Understanding Grain Boundaries Through Machine Learning

in Regards to Hydrogen Embrittlement of Steel

Felicity Nielson

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

Understanding Grain Boundaries Through Machine Learning
in Regards to Hydrogen Embrittlement of Steel

Felicity Nielson
Department of Physics and Astronomy, BYU
Bachelor of Science

Steel is an incredibly valuable, versatile material. Unfortunately, high-strength steels are vulnerable to hydrogen embrittlement, a process that describes the degradation of a crystalline-structured material when too much hydrogen is absorbed. When enough hydrogen builds up, it can lead to early and unexpected failure of the material, which is both costly and dangerous.

Recent decades have seen a surge of efforts to solve this problem, but a general, viable solution has yet to be found. In this paper, we continue a new method using machine learning techniques in conjunction with atomic environment representations to predict global properties based on local atomic positions. Steel is comprised mostly of the base element iron. The defects in the iron crystal structure are where hydrogen prefers to adsorb. By developing a technique that will allow us to understand the global properties in these areas, future research will lead to predicting where the hydrogen will adsorb so that we can find another element that will non-deleteriously adsorb to those same sites, thus blocking the hydrogen and preventing hydrogen embrittlement.

This methodology can further be applied to any crystalline material, allowing engineers to understand the basic building blocks of what gives a material its properties. Its application will help improve the versatility of materials manufacturing, allowing manufacturers to precisely design a material with whatever properties a customer desires, enhance the properties of existing materials, and stabilize materials that so far only exist in theory.

Keywords: hydrogen embrittlement, steel, materials science, materials engineering, physics, grain boundaries, grain boundary engineering, grain boundary model, soap, machine learning, smooth overlap of atomic positions, local atomic environments, local environment representation, condensed matter physics

ACKNOWLEDGMENTS

I would like to acknowledge Dr. Gus Hart and Dr. Conrad Rosenbrock for their aid in my development as a student and researcher which has led to this paper. I would also like to acknowledge Wiley Morgan for answering my numerous questions, and Mark O'Neill for generously allowing me to steal a little bit of his time, knowledge, and resources.

Contents

Chapter 1

Introduction

1.1 Hydrogen Embrittlement

Hydrogen embrittlement is the name of a phenomenon that has plagued the steel industry since its inception. It describes the degradation of a material like steel due to detrimental hydrogen invasion. Steel is made by mixing small percentages of carbon and other alloying elements to a base of iron to make a material that can withstand intense stresses, pressures, temperatures, etc. Steel's properties have allowed us to build a plethora of useful inventions, from skyscrapers to stainless steel cutlery, and wind turbines to jet engines. This is possible because of iron's true greatest strength–its versatility. However, it is this versatility that is also iron's greatest weakness. Just as adding an alloying element has the potential to enhance the beneficial attributes of steel, adding the wrong element also has the potential to weaken it.

Hydrogen is one such problem element. Being a single proton, it is small and abundant in our environment. It gets into steel during manufacturing and after production during welding, galvanic corrosion, and other processes (Chen et al. 2017; Djukic et al. 2015). Too much hydrogen build up leads to early failure in steel (Chen et al. 2017; Tavares et al. 2017). In fact, there is a new steel that

is one third the weight of current steel while being just as strong, but it is so susceptible to hydrogen embrittlement (HE) that it cannot be used until a proper solution to the centuries old HE problem is found.

The difficulty with solving HE problem is that hydrogen will adsorb in areas of defects in the otherwise perfect crystalline lattice structure (Dunne et al. 2016; Jemblie et al. 2016; Luppo & Ovejero-Garcia 1991). The vast majority of defects constitute a high-dimensional space known as grain boundaries (GBs). GBs are dynamic surfaces, occurring where two or more crystals, or grains, with different orientations meet (Janssens et al. 2006; Lambert et al. 2017). The periodic structures of the lattices cannot match at the GB, so the atoms settle somewhere in between. The resulting environment is at a higher energy than it would have been had the perfect crystal structure been able to continue uninterrupted.

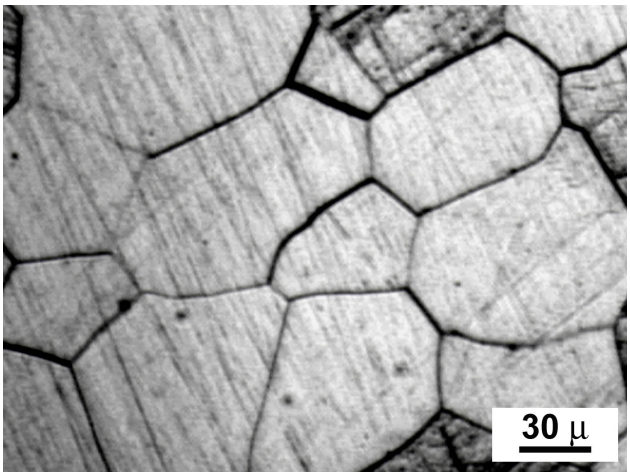

Figure 1.1 Grain boundaries in a polycrystalline metal. Image by Edward Pleshakov. Note that the grain boundaries in this image have been made more apparent by acid etching.

Many experiments have been done in hopes of understanding how hydrogen affects GBs (de Assis et al. 2017; Dunne et al. 2016; Randle 2006; Sirois & Birnbaum 1992; Takahashi et al. 2016; Zhou & Song 2017), and many models have been made in hopes of finding a solution to HE on GBs (Djukic et al. 2015; Gobbi et al. 2014; Jemblie et al. 2017; Pressouyre 1980; Tehranchi &

Curtin 2017; Wang et al. 2016). However, because of the numerous steel types, each with their own cocktail of GBs, and because of the randomness of the initial orientations and resulting interfaces, there is not yet a viable general model to fully explain or predict hydrogen's effect on GBs. Each case has to be done specifically to the steel type, of which there are thousands, each with unique properties.

There has been a recent development using machine learning to study GBs by Rosenbrock et al. (Rosenbrock et al. 2017). Our direction is to find a connection between local properties (atomic scale) and global properties (GB scale). By carefully representing atomic environments, we can use machine learning algorithms, such as decision trees, to predict GB properties knowing only what the Local Atomic Environments (LAEs) look like. Machine learning and decision trees are briefly discussed in Chapter 4, and LAEs are defined in Chapter 3, where we also define Local Environment Representations (LERs), an advancement on the LAE representation. This methodology has the advantage of being generalizable to any GB configuration and giving key insights into the physics behind the predictions. The ultimate goal for solving HE in steel is to find out where hydrogen preferentially adsorbs onto steel GBs, and then to find another element or material that will non-deleteriously adsorb to those same sites. For this reason, we need a hydrogen-steel database, but it turns out that building a database takes years, so while we wait for the hydrogen-steel database to be built, we have tested our methodology on the Imeall α-Fe GB database. This paper acts as sister to "Discovering the building blocks of atomic systems using machine learning: application to grain boundaries", by Rosenbrock, et al., to verify that the methodology works on an iron database just as well as it did on nickel.

1.2 Grain Boundary Engineering

There are two significant outcomes to this research. The first is directly solving the HE on steel problem. The second is its application to a more general GB engineering pursuit; purposely crafting GBs such that a material can be designed to have specifically chosen properties. The idea of GB engineering first started when interfaces between orientations were discovered in the 1880's, but the attention to GB engineering and its potential has grown exponentially in recent decades (Watanabe 2011). This ongoing area of research has already produced many significant results. One of these results, for example, is the realization that increasing the ratio of low-angle boundaries (or reducing the ratio of high-angle boundaries) reduces intergranular brittleness that can later lead to fracture, not unlike the degradation we see with HE. Since this revelation, engineers have been deliberately developing methods to influence the grain growth in favor of low-angle boundaries. Some of the methods include doping with the right element, laser surface processing, and rapid solidification followed by subsequent annealing. The method of rapid solidification with subsequent annealing has recently been employed with the addition of high strength magnetic fields to expel Sn from GBs in recycled steel (Watanabe 2011). The continuance of GB engineering will help lead to society's ability to master its own materials. It is our aim that the methodology presented in this paper will hasten this project by reducing the complexity of atomic space to reveal the key underlying physics.

When selecting the relative orientation of two crystals to produce a GB, there are 5 degrees of freedom that are macroscopic and experimentally measurable using backscattering techniques. An additional 3 degrees of freedom per atom exist that are microscopic and too difficult to measure experimentally with current technology. They deal with the placement of individual atoms at the interface. However, using known physics we can simulate the environments computationally. This allows us to measure the computed properties and match them to their experimental counterparts. We can then look for the important atom arrangements that contribute most to the global properties. In essence, we create a statistical distribution of the microscopic environment to the macroscopic

environment.

In terms of an analogy, if you have a class of 10 year olds and you want to know the height of a "10 year old" you would measure all of their heights and end up with a statistical distribution of their individual heights. The distribution looks very similar to a bell curve. The actual heights of the children are not random, they are determined by factors that are unknown to us, extra variables, extra "degrees of freedom," like their genetics, where they grew up, what they ate growing up, etc. In GBs, the extra unknowns are the extra $3N$ (N being the number of atoms) degrees of freedom that we cannot measure. In statistical GB engineering, we can map the microscopic degrees of freedom to the experimentally verifiable 5 degrees of freedom. This allows us to select and choose the macroscopic orientations that give us the most favorable attributes. The breakdown of how computational orientations are created is discussed in the next chapter.

We look towards a future where a manufacturer can pick out the orientations and alloying element combinations necessary to give the material he or she is designing its desired properties, and then make that material. While perfecting GB engineering may be far into the future, it is an inevitable road. Existing materials can also be altered, and with better methods, their sought-after attributes enhanced, and weaknesses minimized.

Chapter 2

The Database

The Imeall α-Fe database is computationally constructed using Embedded Atom Method (EAM) potentials. There are currently four variations on EAM potentials used for this database: Dudarev, Minish, Mendelev, and Ackland (Lambert et al. 2017). Metallic materials like steel are crystalline, meaning their base element atoms settles into predicable, periodic lattice structures. Steel's base element is iron. α-Fe is an allotrope, or one of multiple physical forms, that has a body centered cubic (bcc) lattice structure with eight nearest neighbors. See fig 2.1.

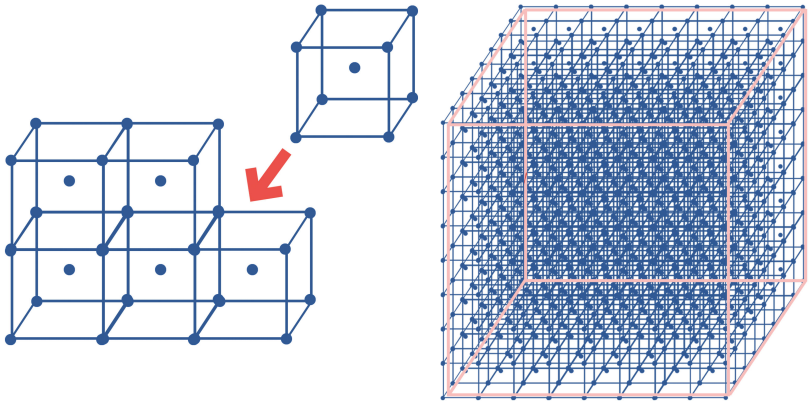

Figure 2.1 Periodic body centered cubic lattice structure. Each blue dot is an atom. Every atom has eight nearest neighbors.

Computational construction of a GB starts first with two superimposed crystals as seen in fig 2.2. One of the crystals is held in place while the other is rotated (Lambert et al. 2017). The first three degrees of freedom are determined when selecting rotation along the three Cartesian axes. Another two degrees of freedom are determined by selecting the position of the resulting plane or interface. The atoms that overlap beyond this plane into the other crystal are then deleted. These first five degrees of freedom are the macroscopic. An additional microscopic 3 degrees of freedom per atom result from selecting individual atom positions that may overlap or otherwise be in contest at the interface.

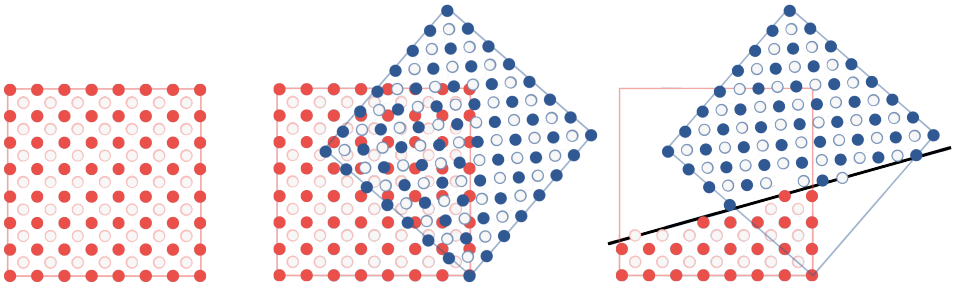

Figure 2.2 Crystal rotation and plane selection constitute the five macroscopic degrees of freedom when creating a GB.

After initial construction, the atoms are relaxed to their minimum energy using calculators in quippy, a software package that automates such processes (Bartók et al. 2013; 2010). In all, the Imeall database presents an exhaustive set of GB orientations of over 2800 GBs and several million atoms total. For more on building the Imeall database, see (Lambert et al. 2017).

Chapter 3

Representation of the Grain Boundary Environment

The GBs are isolated from the bulk of the atoms using common neighbor analysis (CNA) via the Ovito package (Stukowski 2010). CNA systematically runs through every atom in a given structure, identifying the target atom's neighbors and then classifying the target atom based on the separation distances between the neighbors. If the neighbors form a structure geometrically equivalent to a recognized structure in Ovito's CNA package, for instance bcc, it will then classify the target atom as bcc. If there is no recognizable atomic lattice, the Ovito package will classify the atom as 'other.' In this way, we can automatically locate the GB, where the atomic lattices misalign and form unexpected or random structures.

To represent the atoms, we use a descriptor based on the Smooth Overlap of Atomic Position (SOAP) method. Without a descriptor, the raw atoms represented by a database like Imeall's α-Fe GBs exist only as xyz values in a Cartesian coordinate system with corresponding calculated force vectors, CNA values, etc. This makes handling and comparing structures and environments difficult. For materials purposes, a good descriptor will be invariant to global translation, global rotation,

permutations of identical atoms, and will have compact support (brings an otherwise diverging function smoothly to zero). Essentially it needs to uniquely describe distinct environments in a way that provides access to accurate and efficient similarity calculations between two structures. SOAP provides this as well.

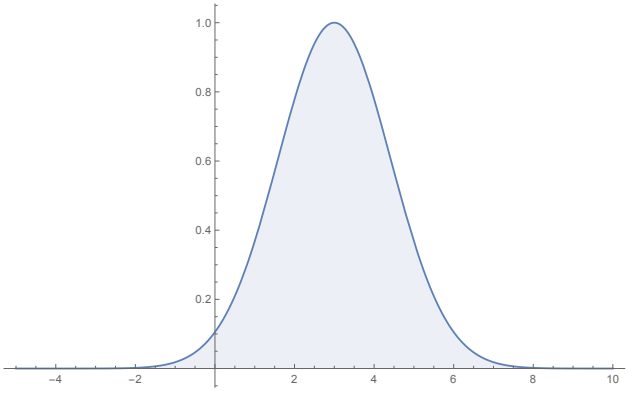

Figure 3.1 We use Gaussian density functions so that we can overlap the representation of an atom's influence and so that we can take advantage of the computational convenience of a Gaussian function. The figure shows a one dimensional Gaussian density function centered over atom 3.

The processes of representing a Local Atomic Environment (LAE) using SOAP descriptions starts by placing a Gaussian density function of the form,

$$e^{-(x-x_0)^2/\sigma^2}$$

over every atom, superimposing the functions within a chosen cutoff radius around an atom, and then projecting the superimposed function onto a spectral basis comprised of radial and spherical components. It is somewhat analogous to using a Fourier Transform to represent EM signals in three-dimensions. The resulting SOAP vectors of a single GB together form a matrix that represents the GB as a whole.

For the Imeall database we used a cutoff radius of 3.25 Å resulting in a LAE vector length of 1015. The average GB matrix had dimensions of (13640, 1015), but these dimensions varied depending upon how many atoms were in each GB, making comparison between them difficult.

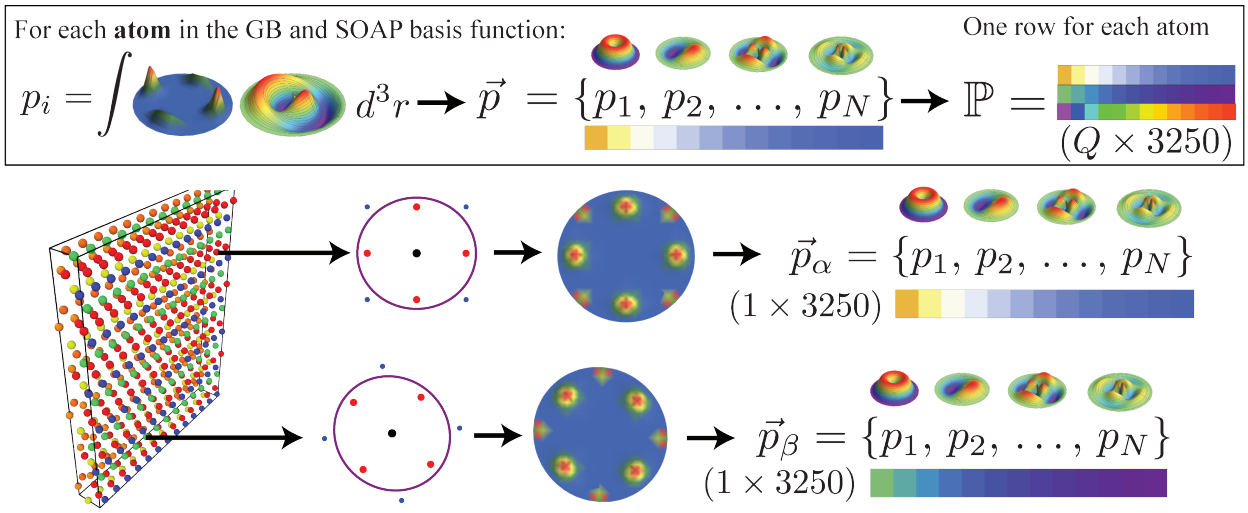

Figure 3.2 Process of creating SOAP vectors. Each atom in a grain boundary is iteratively selected. All atoms within a cutoff radius around a selected atom, excluding the atom itself, become part of the Local Atomic Environment (LAE) of that atom. A Gaussian density function is placed over each atom in the LAE, and then the LAE is projected onto a spectral basis. This is done by multiplying the LAE by one component of the spectral basis and then integrating to get a number. This number becomes one of the coefficients in the SOAP vector, telling us how much of that basis component is in the LAE. This is done for all basis components and all atoms in the grain boundary, resulting in a $Q \times N$ matrix $\mathbb{P}$ whose rows are LAEs and column weights in the spectral basis. Figure used with permission from (Rosenbrock et al. 2017).

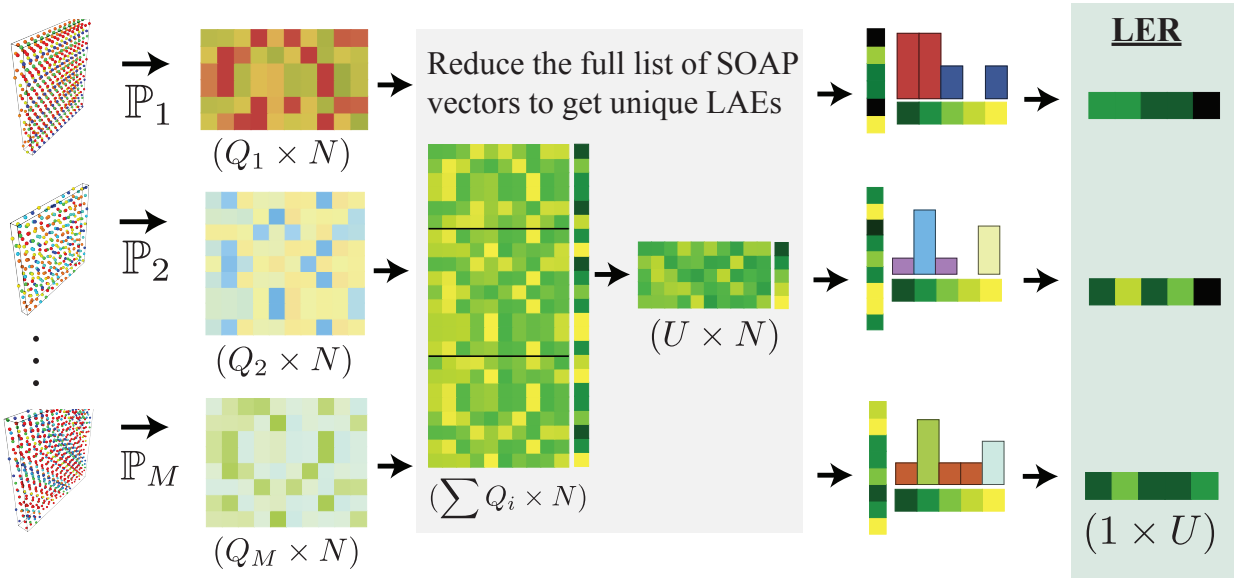

Figure 3.3 Reduction of Local Atomic Environment (LAE) representation to Local Environment Representation (LER). After calculating the SOAP vectors for each atom, we are left with the grain boundaries (GBs) being represented as matrices $\mathbb{P}_M$, where M is the number of the GB. Because the matrices are of different sizes, making their comparison difficult, and because there are too many LAEs acting as features, we make a reduction. We compare all of the LAEs and keep only the unique LAEs. The GBs are now represented as a new vector who's coefficients are percentages of how much of each unique LAE it contains. Figure used with permission from (Rosenbrock et al. 2017) with alterations.

Moreover, in total for the Dudarev EAM potential, there were over 12 million LAE vectors. This means we have over 12 million features to optimize parameters on, which is computationally too expensive to work with. To reduce the dimensionality of the problem (to reduce millions of features to something computationally feasible), we use SOAP's dissimilarity metric s to compare every LAE to every other LAE in the set, keeping only the unique LAEs within a tolerance, ε. The similarity metric can be seen in the equation below as shown. To get this equation, we take the difference between the two vectors we are comparing to get a new vector. We then dot this vector with itself and take the square root to get its magnitude. In this way we are looking at the magnitude of the difference between the vectors.

$$s = ||\vec{a} - \vec{b}||$$

$$s = \sqrt{(\vec{a} - \vec{b}) \cdot (\vec{a} - \vec{b})}$$

$$s = \sqrt{\vec{a} \cdot \vec{a} + \vec{b} \cdot \vec{b} - 2\vec{a} \cdot \vec{b}}$$

Reducing the number of features by keeping only the unique LAEs is known as Local Environment Representation (LER). The GBs are now represented as a new vector, the LER vector, who's coefficients are percentages of how much of each unique LAE it contains. We prefer this final representation over other representations because it lends itself readily to physical interpretation, making GB comparison easy. It also produces efficient features for machine learning.

Chapter 4

Machine Learning on α-Iron Grain Boundaries

To describe conventional supervised machine learning (using features) as simply as possible, it is optimizing a prototype function to follow the trend of the database and provide accurate predictions. This is made possible by allowing built-in or added parameters to vary until they are optimized, yielding, as is the goal, a function that does not over-fit nor under-fit the database's trend. There are many algorithms and tools that have been created to increase the efficiency, accuracy, interpretability, etc. of the final function, yet the basic idea of fitting a prototype function still remains.

As an example, say you want to predict how much it is going to rain in the next week. Over the past month you have been gathering data, keeping a record of the various cloud formations, wind velocities, air pressures, temperatures, and the height of rainfall in millimeters outside your house everyday. In this scenario, the cloud formations, wind velocities, air pressures, and temperatures are your "features," and the rain height is the "label," or the answer to the features. For instance, overcast with an average of 8 mph SE winds, average barometric pressure of 29.6" Hg, and average temperature of 52 °F corresponded on a given day to 6.6 *mm* of rainfall. Next you want to build

your model so that you can put this morning's data in and get a prediction on how much it is going to rain today. You can develop your own starting function, or use an existing one. The simplest prototype function for a scenario like this is a linear regression model.

$$y = ax + b$$

where a and b are the varying parameters, and x is a feature. The number of features can be extended, each given their own variable parameter which will control how much weight that feature has in the final function,

$$y = a_0 + a_1x_1 + a_2x_2 + a_3x_3 + ...$$

or the features can be presented as higher powers of x, in which case the function is no longer linear,

$$y = a_0 + a_1x_1 + a_2x_2^2 + a_3x_3^3 + ...$$

Once a starting function is chosen, you use your dataset to "train" the function (aka the "machine") until the parameters fit the function to the dataset as much as it can or within a chosen tolerance. This often requires another algorithm that will find the minimum distances between relationships of the features and the label. The target function y you end up with in the end is your final model. You can feed it your set of features $\{x_1, x_2, x_3, ...\}$ and get out a prediction $y = 5.2\ mm$.

In the present work, we use gradient boosted decision trees as the prototype function with cross-validation techniques to predict GB energy on the α-Fe database. Decision trees are known for being robust while also readily transparent. A lot of machine learning algorithms have historically acted as black boxes, meaning you put data in and you get results out without knowing what went on in between because it is too complex for a human to follow. Increasingly there has been a development of supplementary algorithms designed to simplify the complexity and make it human-interpretable (for instance, Principle Component Analysis, which projects a high-dimensional feature set into

something you can visualize such as three or two dimensions). For our purposes, we found decision trees, which are naturally interpretable, a good fit.

The final function of a decision tree is a system of paths that branch out not unlike a tree. The paths can be represented pictorially and are readily followed. At each leaf, or node, a condition is presented. The answer to the condition will determine which of the adjoining paths are taken.

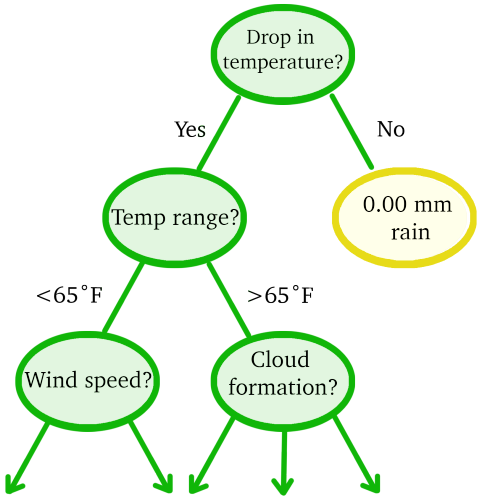

Figure 4.1 Decision tree example for weather prediction.

Gradient boosting increases the efficiency of optimization, and cross-validation serves to utilize as much of the data available as possible for both training and testing. In supervised machine learning using features, you split the dataset up into a training set–part of the data to optimize the function with, and a testing set–the other part of the data to test the accuracy, variances, biases, etc. of the resulting function. This straightforward split has the caveat that you never use the test data to make the function better, and you never use the training data to test the validity of function. When data is already scarce, your function is unlikely to work well. When using cross-validation, you still need to keep a separate test set that your function will never see (otherwise there is no way of knowing how good your final function really is), but we can increase the usefulness of the data in the training set by splitting it further into more training and testing sets. k-fold cross-validation

means you split the training set into k sub-sets. $k-1$ of the sets are used for training, and the left out set is used for testing. Each subset is iterated through to be given a chance to be the test set. In this way, k separate functions are created that can then be averaged to make a final function. For the present work, we use 5-fold cross-validation.

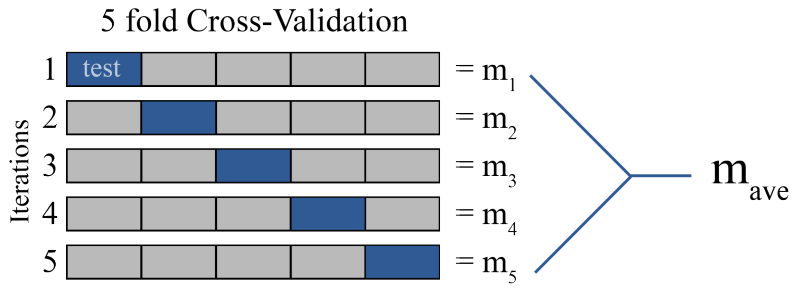

Figure 4.2 Training and testing sets are split five ways to train five separate machines which are then averaged to give the final machine.

Chapter 5

Results

5.1 Predicted Energies

GB energies were predicted on the α-iron, Dudarev EAM dataset. In total there were 907 different GBs as a result of the varied crystal orientations. The machine learning was run for different splits of the training and testing sets. As expected, increasing the training set improved the accuracy of the predictions. This increase, or learning rate, was about a 5–6% going from a 50/50 training-testing split to a 90/10 split, see fig. 5.2. This was done for a cutoff radius of 3.25 Å, $lmax = 12$, $nmax = 12$, $\varepsilon = 0.0025$, $\sigma = 0.5$, where *lmax* and *nmax* determine the angular and radial components of the spectral basis respectively, σ is the Gaussian density parameter, and ε is the unique LAE tolerance.

The actual versus predicted energies for a 50/50 split and a 80/20 split are as show in fig. 5.1. This was for an initial run without adjusting the SOAP parameters, LAE uniqueness tolerance ε, or manually input machine learning parameters. For the 50/50 split, the accuracy, averaged over 100 independent fits, was $57 \pm 3.5\%$, and for the 80/20 split it was $61 \pm 4.3\%$. The fluctuation in error is calculated using standard deviation from the mean. This means that instead of taking the full range of percents over the 100 iterations as our fluctuation range, we account for outlying points (outliers)

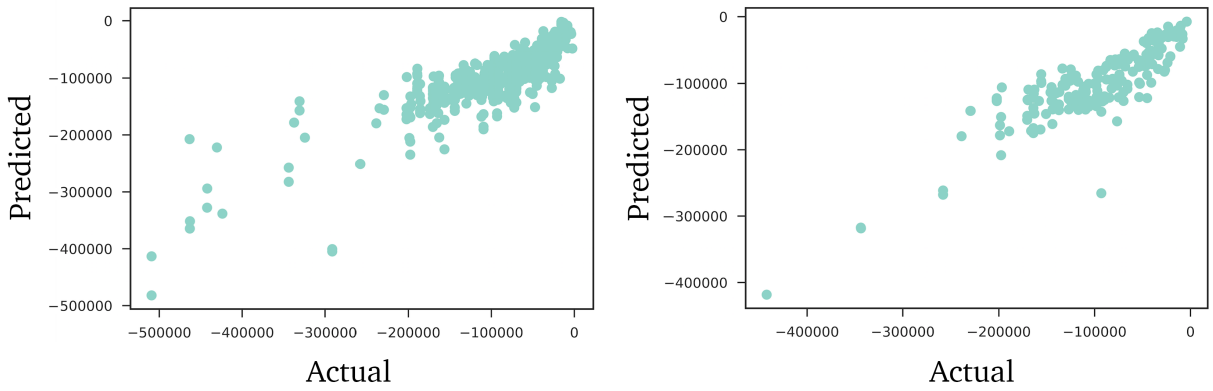

Figure 5.1 Left: Energy prediction results doing a 50/50 pessimistic split of the dataset. Right: Energy prediction results doing an 80/20, training-testing, split of the dataset. This is a more typical split in machine learning. Energies are in Joules.

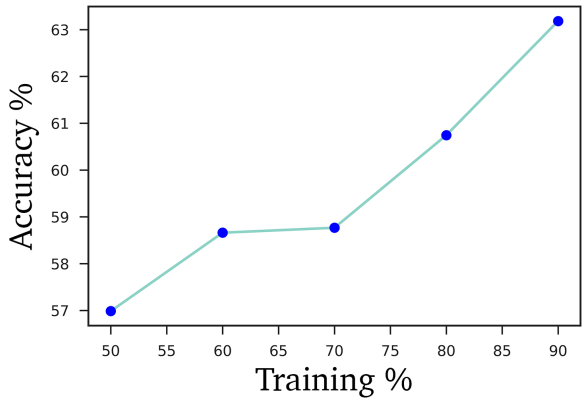

Figure 5.2 Learning rate on the Dudarev EAM potential, α-iron dataset with initial parameters. Learning rate was averaged over 25 independent fits for 50, 60, 70, 80, and 90 percent training set splits.

when calculating the fluctuation range, see fig. 5.3. Outliers can result from a bad random split or other another statistical error. Excessive outliers may also indicate a bug in the code.

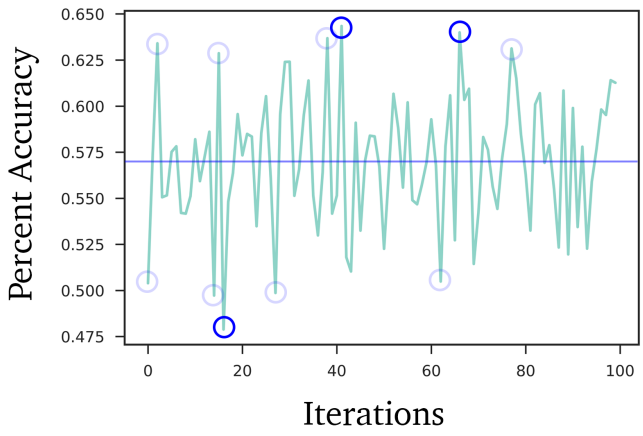

Figure 5.3 Plot of the percent accuracy over 100 iterations of the 50/50 training-testing split. The average percent accuracy is 57%. The total fluctuation has a range of 9% above and 14% below the 57%. This range is a bit skewed because of outliers, probably coming from how the training set is split each time. Standard deviation accounts for these outliers, giving us a range of $\pm 3.5\%$.

5.2 Discussion and Analysis

There is a clear correlation between the actual and predicted energies. In a perfect correlation, all of the points in the actual versus predicted energy graphs would lie exactly on the 45° axis. The results for the initial run are promising. The next step is to adjust the parameters until we get a vetted model, looking for an energy prediction accuracy of 80% or greater on the 50/50 split.

Learning rates for cutoff radii 3.25, 3.6, 4.0, and 4.4 Å were tested on 1/4th of the Dudarev set. 4.0 Å showed the most promise, producing a learning rate of 7–12%. This is a proof of concept that we can work the parameters to get better results.

At the moment calculating the unique LAEs on 1/4th of the dataset for a cutoff radius of 4.0 Å takes 13 hours on a robust computer. This is because the algorithm to compare each LAE goes as

x^N. If the algorithm is rewritten to model advanced sorting algorithms such as quicksort or heap sort, it can be reduced to going as $\log(N)$, cutting a computation time of 13 hours down to as much as an hour or less. Switching from using quippy, a large software package, to soapxx, a much smaller software package, will likely cut down on the computation time and expense as well, since calculating the SOAP vectors on the full dataset can take as much as 5 hours or more depending upon the variables.

Once the computation time is reduced to something more workable, the next step is to run a parameter grid to test the various parameter combinations. A lot of the parameters are interdependent. For instance, once we increase the cutoff radius to 4.0 Å, the angular resolution of the SOAP vectors decreases. The angular resolution of a disk, or sphere, is dependent on both the radius of the disk, and the angle (given in the SOAP parameters by *lmax*). Higher angular resolution allows us to distinguish smaller details. A simple example is the difference between two very close atoms being recognized as one atom, or two. If we have a really long cutoff radius but not high enough of an angle factor, two very close atoms can appear to look like one, or closer than they really are.

Already, the model predicts better than random guessing, which indicates that it has learned some physics. Once the model is optimized, we will look for the LAEs that are most important in the model's decision making process. Similar to when a Ni database was tested by Rosenbrock et al., we expect to find certain LAEs corresponding to high-energy GBs, and other LAEs corresponding to low-energy GBs. In their paper, there were 10 identified most important LAEs for energy prediction, with one LAE carrying as much as 5.5% of the importance in the decision making process on its own.

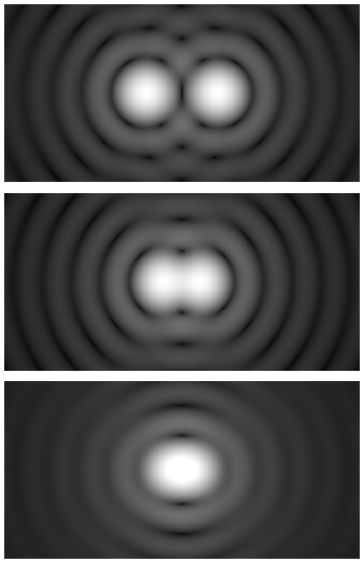

Figure 5.4 Discernment between diffraction patterns of light as angular resolution is decreased, similar to how two atoms might be seen as the SOAP cutoff radius is increased, but *lmax* (the angular dependence) is not. Image by Spencer Bliven.

5.3 Conclusion

The methodology presented in this paper shows promise of being able to predict GB properties, specifically the GB energies of α-Fe. We have good reason to believe a vetted model can be achieved with continuation of the research. Presuming that a satisfactory model is achieved, we will have established significant evidence that the methods employed in this paper are generalizable, since they have previously performed well on a Ni database.

If a satisfactory model is not achievable using these methods, then we will have shown that some significant physics is being overlooked by the methodology. Looking for and accounting for the missing physics in an augmented methodology will help us gain new ground. Or perhaps in trying to vet a new model, we will discover key underlying physics of the GB space that was not previously known.

We hope we are one step closer to solving the HE problem, and one step closer to helping make quick, versatile GB engineering a realizable future.

Bibliography

Bartók, A. P., Kondor, R., & Csányi, G. 2013, Physical Review B - Condensed Matter and Materials Physics, 87, 1

Bartók, A. P., Payne, M. C., Kondor, R., & Csányi, G. 2010, Physical Review Letters, 104, 1

Chen, Y.-S., et al. 2017, Science, 355, 1196

de Assis, K. S., Lage, M. A., Guttemberg, G., dos Santos, F. P., & Mattos, O. R. 2017, Engineering Fracture Mechanics, 176, 116

Djukic, M. B., Sijacki Zeravcic, V., Bakic, G. M., Sedmak, A., & Rajicic, B. 2015, Engineering Failure Analysis, 58, 485

Dunne, D. P., Hejazi, D., Saleh, A. A., Haq, A. J., Calka, A., & Pereloma, E. V. 2016, International Journal of Hydrogen Energy, 41, 12411

Gobbi, G., Colombo, C., Miccoli, S., & Vergani, L. 2014, Procedia Engineering, 74, 460

Janssens, K. G., Olmsted, D., Holm, E. A., Foiles, S. M., Plimpton, S. J., & Derlet, P. M. 2006, Nature Materials, 5, 124

Jemblie, L., Olden, V., & Akselsen, O. M. 2016, International Journal of Hydrogen Energy, 42, 11980

—. 2017, International Journal of Hydrogen Energy, 42, 11980

Lambert, H., Fekete, A., Kermode, J. R., & De Vita, A. 2017

Luppo, M., & Ovejero-Garcia, J. 1991, Corrosion Science, 32, 1125

Pressouyre, G. M. 1980, Acta Metallurgica, 28, 895

Randle, V. 2006, Journal of Microscopy, 222, 69

Rosenbrock, C. W., Homer, E. R., Csányi, G., & Hart, G. L. W. 2017, npj Computational Materials, 3, 29

Sirois, E., & Birnbaum, H. K. 1992, Acta Metallurgica Et Materialia, 40, 1377

Stukowski, A. 2010, Modelling and Simulation in Materials Science and Engineering, 18

Takahashi, Y., Kondo, H., Asano, R., Arai, S., Higuchi, K., Yamamoto, Y., Muto, S., & Tanaka, N. 2016, Materials Science and Engineering A, 661, 211

Tavares, S., Machado, C., Oliveira, I., Martins, T., & Masoumi, M. 2017, Engineering Failure Analysis

Tehranchi, A., & Curtin, W. A. 2017, Journal of the Mechanics and Physics of Solids, 101, 150

Wang, S., Martin, M. L., Robertson, I. M., & Sofronis, P. 2016, Acta Materialia, 107, 279

Watanabe, T. 2011, Journal of Materials Science, 46, 4095

Zhou, X., & Song, J. 2017, Materials Letters, 196, 123

Index