Exact expressions for structure selection in cluster expansions

The MIT Faculty has made this article openly available. Please share how this access benefits you. Your story matters.

<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>As Published</td>
<td><a href="http://dx.doi.org/10.1103/PhysRevB.82.184107">http://dx.doi.org/10.1103/PhysRevB.82.184107</a></td>
</tr>
<tr>
<td>Publisher</td>
<td>American Physical Society</td>
</tr>
<tr>
<td>Version</td>
<td>Final published version</td>
</tr>
<tr>
<td>Accessed</td>
<td>Wed Mar 16 19:04:36 EDT 2016</td>
</tr>
<tr>
<td>Citable Link</td>
<td><a href="http://hdl.handle.net/1721.1/63804">http://hdl.handle.net/1721.1/63804</a></td>
</tr>
<tr>
<td>Terms of Use</td>
<td>Article is made available in accordance with the publisher's policy and may be subject to US copyright law. Please refer to the publisher's site for terms of use.</td>
</tr>
<tr>
<td>Detailed Terms</td>
<td></td>
</tr>
</tbody>
</table>
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 this value.14 More specifically, 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,

\[
\Theta_0(s_j) = 1,
\]

\[
\Theta_1(s_j) = \cos(\pi s_j) = \pm 1.
\]

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 \(b\),

\[
\Phi_b(s) = \prod_j \Theta_b(s_j),
\]

where \(s\) is the set of all site variables and \(b_j\) and \(s_j\) are the \(j\)th elements of \(b\) and \(s\) respectively. In general, \(\Theta_0\) 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,

\[
\sum_{j=1}^{N_{s_i}} \Phi_{b_i}(s_i) \Phi_{b_j}(s_j) = \delta_{bb'},
\]

where \(s_i\) is the \(i\)th set of possible values for the site variables and the sum is over all \(N_s\) such sets.

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

\[
F(s) = \sum_{i=1}^{N_s} F_i \Phi_i(s).
\]
\[ F(s) = \sum_b V_b \Phi_b(s), \]  

where \( V_b \) 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,

\[ F(s) = \sum_{\alpha} \sum_{b \in \alpha} \Phi_b(s), \]  

where \( \alpha \) 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

\[ f(s) = \sum_{\alpha} V_{\alpha} m_{\alpha} \varphi_{\alpha}(s), \]  

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

\[ \sum_{b \in \alpha} \Phi_b(s) = \sum_{b \in \alpha} \frac{\varphi_b(s)}{N_{\alpha}}. \]  

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 \( X \) in which

\[ X_{\alpha i} = \varphi_{\alpha}(s_i), \]  

where \( s_i \) is the set of site variables for the \( i \)th element of the training set. The ECI may then be estimated using a least-squares fit,

\[ \hat{\psi} = (X^T X)^{-1} X^T y, \]  

where \( \hat{\psi} \) is a column vector in which the \( \alpha \)th element is the predicted value for \( m_{\alpha} V_{\alpha} \), and \( y \) is a column vector in which \( y_i \) is the value for the property of interest for the \( i \)th 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 \( s \) is given by \( \langle (xM^t)_{\alpha i} \rangle \) \[ M = (X^T X)^{-1}\sigma^2. \]  

Although \( \sigma^2 \) 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 \( X \) 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, \( \psi \) the predicted ECI is given by

\[ \psi = (X^T X + A)^{-1} X^T y, \]  

where \( A = (X^T X + A)^{-1}\sigma^2 \) 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. It can be shown that the covariance matrix for the predicted ECI for a Bayesian cluster expansion is

\[ \hat{\psi} = (X^T X + A)^{-1} X^T y, \]  

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

\[ \langle x M^t \rangle_{\text{pop}} = M \langle x^t x \rangle_{\text{pop}}, \]  

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

\[ A : B = \sum_i \sum_j A_{ij} B_{ij}. \]  

We will call the matrix \( (xM^t)_{\text{pop}} \) the domain matrix and represent it with the symbol \( D \). As long as the domain matrix is known, the dimensionless quantity

\[ \frac{M : D}{\sigma^2} \]  

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 \( \Lambda \) to represent the expression in Eq. (19), but to avoid confusion with the symbols used in the formalism of the Bayesian cluster expansion, we will use the symbol \( \tau \). For a least-squares fit,

\[ \tau = (X^T X)^{-1} : D \]  

and for a Bayesian cluster expansion,
\[ \tau = (X^T X + \Lambda)^{-1} D, \]  
(21)

where once again the symbol : represents the Frobenius inner product. Attempts to find good training sets have focused on minimizing \( \tau \), using either approximate methods or statistical sampling \( ^{15} \) to calculate \( D \). 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

\[ D_{\alpha\beta} = \langle s, x_{\alpha\beta} \rangle_{\text{pop}} = \langle \varphi_{\alpha}(s) \varphi_{\beta}(s) \rangle_{\text{pop}} + \sum_{b=\alpha} \sum_{b'=\beta} \langle \Phi_b(s) \Phi_{b'}(s) \rangle_{\text{pop}}. \]

where once again \( \langle \cdot \rangle_{\text{pop}} \) represents the average over a population of structures, and \( N_\alpha \) and \( N_\beta \) are the number of cluster functions in orbits \( \alpha \) and \( \beta \), 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

\[ D_{\alpha\beta} = \frac{\sum_{b=\alpha} \sum_{b'=\beta} \delta_{bb'}}{N_\alpha N_\beta} = \frac{\delta_{\alpha\beta}}{N_\alpha}. \]

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 \( b \) that represent symmetrically equivalent cluster functions. For these cluster functions, \( N_\alpha \) 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 \( b \) that represents the empty cluster function, even in an infinite crystal. For the empty cluster function is assigned the index \( \alpha = 0 \), and for an infinite crystal \( D_{\alpha0} \) is typically the only nonzero element,

\[ D = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{bmatrix}. \]

Equations (17) and (24) yield the result that the average variance for the predicted property values over all structures is simply \( M_{\alpha0} \).

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

\[ \langle \Phi_b(s) \Phi_{b'}(s) \rangle_{\text{pop}} = \langle \Phi_b(s) \rangle_{\text{pop}} \langle \Phi_{b'}(s) \rangle_{\text{pop}}. \]

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

\[ D_{\alpha\beta} = \sum_{b=\alpha} \sum_{b'=\beta} \langle \Phi_b(s) \rangle_{\text{pop}} \langle \Phi_{b'}(s) \rangle_{\text{pop}} \frac{N_\alpha N_\beta}{N_\alpha N_\beta} = \langle x_\alpha \rangle_{\text{pop}} \langle x_\beta \rangle_{\text{pop}} \]

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

\[ \langle x M x^T \rangle_{\text{pop}} = \langle x \rangle_{\text{pop}} M \langle x \rangle_{\text{pop}}. \]

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 \( \Theta_1(s) \) are +1 and −1. If \( c \) is the concentration of the element assigned a value of +1, then the expected value of \( \Theta_1(s) \) over all structures with concentration \( c \) is \((2c−1)/2\). If a cluster expansion is to be applied to the population of all structures with concentration \( c \), the elements of the domain matrix are

\[ D_{\alpha\beta} = (2c - 1)^{n_\alpha n_\beta}, \]

where \( n_\alpha \) and \( n_\beta \) represent the number of sites upon which the cluster functions in orbits \( \alpha \) and \( \beta \) 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

\[ D_{\alpha\beta} = \int_{c=0}^1 (2c - 1)^{n_\alpha n_\beta} dc = \begin{cases} 1 & \text{if } (n_\alpha + n_\beta + 1) \text{ is even} \\ 0 & \text{if } (n_\alpha + n_\beta) \text{ is odd} \end{cases} \]

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 \( \tau \). 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-
the density of structures in the training set is the highest, and Fig. 1 composition. The results of these calculations are given in these training sets, it is possible to use Eq. 28-structure training set containing 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 $\tau$. 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 $\tau$ for the population of structures at a given composition. The results of these calculations are given in Fig. 1(a). In general, $\tau$ is lowest at the compositions at which the density of structures in the training set is the highest, and $\tau$ 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 only symmetrically distinct structures are included in the training set but the value of $\tau$ 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.

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. Seko et al. 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, each containing up to five atoms per unit cell, that are similar to the case in which only symmetrically distinct structures are included in the training set but the value of $\tau$ 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.

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)].
minimizes the value of $\tau$. The resulting minimal values of $\tau$ for the four different domain matrices are 7.303, 0.274, 0.101, and 0.301, respectively. The high value for $\tau$ 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 $\tau$ 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 $\tau$ 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 $\tau$ for all compositions. Although the value of $\tau$ 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.\textsuperscript{15} manually construct an initial set of training structures and then add structures that minimize $\tau$ 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 $\tau$ 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\textsuperscript{22} that this problem may be alleviated by using a concentration-dependent basis such as the one proposed by Asta et al.\textsuperscript{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.\textsuperscript{24} If all such cluster functions are included, the transformation to a concentration-dependent basis can be accomplished using a linear operator $A$.\textsuperscript{22}

$$x' = xA,$$

$$X' = XA,$$  \hspace{1cm} (30)

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

$$\tau' = (x'(X'TX')^{-1}x'T)_{pop} = (x(A'TXA)^{-1}A'Tx)_{pop} = \tau.$$  \hspace{1cm} (31)

A similar result holds for Bayesian cluster expansions where $A' = A'TAA$. Thus the value of $\tau$ 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.\textsuperscript{14} More accurate domain matrices yield better estimates of the variance of predicted property values, enabling

FIG. 3. (Color online) The effect of initial training set selection on the value of $\tau$ 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.
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.