Predicting Polymer Membrane Permeability Using Gradient Boosting Regression

Abdul Fayeed Abdul Kadir and Aamani Ponnekanti

CHENE 4670 – Chemical Engineering Data Analysis

Introduction

Polymer membranes are important for a variety of applications, from carbon capture to desalination.

Considering the wide variety of membranes available, as well as the myriad of applications, advances in the evaluation of polymer/gas pairings are in high demand. As a response, machine learning (ML) has emerged as a powerful tool for evaluation of membrane performance.

Previous work by Barnett et al. explored the utilization of Gaussian process regression to predict polymer membrane performance based on permeability to the specified gas.

The objective of this work is to develop a new ML model for the same set of data which can predict the membrane permeability to a greater degree of certainty.

Methodology

Dataset Generation

This study utilizes the database compiled by Barnett et al. of literature-reported polymer membranes and their performance versus helium, hydrogen, carbon dioxide, nitrogen, and methane. Only permeability was used to quantify performance.

Features that could affect permeability were compiled. The RDKit fingerprinting algorithm was used to generate a unique numerical fingerprint describing each molecule based on chemical connectivity between units. Molecular weight (MW) of the polymers, as well as the temperature at which permeability was reported were also included.

Instead of using a holdout set for model validation, cross-validation is used, where each subset of the data is used both as a training and testing sets. Cross-validation with 5 folds were used, where the data were split into 5 groups.

Machine Learning Methods

Gradient Boosting Regressor (GBR) was used. This ensemble method combines the results of multiple models to improve the prediction. The sklearn.ensemble package leveraged combines two models. GBR is also a boosting ensemble method, as opposed to stacking or bagging. Whereas stacking and bagging learn and combine heterogeneous or homogeneous models in parallel, boosting creates sequential models which learn from the errors of the previous models.

Bayesian Optimization was used to tune the hyperparameters. This method generates a surrogate model to decide which hyperparameters to sample, reducing the number of iterations necessary as compared to grid or random search methods. Furthermore, the built-in "feature_importances_" function was used to scale the effect of the three features on the model.

Performance was evaluated using mean absolute error (MAE), mean squared error (MSE), and testing R^2 values, calculated using built-in functions.

Results

Default Gradient Boosting Regression

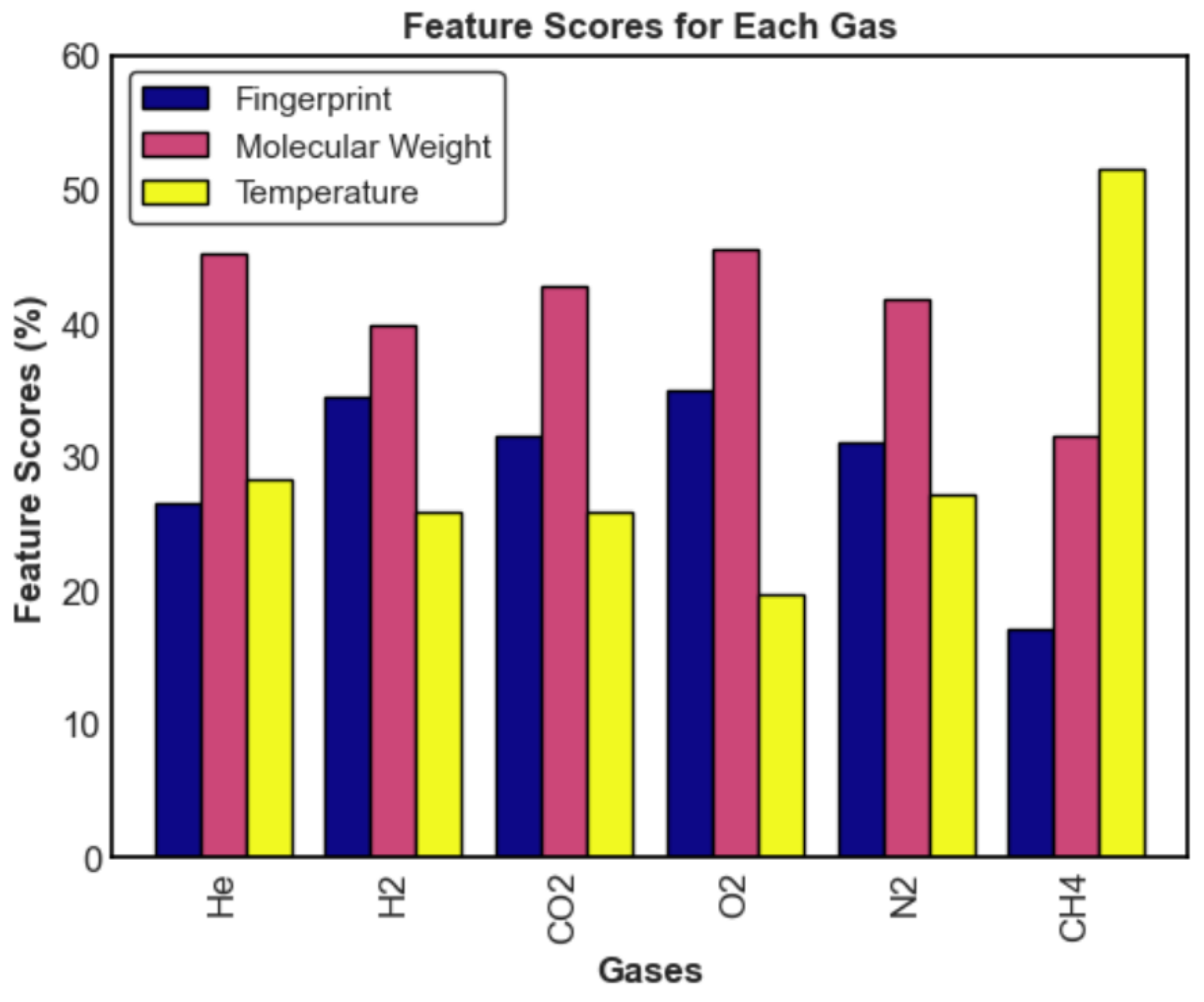
The following scores were generated using GBR with default hyperparameters (number of estimators: 100, learning rate: 0.1, maximum depth: 3), ***only using molecular fingerprint*** as a feature, and without performing cross validation (instead 80% of the dataset was randomly selected to train, and the remaining 20% to test).

Table 1: Standard Gradient Boosting Regression Errors

Gases	MAE	MSE	R^2
He	0.448	0.381	0.652
H_2	0.373	0.290	0.621
CO_2	0.601	0.858	0.570
O_2	0.465	0.447	0.707
N_2	0.644	0.913	0.634
CH_4	0.524	0.672	0.675
Average	0.509	0.594	0.643

Feature Engineering

To improve the predictive power of the model, ***molecular weight*** and ***temperature*** were added as features. To demonstrate their effect, their importance was calculated and used to scale the GBR. Typically, molecular weight has the strongest correlation with permeability, followed by fingerprint, and temperature. CH_4 and He are exceptions as fingerprint is the least important. Additionally, for CH_4, temperature dominates as the most important feature.



Hyperparameter Tuning

To improve the model performance, Bayesian optimization was implemented to generate new hyperparameters for each gas. This method generated the following values, which were then plugged into a new GBR.

Table 2: Optimized Hyperparameters

Gases	Number of Estimators	Learning Rate	Maximum Depth
He	97	0.114	5
H_2	180	0.058	4
CO_2	157	0.081	3
O_2	180	0.163	4
N_2	103	0.061	4
CH_4	158	0.084	3

Tuned GBR Results

The following scores were generated from the improved GBR.

Table 3: Improved Gradient Boosting Regression Errors

Gases	MAE	MSE	R^2
He	0.327 ± 0.062	0.232 ± 0.095	0.734 ± 0.095
H_2	0.363 ± 0.076	0.293 ± 0.120	0.736 ± 0.126
CO_2	0.489 ± 0.066	0.531 ± 0.169	0.674 ± 0.086
O_2	0.461 ± 0.085	0.487 ± 0.169	0.709 ± 0.083
N_2	0.515 ± 0.076	0.578 ± 0.190	0.715 ± 0.080
CH_4	0.518 ± 0.077	0.575 ± 0.148	0.746 ± 0.062
Average	0.446 ± 0.081	0.449 ± 0.150	0.719 ± 0.026

Discussion

The initial average training **R^2 value** for all gases using default settings for GBR is **0.6432**. This indicates that the model is not particularly strong. Tuning the hyperparameters and including new features with scaled importance modestly increased this value to **0.7190**, with increases of the R^2 of all the gases. In both cases, the R^2 value for CO_2. was the lowest. It is possible that the dataset did not have as much data for CO2. Alternatively, CO_2 could possess other features which affect its ability to permeate membranes.

Feature engineering also revealed interesting insights. For one, the data shows that molecular weight of the polymer is typically more important than chemical connectivity. This could be because molecular size has a greater impact on the ability of gases to permeate the material than the bonds. Temperature proved to be the least important in most cases, which is somewhat surprising as permeability often increases with temperature. It is possible that the temperature range represented in the dataset is too small to have much effect.

It is possible that in the case of He, molecular fingerprint loses some importance as the atom's small size is less affected by the membrane molecular structure. Further studies should be done to understand the deviations shown be He and CH_4.

Conclusions and Future Work

1. While GBR alone has weak predictive power, the inclusion of hyperparameter optimization and feature engineering can modestly improve performance.
2. There is significant room for improvement.
   - While sklearn.ensemble only combines two models, combining more could improve the prediction, although it would take more time
   - Other features could also be included in the dataset.
   - A more robust dataset could also lead to more accurate predictions.
3. Boosting algorithms are good for low-noise data sets and reducing bias, but stacking algorithms are particularly strong for generating more general models when features have more complex relationships with the target variable.
4. Given that there is not a direct correlation between polymer structure and permeability, stacking might perform better than boosting.

References

1. Kalirane, Mbali. "Ensemble Learning Methods: Bagging, Boosting and Stacking." *Analytics Vidhya*, 20 Jan. 2023, https://www.analyticsvidhya.com/blog/2023/01/ensemble-learning-methods-bagging-boosting-and-stacking/.
2. Stevens, Kevin A., et al. "Influence of temperature on gas transport properties of tetraaminodiphenylsulfone (TADPS) based polybenzimidazoles." *Journal of Membrane Science* 593 (2020): 117427.
3. Barnett, J. Wesley, et al. "Designing exceptional gas-separation polymer membranes using machine learning." *Science advances* 6.20 (2020): eaaz4301.
4. Bishop, Kyle. "22_Hyperparameters_and_Model_Validation." CHENE 4670. 12 April. 2023, Columbia University. Class lecture.
5. Wade, Corey, and Kevin Glynn. *Hands-On Gradient Boosting with XGBoost and scikit-learn: Perform accessible machine learning and extreme gradient boosting with Python*. Packt Publishing Ltd, 2020.