Acta Materialia 188 (2020) 49–62

Contents lists available at ScienceDirect

Acta Materialia

journal homepage: www.elsevier.com/locate/actamat

Full length article

Semi-supervised learning approaches to class assignment in ambiguous microstructures

Courtney Kunselman*[,a], Vahid Attari[a], Levi McClenny[b], Ulisses Braga-Neto[b], Raymundo Arroyave[a,c,d]

[a] Department of Materials Science and Engineering, Texas A&M University, College Station, TX 77843, United States
[b] Department of Electrical Engineering, Texas A&M University, College Station, TX 77843, United States
[c] Department of Mechanical Engineering, Texas A&M University, College Station, TX 77843, United States
[d] Department of Industrial and Systems Engineering, Texas A&M University, College Station, TX 77843, United States

ARTICLE INFO

Article History:
Received 9 November 2019
Accepted 23 January 2020
Available online 31 January 2020

Keywords:
Machine learning
Microstructure classification
Support vector machines
Semi-supervised learning methods
Unsupervised error estimation

ABSTRACT

Uncovering links between processing conditions, microstructure, and properties is a central tenet of materials analysis. It is well known that microstructure determines properties, but expressing these structural features in a universal quantitative fashion has proved to be extremely difficult. Recent efforts have focused on training supervised learning algorithms to place microstructure images into predefined classes, but this approach assumes a level of *a priori* knowledge that may not always be available. In this paper, we expand this idea to the semi-supervised context in which class labels are known with confidence for only a fraction of the microstructures that represent the material system. It is shown that classifiers which perform well on both the high-confidence labeled data and the unlabeled, ambiguous data can be constructed by relying on the labeling consensus of a collection of semi-supervised learning methods. We also demonstrate the use of novel error estimation approaches for unlabeled data to establish robust confidence bounds on the classification performance over the entire microstructure space.

1. Introduction

1.1. Motivation

A basic goal of materials data analysis is to extract useful information from materials datasets that can in turn be used to establish connections along the processing-structure-properties chain. As the volume, variety and complexity of the datasets increases, extracting such information will likely be increasingly reliant on automated frameworks that facilitate the uncovering of distinctive features and patterns that can be used for further analysis in the context of Integrated Computational Materials Engineering (ICME). A central challenge exists in arriving at such a framework in the case of microstructure image data stemming from the fact that variation in a material's internal structure is high and exists in a truly multi-dimensional (feature) space that is often times difficult to navigate without the aid of sophisticated analysis tools [1,2]. The microstructure space is difficult to navigate in part because of the challenges associated with establishing the most important features that can in turn be used to establish differences among microstructures as well as the complex, multi-dimensional and often times non-linear relationships between such features and materials behavior. While expert human annotation is highly effective, problems arise when the microstructure datasets are large. In this work, we present a framework for the semi-supervised learning of the microstructure space that moves towards addressing some of these challenges.

1.2. Background

It is well known that computational methods have been identified as a cost-effective way of solving the inverse mapping problem of properties to structure to processing conditions for materials design. Forging these links requires quantitative analysis, and while processing parameters and property observations are generally easily quantifiable—they tend to be represented as objects that exist in a relatively low dimensional space—(micro)structure, the central link in the ICME chain, presents a much more challenging characterization obstacle.

For well-studied material systems, expert identification of features and the resulting demarcation of microstructure images into predefined classes is a start [3–5]. Unfortunately, these human-assigned labels can be too subjective in the face of large structural diversity, and with recent advances in simulation capabilities [6–10], physics-based models can generate massive microstructure datasets for which human annotation is prohibitively expensive [11]. Automated classification models (e.g. support vector machines and neural networks)

* Corresponding author.
E-mail address: cjkunselman18@tamu.edu (C. Kunselman).

https://doi.org/10.1016/j.actamat.2020.01.046
1359-6454/

address both of these concerns by removing the human decision-maker after initial training, making them popular microstructure analysis tools [4,11–17].

These classification algorithms are in the category of supervised learning methods. Models built from these approaches are inductive predictors, implying that they are mappings from inputs to outputs, including inputs which the model has not yet seen. This is in contrast to unsupervised learning methods which are transductive algorithms used to infer relationships between data points within a given set for which no outputs are known. These relational inferences are commonly used for tasks such as dimensional reduction and clustering, which provide insight into feature redundancy and data structure, respectively. Because they do not have an output to train over (or validate with), these methods generally make much stronger assumptions about data distributions than their supervised counterparts. We note, however, that when these assumptions align well with the problem at hand, insights from unsupervised methods can be helpful in the construction of higher-performing supervised models [18].

Just as with many other automated tools, a supervised classifier's performance is limited by the assumptions it 'is told' to make and the data on which it is trained [19]. In general, a successful classifier requires (I) a robust labeling of the training set, (II) a discriminative feature set, and (III) an appropriate choice of model assumptions and hyperparameters. Due to the rigorous theoretical framework of most mainstream supervised classification algorithms, meeting the third requirement is usually a straightforward exercise when the first and second requirements are readily available. Unfortunately, for a given microstructure classification problem, attaining the first two requirements is anything but straight-forward because their acquisition usually requires an appreciable amount of *a priori* information. Consider the requirement of a robustly labeled training set, recalling that a class of microstructures is most helpful to the design problem if its members share a set of structural features which map to a tight region in some property space of interest. For material systems in which the relationship between (micro)structural features and resulting properties is not well-studied or for a more general problem where multiple material systems are involved, identifying the correct number of classes and confidently assigning a discrete label to each microstructure image becomes a highly subjective, nontrivial task. Consequently, expense and/or uncertainty can lead to only a fraction of the available data being labeled, further deteriorating the robustness of the training set.

Fortunately, semi-supervised learning methods have been developed specifically to address the challenges associated with partially labeled training data. While examples of these methods are sparse in the available literature investigating microstructure classification—Okaro et al. did make use of them to detect faults in the microstructures of additively manufactured parts [20]—semi-supervised learning has been used successfully in a variety of other image classification studies [21–24]. Just as with the supervised case, traditional semi-supervised methods can still only assign data points to predefined classes, which can be a concern if the unlabeled dataset contains data points from classes beyond those that are known *a priori*. To address this issue, approaches which allow for new class discovery have been explored in contexts outside materials science [25–27], with a large portion of the work focusing on identifying new cancer classes, for example, through patterns in gene expression.

Similar to the unsupervised learning problem, most semi-supervised methods operate under the assumption that data points which are close to each other in the feature space (based on a given distance metric) have a high chance of belonging to the same class. Supervised classifiers do not necessarily make this assumption, but as the second requirement above implies, it is to be expected that in general the larger the degree of separation between classes in the feature space, the more successful the classifier will be. However, finding such a feature set which is simultaneously adequately discriminative and computationally tractable can be extremely difficult, and even if it is found for one material system, there is no guarantee that it will generalize with success. In response to this problem, DeCost and Holm [28] explored a classification framework which applies the 'bag of visual features' methodology [29] to build a discriminative feature set which requires no *a priori* knowledge of the relevant structural features present in the dataset. DeCost and Holm demonstrated the utility of this method of characterization with an 83% classification accuracy on microstructure images from seven different material systems, and subsequent studies have utilized the bag of visual features approach for their own classification models [30,31].

While DeCost and Holm have presented a method for handling a lack of *a priori* information for the requirement of a discriminative feature set, as far as we know corresponding work has yet to be done for the situation of human annotation by visual inspection leading to uncertainty in class taxonomy and initial label assignment for the microstructure classification problem. The following notional example illustrates why this problem is worth considering.

1.3. Notional example

Similar to many problems involving the exploration of microstructure spaces [7], color is a continuous physical feature which is often discretized for a given application. Suppose we are tasked with building a color classification model using colors (a–g) shown in Fig. 1. We are given no information about how many classes there should be or what they should contain. Upon visual inspection of our sample set, we decide to keep it simple and define two widely-recognizable classes: purple and blue. Furthermore, we are confident that colors (a) and (e) belong to the purple class and that colors (c) and (d) belong to the blue class. However, the remaining colors could really belong to either class, so we leave them unlabeled to start. We find a feature space which clusters our confidently labeled colors well, and following our intuition, colors (b), (f), and (g) lie in a space between the two clusters. From here, we are unsure of how to provide labels to the remaining three colors.

One option is to claim that classification is inherently subjective and label them according to our expert opinion. Another is to leave them unlabeled, train a classifier on the high-confidence colors, and then use this model to label the ambiguous colors. However, the first approach has the potential to provide the model with bad training information while the second may not provide enough information. In response to this predicament, we could define a third class, say indigo, which is somewhere between purple and blue. But now our dilemma is doubled because we have to decide between purple and indigo on one front and blue and indigo on another, and our confidence level on previously labeled colors could drop now that a new class is available.

The above example illustrates that when class taxonomy is uncertain, there can exist a tension between providing enough labeled data for the training set while avoiding the addition of detrimental data or information that may compromise the performance of the model. The remainder of this paper proposes a framework to address this tension in a data-driven manner which involves appending the high-confidence training set with a subset of the ambiguous data identified through the application of a collection of semi-supervised learning methods. The idea is that if a 'safe' subset of the ambiguous data can be labeled and added to the training set, the supervised model trained over this set will gain valuable information at little risk of degrading performance.

2. Dataset creation

A multi-scale elasto-chemical phase-field approach based on Cahn–Hilliard (C–H) formalism [7,32] is used to generate synthetic microstructure dataset used in this study. The total free energy

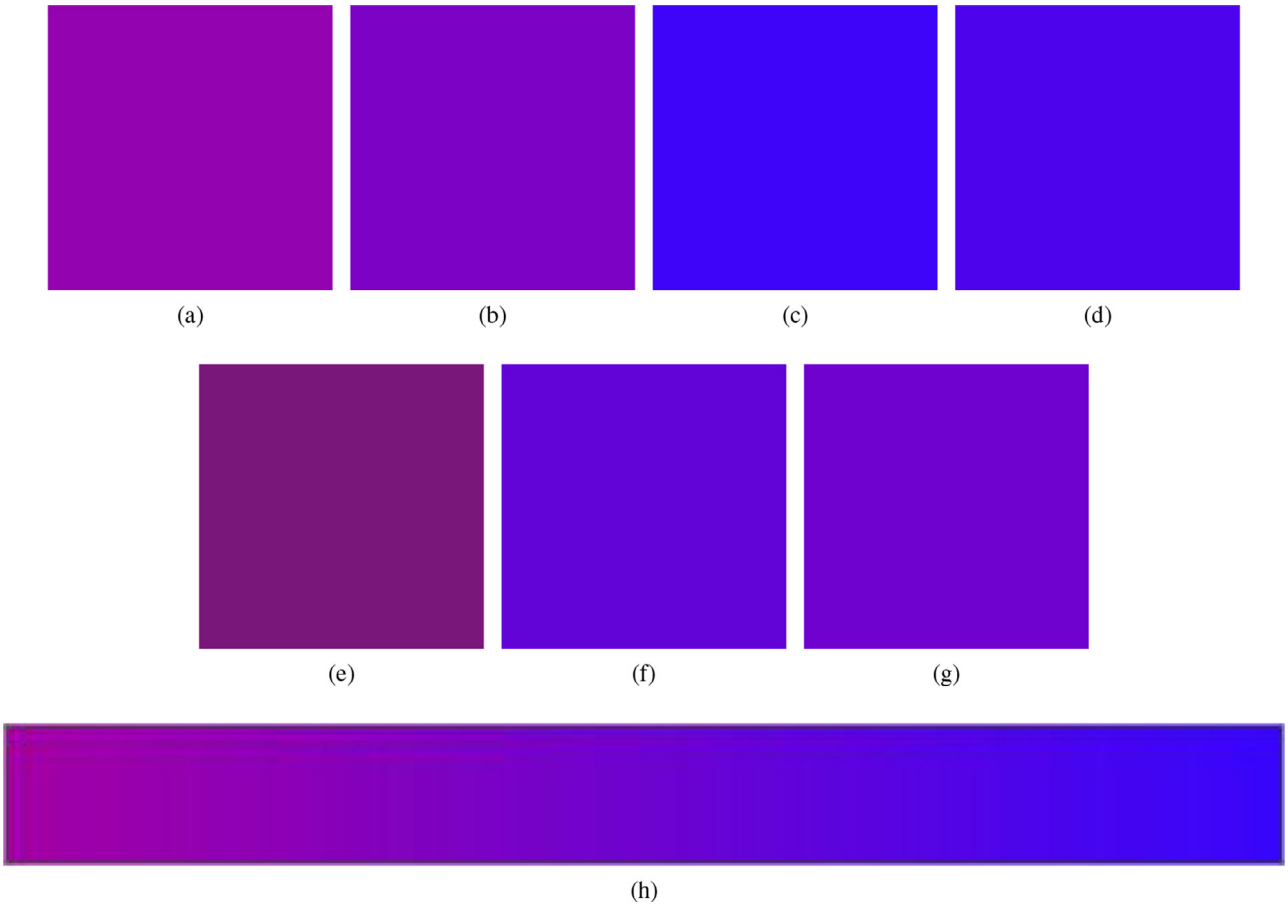


Fig. 1. Sample set of colors (a–g) from the spectrum in (h). For interpretation of the colors, refer to the online version of this document.

functional ($\mathscr{F}^{tot}$) for a heterogeneous solid medium as the sum of contributing fields over the domain (Ω) is:

$$\mathscr{F}^{tot} = \int_{\Omega} \left(f_{bulk} + f_{interfacial} + f_{elastic}\right) d\Omega \tag{1}$$

where bulk free energy, f_{bulk}, interfacial free energy, $f_{interfacial}$, and elastic strain energy, $f_{elastic}$ are:

$$f_{bulk} = f^0(c, T), \tag{2}$$

$$f_{interfacial} = \frac{1}{2}\kappa |\nabla c|^2, \tag{3}$$

$$f_{elastic} = \frac{1}{2}\sigma_{ij}\varepsilon_{ij}^{el} \tag{4}$$

where c is the composition field, $f^0(c, T)$ is the free energy of a unit volume of homogeneous material for a given temperature (T), κ is the gradient energy coefficient, ε_{ij}^{el} and σ_{ij} are the local elastic strain and stress in the material, respectively.

We postulate the following form of the (C-H) kinetic equation (Eq. 5) along with the micro-elasticity equations (Eqs. (6)–(8)) to generate the synthetic microstructure space by tracking the evolution of composition field. We start with a uniform state where the composition is randomly perturbed only ± 2% around alloy composition, and let the system evolve based on the given input material parameters:

$$\frac{\partial c}{\partial t} = \nabla \cdot M \nabla \left(\frac{\delta \mathscr{F}^{tot}}{\delta c}\right), \tag{5}$$

$$\begin{cases} \dfrac{\partial}{\partial r_j}\left\{C_{ijkl}\left(E_{kl} + \varepsilon_{kl}^{\bigstar} - \varepsilon_{kl}^{0}\right)\right\} = 0 & \text{on} \quad \Omega \\ \varepsilon_{kl}^{\bigstar} & \text{periodic on} \quad \Omega \end{cases}, \tag{6}$$

$$\varepsilon_{ij}^{el} = \varepsilon_{ij}^{tot} - \varepsilon_{ij}^{0}, \tag{7}$$

$$\varepsilon_{ij}^{tot} = E_{ij} - \varepsilon_{ij}^{*} = E_{ij} - \frac{1}{2}\left(\frac{\partial u_i^{\bigstar}}{\partial r_j} - \frac{\partial u_j^{\bigstar}}{\partial r_i}\right) \tag{8}$$

where M, C_{ijkl}, E_{ij}, $\varepsilon_{ij}^{\bigstar}$, ε_{ij}^{0} are mobility, elastic constant tensor, mean of total strain (ε_{ij}^{tot}), periodic strain, and stress-free transformation strain, respectively. ε_{ij}^{el} is elastic strain, and ε_{ij}^{*} is the periodic fluctuation strain field given by period displacement (u^*). The eigenstrain term is interpolated over the domain by $\varepsilon_{ij}^{0} = \varepsilon^T \delta_{ij} h(c)$, where ε^T is the strength of the mismatch, δ_{ij} is the Kronecker-delta function and $h(c) = c^3(10 - 15c + 6c^2)$ is an standard interpolation function. A detailed description of the microelasticity model is provided in detail in [7]. An efficient method is used to sample input parameters out of prior probably distributions of input parameters to minimize the number of samples and these samples are fed to the phase-field model to generate the microstructure dataset. For a complete explanation of the method for propagation of the uncertainty in microstructure space and generation of the microstructure dataset refer to [8]. The microstructure dataset is curated in Open Phase-field Microstructure Database (OPMD) website [33].

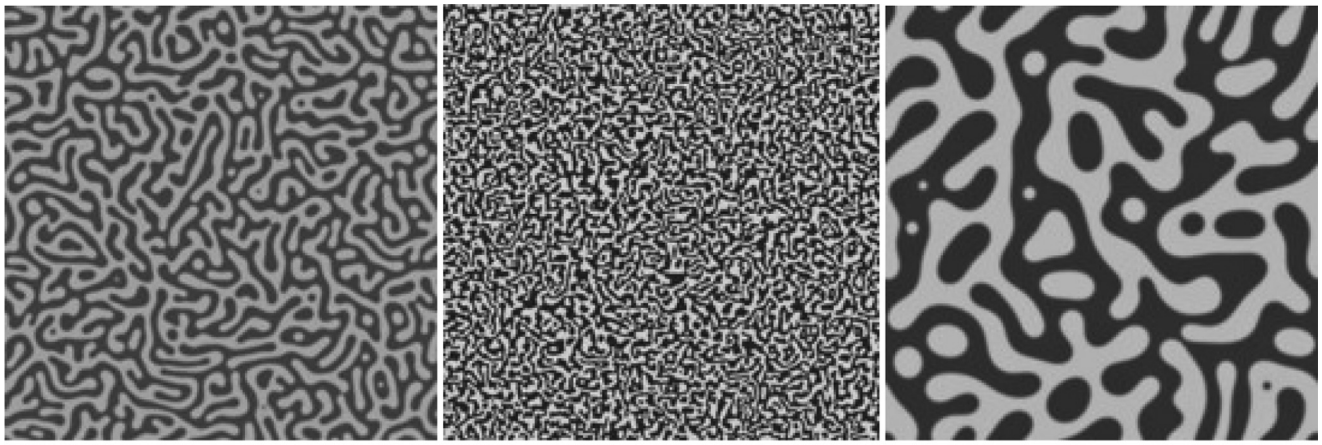
3. Microstructure characterization

3.1. Data labeling and pre-processing

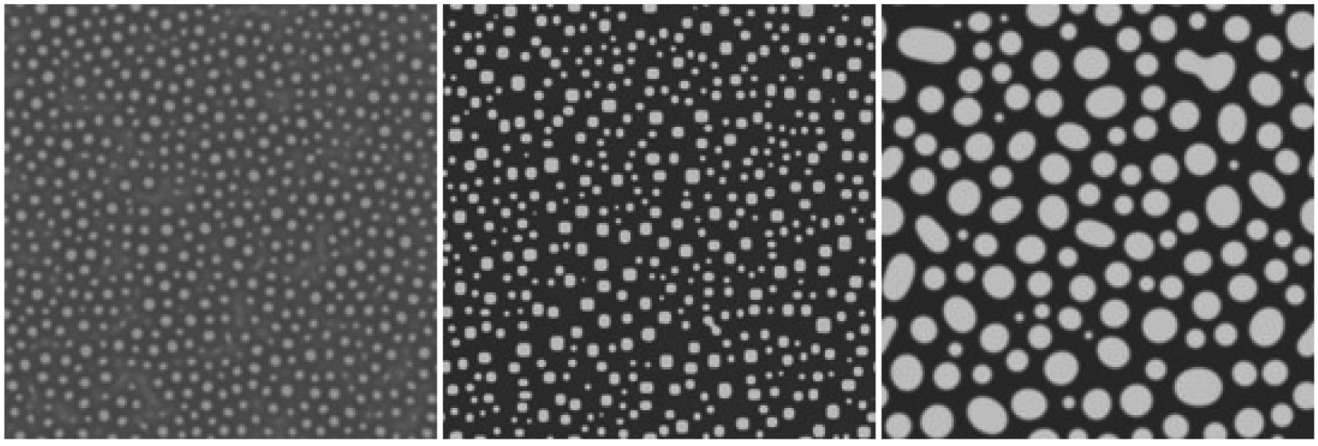
Phase-field simulations produced ten thousand 512 × 512 microstructure images for characterization and classification [8]. Of these ten thousand images, 2,439 were determined to have undergone phase

decomposition (i.e. they self-organized into two phases). Through visual inspection, 1,920 of the two-phase microstructures were labeled as either 'Bicontinuous' or 'Precipitate.' The remaining 519 images resembled a weighted blend of these two classes; thus, they were initially left unlabeled since class assignment could not be made with confidence. Fig. 2 provides examples of this labeling scheme.

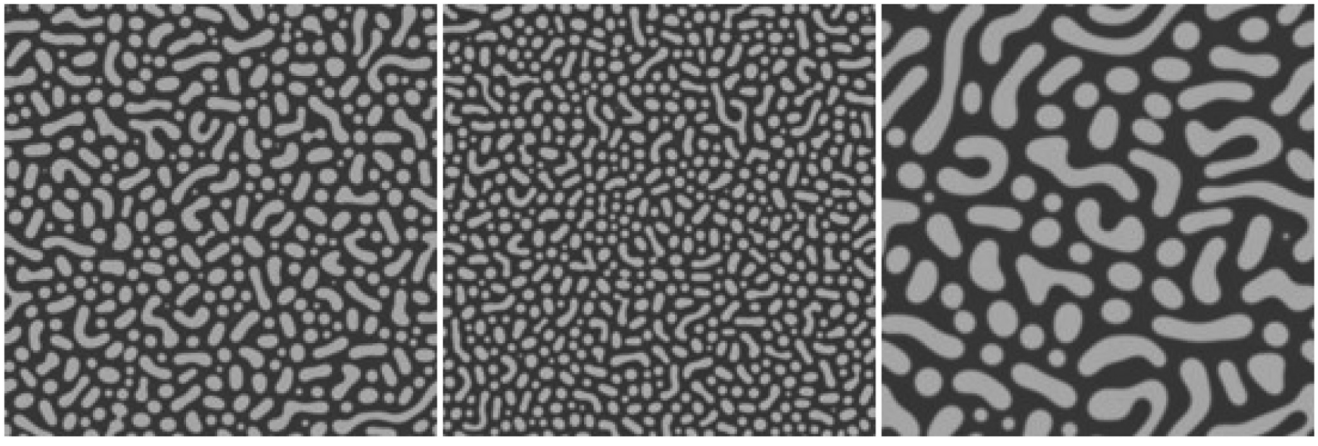
Following label assignment, the two-phase dataset was reduced to binary images using Otsu Thresholding [34]. This popular image segmentation technique iterates through all possible threshold values and chooses the value which minimizes the sum of intra-class variance of pixels above and below the threshold. Final processing consisted of applying opening, a mathematical morphology operator used to eliminate small foreground islands in binary images, to those reduced images requiring noise removal [35]. Fig. 3 illustrates this process. Image processing methods were implemented using the scikit-image package in python [36].



(a) Bicontinuous



(b) Precipitate



(c) Unlabeled

Fig. 2. Examples of initially labeled and unlabeled microstructures.

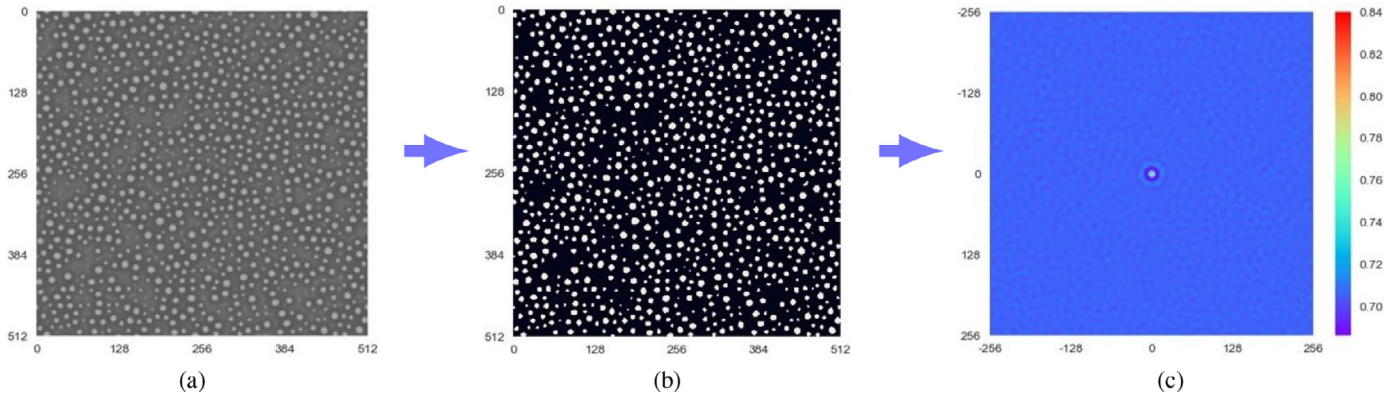


Fig. 3. Microstructure characterization process flow. The raw image (a) is binarized using the Otsu method and noise is removed through opening (b). The image is then ready to be characterized using the black autocorrelation function (c).

3.2. Two-point correlation function

Statistical characterization of microstructures is pervasive throughout recent literature investigating classification, reconstruction, and structure-property linkages [12,30,37–39]. Statistical functions provide probabilistic spatial correlation information and have interpretations based in the random placement of a given polyhedron. Specifically, the two-point correlation function $P_{l,l'}(\boldsymbol{r})$ can be interpreted as the conditional probability of finding local state l at the first endpoint and local state l' at the second endpoint of the vector $\boldsymbol{r}$ after it is randomly placed into the microstructure where local state is a set of structural characteristics needed to distinguish one possible material state from another at the length scale of interest. In [40], Niezgoda et al. give a mathematically rigorous explanation which starts by defining the microstructure function $m(\boldsymbol{x}, h)$ as a wide sense stationary stochastic process in which h is a random variable associated with finding a specific local state at spatial position $\boldsymbol{x}$. The two-point correlation function is then expressed as

$$P_{l,l'}(\boldsymbol{x}_1,\boldsymbol{x}_2) = E[m(\boldsymbol{x}_1,l)m(\boldsymbol{x}_2,l')], \tag{9}$$

which can be reduced to only a function of the spatial variable $\boldsymbol{r} = \boldsymbol{x}_2 - \boldsymbol{x}_1$ since $m(\boldsymbol{x}, h)$ is assumed to be stationary:

$$P_{l,l'}(\boldsymbol{r}) = E[m(\boldsymbol{x},l)m(\boldsymbol{x}+\boldsymbol{r},l')]. \tag{10}$$

When local state information is limited to realizations over a uniform grid in spatial position, the two-point correlation function is given by

$$P_{l,l'}(\boldsymbol{r}) = \frac{1}{S}\sum_{\boldsymbol{s}} M_{\boldsymbol{s}}^{l} M_{\boldsymbol{s}+\boldsymbol{r}}^{l'} \tag{11}$$

where S is the total number of grid points, $\boldsymbol{s}$ is a position in the grid, and $M_{\boldsymbol{s}}^{l}$ is an indicator function which equals one when local state l is at spatial position $\boldsymbol{s}$ [41]. For the case of a binary microstructure with white (w) and black (b) phases, four two-point correlations are defined for a given vector $\boldsymbol{r}$: $P_{w,w}(\boldsymbol{r})$, $P_{b,b}(\boldsymbol{r})$, $P_{w,b}(\boldsymbol{r})$, and $P_{b,w}(\boldsymbol{r})$. However, as the following system of equations demonstrates, only one of these correlations is independent [42]:

$$P_{w,w}(\boldsymbol{r}) + P_{b,b}(\boldsymbol{r}) + P_{w,b}(\boldsymbol{r}) + P_{b,w}(\boldsymbol{r}) = 1, \tag{12}$$

$$P_{w,b}(\boldsymbol{r}) = P_{b,w}(\boldsymbol{r}), \tag{13}$$

$$P_{w,w}(\boldsymbol{r}) + P_{w,b}(\boldsymbol{r}) = \phi_w, \tag{14}$$

$$P_{b,b}(\boldsymbol{r}) + P_{b,w}(\boldsymbol{r}) = \phi_b = 1 - \phi_w \tag{15}$$

where ϕ_i is the volume fraction of phase i. Thus, for this study, only $P_{b,b}(\boldsymbol{r})$ (also known as the black phase autocorrelation) was considered. These autocorrelations were computed using the PyMKS framework, where a primitive basis and periodic boundaries were assumed [43]. Fig. 3 provides a flow from raw input image to black phase autocorrelation. Note that the axes for the autocorrelation image define the vector $\boldsymbol{r}$ which is being placed into the microstructure and not a spatial position in the microstructure.

3.3. Normalization of the two-point statistics and dimension reduction

The black phase autocorrelations calculated for the binary microstructures lie in a 512×512 dimensional space, making the data an impractical input for a classifier due to computational efficiency, classification accuracy, and data visualization concerns. Principal Component Analysis (PCA) is a popular unsupervised dimension reduction tool in machine learning which builds an orthonormal basis corresponding to directions of most variance in the input data. These basis vectors, known as Principal Components, are normalized linear combinations of the original features, allowing for the reduced representation to be easily inverted. Principal Components are determined through an eigenvalue decomposition of the covariance matrix C:

$$C = W\Lambda W^{-1} \tag{16}$$

where W is a matrix of eigenvectors of C and Λ a diagonal matrix of corresponding eigenvalues. These eigenvectors are the Principal Components. Thus, reduction to dimension k is accomplished through multiplying the feature matrix by the first k columns of W [44]. However, direct computation of this decomposition is often not practical for large datasets due to computational expense and finite memory constraints. In response to this dilemma, the machine learning community developed a class of methods known as Incremental PCA (IPCA). These methods either incrementally build the eigenvectors without constructing the covariance matrix or estimate eigenvalue decompositions by incorporating the data samples in batches [45]. Following the latter strategy, Ross et al. [46] developed an IPCA algorithm which is computationally efficient when the number of features is much greater than the number of observations. An additional advantage of this method and those that are similar is that, since these algorithms process the training inputs in batches, additions to the training set can be incorporated into the model without having to reprocess the old data.

PCA has been demonstrated to be an effective dimension reduction technique for two-point correlation data [4,30]. However, when the volume fraction of the microstructure sample has high variability, the first Principal Component, which has been shown to be highly related to volume fraction, can have an extremely large eigenvalue relative to the other eigenvectors [40]. If this is the case, then all of the data points in the high-dimensional feature space could be arranged around a line closely related to volume fraction. This implies that the first few eigenvectors of the decomposition provide very little discriminative (micro)structural information beyond

volume fraction while apparently capturing a large fraction of the variance of the original data. While volume fraction can be an important discriminative feature for many material systems, for this study we aim to craft a classifier which is sensitive to higher order (micro) structural features but robust to varying volume fraction. To address this concern, we introduce the correlation function

$$\mathrm{corr}_{l,l'}(\boldsymbol{r}) = \frac{E[m(\boldsymbol{x},l)m(\boldsymbol{x}+\boldsymbol{r},l')] - E[m(\boldsymbol{x},l)]E[m(\boldsymbol{x}+\boldsymbol{r},l')]}{\sqrt{\mathrm{Var}[m(\boldsymbol{x},l)]}\sqrt{\mathrm{Var}[m(\boldsymbol{x}+\boldsymbol{r},l')]}}. \tag{17}$$

By combining Eqs. (10) and (17) and exploiting the stationarity of $m(\boldsymbol{x}, h)$, $\mathrm{corr}_{l,l'}(\boldsymbol{r})$ can be expressed as a function of $P_{l,l'}(\boldsymbol{r})$ and the volume fractions of local states l, l'

$$\mathrm{corr}_{l,l'}(\boldsymbol{r}) = \frac{P_{l,l'}(\boldsymbol{r}) - \phi_l\phi_{l'}}{\sqrt{(\phi_l - \phi_l^2)(\phi_{l'} - \phi_{l'}^2)}}. \tag{18}$$

Normalizing $P_{l,l'}(\boldsymbol{r})$ in this fashion removes the strong relationship with volume fraction, which allows a PCA decomposition of $\mathrm{corr}_{l,l'}(\boldsymbol{r})$ to be used as a discriminative feature space based on structural information which is robust to varying volume fraction.

Thus, for this investigation, the black phase autocorrelations calculated using PyMKS [43] were normalized using Eq. (18). Those normalized correlations corresponding to labeled microstructures were split into a training set of 1,536 and a test set of 384 data points. The training set was combined with the ambiguous set, and the IPCA method of Ross et al. was then applied to this combined set using the implementation developed by scikit-learn [47]. All normalized correlations were then projected into the subspace defined by the first fifty Principal Components, which cumulatively explained about 60% of the variance (see Table 1). Following projection into the PCA subspace, the data was mean-centered at zero and scaled to unit variance in preparation for classification.

For purposes of comparison, while the PCA decomposition of $P_{b,b}(\boldsymbol{r})$ was not used for classification, the method described above was also followed for the black phase autocorrelation data and the results are presented in Table 1. As expected, the first Principal Component, which we know to be highly related to volume fraction, explains a very large fraction of the variance while the contributions from the succeeding eigenvectors are negligible. Pearson correlation coefficients were also calculated for black phase volume fraction with the first Principal Component from the decomposition of each of the correlation functions. As expected, the Pearson correlation coefficient between black phase volume fraction and the first Principal Component of $P_{b,b}(\boldsymbol{r})$ is quite high at 0.9952, whereas the coefficient for black phase volume fraction and the first Principal Component of $\mathrm{corr}_{b,b}(\boldsymbol{r})$ demonstrates very little correlation at a value of –0.1901. Fig. 4 gives a graphic representation of this result.

4. Classification of labeled data

As articulated in the introduction, the aim of this investigation is to develop a data-driven approach of assigning labels to a select subset of the unlabeled, ambiguous microstructures in order to train the decision-making mechanism of a supervised classifier on a more comprehensive representation of the data. However, any addition to the training set that deteriorates the classifier's performance on the confidently-labeled data must be avoided. To this end, it is necessary to construct a classifier only on the labeled microstructures for the sake of providing a performance baseline. Sundararaghavan and Zabaras [16,17] and later Niezgoda et al. [4] successfully employed microstructure classification schemes involving characterization via two-point statistics, dimension reduction using PCA, and class assignment through support vector machines (SVM). In simple terms, a SVM is a binary classification method which constructs an optimal separating hyperplane in the feature space by maximizing the distance between the hyperplane and the nearest data points in the training set. A more detailed explanation is given below.

Table 1
Explained variance of the first five Principal Components (PC) for decompositions of $P_{b,b}(\boldsymbol{r})$ and $\mathrm{corr}_{b,b}(\boldsymbol{r})$.

PC number	$P_{b,b}(\boldsymbol{r})$	$\mathrm{corr}_{b,b}(\boldsymbol{r})$
1	0.99801	0.19101
2	0.00037	0.08322
3	0.00017	0.03055
4	0.00006	0.02508
5	0.00004	0.02041

4.1. Support vector machines

Suppose we are given the labeled feature set $(\boldsymbol{x}_1, y_1)$, $(\boldsymbol{x}_2, y_2)$, ⋯, $(\boldsymbol{x}_m, y_m)$ where $\boldsymbol{x}_i \in \mathbb{R}^n$ and $y_i \in \{-1, 1\}$ for $i = 1, 2, \cdots, \mathrm{m}$. The goal is to find a decision function

$$D_{\boldsymbol{w},b}(\boldsymbol{x}) = \mathrm{sgn}(\langle \boldsymbol{w}, \boldsymbol{x} \rangle + b) \tag{19}$$

such that

$$D_{\boldsymbol{w},b}(\boldsymbol{x}_i) = y_i \tag{20}$$

for $i = 1, 2, \cdots, \mathrm{m}$ and where $\langle \cdot\,, \cdot \rangle$ denotes a dot product in the feature space. The hyperplane itself is defined where $D_{\boldsymbol{w},b}(\boldsymbol{x}) = 0$. To determine $\boldsymbol{w}$ and b, the objective

$$J(\boldsymbol{w}) = \frac{1}{2}||\boldsymbol{w}||^2 \tag{21}$$

is minimized such that

$$y_i(\langle \boldsymbol{w}, \boldsymbol{x}_i \rangle + b) \geq 1 \tag{22}$$

are satisfied for $i = 1, 2, \cdots, \mathrm{m}$. Introduction of the Lagrangian leads to the conclusion that the solution vector $\boldsymbol{w}$ is of the form

$$\boldsymbol{w} = \sum_{i=1}^{m} \alpha_i y_i \boldsymbol{x}_i \tag{23}$$

where $\alpha_i \geq 0$. Since these α_i's are Lagrange multipliers, they are only nonzero when the corresponding constraints of Eq. (22) are active. Feature vectors with $\alpha_i > 0$ define the distance between classes, and they are known as support vectors.

At this point, we have assumed that all of the constraints in Eq. (22) can be satisfied. However, this is not always the case, especially when outliers are present. This concern is alleviated through the addition slack variables $\xi_i \geq 0$ and an adjustment of the constraints in Eq. (22)

$$y_i(\langle \boldsymbol{w}, \boldsymbol{x}_i \rangle + b) \geq 1 - \xi_i \tag{24}$$

for $i = 1, 2, \cdots, \mathrm{m}$. Inspection of Eq. (24) reveals that these constraints will always be met if the slack variables are allowed to be arbitrarily large. To address this issue, we add them to the objective in Eq. (21)

$$J(\boldsymbol{w}, \boldsymbol{\xi}) = \frac{1}{2}||\boldsymbol{w}||^2 + C\sum_{i=1}^{m} \xi_i \tag{25}$$

where $C > 0$ is a predetermined parameter. This is known as a soft margin SVM. Since we are minimizing J, only those data points in the training set which violate Eq. (22) will have nonzero slack variables. Thus, the larger C gets, the stricter the boundary becomes.

The above discussion only covers the case of a linear boundary. However, positive definite kernels can be used to transform the training set into a higher dimensional space in order to provide a more general decision boundary. This is simply accomplished by replacing instances of $\boldsymbol{x}_i$ in the above equations with $\Phi(\boldsymbol{x}_i)$, an appropriate mapping into a higher-dimensional space. This leads to solution

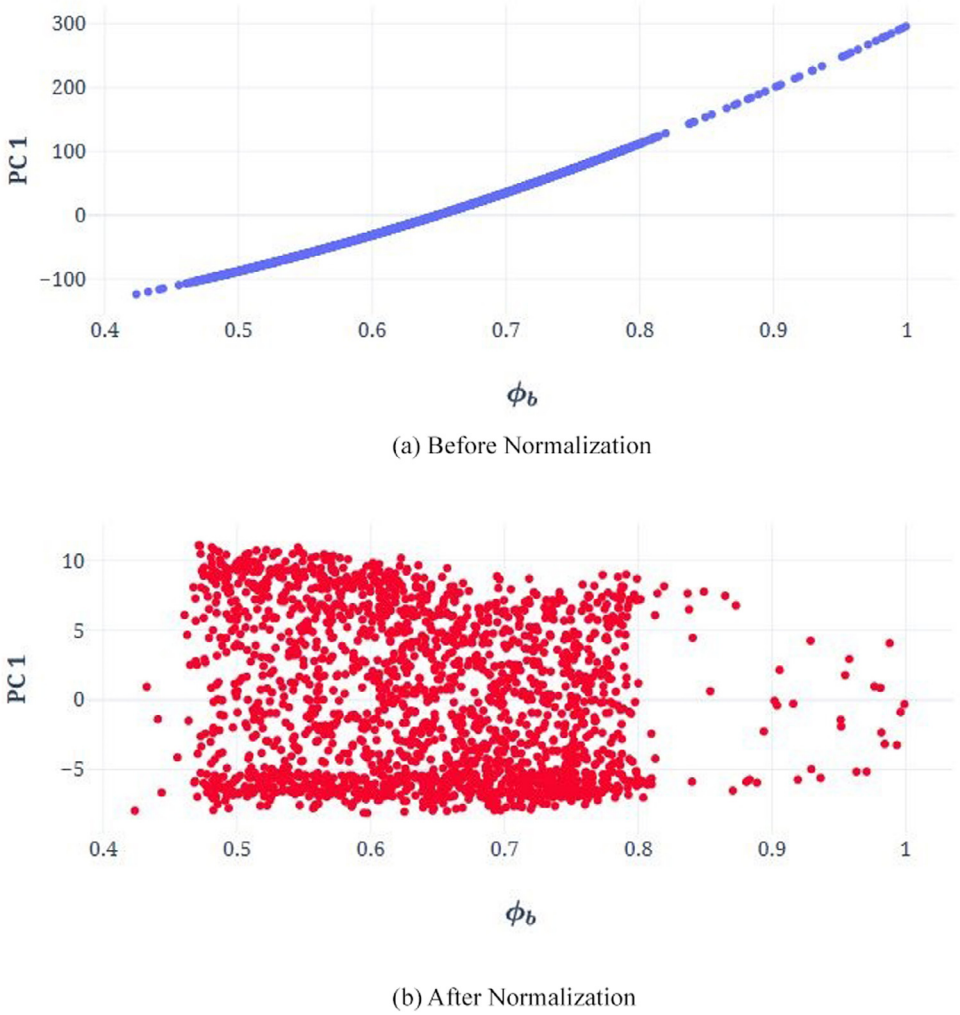


Fig. 4. Principal Component 1 (PC 1) versus black phase volume fraction (ϕ_b) for (a) the PCA decomposition of $P_{b,b}(\boldsymbol{r})$ and (b) the PCA decomposition of $\text{corr}_{b,b}(\boldsymbol{r})$.

vectors of the form

$$\boldsymbol{w} = \sum_{i=1}^{m} \alpha_i y_i \Phi(\boldsymbol{x}_i) \tag{26}$$

with $\alpha_i > 0$ once again indicating that $\boldsymbol{x}_i$ is a support vector. Common kernels include polynomial

$$\langle \Phi(\boldsymbol{x}), \Phi(\boldsymbol{x}_i) \rangle = \langle \boldsymbol{x}, \boldsymbol{x}_i \rangle^d \tag{27}$$

where d is the degree of the polynomial and Gaussian

$$\langle \Phi(\boldsymbol{x}), \Phi(\boldsymbol{x}_i) \rangle = \exp(-\gamma \| \boldsymbol{x} - \boldsymbol{x}_i \|^2) \tag{28}$$

where $\gamma > 0$. Further details can be found in [48].

4.2. Classification of labeled data using support vector machines

For this study, soft margin SVMs employing linear and Gaussian kernels were considered, and the python package scikit-learn was used to implement them [47]. The penalty hyperparameter C and Gaussian kernel hyperparameter γ were optimized through an exhaustive grid search using five-fold cross validation on the specified training set. For purposes of classification performance comparison, hyperparameters for a baseline SVM were optimized on the initially labeled training set, resulting in $C = 10$ and $\gamma = 0.01$. The ensuing classifier had a training error estimate of 0.0358, and application of the labeled test set produced an unbiased error estimate of 0.0547. These error estimates show that the constructed baseline classifier discriminates well between high-confidence data points.

5. Assignment of labels to a subset of the unlabeled data

While the results of the previous section provide an excellent performance diagnostic, it must be remembered that the baseline SVM only demonstrated an ability to correctly classify high-confidence data points. The decision boundary is still completely uninformed by the initially unlabeled microstructures, many of which could be in close proximity to the boundary. To explore the distributions of both ambiguous and high-confidence data points relative to the decision boundary, the decision function given in Eq. (19) was calculated for each data point in the labeled training and ambiguous sets. The results are displayed in Fig. 5. As expected from the low training error estimate, the densities for the high-confidence 'Precipitate' and 'Bicontinuous' microstructures are well separated. Of further interest is that few of the high-confidence data points have values of the decision function at values close to zero whereas the distribution for the ambiguous set appears to be centered close to zero. This means that many of the ambiguous data points are close to the decision boundary relative to the high-confidence points, implying that the decision boundary is not informed by much of the data closest to it. Therefore, in its current state, the baseline SVM is not necessarily a reliable tool for classifying the initially unlabeled microstructures.

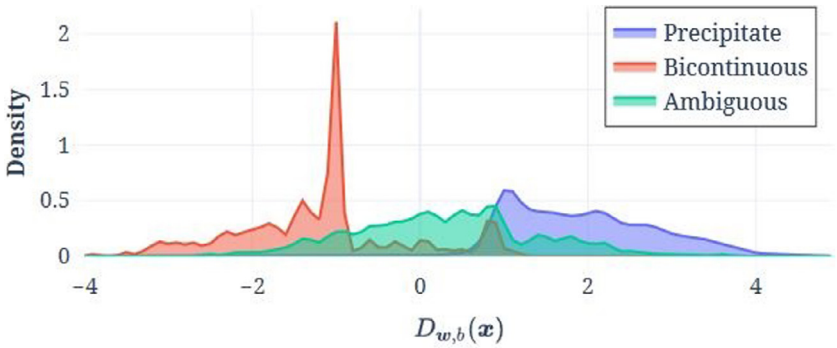


Fig. 5. Distributions of the decision function $D_{w,b}(x)$ of the baseline SVM for the labeled training and ambiguous sets.

To address this concern, a collection of semi-supervised methods involving the labeled training set were used to provide labels for the initially unlabeled data. In contrast to traditional classification which is a completely supervised exercise, semi-supervised classification makes use of additional unlabeled training data to build more accurate classifiers [49]. This can be useful when labeled data is expensive or label assignment is uncertain. Many semi-supervised methods are transductive, implying that a traditional supervised classifier is often trained over the results of the semi-supervised labeling to provide an inductive model for the classification of future observations. In this study, the collection of semi-supervised methods acts as a transductive algorithm which assigns labels only to that subset of the unlabeled data which receives a unanimous labeling vote. This subset is then added to the original training set in order to train a new SVM.

As mentioned previously, any addition to the training set that weakens classifier performance on initially labeled data should be avoided. This is an important consideration with semi-supervised methods because it can be very difficult to correctly fit model parameters to the problem at hand when labels are missing [50]. Thus, to mitigate the risk of a poor match between the problem and the model, a variety of methods with a variety theoretical frameworks make up the collection of methods being used for the transductive step described above. The tested methods are described below. Note that for all four semi-supervised methods, the original labeled training set was used as the set of labeled data.

5.1. Method 1: modified yarowsky algorithm (MY)

Self-training methods are semi-supervised classification tools which wrap around an existing supervised classifier, known as the base classifier. In general, they are easy to understand and implement, making them a common starting point for semi-supervised investigations. Because they are wrappers, self-training methods can be applied to almost any complicated classification framework [50]. In 1995, Yarowsky introduced an iterative rule-based self-training algorithm for classification problems in computational linguistics [51]. The algorithm consists of the following steps:

1. Train the base classifier using the available labeled data.
2. Feed all initially unlabeled data into the trained classifier. For each data point, if the probability of belonging to a certain class is greater than a predetermined threshold, add that data point with its corresponding label to the original labeled training set.
3. Retrain the base classifier using the updated training set.
4. Repeat steps (2) and (3) until label convergence is reached.

Note that since the labels of all initially unlabeled data are reassigned based on class probability for each iteration, label assignments in earlier iterations can be changed in later iterations. While this method was successful for Yarowsky's purposes, the algorithm did not receive a robust mathematical analysis until Abney's investigation in 2004 [52]. In that work, Abney showed that a slightly modified version of Yarowsky's algorithm minimizes a reasonable objective function. His modifications include fixing the probability threshold at $1/L$ where L is the number of classes and imposing the condition that once a data point gains a label, it can change labels, but it cannot become unlabeled.

In this study, the base classifier is the SVM used to establish the baseline classification performance. Only five iterations were necessary to achieve label convergence.

5.2. Method 2: safe semi-supervised support vector machine (S4VM)

Transductive SVMs (TSVM) are a class of semi-supervised classification methods derived from the framework of traditional SVMs. Consider our SVM framework above. Now, suppose we introduce an unlabeled feature set $\hat{x}_1, \hat{x}_2, \cdots, \hat{x}_u$ where $\hat{x}_j \in \mathbb{R}^n$ for $j = 1, 2, \cdots, u$. These unlabeled data have corresponding slack variables $\hat{\xi}_j \geq 0$, and constraints similar to those in Eq. (24). However, the labels $\hat{y}_j$ are variables instead of parameters since they are not assigned. For the TSVM problem, Eq. (25) becomes

$$J(w, \xi, \hat{\xi}) = \frac{1}{2}||w||^2 + C_1 \sum_{i=1}^{m} \xi_i + C_2 \sum_{j=1}^{u} \hat{\xi}_j \tag{29}$$

where $C_1, C_2 > 0$ and $\hat{y}_j \in \{-1, 1\}$ for $j = 1, 2, \cdots, u$ [53].

Minimizing Eq. (29) in its current form is a combinatorial nightmare and is known to be NP-hard [54]. Thus, many less expensive methods aimed at approximating the TSVM solution have been proposed [53–57]. While these algorithms have made the problem tractable, they still must be used with caution because properly tuning hyperparameters in a semi-supervised environment is difficult and converging to a decision boundary which hurts classifier performance is an ever-present danger. Considering these concerns, the Safe Semi-Supervised SVM (S4VM) algorithm developed by Li and Zhou [57] is of particular interest. In contrast to other TSVM implementations which converge to a single optimal decision boundary, the S4VM algorithm builds a pool of candidate low-density separators and then chooses labels for the unlabeled data which maximize the performance for any separator. Li and Zhou determined these labels through simulated annealing and heuristic representative sampling approaches. Most importantly, Li and Zhou showed that S4VM is relatively insensitive to choice of hyperparameters and claimed that the algorithm never performs significantly worse in a statistical sense than an SVM trained only on the labeled data.

For these reasons, S4VM is used to approximate the TSVM solution in this study. For computational efficiency, the sampling technique was used instead of simulated annealing, and the algorithm was implemented using Li and Zhou's MATLAB package. While most parameters were left at default values, the γ parameter for the Gaussian kernel and the C_1 penalty parameter were set to 0.01 and 10 respectively to mirror the optimized hyperparameters for the baseline SVM.

5.3. Method 3: label propagation (LP)

Many methods developed for the semi-supervised classification problem are graph-based—see [50] for a thorough overview. In discrete mathematics, a graph is an abstract entity which captures pairwise relationships between elements within a finite set. Formally, a graph (also known as a simple graph) is an ordered pair $G = (V, E)$ comprised of a finite, nonempty set V of elements called vertices and a set E containing pairs of distinct elements of V called edges. Graphs can be thought of as geometrical objects by representing vertices as points and corresponding edges as lines between them in a planar space. An example is given in Fig. 6. In a weighted graph, each edge $\alpha = \{x, y\} \in E$ is assigned a non-negative number $c(\alpha)$ (called as its weight) through the weight function c. Weights generalize graphs by allowing the relationship between two vertices to go beyond the existence or absence of an edge, allowing physical concepts such as distance or cost to be modeled. Further details on graph theory can be found in [58].

Zhu and Ghahramani created a graph-based semi-supervised learning algorithm based on the construction of a weighted graph [59]. It is known as label propagation. This algorithm 'pushes' labels from labeled data points to unlabeled data points under the assumption that close proximity implies similarity. Suppose we have l labeled and u unlabeled data points where $Y_L = \{y_1, y_2, \cdots, y_l\}$ are the observed labels and $Y_U = \{y_{l+1}, y_{1+2}, \cdots, y_{1+u}\}$ are unobserved. Let there be C possible class labels. The graph is built by first placing all labeled and unlabeled data points into the set of vertices. Each vertex is then connected to every other vertex in the set of edges, producing what is known as a complete graph. For edge $\alpha = \{x, y\}$ where x, y are vertices, the weight function is given by

$$c(\alpha) = \exp\left(-\frac{d_{xy}^2}{\sigma^2}\right) \tag{30}$$

where d_{xy} is the Euclidean distance in the feature space between the data points represented by the vertices x, y and σ is a previously given parameter. Thus, data points which are closer in the feature space have larger edge weights.

Now, define a $(l+u) \times (l+u)$ matrix T

$$T_{ij} = \frac{c(\{i,j\})}{\sum_{k=1}^{l+u} c(\{k,j\})} \tag{31}$$

where T is interpreted as the probability of traveling from vertex j to vertex i. Furthermore, let Y_0 be a $(l+u) \times C$ matrix where the ith row contains the label probability distribution for data point i. Assume that the first l rows correspond to the labeled data points. The rows of labeled data points contain a 1 in the appropriate column and 0 in all others; the distributions for the unlabeled data points are arbitrarily assigned. The algorithm is then conducted through the following procedure where t denotes the current iteration number:

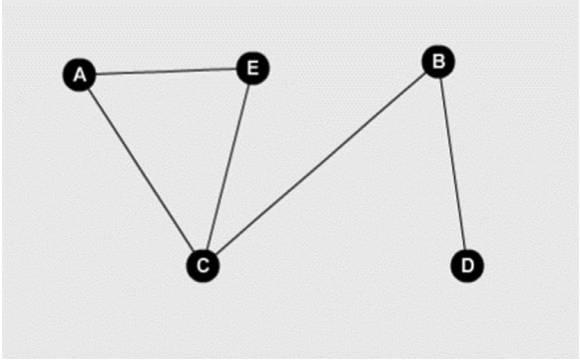


Fig. 6. Geometrical representation of graph $G = (V, E)$ where $V = \{v_A, v_B, v_C, v_D, v_E\}$ and $E = \{\{v_A, v_C\}, \{v_A, v_E\}, \{v_B, v_C\}, \{v_B, v_D\}, \{v_C, v_E\}\}$.

1. Propagate the labels by applying the transition matrix to the current label distribution matrix $Y_t = TY_{t-1}$.
2. Row normalize Y_t to maintain the probability interpretation.
3. Set the first l rows of Y_t equal to the first l rows of Y_0. This is known as clamping the labels.
4. Repeat until desired convergence criteria are met. Once converged, unlabeled data points are assigned that label with the highest probability.

In their paper, Zhu and Ghahramani prove this algorithm converges to a fixed point solution regardless of the initial label probability distributions for the unlabeled data. Step 3 ensures that a constant push is provided from initially labeled data points and that classes with fewer labeled data points are not pushed out.

At this point, choice of the σ parameter has yet to be discussed. In place of optimization through cross validation on the labeled data, Zhu and Ghahramani propose a heuristic based on the 3σ rule of a Gaussian distribution, but for our problem this heuristic produced a value of σ on the order of 10^{-3}, which caused the algorithm to become numerically ill-conditioned as $c(\alpha)$ approached zero for large values of d_{xy}^2. Further investigation showed that this ill-conditioning persisted for values of $\sqrt{\sigma}$ on the order of 10^{-1} and lower. Conversely, values of $\sqrt{\sigma}$ on the order of 10 or higher made the radius of influence of each vertex so large that the algorithm assigned the label of the class with the majority of data points in the training set (Precipitate) to each data point in the ambiguous set. However, setting $\sigma = 1$ avoided both numerical ill-conditioning and assignment of the same label to the entire ambiguous set; thus, we adopted this value of σ for our problem. Label propagation was implemented through scikit-learn [47].

5.4. Method 4: COP-KMEANS clustering (CKM)

Semi-supervised clustering methods are adaptations of traditional unsupervised clustering algorithms designed to take advantage of partial information. In contrast to the other semi-supervised methods discussed above, this information is not necessarily labeled data and often takes the form of pairwise linkage constraints. Two common types of linkage constraints are 'must-link,' in which two data points must be in the same cluster, and 'cannot-link,' where two data points cannot be in the same cluster [60].

Many studies modifying clustering algorithms to exploit partial information start with K-means clustering [61–65]. For a given K, this method partitions the data set into K clusters such that the sum of intra-cluster variance (based on Euclidean distance) is minimized. An outline of the K-means algorithm is presented below:

1. Randomly assign each data point a cluster label from 1 to K.
2. Compute the centroid, or vector of means for each dimension in the feature space, of each cluster.
3. Reassign each data point to the cluster whose centroid it is closest to.
4. Repeat steps (2) and (3) until cluster assignments converge.

A rigorous mathematical description can be found in [18].

Wagstaff et al. [65] introduced must-link and cannot-link constraints into this algorithm by slightly adjusting step (3). That is, instead of each data point being assigned to the cluster whose centroid is closest, each data point is assigned to the closest cluster which does not violate any of the given constraints. Their method is known as COP-KMEANS.

In order to utilize COP-KMEANS for this study, K was chosen to be 2 and linkage constraints were derived from the labeled training set by giving must-link conditions to those microstructures of the same class and cannot-link to those of opposite classes. However, specifying all of these constraints introduced a large degree of redundancy and complication into the model. Thus, to relieve model complexity, must-link

constraints were established between one precipitate microstructure P_i and all other precipitate microstructures in the training set. The same procedure was followed for one bicontinuous microstructure B_l, and then one cannot-link constraint was specified between P_i and B_l. Once all of the necessary constraints were specified, COP-KMEANS was applied using Babaki's implementation in python [66].

6. Classification of labeled data using the updated training set

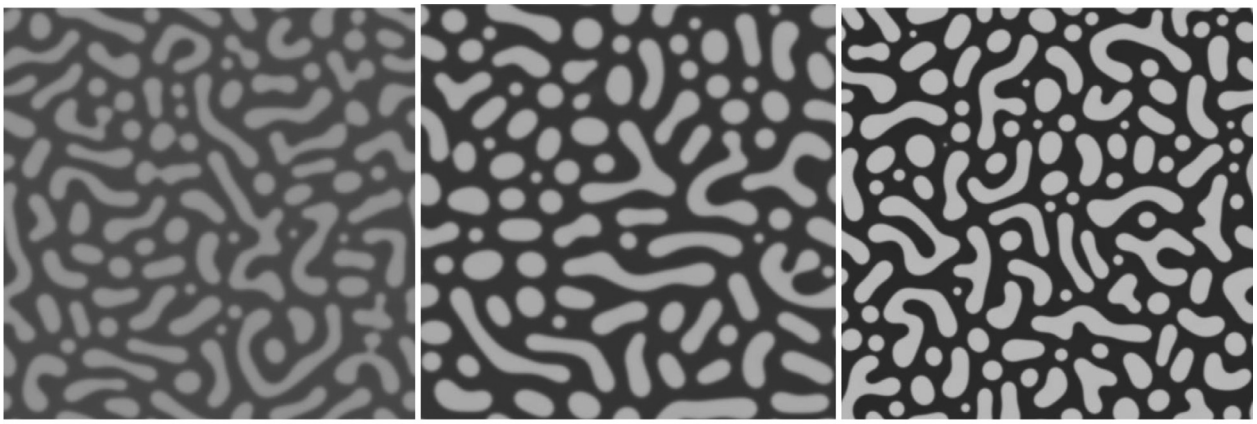
To reiterate, the purpose of applying the collection of semi-supervised methods described above was to identify a subset of the initially unlabeled microstructures for which a labeling consensus could be reached. This subset is then added to the initially labeled training set in order to train a new SVM with a decision boundary which is informed by both high-confidence and ambiguous examples of each class. The idea is that only adding this subset to the training set will lessen the risk of degrading classifier performance posed by many semi-supervised methods. The four semi-supervised learning methods agreed on 301 of 519 initially unlabeled microstructures (about 58%). The 301 microstructures from the initially unlabeled set which received an identical label vote from each of the four semi-supervised methods were added to the original training set with their corresponding labels.

Fig. 7 shows examples of initially ambiguous microstructures that were subsequently assigned labels through consensus. As can be seen, the example microstructures are truly ambiguous in that it is somewhat challenging to decide on the class they belong to. However, closer inspection of the dominant features in each of the two subsets seems to make intuitive sense: microstructures labeled as 'Bicontinuous' tend to have more elongated and tortuous single-phase domains, while those labeled as 'Precipitate' tend to have motifs that are closer to precipitate-like morphologies (or at least have them in greater numbers).

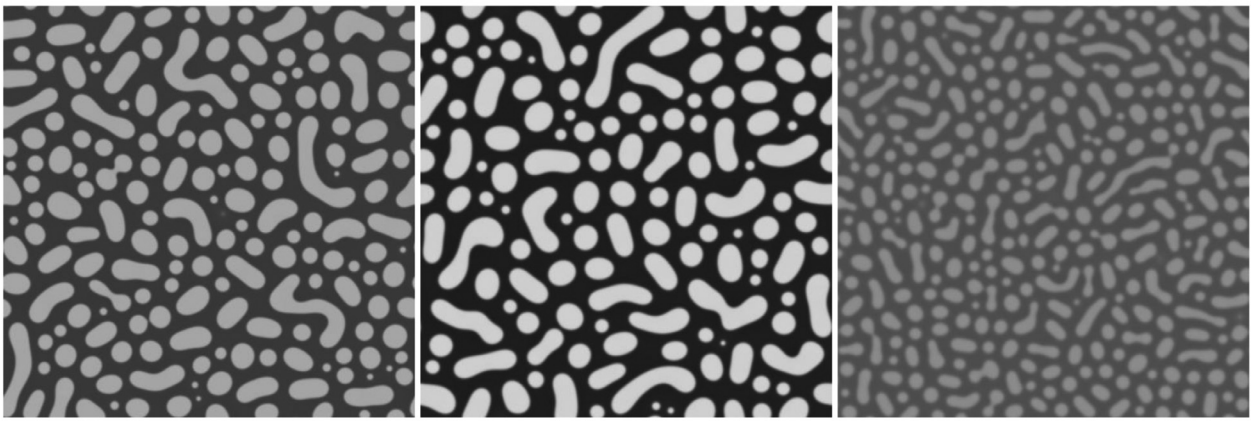
Following the same procedure used for the baseline SVM, kernel and hyperparameter selection was optimized through an exhaustive grid search employing five-fold cross-validation. This resulted in a Gaussian kernel with $\gamma = 0.01$ a penalty parameter $C = 10$. This classifier, which will be referred to as the updated SVM, performed similarly to the baseline SVM on the high-confidence data, with estimated error rates of 0.0397 from the initially labeled training set and 0.0599 from the labeled test set.

While the error estimates for updated SVM regarding the initially labeled data are low, they are still higher than those of the baseline SVM. To determine whether there is a statistically significant difference between the performance of the baseline and updated SVMs on the high-confidence data, McNemar's test was employed. McNemar's test is a non-parametric statistical hypothesis test used to compare dependent categorical outputs, making it useful for evaluating relative classifier performance when resampling and retraining is too expensive or when test data is limited [67]. The test statistic for McNemar's test is determined through a contingency table, which summarizes how the two classifiers agree and disagree on the test set. An example is given in Table 2 where a is the number of points from the test set of size n which both classifiers labeled correctly, b denotes the number of points which the first classifier labeled correctly but the second classifier did not, and so on.

The null hypothesis is that the proportion of correctly classified points made by the first classifier is equal to that of the second



(a) Bicontinuous



(b) Precipitate

Fig. 7. Examples of ambiguous microstructures which were assigned to the (a) bicontinuous or (b) precipitate class by each semi-supervised method.

Table 2
Example of a contingency table used for calculating the test statistic for McNemar's test.

		Classifier 1	
		Correct	Incorrect
Classifier 2	Correct	a	b
	Incorrect	c	d

classifier. That is,

$$H_0 : \frac{a+b}{n} = \frac{a+c}{n}. \tag{32}$$

The test statistic is given by

$$\chi^2 = \frac{(b-c)^2}{b+c}. \tag{33}$$

Under the null hypothesis, this test statistic has a chi-squared distribution with one degree of freedom for sufficiently large values of $b+c$. As a rule of thumb, when $b+c<10$, the exact binomial variant is recommended [68]. Under the null hypothesis, the distribution of B, the random variable associated with b, conditioned on the number of discordant pairs $b+c$ is the binomial distribution $Binomial(b+c, 0.5)$. A two-sided p-value is then calculated from the observed value $B=b$ by multiplying the minimum of the upper and lower tail p-values by two [69]. As per popular convention, in this work the null hypothesis will be rejected if the p-value is less than 0.05. The contingency table for the baseline and updated SVMs regarding the labeled test set is given in Table 3.

The sum of discordant pairs is less than 10, so the exact variant of the test was used, resulting in a p-value of 0.625. Thus, we do not have evidence to reject the null hypothesis at the 95% confidence level, and we conclude that the difference in performance of baseline and updated SVMs on the high-confidence data is not statistically significant.

7. Estimating semi-supervised classification error

Estimating model error is of paramount importance in machine learning. Presently, estimating classification error when some or all labels are missing is a difficult problem which has yet to be studied extensively. To estimate the error for semi-supervised problem encountered in this study, we propose using a convex combination of error estimates for the high-confidence labeled and ambiguous unlabeled data. A rigorous mathematical discussion is given below.

7.1. Definitions

Let $\boldsymbol{X}$ be a feature vector of length d which is a member of either the labeled subpopulation π_L or the unlabeled subpopulation π_U. Regardless of which subpopulation $\boldsymbol{X}$ belongs to, it will have a corresponding true class label $Y \in \{0,1\}$. We define the error rates of a classifier $\psi : \mathbb{R}^d \to \{0,1\}$ with respect to these subpopulations as

$$\epsilon_U = P\Big(\psi(\boldsymbol{X}) \neq Y | \boldsymbol{X} \in \pi_U\Big) = E[|Y - \psi(\boldsymbol{X})| | \boldsymbol{X} \in \pi_U], \tag{34}$$

Table 3
Contingency table for the baseline and updated SVMs on the labeled test set.

		Baseline	
		Correct	Incorrect
Updated	Correct	360	3
	Incorrect	1	20

$$\epsilon_L = P\Big(\psi(\boldsymbol{X}) \neq Y | \boldsymbol{X} \in \pi_L\Big) = E[|Y - \psi(\boldsymbol{X})| | \boldsymbol{X} \in \pi_L]. \tag{35}$$

We note that the expectation of a random variable Z given event W can be expressed as

$$E[Z|W] = \frac{E[ZI_W]}{P(W)}, \tag{36}$$

where I is an indicator function. This allows us to rewrite Eqs. (34) and (35) as

$$\epsilon_U = \frac{E[|Y - \psi(\boldsymbol{X})| I_{\boldsymbol{X} \in \pi_U}]}{P(\boldsymbol{X} \in \pi_U)}, \tag{37}$$

$$\epsilon_L = \frac{E[|Y - \psi(\boldsymbol{X})| I_{\boldsymbol{X} \in \pi_L}]}{P(\boldsymbol{X} \in \pi_L)}. \tag{38}$$

It then follows that the overall error rate ϵ is given by

$$\begin{aligned} \epsilon &= P\Big(\psi(\boldsymbol{X}) \neq Y\Big) = E[|Y - \psi(\boldsymbol{X})|] \\ &= E[|Y - \psi(\boldsymbol{X})| I_{\boldsymbol{X} \in \pi_U}] + E[|Y - \psi(\boldsymbol{X})| I_{\boldsymbol{X} \in \pi_L}] \\ &= P(\boldsymbol{X} \in \pi_U)\epsilon_U + P(\boldsymbol{X} \in \pi_L)\epsilon_L, \end{aligned} \tag{39}$$

which is a convex combination of ϵ_U and ϵ_L weighted by the probabilities of the feature vector $\boldsymbol{X}$ belonging to π_U and π_L, respectively.

7.2. Labeled error estimation

Although the previous result provides us with a theoretical definition of the overall rate, in practice we can only estimate this value from available data. Suppose that this data is an i.i.d. sample $S_{n+m} = \{(\boldsymbol{X}_1, Y_1), \cdots, (\boldsymbol{X}_n, Y_n), \boldsymbol{X}_{n+1}, \cdots, \boldsymbol{X}_{n+m}\}$ where $\boldsymbol{X}_1, \cdots, \boldsymbol{X}_n \in \pi_L$ and $\boldsymbol{X}_{n+1}, \cdots, \boldsymbol{X}_{n+m} \in \pi_U$. From Eq. (38), we see that the sample estimator of ϵ_L is given by

$$\begin{aligned} \widehat{\epsilon}_L &= \frac{\widehat{E}[|Y - \psi(\boldsymbol{X})| I_{\boldsymbol{X} \in \pi_L}]}{\widehat{P}(\boldsymbol{X} \in \pi_L)} = \frac{\frac{1}{n+m}\sum_{i=1}^{n} |Y_i - \psi(\boldsymbol{X}_i)|}{\frac{n}{n+m}} \\ &= \frac{1}{n}\sum_{i=1}^{n} |Y_i - \psi(\boldsymbol{X}_i)|, \end{aligned} \tag{40}$$

which is the well-known result for supervised error estimation. In this study, all supervised classifiers are independent of the sample being used to estimate the labeled error rate, so we can assume that the bias is negligible. Due to the lack of labels, ϵ_U cannot be estimated in the same fashion. We discuss our procedure for estimating this quantity below.

7.3. Unlabeled error estimation using agreement rates of multiple classifiers

For many recent machine learning problems, data generation and/or collection outpaces the labeling process, resulting in a plethora of information which is practically useless for traditional error estimation techniques. In response, exploring the unsupervised error estimation problem has gained traction over the last ten years. Only a few methods have been proposed, with most making limiting assumptions such as the label distribution being known [70] or that all classifiers make independent errors [71]. However, in [73], Platanios et al. introduce a simple algorithm which uses the sample agreement rate estimates of a collection of classifiers on only unlabeled data to estimate the individual and joint error rates. It requires no prior knowledge of the label distribution of the sample data used to estimate agreement rates, and it relaxes the independence assumption by turning it into the objective in an optimization problem. Let A be a set of classifiers and a_A and e_A be the agreement rate (the probability that all classifiers in A assign the same label) and error rate (the probability that all classifiers in A make the wrong prediction) for

that set of classifiers, respectively. Specifically, the objective to be minimized is given by

$$c_1(\boldsymbol{e}) = \sum_{A:|A|\geq 2} \left(e_A - \prod_{i \in A} e_i \right)^2, \tag{41}$$

which is effectively minimizing error rate dependence by making the joint error rates close in value to the product of their marginal rates. Equality constraints which relate sample agreement rate estimates to error rates are given by

$$\hat{a}_A = e_A + 1 + \sum_{k=1}^{|A|} [(-1)^k \sum_{\substack{I \subset A \\ |I| = k}} e_I], \tag{42}$$

where $\hat{a}_A$ is simply the number of data points which received the same prediction from each classifier in A divided by the sample size. Furthermore, the following inequality constraints

$$e_A \leq \min_{i \in A} e_{Ai} \tag{43}$$

for $|A| \geq 2$ ensure that all joint error rates are properly bounded by the values of their corresponding marginal error rates. Platanios et al. go on to recommend constraining some fraction of the individual error rates to be less than 0.5 in order to avoid solutions which imply that most of the classifiers perform worse than chance. We decided to implement this idea through the following constraint

$$\min_{i \in A} e_i \leq 0.5 \tag{44}$$

which simply forces at least one of the individual error rates to be less than 0.5. We also considered a second objective which attempts to minimize the sum of all of the individual error rates

$$c_2(\boldsymbol{e}) = \sum_{i \in A} e_i. \tag{45}$$

While c_2 has the potential to give a more optimistic solution than c_1, it takes the independence of error rates out of the problem completely.

Since the framework above allows for the estimation of error of multiple classifiers, we decided to create four new training sets by combining the original labeled training data with the output of each semi-supervised method. A new SVM was then trained over each of these training sets, giving us a set of five classifiers including the updated SVM. The predictions of all five classifiers were then collected on the ambiguous set in order to estimate agreement rates. Optimization was implemented using Sequential Quadratic Programming in Matlab.

7.4. Overall error estimation

Once $\hat{\epsilon}_L$ and $\hat{\epsilon}_U$ are available, an estimate of the overall error rate $\hat{\epsilon}$ can be obtained from Eq. (39):

$$\hat{\epsilon} = \hat{P}(\boldsymbol{X} \in \pi_U)\hat{\epsilon}_U + \hat{P}(\boldsymbol{X} \in \pi_L)\hat{\epsilon}_L = \frac{m}{m+n}\hat{\epsilon}_U + \frac{n}{m+n}\hat{\epsilon}_L. \tag{46}$$

The labeled, unlabeled, and overall error estimates for all five classifiers are given in Tables 4 and 5.

There are a few observations of note. The first is that optimization of both objectives resulted in very consistent solutions for unlabeled error estimation. This could be a result of the constraints. That is, while there are infinitely many solutions to the equality constraints (which is why optimization was necessary in the first place), the additional inequality constraints could have resulted in an extremely small feasible region in the design space, leading to similar solutions for both objectives. Along these lines, the addition of the constraint given in Eq. 44, which only forced one of the five individual error rates to be less than 0.5, led to solutions in which all individual error rates are below this threshold, which still aligns with the assumption made by Platanios et al. that most of the classifiers must have error rates better than chance. The second observation is that the updated SVM, which was trained on the initially labeled training data and the subset of ambiguous data which received a labeling consensus from all four semi-supervised methods, had the lowest labeled, unlabeled, and overall error estimates. Thus, for this particular problem, only adding this subset of the initially unlabeled data to the training set helped to avoid potential performance degradation on the high-confidence data and resulted in the best decision boundary for the ambiguous data. Lastly, both labeled and unlabeled error estimates for the SVMs trained over the results of the individual semi-supervised methods show that the Modified Yarowsky and S4VM algorithms assigned labels in a less detrimental way than Label Propagation and COP-KMEANS. This could be an artifact of the degree to which algorithm assumptions matched the given problem and is further evidence that more than one semi-supervised method should be considered when little is known about the distribution(s) of the data.

Table 4
Error estimation results using objective c_1. The SVMs trained on the results of a specific semi-supervised method are denoted by the abbreviation of that semi-supervised method.

Classifier	$\hat{\epsilon}_L$	$\hat{\epsilon}_U$	$\hat{\epsilon}$
MY	0.0625	0.0929	0.0690
S4VM	0.0677	0.0538	0.0647
LP	0.0911	0.2038	0.1151
CKM	0.0964	0.1557	0.1090
Updated	0.0599	0.0271	0.0529

Table 5
Error estimation results using objective c_2. The SVMs trained on the results of a specific semi-supervised method are denoted by the abbreviation of that semi-supervised method.

Classifier	$\hat{\epsilon}_L$	$\hat{\epsilon}_U$	$\hat{\epsilon}$
MY	0.0625	0.0963	0.0701
S4VM	0.0677	0.0520	0.0644
LP	0.0911	0.2004	0.1144
CKM	0.0964	0.1522	0.1082
Updated	0.0599	0.0289	0.0533

A complete suite of scripts with full implementation of all methods on an abridged version of the dataset for replication and verification can be found in [73].

8. Conclusions

Microstructure characterization and classification has been identified as an important step in building processing-structure-property linkages for the ultimate goal of materials by design. While the supervised classification problem is straight-forward given an appropriate metric and a robustly labeled training set, the high level of variation in a material's internal structure often hinders their acquisition. As we move to generalize the microstructure classification problem to encompass both established and emerging material systems, we must recognize that class taxonomy will be an ambiguous and dynamic entity which will require tools beyond human inspection to define and update. In response, we considered the specific problem of binary classification where class assignment was certain for some microstructures and ambiguous for others and proposed a data-driven classification framework which uses a collection of semi-supervised learning methods to identify the largest 'safe' subset of the ambiguous microstructures to label and add to the training set. We showed that the addition of this subset, consisting of almost 58% of the ambiguous

sample, to the training set did not degrade supervised classifier performance on high-confidence microstructures and that reliance on the consensus of multiple semi-supervised methods mitigated the risk of adding detrimental information to the training set. We also showed how classifier error can be estimated for the semi-supervised problem when it cannot be assumed that classifiers make independent errors. Although this paper has made an important step, future work will have to address changes in class taxonomy through the identification of emerging classes, the partitioning of old, broad classes into new, more specific subclasses, and so on. It must be stressed that the aim of this work was not only to demonstrate that semi-supervised learning methods can be used to train high-performing microstructure classification models. We also showed that automated, data-driven tools can be used in conjunction with human experience and rationality to uncover subtle relationships in complex microstructural systems. It is our hope that future studies on the microstructure classification problem will leverage the paradigm of data-driven science to accelerate the discovery of new useful information about the materials microstructure space, rather than simply training supervised classifiers in order to automate otherwise tedious tasks.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The authors would like to acknowledge the #Terra supercomputing facility of the Texas A&M University, for providing computing resources useful in conducting the research reported in this paper. The data is currently hosted by this facility. VA and RA thank the support by the National Science Foundation under NSF Grant No. CMMI-1462255 and NSF-CMMI-1663130. CK also acknowledges NSF support through Grant No. 1545403 (NRT-Data Enabled Discovery and Design of Energy Materials, D^3EM).

References

[1] S.R. Kalidindi, M. De Graef, Materials data science: current status and future outlook, Annu. Rev. Mater. Res. 45 (2015) 171–193.
[2] R. Arróyave, D.L. McDowell, Systems approaches to materials design: past, present, and future, Annu. Rev. Mater. Res. 49 (2019).
[3] R. Bostanabad, Y. Zhang, X. Li, T. Kearney, L.C. Brinson, D.W. Apley, W.K. Liu, W. Chen, Computational microstructure characterization and reconstruction: review of the state-of-the-art techniques, Prog. Mater. Sci. 95 (2018) 1–41.
[4] S.R. Niezgoda, A.K. Kanjarla, S.R. Kalidindi, Novel microstructure quantification framework for databasing, visualization, and analysis of microstructure data, Integr. Mater. Manufact. Innov. 2 (1) (2013) 54–80.
[5] B.L. DeCost, T. Francis, E.A. Holm, Exploring the microstructure manifold: image texture representations applied to ultrahigh carbon steel microstructures, Acta Mater. 133 (2017) 30–40.
[6] D.R. Gaston, C.J. Permann, J.W. Peterson, A.E. Slaughter, D. Andrš, Y. Wang, M.P. Short, D.M. Perez, M.R. Tonks, J. Ortensi, L. Zou, R.C. Martineau, Physics-based multiscale coupling for full core nuclear reactor simulation, Ann. Nucl. Energy 84 (2015) 45–54.
[7] V. Attari, A. Cruzado, R. Arroyave, Exploration of the microstructure space in TiAlZrN ultra-hard nanostructured coatings, Acta Mater. 174 (2019) 459–476.
[8] V. Attari, P. Honarmandi, T. Duong, D.J. Sauceda, D. Allaire, R. Arroyave, Uncertainty Propagation in a Multiscale CALPHAD-Reinforced Elastochemical Phase-field Model, Acta Mater 183 (2019) 452–470.
[9] M. Sanghvi, P. Honarmandi, V. Attari, T. Duong, R. Arroyave, D.L. Allaire, Uncertainty propagation via probability measure optimized importance weights with application to parametric materials models, in: Proceedings of the AIAA Scitech 2019 Forum, 2019, p. 0967.
[10] S.G. Fries, B. Boettger, J. Eiken, I. Steinbach, Upgrading CALPHAD to microstructure simulation: the phase-field method, Int. J. Mater. Res. 100 (2) (2009) 128–134.
[11] S. Curtarolo, G.L.W. Hart, M.B. Nardelli, N. Mingo, S. Sanvito, O. Levy, The high-throughput highway to computational materials design, Nat. Mater. 12 (3) (2013) 191–201.
[12] R. Bostanabad, A.T. Bui, W. Xie, D.W. Apley, W. Chen, Stochastic microstructure characterization and reconstruction via supervised learning, Acta Mater. 103 (2016) 89–102.
[13] S.M. Azimi, D. Britz, M. Engstler, M. Fritz, F. Mücklich, Advanced steel microstructural classification by deep learning methods, Sci. Rep. 8 (1) (2018).
[14] R. Kondo, S. Yamakawa, Y. Masuoka, S. Tajima, R. Asahi, Microstructure recognition using convolutional neural networks for prediction of ionic conductivity in ceramics, Acta Mater. 141 (2017) 29–38.
[15] J. Gola, D. Britz, T. Staudt, M. Winter, A.S. Schneider, M. Ludovici, F. Mücklich, Advanced microstructure classification by data mining methods, Comput. Mater. Sci. 148 (2018) 324–335.
[16] V. Sundararaghavan, N. Zabaras, Representation and classification of microstructures using statistical learning techniques, AIP Conference Proceedings, 712, AIP, 2004, pp. 98–102.
[17] V. Sundararaghavan, N. Zabaras, Classification and reconstruction of three-dimensional microstructures using support vector machines, Comput. Mater. Sci. 32 (2) (2005) 223–239.
[18] G. James, D. Witten, T. Hastie, R. Tibshirani, An Introduction to Statistical Learning, Springer New York, 2013.
[19] S.B. Kotsiantis, I. Zaharakis, P. Pintelas, Supervised machine learning: a review of classification techniques, Emerg. Artif. Intell. Appl. Comput. Eng. 160 (2007) 3–24.
[20] I.A. Okaro, S. Jayasinghe, C. Sutcliffe, K. Black, P. Paoletti, P.L. Green, Automatic fault detection for laser powder-bed fusion using semi-supervised machine learning, Addit. Manuf. 27 (2019) 42–53.
[21] G. Camps-Valls, T.V.B. Marsheva, D. Zhou, Semi-supervised graph-based hyperspectral image classification, IEEE Trans. Geosci. Remote Sens. 45 (10) (2007) 3044–3054.
[22] M. Guillaumin, J. Verbeek, C. Schmid, Multimodal semi-supervised learning for image classification, in: Proceedings of the 2010 IEEE Computer Society Conference on Computer Vision and Pattern recognition, IEEE, 2010, pp. 902–909.
[23] D.P. Kingma, S. Mohamed, D.J. Rezende, M. Welling, Semi-supervised learning with deep generative models, in: Proceedings of the Advances in Neural Information Processing Systems, 2014, pp. 3581–3589.
[24] L. Bruzzone, M. Chi, M. Marconcini, A novel transductive svm for semisupervised classification of remote-sensing images, IEEE Trans. Geosci. Remote Sens. 44 (11) (2006) 3363–3373.
[25] T.R. Golub, D.K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J.P. Mesirov, H. Coller, M.L. Loh, J.R. Downing, M.A. Caligiuri, et al., Molecular classification of cancer: class discovery and class prediction by gene expression monitoring, Science 286 (5439) (1999) 531–537.
[26] D.J. Miller, J. Browning, A mixture model and em-based algorithm for class discovery, robust classification, and outlier rejection in mixed labeled/unlabeled data sets, IEEE Trans. Pattern Anal. Mach. Intell. 25 (11) (2003) 1468–1483.
[27] A. Mackay, B. Weigelt, A. Grigoriadis, B. Kreike, R. Natrajan, R. A'Hern, D.S. Tan, M. Dowsett, A. Ashworth, J.S. Reis-Filho, Microarray-based class discovery for molecular classification of breast cancer: analysis of interobserver agreement, JNCI J. Natl. Cancer Inst. 103 (8) (2011) 662–673.
[28] B.L. DeCost, E.A. Holm, A computer vision approach for automated analysis and classification of microstructural image data, Comput. Mater. Sci. 110 (2015) 126–133.
[29] C.-F. Tsai, Bag-of-words representation in image annotation: a review, ISRN Artif. Intell. 2012 (2012) 1–19.
[30] A. Choudhury, Y.C. Yabansu, S.R. Kalidindi, A. Dennstedt, Quantification and classification of microstructures in ternary eutectic alloys using 2-point spatial correlations and principal component analyses, Acta Mater. 110 (2016) 131–141.
[31] B. Gallagher, M. Rever, D. Loveland, T.N. Mundhenk, B. Beauchamp, E. Robertson, T. Han, Predicting compressive strength of consolidated molecular solids using computer vision and deep learning, arXiv preprint arXiv:1906.02130 (2019).
[32] S.-i. Yi, V. Attari, M. Jeong, J. Jian, S. Xue, H. Wang, R. Arroyave, C. Yu, Strain-induced suppression of the miscibility gap in nanostructured Mg 2 Si–Mg 2 Sn solid solutions, J. Mater. Chem. A 6 (36) (2018) 17559–17570.
[33] V. Attari, Open Phase-field Microstructure Database (OPMD), 2019. http://microstructures.net.
[34] N. Otsu, A threshold selection method from gray-level histograms, IEEE Transa. Syst. Man Cybern. 9 (1) (1979) 62–66.
[35] E.R. Dougherty, R.A. Lotufo, Hands-on Morphological Image Processing, 59, SPIE Press, 2003.
[36] S. van der Walt, J.L. Schönberger, J. Nunez-Iglesias, F. Boulogne, J.D. Warner, N. Yager, E. Gouillart, T. Yu, scikit-image: image processing in python, 2014.
[37] H. Xu, R. Liu, A. Choudhary, W. Chen, A machine learning-based design representation method for designing heterogeneous microstructures, J. Mech. Des. 137 (5) (2015) 051403.
[38] T.-S. Han, X. Zhang, J.-S. Kim, S.-Y. Chung, J.-H. Lim, C. Linder, Area of lineal-path function for describing the pore microstructures of cement paste and their relations to the mechanical properties simulated from μ-CT microstructures, Cem. Concr. Compos. 89 (2018) 1–17.
[39] M.V. Karsanina, K.M. Gerke, E.B. Skvortsova, D. Mallants, Universal spatial correlation functions for describing and reconstructing soil microstructure, PLOS ONE 10 (5) (2015) e0126515.
[40] S.R. Niezgoda, Y.C. Yabansu, S.R. Kalidindi, Understanding and visualizing microstructure and microstructure variance as a stochastic process, Acta Mater. 59 (16) (2011) 6387–6400.
[41] D.T. Fullwood, S.R. Niezgoda, S.R. Kalidindi, Microstructure reconstructions from 2-point statistics using phase-recovery algorithms, Acta Mater. 56 (5) (2008) 942–948.
[42] A. Gokhale, A. Tewari, H. Garmestani, Constraints on microstructural two-point correlation functions, Scr. Mater. 53 (8) (2005) 989–993.
[43] D. Wheeler, D. Brough, T. Fast, S. Kalidindi, A. Reid, Pymks: Materials knowledge system in python, 2014. http://pymks.org/en/latest/rst/README.html.

[44] H. Abdi, L.J. Williams, Principal component analysis, Wiley Interdiscip. Rev. Comput. Stat. 2 (4) (2010) 433–459.
[45] G. Duan, Y.-W. Chen, Batch-incremental principal component analysis with exact mean update, in: Proceedings of the 2011 Eighteenth IEEE International Conference on Image Processing, IEEE, 2011, pp. 1397–1400.
[46] D.A. Ross, J. Lim, R.-S. Lin, M.-H. Yang, Incremental learning for robust visual tracking, Int. J. Comput. Vis. 77 (1–3) (2008) 125–141.
[47] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay, Scikit-learn: machine learning in python, J. Mach. Learn. Res. 12 (2011) 2825–2830.
[48] B. Scholkopf, A.J. Smola, Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond, MIT Press, Cambridge, MA, USA, 2001.
[49] X. Zhu, A.B. Goldberg, Introduction to semi-supervised learning, Synthesis Lectu. Artif. Intell. Mach. Learn. 3 (1) (2009) 1–130.
[50] X. Zhu, Semi-supervised Learning Literature Survey, Technical Report, University of Wisconsin-Madison Department of Computer Sciences, 2008.
[51] D. Yarowsky, Unsupervised word sense disambiguation rivaling supervised methods, in: Proceedings of the Thirty-third Annual Meeting of the Association for Computational Linguistics, 1995, pp. 189–196.
[52] S. Abney, Understanding the yarowsky algorithm, Comput. Linguist. 30 (3) (2004) 365–395.
[53] T. Joachims, Transductive inference for text classification using support vector machines, in: Proceedings of the ICML, 99, 1999, pp. 200–209.
[54] J. Wang, X. Shen, W. Pan, On transductive support vector machines, Contemp. Math. (2007) 7–19.
[55] K. Bennett, A. Demiriz, Semi-supervised support vector machines, in: Proceedings of the Advances in Neural Information Processing Systems, 1999, pp. 368–374.
[56] F. Gieseke, A. Airola, T. Pahikkala, O. Kramer, Fast and simple gradient-based optimization for semi-supervised support vector machines, Neurocomputing 123 (2014) 23–32.
[57] Y.-F. Li, Z.-H. Zhou, Towards making unlabeled data never hurt, IEEE Trans. Pattern Anal. Mach. Intell. 37 (1) (2014) 175–188.
[58] R. Brualdi, Introductory combinatorics, fifth ed., Pearson Modern Classic, Pearson, 2018.
[59] X. Zhu, Z. Ghahramani, Learning from Labeled and Unlabeled Data with Label Propagation, Technical Report, Carnegie Mellon University, 2002.
[60] E. Bair, Semi-supervised clustering methods, Wiley Interdiscip. Rev. Comput. Stat. 5 (5) (2013) 349–361.
[61] S. Basu, A. Banerjee, R.J. Mooney, Active semi-supervision for pairwise constrained clustering, in: Proceedings of the 2004 SIAM International Conference on Data Mining, SIAM, 2004, pp. 333–344.
[62] S. Basu, M. Bilenko, R.J. Mooney, A probabilistic framework for semi-supervised clustering, in: Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2004, pp. 59–68.
[63] P. Bradley, K. Bennett, A. Demiriz, Constrained k-means clustering, Microsoft Technical Report (2000).
[64] M. Bilenko, S. Basu, R.J. Mooney, Integrating constraints and metric learning in semi-supervised clustering, in: Proceedings of the Twenty-first International Conference on Machine Learning, ACM, 2004, p. 11.
[65] K. Wagstaff, C. Cardie, S. Rogers, S. Schrödl, et al., Constrained k-means clustering with background knowledge, in: Proceedings of the ICML, 1, 2001, pp. 577–584.
[66] B. Babaki, Cop-kmeans version 1.5, 2017. https://doi.org/10.5281/zenodo.831850.
[67] T.G. Dietterich, Approximate statistical tests for comparing supervised classification learning algorithms, Neural Comput. 10 (7) (1998) 1895–1923.
[68] K. Rufibach, Assessment of paired binary data, Skeletal Radiol. 40 (1) (2010) 1–4.
[69] P.H. Westfall, J.F. Troendle, G. Pennello, Multiple MCnemar tests, Biometrics 66 (4) (2010) 1185–1191.
[70] P. Donmez, G. Lebanon, K. Balasubramanian, Unsupervised supervised learning i: estimating classification and regression errors without labels, J. Mach. Learn. Res. 11 (Apr) (2010) 1323–1351.
[71] A. Jaffe, B. Nadler, Y. Kluger, Estimating the accuracies of multiple classifiers without labeled data, in: Proceedings of the Artificial Intelligence and Statistics, 2015, pp. 407–415.
[72] E.A. Platanios, A. Blum, T. Mitchell, Estimating accuracy from unlabeled data, in: Proceedings of the Thirtieth Conference on Uncertainty in Artificial Intelligence, AUAI Press, 2014, pp. 682–691.
[73] C. Kunselman, Class Assignment in Ambiguous Microstructures, 2019. https://github.com/cjkunselman18/Class-Assignment-in-Ambiguous-Microstructures.