Using Machine Learning in Materials Science

Marco Fronzi

1 Introduction

Machine learning in materials science enables predictions of material properties, the discovery of new materials, and insights into material behaviors. This tutorial demonstrates practical applications of ML models, from data preparation to model evaluation.

2 Data Collection and Preparation

2.1 Data Sources

Materials science data often comes from databases like the Materials Project, OQMD, and ICSD. These databases provide structured data on material properties, which are crucial for training predictive models.

2.2 Data Cleaning

Data must be clean and reliable. Common steps include:

- Removing samples with missing data.
- Normalizing data to a standard scale.
- Encoding categorical variables.

Mathematically, cleaning may involve transformations such as:

$$x_{\text{normalized}} = \frac{x - \mu}{\sigma}$$

where μ and σ are the mean and standard deviation of the data, respectively.

3 Feature Engineering

Feature engineering transforms raw data into a format that machine learning algorithms can work with effectively.

3.1 Feature Generation from Chemical Formulas

For instance, converting chemical formulas into numerical features:

```
# Using MatMiner for feature extraction
from matminer.featurizers.composition import ElementProperty
featurizer = ElementProperty.from_preset("magpie")
data['features'] = data['formula'].apply(lambda x: featurizer.featurize(x))
```

This process typically involves computing statistical properties like atomic weight, electronegativity, and radius, aggregated over the elements present in the formula.

4 Model Selection

4.1 Choosing the Right Model

Model selection depends on the problem type (regression or classification). For regression, models like Random Forests or Neural Networks are typical.

4.2 Model Theories

Random Forests, for example, are an ensemble learning method for regression (or classification) that operates by constructing a multitude of decision trees at training time. The output of the Random Forest is the mean prediction of the individual trees.

$$Y = \frac{1}{n} \sum_{i=1}^{n} f_i(X)$$

where n is the number of trees, and f_i is the prediction of the i-th tree.

5 Training the Model

```
# Training a Random Forest model
from sklearn.ensemble import RandomForestRegressor
model = RandomForestRegressor(n_estimators=100)
model.fit(train_features, train_labels)
```

6 Model Evaluation

Model evaluation assesses the accuracy and robustness of the model.

6.1 Metrics

For regression tasks, the Mean Squared Error (MSE) is a common metric:

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

where y_i and $\hat{y}_i$ are the true and predicted values, respectively.

```
# Calculate MSE
from sklearn.metrics import mean_squared_error
mse = mean_squared_error(test_labels, predictions)
print(f'Mean Squared Error: {mse}')
```

7 Conclusion

This tutorial presented a comprehensive approach to applying machine learning in materials science, emphasizing practical steps from data preparation to model evaluation.