TIF345

ADVANCED SIMULATION AND MACHINE LEARNING

Project 2a: Alloy cluster expansions

Authors:
Simon Josefsson (simjos)
Tomas Lundberg (lutomas)

Examinator:
Paul Erhart

Tutor:
Eric Lindgren

December 8, 2021

Introduction

In this report we explore five different linear models; ordinary least squares (OLS), Ridge regression, a phyiscal intuitive model through a covariance matrix, full Bayesian analysis and automatic relevance feature detection (ARDR). We use these models to fit effective cluster interactions (ECIs) $\boldsymbol{J}$ according to

$$X\boldsymbol{J} = \boldsymbol{E}_{mix} \tag{1}$$

where $\boldsymbol{E}_{mix}$ is the mixing energy and X is a matrix of cluster vectors [1]. A cluster vector for a particular structure is a vector where each element correspond to the average number of times a cluster occurs in the structure. We compare the performance of the different models in task 2-5 and predict the ground state structure and energy on unseen data in task 6.

Task 1 - Prepare Data

Method: First, in this task the data from the supplied database `AuCu-structures.db` was read. With the use of the `icet` package we could obtain X and $\boldsymbol{E}_{mix}$, see eq. 1, from the data with cutoff radii of of 8Å, 6Å and 5Å for pairs, triplets and quadruplets respectively. From the information retrieved from the data the Cu concentration was calculated by taking the number of Cu atoms divided by the number of atoms for each structure. Finally, we standardized the data, both X and $\boldsymbol{E}_{mix}$, to have zero mean and unit variance.

Results and Discussion: The mixing energy per atom for different Cu concentrations is presented in figure 1a. Although, the data seem to be rather noisy, there is an indication that concentrations close to ~ 0.6 have the lowest mixing energies and that the variance goes down for both lower and higher concentrations.

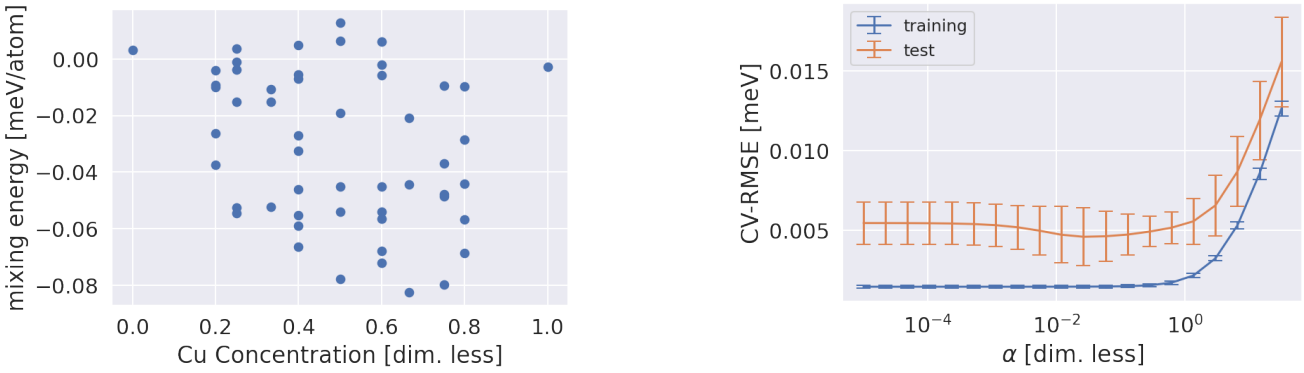
The standardisation of data is used when different data features is measured at different scales (have different means and or variances) and hence cannot be compared on equal footing. Standardisation is therefore incorporated to improve learning. To show how this might be helpful, consider the case where the magnitude of the features is important for the choice of hyperparamters. For instance, when using the Ridge model we set a hyperparameter that penalizes large values. Without standardisation a variable with large variance will for example outweigh a variable with small variance. In our case, both X and $\boldsymbol{E}_{mix}$ *are* at different scales, so standardizing the data is necessary[1].

Task 2 - OLS and Ridge Regression

Method: In this task the OLS and ridge models were used to fit the ECIs for each structure with the help of the `sklearn` library. The ridge model uses a hyperparameter, α, to penalize large values of ECIs. To find a good value for α a 5-fold cross validation (CV) was used, evaluated on the root-mean-square error (RMSE).

Results and Discussion: From figure 1b a good value for α could be estimated to lie between 0.05 and 0.5, we have chosen a value of α=0.1 based on the test data mean and standard deviation. The fitted ECIs are presented in figure 2 (*left*) (together with

[1] In fact, training the models on unstandardized data yields notably different results.



(a) (b)

Figure 1: (a) The Cu concentration against the mixing energy for all structures in the dataset with cutoffs of 8Å, 6Å and 5Å for pairs, triplets and quadruples respectively. (b) The 5-fold cross validation root-mean-square error (CV-RMSE) for different values of the α parameter used in Ridge regression on test and training data. Error bars indicate $\pm\sigma$. We note a minimum of the CV-RMSE on the test data for $\alpha \in [0.05, 0.5]$.

the ECIs of the later models in interest of space). Note that the ECIs in the figure are those obtained with the standardized data and are thus dimensionless, i.e. the actual values does not have a physical meaning. For simplicity we have opted to keep it this way. Nevertheless, the ECIs in figure 2 are slightly smaller in magnitude for OLS in comparison to Ridge for smaller orbit indices and vice versa for larger indices. This is to be expected for two reasons. First, for physical reasons, J_i for smaller orbit indices are anticipated to be larger. Second, the α parameter penalizes larger ECIs in the Ridge model. Therefore, the Ridge model decreases the parameter values for the larger ECIs (lower indices) and instead increase the smaller ECIs (higher indices), where the incurred α-penalty is lower, to best fit the data.

Additionally, in figure 2 (*right*) the mean and standard deviation of the CV-RMSE for the models are presented. Recall that if α is set to be zero, Ridge becomes OLS and larger values of α will penalize larger values of ECIs. The Ridge model hence improves upon the OLS model to better avoid overfitting the data. Moreover the mean of the RMSE is lower for Ridge, even though most of interval indicated by the $\pm$ standard deviation is overlapping for the two models.

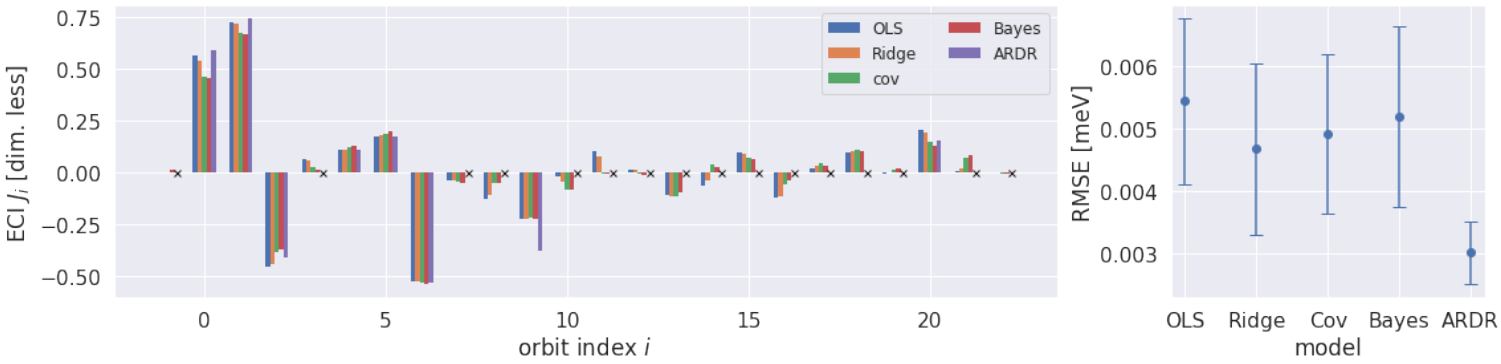


Figure 2: (*left*) The (scaled) ECIs, J_i, for orbit index $i = -1, 0, 1, \ldots, 22$ for all models. A cross indicates a zero-parameter for ARDR. (*right*) The root mean squared error (RMSE) with $\pm\sigma$ for all models. The RMSE was calculated through 5-fold cross validation on test data for all models but Bayes, where it instead was calculated through the MCMC samples.

Task 3 - Covariance matrix for cluster similarity

Method: In the Ridge model we used a hyperparameter α that penalized larger values of ECIs. In this task we are going to explore a model where we incorporate our physical intuition. Namely, that we expect orbits with more sites n and larger radius r to contribute less. This is implemented using a regularization matrix Λ, which also can be interpreted as the inverse of the covariance matrix, hence the model is referred to as the Cov model. We are going to focus on the diagonal terms, Λ_{ii} while setting the off diagonal terms to zero. To capture our physical intuition, we set the diagonal terms Λ_{ii} to

$$\lambda_i(n, r) = \gamma_1(\gamma_2 r + \gamma_3 + 1)^{\gamma_4 n + \gamma_5}. \tag{2}$$

So, instead of finding λ_i individually, the problem is reduced to finding the five hyperparameters $\boldsymbol{\gamma}$. By using the `scipy` package with the function `optimize.minimize` we found the five hyperparameters $\boldsymbol{\gamma}$ by minimizing the mean of the test RMSE for a 5-fold CV.

Results and Discussion: The minimization results in the optimal values $\boldsymbol{\gamma} = \{0.66, 0.49, 0.74, 0.034, 0.53\}$ which corresponds to a penalty λ_i^{opt} as shown in figure 3, together with r and n. This gives us ECIs that are in general similar as those found for OLS and Ridge in the previous task, but slightly lower in magnitude, see figure 2 (*left*). Even though larger order indices are given larger penalties, the penalty λ_i^{opt} is still roughly one order of magnitude larger than $\alpha = 0.1$ used in the Ridge model which serves as an explanation for why the ECIs are in general smaller in magnitude than those for found by the Ridge model.

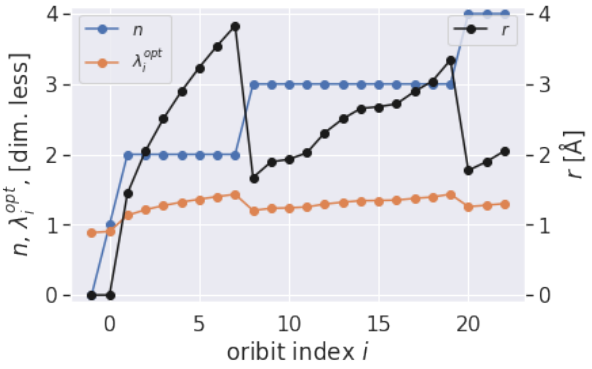


Figure 3: The resulting penalty λ_i^{opt}, see eq. 2, for optimal $\boldsymbol{\gamma}$ shown together with radius r and sites n.

When including the Λ, each orbit incurrs a unique penalty λ_i, which depends on the properties n and r of the orbits. In contrast, the Ridge model uses the same parameter α for every orbit. The RMSE presented in figure 2 (*right*) indicate an improvement in the fitting of our data with the covariance matrix compared to the OLS model, but the Ridge model still fits the data better in terms of the mean RMSE[2]. This could be caused by the fact that different entries of $\boldsymbol{\gamma}$ have different units which makes optimization difficult and perhaps even nonsensical. Even though incorporating our physical intuition seem like a good idea, when combining this strategy with machine learning, great care has to be taken with regards to units of the parameters such that it makes sense from both a physical and machine learning perspective.

Task 4 - Bayesian Cluser Expansion

Method: We performed a full Bayesian analysis through Markov chain Monte Carlo (MCMC) sampling. 26 parameters were included in the sampling, the first 24 were the ECIs J_i with orbit index $i = -1, 0, 1, \ldots, 22$ and the last two the error scales σ and α for

[2]Note that the $\pm$ standard deviation interval shows significant overlap between OLS, Ridge and Cov so nothing conclusive can be said.

the likelihood and the priors over **J** respectively. We made use of a Gaussian prior for the ECIs with zero mean as

$$P(\boldsymbol{J}) = \frac{1}{(2\pi\alpha^2)^{N_p/2}} \exp\left(-||\boldsymbol{J}||^2/2\alpha^2\right), \tag{3}$$

where N_p is the length of the $\boldsymbol{J}$-vector, and a inverse gamma ($\mathcal{IG}$) distribution for α as $\mathcal{IG}(0.001, 0.05)$ to represent our prior knowledge obtained from the previous task that α is approximately between 0.05 an 0.5[3]. Moreover, we used a Gaussian likelihood

$$P(D \mid \boldsymbol{J}, \sigma) = \frac{1}{(2\pi\sigma^2)^{N_p/2}} \exp\left(-\frac{1}{2\sigma^2}||\boldsymbol{E} - X\boldsymbol{J}||^2\right), \tag{4}$$

and $\mathcal{IG}(0.01, 0.1)$ for the prior of σ to capture that σ is approximately between 0.1 and 1 as hinted by the ECIs in figure 2. The posterior probability distribution was then obtained as

$$P(\boldsymbol{J}, \sigma, \alpha|D) \propto P(D|\boldsymbol{J}, \sigma)P(\boldsymbol{J})P(\sigma)P(\alpha). \tag{5}$$

For numerical stability we sampled the log of the distributions. We used 60 walkers and 50,000 steps per walker[4].

Results and Discussion: In figure 4b, the chains for all parameters are shown together with the used burn-in (*dashed black*) after which the walkers had stabilized. In figure 4a, the posterior distributions over the (scaled) ECIs are presented[5]. The orbit indicies in bold (see *y*-axis) are those ECIs that are likely to be non-zero; explicitly where 0 is not included in $\pm$ two standard deviations. These ECIs, $i = 0, 1, 2, 4, 5, 6$, all have low orbit indices while the higher are centered close to zero with just a few exceptions. Of course, choosing, say $\pm$ one standard deviations would yield more "necessary" parameters but the key take away is that a relatively small fraction of the ECIs are necessary for a good fit[6].

If we where to set priors to represent something "unphysical", for instance to favor 3rd and 4th order clusters we might get very different results. If they are only slightly favored, our results would probably not differ too much but while we sharpen the prior distribution to favour non-zero ECIs of higher order clusters our posterior distribution would be dominated by the prior. Therefore, naturally, the higher order terms would tend to be non-zero. Without testing this, it is difficult to predict how the lower order terms would be affected, but in general they would have to compensate for the, likely, erroneous higher order ECIs and thus their values might vary a lot. It is likely that the ECIs would be heavily overfitted on the training data and hence would generalize poorly on unseen data.

Task 5 - ARDR Feature selection

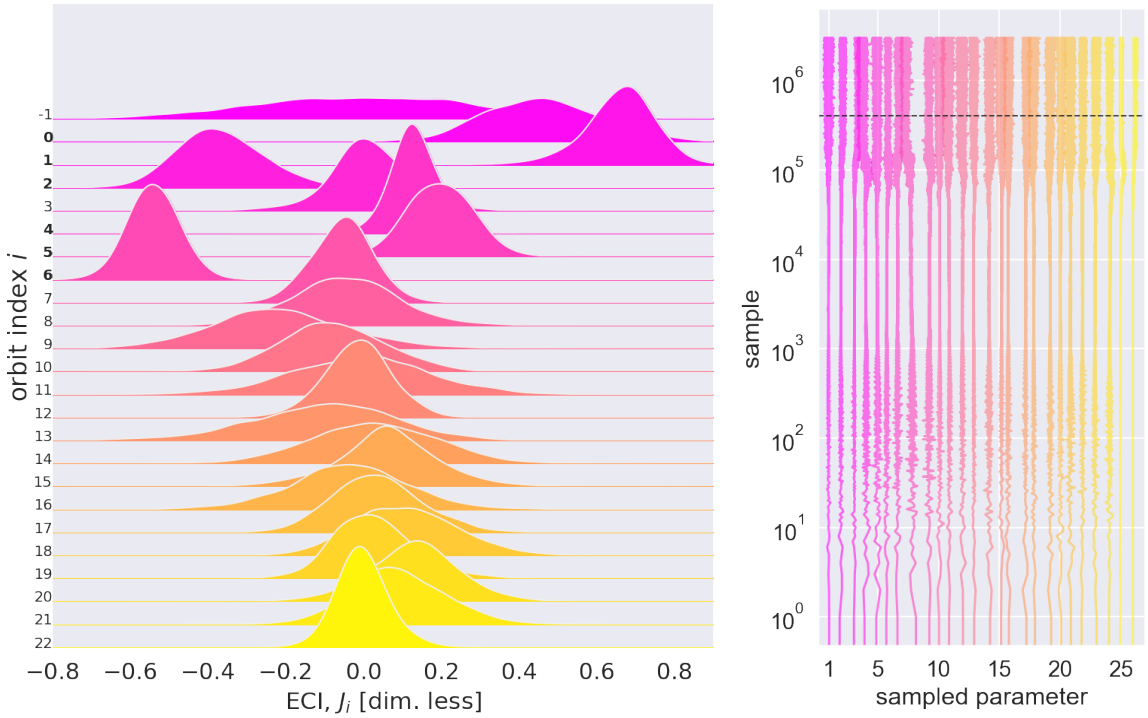
Method: We used Automatic Relevance Feature Detection Regression (ARDR) on data with higher cutoffs than previously to demonstrate the power of ARDR; 13Å, 8Å and 6Å for pairs, triplets and quadruples respectively. In short, ARDR has a hyper

[3]See plots of the priors generated in the appended notebook.

[4]See appended notebook for details on e.g. initialization.

[5]Not shown in the figure are the results of α and σ. We present them here in brevity: $\alpha = 0.29 \pm 0.03$ and $\sigma = 0.27 \pm 0.02$ ($\pm$ one standard deviation).

[6]See next task for more discussion on this.



(a) (b)

Figure 4: *(a)* The (scaled) ECI distributions (*x-axis*), i.e. J_i for the orbit indices $i = -1, 0, 1, \ldots 22$ (*y-axis*) obtained through MCMC sampling. The bold orbit indicies (0, 1, 2, 4, 5 and 6) indicate parameters that are likely to be non-zero; explicitly where 0 is not included $\pm 2\sigma$. *(b)* The MCMC chains of all 26 sampled parameters. 1-24 are $\boldsymbol{J}$ and 25 and 26 are σ and α respectively.

parameter λ_{lim} that controls the pruning of weights on the parameters, in our case J_i. A low λ_{lim} means that weights are likely to be set to 0 and if λ_{lim} is large, the parameters are more likely to be included in the fit. We performed a 5-fold CV for 30 values of λ_{lim} log-spaced between 10^0 and 10^4. We evaluated the models for a given λ_{lim} on the RMSE and the Akaike and Bayes information criteria, AIC and BIC on the test data.

Results and Discussion: In figure 5 we present the results of the λ_{lim} scan. Every model fit with a particular λ_{lim} corresponds to a certain number of non-zero model parameters, i.e. non-zero ECIs. These are shown on the common x-axis. The CV-RMSE with corresponding standard deviations are presented *above* in figure 5, the information criteria in the *middle* and the relation between λ_{lim} and the number of non-zero parameters *below*. At 10 non-zero parameters we note a maximum of both ICs, and at the same point the training and test CV-RMSE seem to deviate, indicating that we are slightly overfitting for larger number of non-zero parameters. 10 non-zero parameters corresponds to $\lambda_{lim} \sim 10^2$.

For a fair comparison with the other methods, we performed a similar analysis with the original dataset (smaller cutoffs). The optimal number of non-zero parameters were then 8, and the ECIs are presented in figure 2, together with those of the other models. Here, the zero parameters, are indicated by a black cross. the *non*-zero parameters according to the optimal ARDR model are $i = 0, 1, 2, 4, 5, 6, 9, 20$. Based on our analysis, these seem to be the features that are the most suitable to include in a final model. Moreover, note that the first 6 clusters are those found to be non-zero by our Bayesian analysis in the previous task. In addition, there seem to be a correlation

between the chosen indices and a low value of our "physically intuitive" λ_i^{opt} in figure 3. This indicates a coherence between different models and hence a higher certainty on our results. In comparison with the ECIs of the other models in figure 2 (*left*), obviously the zero-paremters differ, but otherwise not too much, with a few exceptions, most notably $i = 9$. If we look at the *right* of figure 2, we see that the CV-RMSE is notably lower both in the mean and the standard deviation. This indicates that the ARDR model generalize much better than the other models, and in extension, perhaps that the other models are overfitting with the use of all parameters. In conclusion, with our chosen metrics, ARDR seem to be the best model.

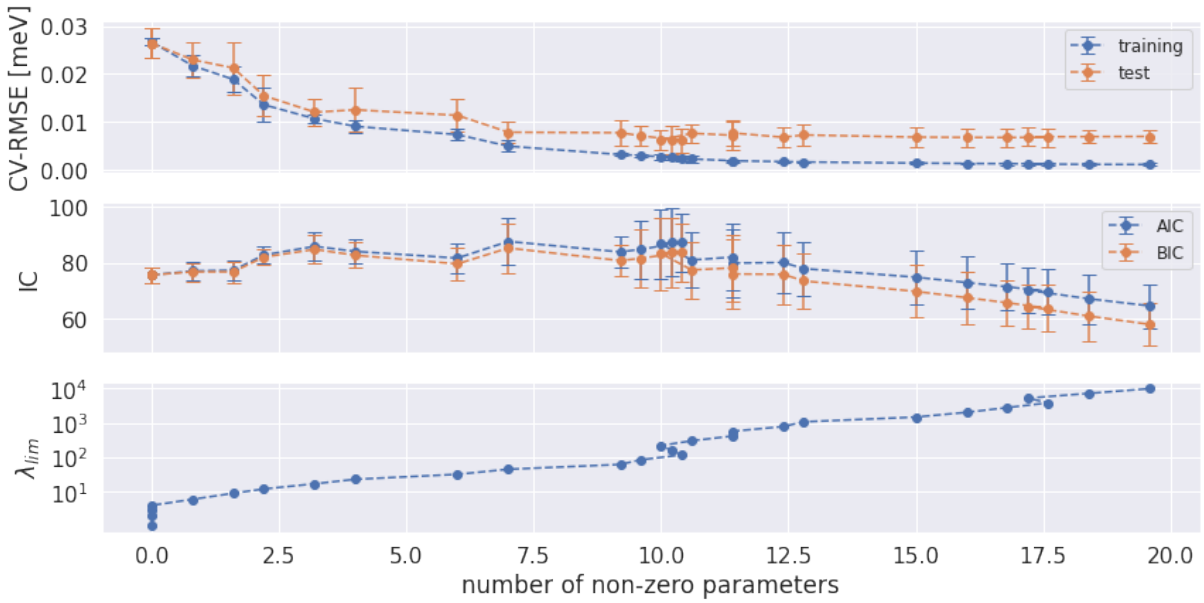


Figure 5: (*top*) The 5-fold cross validation root mean squared error (CV-RMSE) on training and test data with $\pm\sigma$; (*middle*) the Akaike information criterion (AIC) and Bayesian information criterion (BIC) on test data; and (*bottom*) the model hyper parameter λ_{lim} as a function of non-zero parameters obtained with ARDR. We note that the test CV-RMSE stops decreasing at roughly 10 non-zero parameters and that the BIC and AIC have a local maximum at the same value. This corresponds to $\lambda_{lim} \sim 10^2$.

Task 6 - The ground state

Method: In the final task, we used the ECIs obtained from all models to predict the ground state energy and structure of an unseen dataset with 57 structures indexed $0, 1, 2 \ldots 56$. For the ECIs obtained through the full Bayesian analysis, we used the samples from the MCMC chain to obtain a probability distribution over the most likely ground state as well as over the ground state energy. For the other models, we simply acquired point estimates.

Results and Discussion: The results are shown in figure 6. We see the probability distribution and point predictions of the ground state structure (*left*) and the ground state energy (*right*) for all models. The Bayesian model give the highest probability for structure index 28 (46%), followed by 3 (38%) while all other models predict structure 3 to be the ground state structure. Moreover, the point predictions of the ground state energy vary between -11.7 meV and -11.2 meV and these coincide with the maxi-

mum a posteriori for the Baysian model. The consistency of the predictions are not too surprising given that the ECIs shown in figure 2 (*left*) are very similar for all models.

In this particular problem, the Bayesian model offers a clear advantage in terms of assigning probabilities over all possible ground state structures as well as over the ground state energy. This is in stark contrast to the other models where, for instance, the likelihood of structure 28 being the ground state structure is not reflected.

Moreover, since we are dealing with a physical problem where we have clear intuitions on what is to be expected, i.e. prior knowledge, the models that can incorporate prior knowledge should perhaps be favoured in a task like this. The Covariance[7] and full Bayesian models do offer this possibility. The automatic feature detection models, especially ARDR, have shown to perform well in terms of RMSE, but lack in incorporating prior belief. Therefore, in a task like this, perhaps the more hands-on physical intuition based approaches should be favoured. Nevertheless, for this particular dataset, the predictions are in agreement.

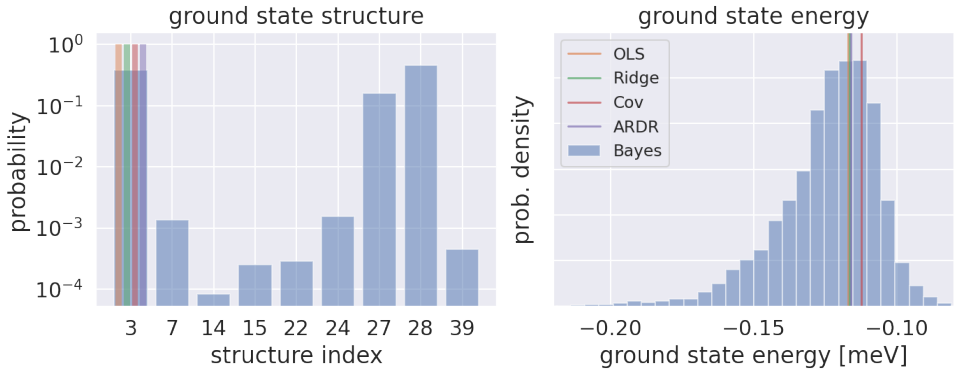


Figure 6: (*left*) The point predictions and (log-scale) probability distributions for the ground state structure over the structure indices with non-zero probabilities for all models (see legend to the *right*). (*right*) The point predictions and probability distribution over the ground state energy.

References

[1] P. Erhart and E. Lindgren, "Tif345 project 2a: Alloy cluster expansions," November 2021.

[7] Not mentioning the issuse already discussed previously.