Statistical Learning for Halide Perovskite Discovery

Panayotis Manganaris[1]

[1]Purdue Materials Science and Engineering Mannodi Group

June 16, 2022

Outline

Artificial Intelligence I

The Four Approached to AI

Thinking Humanly	Thinking Rationally
- Turing test approach	- Laws of Thought
(The Six Fields of AI)	– logical positing
– NLP	– proven algorithms
– Knowledge Representation	– correct inference
– automated reasoning	– syllogistic reason
– Machine Learning	
– computer vision	
– robotics	
Acting Humanly	**Acting Rationally**
- cognitive modeling approach	- The rational agent
– neuromorphic algorithms	– inference + reflex
	– inference vs deduction

Russell and Norvig [2010]

Machine Learning I

ML Contributes to AI

- Adaptable agent
  - Contextual judgment of percept relevance
  - Autonomous utilization of percept sequence
- Learning
  - function performance improves with exposure to more percepts

Definition (Artifical Agency)

agent self-contained sensor->function->action pipeline

function Set of all possible responses for all possible percepts

percept sensory input

percept sequence history of sensory input

Machine Learning II

Supervised Training

Encourage the agent to behave "correctly"

1. Minimize Loss
2. Maximize Score

Unsupervised Training

The agent determines something principally true about its environment using mathematical/logical characterization methods.

- find eigenvectors and eigenvalues
- differentially calculate optima

Inverse Design

A Type of AI Implementation

senses	maps points in many dimensions
function	reliably navigates it's environment searching for optima
action	returns its findings to human interpreters

Perovskite Structure and Chemistry

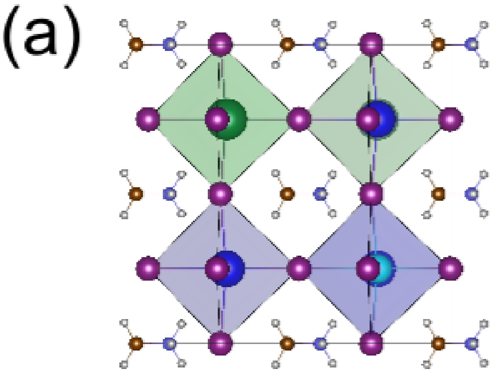


Figure: Example of hybrid organic-inorganic $MAPbI_3$ Mannodi-Kanakkithodi and Chan [2022]

Our Dataset

DFT Simulations

1. geometry optimization
2. Static band structure and optical absorption

Levels of Theory

- PBE
- HSE06
- PBE+HSE06(SOC)
- Experimental

Formula	bg_{eV}	η	LoT
MAPbCl3	3.0300	0.0020	EXP
CsPbI0.375Br2.625	1.6880	0.1532	PBE
RbSnBr2.625Cl0.375	1.4467	NaN	HSE
CsGeCl3	1.0510	0.1767	PBE
MASr0.5Pb0.5Cl3	5.3125	NaN	HSE
MABa0.25Pb0.75I3	1.9980	0.0155	PBE
MASnI3	2.5741	NaN	HSE
MACa0.5Pb0.5Cl3	5.3219	NaN	HSE
...	...	...	...

Band Gap Fidelity I

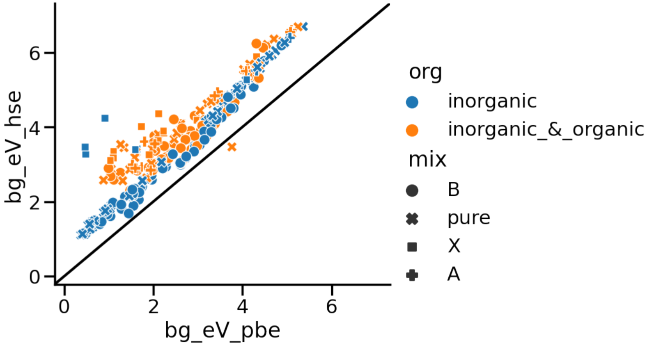


Figure: PBE vs HSE Band Gaps

Band Gap Fidelity II

Almora et al. [2020]

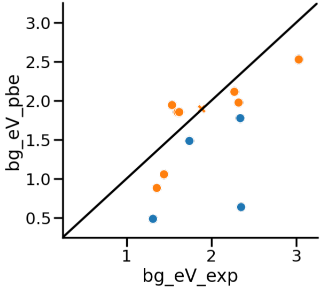


Figure: PBE vs Almora BG

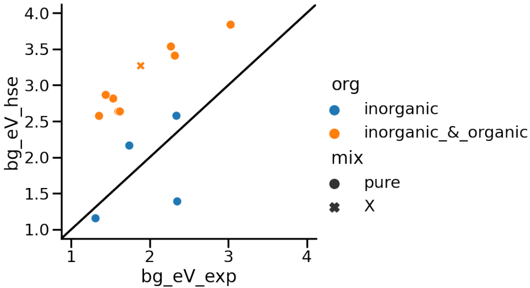


Figure: HSE vs Almora BG

Data Pre-Processing

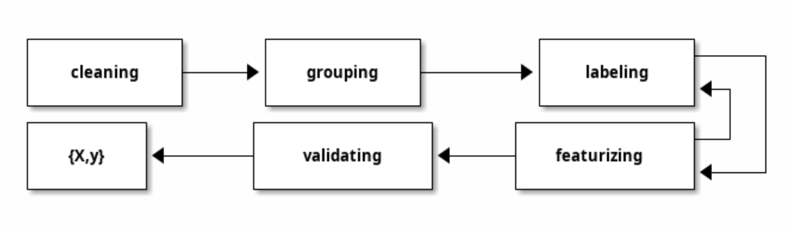


Figure: Data Preprocessing Workflow to Implement with Python Pandas

Machine Learning Pipeline

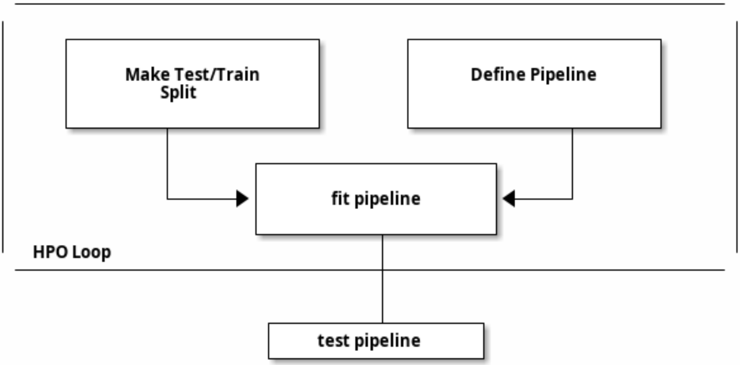


Figure: Machine Learning Pipelin to Implement with Python SciKit-Learn

Implementation in Jupyter Python I

```
import sys, os
sys.path.append(os.path.expanduser("~/src/cmcl"))
sys.path.append(os.path.expanduser("~/src/spyglass"))
import pandas as pd
import numpy as np
import cmcl
from spyglass.model_imaging import parityplot
from sklearn.pipeline import make_pipeline
from sklearn.<module> import NumPreProcessor1
from sklearn.<module> import CatPreProcessor1
from sklearn.<module> import NumPreProcessor2
from sklearn.<module> import CatPreProcessor2
from sklearn.<module> import Estimator

df = pd.read_<data>('./file.<data>')
df = df.groupby('Formula', as_index=False).agg(
    {'bg_eV':'median',
     'efficiency':'median'})
```

Implementation in Jupyter Python II

```
dc = df.ft.comp()
dc = dc.assign(label='label')

numeric_features = dc
.select_dtypes(np.number)
.columns
.to_list()
numeric_pipeline = make_pipeline(NumPreProcessor1(),
                                 NumPreProcessor2())
categorical_features = mc
.select_dtypes('object')
.columns
.to_list()
catagorical_pipeline = make_pipeline(CatPreProcessor1(),
                                     CatPreProcessor2())
```

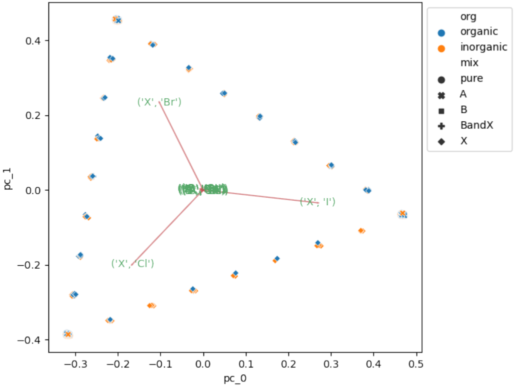
Implementation in Jupyter Python III

```
preprocessor = colt(
    transformers=[
        ("num", numeric_pipeline, numeric_features),
        ("cat", categorical_pipline, categorical_features),
    ]
)

ss = ShuffleSplit(n_splits=1, train_size=0.8,
                  random_state=None)
train_idx, test_idx = next(ss.split(dc))
dc_tr, dc_ts = dc.iloc[train_idx], dc.iloc[test_idx]
df_tr, df_ts = df.iloc[train_idx], df.iloc[test_idx]

pipe = make_pipeline(preprocessor, Estimator())

pipe.fit(dc_r, df_tr.<target>)
```

Implementation in Jupyter Python IV

```
p, data = parityplot(pipe,
                     dc_ts, df_ts.<target>.to_frame(),
                     aspect=1.0)
p.figure.show()
```

PCA



$$UAU^{\dagger} = Q^{-1}SQ$$

Figure: Learn transformation matrix U to diagonalizes the matrix A. The Principal Components in Q corresponding to the largest two Singular Values in S contain the majority of the variance in the data.

tSNE

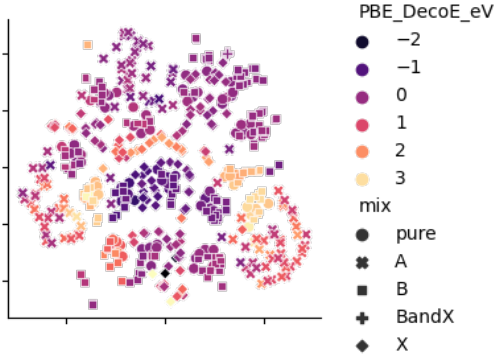


Figure: Learn a low-dimensional (2 or 3D) embedding space in which statistical similarity governs the proximity of high-dimensional data points

UMAP

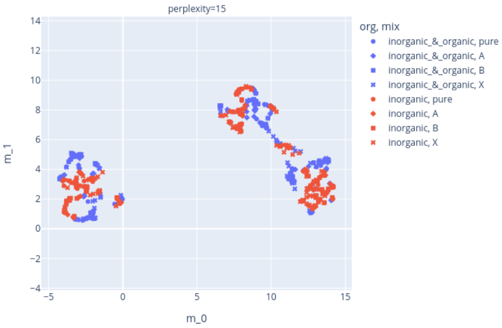


Figure: Learn a manifold embedding space in which nearest neighbors form clusters

Linear regression on BG I

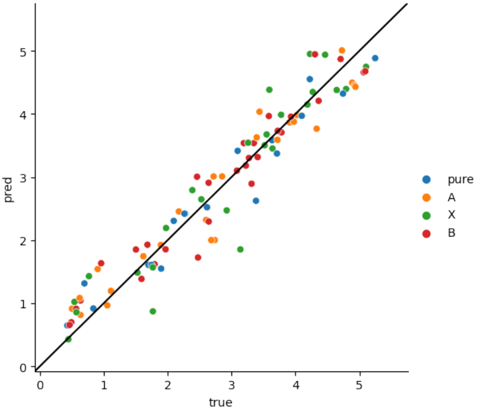


Figure: OLS determines $\vec{w}$ so that $f(x) = \vec{x}^T \vec{w}$, $y_i = f(x_i) + \epsilon_i$ and all ϵ_i are as small as possible

Linear regression on BG II

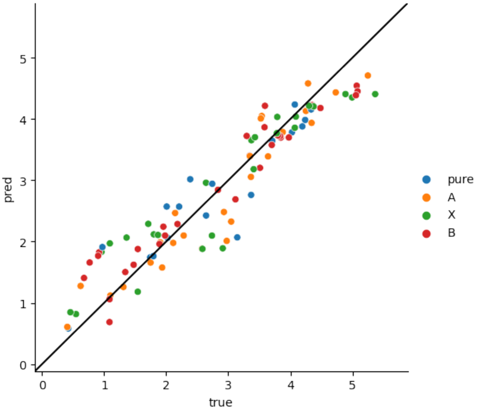


Figure: elasticnet determines $\vec{w}$ as before, but also works to sparsify the model

OLS weights

site	element	
A	Cs	23.771206
A	FA	25.794831
A	K	22.774475
A	MA	25.452629
A	Rb	23.282988
B	Ba	-32.603053
B	Ca	-31.378385
B	Ge	-45.001044
B	Pb	-42.526511
B	Sn	-46.868114
B	Sr	-32.068490
X	Br	0.939374
X	Cl	1.769032
X	I	0.140658

	RSS
A	54.213044
B	95.426246
X	2.007905

.

elasticnet weights

site	element	
A	Cs	-0.191057
A	FA	1.589015
A	K	-1.081903
A	MA	1.214167
A	Rb	-0.530437
B	Ba	5.139688
B	Ca	6.424156
B	Ge	-5.879154
B	Pb	-3.673012
B	Sn	-7.689152
B	Sr	5.678253
X	Br	0.000000
X	Cl	0.819669
X	I	-0.786942

	RSS
A	2.342552
B	14.391222
X	1.136281

Gaussian Process or BG I

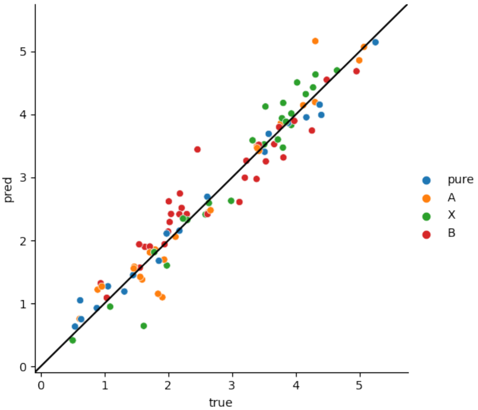


Figure: GPR picks functions from a distribution derived from the data covariance. The functions that satisfy the data form the fit.

Gaussian Process or BG II

Regularization with Priors

Conditional Probablity $P(x|y) = \frac{P(x)P(y|x)}{P(x)}$

Conditional Odds $O(x|y) = O(x)\frac{P(x|y)}{P(x|\neg y)}$

Isolated Bayesian Prior $B = \frac{P(x|y)}{P(x|\neg y)}$

Osbel Almora, Derya Baran, Guillermo C. Bazan, Christian Berger, Carlos I. Cabrera, Kylie R. Catchpole, Sule ErtenEla, Fei Guo, Jens Hauch, Anita W. Y. HoBaillie, T. Jesper Jacobsson, Rene A. J. Janssen, Thomas Kirchartz, Nikos Kopidakis, Yongfang Li, Maria A. Loi, Richard R. Lunt, Xavier Mathew, Michael D. McGehee, Jie Min, David B. Mitzi, Mohammad K. Nazeeruddin, Jenny Nelson, Ana F. Nogueira, Ulrich W. Paetzold, NamGyu Park, Barry P. Rand, Uwe Rau, Henry J. Snaith, Eva Unger, Lídice VaillantRoca, HinLap Yip, and Christoph J. Brabec. Device performance of emerging photovoltaic materials (version 1). *Advanced Energy Materials*, 11(11):2002774, 2020. doi: 10.1002/aenm.202002774. URL http://dx.doi.org/10.1002/aenm.202002774.

Arun Mannodi-Kanakkithodi and Maria K. Y. Chan. Data-driven design of novel halide perovskite alloys. *Energy Environ. Sci.*, 15:1930–1949, 2022. doi: 10.1039/D1EE02971A. URL http://dx.doi.org/10.1039/D1EE02971A.

Stuart Russell and Peter Norvig. *Artificial intelligence : a modern approach*. Prentice Hall, Upper Saddle River, New Jersey, 2010. ISBN 9780136042594.