Mapping Grain Boundary Metastable States with Unsupervised Learning

Derek Hensley

A senior thesis submitted to the faculty of
Brigham Young University
in partial fulfillment of the requirements for the degree of

Bachelor of Science

Gus Hart, Advisor

Department of Physics and Astronomy

Brigham Young University

ABSTRACT

Mapping Grain Boundary Metastable States with Unsupervised Learning

Derek Hensley
Department of Physics and Astronomy, BYU
Bachelor of Science

Grain Boundaries (GBs), the interfaces between individual crystals in metals, influence many of the physical properties observed in metals such as corrosion, electrical conductivity, and strength. I look to map the metastable states, states that are stable but not at the lowest energy, of specific GB subsets where the macroscopic parameters are kept constant. By using machine learning, I am able to cluster these GB subsets to possibly find the unique metastable states. Applying this technique to 1797 Σ5-(012) symmetric twist GBs, I found the optimal number of clusters was based on the representation of the GB that was used. While these clusters cannot be proven to correspond to the metastable states, analyzing the clusters based on Principal Component Analysis and energy gives confidence that they do. With knowledge of these metastable states, material design and GB engineering, the deliberate manipulation of GBs to improve properties, can be improved.

Keywords: Grain Boundary, Grain Boundary Engineering, Machine Learning, clustering, Grain Boundary metastability

Contents

List of Figures

Chapter 1

Introduction

1.1 Grain Boundaries

Most metals form by growing individual crystals, called grains, that pack togehter to form the metal. Of particular interest to material scientists are the regions where these grains meet. These interfaces, called Grain Boundaries (GBs) are depicted in Figure 1.1 (dark lines) and influence many of the physical properties of these materials [1, 2]. Corrosion [3], electrical conductivity [4], and strength [5] have all been shown to be influenced by GBs.

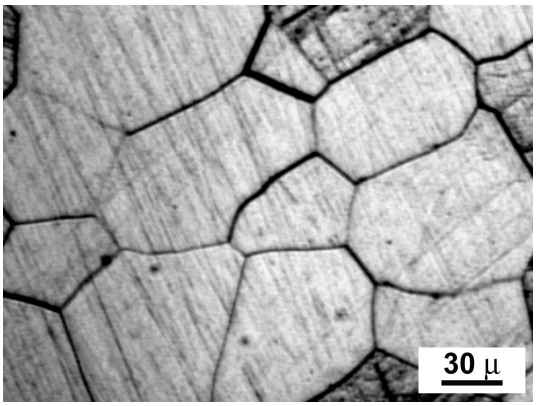

Figure 1.1 Grain Boundaries found in a metal. Image created by Edward Pleshakov.

While GBs are known to affect the physical properties of materials, this relationship is still poorly understood due to the complexity of GBs. To describe the just macroscopic structure of a GB requires five degrees of freedom (DOFs). Two DOFs are used to describe the orientation of the boundary plane separating the two grains, and the other three DOFs describe the orientation of one grain relative to the other. However, as GBs are made of many atoms, they also have a microscopic structure that requires more DOFs to fully describe the GB. In this thesis, I will be analyzing GBs that all have the same macroscopic five degrees DOFs, but differ in microscopic DOFs that will be discussed later.

1.2 Metastability

Metastability is when a system is in a stable state, but is not necessarily in its lowest energy state. For example, Figure 1.2 depicts a contrived energy landscape of some fictional system. The arrow points to a metastable state, as that state is stable, since it would require adding energy to change states, but is not the system's lowest possible energy. GB metastability is similar, but with multiple DOFs. GBs metastability involves more than just one dimension as depicted in Figure 1.2.

By studying microscopic metastable states within GBs, Han et al. [6] were better able to understand properties of GBs. This has implications for GB engineering, or the deliberate manipulation of GBs to improve the physical properties of the material [6]. Han et al. did this by analyzing GBs with the same five macroscopic degrees of freedom, but differing in two microscopic DOFs. In GBs, it is possible to shift one crystal with respect to the other, called a rigid body translation, that changes the microscopic structure, but leaves the macroscopic structure unchanged. Thus by performing two dimensional rigid body translations, the authors kept the five macroscopic DOFs constant, while adjusting two microscopic DOFs.

In this thesis I seek to expand on the work done by Han et al. [6] by adding an additional four

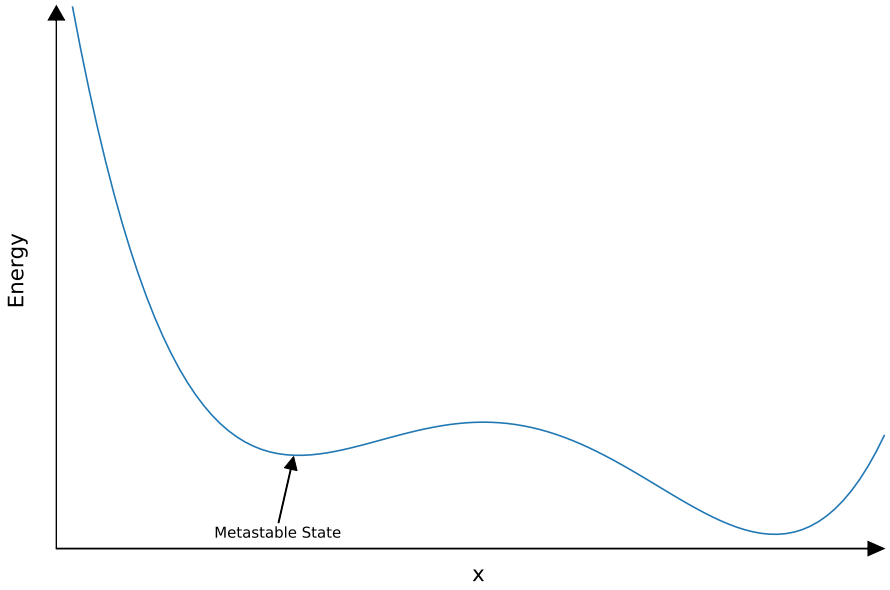

Figure 1.2 Contrived energy landscape that depicts a state that is not the lowest energy, but is still stable, i.e. metastable state.

microscopic DOFs. First I will perform a three dimensional, instead of two dimensional, rigid body translation adding one additional DOF. Another DOF comes from the position of the plane separating the two grains. The last two DOFs deals with the closeness of atoms. One DOF is the distance atoms can come together before overlap is too significant, and the other DOF is determining what to do with these atoms. In my case there are three possibilities when two atoms, atom A and atom B, get close enough together, either delete atom A, atom B, or delete both and place atom C at the midpoint of them both. Additional details on these DOFs can be found in work done by Homer [7]. Due to the complexity added by these additional DOFs, I use Machine Learning to analyze the data.

1.3 Machine Learning

Machine learning is a subset of artificial intelligence where the machine is able to take data and find patterns, thus 'learning'. By using machine learning, large and complex datasets can be analyzed. This is how large companies such as Google and Facebook are able to process the datasets they use. As the GB data is complex, machine learning would be a good choice to analyze it.

In machine learning there are many different types of learning, but here we will be using algorithms that fall under the unsupervised type. Unsupervised machine learning deals with datasets where only the data points themselves are known. The most common form of unsupervised learning, and the one used in this work, is *cluster analysis* where the machine finds patterns or groupings in the data. By clustering GBs, we will be able to see groupings of GBs that should correspond to metastable states.

1.4 Representations

The key to all machine learning algorithms is not the actual algorithm used, but how the data is represented. Imagine that you are conversing with someone that speaks a different language. While you both may be speaking about the same thing, it is still difficult to understand each other due to this language barrier. This is the same with computers. Thus the GBs need to be represented in a way that is descriptive enough that the computer understands.

My group previously studied several representations for describing GBs. With the Smooth Overlap of Atomic Positions (SOAP) algorithm [8], the group created two representations, the Averaged SOAP Representation (ASR) and the Local Environment Representation (LER) . Both of these have shown to be effective representations on GB datasets [9]. My group also studied the Scattering Transformation (ST), which proved effective, but not as good as the ASR and LER [10]. With these representations I have the tools necessary to represent our GB data in away the computer understands. Thus I can tackle this metastability problem using unsupervised machine learning.

1.5 Overview

In this thesis I discuss how to use clustering algorithms to map out the metastable states of GB systems. I first discuss the dataset that was used and how it was generated. Next, I describe the representations used to describe the GBs, along with the machine learning algorithms used to find GB metastable states. Finally, the results and conclusions of applying the technique to GBs are presented.

Chapter 2

Dataset

2.1 Generation

The first step to mapping the metastable states was to generate the required data. The generation process used is the same as used by Homer [7]. In short, the process starts with a representative GB that has the five degrees of freedom fixed. A grid search is then systematically performed by varying the microscopic DOFs to find potential GB structures with the same macroscopic DOFs. This is done by performing rigid body translations, adjusting the plane offset, adjusting the distance of atom overlap, and deciding how to manage overlapped atoms as described in section 1.2. Finally, atomic simulations can be run on these potential structures, using LAMMPS [11], to relax them to their local minimum energy using the conjugate gradient. By following this process, data can be generated for any GB system.

While this process is purely computational, studies have shown that these computationally generated GBs do, in fact, relate to physical ones. For example, Meiners et al. experimently showed GB structures predicted computationally [12]. This gives confidence that my computational work does have basis in physical GBs.

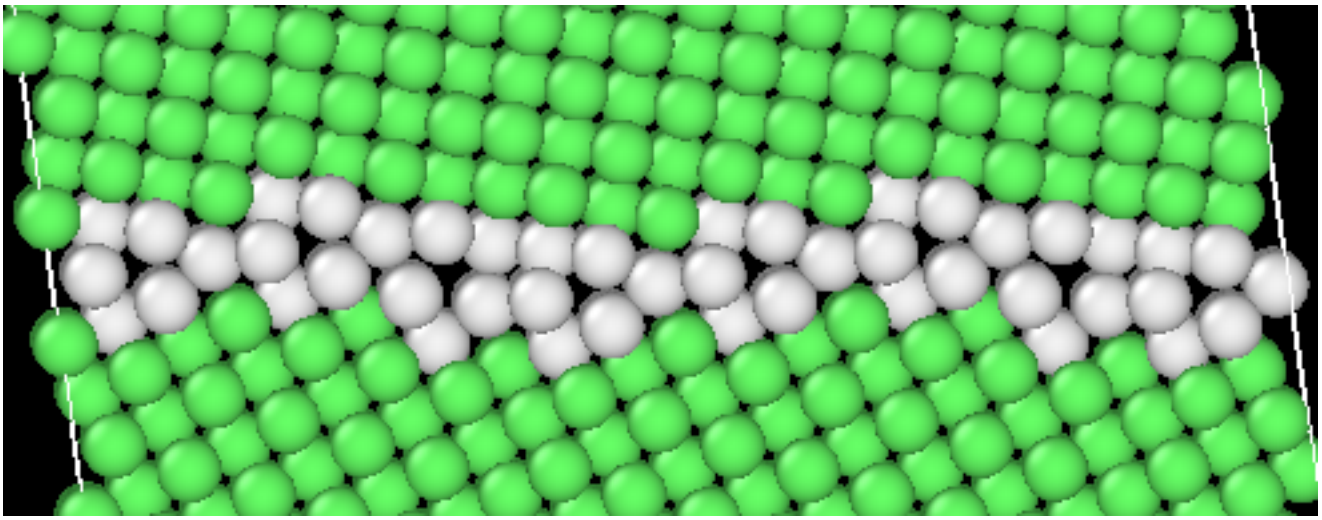

Figure 2.1 An example of one of the 1797 Σ5-(012) symmetric twist GBs. The grey atoms are atoms that make up the Grain Boundary, while the green ones are part of either crystal.

2.2 Description

Using generation process described in section 2.1, data was generated for the Σ5-(012) symmetric twist GB system (Figure 2.1). After generation and relaxation with LAMMPS, the dataset contained 1797 unique GBs with various microscopic DOFs, but the same macroscopic DOFs.

Chapter 3

Representations

3.1 Smooth Overlap of Atomic Position

The Smooth Overlap of Atomic Position (SOAP) formulation [8], depicted in Figure 3.1, can be used to obtain a representation for each GB in the dataset that can be used with clustering algorithms. Briefly, the SOAP descriptor is obtained by first placing a Gaussian function at each atomic position. The sum of these local Gaussians is then expanded using an orthonormal basis of radial functions and spherical harmonics. Next, the expansion coefficients are collected into a vector. This vector describes the atom's local neighbors. By following this procedure for each atom in the GB, a $Q \times N$ matrix is obtained where Q is the number of atoms in the GB and N is the number of basis functions used (which depends on tunable parameters).

SOAP has been implemented in a couple of Python packages, and in the present work, the Python package `pycsoap`[1] [13] was used to generate each SOAP matrix. There are three free parameters; r_{cut}, the distance describing an atom's local neighbors, n_{max}, the number of radial basis functions, and l_{max}, the number of spherical harmonics. I refer to work done by Rosenbrock et al. [14] for a discussion on choosing these parameters. In the present thesis, r_{cut} was chosen to be 5

[1] This is available from the Python Package Index using `pip install pycsoap`.

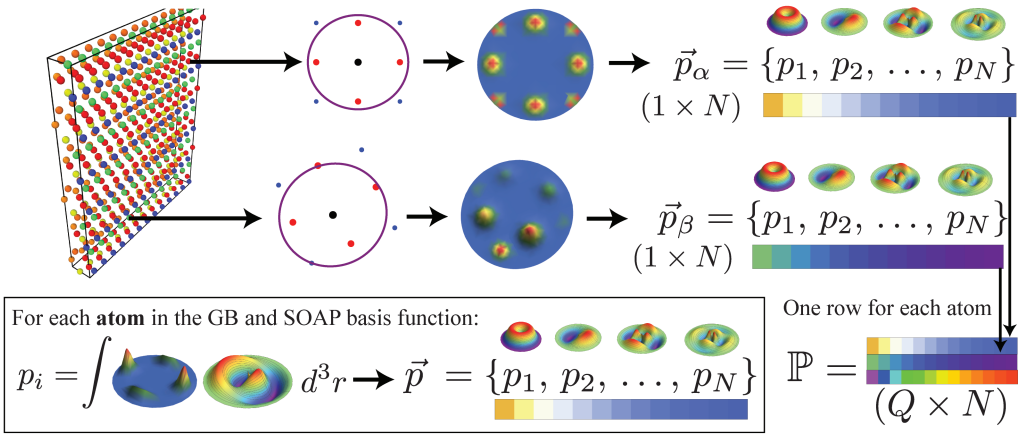

Figure 3.1 A schematic drawing of the SOAP descriptor. We place a Gaussian function on each atom in a local environment. The summation of these functions can be projected into a spectral basis consisting of radial basis functions and spherical harmonics, as shown in the boxed region. Following this process for each atom, a matrix of size $Q \times N$ is formed where Q is the number of atoms and N is the number of basis functions. This diagram was used with permission [9].

angstroms, and both n_{max} and l_{max} were 9.

While SOAP does produce a representation matrix for each GB, this representation cannot be directly used in any machine learning algorithm. Machine learning algorithms require a single matrix, called a feature matrix, where each row is a fixed length vector for each data point in the dataset. SOAP produces a matrix for each GB that is a variable sized matrix dependent on the number of atoms in the GB. Thus the SOAP matrix will be used as a starting point to generate two other representations, the Average SOAP Representation (ASR) and the Local Environment Representation (LER).

3.2 Averaged SOAP Representation

The first representation generated using the SOAP matrix is the Averaged SOAP Representation (ASR) [9]. The ASR is obtained by a simple average of each column of our $Q \times N$ matrix, i.e.,

averaging over the atoms. Averaging produces a single vector of length N that represents the average local atomic environment of all the atoms in the GB. The ASR is shown graphically in Figure 3.2. By collecting each of these ASR vectors from each GB in the dataset, a feature matrix is produced that can be used for machine learning.

3.3 Local Environment Representation

The other representation that I generate from SOAP is the Local Environment Representation (LER) [9] shown in Figure 3.2. I start by reducing the full SOAP matrices from each GB into unique SOAP vectors, a row from the SOAP matrix, clustered using the Euclidean distance. Two SOAP vectors are considered to be a part of the same cluster if their Euclidean distance is less than some distance threshold ε. After finding these unique clusters, the matrices are looped through to classify each SOAP vector into its cluster. With each SOAP vector classified, a histogram is compiled of the number of each unique vector for each GB. This histogram then produces an LER vector of the fractional abundance of each unique vector. Once again collecting the LER vector from each GB produces a feature matrix.

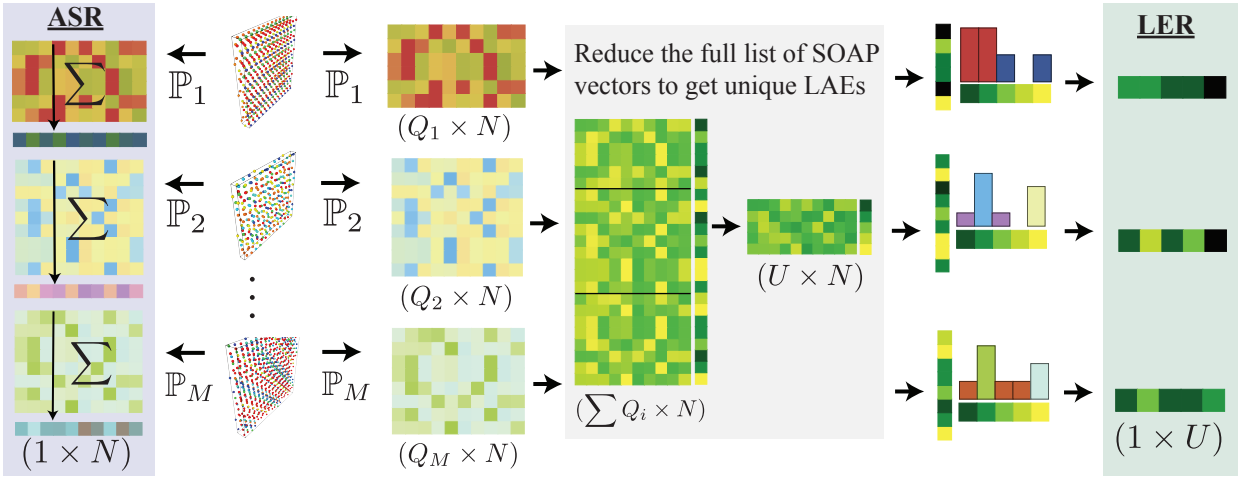

Figure 3.2 An illustration that describes how to produce both the Averaged SOAP Representation (ASR) and the Local Environment Representation (LER). On the left we just average the columns of the SOAP matrix to produce a single ASR vector. On the right we first reduce the full set of soap vectors from each GB to a set of unique vectors, called local atomic environment (LAE), based on Euclidean distance. Than a histogram is generated of how many unique vectors are in a GB's SOAP matrix. This histogram than produces a new LER vector of fractional abundances, whose components add up to 1. This diagram was used with permission [9].

Chapter 4

Unsupervised Machine Learning

4.1 Clustering Algorithms

With feature matrices that represent our GBs, I now apply different machine learning algorithms to help understand them. For this thesis, unsupervised machine learning algorithms are used. More specifically two different clustering algorithms, KMeans and Agglomerative, will be used to find clusters within the dataset.

4.1.1 KMeans

KMeans starts by initializing k random cluster centers, where k is defined by the user. Each point is then assigned to a cluster based on the distance to each cluster center. The cluster centers move to minimize the distance between itself and each point in its cluster. The points are than reassigned based on the new cluster centers, and the process repeats until the cluster centers stay in the same position for two iterations. In the present thesis, the KMeans implementation found in the Python package `scikit-learn`[1] was used.

[1] This is available from the Python Package Index using `pip install scikit-learn`.

4.1.2 Agglomerative

To help validate the clusters found by KMeans, Agglomerative clustering is also used to cluster the data. Agglomerative clustering starts by labeling each data point as its own cluster. At each step, two clusters are merged together into one based on user defined criteria. In my case, I use Ward agglomerative clustering. Ward's method is to merge the two clusters that minimizes the increase of within-cluster variance, i.e. the squared euclidean distances between points within the cluster. This process continues until there are only k clusters left, where k is a parameter defined by the user. The Agglomerative clustering implementation found in the Python package c was used in the present thesis.

4.2 Analysis Methods

4.2.1 Silhouette Score

Both of the machine learning algorithms used here require that the number of clusters be specified beforehand. However, one of the major goals is to determine the optimal number of clusters, i.e., metastable states. To overcome this challenge, the silhouette score will be used. After the data has been clustered by some algorithm, this silhouette score measures how similar each point in a cluster is to every other point in the same cluster, as well as how dissimilar each point in the cluster is to every point outside its cluster. This is done by the following formulation for a single point, which can than be applied to every point and averaged to determine the overall silhouette score. For a single point in cluster C_i, its dissimilarity to every other point in C_i is calculated as

$$a(i) = \frac{1}{|C_i| - 1} \sum_{j \in C_i, i \neq j} d(i, j) \tag{4.1}$$

where $d(i, j)$ is the euclidean distance between i and j. Then the point's dissimilarity to every other cluster is also calculated as

$$b(i) = \min_{k \neq i} \frac{1}{|C_k|} \sum_{j \in C_k} d(i, j) \tag{4.2}$$

The silhouette value is defined for point i as

$$s(i) = \begin{cases} \frac{b(i)-a(i)}{\max(a(i),b(i))} & |C_i| > 1 \\ 0 & |C_i| = 0 \end{cases} \tag{4.3}$$

Thus, for the silhouette score to be maximized $a(i)$ has to be small, i.e., the average distance to points within its cluster is small, while $b(i)$ needs to be large, i.e., the average distance to its next closest cluster is large. Figure 4.1 shows how the silhouette scores varies based on different scenarios. Notice how the silhouette score is maximized with clusters that make the most intuitive sense.

I can use the silhouette score to find the optimal number of clusters by first using each algorithm for a range of cluster numbers. For each number of clusters I calculate the silhouette score. The max of these is the optimal number of clusters since it has clusters that are most similar to themselves, and most dissimilar to each other.

4.2.2 Visualizing the Data

After I apply the machine learning algorithms and find the optimal number of clusters, I also desire a way to view the data. However, both the LER and ASR representations are high dimensional, making them extremely difficult to visualize. Thus I will use two different methods to help visualize the data.

The first method is Principal Component Analysis (PCA) [15] which can be used to reduce the dimensions down to two. By plotting the PCA, the overall dataset can be visualized along with the clusters found allowing for inspection and verification. In this work, the PCA implementation found

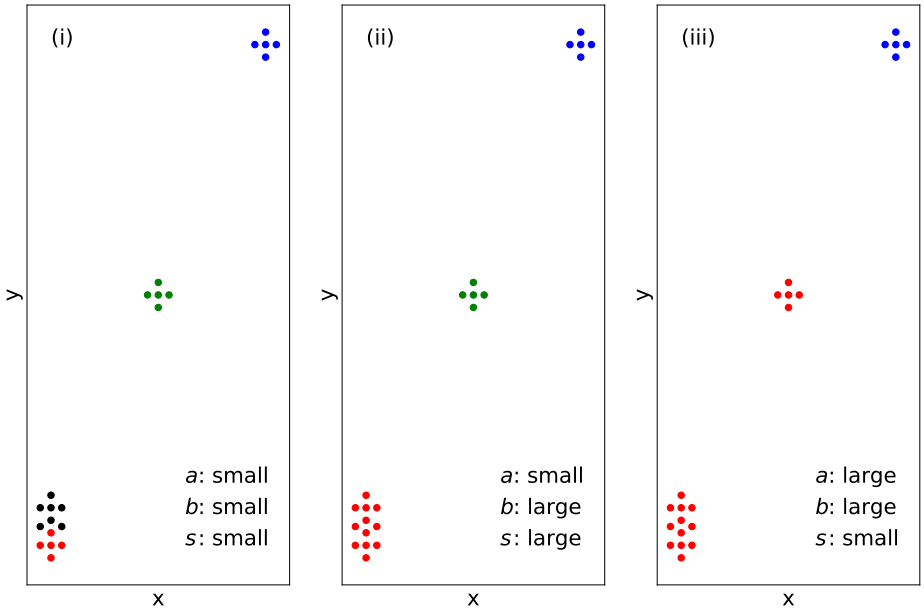

Figure 4.1 A diagram depicting silhouette score variation where colors represent clusters found by some algorithm. (i) has a smaller distance between different clusters (b) reducing the score. (iii) has a large distance between different clusters (b) and small distances within clusters (a) providing the best score. (iii) has a large distance within clusters (a) which reduces the score.

in the Python package `scikit-learn` was used.

The second method is to simply plot the GBs based on their energy and cluster. This will allow a comparison of how clusters compare based on energy.

Chapter 5

Results and Conclusions

5.1 Results and Discussion

With these tools I can now start mapping the metastable states of GBs. The first step was to calculate the silhouette score for various number of clusters to find the optimal number. Figure 5.1 graphically shows this silhouette scores for various numbers of clusters based on representation and algorithm. The peaks of the graph correspond to optimal cluster numbers since the silhouette score is maximized. While each representation shows consistency between both machine learning algorithms in the optimal number of clusters, each representation varies widely from each other. For instance ASR, shows that around 30 clusters is optimal, however LER shows closer to 800 clusters.

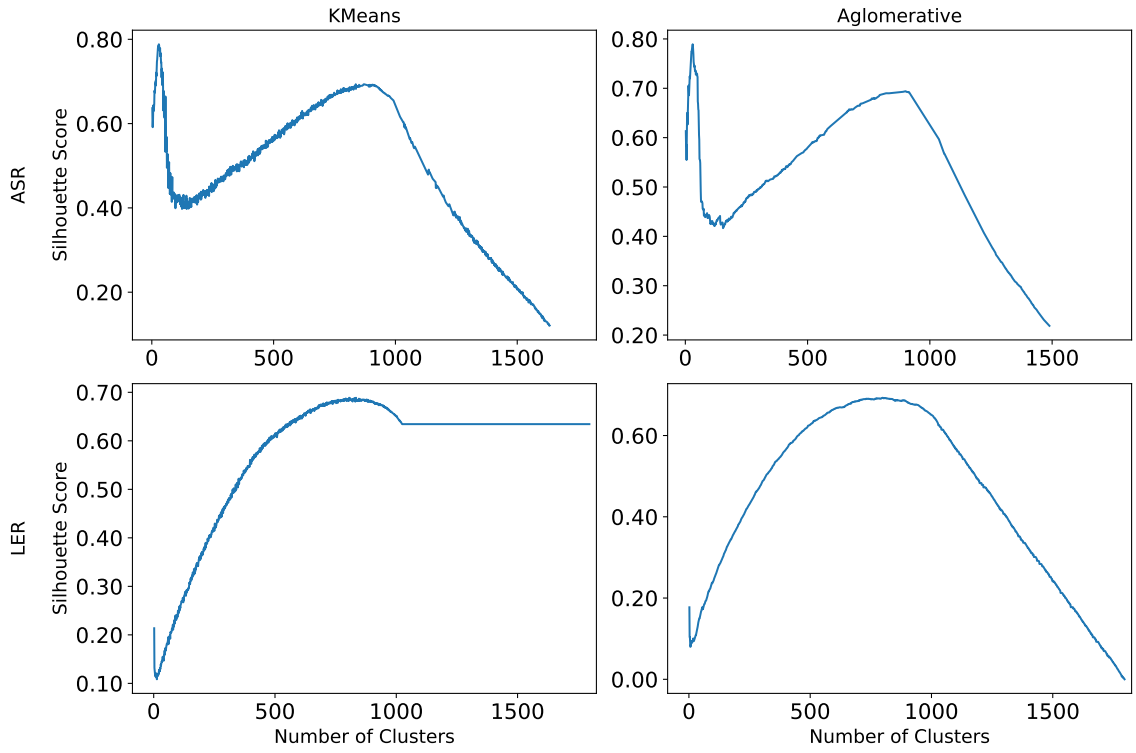

Figure 5.1 The silhouette scores for varying number of clusters based on both representation and algorithm. The optimal number of clusters corresponds to the peak silhouette score.

While it may seem concerning that each representation differs in the number of clusters found, each representation is finding metastable states based on different information. For a simple illustration, Figure 5.2 shows a contrived potential energy landscape. Note that in the circled regions there are multiple small dips in the graph, indicating multiple metastable states. However, each of these regions also belong in a larger basin and thus could be classified as a single metastable state.Thus it is possible that LER is locating more of these smaller metastable state, while ASR picks up more global ones.

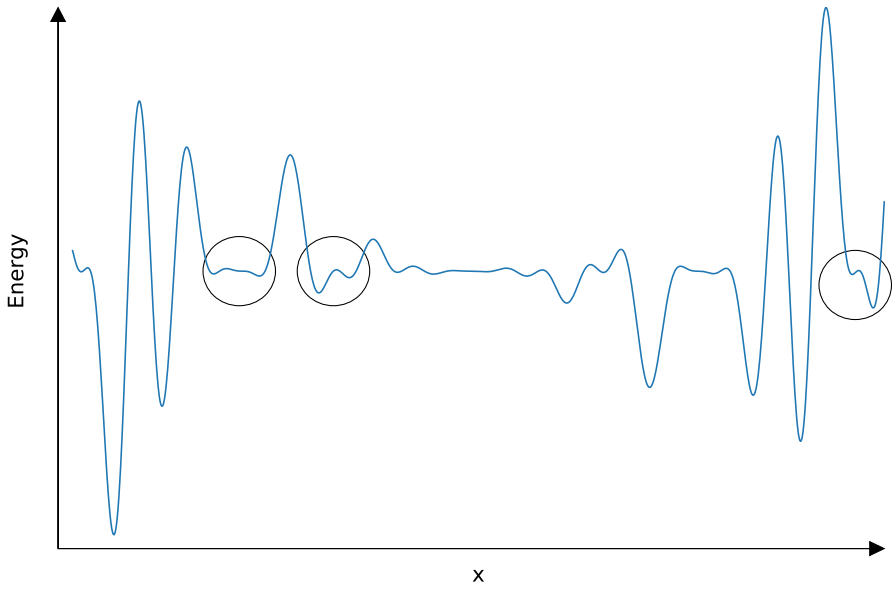

Figure 5.2 Contrived potential energy landscape. The circles show multiple metastable states that could be classified as one since they are part of a larger basin.

Examining the actual clusters gives added confidence that these clusters correspond to GB metastable states. Figure 5.3 depicts these clusters mapped down to 2D using PCA. Note that the the clusters found by each algorithm are well-defined with clear distinction between the clusters. While this gives added confidence that these clusters represent metastable states, as metastable state clusters would be well-defined, this does not prove the clusters are the metastable states. Thus further work must be done to prove these clusters correspond to the metastable states.

Plotting the clusters by energy also adds insight as to if these clusters correspond to the metastable states. In Figure 5.4 each GB is plotted based on its energy, and colored based on its cluster. The ASR graph adds credence to the fact that the clusters correspond to the global metastable states. If ASR was in fact finding the global metastable states, it would be expected

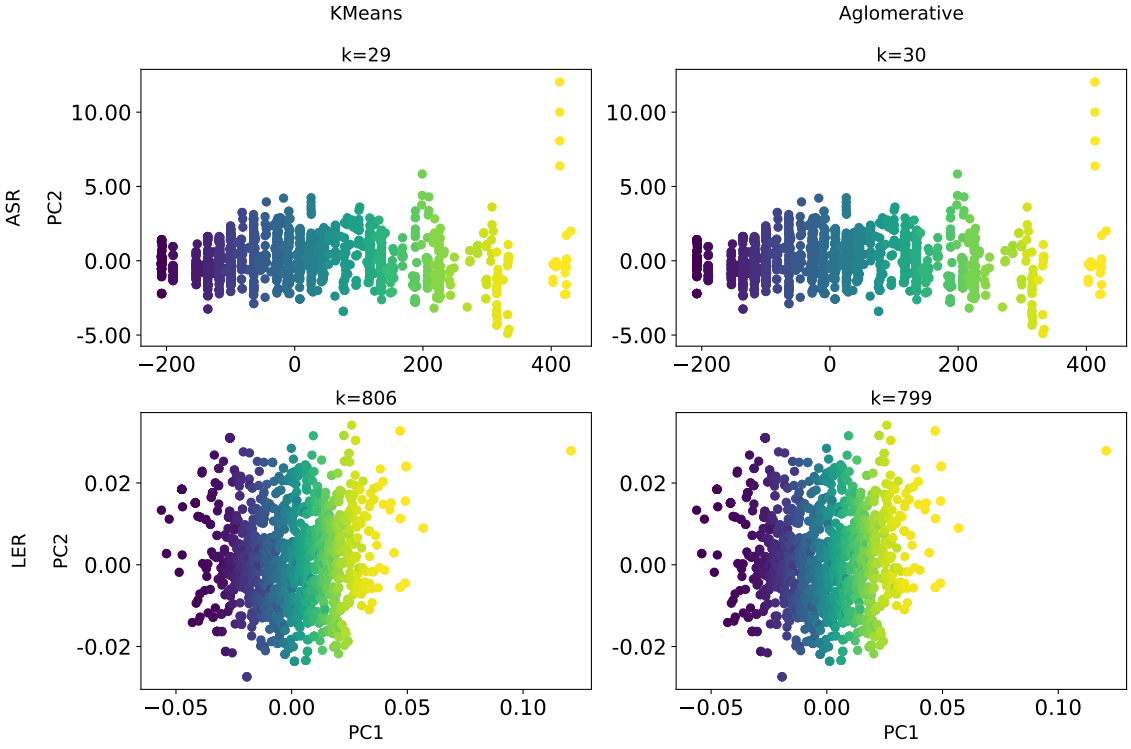

Figure 5.3 The clusters, depicted by different colors, found by various Machine Learning Algorithms (columns), using various representations mapped to 2D with PCA. k is the number of clusters found.

that GBs within clusters would not necessarily have the same energy, since there could be multiple smaller basins within the larger basin found. It would also be reasonable to find multiple clusters to have the same energy as there could be multiple basins in the potential energy map with the same energy. In the ASR graph, these are the patterns seen as there are points with the same color, i.e. the same cluster, at different energies levels and multiple colored points at the same energy. On the other hand, the LER graph also gives credence that the clusters correspond to more find tuned metastable states since there is a smooth change in energy that could correspond metastable states. Thus, while this again adds confidence that these clusters do in fact correspond to metastable states, it can not be proven and further work is needed.

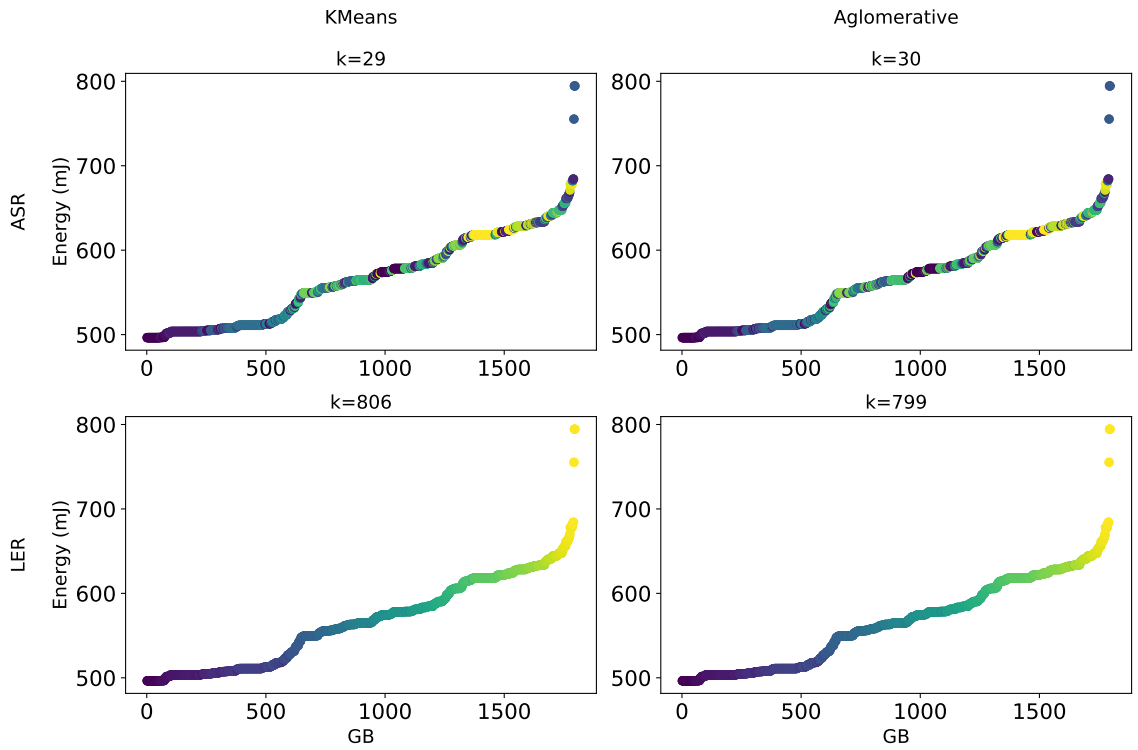

Figure 5.4 The clusters, depicted by different colors, found by various Machine Learning Algorithms (columns), using various representations plotted by energy.

5.2 Further Work

While initial results from this technique are positive, they do not prove its accuracy. Thus the next step is to prove that this technique is accurate. This can be done by annotating the dataset, or using slower, more computationally expensive methods where the metastable states can be more easily analyzed. Comparing the results from each method will validate the accuracy of this technique.

Further avenues of application deal with the enormous amount of data generated for GBs. As part of this research, more data has been generated for GBs with different macroscopic DOFs. Generating all these GB systems provides too much data; it is difficult to store it all. Once the accuracy of this technique is assured, applying this technique to these other systems will allow a fast way to map the metastable states of these other systems. Not only will this help in understanding the

atomic structure of these other GB systems, it will help solve this data storage problem. Mapping these metastable states and only storing their unique metastable states would still provide an accurate sampling of the GB system. Even reducing thousands of GBs down to hundreds will drastically save storage space, and allow for the generation of even more data.

Not only could this technique help with data storage, once its accuracy is proven, it could be useful in material design. As discussed above in section 1.2, understanding metastability of GBs improves Grain Boundary engineering. By applying this technique to other systems, their metastable states could be quickly mapped and understood for use in material design.

5.3 Conclusions

In this thesis I attempted to map the metastable states of Grain Boundary systems using unsupervised machine learning. While this technique's accuracy could not be proven, it shows promise in analyzing large GB datasets and their metastable states quickly. With greater understanding of more GBs, GB engineering will become more efficient and improve the materials we use everyday.

Bibliography

[1] E. R. Homer, S. Patala, and J. L. Priedeman, "Grain Boundary Plane Orientation Fundamental Zones and Structure-Property Relationships," Scientific Reports **5,** 1–13 (2015).

[2] D. Wolf and S. Yip, *Materials Interfaces: Atomic-Level Structure and Properties* (Springer Science & Business Media, 1992).

[3] C. M. Barr, S. Thomas, J. L. Hart, W. Harlow, E. Anber, and M. L. Taheri, "Tracking the Evolution of Intergranular Corrosion through Twin-Related Domains in Grain Boundary Networks," npj Materials Degradation **2,** 1–10 (2018).

[4] J. Jose and M. Abdul Khadar, "Role of Grain Boundaries on the Electrical Conductivity of Nanophase Zinc Oxide," Materials Science and Engineering: A **304-306,** 810–813 (2001).

[5] S. Tsurekawa, T. Tanaka, and H. Yoshinaga, "Grain Boundary Structure, Energy and Strength in Molybdenum," Materials Science and Engineering: A **176,** 341–348 (1994).

[6] J. Han, V. Vitek, and D. J. Srolovitz, "Grain-Boundary Metastability and Its Statistical Properties," Acta Materialia **104,** 259–273 (2016).

[7] E. R. Homer, "Investigating the Mechanisms of Grain Boundary Migration during Recrystallization Using Molecular Dynamics," IOP Conference Series: Materials Science and Engineering **89,** 012006 (2015).

[8] A. P. Bartók, R. Kondor, and G. Csányi, "On Representing Chemical Environments," Physical Review B **87,** 184115 (2013).

[9] C. W. Rosenbrock, E. R. Homer, G. Csányi, and G. L. W. Hart, "Discovering the Building Blocks of Atomic Systems Using Machine Learning: Application to Grain Boundaries," npj Computational Materials **3,** 1–7 (2017).

[10] E. R. Homer, D. M. Hensley, C. W. Rosenbrock, A. H. Nguyen, and G. L. W. Hart, "Machine-Learning Informed Representations for Grain Boundary Structures," Frontiers in Materials 6 (2019).

[11] S. Plimpton, "Fast Parallel Algorithms for Short-Range Molecular Dynamics," Technical Report No. SAND-91-1144, Sandia National Labs., Albuquerque, NM (United States) (1993) .

[12] T. Meiners, T. Frolov, R. E. Rudd, G. Dehm, and C. H. Liebscher, "Observations of Grain-Boundary Phase Transformations in an Elemental Metal," Nature **579,** 375–378 (2020).

[13] A. H. Nguyen and C. W. Rosenbrock, "Streamlined Generation of SOAP Descriptors," , to be submitted.

[14] C. W. Rosenbrock, J. L. Priedeman, G. L. W. Hart, and E. R. Homer, "Structural Characterization of Grain Boundaries and Machine Learning of Grain Boundary Energy and Mobility," arXiv:1808.05292 [cond-mat, physics:physics] (2018).

[15] S. Wold, K. Esbensen, and P. Geladi, "Principal Component Analysis," Chemometrics and Intelligent Laboratory Systems **2,** 37–52 (1987).

Index