Predicting the Efficiency of Organic Photovoltaics

Angela Jiang, Tony Li, George Zhang

CS181 Practical 1 Writeup

https://github.com/gzhang01/cs181practicals/tree/master/practical1

Abstract

Our task was to predict the HOMO-LUMO energy gap in various molecules given a SMILES string representation of the molecule. We used random forest regression with new features we extracted using RDKit, including the types of atoms in the molecule, the types of bonds in the molecule, and the number of valence electrons in the molecule. Our model ultimately produced a RMSE value of 0.18849, which beat the benchmarks set by linear regression with the given features (0.29846) and random forest regression with given features (0.27207).

Technical Approach

We began by experimenting with the given features. However, we soon noticed that there was no information about what the features actually were and that all the features were denoted as either present or absent. We decided that we wanted features that we could identify, so we would know how they might predict the HOMO-LUMO gap, and that we wanted more quantitative features. Thus we shifted to focus on extracting additional features using RDKit.

We first extracted chemical features such as number of electron donors / acceptors within the molecule. The process we used ended up extracting 7 different features. However, we found that using these features in a linear regression produced no increase in performance when we increased the data (see table 2). As a result, we concluded that these features were not adequate in predicting the gaps, and so we tried another approach.

After a bit of research into what determines the HOMO and LUMO orbitals, we decided that number of atoms, number of specific types of bonds, and number of valence electrons would be useful in predicting the HOMO-LUMO gap. Since the HOMO-LUMO gap is exactly determined by the difference in energy of the orbitals, and the energy level of the orbitals depend on what type of bonds exist in the molecule, we thought these features would be useful in our prediction. Using RDKit, we extracted information on the number of carbon, oxygen, sulfur, nitrogen, silicon, and selenium atoms; the number of single bonds, double bonds, and aromatic bonds; and the number of valence electrons (see ExtractFeature.ipynb). The result of linear regression based on these features is in table 3.

Happy with our feature extraction, we moved onto choosing an appropriate model. The two models we experimented with were linear regression and random forest regression. After running several trials with each method, we determined that the random forest regression provided us with the best estimates based on our calculated RMSE values. Thus we used a random forest regression with our new features to make our final predictions.

Results

One of the first things we did was build a test suite that implemented k-fold validation (see testsuite.ipynb). The purpose of this file was to test our model and compare RMSE values against the baseline models without having to upload to Kaggle. Running the model with k folds produced k RMSE values based on the different train / test data, and the mean, variance, and standard deviation of those values were printed and used for analysis. We also wrote a script to cut the amount of data we were working with, so our initial attempts could be run on smaller sets of data (see shrinkTrain.py). Using these two files, we established baseline RMSE values for linear regression with given features for k = 5, n = 1000 and k = 5, n = 10000:

	k = 5, n = 1000	k = 5, n = 10000
mean	0.3546	0.3044
var	0.001203	9.265e-05
std dev	0.0347	0.009625

Table 1: Benchmark RMSE Values from Linear Regression

The results of our first attempt at modeling with new features is in table 2. We used 7 new features we extracted using RDKit's Chemical Features module. These included properties such as electron donors / acceptors, atom donors / acceptors, and polarity.

	k = 5, n = 1000	k = 5, n = 10000
mean	0.3204	0.3209
var	3.9448e-05	3.7402e-06
std dev	0.006281	0.001934

Table 2: RMSE Values from Linear Regression Using Chemical Features

The results of modeling with atoms, bonds, and valence electrons as features are below. As stated in the technical approach section, our features were the number of carbon, oxygen, sulfur, nitrogen, silicon, and selenium atoms; the number of single bonds, double bonds, and aromatic bonds; and the number of valence electrons.

	k = 5, n = 10000
mean	0.2617
var	2.5700e-06
std dev	0.009625

Table 3: RMSE Values from LR Using Atoms, Bonds, and Valence Electrons

The low RMSE values we obtained from our new features convinced us that we had a better set of features than the given ones. We then attempted to determine whether to use a linear regression or a random forest regression. We first ran trials with k = 5, n = 10000 to get a basic sense of how good the models were. After we had this data, we ran several more trials, this time using all the training data with various values of k. The results are shown in the tables below.

	model (features)	mean	variance	std dev

k = 5
n = 10000
	LR (old)	0.3044	9.265e-05	0.001603

	LR (new)	0.2617	2.5700e-06	0.009625
	RF (old)	0.2891	1.3563e-05	0.003683
	RF (new)	0.2373	3.0366e-06	0.001743

k = 2
n = all
	LR (old)	0.2991	9.9973e-09	9.9986e-05

	LR (new)	-	-	-
	RF (old)	-	-	-
	RF (new)	0.1906	5.2525e-08	0.0002292

k = 3
n = all
	LR (old)	0.2989	2.6188e-07	0.0005117

	LR (new)	0.2610	1.8260e-08	0.0001351
	RF (old)	0.2726	2.5348e-07	0.0005035
	RF (new)	0.1898	1.7538e-07	0.0004188

k = 4
n = all
	LR (old)	0.2989	2.6780e-07	0.0005175

	LR (new)	-	-	-
	RF (old)	0.2726	3.6953e-07	0.0006079
	RF (new)	0.1894	1.5194e-07	0.0003898

Table 4: Comparison of RMSE values using various methods (k = 5, n = 10000)

With this data, we concluded that our new features, combined with a random forest regression produced the best results. Therefore, we decided to submit our predictions based on this model. The final RMSE values calculated by Kaggle are shown in table 5.

	our model	LR baseline	RF baseline
RMSE	0.18849	0.29846	0.27207

Table 5: Kaggle-produced RMSE values

Discussion

Our initial attempts to model the data centered on the features we were given. However, we quickly realized that they were all qualitative; that is, either the feature was present or it wasn't. Without knowing what the features themselves were, we found it difficult to manipulate them in any meaningful way to produce better predictions. As a result we aimed to extract new features for our model.

Our first attempt in this regard used the Chemical Features module. We extracted 7 features using that module, and built a linear regression model based on those features. When we tested our new model with 1000 data points, we found that it did mildly better than the linear regression model using the given features with 1000 data points (note that the baseline model did worse with 1000 data points because there was less data to use to train). When we attempted with 10000 data points, however, the baseline model improved, while ours did not (see table 2). Since an increase in data did not result in an increase in performance, we concluded that the model and features we had likely did not correlate to gap measurements very well. We confirmed this by plotting these features against the measured gap (plots not shown). This apparently lack of correlation led us to abandon these features and attempt to find new ones.

Our goal then was to find a set of features that would produce improved results. Since the energy levels of the HOMO and LUMO orbitals depend on the orbitals present in the molecule, we decided to consider the types of atoms and types of bonds present in the molecule. In addition, since the HOMO and LUMO are decided based on how many electrons have filled up the orbitals, we decided to keep track of the number of valence electrons. Using RDKit to extract these new features, we produced a new set of features which we used with the linear regressor and ran our test suite on it. We saw that our mean RMSE improved significantly (see table 3). From this, we concluded that our new set of features were more useful than the previous two sets we worked with, and so we were satisfied enough to move onto model selection.

Our attention then turned to determining which model to use. We ran linear regression and random forest regression on both the given features and our newest features. We tested each of these models for various number of folds and various number of data points and produced the data in table 4. In each instance, random forest regression with our new features produced the best mean RMSE results. Thus, we were relatively satisfied with this model. We proceeded to extract these new features from the test data and ran our predictor (see makepredict.ipynb) on the data. We then uploaded our final predictions onto Kaggle. Table 5 shows the final RMSE value calculated for our model using 49% of the training data, along with the baseline RMSE values on the leaderboard.

Conclusion

We were tasked with producing a model that could predict HOMO-LUMO energy gaps for a given molecule. Our approach centered on feature engineering, as we believed that having the proper inputs into our model would be more beneficial than attempting to fiddle with the given features using various models. Unfortunately, the learning curve associated with RDKit prevented us from moving onto more model exploration, which is something we hope to expand upon next time, but ultimately, we were able to produce a model that outperformed the linear regression and random forest regression models that used the given data.