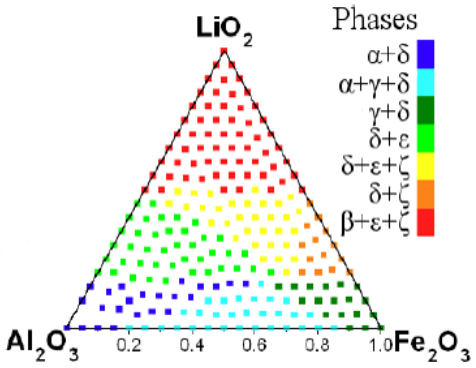

Figure 3: An example of a phase diagram, represented as a simplex. Each vertex corresponds to one of the original compounds in the alloy, colors correspond to similar phase structure between those points, and the Greek symbols in the legend represent the different phases seen in the material [1].

light used in the x-ray diffraction process. One can also notice a shifting of the position of seen peaks over different light intensities, which is a source of error and something that has to be accounted for. Using this data, we can recognize the constituent phases seen at each point in the material along with their respective proportions, and a phase diagram can be made like the one seen in Figure 3 [1]. The alloyed material we wish to sample is usually on a circular thin film, yet we transform the data taken from this shape into a simplex, where each vertex corresponds to the locations of the three initial compounds at the start of the mixing process. Each dot or marker on the simplex corresponds to a probed sample point. Different colors represent areas/clusters within the material where similar phase structure is seen, indicating that these regions will have similar intrinsic chemical properties.

2 Project Objective

Previously, these phase diagrams were done by hand, eyeing the proportions of the constituent phase composition. This process took so long that a library of materials, called the Inorganic Crystal Structure Database, has already been created which have yet to be analyzed. Thus, the White House Materials Genome Initiative was started in order to encourage development of an algorithm to take in this structure and composition data as an input and produce the desired phase diagrams and phase composition data as output. More information is available at *https://www.whitehouse.gov/mgi*). This algorithm must accurately identify the individual basis phases as well as regions or clusters of similar phase composition while also obeying the laws of physics. Furthermore, it must do all of this in an efficient manner so more materials can be evaluated [3].

Current attempts at algorithm development focus on pattern decomposition. Given a set of diffraction patterns at N points of a given system, it is assumed these can be described as a combination of D basis patterns. We seek to resolve these basis patterns, which in this case are the D constituent phases that contribute to the diffraction patterns seen in the material. In other words, if we think of the entire material's diffraction spectrum as a vector

space, we wish to find the basis vectors of the space. In the traditional method of pattern decomposition, there are two main steps, the first being spectral clustering. Here a similarity matrix is constructed to group points in the material with analogous diffraction patterns, with each group of points being called a *cluster*. This splits up our entire dataset into smaller subproblems, allowing the algorithm to run more efficiently. Second, nonnegative matrix factorization is used to identify the constituent phases and their proportions within each cluster [1]. These steps will be explained in detail in Section 3.

One example of such an algorithm is Graph-based Endmember Extraction and Labeling (GRENDEL). *Endmember* is another word for the basis constituent phase which makes up the diffraction pattern of a given point or region within the material, so both terms can be used interchangeably. This method seeks to minimize an objective function during the pattern decomposition process, which looks at how well our estimated phase proportions match up with the raw diffraction patterns both within the clusters and over the entire material. GRENDEL runs very fast, with computation times under a minute, but fails to properly take into account physical constraints which leads to inaccuracy [3]. Another attempt at an algorithm, Alternating Mixed Integer Quadratic Optimization (AMIQO), which uses a combination of several mixed integer quadratic problems to minimize an error function, such as the least squares error between the original structure data and the hypothesized phase structure. Yet this method uses prior knowledge to add in physical constraints. It recognizes certain pairs of points and phases that Must-Link and Cannot-Link together, which leads to extremely accurate results [4] (Section 4.2 will explain in detail a similar Cannot-Link procedure). AMIQO runs on the order of days, however, making it too slow for an ideal method. The latest attempt of a pattern decomposition algorithm is AgileFD, which relies on convolutive nonnegative matrix factorization, physical constraints, and lightweight update rules of the basis phases derived from the Kullback-Leibler divergence loss function to obtain accurate estimates of the basis phases in an efficient manner [5],[6]. Yet AgileFD omits the clustering step of isolating regions of similar phase structure, so we wish to combine the accuracy of AgileFD with the speed and clustering of GRENDEL.

In summary, current approaches at an algorithm are missing at least one key goal of the White House Materials Genome Initiative. Our project objective is to address these issues in one algorithm to combine speed and accuracy. To do so, will be working to extend the GRENDEL algorithm. GRENDEL begins with a spectral clustering step in order to create initial cluster assignments for all of our sample points within our given material dataset. Then, an iterative two-step process of nonnegative matrix factorization and the Graph Cut package is run to find a local minimum of the objective function while simultaneously updating cluster assignments for the entire material [7],[9],[8],[10]. Once convergence is attained, GRENDEL will output cluster assignments for each of the sample points within the material as well as a set of constituent basis phases/endmembers for each cluster [3]. From this output, phase diagrams outlining regions of similar phase structure and constituent phase compositions of each cluster can be generated, with an example of a desirable phase diagram seen in Figure 3.

One of the new components which we will add to GRENDEL is a connectivity constraint algorithm called Cannot Link, which is designed to generate our own novel constraint to mirror those used by AMIQO [4]. Cannot Link seeks to induce a clustering result which will abide by the law of physics, particularly the connectivity of clustering regions within the

material. A more detailed explanation will be given in Section 4.2. The second algorithm we wish to extend GRENDEL with is a version of nonnegative matrix factorization, called ShiftNMF, which will take into account peak-shifting, a result of the alloying process used to produce our ternary material [11]. This physical effect of material creation affects the scattering angle in which our x-ray diffraction detector detects peaks. This leads to error in our basis phase recognition via regular nonnegative matrix factorization. As stated previously, the scattering angles of peaks is the primary feature of a basis phase we wish to unearth for our application in materials science, so addressing peak-shifting in our algorithm is of paramount importance. Further explanation of ShiftNMF is given in Section 4.3.

3 Algorithm - GRENDEL

Figure 4 is a flow chart of the current implementation of GRENDEL. As input, both structure and composition data from the Inorganic Crystal Structure Database and other material libraries can be utilized. If X is the input diffraction waveforms (labeled in Figure 4 as "structure" data) for the whole material of N sample points, GRENDEL looks at each individual sample diffraction waveform X_i. X_i is a vector with dimensions $1 \times M$, where M is the dimension of the waveform. Typically, M is either the number of scattering angles observed or the number of grid points in the Q-spacing. Q-spacing is analogous to scattering angles, yet can have a wider range of values than $[0, 90]$. Certain x-ray diffraction machines record data with scattering angles, while others utilize Q-spacing. For uniformity, all of the results described in this paper will have dimension M to refer to the unit-less general length of the waveform our of initial data. Using this method, M can range between sample materials, although typical values range from 500 to 2000.

A given element $X_{i,j}$ is itself a scattering intensity value seen at the given j^{th} scattering angle by the detector. To graphically see where each of the N sample points are in terms of our phase diagram, each marker seen on the simplex of Figure 3 is a sample point. Composition data C of the material is used in Section 3.2 to place a given sample point i in the correct position on the simplex. The cluster assignment of point i is seen as the particular color of its corresponding marker.

3.1 Spectral Clustering

Spectral clustering seeks to separate the material data into regions or clusters of similar structure, thus allowing the proceeding steps to be run on smaller subsets to speed up computation. These clusters will also be areas of analogous chemical properties. First, a similarity measure is used to compare how close the diffraction patterns are between two given sample points [12]. This metric is the cosine distance between two sample point waveform vectors X_i and X_j, given by

$$\delta_{\cos}(X_i, X_j) = 1 - \cos(X_i, X_j), \tag{2}$$

where $\cos(X_i, X_j)$ is the cosine similarity between the two vectors, defined by

$$\cos(X_i, X_j) = \frac{X_i \cdot X_j}{||X_i||\ ||X_j||}. \tag{3}$$

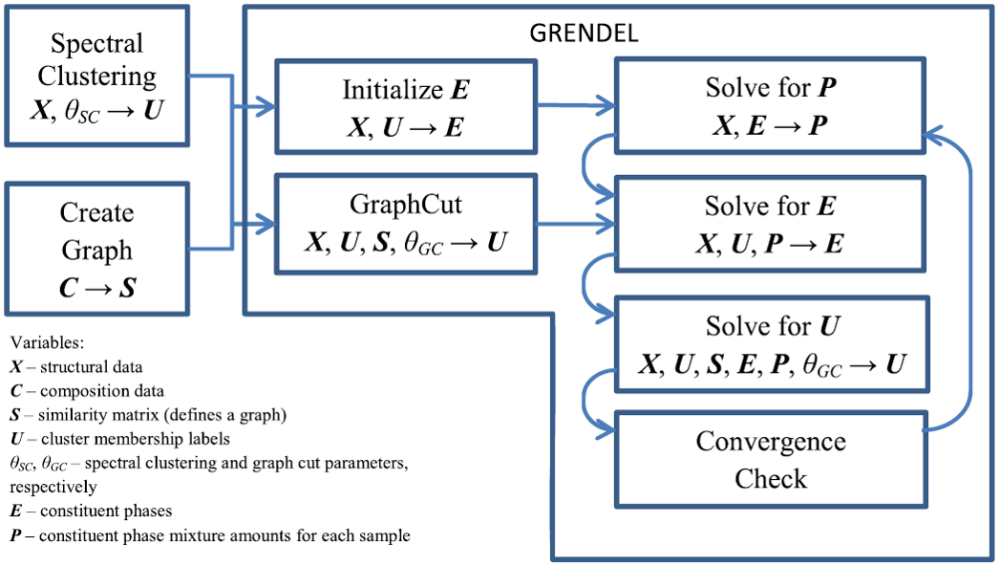

Figure 4: A flow chart of the GRENDEL algorithm. Initial structure X and composition C data is input, and spectral clustering is done to find areas of similar diffraction spectra. Then, an iterative process of the Graph Cut algorithm and NMF (seen here in the last column of the flow chart) is implemented in order minimize the objective function which resolves cluster boundaries and the constituent phases within each cluster. Final output plots are the phase diagram and constituent phase plot of the basis phase waveforms [3].

Here, $\cdot$ is the dot product and $||\cdot||$ is the L2 norm. Thus, the cosine distance between our points X_i and X_j will be near zero if the diffraction patterns match well, near 1 if they are orthogonal, and near 2 if they are completely contradictory [3]. Then, a matrix W is created from these cosine distances,

$$W_{ij} = e^{\frac{-\delta_{\cos}(X_i, X_j)}{2\sigma^2}}, \tag{4}$$

referred to as a similarity matrix, where $\theta_{SC} = \sigma$ is the spectral clustering bandwidth parameter specific to the given material being observed. Thus, W is a $N \times N$ matrix.

With this, a diagonal matrix G is created by summing the rows of W. Then, eigenvalue decomposition used on the Graph Laplacian defined by

$$L = G^{-1}W \tag{5}$$

to find the eigenvectors corresponding to the K smallest nontrivial eigenvalues of L. Here, K is an input parameter specifying how many different clusters are expected to be seen. This varies from material to material, and is usually determined by the user in conjunction with analyzing results of previous experiments with the materials. For example, it is advised to select the number of clusters K to be similar to the number of basis phases expected, which is on the order of 5 to 7.

With these K eigenvectors of length N, GRENDEL utilizes the K-means function within MATLAB to identify individual clusters of points with similar structures. The eigenvectors are assigned to columns of a given matrix Z (so each row of Z corresponds to a data point in the material), and clustering is initialized by finding the K cluster means (note these are

vectors of length K as well). The first is chosen at random from the rows of Z. The other initial means are then chosen from the remaining rows of Z randomly with a probability weighted proportionally to the cosine distance metric between each row and its most-similar cluster mean already selected. The mean initial composition of the k^{th} cluster is given by $\bar{v}_k$. With these, an initial clustering assignment can be made for all N points by assigning each row of Z, where the i^{th} row of Z is z_i, to the cluster whose mean composition $\bar{v}$ is most similar by means of cosine distance:

$$C(i) = \underset{1 \leq k \leq K}{\arg\min}\, \delta_{\cos}(z_i, \bar{v}_k), \tag{6}$$

where $C(i)$ is the current clustering assignment of the i^{th} data point. $C(i)$ ranges from 1 to K.

Then, the cluster means $\bar{v}$ are updated by minimizing the total cosine distance sum between the each z_i in the k^{th} cluster C_k and the desired cluster mean composition v_k,

$$\min_{C, \{\bar{v}_k\}_1^K} \sum_{k=1}^{K} N_k \sum_{C(i)=k} \delta_{\cos}(z_i, \bar{v}_k), \tag{7}$$

where N_k is the number of points assigned to the k^{th} cluster currently. This two-step process defined by equations (6) and (7) is repeated until the cluster assignments no longer change. The final output of this spectral clustering step is the $K \times N$ cluster membership matrix U, where $U_{k,i} = 1$ if the i^{th} data point belongs to the k^{th} cluster [12]. Furthermore, GRENDEL finds the mean spectral composition $\bar{X}_k$ of each cluster by averaging the spectral data of all points within the k^{th} cluster.

3.2 Creating the Simplex

The "Create Graph" portion of the flow chart in Figure 4 refers to a transformation of the input spatial composition data C for each data point into respective coordinates on the simplex. The simplex is created using the Delaunay tessellation function in MATLAB yet only including edge connections to nearest neighbors of each point. This transforms the circular thin film shape of our structure data into a simplex via triangulation. The vertices of the simplex correspond to the three initial metal compounds used to make the ternary system, meaning points closer to these vertices implies the primary component in the mixture at this point will be this particular compound [3].

3.3 PCOMMEND - Lightweight Nonnegative Matrix Factorization

The main portion of GRENDEL is minimizing the objective function, defined as follows:

$$J(X, E, P, U) = \sum_{k=1}^{K} \Big(\sum_{i=1}^{N} u_{ki} (X_i - p_{ki} E_k)^T (X_i - p_{ki} E_k) + \alpha \sum_{h=1}^{D-1} \sum_{l=h+1}^{D} (e_{kh} - e_{kl})^T (e_{kh} - e_{kl}) \Big) \tag{8}$$

Here, X_i is the $1 \times M$ diffraction spectrum for the i^{th} sample point in the material, K is the number of estimated clusters, N is the number of sample points, and u_{ki} is an element of the binary cluster membership matrix U that is 1 if the i^{th} point belongs to cluster number k and 0 otherwise. In addition, D is the number of endmembers (another word for constituent basis phases) in a given cluster, and E_k is a $D \times M$ matrix where the rows are the individual basis phase waveforms that make up the set of basis phases of the i^{th} cluster. That is, the h^{th} column of E_k, symbolized as e_{kh}, is the diffraction spectra of the h^{th} phase of the k^{th} cluster. Moreover, p_{ki} is a $1 \times D$ vector of the proportion values for each basis phase used for the i^{th} sample point. Thus, p_{ki} is a row vector of proportion weights for the basis phases of the k^{th} cluster of the i^{th} sample point. The parameter α is set to 0.0001 to balance the importance of each of the summations [7].

The key assumption is that the input diffraction data at each sample point, X_i, can be approximated as a linear combination of the basis phases. Thus, $\{p_{ki}\}$ is the set of proportion weights applied to the basis phases E_k to make up this combination

$$X_i \approx p_{ki}\ E_k. \tag{9}$$

The first summation term in (8) corresponds to a least-squares residual between our input diffraction patterns X_i and the desired linear combination, $p_{ki}E_k$, while the second summation can be thought of as a volume constraint on the basis waveform vectors themselves. As stated previously, the position/scattering angle of the peaks within the waveform are determined by the lattice spacing term d in the Bragg Equation (1). Since d is a fixed value at a given point in the material, then logically the position of the peaks for each basis phase observed at this point should be approximately equal. If this is not the case, the second summation will be large, implying error in the estimate of the basis phases.

Note that the objective function requires matrices X, U, P, and E. We already have X and an initial guess at U from spectral clustering. We create an initial guess for P by setting all proportions equal to $1/D$, and an initial E is obtained using the *nnmf* function of MATLAB, which outputs a guess at the basis phases themselves for each cluster by seeking to minimize the first summation in Equation (8). Note that since proportions and basis phases are necessary for each cluster, the matrix P is $K \times N \times D$ and E is $K \times D \times M$.

To save computation time, GRENDEL applies lightweight update rules for the matrices E and P using the Piece-wise Convex Multiple Model Endmember Detection algorithm [7]. The *PCOMMEND* method is utilized to quickly find the local minimum of the objective function. First, the equation $\partial J/\partial E_k = 0$ is solved to update our guess for the basis phase matrix of the k^{th} cluster, yielding the equation

$$E_k = \Big(\sum_{i=1}^{N} u_{ki} p_{ki}^T p_{ki} + 2\alpha(DI_{D\times D} - 1_{D\times D}) \Big)^{-1} \Big(\sum_{j=1}^{N} u_{kj} p_{kj}^T X_j \Big), \tag{10}$$

where I and 1 are the $D \times D$ identity and ones matrices, respectively. We assume endmembers/basis phases must be positive in order to resemble a physically-accurate diffraction pattern, so if an element of E_i is negative, that value is set to zero and the matrix is recomputed via Equation (10).

Second, Equation (8) is minimized with respect to p_{ki}. Our proportions of basis phases in cluster i must sum to 1 for each sample point X_i in the cluster in order to be physically

realistic as well, $\sum_{h=1}^{D} p_{kih} = 1$. To ensure this, a Lagrange multiplier λ_k is used. Proportions must be nonnegative as well, so our update of p_{ij} becomes

$$p_{ki} = \max\left(\{(E_k E_k^T)^{-1}(E_k X_i^T - \lambda_k 1_{D\times 1})\}^T, 0\right) \tag{11}$$

with

$$\lambda_k = \frac{1_{1\times D}(E_k E_k^T)^{-1} E_k X_i^T - 1}{1_{1\times D}(E_k E_k^T)^{-1} 1_{D\times 1}} \tag{12}$$

If a particular proportion value is chosen to be 0 because the first term in Equation (11) is negative, then the other proportions for the i^{th} sample point must be normalized in order to have them sum to one.

These two updates are repeated for all K clusters and over multiple iterations along with Graph Cut, to be explained in Section 3.4, to try and locally minimize our objective function (8) [7]. By finding a local minimum, it is understood that the steepest descent-like nature of our updates mean that GRENDEL cannot guarantee convergence to the absolute minimum of the objective function, only that the objective function is minimized within a certain neighborhood of potential solutions for E and P. Depending on different initial seed guesses at U, P, and E, our PCOMMEND update procedures may converge to different final results, although the previous authors of GRENDEL only used the initialization procedure described above [3].

3.4 Graph Cut Algorithm

Now that GRENDEL updated its basis phases E and proportions P, Graph Cut is used to compute the update of these cluster membership matrix U each iteration of GRENDEL [3],[9],[8],[10]. The PCOMMEND updates described above also have an update rule for U along with E and P; however, Graph Cut is chosen to make our cluster membership guess U more accurate. We use a specific MATLAB wrapper available online at

http://www.wisdom.weizmann.ac.il/∼bagon/matlab.html.

The update of U is done by minimizing a cost function, V. The general cost V of the cluster labeling of all input spectral data $X_i, i \in [1, N]$, is described as

$$V = \lambda_d \sum_{i}^{N} V^i(L_i) + \lambda_s \sum_{i=1}^{N} \sum_{j=neighbor}^{all\ neighbors} V^{i,j}(L_i, L_j), \tag{13}$$

where $L_i = k$ is the cluster index of point i, corresponding to $U_{ki} = 1$ if point i is in the k cluster. $V^i(L_i)$ is the data cost for a point i, or the cost to assign a cluster label L_i to i, and $V^{i,j}(L_i, L_j)$ is the smoothness cost, or the cost to assign the labels L_i and L_j to the neighboring points i and j. Note that the values of L range from 1 to K, corresponding to the K clusters. Referring to Figure 4, the Graph Cut parameters $theta_{GC}$ are the scalars λ_d and λ_s, the data cost and smoothness cost weights. These are parameters chosen to balance the smoothness cost, which emphasizes connectivity of clusters so they are all closed regions, and data cost, which emphasizes the similarity of points within a given cluster.

The data cost in Equation (13) is given by

$$V^j(L_i = k) = \frac{3}{4}\delta_{\cos}(X_i, \bar{X}_k) + \frac{1}{4}\frac{||X_i - p_{ki}E_k||_{L_2}}{\sum_k ||X_i - p_{ki}E_k||_{L_2}}, \tag{14}$$

where $\delta_{\cos}(X_i, \bar{X}_k)$ is the cosine distance between diffraction peaks of sample point i and the mean spectra of the currently assigned cluster $L_i = k$, $||\cdot||_{L_2}$ is the L_2 norm, and E_k and p_{ki} are defined as in Section 3.3. The first term makes sure that the spectral data (diffraction pattern) of a point X_i matches with the assigned cluster's mean spectra, similar to the spectral clustering step. The second term makes sure that this cluster's basis phase composition correctly represents the sample point's spectral data X_i, similar to the first summation of the objective function in Equation (8).

The smoothness cost $V^{i,j}(L_i, L_j)$ is 0 if points i and j belong to the same cluster and 1 if they do not. As seen in Equation (13), the smoothness cost summation is restricted to only neighboring points i and j rather than summing over all possible pairs of points. This makes sense- for smooth and continuous clusters, it is expected that most of the adjacent data points to sample point i should also be in the same cluster unless it is on a boundary. Adding these two terms together, V is minimized and all sample points are reassigned into the clusters based on this minimized result.

To minimize V, however, Graph Cut utilizes something called the Max Flow Algorithm [9]. This iterates over all K clusters, and looks at all N data points at one time. In one iteration, looking at cluster κ ($\kappa \in \{1, 2, ..., K\}$), then for each point data point X_i Max Flow looks at the cost of assigning this point into cluster κ versus its current cluster assignment. Specifically, it takes the *residual* between these two costs, and uses this to determine if it should switch the current cluster assignment to cluster κ. The residual refers to the difference in the total costs (data cost + smoothness cost) between a given point i being in its current cluster assignment versus being assigned to cluster κ. Thus, if the residual for point i is positive, Max Flow would say that it is more costly to keep the point in its current cluster, and the cluster assignment of point i should be changed to cluster κ. But if the residual is negative, i should be kept in its current cluster.

The novel idea though is to think of the points in cluster κ as belonging to a "source" tree of flow (positive residuals), and the points remaining in their original cluster assignment as belonging to a 'sink' tree (negative residuals). There must be a continuous path from the highest-level parent nodes of the source and sink tree. This idea is illustrated in Figure 5 [9]. The "A" and "P" labels of the points correspond to whether or not a point is an active or passive node in the tree, which is just terminology to say whether or not nodes are on the boundary of their respective trees (or the boundary of the cluster itself, thinking about the ternary diagram like in Figure 1). Note, however, that while this does enforce semi-connectivity of cluster assignments of the entire material, there are certain points (seen as the white points in Figure 5) that may be disconnected in terms of cluster assignment. This warrants more connectivity constraints to ensure the laws of physics are obeyed.

One might notice in column 2 of the flowchart for GRENDEL (Figure 4), that there seems to be a performance of Graph Cut prior to going into this iterative loop. A simpler version Graph Cut is run prior to this, yet without an initial guess of the matrices E and P. For this run of Graph Cut, the data cost matrix is defined to be only the cosine distance

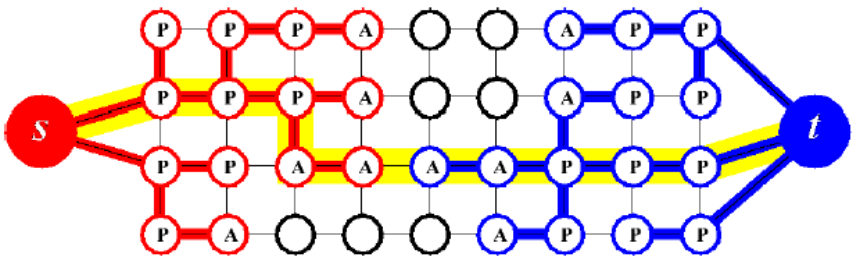

Figure 5: Illustration of the methodology of Graph Cut to update cluster assignments of sample data points. The red points belong to the source tree corresponding to cluster κ, while the blue points belong to the sink tree corresponding to all other cluster assignments. Note the highlighted path of "max flow" between the source and sink parent nodes [9].

metric,

$$V^j(L_j = i)_{\text{simple}} = \delta_{\cos}(X_j, \bar{X}_i), \tag{15}$$

with all other aspects of Graph Cut described above remaining unchanged. This is meant to create a better initial guess at U prior to running the nonnegative matrix factorization portion of GRENDEL.

If, after Graph Cut, there is a cluster where less than 3 data points are assigned membership, these near-empty clusters are eliminated and Graph Cut is re-run. This is meant to overcome potential over-fitting of cluster memberships if our initial guess of the number of clusters, K, is too large.

Graph Cut, along with the nonnegative matrix factorization updates in Section 3.3, are repeated over a certain number of iterations until the convergence criterion of the *condition number* is met. The condition number for iteration *iter* is just the summation of the norms of the difference between the E, P, and U matrices from iteration *iter* and *iter* -1, that is, how much our guesses at these three matrices have changed in one update. If this condition number does not change by a certain threshold (10^{-10}) between iterations, the authors of GRENDEL take this to mean that the algorithm is switching between solutions around a local minimum and the algorithm is stopped. Note, however, that the original GRENDEL algorithm does not guarantee convergence, one of the pitfalls of the algorithm. This is seen in Figure 25.

4 Approach to Extend GRENDEL

4.1 Constraint Already Applied - Gibbs' Phase Rule [1]

One example of a physical constraint is the Gibbs phase rule. Our material is considered to be in equilibrium or steady-state. That is, it is not undergoing any chemical processes such as melting or evaporation, and the chemical composition is stable. At equilibrium, a compound or element must be in a set crystalline structure, corresponding to a set phase. Thus, within our ternary system there can only be three phases seen at a given point due to the three input compounds. Every point assigned to a given cluster k should also be

represented by the same set of endmember basis phases E_k, so this means that at most 3 phases can be seen in a given cluster [1]. As D is defined as the number of basis phases seen in a given cluster, this constraint is written as

$$D \leq 3. \tag{16}$$

Thus, this law of physics is already applied in GRENDEL by us setting $D = 3$ (so the matrix of basis phases E_k, is $3 \times M$, and p_{ki} is a 1×3 vector) during our updates as defined in Section 3.3. No validation is required, although if our sample material only had 2 input metal compounds, $D \leq 2$.

4.2 Connectivity - Cannot Link Algorithm

Due to the continuity of the mixing and alloying process of creating the ternary system of metal compounds, another law of physics that must be upheld is connectivity of the regions within the material with the same basis phases present. In GRENDEL, this means that the clusters themselves within the material must be fully connected, as each cluster should compromise of the same 3 basis phases. A way of visualizing this is through mixing colors while painting. If one mixes red and yellow paint loosely together with a paintbrush, knowing that you started with all of the red paint on the left side of the pallet and yellow on the right, one would expect areas of red, orange, and yellow. But one would not see a two regions of red paint completely isolated from each other without at least connecting path of orange paint (a mixture of red and yellow). Otherwise, it would mean that these red regions somehow split without leaving some sort of trail between them, which is physically impossible. In that same vein, if we mix three metallic compounds, we expect a continuous distribution/path for each of them throughout the entire alloyed material. The basis phases seen at a given point in the material correspond to these input metal compounds (hence Gibbs' phase rule requires $D = 3$), so we expect connected regions of these basis phases as well.

While the utilization of Graph Cut does a decent job of initially enforcing connected cluster regions, discontinuity of certain clusters may occur as we update matrices E and P through nonnegative matrix factorization. To prevent this, we use "prior knowledge" to enforce greater connectivity of clusters. By prior knowledge, we mean that the constraints used are not exactly a scientific or physical law, but the methodology utilized makes our results enforce the laws of physics. We talked about the AMIQO algorithm in Section 2, which applies something called Must-Link and Cannot-Link pairs of data points. Essentially, if the user of the AMIQO algorithm knows prior to analysis that a certain pair of points i and j, having spectra X_i and X_j, are in the same cluster, then they say that these two points in the material must be linked together, regardless of what cluster this pair is assigned into. And if we know that a pair of points are not contained in the same cluster, then this pair cannot be linked in the same cluster [4].

The issue with this method of constraints is that this requires omniscience regarding cluster assignments of certain data points within the material. Therefore, we use our own algorithm, CannotLink, which determines pairs of points that cannot be linked in the same cluster using methodology similar the spectral clustering step described in Section 3.1. Here, we used the cosine distance between two input diffraction waveform vectors, $\delta_{\cos}(X_i, X_j)$, as

a similarity metric. To determine which particular pairs of points that cannot be linked together in the same cluster, we assume that the cosine distance between the two waveforms must be large. We assign the top $\rho\%$ of pairs into a Cannot Link array, given by CL (so CL is a $P \times 2$ matrix, where $P = 0.01\rho \sum_{i=1}^{N-1} i$), and check to make sure no CL pairs are assigned into the same cluster after the Graph Cut portion in GRENDEL. If they are, whichever point in the pair was the latest to switch into the shared cluster is reverted to the cluster assignment of the previous iteration. Cannot Link is only run after each Graph Cut step, as Cannot Link requires the cluster membership matrix U outputted by Graph Cut.

In summary, the Cannot Link algorithm is described below:

```
for i = 1 : size(CL, 1)
    if CL(i, 1) and CL(i, 2) are in the same cluster;
        if point CL(i, 1) changed cluster assignment last
            U(:, CL(i, 1)) = U_old(:, CL(i, 1));
        else
            U(:, CL(i, 2)) = U_old(:, CL(i, 2));
        end
    end
end
```

If both the previous iteration's cluster membership, U_{old}, and the current cluster membership, U, have a CL pair in the same cluster, we eliminate that CL pair from our array. Also, because a given data point i can have multiple Cannot Link pairs, the order in which we loop through the CL pairs can lead to situations where, at the end of the algorithm, certain pairs of points have reverted cluster assignments to again be in the same cluster. This only happens in the first iteration of Graph Cut unless an empty cluster is eliminated at some point in the process. Thus, the algorithm seen above is ran in a while loop. After each run of the Cannot Link algorithm, a check is done to see if any pairs of points in CL are still paired together. If this is the case, Cannot Link is ran until all CL pairs are indeed not in the same clusters.

The parameter ρ should be tested for its optimal value for each given sample material. For our data sets, to be described later, this turned out to be $\rho = 75$. We assumed ρ should be approximately the 1 minus the ratio of the size of the largest cluster to N, the overall number of data points. For our synthetic data set seen in Figure 10, the largest cluster had 61 data points, and with $N = 219$, mean $\rho \approx 1 - 61/219 = .7215$, or 72.15%. Both $\rho = 70\%$ and $\rho = 75\%$ were tested as well as multiple other values to ensure our logic was sound, with $\rho = 75\%$ creating consistently connected results.

4.3 Peak Shifting

The last physical constraint needed to be implemented concerns peak-shifting of the input waveforms in X. It was described in Section 1 how the scattering angle of a given peak an x-ray diffraction pattern is dependent upon the lattice spacing of the material's crystalline structure. It is also assumed that the lattice is flat and uniform, that is, no bumps or

aberrations. But due to the alloying process, the material may not have a perfect lattice structure once it has cooled back down, and thus the lattice spacing may be a little bit off or nonuniform at a given point in the material. This creates a shifting of the diffraction peaks in the waveforms of our sample points, X_i, and this generates error considering the exact location of the peaks of the sample point's waveform is needed to determine its exact basis phases.

An analogy to help understand this can be made with ice cubes. Say you have a perfect ice cube, completely uniform in atomic structure at every point. It is then melted into water, and this water is put into a ice cube tray to be put back in the freezer. Once the ice is solidified again though, the new ice cube may not be exactly the same as the old one - air bubbles may have been trapped in the water during the freezing process, or the tray may not have been completely level when put into the freezer. This would distort the ice cube's atomic structure slightly. A similar idea occurs when alloying the three ternary compounds together. Slight shifts in the atomic lattice structure, while unnoticeable to the naked eye, can be seen through the shifting of peaks in the waveform. This can cause GRENDEL to incorrectly say we have two separate basis phases present in the material, when in reality they are just shifted versions of the same one. This error must be accounted for in order to have accurate clustering diagrams as well as accurate guesses at the basis phase patterns.

To extend GRENDEL to account for peak-shifting, we implement a version of a previously-published algorithm, ShiftNMF [11], which is short for Shifted Nonnegative Matrix Factorization. ShiftNMF takes into account peak-shifting through applying a peak-shifting parameter value to the basis phases corresponding to each individual data point. While both our version and the original authors' ShiftNMF is based upon the same mathematical principles, we seek to make our algorithm more robust and geared to adapt to diffraction patterns of materials. The original ShiftNMF is used for signal processing and includes nuances such as regularization and smoothing, yet does not do well with noisy data. Our version cuts out some extra features of the original ShiftNMF to for use in GRENDEL. Most importantly, our version is set up to allow for constraints to be added in order to account for Gibbs Phase Rule. While a constraint algorithm for Gibbs Phase Rule could not be implemented in the timeline of the semester, it is a subject of future work. For now, we seek to just replicate results in accordance with the original authors.

Compared to the objective function of GRENDEL's nonnegative matrix factorization step, Equation (8), ShiftNMF uses a least-squares objective function

$$J_{LS}(X, E, P, T) = \frac{1}{2}||X - PE||_{L_2}^2 = \frac{1}{2\beta}||X_f - (P \cdot \exp(i\omega T))E_f||_{L_2}^2 \tag{17}$$

Note, Equation (17) does not take into account clustering or the matrix U. ShiftNMF can be run inside a given cluster on a subset of the full initial data X that is assigned to that given cluster, something that will be explored in Section 8.1 . X is still our input diffraction patterns matrix, with dimension $N \times M$ for the N data points and M being the dimension of the waveforms. E is the basis phase matrix, with dimension $D \times M$ for the D basis phases, while P is our proportion matrix, or the weights applied to the phases in E in order to make the linear combination of of basis phases meant to reconstruct our initial data. Thus, P is $N \times D$, with weights for each data point or all basis phases. What is new is T, which is a matrix for a value/magnitude of peak-shifting. It is $N \times D$, to quantify the amount of

peak-shifting for each basis phases for each data point.

The subscript f is used to indicate that the Discrete Fourier Transform of a given matrix has been applied. The peak-shifting values are incorporated in Fourier space using the exponential $\exp(i\omega T)$, with ω a given frequency in Fourier space, which upon applying a Inverse Fourier Transform, equates to a linear shift in the basis phases. The frequencies correspond to the M discrete data points of the waveforms. Both of the representations of our new objective functions, outside and inside Fourier space, are equivalent due to Parseval's identity, which states that the sum of a square of a function, such as least-squares error, is equal to sum the of the square of its Fourier transform, scaled by a parameter β. For our discrete case, $\beta = M$, the length of the waveform.

We seek to minimize the objective function (17) in the same way that regular nonnegative matrix factorization does so. Note all of the following explanations of our update rules of P, E, and T matrices, for brevity and clarity, have been written in vectorized form. Yet in reality, these derivations are done element-wise for each matrix, as this is the only way for the dimensions of X, P, E, and T to match up in Fourier space. Part of the computation rigor of implementing this algorithm is finding clever ways of vectorizing these update rules.

To update the matrix E, our method is as follows:

$$
\begin{aligned}
&P_T = P \cdot \exp\left(i\omega T\right) \\
&\text{grad}_E = \frac{-1}{M} P_T^H (X_f - P_T E_f) \\
&\text{grad}_E^+ = \frac{1}{M} P_T^H P_T E_f \\
&\text{grad}_E^- = \frac{1}{M} P_T X_f \\
&G^+ = \text{ifft}(\text{grad}_E^+), \quad G^- = \text{ifft}(\text{grad}_E^-) \\
&E = E \cdot \left(\frac{G^-}{G^+}\right)^\alpha \\
&\text{If } J_{new} \geq J_{old}, \quad \text{then reduce } \alpha \text{ until } J_{new} < J_{old}
\end{aligned}
$$

To explain, we apply Discrete Fourier Transforms to our X and E matrices using the *fft* function in MATLAB, apply the peak-shifting values T to the matrix P, and set $\partial J / \partial E_f = 0$ in Fourier space to get the gradient. A superscript H denotes the conjugate transpose of the given matrix. We then separate this gradient into the positive and negative term (lines 3 and 4 seen above, respectively) and apply Discrete Inverse Fourier Transforms using the *ifft* function in MATLAB. E is then updated by element-wise multiplication of E with the ratio of the negative to positive parts of the gradient $\partial J / \partial E$ to the power α, which is a convergence parameter. α is tuned to ensure that the objective function is reduced each iteration.

In the same vein, our method to update P is:

$$
\begin{aligned}
&E_{f,T} = E_f \cdot \exp\left(i\omega T\right) \\
&E_T = \text{ifft}(E_{f,T}) \\
&\text{grad}_P = -(X - PE_T)E_T' \\
&\text{grad}_P^- = XE_T' \\
&\text{grad}_P^+ = PE_TE_T' \\
&P = P \circ \left(\frac{\text{grad}_P^-}{\text{grad}_P^+}\right)^\alpha \\
&\text{Guaranteed convergence for } \alpha = 1
\end{aligned}
$$

Here, we apply a Fourier Transform to E and then apply the peak-shifting values. After taking an Inverse Fourier Transform to get back E, only this time with the basis phases shifted, we set $\partial J/\partial P = 0$. To avoid confusion with the matrix T, the transpose of E_T, the basis phases with peak-shifting applied, is denoted by E_T'. As this gradient is taken in real space, this update rule has been shown to be always convergent for $\alpha = 1$ [11].

To update our peak-shifting matrix T, we utilize the Newton-Raphson method. Again, note that the following method is abbreviate in vector form, but in order to take all gradients and Hessians, element-wise derivatives must be taken:

$$
\begin{aligned}
&P_T = P \cdot \exp\left(i\omega T\right) \\
&Q_f = P_T E_f \\
&Y_f = X_f - Q_f \\
&\text{grad}_T = g = \frac{-1}{M}\sum_\omega 2\omega \text{Im}[Q_f Y_f^*] \\
&\text{Hessian}_T = B = \left\{ \begin{array}{ll} \frac{-2}{M}\sum_\omega \omega^2 \text{Re}[Q_f \bar{Q}_f], & \text{for diagonal entries} \\ \frac{-2}{M}\sum_\omega \omega^2 \text{Re}[Q_f(\bar{Q}_f + \bar{Y}_f)], & \text{else} \end{array} \right\} \\
&T = T - \eta B^{-1} g \\
&\text{If } J_{new} \geq J_{old}, \quad \text{then reduce } \eta \text{ until } J_{new} < J_{old}
\end{aligned}
$$

The terms Q_f and Y_f are used to simplify the visualization of the gradient and Hessian, and $\bar{Q}_f$ and $\bar{Y}_f$ denotes the conjugate of the respective matrix. Similar to our update of E, η is a convergence parameter tuned each iteration to ensure the objective function is always reduced.

Due to the complexity of minimizing the objective function (17) when matrix dimensions get large, this iterative method is subject to finding solutions which are local minima. In order to try and combat this, a cross-correlation step is applied every 20 iterations to mix up the peak-shifting values within T. This seeks to update each individual element of the matrix T, as opposed to the vectorized version of the T update above:

Randomly select d$'$ phase, n$'$ data point

Let $X_{n',f} = \text{fft(X) at n}'$

Let $E_{d',f} = \text{fft(E) at d}'$

$$R_{n',f} = X_{n',f} - \sum_{d \neq d'} P_{n',f} E_{d,f} \cdot \exp\left(i\omega T_{n',d}\right)$$

$$C_{n',f} = R^*_{n',f} E_{d',f}$$

$$C'_n = \text{ifft}(C_{n',f})$$

$$t = \arg\max C_{n'}$$

$$T_{n',d'} = t \quad \text{(transformed to fit range of possible peak-shifting values)}$$

In words, the cross-correlation step does a random permutation of all data points and all basis phases, with n' and d' denoting the data point and basis phases indices for the given iteration. We take the Fourier Transform of X and E, and then subtract out all contributions from the other basis phase combinations, with peak-shifting applied (seen as $R_{n',f}$). This is equivalent to the contribution of the d' basis phase for a given diffraction pattern for sample point n'. The cross-correlation between the basis phase d' and the n' diffraction pattern is given by $C_{n',f}$. We take the Inverse Fourier Transform to this to get $C_{n'}$, which is a $1 \times M$ vector. The index of the maximum value of $C_{n'}$ is then taken to be the new peak-shifting value for the d' phase for point n', after transforming this positive index to range from $[-M, M]$.

The updates for P, E, and T matrices are repeated iteratively, with the convergence progress being updated each iteration. The convergence criteria for ShiftNMF is defined as:

$$R^2 = \frac{\text{SST} - \text{SSE}}{\text{SST}}, \tag{18}$$

with SST defined as

$$\text{SST} = ||X||_{L_2}$$

and SSE

$$\text{SSE} = \frac{1}{2M} ||X_f - (P_f \bullet \exp\left(i\omega T\right)) E_f||^2_{L_2}$$

What we denote as R^2 is really the percent of the variance in the initial data explained by our reconstructed solution, PE. In statistics, R^2 is the coefficient of determination, the square of the correlation coefficient between two data sets. Our definition is not the true R^2 in this sense, although we have used this terminology to illustrate the fact that both interpretations are the percent of the variance of an initial dataset explained by reconstructed solution. The only difference is that the coefficient of determination is bounded by $[0, 1]$, whereas our R^2 is not bounded in the negative direction if our reconstruction solution has extreme error. The key, however, is as both versions of R^2 approach a value of 1, it implies convergence to complete agreement between the initial data and our reconstructed solution.

Thus, to summarize, this paper defines R^2 to be a percent of the variance in our initial data X explained by our reconstructed solution of P, E, and T. A value of R^2 being 1

means that our solution is a perfect linear combination of basis phases for our data X. Our iterative process of updating P, E, and T is repeated until a maximum number of iterations is reached, or if the change in the objective function (least squares error) between iterations is under a certain threshold (10^{-8}).

5 Implementation

The overall GRENDEL algorithm and all of the constraint programming is written in MATLAB R2017a. The Graph Cut portion is coded in C++, yet our goal is to not change this function as it has been optimized over years of research [9],[8],[10]. Both the Cannot Link and ShiftNMF algorithm were written in MATLAB R2017a. THe attempts to fit ShiftNMF into GRENDEL, outlined in Sections 8.1 and 9, were written in MATLAB as well. The code is run on a personal ASUS laptop with a 2.4 GHz Intel processor and 8 GB of RAM.

Statistic analysis, including the calculation of p-values discussed in Section 8.1, was done using the Data Analysis package in Excel 2013.

6 Datasets

Four different data sets are used. The first is a synthetic data set given to us by the creators of the GRENDEL algorithm [3],[1]. This diffraction data has been generated for validation testing purposes, as we know the basis phase patterns in E, the proportion of basis phases P, and the cluster membership U for each given data point in the material. For validation of the Cannot Link algorithm, we use the $(Fe\text{-}Al\text{-}Li)O_x$ data set, which is known to have k=7 clusters and 6 basis phases.

The second set of data is taken from the Inorganic Crystal Structure Database (ICSD), a large library of material data. Diffraction spectral data and spatial composition data for the $Fe\text{-}Ga\text{-}Pd$ ternary system from the ICSD courtesy the authors of GRENDEL [3]. This the true phases of this material, however, is not known and thus validation cannot be done on this dataset. An example of the pattern decomposition of the $Fe\text{-}Ga\text{-}Pd$ ternary system from the original GRENDEL algorithm Figure 6. After validation, we wish to test ShiftNMF on this data set, as it is the only diffraction data that has noise, a physically realistic element of diffraction patterns.

The last two data sets are synthetic spectral data used to validate our version ShiftNMF. One of the initial data sets, generated by the authors of the original ShiftNMF algorithm, is in accordance with the validation procedure of the original authors of ShiftNMF [11], while the other data set was generated by us for further validation.

7 Validation Methods

7.1 Cannot Link

Running the original algorithm, prior to applying the connectivity constraint Cannot Link, the results of GRENDEL on the $(Fe\text{-}Al\text{-}Li)O_x$ ternary system is seen in Figures 7 and 8.

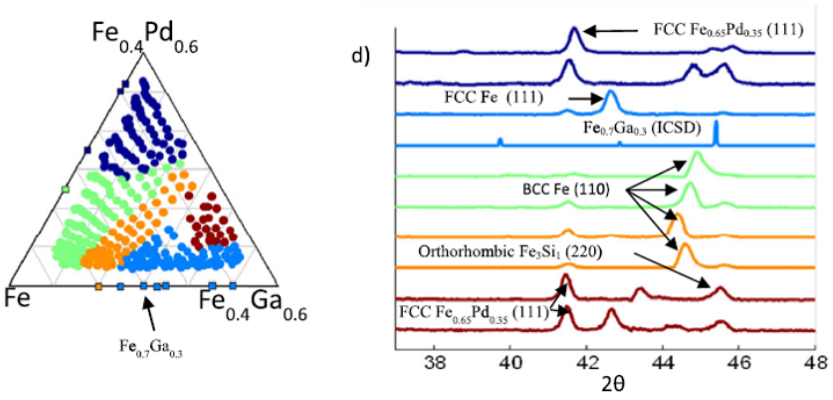

Figure 6: Left: the *Fe-Ga-Pd* system ternary diagram, illustrating the clustering of a previous GRENDEL experiment. Right: the constituent phase plot of the 10 basis phases seen in the material [3]. Since we do not know true values for this data set, this material will be analyzed after validation procedures are completed.

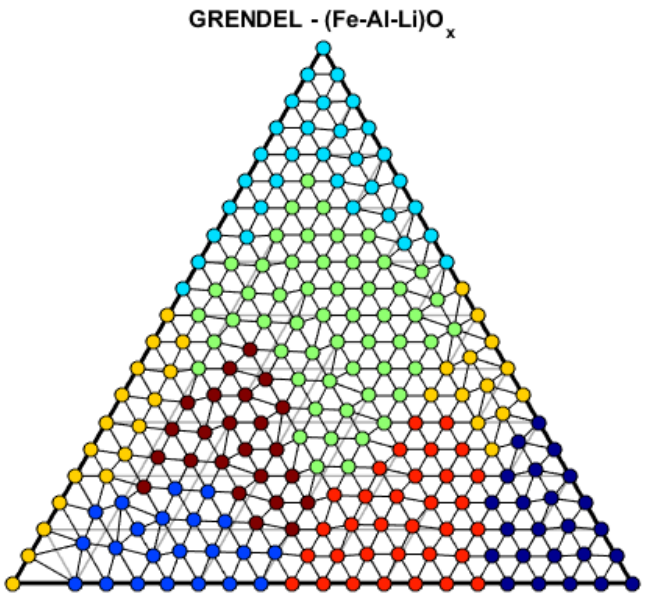

Figure 7: The ternary clustering diagram for the $(Fe\text{-}Al\text{-}Li)O_x$ system for the original GRENDEL algorithm.

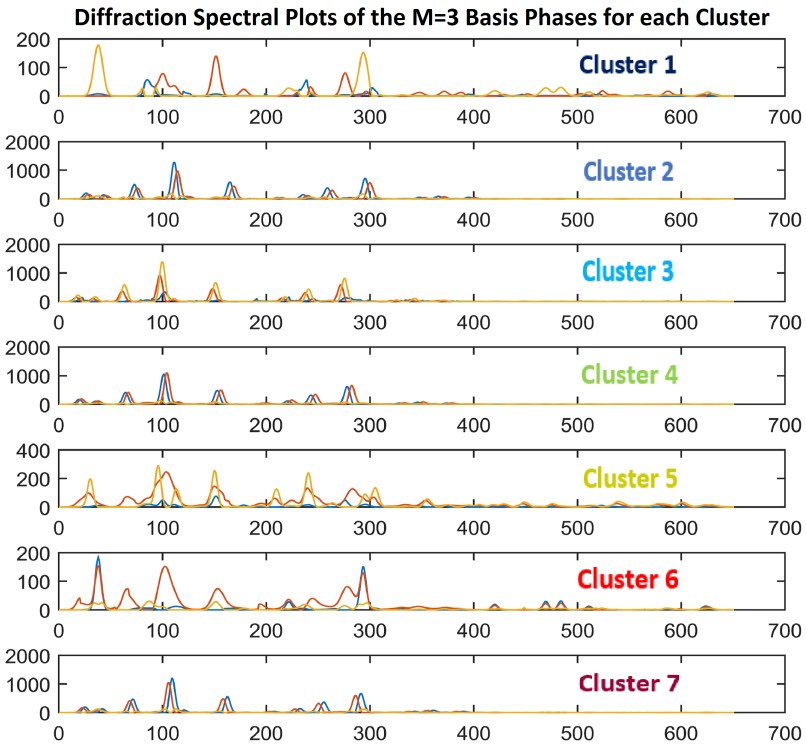

Figure 8: The basis phase waveforms for each of the k=7 clusters. The legend to the right is to associate the given spectral phase plots to their respective cluster color in Figure 7. See that at most 3 waveforms are seen in each cluster, verifying Gibbs Phase Rule has been upheld.

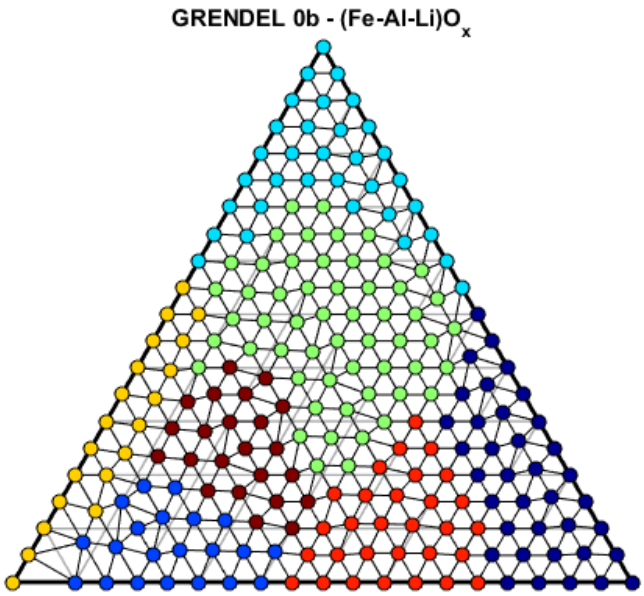

Figure 9: Ternary clustering diagram after adding in Cannot Link algorithm. Note that all k=7 clusters are fully connected now.

Gibb's Phase Rule is seen when looking at Figure 8, the spectral basis phase plots for each of the k=7 clusters. Each cluster is represented by a subplot, and at most three waveforms are seen (indicated by the orange, yellow, and blue waveforms). The plots are labeled accordingly by cluster corresponding to the colored regions seen in Figure 7.

Regarding validation of the Cannot Link, the number of Cannot-Link pairs in the array CL which belonged to the same array after each iteration was documented, as well as the final count of Cannot-Link pairs in the same cluster at the end of the algorithm. Replicated for 50 trials, absolutely zero CL pairs were in the same cluster. This makes sense, as the construction of the Cannot Link algorithm requires this fact in order to advance further. To prove this fact, validation of Cannot Link is built-in as an output of all codes to be delivered in conjunction with this project.

For further visualization of how Cannot Link increases connectivity of clusters, see Figure 9. This is in comparison to the original GRENDEL's clustering, which we see has the disconnected yellow cluster in Figure 7. Over the 50 trials, only 3% to 4% of the CL pairs needed to be eliminated from our array, implying the ρ parameter for the percent of CL pairs is not too large. Note that our clustering does not exactly match up with the true clustering scheme seen in the ternary diagram of Figure 10, yet note this is not possible until peak-shifting has been accounted for.

7.2 ShiftNMF

To validate our version of the ShiftNMF algorithm, we tested it on the synthetic data set used in [11]. This set consisted of $N = 9$ data points, $D = 3$ basis phases, and a waveform

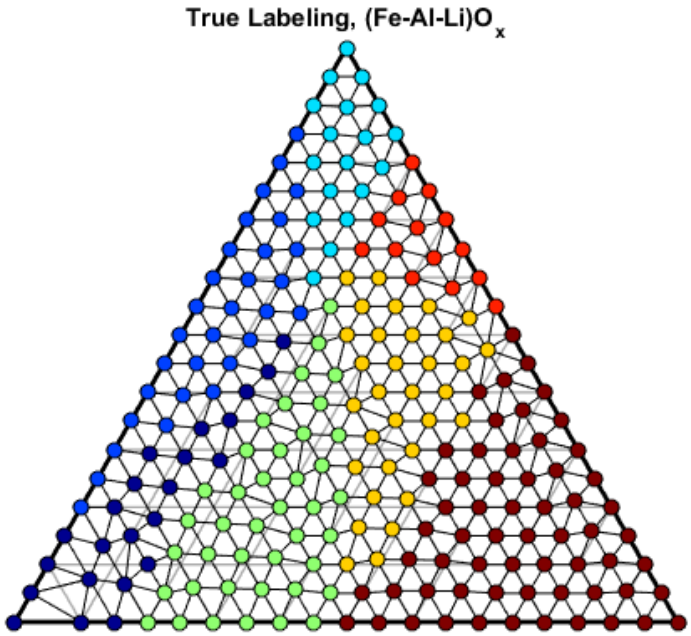

Figure 10: The true ternary clustering diagram for the $(Fe\text{-}Al\text{-}Li)O_x$ synthetic data set.

length $M = 1400$. Again, note that for plotting purposes, M is a dimensionless grid spacing representing either scattering angle or Q-spacing of the diffraction pattern. The intensity values (y-axis) was scaled for comparison purposes. Note in all of the proceeding figures, the key aspect to analyze is the exact position of the basis phases recognized by ShiftNMF in comparison to the true values. Regardless of the height/intensity of the basis phase, the scattering angle/Q-spacing value of the peaks of a given basis phase provide the information about the chemical properties of the material which we desire in application. Figures 11 and 12 show that our reconstructed solution of the P, E, and T matrices match the true values.

To further show our version of ShiftNMF can resolve the true basis phases, proportions, and peak-shifting values, we also tested it on a harder data set, seen in Figure 13. This data set had $N = 12, D = 4, M = 1500$. Again, our reconstructed solution matches with the true values, in particular with the basis phases. The 12^{th} data point is the only source of error in our solution; however this error manifested in the proportions and the peak-shifting values rather than the basis phases. As ShiftNMF is not deterministic due to the random component of our convergence methods, this will not be the case for every run of ShiftNMF. As a part of this project, code will be delivered with the specific seeds of the random number generation included in order to replicate these figures.

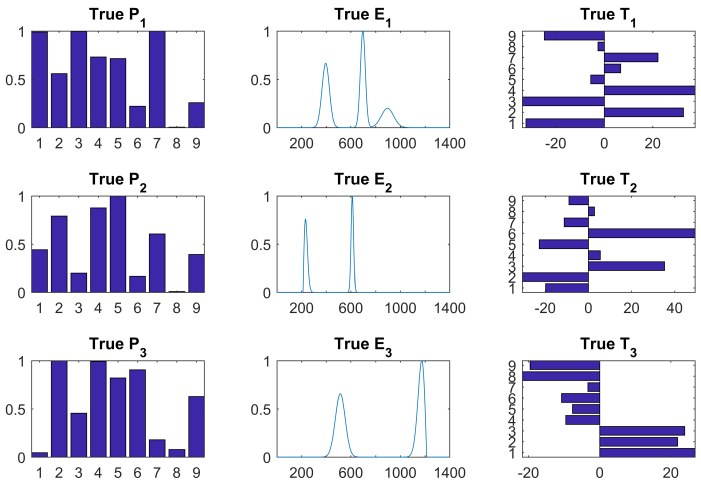

Figure 11: True values of P, E, T for the 3 basis phases.

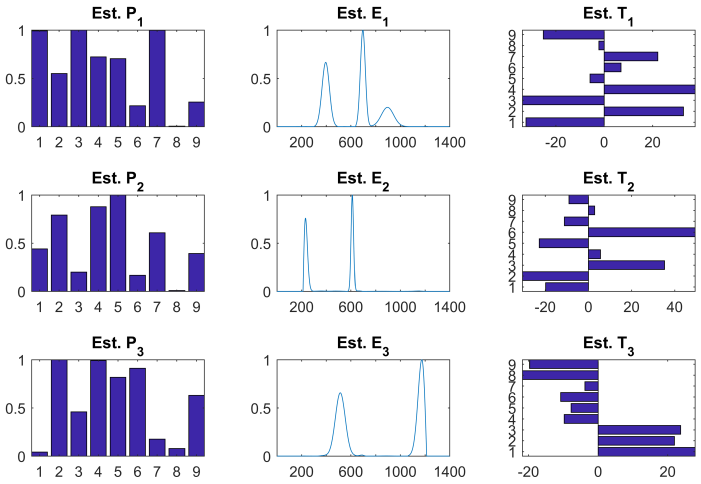

Figure 12: Results P, E, T of ShiftNMF, after 2000 iterations. An R^2 value of 1.0000 was observed.

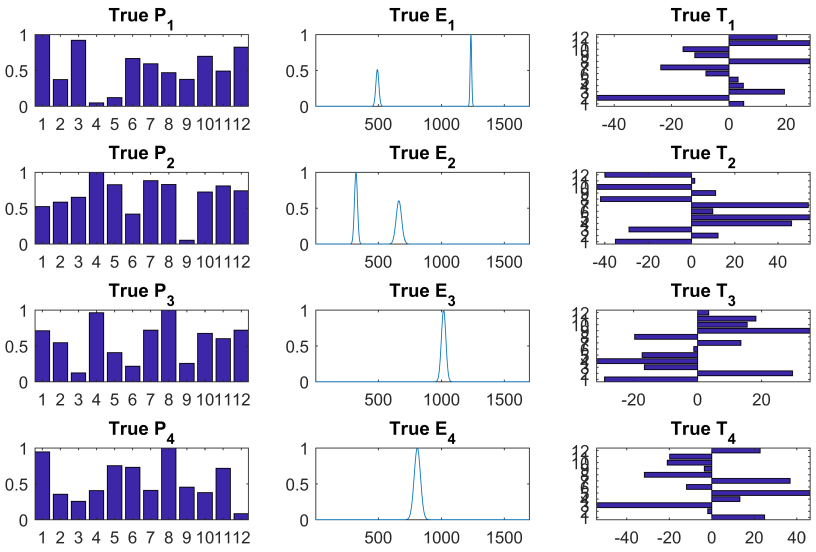

Figure 13: True values of P, E, T for the 4 basis phases.

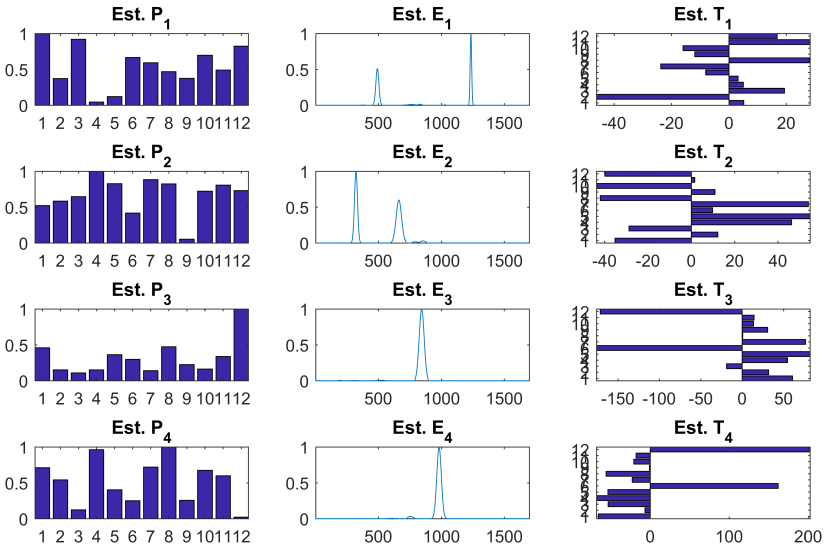

Figure 14: Results P, E, T of ShiftNMF, after 1743 iterations. An R^2 value of 0.9993 was observed.

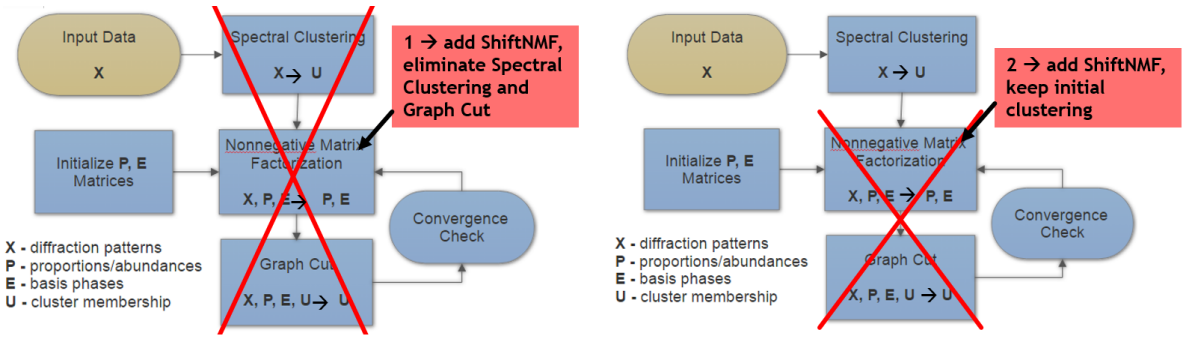

Figure 15: The two options of implementing ShiftNMF within GRENDEL. To the left, applying ShiftNMF without any clustering. To the right, replacing the original nonnegative matrix factorization step with ShiftNMF, yet keeping the initial clustering steps.

8 Testing

8.1 Testing ShiftNMF within GRENDEL

The Cannot Link algorithm has already been implemented in GRENDEL, yet ShiftNMF still has to integrated. There are two options we explored, seen in the flow charts in Figure 15. First, we tested just running ShiftNMF on our synthetic spectral data set $(Fe\text{-}Al\text{-}Li)O_x$ without clustering. Note though, this idea does not adhere to Gibbs Phase Rule, a major law of physics violation.

The second option is to replace the nonnegative matrix factorization step described in Section 3.3 with ShiftNMF and keep just the initial clustering steps, the spectral clustering and the run of the simpler version of Graph Cut. As a reminder, the data cost function of this simpler version of Graph Cut is defined in Equation (15). In the step of the project, we were testing whether or not it is advantageous to cluster before or after running ShiftNMF in the constructs of GRENDEL. So, we decided to cut out all advanced Graph Cut steps, as we had yet to alter the data cost function described in Equation 14 to take into account the peak-shifting values T. This method of using ShiftNMF with clustering adheres to Gibbs Phase Rule, yet the initial clustering steps do not take into account the peak-shifting error.

Both of the strategies, ShiftNMF without clustering and ShiftNMF with clustering, were coded and tested in comparison to GRENDEL. As the ShiftNMF algorithm had to be altered in each testing method, to validate that the two new strategies still performed as they should, the synthetic data set with $N = 9$ points and $D = 3$ phases used to originally test ShiftNMF was again replicated, only with new true values of P and T for each data point [11]. These correspond to Figures 16, 17, 18, and 19.

We looked at specifically how well the variance in the initial data is explained by our reconstructed solution with P, E, and T. As stated in Section 4.3, we call this our R^2 statistic. Specifically, we ran 30 trials of ShiftNMF without clustering, ShiftNMF with clustering, and the original GRENDEL code. This was run for two types of initialization of the matrices P and E - randomized numbers, in accordance to the initial conditions of ShiftNMF, and

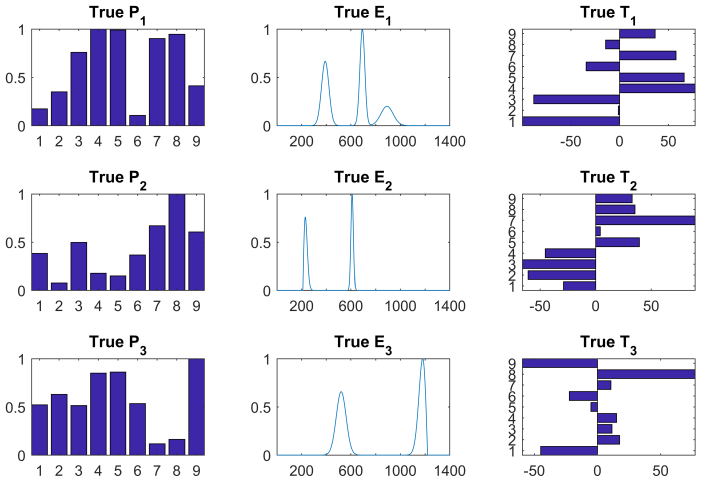

Figure 16: True values of P, E, T for the 3 basis phases used to validate ShiftNMF within Strategy 1 (without clustering).

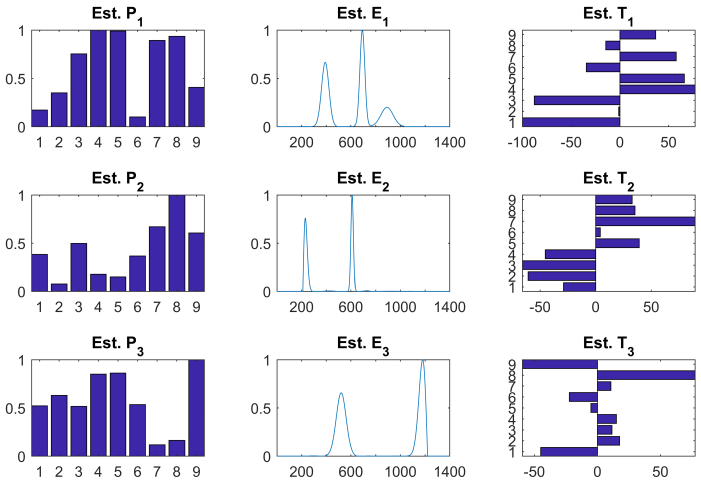

Figure 17: Results P, E, T of Strategy 1, ShiftNMF without clustering, after 2000 iterations. An R^2 value of 1.0000 was observed.

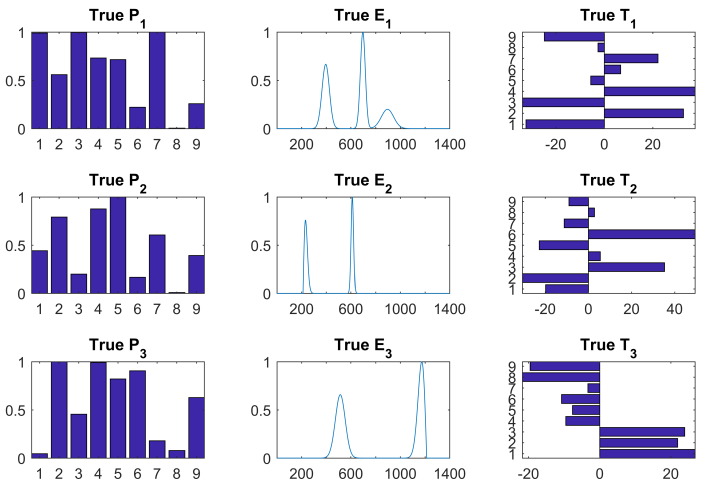

Figure 18: True values of P, E, T for the 3 basis phases used to validate ShiftNMF within Strategy 2 (with clustering).

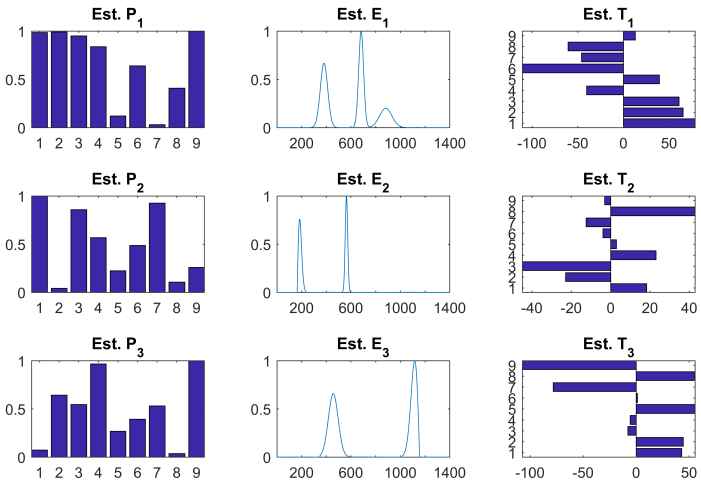

Figure 19: Results P, E, T of Strategy 2, integrating ShiftNMF within the GRENDEL clustering scheme, after 2000 iterations. An R^2 value of 1.0000 was observed.

using the initial conditions of the original GRENDEL algorithm (uniform proportions for P, using the output of *nnmf* MATLAB function).

The results of the testing are summarized below:

Implementation	Initialization Procedure	Mean R^2	t-statistic	p-value
GRENDEL	Randomized	0.9493	N/A	N/A
GRENDEL	nnmf	0.9450	N/A	N/A
ShiftNMF without clustering	Randomized	0.9607	6.308	3.91e-7
ShiftNMF without clustering	nnmf	0.9721	30.385	9.41e-28
ShiftNMF with clustering	Randomized	0.9872	77.236	3.71e-45
ShiftNMF with clustering	nnmf	0.9867	97.088	6.70e-66

We see that the mean values of R^2 are indeed higher for our two strategies compared to the original GRENDEL. Yet this does not tell us as much the t-statistic and p-values that were calculated through 2-sample t-tests with unequal variances. This is an example of hypothesis testing. For this, we create a null hypothesis, in this case, that the final mean R^2 values our ShiftNMF strategies is the same as the mean R^2 of GRENDEL's output. Assuming this null hypothesis to be true. A probability distribution is then created to mirror this null hypothesis. Because the true mean and standard deviation in the results of GRENDEL are not known, we use what is known as a t-distribution. This is similar to a standard normal distribution, only it accounts for our lack of prior knowledge of the mean and standard deviation. Using this probability model, we test our null hypothesis by taking the difference in our mean R^2 values. After scaling this by the standard error, a combination of the standard deviations of each of the R^2 data sets, we are given a *t-statistic* value. This statistic directly corresponds to the probability that this magnitude of difference (or larger differences) in mean R^2 values would be seen in the t-distribution assuming the null hypothesis, which is the *p-value*.

In general, a p-value of 0.05, or a 5% probability that the results seen can be explained by the null hypothesis, is seen as statistically significant. By that, we mean that we can reject the null hypothesis and say that the mean R^2 between GRENDEL and each of our ShiftNMF strategies are not equal. For example, the p-value of 3.91e-7 implies that, assuming a probability distribution that the mean R^2 values are equal, that there is a probability of 0.000000391% that the observed mean R^2 difference in the algorithms are explained by the null hypothesis. Thus, for both initializations and for both strategies, we can reject the null hypothesis, and we can say our implementations of ShiftNMF yield different R^2, specifically that our strategies yield better R^2 convergence results.

This, however, does not take into account certain physical aspects of our solutions in each strategy. To illustrate a key point, see Figure 20. Particularly when implementing ShiftNMF within clustering, the clustering portion of GRENDEL does not take into account peak-shifting prior to running ShiftNMF. This leads to error when calculating the basis phases - there is over-fitting and we see many more unique phases than the desired solution. Only a few of phases overlap between the clusters, which is an unacceptable physical result.

In comparison, when running ShiftNMF outside of clustering, we see the correct number of basis phases. Looking at Figure 21, we have been able to recognize some of the true phases

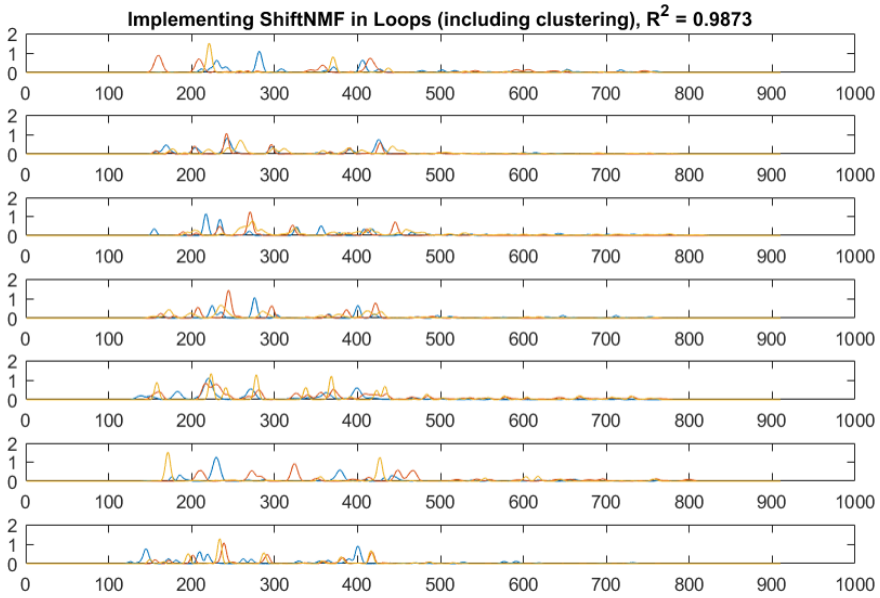

Figure 20: Basis phases of a single run of the strategy of ShiftNMF with clustering. Note that the majority of the 21 phases, spanning 3 phases for each of the 7 clusters, are unique. That is, running ShiftNMF with clustering yields more than the true number of basis phases.

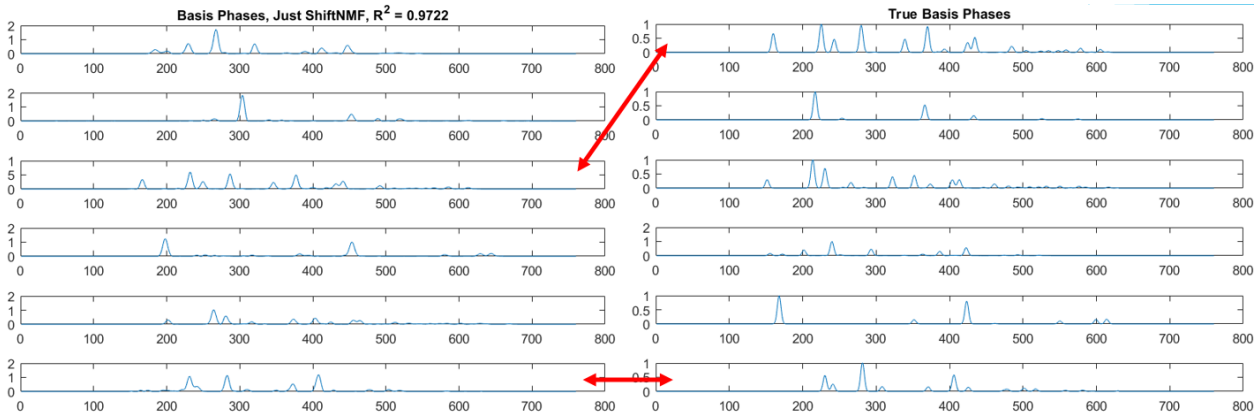

Figure 21: Basis phases of a single run of the strategy of ShiftNMF without clustering (right) compared to the true basis phases of the $(Fe\text{-}Al\text{-}Li)O_x$ synthetic data set. Notice that some agreement is even seen between the experimental basis phases and true phases.

of the $(Fe\text{-}Al\text{-}Li)O_x$ synthetic data set. This leads us to believe that, prior to clustering, that ShiftNMF needs to run in order to take into account peak-shifting prior to any clustering step.

More evidence of a need to implement ShiftNMF prior to clustering is seen when running the two strategies in the real material data from the ICSD, $Fe\text{-}Ga\text{-}Pd$. This is important as this real diffraction data contains noise, something our implementation of ShiftNMF must address in order to be valuable. Random initial conditions were used and ran for 10 trials each. For ShiftNMF without clustering, we set the number of basis phases to be $D = 8$. The number of clusters in GRENDEL and ShiftNMF with clustering were set to be 5. The results are summarized below:

Implementation	Mean R^2	t-statistic	p-value
GRENDEL	0.8871	N/A	N/A
ShiftNMF without clustering	0.9138	23.491	9.45e-11
ShiftNMF with clustering	0.9055	17.450	6.89e-9

We still see statistically-significant p-values for both of our strategies, but now ShiftNMF without clustering performs better in terms of the mean R^2 statistic. This real data set includes more data points and less clustering, meaning less over-fitting when running ShiftNMF within the GRENDEL clustering scheme.

9 New Algorithm - ShiftGRENDEL

With these conclusions in hand, we have developed a new algorithm to try and combine Cannot Link and ShiftNMF into GRENDEL to take into account peak-shifting in both basis phase recognition as well as clustering. We call this ShiftGRENDEL, and a flowchart explaining the methodology is seen in Figure 22.

To summarize, ShiftNMF is run first after initializing the CL array and P, E, and T matrices. The number of basis phases D is also an input, and we output P, E, and T from ShiftNMF. Then, we assign cluster memberships of each of the data points based on the outputted basis phases and proportions for each data point. The output is the cluster membership matrix U as well as the number of unique clusters seen k. Of importance is that we recognize that Gibbs Phase Rule is not enforced in ShiftNMF outside of clustering. This causes most data points X_i to be a linear combination of 4 or 5 phases. This is a focus of future work on this project, to be discussed later. For now, cluster membership is assigned by selecting the 3 basis phases with the highest proportions for a given data point i. Note that within each cluster now, only 3 basis phases are seen, so within each cluster Gibbs Phase Rule is applied similar to the original GRENDEL code.

Then, after cluster membership is assigned, ShiftNMF is now ran within each cluster. This is ran with the entire input data set X in all clusters. Within each cluster, we set the basis phases to be constant, meaning we skip over the E update in ShiftNMF. This updates the proportions P and peak-shifting values T for each of the data points, if we were to assign them into that cluster. The reason for this is to be able to formulate the cost of a given data point i to be assigned into each of the clusters, something necessary to run Graph Cut. The

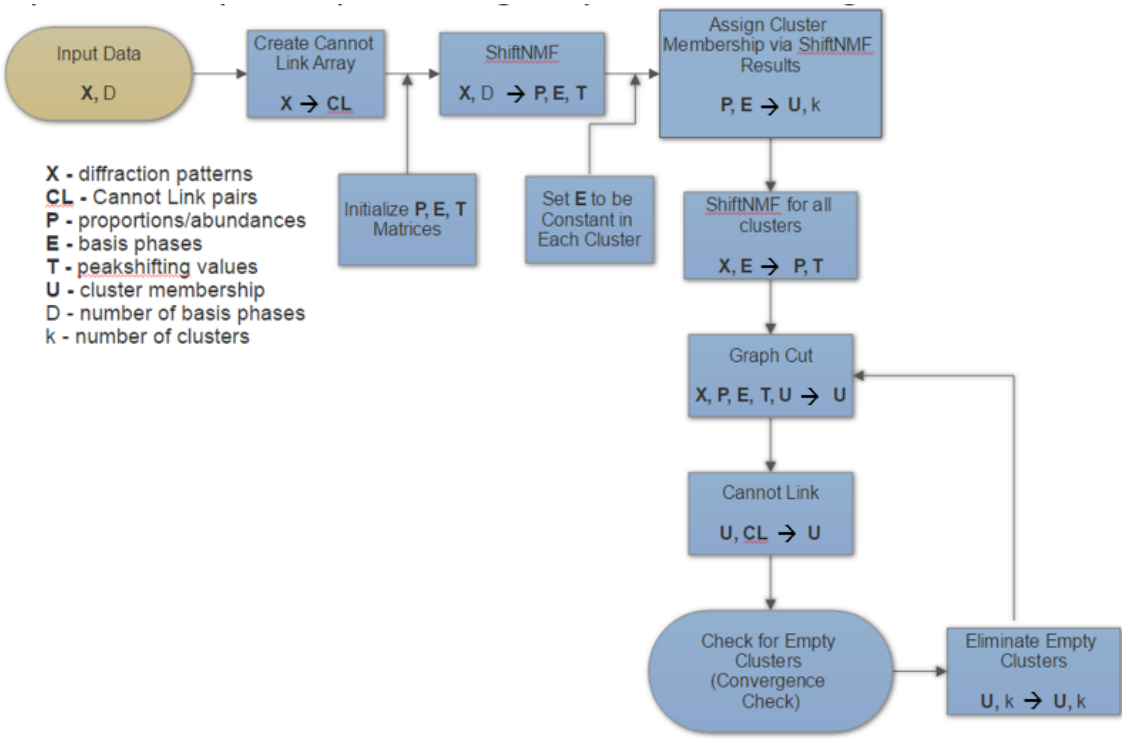

Figure 22: A visual representation of the ShiftGRENDEL algorithm.

data cost function described in Equation (14) is then appended to apply the peak-shifting values T, similar to our ShiftNMF objective function in Equation (17).

After this, Cannot Link is run to update cluster membership. As a convergence check/stopping criterion, we check to see if there are any empty clusters (in the implementation of shift-GRENDEL in my deliverables, this is defined to be clusters with less than 15 data points). Empty clusters are removed if they are present, and Graph Cut and Cannot Link are ran again. Once all clusters are non-empty, ShiftGRENDEL stops.

To validate ShiftNMF within ShiftGRENDEL runs as it should, again the ShiftNMF synthetic data set from [11] was used as input, with Figures 23 and 24. This data set has $D = 3$ basis phases, so by definition Gibbs Phase Rule was upheld with clustering.

To test to see how ShiftGRENDEL performs in terms of convergence, 30 trials were ran on the $(Fe\text{-}Al\text{-}Li)O_x$ data set. The R^2 statistic was recorded after the initial ShiftNMF, prior to clustering. Because of our insufficient method of enforcing Gibbs Phase Rule when clustering after ShiftNMF, misclassification of cluster membership is very high. This leads to inaccurate reconstruction results of P and T within clusters. Once Gibbs Phase Rule is enforced, we will be able to look at statistics such as R^2 after running ShiftNMF in clusters and Graph Cut.

GRENDEL had a mean R^2 value of 0.9484, while ShiftGRENDEL had a mean R^2 of 0.9609. This led to a t-statistic, after doing a 2-sample t-test with unequal variances, of 6.779, corresponding to a p-value of 4.90e-8. Again, we see that we can reject the null hypothesis that the mean R^2 values of GRENDEL and ShiftGRENDEL are assumed to be equal. Thus, we see a better reconstruction of the input data X using the outputted P, E,

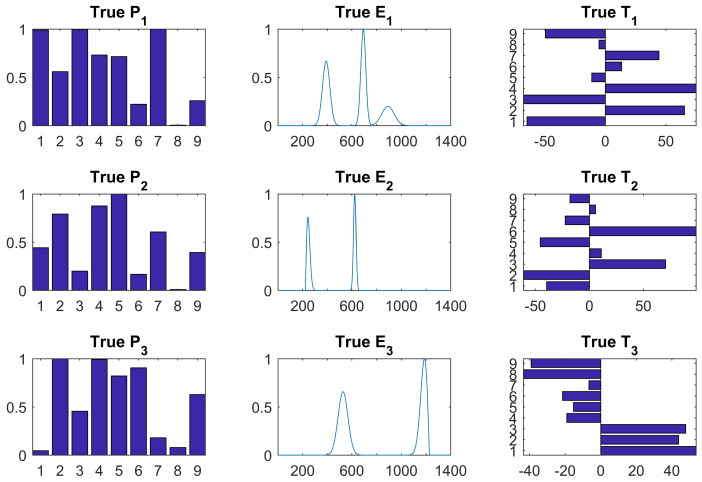

Figure 23: True values of P, E, T for the 3 basis phases used to validate ShiftNMF within ShiftGRENDEL.

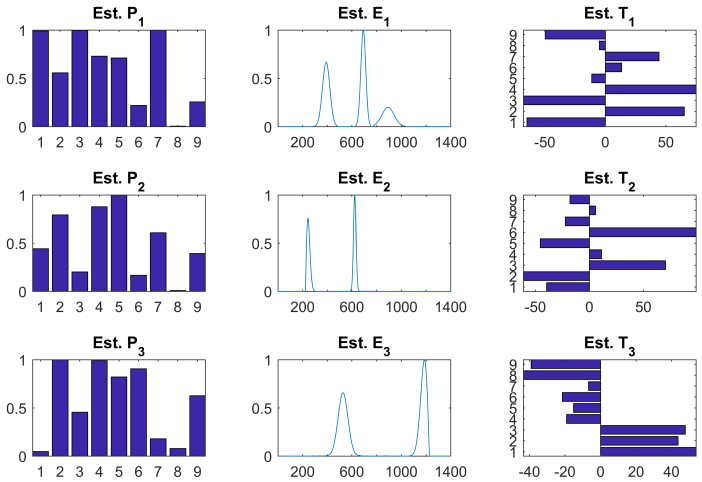

Figure 24: Results P, E, T of the ShiftNMF scheme within ShiftGRENDEL. And R^2 value of 1.000 was observed.

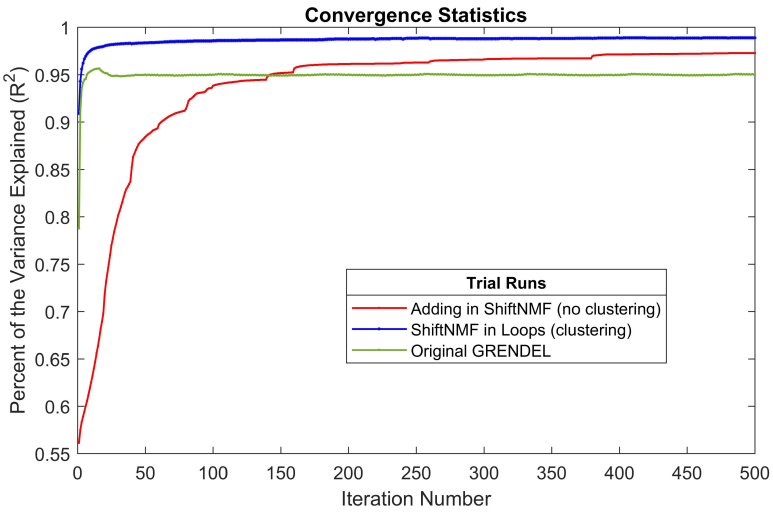

Figure 25: R^2 values at each iteration step for GRENDEL, ShiftNMF without clustering, and ShiftNMF with clustering. Note that convergence to a set R^2 value tends to occur earlier than 500 iterations. This also illustrates a flaw of GRENDEL, that is does not guarantee convergence from iteration to iteration.

and T matrices of ShiftGRENDEL.

While this is nice, it is in no way a satisfactory result yet. Gibbs Phase Rule must be implemented in order to get physically realistic solutions. The next step in this project, to be completed this summer, will be to implement a LASSO-type L1 regularization procedure to restrict the number of basis phases seen at each point within the initial ShiftNMF step, prior to clustering. Also, we have yet to talk about the speed of each of the algorithms. Averaging over 10 trials for 500 iterations, the mean run-time of the original GRENDEL code was 30.1 seconds. Strategy 1 of ShiftNMF without clustering ran in 174.8 seconds, and ShiftNMF within clustering ran in 147.7 seconds. Yet ShiftGRENDEL, due to the repeated running of ShiftNMF, the most computationally-expensive part of the algorithm, ran in 1281.3 seconds, much too high. Future work will also address this. A first idea will be to parallelize the running of ShiftNMF within each cluster, as we can coded this to be able to run independently and simultaneously with affecting results. Second, Figure 25 shows how R^2 converges to a stable value in a seemingly quick fashion. An idea is to see just how many iterations of ShiftNMF is needed in order to achieve acceptable results.

10 Timeline

The project timeline had to be appended several times, with the final revision being early this semester after recognizing the complexity of applying methods to account for peak-shifting. We were able to complete nearly every part of this timeline established:

1. Fully understand GRENDEL, replicate the previous results (*mid/late October*) **completed**

2. Connectivity constraint - Cannot Link
   (a) Write Cannot Link algorithm (*November*) **completed**
   (b) Validate and optimize parameters (*December*) **completed**
3. Peak-shifting - ShiftNMF
   (a) Locate and understand algorithm, ShiftNMF (*January*) **completed**
   (b) Write own version of ShiftNMF (*February*) **completed**
   (c) Validation (*March*) **completed**
4. Implementing Cannot Link and ShiftNMF within GRENDEL
   (a) Test options for ShiftNMF within GRENDEL (*April*) **completed**
   (b) Generate ShiftGRENDEL algorithm (*early May*) **completed**
   (c) Collect final results (*May*) **completed**
   (d) Optimize run-time of ShiftGRENDEL (*May*) **incomplete**

11 Deliverables

I will be delivering packages of each of the algorithms discussed during this project. These will include the Cannot Link algorithm, ShiftNMF, both strategies of implementing ShiftNMF into GRENDEL, and the final ShiftGRENDEL code. Zip files will be created corresponding to packages to validate ShiftNMF outside of GRENDEL on both data sets discussed, as well as the three methods of combining ShiftNMF and GRENDEL set to run the code with the synthetic $(Fe\text{-}Al\text{-}Li)O_x$ data as well as the ICSD $Fe\text{-}Ga\text{-}Pd$ data. A readme.txt file will also be attached that will include directions regarding how to run each code for each data set. All data sets used will be included, as well as an Excel spreadsheet of data taken regarding the testing described in Section 8.1, both included in this report as well as extra statistics taken.

References

[1] Lebras R., Damoulas T., Gregoire J.M., Sabharwal A., Gomes C.P., and van Dover R.B., *Constraint reasoning and kernel clustering for pattern decomposition with scaling*, AAAI **CP'11** (2011), 508–522.

[2] Takeuchi I., *Data Driven Approaches to Combinatorial Materials Science*, Materials Research Society Spring Meeting presentation, University of Maryland, College Park, 2016.

[3] Kusne A.G., Keller D., Anderson A., Zaban A., and Takeuchi I., *High-throughput determination of structural phase diagram and constituent phases using GRENDEL*, Nanotechnology **26** (2015), no. 44, 444002.

[4] Ermon S., LeBras R., Suram S., Gregoire J.M., Gomes C.P., Selman B., and van Dover R.B., *Pattern decomposition with complex combinatorial constraints: application to materials discovery*, AAAI Conference on Artifical Intelligence (1972), available at `http://www.aaai.org/ocs/index.php/AAAI/AAAI15/paper/view/10020`.

[5] Ermon S., LeBras R., Suram S., Gregoire J.M., Gomes C.P., Selman B., and van Dover R.B., *Automated Phase Mapping with AgileFD and its Application to Light Absorber Discovery in the V-Mn-Nb Oxide System*, 2016.

[6] Xue Y., Bai J., LeBras R., Rappazzo B., Bjork J., Longpre L., Suram S., van Dover R.B., Gregoire J.M., and Gomes C.P., *Phase-Mapper: An AI Platform to Accelerate High Throughput Material Discovery*, CoRR **1610** (2016).

[7] Zare A., Gader P., Bchir O., and Frigui H., *Piecewise Convex Multiple-Model Endmember Detection and Spectral Unmixing*, IEEE Transactions on Geoscience and Remote Sensing **51** (2013), no. 5, 2853–2862.

[8] Boykov Y., Veksler O., and Zabih R., *Efficient Approximate Energy Minimization via Graph Cuts*, IEEE Transactions on PAMI **20** (2001), no. 12, 1222–1239.

[9] Boykov Y. and Kolmogorov V., *An Experimental Comparison of Min-Cut/Max-Flow Algorithms for Energy Minimization in Vision*, IEEE Transactions on Geoscience and Remote Sensing **51** (2013), no. 5, 1124–1137.

[10] Kolmogorov V. and Zabih R., *What Energy Functions can be Minimized via Graph Cuts?*, IEEE Transactions on PAMI **26** (2004), no. 2, 147–159.

[11] Morup M., Madsen K.H., and Hansen L.K., *Shifted Non-negative Matrix Factorization*, IEEE International Workshop on Machine Learning for Signal Processing (2007), 139–144.

[12] Hastie T., Tibishirani R., and Friedman J., *The Elements of Statistical Learning - Data Mining, Interference, and Prediction*, 2nd ed., Springer, 2013.