Exact expressions for structure selection in cluster expansions

Tim Mueller and Gerbrand Ceder*

Massachusetts Institute of Technology, 77 Massachusetts Avenue, Building 13-5056, Cambridge, Massachusetts 02139, USA

(Received 30 August 2010; published 12 November 2010)

The cluster expansion has proven to be a valuable tool in materials science to predict properties of configurationally ordered and disordered structures but the generation of cluster expansions can be computationally expensive. In recent years there have been efforts to make the generation of cluster expansions more efficient by selecting training structures in a way that minimizes approximate expressions for the variance of the predicted property values. We demonstrate that in many cases, these approximations are not necessary and exact expressions for the variance of the predicted property values may be derived. To illustrate this result, we present examples based on common applications of the cluster expansion such as bulk binary alloys. In addition we extend these structure selection techniques to Bayesian cluster expansions. These results should enable researchers to better analyze the quality of existing training sets and to select training structures that yield cluster expansions with lower prediction error.

DOI: 10.1103/PhysRevB.82.184107 PACS number(s): 61.50.Ah

In materials science, generalized Ising models[1] known as cluster expansions are widely used to study structure-property relationships among structures that share a similar underlying lattice.[2–12] Cluster expansions predict the value of a material property for a given structure very quickly and accurately, making it computationally feasible to search for structures that have optimal property values or perform statistical sampling to arrive at thermodynamic averages. However for each property and each material, a new cluster expansion must be created by fitting a parametrized function to a set of training data. Generating the training data can be computationally expensive, and in most cases the computational cost of generating a cluster expansion is essentially the cost of generating the training data. To reduce the expense of generating cluster expansions, methods have been developed to select structures for the training set in a way that reduces the amount of training data required to generate cluster expansions with acceptable prediction error.[13–15] Van de Walle and Ceder derived an approximate expression for the variance of the predicted property values and suggested selecting training structures in a way that minimizes this value.[14] More recently, Seko *et al.*[15] have proposed a similar approach in which random sampling is used to estimate the variance. In this paper we demonstrate that these approximate methods may not be necessary, and in many cases it is possible to generate a simple, exact expression for variance of the predicted property values. This result should enable researchers to more quickly and accurately evaluate the quality of cluster expansion training data and select new structures to add to the training set.

We start with an overview of the cluster expansion to introduce basic terms and concepts based primarily on the work of Sanchez *et al.*[16] In a cluster expansion the structure of the material is represented by variables assigned to specific sites, which we refer to as "site variables." For example, site variables are commonly used to specify which element occupies each site, in which case the cluster expansion is used to predict property values for a group of structures with the same underlying topology. For each site variable, a single-site basis is defined. In general an orthonormal single-site basis is used, such that

$$\frac{\sum_{s_j=1}^{N_j} \Theta_b(s_j)\Theta_{b'}(s_j)}{N_j} = \delta_{bb'}, \tag{1}$$

where s_j is the site variable for the jth site,[17] N_j represents the number of values this variable may take, $\Theta_b(s_j)$ is the bth basis function for the jth site, and $\delta_{bb'}$ is the Kronecker delta. For example, for a binary cluster expansion in which each site variable may take on the values of 1 or 2, the following commonly used basis meets the above orthogonality condition,

$$\Theta_0(s_j) = 1,$$

$$\Theta_1(s_j) = \cos(\pi s_j) = \pm 1. \tag{2}$$

The tensor product of all single-site basis functions produces a basis of "cluster functions." Each cluster function can be defined by a single vector $\mathbf{b}$,

$$\Phi_{\mathbf{b}}(\mathbf{s}) = \prod_j \Theta_{b_j}(s_j), \tag{3}$$

where $\mathbf{s}$ is the set of all site variables and b_j and s_j are the jth elements of $\mathbf{b}$ and $\mathbf{s}$ respectively. In general, Θ_0 is always "1," and the cluster function only depends on the cluster of sites for which $b_j \neq 0$. It can be shown that if the single-site basis functions are orthonormal, then the cluster functions must also be orthonormal,[16]

$$\frac{\sum_{i=1}^{N_{\mathbf{s}}} \Phi_{\mathbf{b}}(\mathbf{s}_i)\Phi_{\mathbf{b}'}(\mathbf{s}_i)}{N_{\mathbf{s}}} = \delta_{\mathbf{bb}'}, \tag{4}$$

where $\mathbf{s}_i$ is the ith set of possible values for the site variables and the sum is over all $N_{\mathbf{s}}$ such sets.

If a property of the material can be expressed as a function of the site variables, $F(\mathbf{s})$, then it can be expanded exactly as a linear combination of cluster functions,

$$F(\mathbf{s}) = \sum_{\mathbf{b}} V_{\mathbf{b}} \Phi_{\mathbf{b}}(\mathbf{s}), \tag{5}$$

where $V_{\mathbf{b}}$ are unknown coefficients called effective cluster interactions (ECIs) and the sum is over all cluster functions. Symmetry may be used to reduce the cluster expansion further, resulting in the general expression,

$$F(\mathbf{s}) = \sum_{\alpha} V_{\alpha} \sum_{\mathbf{b} \in \alpha} \Phi_{\mathbf{b}}(\mathbf{s}), \tag{6}$$

where α represents an orbit of symmetrically equivalent cluster functions. Because cluster expansions are often applied to infinite crystals, it is common to normalize all values per unit cell. Equation (6) can then be written as

$$f(\mathbf{s}) = \sum_{\alpha} V_{\alpha} m_{\alpha} \varphi_{\alpha}(\mathbf{s}), \tag{7}$$

where $f(\mathbf{s})$ is the property value per unit cell, m_{α} is the number of cluster functions in α per unit cell, and $\varphi_{\alpha}(\mathbf{s})$, sometimes referred to as a correlation function, is the average value of all cluster functions in α,

$$\varphi_{\alpha}(\mathbf{s}) = \frac{\sum_{\mathbf{b} \in \alpha} \Phi_{\mathbf{b}}(\mathbf{s})}{N_{\alpha}}. \tag{8}$$

Typically a cluster expansion is truncated so that there are a finite number unknown ECI corresponding to cluster functions that are dependent on a finite number of sites. Values for these ECI may be determined using a set of training data. The training data can be represented by a matrix $\mathbf{X}$ in which

$$X_{i\alpha} = \varphi_{\alpha}(\mathbf{s}_i), \tag{9}$$

where $\mathbf{s}_i$ is the set of site variables for the ith element of the training set. The ECI may then be estimated using a least-squares fit,

$$\hat{\mathbf{v}} = (\mathbf{X}^{\mathbf{T}}\mathbf{X})^{-1}\mathbf{X}^{\mathbf{T}}\mathbf{y}, \tag{10}$$

where $\hat{\mathbf{v}}$ is a column vector in which the αth element is the predicted value for $m_{\alpha}V_{\alpha}$, and $\mathbf{y}$ is a column vector in which y_i is the value for the property of interest for the ith element of the training set.

Van de Walle and Ceder proposed selecting training structures by using the fact that the variance of the prediction error for a structure with site variables $\mathbf{s}$ is given by[14]

$$\mathbf{x}\mathbf{M}\mathbf{x}^{\mathbf{T}}, \tag{11}$$

where $\mathbf{M}$ is the covariance matrix of the predicted ECI and $\mathbf{x}$ is a row vector with elements

$$x_{\alpha} = \varphi_{\alpha}(\mathbf{s}). \tag{12}$$

A standard model for linear regression is that the observed property values, $\mathbf{y}$, are related to the training data by

$$\mathbf{y} = \mathbf{X}\mathbf{v} + \tilde{\mathbf{e}}, \tag{13}$$

where $\mathbf{v}$ is a column vector of the true ECI and $\tilde{\mathbf{e}}$ is a vector of randomly distributed noise. In a cluster expansion, the noise is generally due to the truncation of the cluster expansion and, in some cases, the use of nondeterministic algorithms such as Monte Carlo simulations to calculate property values. If the elements $\tilde{e}_i$ are drawn from independent distributions with a mean of zero and variance σ^2, it can be shown that[18]

$$\mathbf{M} = (\mathbf{X}^{\mathbf{T}}\mathbf{X})^{-\mathbf{1}}\sigma^2. \tag{14}$$

Although σ^2 can be estimated statistically, for the purposes of this paper it is safe to treat it as an unknown constant. Because of the matrix inversion in Eq. (14), the covariance matrix can only be calculated if $\mathbf{X}$ contains at least as many columns (structures in the training set) as rows (symmetrically distinct cluster functions included in the fit). When generating initial structures for the training set, this means that some cluster functions that might be significant are ignored. Alternatively, in a Bayesian cluster expansion,[19] the predicted ECI are given by

$$\mathbf{v} = (\mathbf{X}^{\mathbf{T}}\mathbf{X} + \mathbf{\Lambda})^{-1}\mathbf{X}^{\mathbf{T}}\mathbf{y}, \tag{15}$$

where $\mathbf{\Lambda}^{-1}\sigma^2$ is the covariance matrix for a multivariate Gaussian prior distribution for ECI values. Bayesian cluster expansions generally have lower prediction error than standard cluster expansions, in part because an arbitrarily large number of distinct cluster functions may be included, regardless of training set size.[19] It can be shown that the covariance matrix for the predicted ECI for a Bayesian cluster expansion is[20]

$$\mathbf{M} = (\mathbf{X}^{\mathbf{T}}\mathbf{X} + \mathbf{\Lambda})^{-\mathbf{1}}\sigma^2. \tag{16}$$

For either least-squares or Bayesian regression, the expected variance for the predicted property values for a given population of structures is

$$\langle \mathbf{x}\mathbf{M}\mathbf{x}^{\mathbf{T}} \rangle_{pop} = \mathbf{M} : \langle \mathbf{x}^{\mathbf{T}}\mathbf{x} \rangle_{pop}, \tag{17}$$

where $\langle\ \rangle_{pop}$ indicates the average value over all structures in the population and the : symbol represents the Frobenius inner product, defined as

$$\mathbf{A} : \mathbf{B} = \sum_i \sum_j A_{ij} B_{ij}. \tag{18}$$

We will call the matrix $\langle \mathbf{x}^{\mathbf{T}}\mathbf{x} \rangle_{pop}$ the domain matrix and represent it with the symbol $\mathbf{D}$. As long as the domain matrix is known, the dimensionless quantity

$$\frac{\mathbf{M} : \mathbf{D}}{\sigma^2} \tag{19}$$

can be calculated exactly. The numerator of Eq. (19) is the expected variance of the predicted property values and the denominator represents the contribution to the prediction error from factors that have nothing to do with structure selection. The ratio is therefore a measure of how well a given set of training structures reduces prediction error. Seko *et al.* used the symbol Λ to represent the expression in Eq. (19),[15] but to avoid confusion with the symbols used in the formalism of the Bayesian cluster expansion, we will use the symbol τ. For a least-squares fit,

$$\tau = (\mathbf{X}^{\mathbf{T}}\mathbf{X})^{-\mathbf{1}} : \mathbf{D} \tag{20}$$

and for a Bayesian cluster expansion,

$$\tau = (\mathbf{X}^{\mathbf{T}}\mathbf{X} + \boldsymbol{\Lambda})^{-1}:\mathbf{D}, \tag{21}$$

where once again the symbol : represents the Frobenius inner product. Attempts to find good training sets have focused on minimizing τ, using either approximate methods[14] or statistical sampling[15] to calculate $\mathbf{D}$. Here we demonstrate that in many common cases, the domain matrix may be calculated exactly.

The elements of the domain matrix may be written as

$$D_{\alpha\beta} = \langle x_\alpha x_\beta \rangle_{pop} = \langle \varphi_\alpha(\mathbf{s})\varphi_\beta(\mathbf{s})\rangle_{pop} = \frac{\sum_{\mathbf{b}\in\alpha}\sum_{\mathbf{b}'\in\beta}\langle \Phi_{\mathbf{b}}(\mathbf{s})\Phi_{\mathbf{b}'}(\mathbf{s})\rangle_{pop}}{N_\alpha N_\beta}, \tag{22}$$

where once again $\langle\,\rangle_{pop}$ represents the average over a population of structures, and N_α and N_β are the number of cluster functions in orbits α and β, respectively. Equation (22) is valid for any population of structures. For example, if the population contains all possible structures, the orthonormality of the basis [Eq. (4)] yields

$$D_{\alpha\beta} = \frac{\sum_{\mathbf{b}\in\alpha}\sum_{\mathbf{b}'\in\beta}\delta_{\mathbf{b}\mathbf{b}'}}{N_\alpha N_\beta} = \frac{\delta_{\alpha\beta}}{N_\alpha}. \tag{23}$$

In an infinite crystal, for each cluster function that depends on a finite, positive number of sites, there are an infinite number of distinct vectors $\mathbf{b}$ that represent symmetrically equivalent cluster functions. For these cluster functions, N_α is infinite and $D_{\alpha\beta}$ is therefore zero. The only cluster function typically included in the cluster expansion that does not meet this criterion is the "empty" cluster function, represented by $b_j=0$ for all j. There is only one vector $\mathbf{b}$ that represents the empty cluster function, even in an infinite crystal. For convenience the empty cluster function is assigned the index $\alpha=0$, and for an infinite crystal D_{00} is typically the only nonzero element,

$$\mathbf{D} = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & 0 \end{bmatrix}. \tag{24}$$

Equations (17) and (24) yield the result that the average variance for the predicted property values over all structures is simply M_{00}.

It is possible to construct a more general expression for situations in which not all structures are of equal interest. Two cluster functions that are not dependent on any of the same sites are independent of each other, meaning

$$\langle \Phi_{\mathbf{b}}(\mathbf{s})\Phi_{\mathbf{b}'}(\mathbf{s})\rangle_{pop} = \langle \Phi_{\mathbf{b}}(\mathbf{s})\rangle_{pop}\langle \Phi_{\mathbf{b}'}(\mathbf{s})\rangle_{pop}. \tag{25}$$

If each cluster function is dependent on a finite number of sites, the probability that they are independent of each other in the limit of a crystal with an infinite number of sites is 1. In other words, the number of pairs of clusters that overlap is vanishingly small relative to the denominator $N_\alpha N_\beta$ in Eq. (22). Thus under the assumptions that all included cluster functions are dependent on a finite number of sites and the crystal has infinite periodicity, the elements of the domain matrix are given by

$$D_{\alpha\beta} = \frac{\sum_{\mathbf{b}\in\alpha}\sum_{\mathbf{b}'\in\beta}\langle \Phi_{\mathbf{b}}(\mathbf{s})\rangle_{pop}\langle \Phi_{\mathbf{b}'}(\mathbf{s})\rangle_{pop}}{N_\alpha N_\beta} = \langle x_\alpha\rangle_{pop}\langle x_\beta\rangle_{pop} \tag{26}$$

and the average variance of the predicted property values is given by

$$\langle \mathbf{x}\mathbf{M}\mathbf{x}^{\mathbf{T}}\rangle_{pop} = \langle \mathbf{x}\rangle_{pop}\mathbf{M}\langle \mathbf{x}^{\mathbf{T}}\rangle_{pop}. \tag{27}$$

This general result greatly simplifies the problem of calculating the domain matrix. As long as it is possible to determine the expected value of each included cluster function independently, the domain matrix may be calculated exactly. It is trivial to reconstruct from Eq. (26) the special case of Eq. (24).

It is common for the cluster expansion to be used in binary alloy systems where the only allowed values for $\Theta_1(s_j)$ are +1 and −1. If c is the concentration of the element assigned a value of +1, then the expected value of $\Theta_1(s_j)$ over all structures with concentration c is $(2c-1)$. If a cluster expansion is to be applied to the population of all structures with concentration c, the elements of the domain matrix are

$$D_{\alpha\beta} = (2c-1)^{n_\alpha + n_\beta}, \tag{28}$$

where n_α and n_β represent the number of sites upon which the cluster functions in orbits α and β are dependent. When $c=0.5$, Eq. (24) is recovered because in an infinite crystal the distribution of all possible structures is a delta function at $c=0.5$. Thus the average over all possible structures is essentially the same as the average over all structures with $c=0.5$. For the more common situation in which all concentrations are considered equally important, a domain matrix may be constructed by integrating $(2c-1)^{n_\alpha+n_\beta}$ over all concentrations uniformly. This is equivalent to taking a weighted average over all structures, where the weight for a structure with composition c is proportional to the inverse multiplicity of structures with composition c. The elements of the resulting domain matrix are

$$D_{\alpha\beta} = \int_{c=0}^{1} (2c-1)^{n_\alpha+n_\beta}dc = \begin{cases} \dfrac{1}{(n_\alpha+n_\beta+1)} & (n_\alpha+n_\beta) \text{ is even} \\ 0 & (n_\alpha+n_\beta) \text{ is odd.} \end{cases} \tag{29}$$

The domain matrix in Eq. (29) is appropriate for basic binary cluster expansions in which all compositions are of equal interest.

Once the domain matrix for a population of structures is known, the quality of different training sets can be evaluated by calculating τ. For example, consider a binary cluster expansion for an fcc alloy. In this example we include the empty cluster function, the orbit of single-site cluster functions, the orbit of two-site functions up to the fourth-nearest-neighbor, the orbit of three-site functions up to the second-

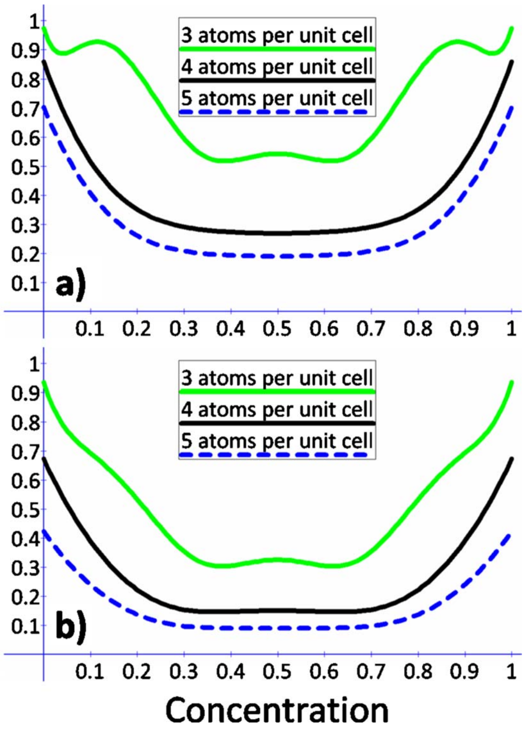

FIG. 1. (Color online) The value of τ for populations of A-B alloys with different concentrations of A. The three lines represent training sets containing (a) all symmetrically distinct structures, (b) all structures, with up to three atoms per unit cell, four atoms per unit cell, and five atoms per unit cell.

nearest-neighbor, and the orbit of four-site nearest-neighbor functions. We do not need to assign ECI values to these cluster functions, as the ECI are not needed to calculate τ. We consider three training sets of increasing size and computational cost: a 10-structure training set containing all symmetrically distinct structures with up to three atoms per unit cell, a 29-structure set containing all symmetrically distinct structures with up to four atoms per unit cell, and a 57-structure training set containing all symmetrically distinct structures with up to five atoms per unit cell. For each of these training sets, it is possible to use Eq. (28) to calculate the value of τ for the population of structures at a given composition. The results of these calculations are given in Fig. 1(a). In general, τ is lowest at the compositions at which the density of structures in the training set is the highest, and τ is highest (i.e., the predictive power of the cluster expansion is weakest) near $c=0$ and $c=1$. For this reason it might make sense to include a higher density of structures near the composition end points if all compositions are considered equally important.

Although training sets typically consist of a set of symmetrically distinct structures, this is not a requirement of the cluster expansion. It is instructive to consider the case in which multiple symmetrically equivalent structures are allowed in the training set as long as the vectors of the site variables, **s**, that characterize the structures are different. This is equivalent to weighting each structure in the training set by the number of distinct vectors **s** that yield symmetrically equivalent structures. As shown in Fig. 1(b), the results are similar to the case in which only symmetrically distinct structures are included in the training set but the value of τ is significantly reduced. However because it is more common to include only symmetrically distinct structures in the training set, we will only consider training sets consisting of symmetrically distinct structures for the remainder of this paper.

a) $$\mathbf{D}=\begin{bmatrix}1&0&0&0&0&0&0&0&0\\0&1&0&0&0&0&0&0&0\\0&0&1&0&0&0&0&0&0\\0&0&0&1&0&0&0&0&0\\0&0&0&0&1&0&0&0&0\\0&0&0&0&0&1&0&0&0\\0&0&0&0&0&0&1&0&0\\0&0&0&0&0&0&0&1&0\\0&0&0&0&0&0&0&0&1\end{bmatrix}$$

b) $$\mathbf{D}=\begin{bmatrix}1&0&0&0&0&0&0&0&0\\0&1/32&0&0&0&0&0&0&0\\0&0&1/192&0&0&0&0&0&0\\0&0&0&1/48&0&0&0&0&0\\0&0&0&0&1/192&0&0&0&0\\0&0&0&0&0&1/48&0&0&0\\0&0&0&0&0&0&1/256&0&0\\0&0&0&0&0&0&0&1/384&0\\0&0&0&0&0&0&0&0&1/64\end{bmatrix}$$

c) $$\mathbf{D}=\begin{bmatrix}1&0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0&0\end{bmatrix}$$

d) $$\mathbf{D}=\begin{bmatrix}1&0&1/3&1/3&1/3&1/3&0&0&1/5\\0&1/3&0&0&0&0&1/5&1/5&0\\1/3&0&1/5&1/5&1/5&1/5&0&0&1/7\\1/3&0&1/5&1/5&1/5&1/5&0&0&1/7\\1/3&0&1/5&1/5&1/5&1/5&0&0&1/7\\1/3&0&1/5&1/5&1/5&1/5&0&0&1/7\\0&1/5&0&0&0&0&1/7&1/7&0\\0&1/5&0&0&0&0&1/7&1/7&0\\1/5&0&1/7&1/7&1/7&1/7&0&0&1/9\end{bmatrix}$$

FIG. 2. The four domain matrices generated for the binary fcc cluster expansion used in the examples. The columns and rows are sorted first by the number of sites in the clusters and then by the maximum distance between sites. The matrices are generated by (a) the method of van de Walle and Ceder, (b) the method of Seko *et al.*, (c) giving all structures the same weight [Eq. (24)], and (d) weighting compositions uniformly [Eq. (29)].

In addition to the methods presented here, two other methods for generating the domain matrix have been proposed. Using the simplifying assumption that the correlations are distributed isotropically in a sphere, van de Walle and Ceder arrived at a result that is equivalent to using an identity matrix as the domain matrix.[14] Seko *et al.*[15] estimate the domain matrix by sampling 10 000 random structures from a 32-atom supercell. It is useful to compare the quality of the training sets generated using these different methods. For the binary fcc cluster expansion, we have used five different methods to generate initial training sets containing ten structures each. As the baseline approach, we include all symmetrically distinct structures with up to three atoms per unit cell. The remaining four methods are based on four different ways to calculate the domain matrix:

(1) The method of van de Walle and Ceder, in which the domain matrix is an identity matrix.

(2) The method of Seko *et al.* To eliminate any noise from sampling we enumerate all possible structures in a 32-atom $2\times2\times2$ supercell using an algorithm similar to the one presented by Hart and Forcade.[20,21] An equivalent result can be obtained by using Eq. (23), where N_α is the number of symmetrically distinct clusters in orbit α for a periodic $2\times2\times2$ supercell.

(3) Using Eq. (24), which assumes the population contains all possible structures equally weighted.

(4) Using Eq. (29), in which the population is weighted uniformly across compositions.

The four domain matrices for this example are shown in Fig. 2. Using each of the above four domain matrices to calculate τ, we use simulated annealing to find the set of ten structures, each containing up to five atoms per unit cell, that

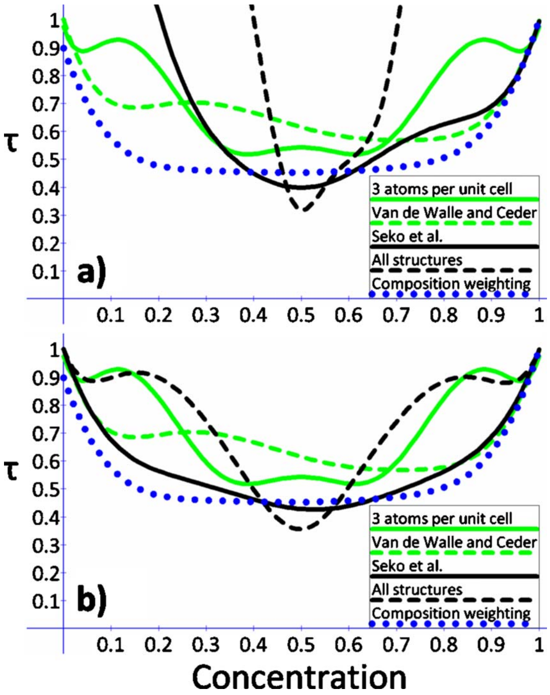

FIG. 3. (Color online) The effect of initial training set selection on the value of τ for populations of A-B alloys as a function of the concentration of A. The five lines represent training sets containing ten structures generating using five different methods. The first is the baseline method of including all structures with up to three atoms per unit cell. The other four are based on four different ways to generate the domain matrix: the method of van de Walle and Ceder, the method of Seko *et al.*, giving all structures the same weight [Eq. (24)], and weighting compositions uniformly [Eq. (29)]. (a) The pure elements may be left out of the training set. (b) The training set must include the pure elements.

minimizes the value of τ. The resulting minimal values of τ for the four different domain matrices are 7.303, 0.274, 0.101, and 0.301, respectively. The high value for τ using the method of van de Walle and Ceder is due to the fact that this method has much higher values for elements of the domain matrix other than D_{00}. This is likely an artifact of the approximations used to construct the domain matrix and it may result in estimates of the variance that are an order of magnitude too high.

For each of the five sets of training structures generated using the above methods, we calculate τ as a function of composition using Eq. (28). The results are shown in Fig. 3(a). The training set that produces the lowest value of τ averaged across all compositions is, by definition, the one generated using the domain matrix in Eq. (29). The method proposed by van de Walle and Ceder results in slightly higher values of τ for all compositions. Although the value of τ estimated by this method is much higher than the others, it produces a reasonably good training set for the prediction of property values across all compositions. The domain matrix for a population containing all structures, equally weighted, selects structures that have a composition near 0.5, resulting in very high prediction errors for other compositions (up to $\tau=12.6$ for $c=1$). This is due to the fact that the distribution of all structures is a delta function at $c=0.5$. The distribution of structures in a 32-atom unit cell is a binomial function of composition with finite width, and for this reason the method of Seko *et al.* does better with compositions near $c=0$ and $c=1$. The latter two methods produce similar domain matrices, as can be seen by comparing the domain matrices in Figs. 2(b) and 2(c). As the size of the supercell is increased, the method of Seko *et al.* becomes equivalent to the method of including all structures with equal weights.

In practice, Seko *et al.*[15] manually construct an initial set of training structures and then add structures that minimize τ to this set. This approach can significantly improve the performance of their method. For example, if the pure elements are manually included in the set of initial training structures and the method of Seko *et al.* is used to find the remaining eight structures, the average value of τ across all compositions is significantly reduced [Fig. 3(b)]. Alternatively, the method that weights all compositions equally can be used to generate, with no manual intervention, an initial set of training structures that includes the pure elements.

In some cases, the magnitude of the ECI might not decay with increasing cluster size, making it impossible to effectively truncate the cluster expansion. It has recently been demonstrated by Sanchez[22] that this problem may be alleviated by using a concentration-dependent basis such as the one proposed by Asta *et al.*[23] Here we demonstrate that changing the basis does not affect the structure selection method described in this paper. We first note that if an included cluster function is dependent on a set of sites, it is important to include all cluster functions that are dependent on subsets of those sites.[24] If all such cluster functions are included, the transformation to a concentration-dependent basis can be accomplished using a linear operator $\mathbf{A}$,[22]

$$\mathbf{x}' = \mathbf{x}\mathbf{A},$$

$$\mathbf{X}' = \mathbf{X}\mathbf{A}, \tag{30}$$

where $\mathbf{x}'$ and $\mathbf{X}'$ are, respectively, the representations of the vector $\mathbf{x}$ and the matrix $\mathbf{X}$ in the concentration-dependent basis. The variance of the predicted property values in the concentration-dependent basis is proportional to

$$\begin{aligned}\tau' &= \langle \mathbf{x}'(\mathbf{X}'^{\mathrm{T}}\mathbf{X}')^{-1}\mathbf{x}'^{\mathrm{T}}\rangle_{pop} = \langle \mathbf{x}\mathbf{A}(\mathbf{A}^{\mathrm{T}}\mathbf{X}^{\mathrm{T}}\mathbf{X}\mathbf{A})^{-1}\mathbf{A}^{\mathrm{T}}\mathbf{x}^{\mathrm{T}}\rangle_{pop} \\ &= \langle \mathbf{x}(\mathbf{X}^{\mathrm{T}}\mathbf{X})^{-1}\mathbf{x}^{\mathrm{T}}\rangle_{pop} = \tau. \end{aligned} \tag{31}$$

A similar result holds for Bayesian cluster expansions where $\mathbf{\Lambda}' = \mathbf{A}^{\mathrm{T}}\mathbf{\Lambda}\mathbf{A}$. Thus the value of τ is independent of the choice of basis. In general we find it most convenient to work with an orthonormal basis as defined in Eq. (4).

Although we have shown simple examples, the concepts we present in this paper can be applied to construct analytical expressions for the domain matrices for a wide variety of problems.[14] More accurate domain matrices yield better estimates of the variance of predicted property values, enabling researchers to both better analyze the quality of existing training data and to better select training structures that minimize the computational cost of generating cluster expansions. The cluster expansion has proven to be a valuable tool in materials science and as the efficiency of generating cluster expansions continues to improve it will become possible to apply this tool to increasingly complex problems.

This work was supported by the U.S. Department of Energy (DOE) under Contract No. DE-FG02-96ER45571.

*Author to whom correspondence should be addressed. FAX: (617) 258-6534; gceder@mit.edu

[1] E. Ising, Z. Phys. **31**, 253 (1925).

[2] V. Ozoliņš, C. Wolverton, and A. Zunger, Phys. Rev. B **57**, 6427 (1998).

[3] M. H. F. Sluiter, Y. Watanabe, D. de Fontaine, and Y. Kawazoe, Phys. Rev. B **53**, 6137 (1996).

[4] A. Van der Ven, M. K. Aydinol, G. Ceder, G. Kresse, and J. Hafner, Phys. Rev. B **58**, 2975 (1998).

[5] N. A. Zarkevich, T. L. Tan, and D. D. Johnson, Phys. Rev. B **75**, 104203 (2007).

[6] B. P. Burton, Phys. Rev. B **59**, 6087 (1999).

[7] C. Wolverton and A. Zunger, J. Electrochem. Soc. **145**, 2424 (1998).

[8] A. Seko, K. Yuge, F. Oba, A. Kuwabara, and I. Tanaka, Phys. Rev. B **73**, 184117 (2006).

[9] B. Kolb and G. L. W. Hart, Phys. Rev. B **72**, 224207 (2005).

[10] A. van de Walle, Nature Mater. **7**, 455 (2008).

[11] A. Franceschetti and A. Zunger, Nature (London) **402**, 60 (1999).

[12] T. Mueller and G. Ceder, Phys. Rev. B **74**, 134104 (2006).

[13] A. Zunger, S. H. Wei, L. G. Ferreira, and J. E. Bernard, Phys. Rev. Lett. **65**, 353 (1990).

[14] A. van de Walle and G. Ceder, J. Phase Equilib. **23**, 348 (2002).

[15] A. Seko, Y. Koyama, and I. Tanaka, Phys. Rev. B **80**, 165122 (2009).

[16] J. M. Sanchez, F. Ducastelle, and D. Gratias, Physica **128A**, 334 (1984).

[17] Frequently the sigma character is used to represent site variables. We use "s" so as not to create confusion with the statistical concept of standard deviation.

[18] J. A. Rice, *Mathematical Statistics and Data Analysis* (Wadsworth, Belmont, California, 1995).

[19] T. Mueller and G. Ceder, Phys. Rev. B **80**, 024103 (2009).

[20] T. Mueller, Ph.D. thesis, Massachusetts Institute of Technology, 2007.

[21] G. L. W. Hart and R. W. Forcade, Phys. Rev. B **77**, 224115 (2008).

[22] J. M. Sanchez, Phys. Rev. B **81**, 224202 (2010).

[23] M. Asta, C. Wolverton, D. de Fontaine, and H. Dreyssé, Phys. Rev. B **44**, 4907 (1991).

[24] N. A. Zarkevich and D. D. Johnson, Phys. Rev. Lett. **92**, 255702 (2004).