Tutorial DS00: Machine Learning in Materials Sciences — From Basic Concepts to Active Learning

Arun Mannodi Kanakkithodi

School of Materials Engineering
Purdue University, West Lafayette, IN

Sunday, Nov 27, 2022, 8.00 pm – 5:00 pm

Email: amannodi@purdue.edu

Tutorial Outline

8.00 am – 9.30 am, Arun Mannodi Kanakkithodi:
Introduction to ML for materials science. Demonstration of training ML models using a DFT dataset of halide perovskite alloys.

9.30 am – 10.00 am: Break

10.00 am – 10.30 am, Austin McDannald:
Gaussian Process Regression: Detailed description using examples.

10.30 am – 11.30 am, Austin McDannald:
Discussion of active learning, Bayesian optimization, autonomous experiments.

11.30 am – 12.00 pm: General Discussion

12.00 pm – 1.30 pm: Lunch Break

1.30 pm – 2.30 pm, Saaketh Desai:
Overview of neural networks for prediction, convolutional neural networks for images.

2.30 pm – 3.00 pm: General Discussion

3.00 pm – 5.00 pm (and beyond):
Battery Informatics and ML Hackathon: A battery dataset will be assigned to the contestants, who will apply some of the methods discussed during the tutorial and use available scripts to train models and make predictions. Please fill this sign-up form if interested and bring your laptop to the in-person launch at 3pm.

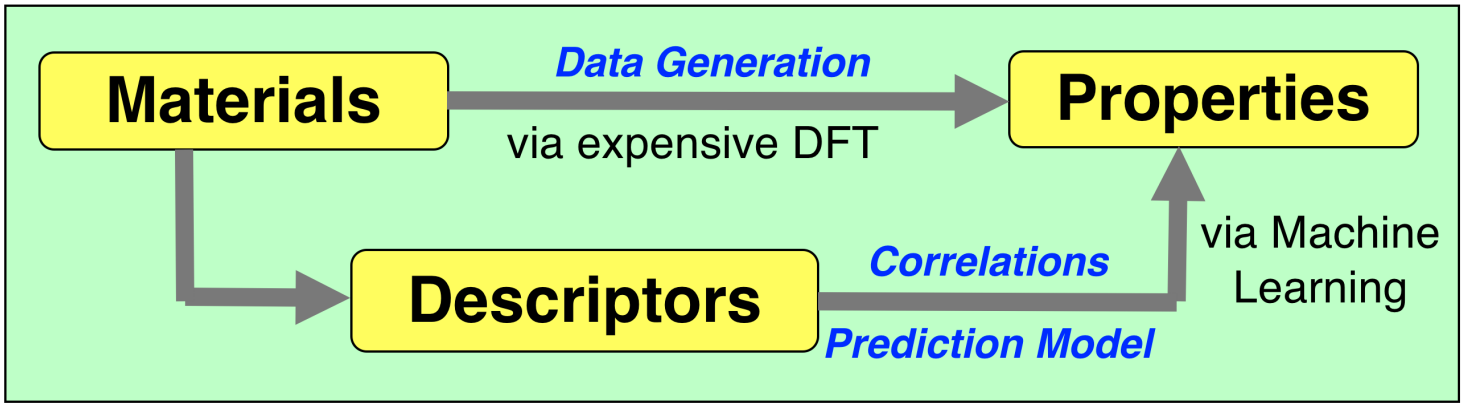
Tutorial Presenters

- Arun Mannodi Kanakkithodi: Assistant Professor, Materials Engineering, Purdue University. Computational materials scientist using high-throughput DFT and ML for materials design.
- Saaketh Desai: Postdoctoral Researcher, Sandia National Laboratory
- Austin McDannald: Materials Research Engineer, National Institute of Standards and Technology
- Shijing Sun: Research Scientist, Energy and Materials, Toyota Research Institute

PART 1:

Introduction to ML in Materials Science

Machine Learning in Materials Science



a Example dataset

Material	Property Value
Material 1	P_1
Material 2	P_2
⋮	⋮
Material *N*	P_N

b The learning problem

Material	Property Value
Material *X*	?

c Fingerprinting, learning and prediction

Material	Fingerprint	Property Value
Material 1	$F_{11}, F_{12}, \ldots F_{1M}$	P_1
Material 2	$F_{21}, F_{22}, \ldots F_{2M}$	P_2
⋮	⋮	⋮
Material *N*	$F_{N1}, F_{N2}, \ldots F_{NM}$	P_N

Fingerprinting

Learning

$f(F_{i1}, F_{i2}, \ldots, F_{iN}) = P_i$

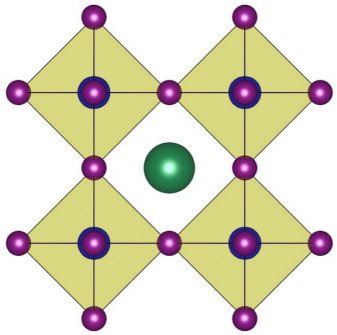
Prediction Model

Key Ingredient of ML: Feature Vectors / Materials Descriptors / Fingerprints

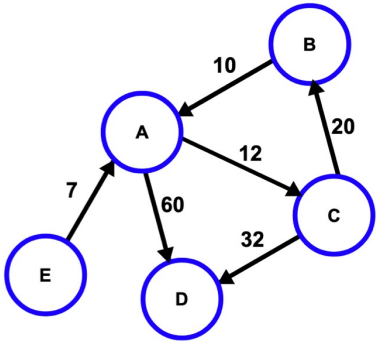
- Numerical representation of materials, input to ML.
- Definition depends on: a) application, b) domain expertise, and c) accuracy desired.
- Requirements: a) intuitive and inexpensive to calculate, b) generalizable to every material in the chemical space, and c) invariant to translation / rotation / permutation of like elements.

Examples of Fingerprints

3D geometry:
Atom i = (Z,x,y,z)



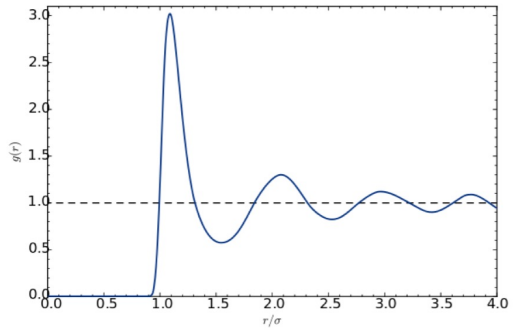
Weighted graph:
atoms & bonds



Coulomb Matrix

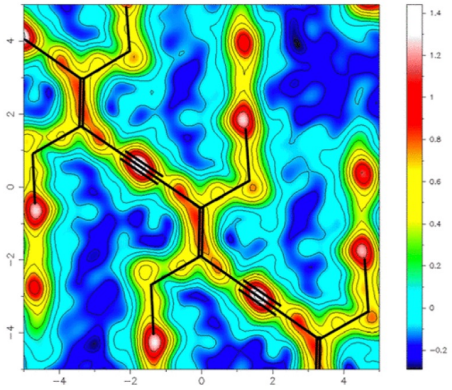
$$M_{IJ} = \begin{cases} 0.5 Z_I^{\,2.4} & \text{for } I = J \\ \dfrac{Z_I Z_J}{\left| \boldsymbol{R}_I - \boldsymbol{R}_J \right|} & \text{for } I \neq J \end{cases}$$

Radial Distribution Function

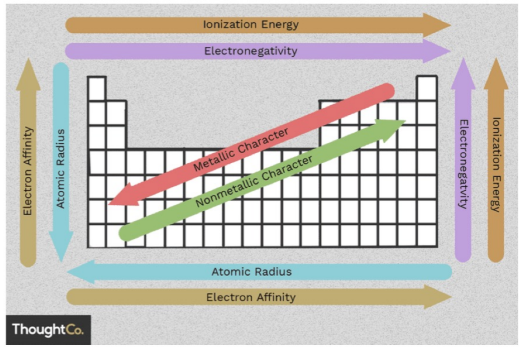


$A_i(\eta) = \exp(-r_{ij}^2/\eta^2)*f(r_{ij})$

Electron Density Distribution

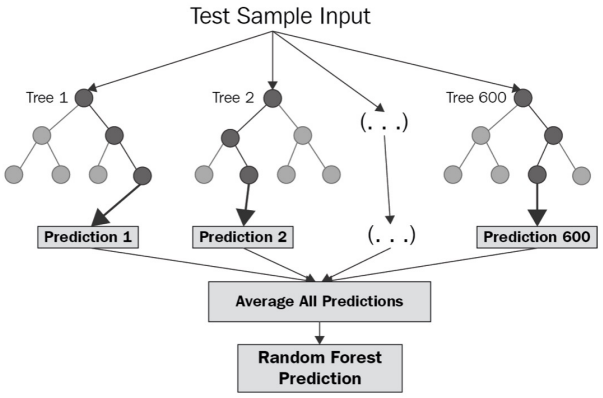


Tabulated elemental properties

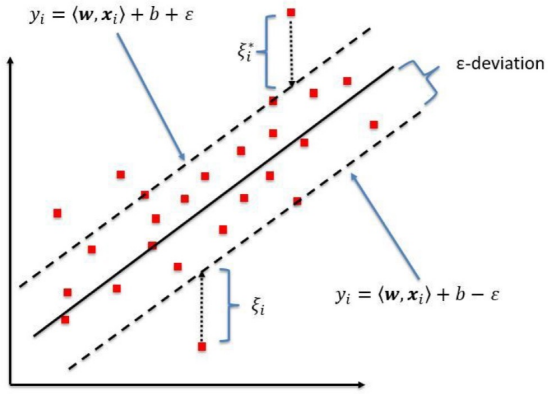


Examples of ML Techniques

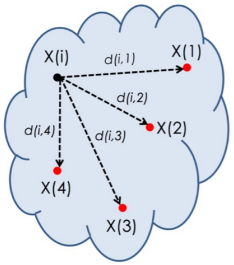
Random Forest Regression



Support Vector Regression



Kernel Ridge Regression



$X(i) = \{x_1, x_2, x_3 \ldots x_m\}$

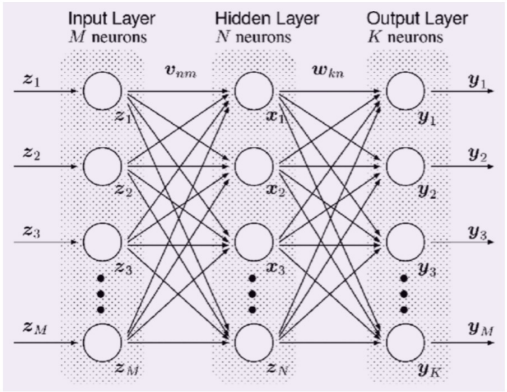
Measure of Similarity: Euclidean Distance

$$d(i,j) = \sqrt{(x_{i1} - x_{j1})^2 + \ldots + (x_{im} - x_{jm})^2}$$

Property = Weighted sum of Gaussians

$$f(i) = \sum_{k=1}^{N} a_k \cdot \exp\left(-\frac{1}{2\sigma^2} \cdot [d(i, i_k)]^2\right)$$

Neural Networks



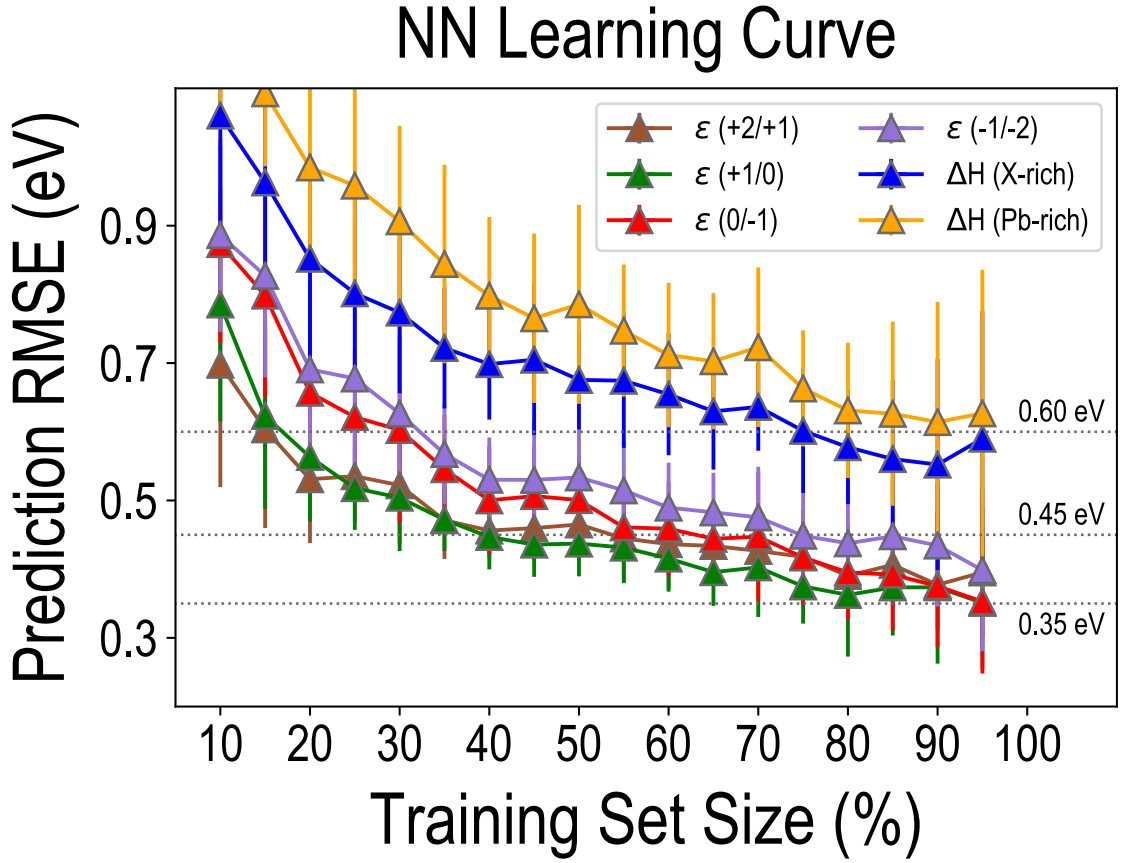
Types of Machine Learning

- Supervised learning: From labeled training data, find the unknown function connecting known inputs to unknown outputs, based on extrapolation of patterns.
- Unsupervised learning: Find patterns in unlabeled data, leading to clustering of samples.
- Semi-supervised learning: Representations learned from a mix of unlabeled and labeled data.
- Reinforcement learning: Finding optimal or sufficiently good actions for a situation to maximize a reward.

Nuts and Bolts of ML

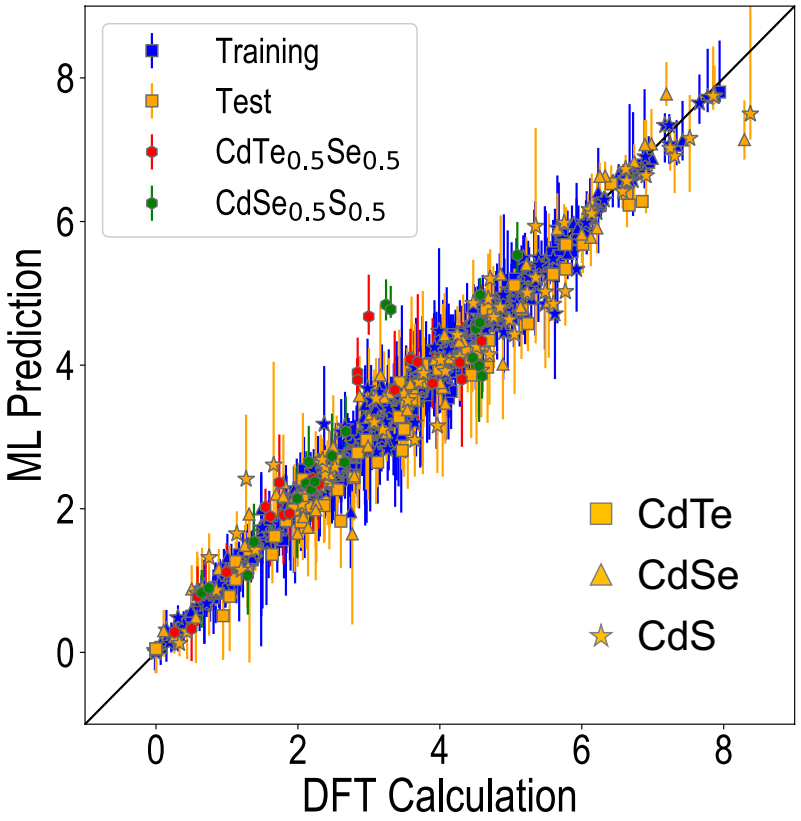
- Data: Divide into training, validation, and test sets.
- Descriptors: enumerate for all data points, perform dimensionality reduction and feature scaling.
- Cross-validation: n-fold, leave-one-out.
- Hyperparameter optimization: grid-search, Bayesian.
- Best model: optimized w.r.t. training data and descriptor size, CV, and HPO; choose error definition.
- Quality and quantity of data and descriptors: prevent underfitting. CV, HPO, regularization: prevent overfitting.

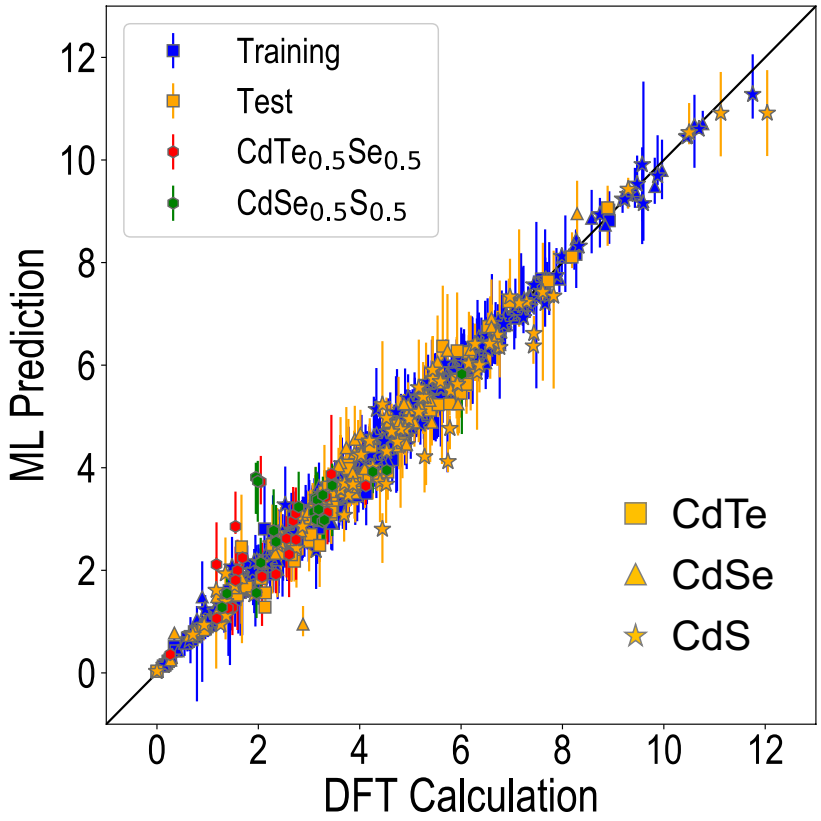
Training Data Size: Learning Curves



Iteratively change training set size (and descriptor dimensions) until test error saturates → learning curve.

Example: ML Regression

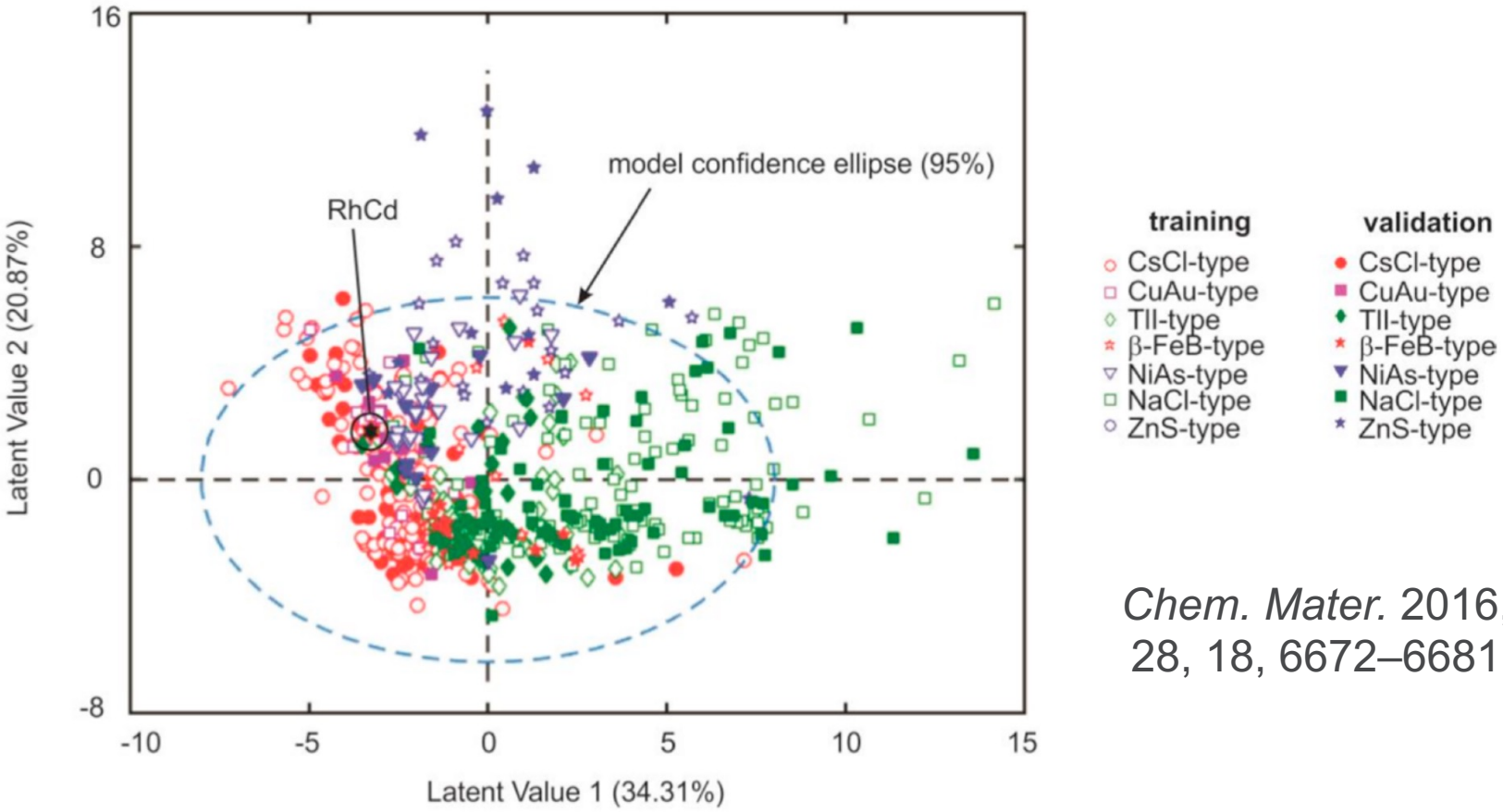




Predicting defect formation energy in semiconductors.

npj Comput Mater. **6,** 39 (2020)

Example: ML Classification



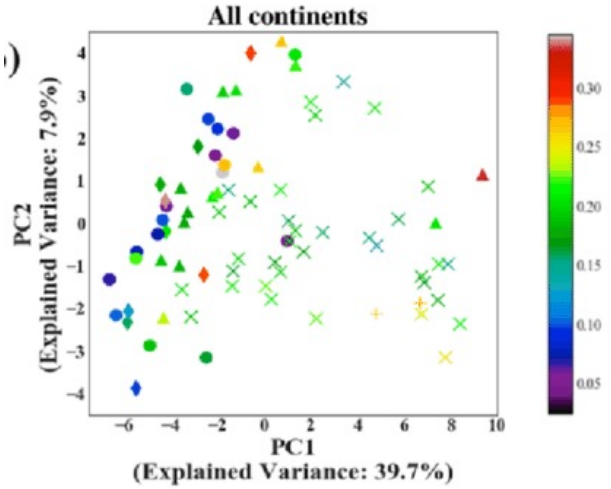
Chem. Mater. 2016, 28, 18, 6672–6681

Classifying AB compounds into different structure types.

Principal Component Analysis

- Projecting data into a hyperplane that lies closest to the data. Used for dimensionality reduction and clustering.

- Convert m-dimensional data into m orthogonal principal components that capture largest to lowest amount of variance in the training data.



- Ideal scenario: PC1 and PC2 together capture all the variance. This allows for real 2D projection and visualization.

- import sklearn.decomposition.PCA

Linear Regression

- Linear regression model prediction form:
  $y = \boldsymbol{\theta}_0 + \boldsymbol{\theta}_1 x_1 + \boldsymbol{\theta}_2 x_2 + \ldots + \boldsymbol{\theta}_m x_m$; or, $y = h_{\boldsymbol{\theta}}(x) = \boldsymbol{\theta}^T.\mathbf{x}$

- MSE cost function for a linear regression model:

$$\text{MSE}(\mathbf{X}, h_\theta) = \frac{1}{m} \sum_{i=1}^{m} \left(\theta^T \cdot \mathbf{x}^{(i)} - y^{(i)}\right)^2$$

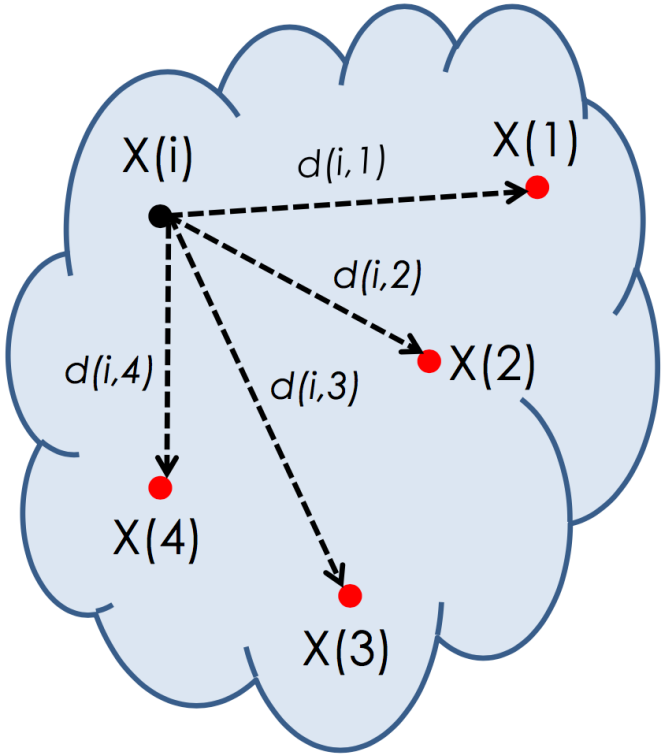
- Direct estimation of $\boldsymbol{\theta}$ using the formula below (indirect method uses GD):

$$\hat{\theta} = \left(\mathbf{X}^T \cdot \mathbf{X}\right)^{-1} \cdot \mathbf{X}^T \cdot \mathbf{y}$$

- Easy extension to Polynomial regression.

Kernel Ridge Regression

Chemical Space



$X(i) = \{x_1, x_2, x_3 \ldots x_m\}$

KERNEL RIDGE REGRESSION (KRR)

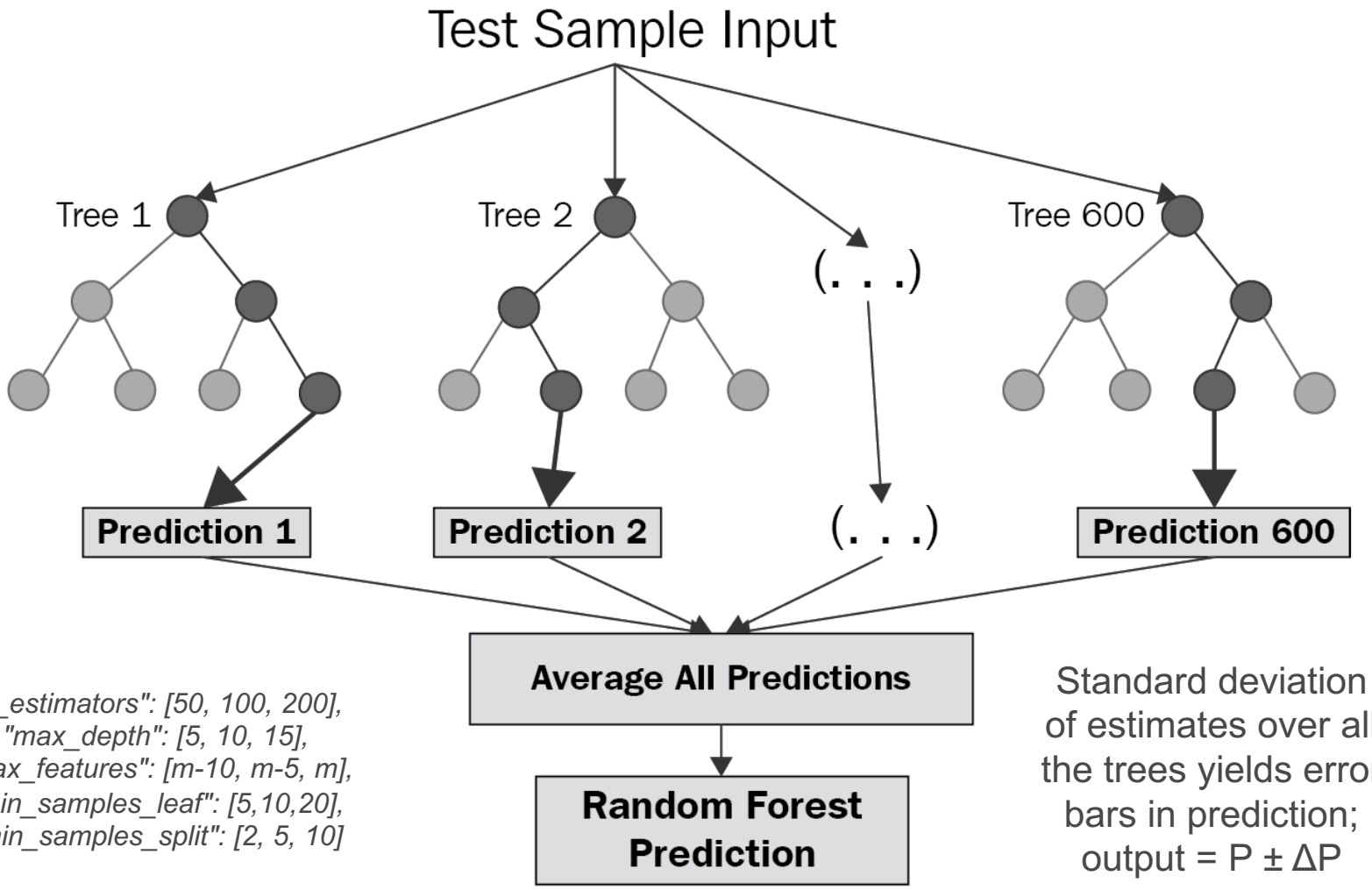
Measure of Similarity: Euclidean Distance

$$d(i,j) = \sqrt{(x_{i1} - x_{j1})^2 + \ldots + (x_{im} - x_{jm})^2}$$

Property = Weighted sum of Gaussians

$$f(i) = \sum_{k=1}^{N} a_k \,.\, \exp\left(-\frac{1}{2\sigma^2} \,.\, [d(i, i_k)]^2\right)$$

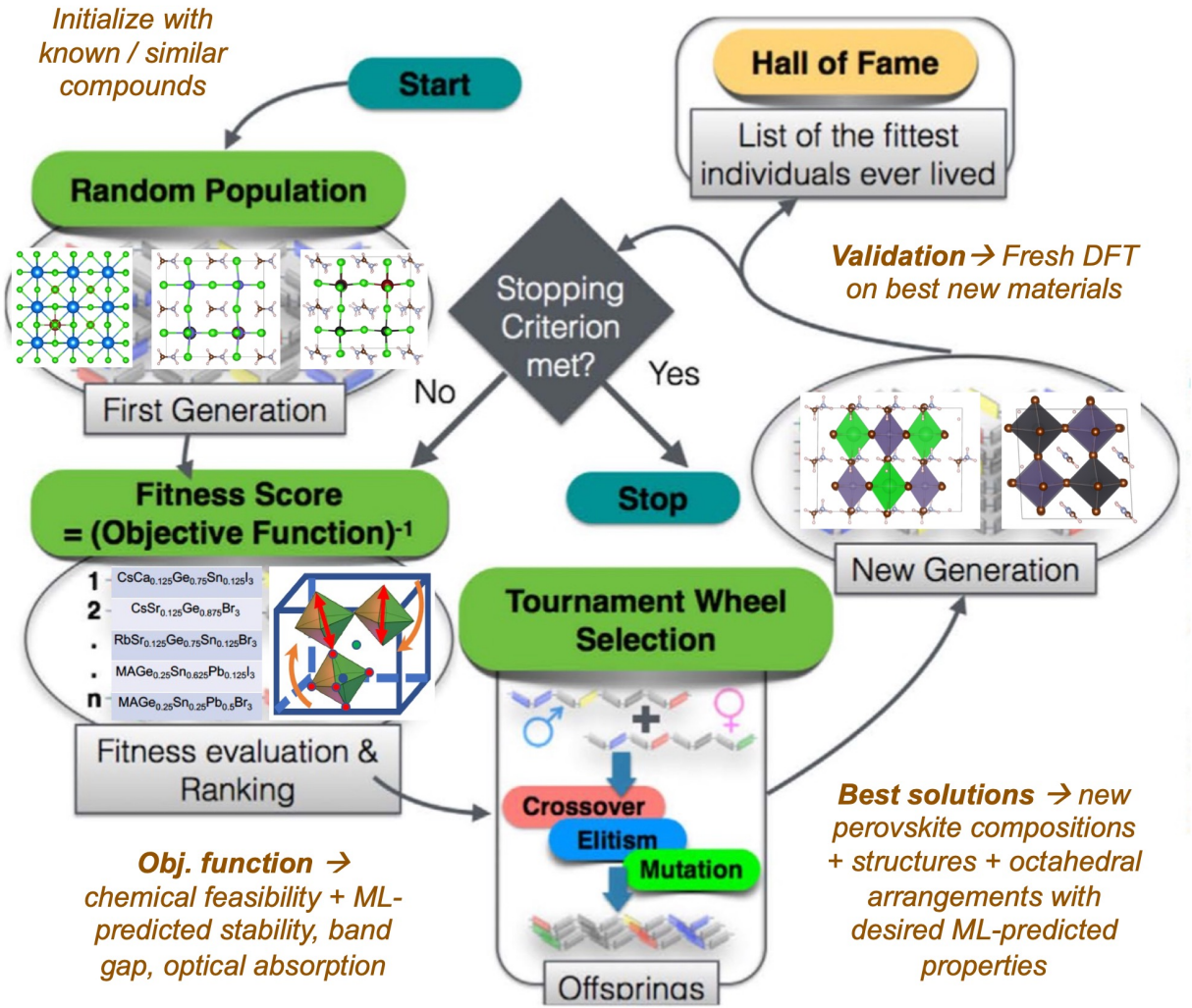
Random Forest Regression



"n_estimators": [50, 100, 200],
"max_depth": [5, 10, 15],
"max_features": [m-10, m-5, m],
"min_samples_leaf": [5,10,20],
"min_samples_split": [2, 5, 10]

Standard deviation of estimates over all the trees yields error bars in prediction; output = P ± ΔP

Genetic Algorithm



PART 1 SUMMARY

- Definition and basic principles of materials informatics.
- Key steps in ML: materials data, materials descriptors, techniques for regression and classification.
- PCA, linear regression, RFR, KRR, GA.
- NEXT: A detailed supervised learning case study → predicting (DFT computed) properties of a chemical space of halide perovskite alloys.

PART 2, CASE STUDY: Data-Driven Design of Halide Perovskite Alloys

https://github.com/mannodiarun/perovs_dft_ml

Or directly open
https://colab.research.google.com/github/mannodiarun/perovs_dft_ml/blob/main/Perovs_DFT_ML_MRS_tutorial.ipynb

https://tinyurl.com/2htafw57

Reduced size: **https://tinyurl.com/4tnmxsys**