PHYSICAL REVIEW B **80**, 024103 (2009)

Bayesian approach to cluster expansions

Tim Mueller and Gerbrand Ceder*
Massachusetts Institute of Technology, 77 Massachusetts Avenue, Building 13-5056, Cambridge, Massachusetts 02139, USA
(Received 30 December 2008; revised manuscript received 17 April 2009; published 2 July 2009)

Cluster expansions have proven to be a valuable tool in alloy theory and other problems in materials science but the generation of cluster expansions can be a computationally expensive and time-consuming process. We present a Bayesian framework for developing cluster expansions that explicitly incorporates physical insight into the fitting procedure. We demonstrate how existing methods fit within this framework and use the framework to develop methods that significantly improve the predictive power of cluster expansions for a given training set size. The key to the methods is to apply physical insight and cross validation to develop physically meaningful prior probability distributions for the cluster expansion coefficients. We use the Bayesian approach to develop an efficient method for generating cluster expansions for low-symmetry systems such as surfaces and nanoparticles.

DOI: 10.1103/PhysRevB.80.024103 PACS number(s): 61.50.Ah, 61.46.−w, 02.90.+p

I. INTRODUCTION

Some properties of crystalline materials can be modeled as a function of variables assigned to a fixed set of sites. For example, magnetic energy can be expressed as a function of the spin states of atoms at a given set of sites[1] and in systems exhibiting substitutional disorder the energy of an arrangement of atoms can be modeled as a function of variables that indicate which species (or vacancy) occupies each site.[2] Often the exact function of site variables that produces the property value is unknown and must be parameterized. Such estimations may be accomplished through the use of cluster expansions.[3]

In a cluster expansion, the function that produces the property value is expanded as a linear combination of basis functions known as cluster functions.[3] Most often these cluster functions are products of single-site descriptors and their expansion coefficient represents an interaction between the sites in the cluster. The coefficients of the expansion are typically estimated from a combination of physical insight and training data. The resulting cluster expansion is capable of calculating property values very quickly, because it is a simple analytical expression, and accurately because an arbitrarily large number of basis functions may be included in the expansion.

The speed and accuracy of the cluster expansion have made it a popular tool in materials science. The Ising model, a simple cluster expansion, is commonly used to study magnetic properties.[4] For systems of substitutional disorder, cluster expansion Hamiltonians have been widely used to identify ground states,[5–12] calculate phase diagrams,[5–9] and study ordering.[10–13] In addition, cluster expansions have been used to model kinetic activation energies,[14] tensor properties,[15] band gaps,[16] orientations of complex ions,[17] amino acid sequences in proteins,[18] and configurational electronic entropy.[19]

The challenge in working with cluster expansions is that a new set of coefficients must be estimated for each system. Generating the required training data can be computationally expensive and it can be difficult to determine how accurately a given cluster expansion predicts a property value. For low-symmetry systems, such as nanoparticles, the number of symmetrically distinct cluster functions that must be included can be large and estimating such a large number of coefficient values typically requires a large set of training data. Compounding the problem, it is often computationally expensive to generate each training data point for low-symmetry systems. The result is that the generation of cluster expansions for low-symmetry systems can be prohibitively expensive.

In this paper we address the above problems by treating the estimation of cluster expansion coefficients as an exercise in statistical function learning. By applying Bayes' theorem,[20] we develop a framework for using physical insights to generate cluster expansion coefficients and demonstrate how existing methods for generating cluster expansions fit within the framework. We use our framework to develop methods for estimating cluster expansion coefficients and demonstrate that on a set of test systems the methods outperform several common methods. Finally, we will demonstrate that our framework may be used to develop cluster expansions for nanoparticles and other low-symmetry systems with a high level of accuracy.

II. CLUSTER EXPANSION OVERVIEW

We will briefly review the mathematical foundation of the cluster expansion, basing the review on the derivation by Sanchez *et al.*[3] The notation introduced in this section will be used throughout the paper.

In a cluster expansion, the value of a property is represented as a function of variables assigned to a set of fixed sites. The variables assigned to a given site represent the state of the site. For example, a site state might be the magnetic spin associated with a site or the chemical species present at a site. For clarity, in this paper we will only consider cases in which there is one variable per site, although extending to multiple variables is straightforward. We will denote the *site variable* for the jth site by s_j and the set of all such site variables by $\vec{s}$. At each site, a single-variable *site basis* of functions is defined. We will use $\Theta_{b,j}$ to represent

the bth basis function for the jth site. The tensor product of all such site bases creates a complete multivariable basis. An extensive material property, F, can be written as a linear combination of these basis functions,

$$F(\vec{s}) = \sum_{\vec{b}} V_{\vec{b}} \prod_{j} \Theta_{b_j,j}(s_j), \tag{1}$$

where b_j, the jth element of $\vec{b}$, is the index of the basis function to be used at site j. The sum is over all possible sets $\vec{b}$. The coefficients for this expansion, $V_{\vec{b}}$, are referred to as effective cluster interactions (ECIs). The basis function $\prod_j \Theta_{b_j,j}(s_j)$ is known as a cluster function and will be denoted by

$$\Phi_{\vec{b}}(\vec{s}) = \prod_{j} \Theta_{b_j,j}(s_j). \tag{2}$$

The number of distinct ECI can be reduced by exploiting symmetry. If two cluster functions are symmetrically equivalent to each other, their corresponding ECI must be equal. Equation (1) can thus be rewritten as

$$F(\vec{s}) = \sum_{\alpha} V_{\alpha} \sum_{\vec{b} \in \alpha} \Phi_{\vec{b}}(\vec{s}), \tag{3}$$

where the set α represents an orbit of symmetrically equivalent cluster functions and the outer sum is over all such orbits.

The value of an extensive property may be normalized per unit of material. For such a property, Eq. (4) can be rewritten as

$$\langle F(\vec{s}) \rangle = \sum_{\alpha} V_{\alpha} m_{\alpha} \langle \Phi_{\vec{b}}(\vec{s}) \rangle_{\alpha}, \tag{4}$$

where $\langle F(\vec{s}) \rangle$ is the average value of the property per formula unit and $\langle \Phi \rangle_{\alpha}$ is the average value of cluster functions in orbit α. The multiplicity, m_{α}, is an integer that represents the number of cluster functions in orbit α per formula unit. If all the ECI were known, Eq. (4) could be used to exactly calculate the normalized property value for a given material state.

For an infinite crystal, Eq. (4) is an infinite sum and cannot be evaluated. This problem can be addressed through truncation of the cluster expansion. The first step in the truncation procedure is to define the single-site bases so that one of the single-site functions is the constant, 1. We will always use the index "0" for the constant function, so that $\Theta_{0,j}$ always equals 1. Equation (2) can therefore be written as

$$\Phi_{\vec{b}}(\vec{s}) = \prod_{j|b_j \neq 0} \Theta_{b_j,j}(s_j). \tag{5}$$

For some cluster functions, Eq. (5) is a product over a finite number of site functions. A commonly applied insight is that the more site functions in a cluster function and the further the distance between the corresponding sites, the smaller the ECI should be.[3] Often, all but a finite number of cluster function orbits will have ECI that are negligible and may be estimated to be zero. This approximation allows for the expansion to be reduced to a finite number of nonzero terms with a typically small loss of accuracy. In the next section we will show how this insight can be more explicitly included in the ECI learning process.

III. BAYESIAN CLUSTER EXPANSION

To construct a truncated cluster expansion, the objective is to find a set of ECI values, $\vec{V}$, that best reproduces the property values. We refer to these values as the optimal ECI. In this work, we seek to find ECI values that are most likely to be optimal given a set of training data. We express the training data as a vector of output values, $\vec{y}$, and matrix of input values, X. The ith element of $\vec{y}$ is the property value for the ith element in the training set and the elements of X are given by $X_{i\alpha} = m_{\alpha} \langle \Phi(\vec{s}_i) \rangle_{\alpha}$. To ensure X has a finite number of columns, cluster functions for which the ECI are likely to be negligible are excluded. The probability density for the optimal ECI, given the training data, is expressed as the conditional probability distribution $P(\vec{v}|X,\vec{y})$, where the variable $\vec{v}$ is defined over possible ECI values.

The key to our approach is to use Bayes' theorem, in conjunction with physical insight, to find the ECI that maximize $P(\vec{v}|X,\vec{y})$. Bayes' theorem, described in more detail in the Appendix, tells us

$$P(\vec{v}|X,\vec{y}) = \frac{P(\vec{y}|\vec{v},X)P(\vec{v}|X)}{P(\vec{y}|X)}. \tag{6}$$

The key to the application of Bayes' theorem is the establishment of the *prior* probability distribution, $P(\vec{v}|X)$. The prior probability distribution represents an educated guess of the likelihood of ECI values before we have calculated property values for the training data. Through the prior probability distribution we may incorporate physical insight into the nature of the ECI. For example, if we are considering formation energies, we may have the *a priori* belief that ECI on the order of meV are more likely than those on the order of keV, given typical values of formation energy.[21] We consider here three examples of such prior insights:

Insight 1: property predictions should be close to those predicted by some simple model. For example, we might expect the energy of an alloy to be close to the weighted average of the energies of the pure substances, or we might expect the energy to be close to that predicted by an empirical potential model. We treat such a model the mean of the prior probability distribution of property values. We define $y_{\Delta} = y - y_{\mu}$, where y is the property value, y_{μ} is the expected value of y according to the simple model, and y_{Δ} is the unknown difference. We can calculate y by using the simple model to calculate y_{μ} and a cluster expansion to calculate y_{Δ}. By definition the prior expected value for y_{Δ} is zero and therefore the prior expected value for each ECI in the cluster expansion of y_{Δ} must also be zero. For this reason, it is convenient to develop a cluster expansion for the transformed property values, y_{Δ}, in place of y. For simplicity, from this point forward we will assume that the variable $\vec{y}$ represents the transformed property values, and consequently the mean of the prior distribution for the ECI is zero.

Insight 2: the greater the number of sites in the cluster, and the greater the distance between sites, the smaller the

ECI should be. This insight is commonly used to truncate the cluster expansion but it may also be applied to cluster functions included in the fit. We define a prior probability distribution for V_α, the αth element of $\vec{V}$ with mean zero and variance σ_α^2. The variance is the expected squared value of the ECI and in the limit $\sigma_\alpha^2 \to 0$ the cluster function is effectively excluded from the expansion. The probability distribution with zero mean and finite variance that maintains the maximum information entropy (i.e., makes the fewest assumptions about the data) is the Gaussian,[22] making it a reasonable choice for our prior distribution. Using Gaussian distributions, the prior probability $P(\vec{v}|X)$ can be expressed as

$$P(\vec{v}|X) \propto \prod_\alpha e^{-{v_\alpha}^2/2{\sigma_\alpha}^2}, \tag{7}$$

where the product is over all included cluster orbits. To build in the expectation that ECI are smaller for larger clusters, σ_α^2 should be a decreasing function of the number of sites in a cluster and the distance between sites.

Insight 3: ECI for similar cluster functions should have similar values. Some clusters of sites that are not symmetrically equivalent may however be similar to each other. For example, a nearest-neighbor pair of sites two layers below a crystal surface is symmetrically distinct from to a nearest-neighbor pair of sites three layers below the surface but both clusters have the same number of sites, the same distance between sites, and the same nearest-neighbor environments. Previous cluster expansions for surfaces have not made use of such similarities.[23–27] The belief that similar clusters should have ECI that are close to each other can be expressed in the prior distribution by setting

$$P(\vec{v}|X) \propto \prod_{\alpha,\beta\neq\alpha} e^{-(v_\alpha - v_\beta)^2/2{\sigma_{\alpha\beta}}^2}, \tag{8}$$

where the product is over all pairs of ECI for similar cluster functions. The variance $\sigma_{\alpha\beta}^2$ indicates the degree of expected similarity between the two ECI. In the limit $\sigma_{\alpha\beta} \to 0$, the ECI for α and β are forced to be identical and in the limit $\sigma_{\alpha\beta} \to \infty$, the ECI are completely decoupled.

In the Appendix we show how the above insights can be combined with Bayes' theorem to derive a maximum likelihood estimate for $\vec{V}$,

$$\hat{\vec{V}} = (X^T W X + \Lambda)^{-1} X^T W \vec{y}, \tag{9}$$

where $\hat{\vec{V}}$ are the estimated ECI, W is a diagonal weight matrix, and Λ is a matrix with elements given by

$$\Lambda_{\alpha\alpha} = \frac{\sigma^2}{\sigma_\alpha^2} + \sum_{\beta|\beta\neq\alpha} \frac{\sigma^2}{\sigma_{\alpha\beta}^2},$$

$$\Lambda_{\alpha\beta} = \Lambda_{\beta\alpha} = \frac{-\sigma^2}{\sigma_{\alpha\beta}^2}, \tag{10}$$

where σ^2 is an unknown constant. We can rewrite Eq. (9) by defining the weighted input matrix and weighted output vector,

$$X_W = W^{1/2} X,$$

$$\vec{y}_W = W^{1/2} \vec{y} \tag{11}$$

to arrive at

$$\hat{\vec{V}} = (X_W^T X_W + \Lambda)^{-1} X_W^T \vec{y}_W. \tag{12}$$

Because of the relationship to Tikhonov[28] regularization, Λ will be referred to as the regularization matrix. Without this matrix, Eq. (12) is just the solution to a standard linear least-squares fit. The regularization matrix is equivalent to using the following prior probability distribution:

$$P(\vec{v}|X) \propto e^{-\vec{v}^T \Lambda \vec{v}/2}, \tag{13}$$

which is a multivariate normal distribution with covariance matrix Λ^{-1}. Thus we can see that Λ may be thought of as the inverse of the covariance matrix for the prior probability distribution of the ECI.

For convenience, we define the orbit regularization parameter $\lambda_\alpha = \frac{\sigma^2}{\sigma_\alpha^2}$ and coupled regularization parameter $\lambda_{\alpha\beta} = \lambda_{\beta\alpha} = \frac{\sigma^2}{\sigma_{\alpha\beta}^2}$. The matrix Λ can be written entirely in terms of these parameters. To better understand the regularization matrix, it is instructive to consider what happens in the limits of the parameters:

(1) $\lambda_\alpha \to \infty$. V_α is forced to be zero and effectively eliminated from the fit.

(2) $\lambda_\alpha \to 0$ and $\lambda_{\alpha\beta} \to 0$. All values for V_α are considered equally likely.

(3) $\lambda_{\alpha\beta} \to 0$. There is no expectation that V_α and V_β are close to one another.

(4) $\lambda_{\alpha\beta} \to \infty$. V_α and V_β are forced to have the same value.

Values for the regularization parameters λ_α and $\lambda_{\alpha\beta}$ may be assigned manually, based on physical expectations. However, it is generally more convenient to have a method to automatically determine reasonable values for λ_α and $\lambda_{\alpha\beta}$. In the next sections, we will provide examples of different automated methods for assigning values to λ_α and $\lambda_{\alpha\beta}$. These methods will then be evaluated against sets of test data to determine which produce the best estimates for the ECI.

A. Methods for generating λ_α

If we restrict $\lambda_{\alpha\beta}=0$ and $\lambda_\alpha \in \{0,\infty\}$, then assigning values to the regularization parameters is equivalent to cluster selection, in which certain cluster functions are selected to be included in the cluster expansion and a least-squares fit is used to estimate the values of the ECI for those cluster functions. The choice of which clusters should be included in the cluster expansion is commonly accomplished using a cross-validation (CV) method.[29] In cross-validation methods, a subset of the training data is set aside and the ECI are estimated using the remaining training data. The estimated ECI are used to predict values for the training data that were set aside and then the process is repeated with another subset of training data set aside. After repeating this process many times, the root-mean-square error of all predictions, known as the cross-validation score, provides an estimate of the prediction error of the cluster expansion. In the context of cluster selection, the cross-validation score is calculated for a number of different sets of included cluster functions and the

set of cluster functions that has the lowest cross-validation score is selected.

The regularization parameters need not be restricted to $\lambda_{\alpha\beta}=0$ and $\lambda_\alpha \in \{0,\infty\}$ but they may still be chosen using cross validation. Directly finding the combination of regularization parameters that minimizes the cross-validation score is one option but this option ignores the physical insights that one may have, such as insights 2 and 3 described above. Alternatively, we may use *regularization functions* to generate the regularization parameters in a way that is consistent with our physical insights. We define regularization functions as functions that generate values for the regularization parameters λ_α and $\lambda_{\alpha\beta}$ given a set of *generating parameters* which we will label γ_i. The optimal values for the generating parameters will be determined using cross validation and these values will be used with the regularization function to generate values for λ_α and $\lambda_{\alpha\beta}$. The extra level of abstraction provided by the regularization functions allows us to incorporate physical insights into the process of generating regularization parameters or ignore our physical insights if we so choose.

We will now describe five sample regularization functions that generate the regularization parameter λ_α. For each of these functions we will set $\lambda_{\vec{0}}$, the regularization variable for the constant cluster function, equal to zero for simplicity. However this choice is not necessary in general.

(a) *Cluster selection.* This regularization function corresponds to the cluster selection method described above. There is one binary generating parameter γ_α for each orbit and the regularization function is

$$\lambda_\alpha = \begin{cases} 0 | \gamma_\alpha = 0 \\ \infty | \gamma_\alpha = 1 \end{cases}.$$

In addition, we enforce a rule that if a cluster is included in the fit, all subclusters must be included as well, consistent with the rules proposed by Zarkevich and Johnson.[30] It was found that this constraint significantly improves the quality of the fit.

(b) *Single width.* This function is equivalent to saying that the expected magnitudes of all ECI are the same. This magnitude is determined by one non-negative continuous input parameter γ and the regularization function is $\lambda_\alpha=\gamma$.

(c) *Independent widths.* In this regularization function, the prior expectations for the magnitude of the ECI are all independent of each other and the expected magnitudes may take on any non-negative values. There is one non-negative continuous input parameter γ_α for each orbit and the regularization function is $\lambda_\alpha=\gamma_\alpha$.

(d) *Inverse cluster density.* Up to this point, we have not explicitly considered the physical expectation that for clusters with a larger number of sites, or clusters with larger distance between sites, the corresponding ECI should be small. To derive a regularization function for this expectation, we consider here the case in which the site variables may take on discrete values. We will also assume that the basis has been constructed so that the cluster functions are orthonormal, where orthonormality is defined as

$$\forall j, \quad \frac{\sum_{i=1}^{N_j} \Theta_{a,j}(s_{j,i})\Theta_{b,j}(s_{j,i})}{N_j} = \delta_{ab}, \tag{14}$$

where N_j is the number of possible values for the site variable for the jth site and $s_{j,i}$ is the ith value for the jth site variable.

The insight that the material property may be adequately represented by a finite set of small, compact clusters may be mathematically expressed as an expectation that the cluster expansion may converge. Here we define convergence by

$$\forall \varepsilon > 0, \ \exists \{n_{\text{cut}}, r_{\text{cut}}\} \text{ such that } \left\langle E\left\{\left[\sum_{\vec{b} \notin B_{\text{cut}}} V_{\vec{b}} \Phi_{\vec{b}}(\vec{s})\right]^2\right\}\right\rangle < \varepsilon, \tag{15}$$

where the function $E(\cdot)$ is the expectation over all structures and B_{cut} is the set of all cluster functions dependent on clusters of no more than n_{cut} sites that are a distance of no more than r_{cut} from each other. Once again we use $\langle\cdot\rangle$ to represent normalization per formula unit. Combining Eqs. (14) and (15) with the fact that $E(V_{\vec{b}})=0$, we see that the convergence condition can be written as

$$\forall \varepsilon > 0, \ \exists \{n_{\text{cut}}, r_{\text{cut}}\} \text{ such that } \left\langle \sum_{\vec{b} \notin B_{\text{cut}}} \sigma_{\vec{b}}^2 \right\rangle < \varepsilon, \tag{16}$$

where $\sigma_{\vec{b}}^2$ is the variance of the prior distribution for $V_{\vec{b}}$.

The condition in Eq. (16) can be met in the limit of $n_{\text{cut}} \to \infty$ and $r_{\text{cut}} \to \infty$ if $\sigma_{\vec{b}}^2$ decreases more rapidly than the number of clusters in B_{cut} increases. As $r_{\text{cut}} \to \infty$ the number of clusters in B_{cut} per formula unit is approximately proportional to $(\gamma_1 r_{\text{cut}})^{\gamma_2 n_{\text{cut}}}$, where γ_1 is a scale factor and γ_2 depends on the number of periodic dimensions. Therefore, the condition in Eq. (16) can be met if, for a cluster with n sites and a maximum distance of r between sites,

$$\sigma(n,r) = (\gamma_1 r)^{-\gamma_2 n} \tag{17}$$

for some non-negative γ_1 and γ_2. More generally, we use the following regularization function, in which all parameters are non-negative:

$$\lambda_\alpha(n,r) = \gamma_1(\gamma_2 r + \gamma_3 + 1)^{\gamma_4 n + \gamma_5}. \tag{18}$$

For some parameter values, the prior distribution will not converge as defined by Eq. (16). However, for cluster expansions that do converge, some set of parameters can be used in Eq. (18) to create a convergent prior distribution.

(e) *All pair distances.* We might want to consider not just the maximum distance between sites but all distances between all sites in the cluster. Below is an example of a function that considers all sites between all clusters and may still converge

$$\lambda_\alpha = \begin{cases} \gamma_3, & \vec{b} \in \alpha, |\vec{b}| = 1 \\ \left\{ \sum_{i|b_i \neq 0} \left[\gamma_i \prod_{j|b_j \neq 0, j \neq i} (1 + \gamma_1 r_{i,j}) \right]^{1/\gamma_2} \right\}^{-\gamma_2}, & \vec{b} \in \alpha, |\vec{b}| > 1 \end{cases}, \tag{19}$$

where i and j are sites, $|\vec{b}|$ is the number of nonzero entries in $\vec{b}$, and $r_{i,j}$ is the distance between sites i and j. The nonnegative parameters γ_i are site dependent and all symmetrically equivalent sites share the same γ_i. The parameter γ_3 is used to treat point clusters specially, although we note that if this parameter is omitted then $\lambda_\alpha = \gamma_i$ for point clusters, creating a similar regularization function with fewer parameters.

B. Methods for generating $\lambda_{\alpha\beta}$

In this section we will consider regularization functions that generate $\lambda_{\alpha\beta}$, which determines the degree to which we expect the ECI for different cluster functions to be similar. These functions will be constructed in a manner similar to how we constructed the regularization functions for λ_α: we will define parameterized functions and use cross validation to determine the parameters.

Methods for generating $\lambda_{\alpha\beta}$ may be closely related to the geometry of a particular problem. Many regularization functions for $\lambda_{\alpha\beta}$ will depend on the concept of "congruent" cluster functions, which we define as cluster functions for which the nonconstant site functions (and underlying sites) are related by an isometry. In congruent clusters, there must be a way to map the sites of one cluster onto the sites of another that preserves all distances and angles between sites. For example, when considering a binary cluster expansion of the surface of an FCC material, all nearest-neighbor pair interactions are congruent to each other, although they are not necessarily symmetrically equivalent. Further examples are given in Fig. 1. We will use the symbol $\cong$ to represent congruency. In general, each of the regularization functions described in this section is a method for determining to what extent congruent cluster functions should be "coupled," or how close we expect their ECI to be, to each other.

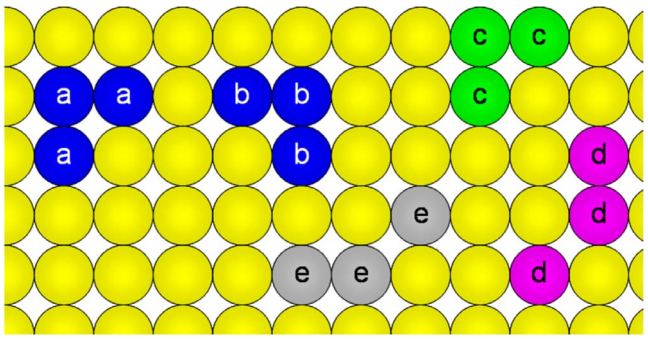

FIG. 1. (Color online) An illustration of symmetric and congruent clusters near a surface on a square lattice. The cluster of sites marked with "a" is symmetrically equivalent to the cluster marked with "b" but it is not equivalent to the cluster marked with "c." Clusters "a" and "c" are congruent, as are "b" and "c." Clusters "d" and "e" are also congruent but not symmetrically equivalent. Clusters "a," "b," and "c" are neither congruent nor symmetrically equivalent to clusters "d" and "e."

We will consider six example regularization functions for $\lambda_{\alpha\beta}$. In each of these methods, $\lambda_{\alpha\beta}=0$ for clusters which are not congruent. It is assumed that the regularization function for λ_α is defined such that if $\alpha \cong \beta$ then $\lambda_\alpha = \lambda_\beta$. To separate the concept of similarity from the expected magnitude of the ECI, each regularization function for $\lambda_{\alpha\beta}$ will first generate a similarity factor, γ_s, which is related to $\lambda_{\alpha\beta}$ as follows:

$$\lambda_{\alpha\beta} = \gamma_s \lambda_\alpha = \gamma_s \lambda_\beta. \tag{20}$$

(a) *No coupling.* This regularization function, $\gamma_s = \lambda_{\alpha\beta} = 0$, has been widely (implicitly) used in cluster expansions. It is appropriate for cases in which any two cluster functions are either symmetrically equivalent or completely distinct.

(b) *Coarse graining.* This regularization function should be used when all congruent cluster functions are to be treated as symmetrically equivalent. The function is defined by $\gamma_s = \infty$ for congruent cluster functions and for all others $\gamma_s = 0$. For example, this approach is similar to the one used in Ref. 23.

(c) *Coarse graining except surface.* This is the same as coarse graining, except cluster functions that are dependent on surface sites are not coupled to congruent cluster functions. The intuition behind the use of this regularization function is that even if congruent bulk clusters may be treated as symmetrically equivalent, it makes less sense to treat a cluster function dependent on surface sites as equivalent to a cluster function dependent on only nonsurface sites. The function is defined by $\gamma_s = \infty$ for congruent cluster functions that do not include surface sites and $\gamma_s = 0$ for all others. An approach similar to this one was used in Ref. 25.

(d) *Uniform coupling.* This function is different from the coarse graining approaches in that the ECI for congruent cluster functions are expected to be close to each other but not necessarily identical. There is one input parameter, γ, which is allowed to have any non-negative value. For congruent cluster functions, $\gamma_s = \gamma$, and for all others $\gamma_s = 0$.

(e) *Similarity to bulk.* If there is no significant surface reconstruction, the cluster functions in orbit α near a surface are congruent to the cluster functions in some orbit α_{Bulk}, defined in an infinite unbroken crystal. In this method, all congruent cluster functions are indirectly coupled through their similarity to the corresponding bulk cluster function. To accomplish this, we assume that V_α is normally distributed around $V_{\alpha_{\text{Bulk}}}$

$$V_\alpha \sim N(V_{\alpha_{\text{Bulk}}}, \sigma^2_{\alpha,\text{Bulk}}), \tag{21}$$

where $N(V_{\alpha_{\text{Bulk}}}, \sigma^2_{\alpha,\text{Bulk}})$ is a normal distribution with mean $V_{\alpha_{\text{Bulk}}}$ and variance $\sigma^2_{\alpha,\text{Bulk}}$. There are a number of ways to

generate $\sigma_{\alpha,\mathrm{Bulk}}$ and we will use the following:

$$\sigma^2_{\alpha,\mathrm{Bulk}} = ae^{-br_\alpha}, \tag{22}$$

where the parameters a and b are scale factors and r_α is the shortest distance between a site in the cluster α and a site on the surface. With this form, the further a cluster is from the surface, the closer its ECI should be to the bulk ECI.

Now consider another cluster function from orbit α' that is congruent to the cluster functions in α. Because these cluster functions are congruent, they must be coupled to the same orbit of bulk cluster functions (i.e., $\alpha'_{\mathrm{Bulk}} = \alpha_{\mathrm{Bulk}}$). For example, in an fcc crystal α might be a nearest-neighbor pair two layers from the surface and α' might be a nearest-neighbor pair three layers from the surface. From Eq. (21), we get

$$V_\alpha - V_{\alpha'} \sim N(0, \sigma^2_{\alpha,\mathrm{Bulk}} + \sigma^2_{\alpha',\mathrm{Bulk}}). \tag{23}$$

Thus, in general the regularization function becomes

$$\gamma_s = \frac{\gamma_1}{e^{-\gamma_2 r_\alpha} + e^{-\gamma_2 r_\beta}} \tag{24}$$

when $\alpha \cong \beta$ and $\gamma_s = 0$ otherwise.

(f) *Local environment similarity.* This is the most complicated approach, in which the similarity between any two congruent cluster functions is determined as a function of the similarity of their local environments. As usual, we define $\gamma_s = 0$ for any two cluster functions that are not congruent. For congruent clusters, $\lambda_{\alpha\beta}$ is determined by a measure of similarity between the local environments of cluster functions in α and β. In this paper, we measure the general similarity by considering each transformation T that maps a cluster function in α to a cluster function in β, excluding constant site functions. We define the set $\mathrm{M}_{T,\alpha}$ as the set of indices of sites that are mapped by T onto sites with identical allowed states and single-site bases, including constant site functions. Likewise, $\mathrm{M}_{T,\beta}$ is defined for the inverse transformation. We define the overlap factor $\mathrm{O}_{\alpha,\beta}$ by

$$\mathrm{O}_{\alpha,\beta} = \max_T \left(\frac{\sum_{i \in M_{T,\alpha}} w(r_{i,\alpha}) + \sum_{i \in M_{T,\beta}} w(r_{i,\beta})}{\sum_i w(r_{i,\alpha}) + \sum_i w(r_{i,\beta})} \right),$$

$$w(r_{i,\alpha}) = (1 + \gamma_1)^{-\gamma_2 r_{i,\alpha}}, \tag{25}$$

where $r_{i,\alpha}$ is the minimum distance between site i (or a periodic image of site i) and a nonconstant site in the cluster function to which the transform was applied. The overlap factor is defined so that if α and β are symmetrically equivalent, $\mathrm{O}_{\alpha,\beta} = 1$. If, on the other hand, there is no overlap among sites between the original and transformed crystal, $\mathrm{O}_{\alpha,\beta} = 0$. From the overlap factor, we define the similarity factor as

$$\gamma_s = \frac{\mathrm{O}_{\alpha,\beta}^{1/\gamma_3}}{1 - \mathrm{O}_{\alpha,\beta}^{1/\gamma_3}}. \tag{26}$$

IV. APPLICATIONS

To evaluate the various methods for generating λ_α, test data were generated for three different binary material systems: Si-Ge, Ag-Au, and Pb-Ir. For the diamond-cubic Si-Ge system, energies were calculated for all structures with up to 14 atoms per unit cell, for a total of 9631 structures. For the fcc Ag-Au and Pb-Ir systems, energies were calculated for all structures with up to nine atoms per unit cell, for a total of 1135 structures each. To calculate the energies for Si-Ge, the Tersoff[31] potential was used. For Ag-Au and Pb-Ir, we used a Sutton and Chen[32] embedded atom potential,

$$E = \varepsilon_{ij} \left[\frac{1}{2} \sum_i \sum_{j \neq i} \left(\frac{a_{ij}}{r_{ij}} \right)^{n_{ij}} - \sum_i c_i \sqrt{\sum_{j \neq i} \left(\frac{a_{ij}}{r_{ij}} \right)^{m_j}} \right], \tag{27}$$

with the following combination rules:

$$a_{ij} = \frac{a_{ii} + a_{jj}}{2},$$

$$n_{ij} = \frac{n_{ii} + n_{jj}}{2},$$

$$\varepsilon_{ij} = \sqrt{\varepsilon_{ii}\varepsilon_{jj}}. \tag{28}$$

The lattice parameters for Au and Ag are very close to each other, resulting in little relaxation of atoms from ideal fcc lattice positions and generally well-converged cluster expansions. The lattice parameter for Pb is 29% larger than the lattice parameter for Ir and there was frequently significant relaxation for this system. No attempt was made to remove structures in which the initial and final atomic positions were significantly different. Hence the Pb-Ir data set is a good test of how well the different methods are able to fit a poorly converging cluster expansion. In contrast, the Si-Ge structures retained the diamond-cubic form and yielded cluster expansions with the lowest average prediction error.

To evaluate the methods, cluster expansions were generated for training sets of 15, 30, 45, 60, and 75 training structures. Four different sets of candidate clusters were generated. Each includes all two-site, three-site, and four-site cluster functions up to a specified cutoff. The four cutoffs considered are first nearest neighbor, second nearest neighbor, third nearest neighbor, and fourth nearest neighbor. The predictive power of the cluster expansion was tested on the complete set of sample structures.

Two different cross-validation methods were considered for parameter selection: leave-one-out cross validation (LOOCV) and generalized cross validation (GCV). Leave-one-out cross validation is the process of leaving one sample out of the training set, training the cluster expansion on the remaining samples, and measuring the predictive error on the sample left out. The LOO CV score is the mean-squared predictive error over all excluded samples and is given by

$$\mathrm{LOOCV} = \sum_i (\hat{y}_{W,\mathrm{CV},i} - y_{W,i})^2, \tag{29}$$

where $\hat{y}_{W,\mathrm{CV},i}$ is the value for the ith training sample as predicted by a cluster expansion fit to the remaining samples,

and $y_{W,i}$ is the ith element of $\vec{y}_{W,i}$. The sum is over all training samples.

The evaluation of Eq. (29) can be sped up by reducing the size of the matrix that needs to be inverted for each excluded sample. Using the Sherman-Morrison-Woodbury formula,[33] it can be shown that the prediction error for a given set of excluded samples is given by

$$\vec{\hat{y}}_{W,\mathrm{CV,out}} - \vec{y}_{W,\mathrm{out}} = (I - X_{W,\mathrm{out}}{}^{T}\Omega^{-1}X_{W,\mathrm{out}})^{-1}(\vec{\hat{y}}_{W,\mathrm{out}} - \vec{y}_{W,\mathrm{out}}), \tag{30}$$

where $X_{W,\mathrm{out}}$ are the rows of X_W corresponding to the excluded samples, $\vec{y}_{W,\mathrm{out}}$ is the equivalent for the output values, $\vec{\hat{y}}_{W,\mathrm{out}}$ are the predicted values for the excluded samples using a full fit, and $\vec{\hat{y}}_{W,\mathrm{CV,out}}$ are the predicted values for the excluded samples using a cluster expansion fit to only the excluded samples. The matrix Ω is given by

$$\Omega = (X_W{}^{T}X_W + \Lambda)^{-1}. \tag{31}$$

It can be shown that the leave-one-out cross-validation score is dependent on the basis used to represent the input data X. A widely used alternative that does not have this dependency is known as GCV.[34] It is equivalent to the leave-one-out cross-validation score for a system in which the input data have been rotated to a standard form. For systems in which the prior is Gaussian, the generalized cross-validation score is given by

$$\mathrm{GCV} = \frac{\sqrt{N\|\vec{\hat{y}}_T - \vec{y}_T\|^2}}{\mathrm{Tr}(I - X_W\Omega^{-1}X_W{}^{T})}, \tag{32}$$

where N is the number of training samples and Ω is given by Eq. (31). In addition to being basis independent, the GCV score has the advantage of being faster to compute than the LOO CV score.

Regularization matrices were generated by finding parameters that yield low cross-validation scores for the regularization function. For cluster selection in cluster expansions in which there were fewer than 30 candidate clusters, the set of clusters that minimized the cross-validation score was found by an exhaustive search of all possible sets of clusters to include in the fit. For cluster expansions with more than 30 candidate structures, simulated annealing was used to search for the ground state.

For the inverse-density, pair-distances, and single-value regularization functions, parameter selection was done in a two-stage process. The first stage of the process was a grid search for the locally minimum cross-validation score on a logarithmic grid, in which neighboring grid points represented parameters that differed by a factor of 2. All parameters were initialized with a value of 1. The grid search was ended when the improvement in the score between neighboring points was less than 0.1 meV. When the grid search was completed, a conjugate gradient algorithm was used to more finely resolve the local minimum. The conjugate gradient algorithm was stopped when the gradient of the score with respect to the natural log of the parameter values was less than 0.01 meV.

For the independent regularization function, the same method was used as the inverse-density, pair-distances, and single-value regularization functions if there were fewer than six parameters. For situations in which there were six or more parameters, the multidimensional grid search rapidly becomes computationally expensive. In such situations, the grid search was skipped and only the conjugate gradient step was used.

The comparison of the mean-squared predictive error for the different candidate cluster sets, training set sizes, cross-validation techniques, and data sets can be seen in Table I. In each data set, the generalized cross-validation score gives on average slightly better results that leave-one-out cross validation, especially for the noisy Pb-Ir data set. This may be due in part to less numerical noise when calculating the GCV score. Because this method is also faster than leave-one-out cross validation, it was used for the remainder of the results.

A. Evaluating methods for generating λ_α

The average prediction errors and cross-validation scores of the five different methods for generating λ_α are shown in Fig. 2. The inverse-density and pair-distances regularization functions consistently outperform the rest and are significantly better for the more challenging systems. The difference between the inverse-density and pair-distances regularization functions is minor, likely due to the fact that they have similar forms and the same number of parameters.

The single-value regularization function generally performs poorly but this poor performance is somewhat reflected in a high cross-validation score. The reason for this high score is likely due to the fact that the prior distribution in which all ECI are expected to have similar values is physically unrealistic.

The independent regularization function and cluster selection typically have the largest number of independent parameters and consistently produce the lowest cross-validation scores. However, the actual prediction error diverges significantly from the cross-validation score as the number of candidate clusters, and hence the number of parameters, increases. This is almost certainly due to over fitting of the cross-validation score; as the number of degrees of freedom increases, the cross-validation score becomes a poor estimate of prediction error. Thus parameters that produce cluster expansions with low cross-validation scores do not reliably produce cluster expansions with low prediction error.

B. Evaluating methods for generating $\lambda_{\alpha\beta}$

To evaluate methods for generating $\lambda_{\alpha\beta}$, cluster expansions were generated for a 201-atom cuboctahedral Ag-Au nanoparticles. A test set of 10 000 randomly chosen structures was generated and energies were calculated using the embedded atom method. Generalized cross validation was used to find the optimal set of parameters. Four candidate cluster sets were generated containing all three-site clusters up to the second nearest neighbor and pairs up to the first, second, third, and fourth nearest neighbors. For the coarse-graining methods, $\gamma_s = 10^7$ was used instead of $\gamma_s = \infty$ for numerical reasons. Two different methods for generating λ_α

TABLE I. (Color online) The root-mean-squared prediction error for different combinations of candidate cluster sets, training set sizes, cross-validation techniques, and data sets. The highlighted cells indicate the cross-validation method that produces the better result.

Training set size	Neighbor beyond which CE is cut off	Ag-Au		Pb-Ir		Si-Ge	
		GCV	LOO CV	GCV	LOO CV	GCV	LOO CV
15	1st nearest neighbor	0.00782	0.00779	0.23749	0.24615	0.00347	0.00340
	2nd nearest neighbor	0.00507	0.00426	0.32017	0.36454	0.00314	0.00323
	3rd nearest neighbor	0.00338	0.00440	0.30208	0.41684	0.00267	0.00261
	4th nearest neighbor	0.00393	0.00366	0.27123	0.35780	0.00238	0.00236
30	1st nearest neighbor	0.00723	0.00723	0.22381	0.22677	0.00369	0.00373
	2nd nearest neighbor	0.00219	0.00220	0.14714	0.14886	0.00256	0.00266
	3rd nearest neighbor	0.00243	0.00250	0.20600	0.22829	0.00222	0.00224
	4th nearest neighbor	0.00317	0.00291	0.29556	0.26109	0.00233	0.00230
45	1st nearest neighbor	0.00709	0.00714	0.21872	0.22082	0.00352	0.00354
	2nd nearest neighbor	0.00189	0.00193	0.12810	0.13159	0.00250	0.00255
	3rd nearest neighbor	0.00158	0.00191	0.18587	0.17511	0.00206	0.00204
	4th nearest neighbor	0.00238	0.00234	0.19524	0.22796	0.00237	0.00224
60	1st nearest neighbor	0.00686	0.00689	0.21002	0.21156	0.00357	0.00358
	2nd nearest neighbor	0.00183	0.00184	0.12387	0.12540	0.00234	0.00231
	3rd nearest neighbor	0.00109	0.00112	0.13336	0.13274	0.00189	0.00188
	4th nearest neighbor	0.00176	0.00181	0.16000	0.16836	0.00193	0.00201
75	1st nearest neighbor	0.00670	0.00681	0.20764	0.20875	0.00370	0.00371
	2nd nearest neighbor	0.00182	0.00181	0.12380	0.12337	0.00226	0.00226
	3rd nearest neighbor	0.00102	0.00106	0.12690	0.12565	0.00192	0.00193
	4th nearest neighbor	0.00123	0.00124	0.15601	0.15968	0.00177	0.00182
Average		0.00352	0.00354	0.19865	0.21307	0.00261	0.00262

were used: the inverse-density regularization function and a variation in the pair-distances regularization function in which λ_α is given by

$$\lambda_\alpha = \left\{ \sum_{i|b_i \neq 0} \left[\gamma_3 \prod_{j|b_j \neq 0, j \neq i} (1 + \gamma_1 r_{i,j}) \right]^{1/\gamma_2} \right\}^{-\gamma_2}, \quad \vec{b} \in \alpha. \tag{33}$$

The advantage to Eq. (33) over Eq. (19) is that it replaces the individual point parameters γ_i with a single parameter γ_3. For a nanoparticle, in which there are a large number of symmetrically distinct sites, this replacement significantly reduces the total number of parameters.

It was found that both the inverse-density regularization function and modified pair-distances regularization functions gave very similar results. The mean-squared prediction error for different combinations of candidate cluster sets, training set sizes, and regularization functions are given in Table II. In general the best cluster expansions are those generated with the largest set of candidate clusters and largest training sets. In the limit of the largest training set and the largest candidate cluster set considered, the best regularization function performs about four to five times better than coarse graining and one and a half to two times as well as the cluster expansion in which similarity between clusters is ignored. On average, the best coupled regularization functions produce better cluster expansions with a training set of 30 structures than the uncoupled regularization function produces with the largest training set evaluated, containing 75 structures.

Coarse graining, even when surface clusters are left out, generally produces cluster expansions with the highest prediction error. The prediction error does not decrease significantly when the size of the training set is increased because there is no way to recover the error introduced into the cluster expansion by the unrealistically restrictive prior. The uniform-coupling regularization function performs the best on average, although the regularization function based on local-environment similarity is most frequently the best. The reason for this difference is that the uniform-coupling regularization function tends to do much better with small training set sizes. The success of the uniform-coupling regularization function for small training sets is likely due to its simplicity; the single parameter reduces the problem of overfitting the cross-validation score. For larger training sets, additional degrees of freedom are not as problematic and the more complicated regularization functions generally start to perform better. However the importance of the prior distribution diminishes with increased training set size, so the uniform-coupling regularization function never falls far behind the more complicated regularization functions.

V. RELATED METHODS

It is useful to compare the Bayesian approach to other methods that have been used to generate cluster expansions. The method of cluster selection is widely used, the use of cross validation with cluster selection was first proposed by Van de Walle and Ceder.[29] The method of independent values is closely related to that proposed by Diaz-Ortiz and co-workers.[35,36] The results in this paper suggest that low

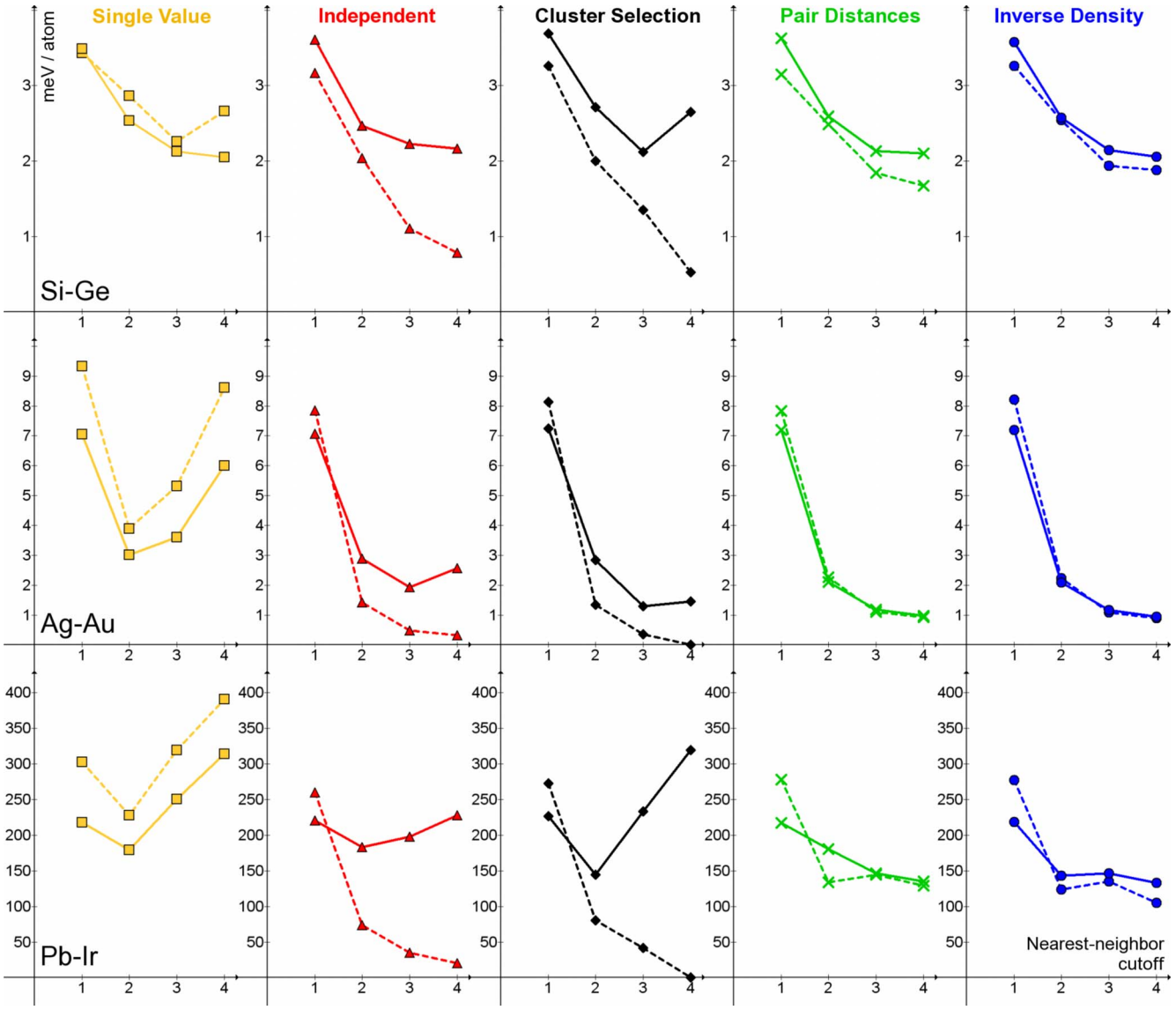

FIG. 2. (Color online) The average leave-one-out cross-validation scores (dashed lines) and root-mean-square prediction error over the entire test set (solid lines) for the Si-Ge (top row), Ag-Au (middle row), and Pb-Ir (bottom) data sets as a function of the cutoff radius of the candidate cluster set. Cluster sets with cutoff radii at the first, second, third, and fourth nearest neighbor were evaluated.

leave-one-out cross-validation scores frequently obtained using these methods may not be reliable indicators of low prediction error, especially if the candidate cluster set is large.

It has been suggested that forms of leave-many-out cross validation, in which multiple elements are left out of the training set at once, may be used in place of leave-one-out cross validation.[37] There is evidence that for cluster selection in systems with only a few significant ECI, leave-many-out cross validation can significantly reduce prediction error compared to leave-one out cross validation but the difference between the two approaches rapidly diminishes as the number of significant ECI increases.[38] In addition, leave-many-out cross validation can be an unreliable indicator of prediction error for cluster selection.[38] In contrast, by reducing the number of degrees of freedom in the model selection procedure, we are able to obtain accurate estimates of prediction error for cluster expansions with a large number of significant ECI.

The single-width regularization function is equivalent to ridge regression, a common statistical tool.[39] Jansen and Popa[40] have recently used a related method to calculate the lateral interaction energies of adsorbates on a surface. Their method is equivalent to using a single-width regularization function with a fixed small value for γ. They use this approach in conjunction with prior probabilities assigned to sets of included cluster functions to perform a Bayesian version of cluster selection, an approach that is proposed as an alternative to the use of cross validation.

The mixed-basis cluster expansion,[37,41] a method developed to address the need to include a large number of pair terms in some cluster expansions, combines concepts from cluster selection and the inverse-density regularization func-

TABLE II. (Color online) The mean-squared prediction error in meV/atom for different combinations of candidate cluster sets, training set sizes, and $\lambda_{\alpha\beta}$ regularization functions. The highlighted cells indicate the best regularization function for each row. The inverse-density regularization function was used to generate λ_α and it was found that the modified pair-distances regularization function in Eq. (33) produced similar results.

		$\lambda_{\alpha\beta}$ generator					
Pair cluster cutoff	**Training set size**	**No coupling**	**Coarse-grained**	**Coarse-grained except surface**	**Uniform coupling**	**Similarity to bulk**	**Local similarity**
1st nearest neighbor	15	2.289	3.329	3.241	2.106	2.127	1.918
	30	1.558	2.967	3.135	1.260	1.536	1.285
	45	1.500	2.973	2.984	1.200	1.202	1.197
	60	1.536	2.884	2.852	1.191	1.191	1.194
	75	1.416	2.875	2.844	1.163	1.166	1.161
1st nearest neighbor average		1.660	3.006	3.011	1.384	1.444	1.351
2nd nearest neighbor	15	2.841	2.865	3.123	2.520	2.659	2.575
	30	1.124	2.799	3.198	0.741	0.803	0.746
	45	0.924	2.746	3.016	0.680	0.702	0.653
	60	0.850	2.746	2.756	0.667	0.715	0.646
	75	0.697	2.733	4.193	0.639	0.643	0.631
2nd nearest neighbor average		1.287	2.778	3.257	1.049	1.105	1.050
3rd nearest neighbor	15	1.849	2.848	3.103	1.162	1.139	1.844
	30	1.291	2.748	3.125	0.833	0.842	0.824
	45	1.096	2.751	3.447	0.660	0.688	0.654
	60	0.879	2.739	2.746	0.627	0.624	0.601
	75	0.953	2.730	2.713	0.640	0.641	0.668
3rd nearest neighbor average		1.214	2.763	3.027	0.784	0.787	0.918
4th nearest neighbor	15	2.521	3.233	3.195	2.020	2.075	3.178
	30	1.792	2.789	3.089	0.801	0.837	0.785
	45	1.474	2.759	3.157	0.658	0.693	0.643
	60	1.158	2.734	2.780	0.640	0.625	0.612
	75	1.161	2.744	2.681	0.585	0.611	0.563
4th nearest neighbor average		1.621	2.852	2.980	0.941	0.968	1.157
Overall average		1.446	2.850	3.069	1.040	1.076	1.119

tion. The mixed-basis cluster expansion is equivalent to using the following regularization function for pair clusters only:

$$\lambda_\alpha(r_\alpha) = \gamma_1 \frac{m_\alpha r_\alpha{}^{\gamma_2}}{\left(\sum_{\alpha'} \sqrt{\frac{r_{\alpha'}{}^{\gamma_2}}{m_{\alpha'}}}\right)^2}, \tag{34}$$

where m_α is the multiplicity of cluster orbit α, r_α is the distance between the two sites, and γ_1 and γ_2 are free parameters. The sum is over all candidate cluster orbits. All point clusters are included without regularization and clusters of more than two sites are selected via cluster selection. To limit the degrees of freedom of the mixed-basis cluster expansion, constraints may be applied to the number of multisite cluster functions with nonzero ECI allowed in the expansion.[37] In contrast, the inverse-density regularization function applies no such limit but reduces the number of degrees of freedom through parameterization while simultaneously encouraging smaller ECI for clusters with a larger number of sites.

The idea of treating surface interactions as a perturbation of bulk interactions has been applied by Müller *et al.*,[42] inspired by the insight of Drautz *et al.*[43] In their approach, surface energies are treated as perturbation of bulk energies and a cluster expansion is developed for the difference between the surface energies and bulk energies. In our framework, this is equivalent to treating the bulk energies as the mean of the prior distribution for surface energies. The cluster expansion of the difference between surface and bulk energies is equivalent to cluster expanding y_Δ as described in the discussion of insight #1. Our approach takes the additional step of asserting that the prior probability of small perturbations is higher than that of large perturbations. In addition, we introduce the ability to couple the ECI for congruent clusters, as in the "similarity to bulk" regularization function, which allows the insight of Drautz *et al.* to be applied without ever needing to calculate a bulk cluster expansion. Using the perturbation method without regularization, Drautz *et al.*[43] found that about 100 distinct significant ECI were required to calculate surface energies for a single binary (100) surface. Determining values for these ECI required a training set of 160 72-atom slabs. It is our belief that the methods described in this paper will significantly reduce

the size of the required training set for such problems.

VI. DISCUSSION

We have presented a Bayesian framework for training cluster expansions. Several existing methods can be expressed in the context of this framework and within this framework we have proposed methods for training cluster expansions that consistently produce cluster expansions with low prediction error. The keys to generating a cluster expansion with low prediction error are as follows:

Use Bayes' theorem, or equivalently, regularization. The use of Gaussian prior distributions as described in this paper is mathematically equivalent to Tikhonov[28] regularization, which improves the convergence of the cluster expansion with respect to training set size. One of the benefits of using regularization is that an arbitrarily large number of ECI may be determined for a given training set, allowing efficient generation of cluster expansions that might have low symmetry, include complex interactions or include significant long-range interactions.

Use a physically meaningful prior distribution. The more physically meaningful the prior distribution is, the more rapidly the cluster expansion will converge. Methods such as cluster selection, the independent regularization function, or the single-value regularization function do not incorporate much physical insight into the prior distribution, and as a result they converge relatively slowly. On the other hand, the prior distribution can be thought of as a "soft" constraint: a good prior distribution will lead to rapid convergence, whereas a bad prior distribution will still lead to convergence but at a slower pace (i.e., more training data will be required).

Use cross validation to determine the prior distribution. Cross validation has been widely used in cluster expansions since initially proposed by Van de Walle and Ceder[29] and it can be incorporated into the Bayesian framework by using parameterized functions to generate prior distributions. We have found that generalized cross validation works about as well as leave-one-out cross validation with the advantage of being faster. Forms of leave-many-out cross validation may also be considered.[37]

Use a regularization function with few parameters. The use of regularization functions with a large number of degrees of freedom, such as cluster selection or the independent regularization function, can lead to overfitting of the cross-validation score. The result is that the cross-validation score ceases to be a meaningful measure of true prediction error and optimizing the cross-validation score does little to improve the prediction error.

Regularized cluster expansions typically include more cluster functions than those generated through cluster selection. Although this usually leads to more accurate cluster expansions for a given training set size, there is a performance penalty when using a cluster expansion with a large number of nonzero ECI. This situation is easily remedied by identifying the smallest ECI determined by the Bayesian approach and removing those clusters from the fit. This is an artificial constraint that will most likely slightly increase the prediction error for the cluster expansion but the benefit of more rapid evaluation may be worth the tradeoff. It is also important to note that no matter how many nonzero ECI are included in the cluster expansion, it is always possible to develop insight into the most dominant interactions by identifying the largest ECI. The values of the largest ECI will be more meaningful for highly accurate cluster expansions, which often require a large number of terms.

As researchers apply the cluster expansion approach to increasingly complex systems, it is important to improve the efficiency of cluster expansions and develop reliable metrics of cluster expansion quality. The Bayesian approach, using regularization functions with few parameters, addresses each of these problems. This approach promises to make the generation of cluster expansions for low-symmetry systems such as surfaces and nanoparticles feasible with a level of accuracy comparable to that of bulk cluster expansions. Using this approach, researchers should be able to develop more physically meaningful methods for generating prior distributions and further improve cluster expansion quality.

ACKNOWLEDGMENTS

This work was funded by the Department of Energy under Grants No. DE-FG02-96ER4557 and No. DE-FG02-05ER46253. Supercomputing resources from the San Diego Supercomputing Center are also acknowledged.

APPENDIX: DERIVATION OF BAYESIAN CLUSTER EXPANSION

The general application of Bayes' theorem to function learning is well known[39] and here we derive the application of Bayes' theorem to the cluster expansion. We start by providing a brief introduction to Bayes' theorem. The joint probability of two events, $P(A,B)$, can be expressed as

$$P(A,B) = P(A|B)P(B) = P(B|A)P(A), \tag{A1}$$

where $P(A|B)$ is the probability of A given B and $P(B|A)$ is similarly defined. For example, the probability that it is raining *and* I am carrying an umbrella is equal to the probability that I am carrying an umbrella given that it is raining, times the probability that it is raining. Rearranging Eq. (A1) gives us Bayes' theorem[20]

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}. \tag{A2}$$

In terms of a cluster expansion, we are trying to maximize $P(\vec{v}|X,\vec{y})$. In the continuous limit, Bayes' theorem tells us[44]

$$P(\vec{v}|X,\vec{y}) = \frac{P(\vec{y}|\vec{v},X)P(\vec{v}|X)}{P(\vec{y}|X)}. \tag{A3}$$

The first term in the numerator on the right, $P(\vec{y}|\vec{v},X)$, is the probability distribution of the training outputs, $\vec{y}$, given the training inputs, X, and the optimal ECI. The second term in the numerator on the right, $P(\vec{v}|X)$, is the prior distribution, which represents a prior expectation for the values of the ECI before the training outputs are generated. The denominator

may be treated as a constant with respect to the ECI. From Eq. (A3) and the fact that the natural log is a monotonically increasing function, it follows that the ECI that maximize $P(\vec{v}|X,\vec{y})$ are given by

$$\hat{\vec{V}} = \arg\min_{\vec{v}}[-\ln(P(\vec{y}|\vec{v},X)) - \ln(P(\vec{v}|X))]. \quad \text{(A4)}$$

To find a closed-form solution for Eq. (A4), we first derive expressions for $P(\vec{y}|\vec{v},X)$ and $P(\vec{v}|X)$.

To derive an expression for $P(\vec{y}|\vec{v},X)$ we start with a linear least-squares regression model, in which property values in the training set are assumed to be generated by an optimal linear combination of included cluster functions plus normally distributed random noise

$$\vec{y} = X\vec{V} + \vec{\tilde{e}}, \quad \text{(A5)}$$

where the ith element of the vector $\vec{\tilde{e}}$, $\tilde{e}_i$, is an independent normally distributed random variable with mean 0 and standard deviation σ_i. The noise reflects the facts that if we are not using the complete basis of cluster functions the cluster expansion will generally not exactly reproduce the training data. In addition, there will be noise if the generation of the property values for the training data is not completely deterministic. From Eq. (A5), we get

$$P(\vec{y}|\vec{v},X) \propto \prod_i e^{-(y_i - \vec{x}_i \cdot \vec{v})^2/2\sigma_i^2}, \quad \text{(A6)}$$

where $\vec{x}_i$ is the ith row of X and the product is over all elements of the training set.

An expression for the prior distribution, $P(\vec{v}|X)$, can be derived by combining Eqs. (7) and (8)

$$P(\vec{v}|X) \propto \prod_\alpha e^{-v_\alpha^2/2\sigma_\alpha^2} \prod_{\alpha,\beta\neq\alpha} e^{-(v_\alpha - v_\beta)^2/2\sigma_{\alpha\beta}^2}. \quad \text{(A7)}$$

It is important to remember that this prior distribution assumes that the training data have been transformed by subtracting out the expected property value for each data point, so that the prior expected value for each ECI is zero.

Combining Eqs. (A4), (A6), and (A7), we get

$$\hat{\vec{V}} = \arg\min_{\vec{v}}\left[\sum_i \frac{(y_i - \vec{x}_i \cdot \vec{v})^2}{2\sigma_i^2} + \sum_\alpha \frac{v_\alpha^2}{2\sigma_\alpha^2} + \sum_{\alpha,\beta\neq\alpha} \frac{(v_\alpha - v_\beta)^2}{2\sigma_{\alpha\beta}^2}\right]. \quad \text{(A8)}$$

The values σ_i, introduced in Eq. (A5), represent the magnitude of the random noise that differentiates the property values in the training set from the values predicted by the target function. They are a metric of how well the target function predicts property values and they are in general unknown because we do not know the error introduced by truncating the cluster expansion. However, for a truncated expansion we may specify the *relative* values of σ_i by defining weights, w_i, for each structure in the training set

$$w_i = \frac{\sigma^2}{\sigma_i^2} \quad \text{(A9)}$$

for some unknown proportionality constant σ^2. The larger the weight w_i, the more accurately the truncated expansion should predict the property value for the ith element of the training set relative to the other elements in the training set. Combining Eqs. (A8) and (A9), and rearranging terms, yields

$$\hat{\vec{V}} = \arg\min_{\vec{v}}\left[\sum_i w_i(y_i - \vec{x}_i \cdot \vec{v})^2 + \sum_\alpha \left(\frac{\sigma^2}{\sigma_\alpha^2} + \sum_{\beta|\beta\neq\alpha} \frac{\sigma^2}{\sigma_{\alpha\beta}^2}\right) v_\alpha^2 - \sum_\alpha \sum_{\beta|\beta\neq\alpha} \left(\frac{\sigma^2}{\sigma_{\alpha\beta}^2}\right) v_\alpha v_\beta\right], \quad \text{(A10)}$$

which can be written in matrix-vector notation,

$$\hat{\vec{V}} = \arg\min_{\vec{v}_\Delta}([\vec{y} - X\vec{v}]^T W[\vec{y} - X\vec{v}] + \vec{v}^T \Lambda \vec{v}), \quad \text{(A11)}$$

where W is a diagonal matrix in which $W_{ii} = w_i$ and the elements of the matrix Λ are given by Eq. (10). The minimum in Eq. (A11) can be directly determined by taking the derivative with respect to $\vec{v}$, yielding Eq. (9).

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

[1] E. Ising, Z. Phys. **31**, 253 (1925).
[2] D. de Fontaine, in *Solid State Physics*, edited by D. Turnbull and F. Seitz (Academic, New York, 1979), Vol. 34, p. 73.
[3] J. M. Sanchez, F. Ducastelle, and D. Gratias, Physica A **128**, 334 (1984).
[4] R. B. Stinchcombe, J. Phys. C **6**, 2459 (1973).
[5] V. Ozolins, C. Wolverton, and A. Zunger, Phys. Rev. B **57**, 6427 (1998).
[6] M. H. F. Sluiter, Y. Watanabe, D. deFontaine, and Y. Kawazoe, Phys. Rev. B **53**, 6137 (1996).
[7] A. Van der Ven, M. K. Aydinol, G. Ceder, G. Kresse, and J. Hafner, Phys. Rev. B **58**, 2975 (1998).
[8] N. A. Zarkevich, T. L. Tan, and D. D. Johnson, Phys. Rev. B **75**, 104203 (2007).
[9] B. P. Burton, Phys. Rev. B **59**, 6087 (1999).
[10] C. Wolverton and A. Zunger, J. Electrochem. Soc. **145**, 2424 (1998).
[11] Atsuto Seko, Koretaka Yuge, Fumiyasu Oba, Akihide Kuwabara, and Isao Tanaka, Phys. Rev. B **73**, 184117 (2006).
[12] Brian Kolb and Gus L. W. Hart, Phys. Rev. B **72**, 224207 (2005).
[13] G. D. Garbulsky and G. Ceder, Phys. Rev. B **49**, 6327 (1994).
[14] A. Van der Ven, G. Ceder, M. Asta, and P. D. Tepesch, Phys. Rev. B **64**, 184307 (2001).
[15] A. Van De Walle, Nature Mater. **7**, 455 (2008).
[16] A. Franceschetti and A. Zunger, Nature (London) **402**, 60 (1999).
[17] Tim Mueller and Gerbrand Ceder, Phys. Rev. B **74**, 134104

(2006).
[18] Fei Zhou, Gevorg Grigoryan, Steve R. Lustig, Amy E. Keating, Gerbrand Ceder, and Dane Morgan, Phys. Rev. Lett. **95**, 148103 (2005).
[19] Fei Zhou, Thomas Maxisch, and Gerbrand Ceder, Phys. Rev. Lett. **97**, 155704 (2006).
[20] M. Bayes and M. Price, Philos. Trans. R. Soc. London **53**, 370 (1763).
[21] S. Curtarolo, D. Morgan, and G. Ceder, CALPHAD: Comput. Coupling Phase Diagrams Thermochem. **29**, 163 (2005).
[22] E. T. Jaynes, *Probability Theory: The Logic of Science* (Cambridge University Press, Cambridge, UK, 2003).
[23] R. V. Chepulskii and W. H. Butler, Phys. Rev. B **72**, 134205 (2005).
[24] B. C. Han, A. Van der Ven, G. Ceder, and B. J. Hwang, Phys. Rev. B **72**, 205409 (2005).
[25] B. Yang, M. Asta, O. N. Mryasov, T. J. Klemmer, and R. W. Chantrell, Scr. Mater. **53**, 417 (2005).
[26] S. Müller, J. Phys.: Condens. Matter **15**, R1429 (2003).
[27] S. Ouannasser, H. Dreysse, and L. T. Wille, Solid State Commun. **96**, 177 (1995).
[28] A. N. Tikhonov and V. Y. Arsenin, *Solutions of Ill-Posed Problems* (John Wiley & Sons, Washington, DC, 1977).
[29] A. van de Walle and G. Ceder, J. Phase Equilib. **23**, 348 (2002).
[30] Nikolai A. Zarkevich and D. D. Johnson, Phys. Rev. Lett. **92**, 255702 (2004).
[31] J. Tersoff, Phys. Rev. B **39**, 5566 (1989).
[32] A. P. Sutton and J. Chen, Philos. Mag. Lett. **61**, 139 (1990).
[33] H. V. Henderson and S. R. Searle, SIAM Rev. **23**, 53 (1981).
[34] G. H. Golub, M. Heath, and G. Wahba, Technometrics **21**, 215 (1979).
[35] A. Diaz-Ortiz, H. Dosch, and R. Drautz, J. Phys.: Condens. Matter **19**, 406206 (2007).
[36] R. Drautz and A. Diaz-Ortiz, Phys. Rev. B **73**, 224207 (2006).
[37] Volker Blum and Alex Zunger, Phys. Rev. B **70**, 155108 (2004).
[38] K. Baumann, TrAC, Trends Anal. Chem. **22**, 395 (2003).
[39] D. V. Lindley and A. F. M. Smith, J. R. Stat. Soc. Ser. B (Methodol.) **34**, 1 (1972).
[40] A. P. J. Jansen and C. Popa, Phys. Rev. B **78**, 085404 (2008).
[41] D. B. Laks, L. G. Ferreira, S. Froyen, and A. Zunger, Phys. Rev. B **46**, 12587 (1992).
[42] S. Muller, M. Stöhr, and O. Wieckhorst, Appl. Phys. A: Mater. Sci. Process. **82**, 415 (2006).
[43] R. Drautz, H. Reichert, M. Fahnle, H. Dosch, and J. M. Sanchez, Phys. Rev. Lett. **87**, 236102 (2001).
[44] D. G. T. Denison, C. C. Holmes, B. K. Mallick, and A. F. M. Smith, *Bayesian Methods for Nonlinear Classification and Regression* (John Wiley & Sons, Chichester, UK, 2002).