Machine Learning for EDS Data Decomposition

Dhruti Shah (dhruti.shah@epfl.ch)
Kshiteej Sheth (kshiteej.sheth@epfl.ch)
Aditya Vardhan Varre (aditya.varre@epfl.ch)

I. Introduction

Energy Dispersive X-ray Spectroscopy (EDS, EDX or XEDS) [1] is a qualitative and quantitative X-ray micro-analytical technique that can provide information on the chemical composition of a sample. In our context Scanning Transmission Electron Microscopy (STEM)- EDS is consideredd, which converges an electron beam on the sample. The electrons from the primary beam penetrate the sample and interact with the atoms from which it is made. Two types of X-rays result from these interactions: Bremsstrahlung X-rays, which means 'braking radiation' and are also referred to as Continuum or background X-rays, and Characteristic X-rays. The X-rays are detected by an Energy Dispersive detector which displays the signal as a spectrum, or histogram, of intensity (number of X-rays or X-ray count rate) versus X-ray energy. The energies of the Characteristic X-rays allow the elements making up the sample to be identified, while the intensities of the Characteristic X-ray peaks allow the concentrations of the elements to be quantified.
Given the EDS Data of a sample, this project aims to obtain the spectrum of the pure phase present in the sample where several phases are overlapping. The baseline results available are from using Hyperspy [2] - an open source Python library which provides tools to facilitate the interactive data analysis of multi-dimensional datasets that can be described as multi-dimensional arrays of a given signal. The goal of this project is to improve upon the results obtained from Hyperspy, and also generate a mask in Python to obtain the physical locations of the different components in the sample.

II. Problem formulation

We see in Fig 1 how the EDS data is captured. Corresponding to each pixel, we have an energy spectrum of 2048 values. For the scope of this project we consider a sample which has 2 distinct elements - A. Pure Fp (Ferropericlase) phase and B. Pure Brg (Bridgmanite) phase. The Brg phase makes up the matrix phase, while the Fp phase is present in chunks spread across the sample, at varying depths. For clarity we will refer to the Brg matrix phase as "background", which should not be confused with Bremsstrahlung (brake x-rays) in the Electron Microscopy domain. The HAADF(High-angle annular dark-field) image of the sample is shown in Fig 2. It shows the locations the Fp phase spread within the Brg matrix phase in the electron beam illuminated direction. Every pixel either has pure Brg phase or a mixture of Brg and Fp phase. The two major objectives of the project are:

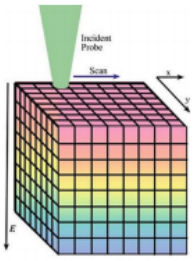

Figure 1. Figure depicting capture of EDS data.

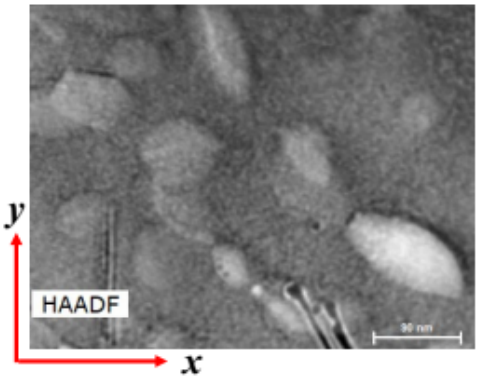

Figure 2. HAADF image of the sample.

1) Mask generation : By identifying and separating the spectra of the two phases, we would like to generate a mask, which shows the physical locations of the presence of the Fp phase.
2) Obtaining pure spectrum : The spectrum we obtain for the Fp phase is not pure and contains characteristics of the Brg phase as well. We would like to separate these using machine learning techniques and domain knowledge to obtain the pure spectra.

This problem is often referred to in literature as 'Hyperspectral data Unmixing'.

One of the most popular methods used is 'Non-negative Matrix Factorization', since constraining the factors to be non-negative ensures that energy is always non-negative across the spectrum. Formally, the data matrix X contains

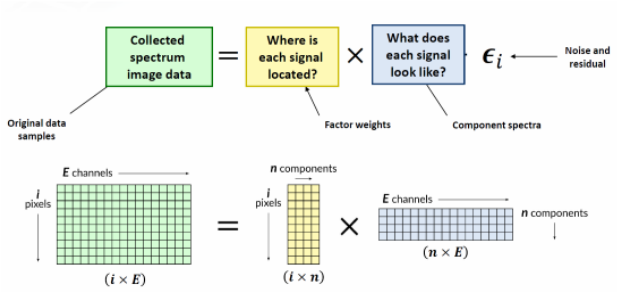

Figure 3. Matrix factorization for Hyperspectral Data Unmixing.

n of pixels rows, one for each pixel containing its energy spectrum of 2048 values, and under the assumption that the spectrum of each pixel is a linear combination spectrums of Fp and Brg phase, our goal is to decompose X into $W \times H$ where $W \in R_+^{n\times 2}$ and $H \in R_+^{2\times 2048}$ In the first part of the project, described in Section III, we show the results obtained using the Hyperspy python library, and the limitations of it. We describe our method to generate the mask of the image in Section IV. We then implement our own version of NMF using alternating least squares, and choose optimum initialization and regularization constraints using domain knowledge, described in Section V. Finally, we apply alternate clustering techniques and Vertex Component Analysis(VCA) for our problem, and compare the results obtained, in Section VI. To reduce computational time we work with 100×100 patch of the original data.

In all the consequent sections, for ease of notation we will refer to the Fp phase as *Phase A* and the Brg phase as *Phase B*.

III. Hyperspy

Hyperspy is an open source Python library which provides tools to facilitate the interactive data analysis of multi-dimensional datasets that can be described as multi-dimensional arrays of a given signal. It is often used to manipulate and work with EDS Data [2]. Hyperspy implements various spectrum unmixing techniques. We first apply the standard NMF - non-negative matrix factorization implementation provided by hyperspy. In Fig 4 we see the decomposition results using the NMF within the Hyperspy library. The two different colored plots represent the estimated separated spectrum of the two phases, phase A and phase B.

In each spectrum, a peak indicates the presence of the element corresponding to that energy level. A peak around 225 eV corresponds to the element Silicon (Si). We know from domain knowledge that Phase A (Fp phase) should not contain any Si, while Phase B (Brg phase) does. But in Fig 4 we see a Si peak for Phase A too.

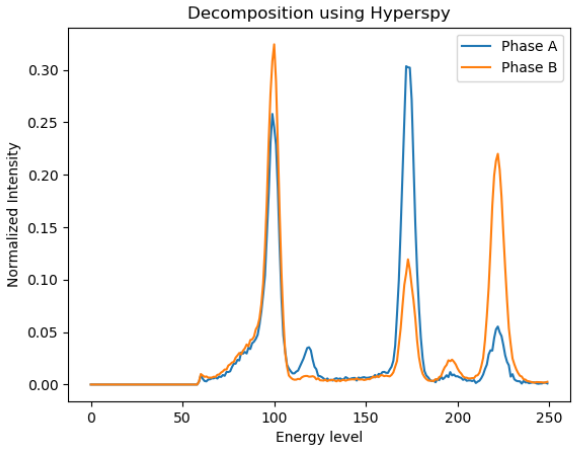

Figure 4. Results of NMF decomposition using Hyperspy, avg normalized intensity of Si - 0.0256.

The decomposition obtained here is used to get the composition of the pure phase present in the material. If the spectrum in decomposition of the pure phase is not accurate this would lead to error in the prediction of pure phase. The other problem with the NMF decomposition is that it is not unique. We would want the decomposition which is relevant with the domain knowledge we have.

IV. Mask Generation

Our first goal is to generate a mask where the black region corresponds to presence of Phase A+B and white region to the presence of Phase B. Let $W \times H$ denote the NMF decomposition obtained from hyperspy and each row of W contains weights corresponding to Fp and Brg phase spectrum for each pixel. We classify a pixel belonging to pure Brg phase if the ratio of the corresponding Brg and Fp phase weights is above a certain threshold. Then we detect the connected components in the intermediate mask and retain only those whose size is significant [3]. Next, we apply a Gaussian blurring, to further smooth the mask and get the final mask. Fig 5 shows the steps and the final mask generated.

V. NMF using ANLS (Alternating Non-negative Least Squares)

As observed in Figure 4, one of the key issues in the NMF decomposition is the presence of a peak corresponding to silicon in the Phase A which, by prior domain knowledge, should not be present. Also the NMF decomposition is not unique and we want a decomposition which is consistent with domain knowledge. We use the Alternating Non negative Least Squares(ANLS) algorithm, an algorithm for performing NMF, and modify it to tackle the previously stated problems.

We want to use two aspects of the domain knowledge

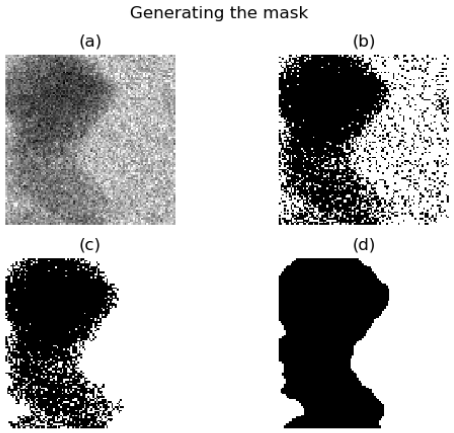

Figure 5. (a) Factor weights matrix for Phase B obtained from NMF. (b) After thresholding. (c) Keeping connected components of significant size only. (d) Applying Gaussian blur on the previous figure.

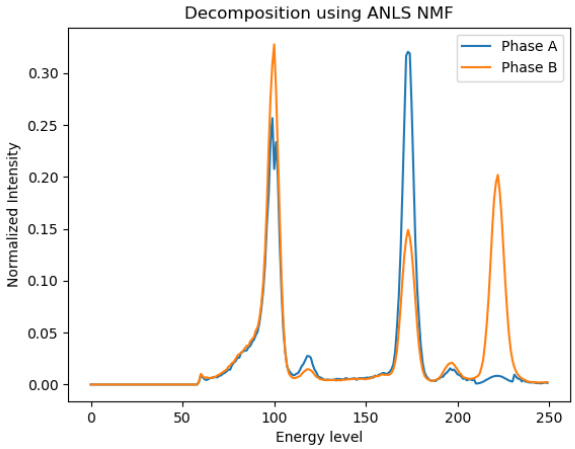

Figure 6. NMF decomposition using ALNS with intensity penalized and smoothness enforced in 210-230 keV (silicon energy level) of Phase A, avg. normalized intensity of Si - 0.005.

- the spectrum of the phase A shouldn't have a peak around Si energy levels.
- the mask generated above gives a heuristic for spectrum of Phase B and we should use the heuristic generated above.

To tackle the first problem, to enforce the algorithm to converge to a decomposition we add a term in the objective function for NMF that penalizes high intensity values corresponding to silicon in Phase A. More formally, we sum the squares of each intensity value in the range 210-230 keV of the Phase A, multiply this sum with a parameter λ_{Si} and add this to the objective function of NMF ($||V - WH||_F^2$ where V is the matrix to be decomposed and W, H are the estimates of the factors). To ensure smoothness of the spectrum in this range we also add the sum of squares of consecutive differences of intensities in the same range multiplied by λ_{Si}.

To generate a heuristic for Phase B, we take a mean along the pure Phase B region along the mask and obtain a heuristic spectrum for Phase B. We employ this heuristic to ensure that the estimated Phase B spectrum in each iteration of ANLS remains close to the heuristic by appending the heuristic Phase B spectrum to the rows of V and a $[w_0, w_1]$ row to the initial W and penalize large value of w_1 by adding a regularizing term $\lambda_{bkg} \times w_1^2$ to the loss and ensuring that it goes close to zero in the decomposition. It can be seen that such a decomposition with w_1 close to zero would result in the Phase B spectrum (the first row of H) close to our heuristic (otherwise the difference along the first row would contribute to the loss).

We implement the Alternating Non-Negative Least Squares (ANLS) algorithm to perform this decomposition and incorporating changes in the updates for each component in each iteration to account for the extra penalty terms to converge to a relevant decomposition. The initialization for each factors (W, H) are random. To enable faster converge of the ANLS decomposition for first few iterations, we run the Alternating Least Squares(ALS) decomposition as described in [4]. Note the ALS is computationally cheaper as it uses a closed form solution to solve the linear regression but it is not possible for ALNS due to the constraint of Non-negativity.

To tune the parameters λ_{bkg} , λ_{Si} and γ(learning rate) we run the algorithm over various range of the parameters and picked the best parameters which gives the lowest average of the intensity in the Si range 210-230 eV. But for higher values of these parameters this gives a decomposition as shown in Figure 7 for $(\lambda_{bkg}, \lambda_{Si}, \gamma) = (0.1, 1, 0.1)$ which has a valley in this range (as higher regularization terms would highly penalize even small intensities) which is not physically desirable, so we only consider decomposition with at least 10^{-4} average Si intensity. The intensities for the higher energy levels(above 800) are very close to zero, so we run our decomposition on this truncated data.

The final decomposition using the best parameters $(\lambda_{bkg}, \lambda_{Si}, \gamma) = (0.01, 1, 0.01)$ obtained is shown in Figure 6. Note that in each spectrum decomposition figure, we normalize each phase by the norm of their corresponding row in H. We only display the energy levels from 0-250 eV in the figures. We clearly observe that the average intensity of Si is lower than in Figure 4, which indicates a desired output. The precise values of avg. Si intensities for each plot are provided in their corresponding captions.

VI. Other approaches

In this section we present various other approaches used in the hyperspectral data unmixing literature and discuss their

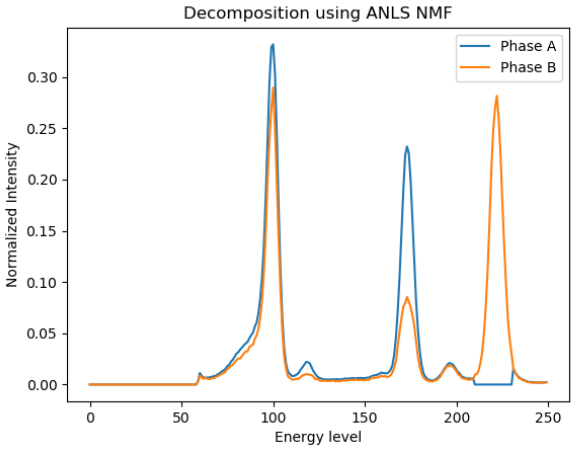

Figure 7. NMF decomposition with a spurious valley for high regularizing coefficients in 210-230 keV (silicon energy level) of Phase A, avg. normalized intensity of Si - 0.00.

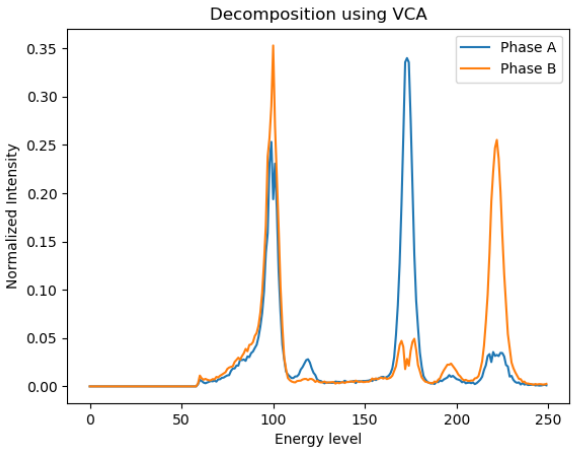

Figure 8. Decomposition using VCA. avg. normalized intensity of Si - 0.022

performance for our problem.

A. Vertex Component Analysis (VCA)

The vertex component analysis (VCA) is a method for unsupervised end-member extraction from hyperspectral data assuming that the data is a linear mixture of pure components (spectra). VCA assumes the presence of pure pixels in the data. It extracts a predefined number of pure components and estimates both the pure spectra and the concentration maps of the found components [5].
We implemented VCA [6] on our data and the results are shown in Fig 8.

Analysis: We see that the spectrum of Phase A and Phase B are both modified from the original. We observe that since VCA assumes the presence of pure components, it tries to cut off the peaks in unnatural ways, which leads to the gap we see, which has no physical significance which is also evident from the average Silicon intensity. Since in our sample, we do not have pure components, this is not a good method.

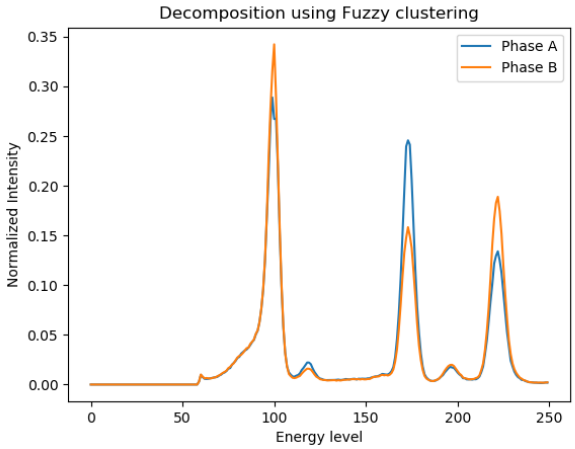

Figure 9. Decomposition using Fuzzy Clustering. avg. normalized intensity of Si - 0.063

B. Clustering

Clustering has been recently proposed in literature [7], [8], [9] because it is insensitive to spectral collinearity which can arise due to phase mixing. In the context of hyperspectral unmixing, clustering can be implemented by considering the energy spectrum of each pixel as the location of that point in a $d(= 1024)$-dimensional space. By clustering these points, using $k = 2$, we get two cluster centroids. These cluster centroids would then correspond to the energy spectrum of the two different components present in the sample.
Mainly 2 types of clustering are used- KMeans [10] and Fuzzy clustering [11]. As a pre-processing step to clustering, it is suggested that dimensionality reduction using Singular Value Decomposition (SVD) be performed. We used SVD to reduce the dimensionality of the given data matrix to 2 (the number of phases present). We then tested both K-Means and Fuzzy clustering with similar results. In Fig 9 we show the results obtained by applying fuzzy clustering on the dimensionality reduced matrix.

Analysis: We clearly see that the two decomposed spectra are quite close to each other, and quite close to the average of the spectra of phases A and B. This happens because in our case, Phase B is largely dominant over Phase A in terms of composition. While clustering is known to work well when the two components are present in approximately equal fractions.The average silicon intensity is also higher.

ACKNOWLEDGEMENTS

We thank Chen Hui for guiding us throughout the project and helping us understand the domain knowledge required in the project. We also thank Prof Cecile Hebert for hosting us to work in her lab.

REFERENCES

[1] "My scope eds description," https://myscope.training/legacy/analysis/eds/mapping/#toggleMenu.

[2] "Hyperspy multi-dimensional data analysis toolbox," https://hyperspy.org/.

[3] "Connected components in python," https://scipy-lectures.org/packages/scikit-image/auto_examples/plot_labels.html.

[4] N. Gillis, "The Why and How of Nonnegative Matrix Factorization," *arXiv e-prints*, p. arXiv:1401.5226.

[5] "Vertex component analysis," http://www.imagelab.at/help/vca_ui.htm.

[6] "Vertex component analysis for python," https://github.com/Laadr/VCA.

[7] C. L. Stork and M. R. Keenan, "Advantages of clustering in the phase classification of hyperspectral materials images," *Microscopy and Microanalysis*, vol. 16, no. 6, pp. 810–820, 2010.

[8] C. M. Parish, "Fuzzy clustering to merge eds and ebsd datasets with crystallographic ambiguity," *Microscopy and Microanalysis*, vol. 25, no. S2, pp. 134–135, 2019.

[9] B. H. Martineau, D. N. Johnstone, J. F. Einsle, P. A. Midgley, and A. S. Eggeman, "Data clustering and scanning precession electron diffraction for microanalysis," *Microscopy and Microanalysis*, vol. 23, no. S1, pp. 116–117, 2017.

[10] "K-means clustering," https://towardsdatascience.com/understanding-k-means-clustering-in-machine-learning-6a6e67336aa1.

[11] "Fuzzy clustering," https://ncss-wpengine.netdna-ssl.com/wp-content/themes/ncss/pdf/Procedures/NCSS/Fuzzy_Clustering.pdf.