Artificial Intelligence and Augmented Intelligence for Automated Investigations for Scientific Discovery

Relative Structural Analysis on Molecular Perovskite

Project Report

Project Dates: 14/06/2021 - 20/08/2021

School of Physics and Astronomy

Project Student: Kevin Daniel Calvache Ramos, School of Physics and Astronomy

Supervised by: Dr Anthony Phillips, School of Physics and Astronomy

Report Date: 30/08/2021

AI3SD-Intern-Series:Report-6_Calvache

Relative Structural Analysis on Molecular Perovskite
AI3SD-Intern-Series:Report-6_Calvache
Report Date: 30/08/2021
DOI: 10.5258/SOTON/AI3SD0147
Published by University of Southampton

Network: Artificial Intelligence and Augmented Intelligence for Automated Investigations for Scientific Discovery
This Network+ is EPSRC Funded under Grant No: EP/S000356/1

Principal Investigator: *Professor Jeremy Frey*
Co-Investigator: *Professor Mahesan Niranjan*
Network+ Coordinator: *Dr Samantha Kanza*

Contents

1 Project Details **1**

2 Project Team **1**

2.1 Project Student 1

2.2 Project Supervisor 1

3 Lay Summary **1**

4 Aims and Objectives **1**

5 Methodology **2**

6 Results **3**

7 Conclusions & Future Work **9**

8 Acknowledgements **10**

1 Project Details

Title	Relative Structural Analysis on Molecular Perovskite
Project Reference	AI3SD-FundingCall3_007
Supervisor Institution	Queen Mary University of London
Project Dates	14/06/2021 - 20/08/2021
Keywords	Perovskite, Descriptors, SOAP, KMeans, Gaussian Mixture

2 Project Team

2.1 Project Student

Name and Title	Mr Kevin Daniel Calvache Ramos
Employer name / University Department Name	Dr Anthony Phillips, School of Physics and Astronomy
Work Email	k.d.calvache@se18.qmul.ac.uk

2.2 Project Supervisor

Name and Title	Dr Anthony Phillips
Employer name / University Department Name	School of Physics and Astronomy
Work Email	a.e.phillips@qmul.ac.uk

3 Lay Summary

Perovskite compounds have gained interest over the years because of their physical properties which include but are not limited to: superconductivity, ferroelectricity and magnetic properties. As such, these materials have found applications in microelectronics such as sensors, fuel cells, capacitors and solar cells. So, in understanding the structure perovskite compounds we can better understand the physical properties that a specific perovskite compound will have. One way to analyse atomic structure is to use descriptors, where it can take in a Crystallographic Information File (cif) and it's resultant output is a constant sized multi-dimensional vector. One such descriptor that has gained interest is the Smooth Overlap of Atomic Positions (SOAP) as it can be used in unsupervised machine learning. Thus we can learn how different/similar different compounds are and how they may form hidden clusters.

4 Aims and Objectives

In this project, I aim to use a descriptor to represent crystal structures and a decomposition method to visually see how the different crystal structure differ from one another. I also aimed to apply different clustering algorithms to find hidden clusters forming from a dataset of different perovskite compounds. Finally, I optimised the number of clusters that results in better models for the dataset I managed to acquire.

5 Methodology

Firstly, I had to make myself a dataset containing several different perovskite compounds. For my project I focused solely on ordered crystal structures at low temperature ranges, $\sim 100K - 298K$. I gathered a total of 56 different perovskite compound from a paper [1], divided into 3 main categories depending on their X-site ligands: formate-bridge, cyano-bridge and azido-bridge. From this paper, I had to trace back each crystal structure to its original paper origin (from the references) and download the relevant cif file. Once I had my dataset, using python 3.7 in jupyter notebooks, I put it through my code which read each of the 56 compounds and obtained a local descriptor by using SOAP from the Dscribe module that I imported. The way SOAP describes a crystal structure is that it encodes a local environment within an atomic structure by using an expansion of a gaussian smeared atomic density based on spherical harmonics and radial basis functions [2]. Its output is a partial power spectrum $\boldsymbol{p}$ whose elements are defined as follows [2]:

$$p_{nn'l}^{Z_1 Z_2} = \pi \sqrt{\frac{8}{2l+1}} \sum_m c_{nlm}^{Z_1}{}^* c_{n'lm}^{Z_2} \tag{1}$$

Where n and n' are indices for the different radial basis functions up to a parameter which you can set called n_{max}. l is the angular degree of the spherical harmonics up to another parameter called l_max and Z_1 and Z_2 are atomic species. The coefficients c_{nlm}^{Z} are defined as the following:

$$c_{nlm}^{Z} = \int \int \int_{R^3} g_n(r) Y_{lm}(\theta, \phi) \rho^Z(\boldsymbol{r}) dV \tag{2}$$

The $\rho^Z(\boldsymbol{r})$ is the gaussian smoothed atomic density for atoms with atomic number Z defined as:

$$\rho^Z(\boldsymbol{r}) = \sum_i^{|Z_i|} e^{-\frac{|\boldsymbol{r}-\boldsymbol{R_i}|^2}{2\sigma^2}} \tag{3}$$

With this descriptor, each different compound has a unique numpy array output of (n x 1050). Where n is each individual atoms/sites in the cif file and 1050 are the different features. In effect, I have 56 different (n x 1050) local descriptions describing different atomic structures. To compare two or more compounds with different number of elements, I take the average of each local descriptor. In other words, I turn every local description into a global description with the shape of a numpy array output of (1 x 1050). In my project I did 3 different types of averaging; first-entry, inner average and outer average. The last two types of averaging is supported by Dscribe where the inner average is taken over the sites before summing up the magnetic quantum number. Outer averaging instead averages over the power spectrum of individual sites. First-entry averaging is the name I gave where I simply take the row of the local descriptor that represents the B-site atom as that is normally the variable that changes (other than the different types of X-sites ligands explained earlier). This averaging is important because this entry corresponds to the metal, so that we are specifically probing the local environment about the B site.

entry (1st).png

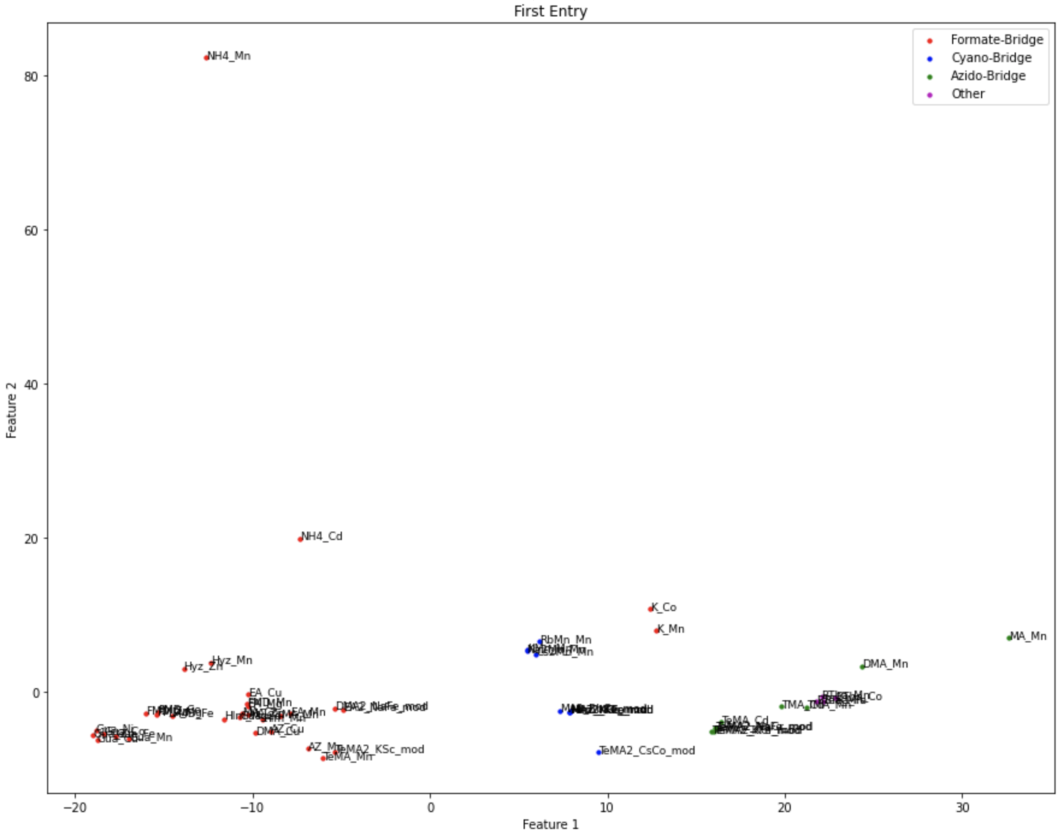


Figure 1: Plot of my 56 perovskite compound using 'first -entry' averaging method.

For each averaging method, I applied standard scalar (from sci-kit learn module) which scales the data in a way so that it has a mean of 0 and variance of 1. I then applied PCA (from sci-kit learn module [3]) which projects the data onto a lower dimensional space that shows greatest difference/variance between the data. In other words, instead of 1050 different features for each perovskite compound, I chose the features to be reduced to 2 so that we can visually see and plot them. In that way I could see how different/similar the compounds are to one another.

I also applied various clustering algorithms to each averaging dataset before applying any PCA reduction as these algorithms can handle multi-dimensional datasets. The cluster algorithms, also from the sci-kit learn module, I used include: MiniBatch-Kmeans, Affinity Propagation, MeanShift, Spectral Clustering, Ward, Agglomerative clustering, DBSCAN, OPTICS, Gaussian mixtures and BIRCH. Finally, I did clustering optimisation for MiniBatch-Kmeans and Gaussian mixtures using silhouette analysis and bayesian information criterion (BIC) respectively (also from sci-kit learn module, more information can be found on their respective documentation).

6 Results

Since I performed three different averaging methods, I obtained three different graphical plots for my 56 perovskite compounds. Starting with 'first-entry' averaging, see fig 1, we can see that there is some clustering between groups of same X-site ligands. However, this is not always the case. From fig 1, the compound $(NH_4)[Mn(HCOO)_3]$ seems to be an outlier and both $(K)[B(HCOO)_3]$, where $B = Co, Mn$, are within the other ligand domains.

average (1st).png

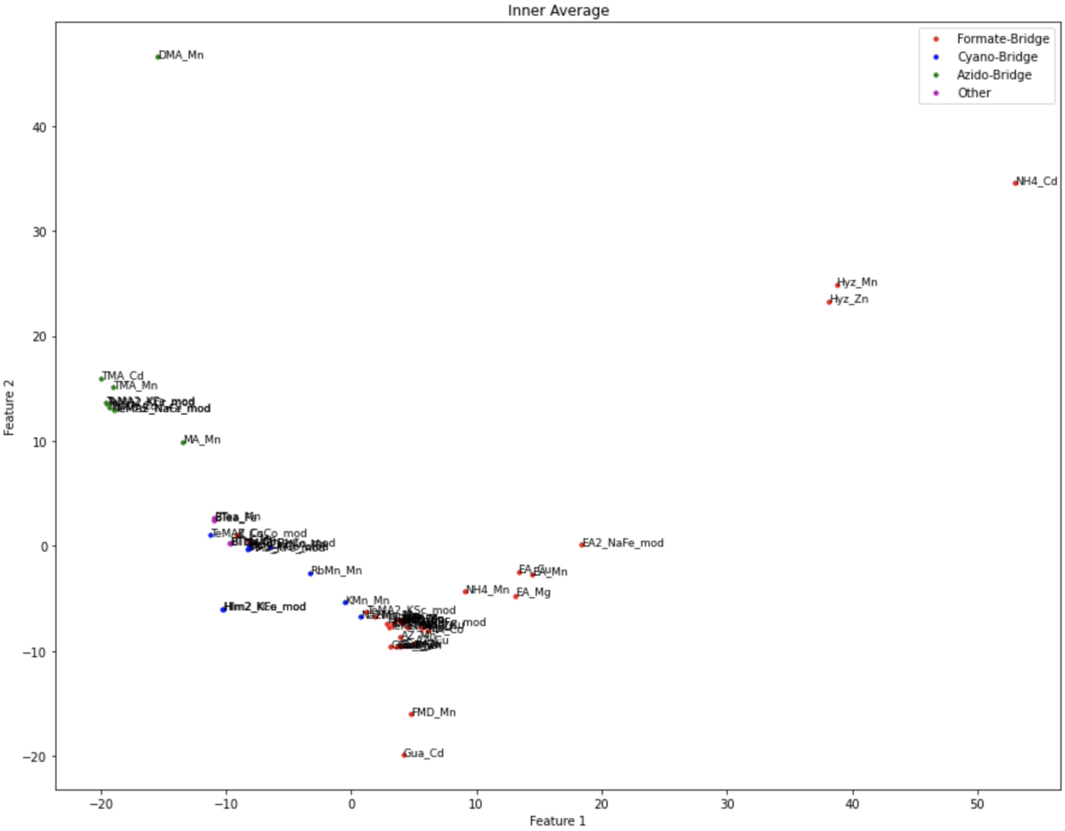


Figure 2: Plot of my 56 perovskite compound using inner averaging method.

With the inner averaging (see fig 2), we see a different pattern but nonetheless, most compounds with the same X-site ligands tend to group together. The same goes with the outer averaging method (see fig 3). However all three plots have different clustering patterns which indicates that their axis represents different properties/features.

With my three different datasets, I then performed the various different clustering algorithms I mentioned earlier to see how they compare with my initial three plots. Some cluster algorithms do have a parameter for the number of clusters you want to put your data in. For those, I put a value of four since my datasets are divided into 4 types of X-site ligands but this doesn't have to be the case as perovskite compounds can have differences/similarities regardless of different X-site ligands. Fig 4 shows my results. At the top of each column shows the clustering algorithm used and the first, second and third row shows the 3 different datasets ('first-entry', inner and outer average respectively). Most of the clustering algorithms agree with one another and most show subgroups within X-site ligands groups like for formate-bridged compounds and this is the case for all averaging methods used.

I chose to do optimisation on both the MiniBatch-KMeeans and Gaussian Mixture algorithms because the latter can be thought as a generalisation of KMeans where it includes information about the covariance structure of the data. MiniBatch-KMeans algorithm is a variant of KMeans which uses MiniBatch to reduce computational time at the cost of some difference in quality of results. I used silhouette analysis on MiniBatch-KMeans clustering, also supported by the sci-kit learn module, to find the optimal number of clusters by measuring how close each point in one cluster is to points in the neighboring clusters. It assigns a value for each point between $[-1, 1]$ where values close to 1 indicates a sample being far away from average (1st).png

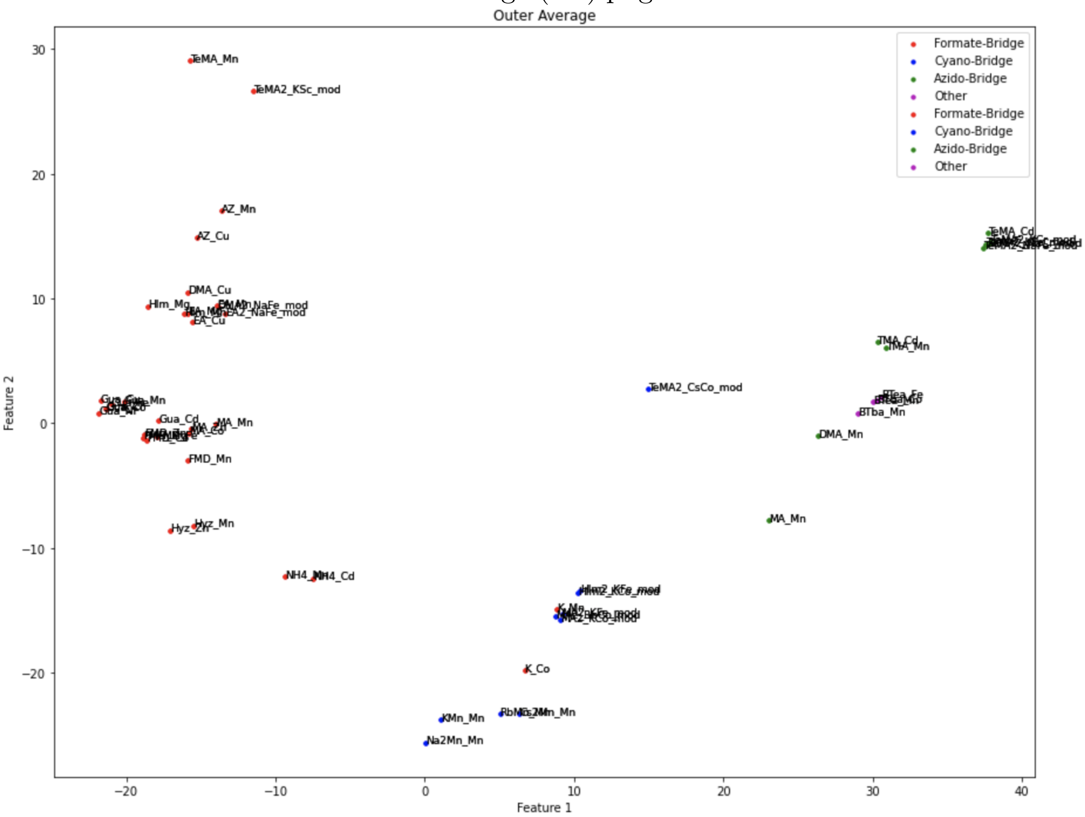


Figure 3: Plot of my 56 perovskite compound using outer averaging method.

(1st).png

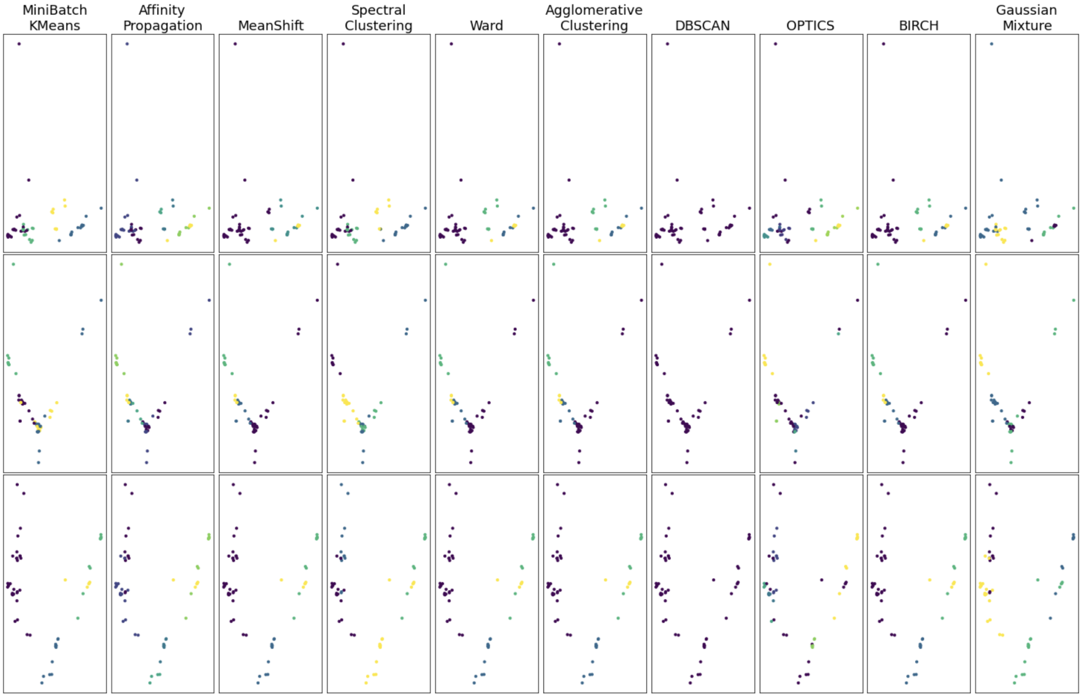


Figure 4: Applying different cluster algorithms to my multi-dimensional dataset first and then using PCA reduction to visually see how the algorithms clustered the data. The first, second and third row are for 'first-entry', inner and outer average respectively.

1st entry (1st).png

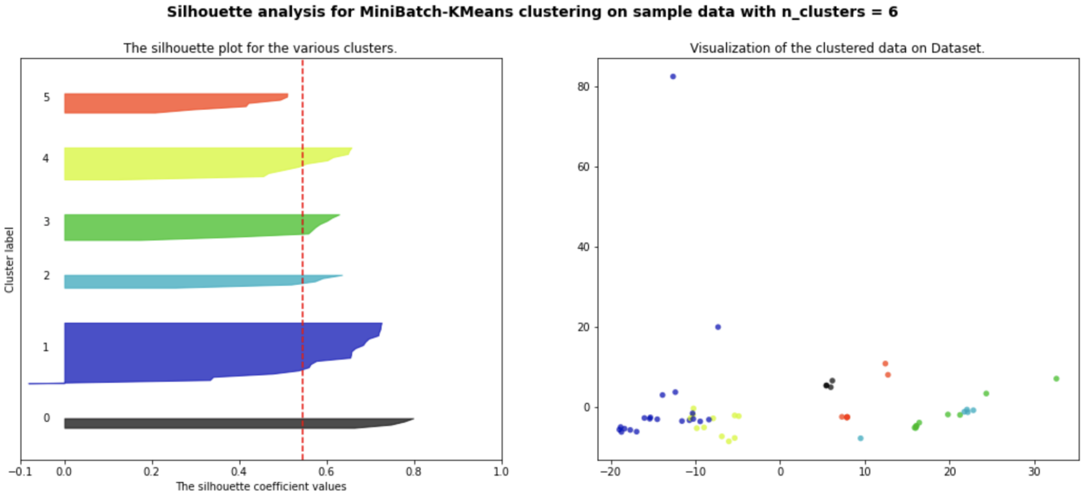


Figure 5: Graph of most optimal number of clusters for MiniBatch-KMeans algorithm using silhouette analysis for 'first-entry' averaging dataset with an average silhouette score of 0.5449464.

nearest neighbouring cluster and −1 indicates sample has a high probability of being assigned to a wrong cluster.

For fig 5 - 7, shows the highest silhouette score for the three datasets corresponding to the three different averaging methods. As you may notice, the highest score is always achieved with six clusters, instead of four, for all three averaging methods. This shows that there is high likelihood of sub-clusters within the dataset not previously known. The graphs to the left visually show the scores for each point and to which cluster they belong (the coloured horizontal bars). So, the larger the cluster, the more thicker the coloured bar it has. The red dotted line is the average silhouette score (see fig 5 - 7 for average scores) and it is a measure of how well overall the model works for dataset where an average score of 1 is considered the best. The graphs to the right just shows how the MiniBatch-Kmeans would group the dataset given the optimal number of clusters. In this case, for all 3 datasets, the optimal number of clusters is six. For the Gaussian mixture algorithm, I considered optimising two parameters. These were the covariance type and number of components (or equivalently, number of clusters). The number of components is self explanatory but for the covariance type there are four options: spherical, tied, diag and full. The covariance type dictates how each sample data produces its own covariance matrix and its relation to other data points (more information can be found on the scikit.mixture.GaussianMixture documentation). I used bayesian information criterion (BIC) to measure how good different combinations of the two parameter values were for the Gaussian mixture. BIC is a likelihood based function which asses how good different fitting models on datasets. So different models might increase the likelihood by adding parameters but this may cause overfitting. BIC overcomes this by introducing a penalty for the numbers of parameters in a model.

The selected parameters is indicated with a star on top of the model. For example, from fig 8, the best selected model is the 'diag' covariance type with 4 number of components (4 number of clusters). From fig 8 - 10 you can see the selected models for all averaging dataset. The top graphs shows how well different Gaussian mixture models fit a specific dataset. For each number of clusters (in the range of 1-6), all 4 different covariance types were applied on each dataset with their score illustrated as column boxes. The lower the score, the better the model in this case. The bottom graphs is a plot of the datasets with the selected/best model inner average (1st).png

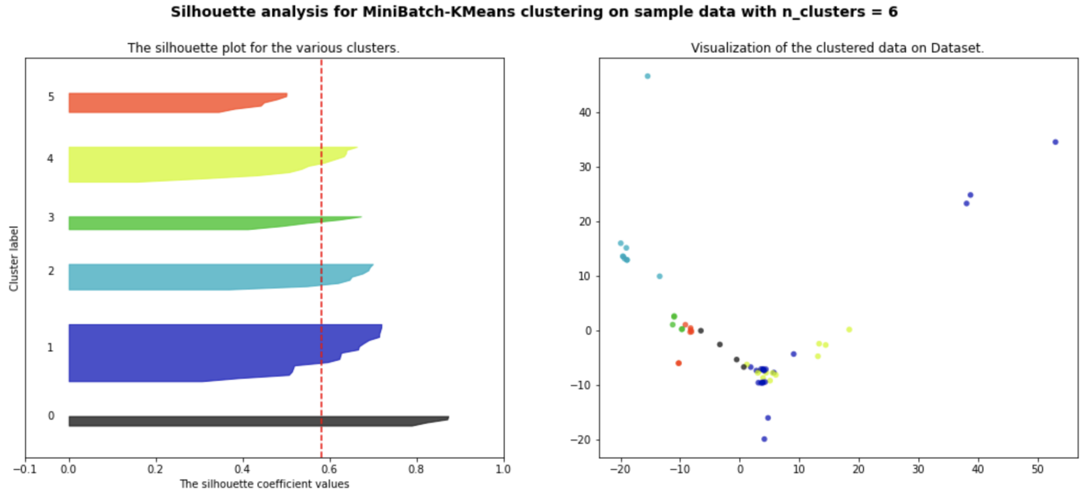


Figure 6: Graph of most optimal number of clusters for MiniBatch-KMeans algorithm using silhouette analysis for inner averaging dataset with an average silhouette score of 0.58103484.

outer average (1st).png

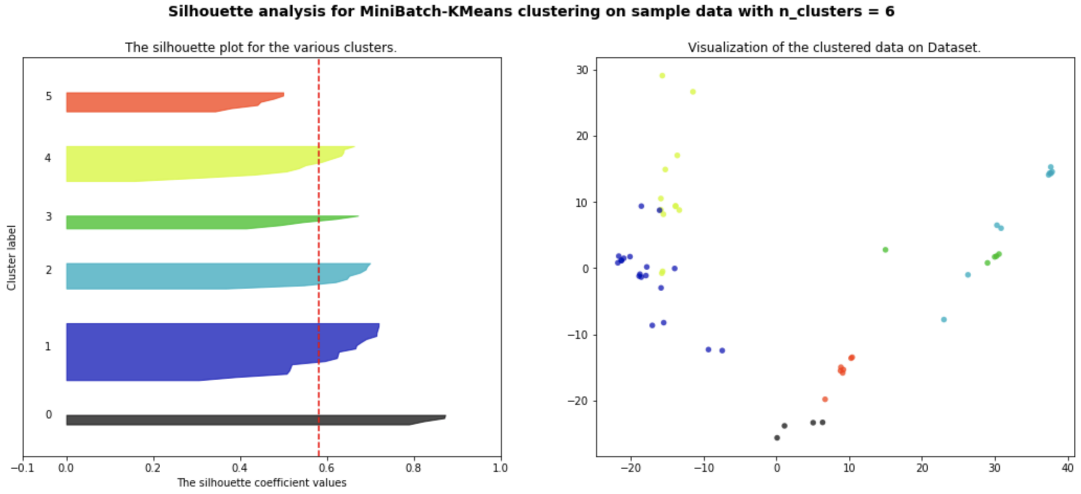
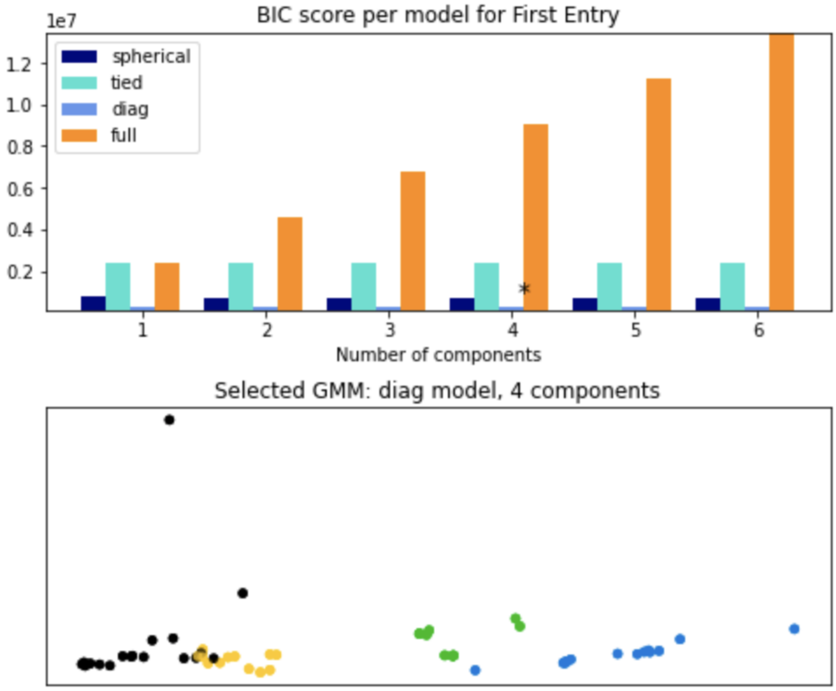
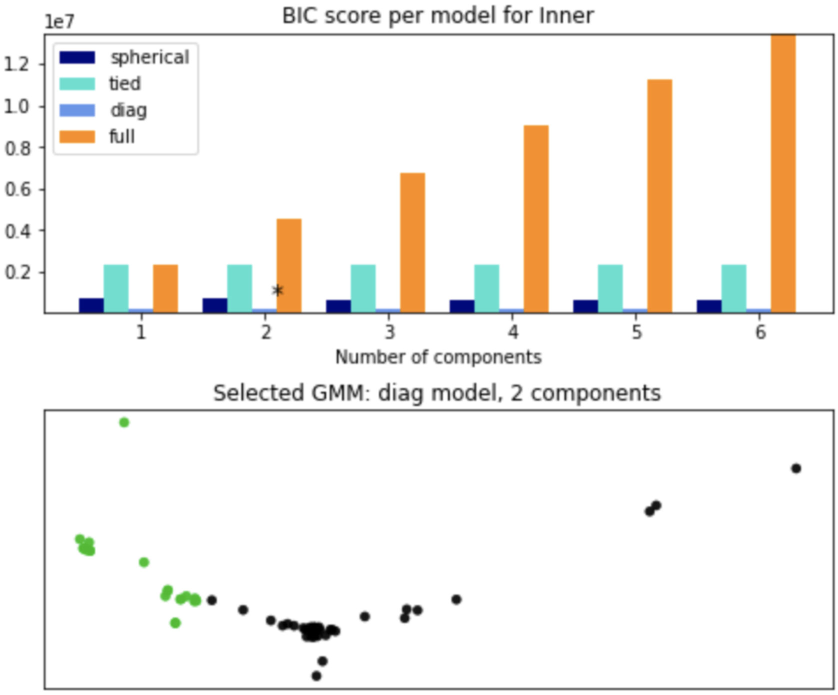


Figure 7: Graph of most optimal number of cluubters for MiniBatch-KMeans algorithm using silhouette analysis for outer averaging dataset with an average silhouette score of 0.5808732.



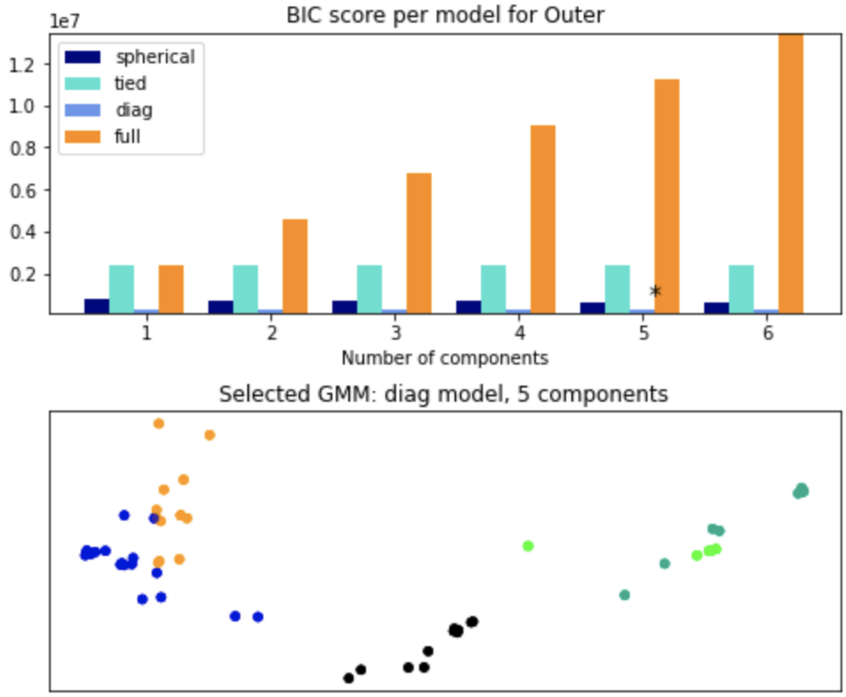
1st entry (1st).png

Figure 8: The first graph showing scores of different models on the 'first-entry' averaging dataset. The second graph showing the selected model. In this case, it is the 'diag' model with 4 components.



inner average (1st).png

Figure 9: The first graph showing scores of different models on the inner average dataset. The second graph showing the selected model. In this case, it is the 'diag' model with 2 components.



outer average (1st).png

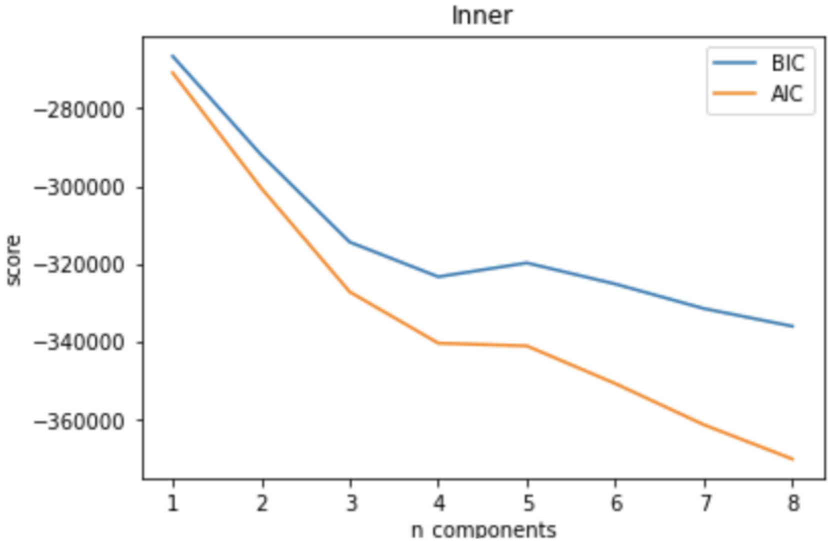
Figure 10: The first graph showing scores of different models on the outer average dataset. The second graph showing the selected model. In this case, it is the 'diag' model with 5 components.

chosen from the top graphs. In each case, they all have different numbers of clusters but they all agree with 'diag' as its covariance type meaning each component has its own diagonal covariance matrix. It is interesting to see how for the inner average, it found that two clusters was best, fig 9. I must point out that for the optimisation on the Gaussian mixture I had to change another parameter (different from the two I'm trying to optimise) for the different covariance types to work. I had to changed the default value of reg_covar=$1e-06$ to reg_covar=1 for my code to work and not run into problems. The reg_covar parameter is essentially a measure of how elongated or blobbed are the Gaussian probability distributions with the value of 1 being more blobbed/spherical.

After seeing fig 9, I did a separate analysis on my datasets where I set the reg_covar parameter as its default value (that being: $1e-06$) but this time I set the covariance type as 'diag' since all datasets agreed on that parameter being the best for it. Fig 11 shows my results for this extra analysis. Here I used also Akaike information criterion (AIC) where it tries to find unknown model that is mutli-dimensional whereas BIC comes across only true models. Both also penalise differently and more information can be found here [4]. Both AIC and BIC agree that the number of components should be four or 5 as seen from the slight dip in score.

7 Conclusions & Future Work

After reviewing my results, it is clear that there are sub-clusters within X-site ligands aswell as compounds with different X-site ligands being grouped in the same clusters. This could indicate perovskite compounds with same ligands have the same general properties but some might have extra properties due to there structure or atoms present in their structure. One downside to my project was that I could not get enough compounds for a better analysis since the tool I was meant to use was not working for the duration of my project. Clearly, the next step would



inner average (1st).png

Figure 11: BIC and Akaike information criterion (AIC) used to asses different number of components when using Gaussian mixture on inner average dataset with the covariance type parameter set to 'diag'.

be to acquire more perovskite compounds run through the same code to see the bigger picture of relative structural differences/similarities between all compounds.

Also, once we have more compounds, further analysis needs to be carried out to figure out what thee axis represent in terms of chemical, atomic or geometric properties. Finally, disordered perovskite compounds in high temperature phase could be studied on its own and then compare it with there low temperature phase counterpart and see how perovskites change behaviour in different temperatures.

8 Acknowledgements

I would like to thank Dr Anthony Phillips for guiding me through a very interesting project and helping me thoughout. I would also like to thank the AI3SD network and PSDS for making the internship possible where I have learnt so much and plan to use it in the future.

References

(1) Xu, W.-J.; Du, Z.-Y.; Zhang, W.-X.; Chen, X.-M. *CrystEngComm* **2016**, *18*, 7915–7928.

(2) Himanen, L.; Jäger, M. O. J.; Morooka, E. V.; Federici Canova, F.; Ranawat, Y. S.; Gao, D. Z.; Rinke, P.; Foster, A. S. *Computer Physics Communications* **2020**, *247*, 106949.

(3) Pedregosa, F. et al. *Journal of Machine Learning Research* **2011**, *12*, 2825–2830.

(4) Kuha, J. *Sociological Methods & Research* **2004**, *33*, 188–229.