NANOx81Review

Shyue Ping Ong

University of California, San Diego

Fall 2022

Overview

1. Review
2. Final Lab

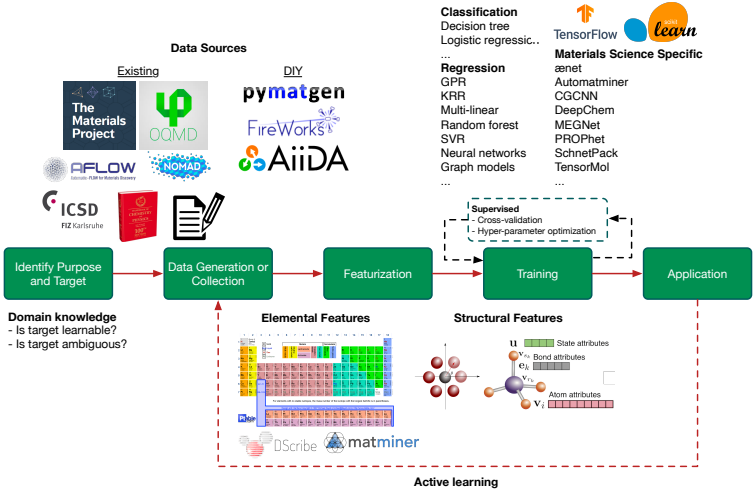
Preliminaries

- In this class, we have provided a broad survey of data science techniques as they are applied to materials science problems.
- In doing so, we have opted for breadth rather than depth, and eschewed most of the mathematical derivations that is typical in data science classes.
- In this final lecture, I want to do an overview of the coverage to reinforce the lessons learnt.
- The structure of this overview is deliberately different from the sequence in which we have done the entire class. Now that you have the details in your mind, this lecture provides a high-level overview of key takeaways and guiding principles.

Why Data Science?

- Materials data is exploding in quantity and quality.
  - High-throughput/combinatorial experiments.
  - High-throughput first principles computations.
- Types of problems that can most benefit from data science:
  - Things that are still too difficult to perform experiments on or compute.
  - Relationships that are beyond our understanding (at the present moment).

ML flowchart



Data and Featurization

- Data generation, collection and wrangling is typically the most time-consuming portion of the whole process.
- Sources: Experimental (ICSD, Pauling file, literature, ...), Computational (Materials Project, OQMD, AFLOW, NOMAD, ...).
- Quality and quantity remains a big issue in materials science.
- Featurization - typically the choice that affects model performance the most
  - Composition-based features (e.g., electronegativity, atomic radii, etc.): intuitively simple, readily available, but clearly cannot capture structural differences.
  - Structure-based features (including graphs): much greater complexity, typically must obey symmetries of system to be effective.

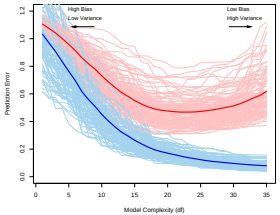
Model fitting

- All model fittings follow the same basic principle - minimizing of some *loss function*, $L(\theta; y_i, \mathbf{x_i})$ either analytically or by numerical procedures (e.g., gradient descent).

Task	Loss function	Equation
Regression	Mean squared error	$\sum_{i=1}^{N}(y_i - f(x_i))^2$
	Mean absolute error	$\sum_{i=1}^{N} \lvert y_i - f(x_i)\rvert$
Classification	Missclassification rate	$I(\text{sign}(f) \neq y)$
	Exponential loss	$e^{-yf(x)}$
	Binomial/multinomial	$-\sum_{k=1}^{K} I(y = G_k) f_k(x) + \log\left(\sum_{l=1}^{K} e^{f_l(x)}\right)$
	Binomial deviance	$\log(1 + e^{-2yf})$

Model assessment and selection

- Model performance is related to its performance on *independent test data*.
  - Training error: L over training set.
  - Test error: L over independent test set.
- Model complexity increases as the number of parameters increases.
- Training errors **always** decrease with increasing model complexity.
- Test errors are high when model complexity is too low (underfitting) or too high (overfitting).



Training, validation and test data

- Model selection: estimating the performance of different models in order to choose the best one.
- Model assessment: having chosen a final model, estimating its prediction error (generalization error) on new data.
  - Training set: For training the model.
  - Validation set: For estimating prediction error to select the model.
  - Test set: For assessing the generalization error of the final model. This should not be used in fitting the model.
- Typical training:validation:test split is 50:25:25 or 80:10:10, or in very data-poor situations, maybe even 90:5:5.

K-fold cross validation (CV)

- Simplest and most widely used approach for model validation.
- Data set is split into K buckets (usually by random).
- Typical values of K is 5 or 10. $K = N$ is known as "leave-one-out" CV.

Train	Train	Validate	Train	Train

- CV score is computed on the validate data set after training on the train data:

$$CV(\hat{f}^{-k(i)}, \alpha) = \frac{1}{N_{k(i)}} \sum_{i=1}^{N_{k(i)}} L(y_i, \hat{f}^{-k(i)}(x_i, \alpha))$$

- assuming the k^{th} data bucket has $N_{k(i)}$ data points and $\hat{f}^{-k(i)}$ refers to the model fitted with the k^{th} data left out.

Regularization

- Often, one starts from a model that is more complex first, and then reduce model complexity.
- Reducing model complexity decreases risk of overfitting (improve generalizability).
  - Shrink coefficients or weights, sometimes to zero.
  - Subset selection / tree pruning.
- General approach is to add a *penalty term* to loss functions that penalizes overcomplex models, e.g., sum of squares or absolute value of coefficients (e.g., MLR) or weights (NNs), tree size (decision trees), etc. Controlled by some parameter to be specified by model developer that determines size of penalty.
- Bias-variance trade-off:

$$\mathrm{MSE} = \mathrm{var}(\hat{\theta}) + [E(\hat{\theta}) - \theta]^2$$

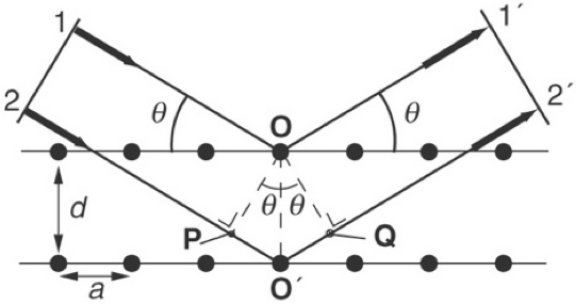
Types of ML models

- Supervised learning
  - Linear (MLR, Ridge, Lasso, Linear/Quadratic Discriminant)
  - Local/Kernel methods (kNN, kernel density estimation, Gaussian mixture models)
  - Trees (CART, Adaboost, Gradient Boosting, Random Forest)
  - Neural networks
- Unsupervised learning
  - Principal Component Analysis
  - K-means
  - Hierarchical clustering
  - DBSCAN
- Most models can be made more flexible through basis expansions (polynomial, Gaussian, cubic splines, etc.)
- Many other models have not been covered (e.g., support vector machines, graph-based models, etc.)

Final Lab

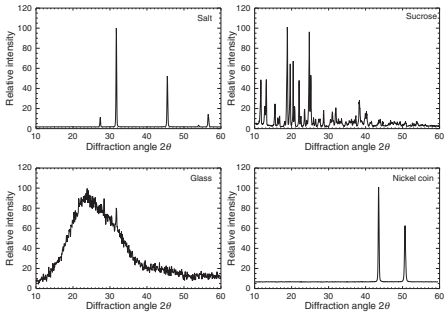
- Working on an **open** problem in materials science.
- Will be held as a Kaggle competition - see Canvas for link.
- If the results are sufficiently good, the whole class will write a journal article on the results.
- Top three results within 10% in accuracy of each other will be co-first authors. All students will be co-authors.
- Note that the actual results have no impact on grades - for grading purposes, we are looking at proper application of data science techniques for the problem, not raw accuracy. Of course, if the accuracy is far lower than what others have obtained, it typically means you are not doing something correctly.

X-ray diffraction



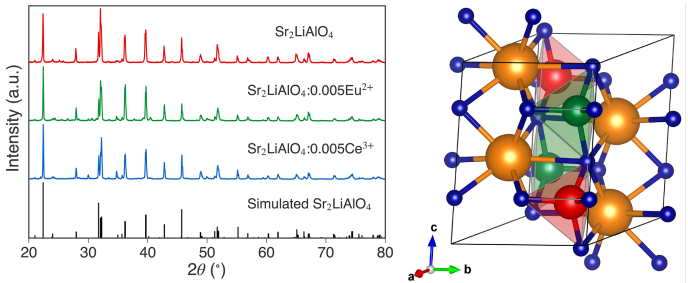
- X-ray diffraction (XRD) is a technique for determining the structure of crystals and is frequently used for phase identification. Planes in a crystal cause a beam of incident X-rays to diffract into many specific directions.

XRD pattern



- Peak positions (given in 2θ) are given by Bragg's equation: $2d \sin\theta = \lambda$, where λ is the wavelength of the X-ray and d is the interplanar spacing.
- Peak intensities are given the nature of the atomic species and the presence of various symmetries (e.g., centering atoms).

The challenge



- Computing the XRD pattern given a crystal structure is easy.
- Inverse problem of predicting crystal structure given an XRD pattern is **very hard** - phase identification is usually done by matching to database of patterns of *known* crystals using Rietveld refinement (see earlier lecture).
- Question: Can we use ML to directly predict crystal structure from a given XRD pattern?

Is it doable?

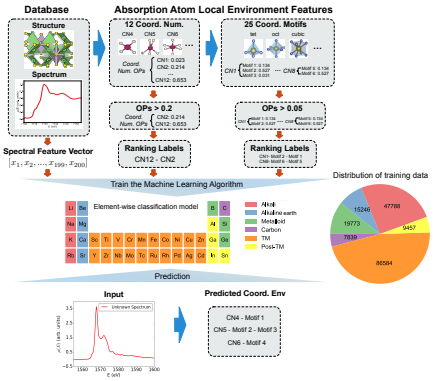


Figure: Workflow for classification of K-edge XANES spectra into one of 25 coordination environments.[1]

Is it doable, contd?

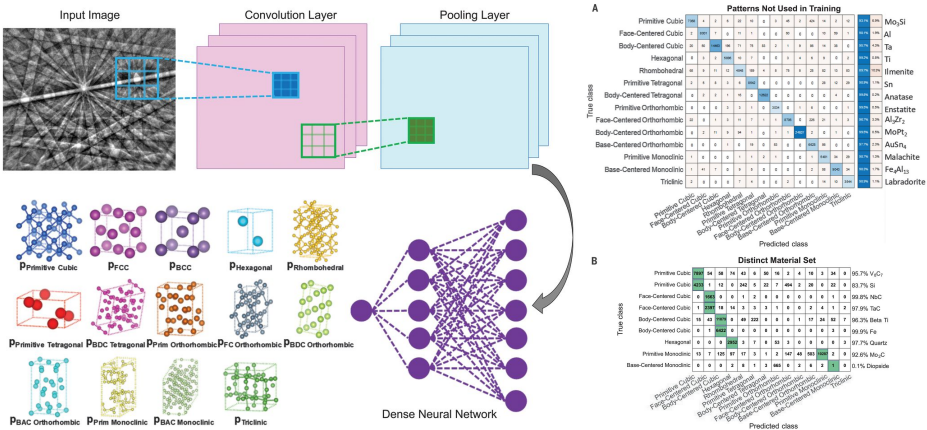
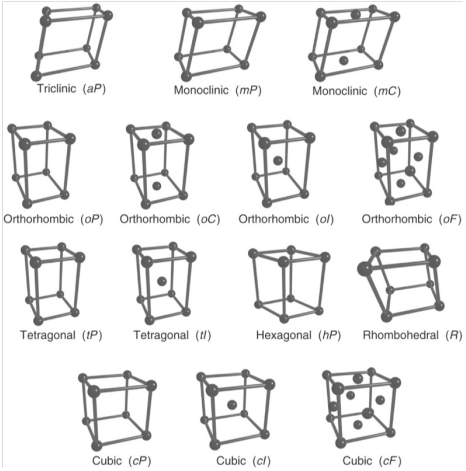


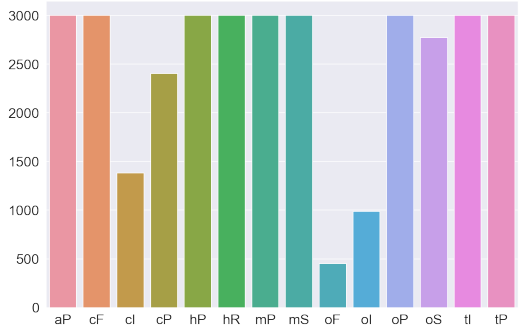
Figure: CNNs to determine crystal symmetry from electron diffraction patterns with $> 90\%$ accuracy.[2]

The 14 3D Bravais lattices



Dataset

- $\sim$ 42,000 crystals from Crystallography Open Database
- We have already sampled the crystals in a way to somewhat equalize the number of crystals in each Bravais lattice type (note that the actual distribution in the literature is highly biased towards certain lattices).



Dataset, contd.

cod_id	formula	spacegroup_number	Bravais_lattice	y0	y1	y2	y3	y4	y5	y6	...	
7018382	CoH34C36(N3O4)2	2	aP	4.805371e-224	1.320895e-190	7.009208e-160	7.180078e-132	1.419871e-106	5.420359e-84	3.994540e-64	...	0.02
4083141	PH33RhC30Se2ClF6	2	aP	7.333743e-146	1.641510e-119	7.092850e-96	5.916401e-75	9.526949e-57	2.961485e-41	1.777153e-28	...	0.03
7216537	H12C14I(NO2)3	2	aP	0.000000e+00	3.290641e-285	2.474378e-247	3.591791e-212	1.006504e-179	5.444771e-150	5.685951e-123	...	0.05
4514227	H31C31N3O4	2	aP	7.706884e-298	4.222192e-259	4.480743e-223	9.222289e-190	3.687103e-159	2.869166e-131	4.356285e-106	...	0.02
7212013	H24C10SN3Cl3O14	2	aP	3.295902e-256	1.694053e-220	1.680889e-187	3.219666e-157	1.190535e-129	8.498313e-105	1.171070e-82	...	0.04

Task

- Develop a ML model to classify an XRD pattern into one of the 14 Bravais lattices.
- Experiment with any of the ML models that you have learnt, and play around with various parameters.
- **Hint:** use smaller sample of 10,000 data points to experiment with different models first, before using full data set to do serious model building and optimization.
- Show all parameter optimizations carried out, e.g., grid search. Demonstrate the usage of at least two types of ML model.
- Submission should be done in two places:
  - Kaggle, which will have a leaderboard.
  - Google classroom, where you will submit your Jupyter notebooks.
- Discuss the results you have obtained, offering interpretations of how your model is achieving the performance claimed. Compare the results you have obtained with what would have been achieved based on random guessing. Is the ML model doing anything useful?

Evaluation Criteria

- Grading will be done on the basis of the proper use of data science techniques, e.g., separation of training and test data, validation, parameter search and optimization, coding style and materials science insights. Raw accuracy is not the goal for grading.
- Competition evaluation will be based on robust performance achieved.

Next steps

- Sign up for Kaggle account.
- Download the dataset.
- Start building your models.
- Submission: Please submit at least one model

Optional challenge problem

- Instead of classification into 14 Bravais lattices, can an ML algorithm classify an XRD pattern into one of 230 space groups?

Bibliography

Chen Zheng, Chi Chen, Yiming Chen, and Shyue Ping Ong.
Random Forest Models for Accurate Identification of Coordination Environments from X-Ray Absorption Near-Edge Structure.
Patterns, page 100013, April 2020.

Kevin Kaufmann, Chaoyi Zhu, Alexander S. Rosengarten, Daniel Maryanovsky, Tyler J. Harrington, Eduardo Marin, and Kenneth S. Vecchio.
Crystal symmetry determination in electron diffraction using machine learning.
Science, 367(6477):564–568, January 2020.

The End