Practical 1: Predicting the Efficiency of Organic Photovoltaics

Yi Ding, Linying Zhang, Danny Zhuang
Kaggle: LYD

February 10, 2017

1 Technical Approach

Our approach consisted of two main steps: feature engineering and model selection.

- **Feature Engineering**

  First, we eliminated 225 non-informative features from the given 256 binary features because they either consisted of all 0s or all 1s, which left us with 31 usable binary features and 1 SMILES string.

  Next, we decided to extract additional features from the given smile string. Given all features provided in the dataset were binary features, only indicating the existence/non-existence of certain chemical structure, we believed that adding quantitative features about molecular substructure would potentially improve our prediction. Based on literature review, we found that the size of HOMO-LUMO gap is closely related to the delocalization ability of electrons in the chemical structure, which is associated with chemical features such as the number of aromatic rings, unsaturated bonds, and rotatable bonds.[1][2] We used the rdMolDescriptors module in RDKit to extract 16 such features from the SMILES string.

  We also extracted an additional 26 quantitative features by counting the number of certain atom and atom-atom combinations in the SMILES strings, such as semiconduct elements Se and Si, and number of carbons c. At the end, we had $31 + 27 = 68$ total features.

  As a side note, in the process of fitting linear regression models, we also considered two-way interaction of the 31 binary variables since coexistence of two different features may enhance or attenuate their individual influence on HOMO-LUMO gap.

- **Model Selection**

  We were dealing with a regression problem. so we started with some linear regression models, from standard linear regression to regressions with regularization (e.g. Ridge and Lasso) and regression with interaction terms. We also tried a non-linear model, in this case, random forest for reasons explained in the following paragraphs.

  The idea behind using regularization was to avoid the problem of overfitting. Based on the loss function of Ridge and Lasso, the tuning parameter lambda serves to control the relative impact of the least squares and the coefficient term in minimizing the loss function. As

Model	Before feature engineering	After feature engineering
Ridge	0.460	0.671
Ridge with interaction	0.525	0.666
Lasso	0.457	0.657
Lasso with interaction	0.524	0.667
Random Forest	0.545	0.933

Table 1: R^2 of models before and after feature engineering.

lambda increases, the impact of the shrinkage penalty grows, and the regression coefficient estimates will approach zero. An extra benefit from Lasso was that it could be also used for variable selection as l1 penalty has the effect of forcing some of the coefficient estimates to be exactly zero.

Ridge Regression (squred ℓ_2 norm) $\quad \min_{\mathbf{w}} \ \mathcal{L}(\mathbf{w}) + \lambda||\mathbf{w}||_2^2$

LASSO(ℓ_1 norm) $\quad \min_{\mathbf{w}} \ \mathcal{L}(\mathbf{w}) + \lambda||\mathbf{w}||_1$

Random forest was tuned for this project primarily because it contrasts the other methods by modeling the data in a non-linear form, which is likely the true scenario given there are more than sixty features and complicated effect modification between features. Another advantage of random forest is that it has low bias and low variance. The recursive partitioning of the feature space (characteristic of decision trees) means that random forests have low bias as they have the flexibility to fit any relationship given that the depth parameter is high enough. The variance of random forest is low compared to other tree methods like bagging since the trees are much less correlated as a consequence of randomizing features at each split[3].

For the convenience of computation, we use R^2 instead of RMSE criteria on kaggle to evaluate our models. These two criteria are consistent by $R^2 = 1 - \frac{N(RMSE)^2}{SSTO}$

2 Results and Discussions

All five models Ridge, Ridge with interaction terms, Lasso, Lasso with interaction terms and Random Forest were performed before and after feature engineering. The performance of these models were summarized in Table 1.
Overall, random forest had the best R^2 compared to the linear regression methods under the same condition. With new features added, all models had a dramatic increase of R^2: the linear models had an increase about 20% from 0.45 to 0.65, while R^2 of random forest jumped from 0.55 to 0.93. The increase of R^2 indicates that the features we extracted dramatically improved the prediction.

- **Linear Regressions**
  Before feature engineering, our baseline linear regression achieved R^2 of 0.45 on the validation set. We then moved on to tuning the penalty parameter lambda from 10^{-7} to 10^7 for both Ridge and Lasso through 5-fold cross-validation. The best R^2 achieved from Ridge and Lasso was about the same as that from linear regression. We also noticed that as lambda increased,

all coefficients quickly went down to zero and the R^2 also went down to zero (Figure 1), suggesting that linear regression without shrinkage was actually optimal in this case. (All 31 features are useful for accurate prediction of HOMO-LUMO gaps.) After including interaction terms, the R^2 increased to above 0.52, which indicated that non-linear effects are necessary in predicting HOMO-LUMO gap.

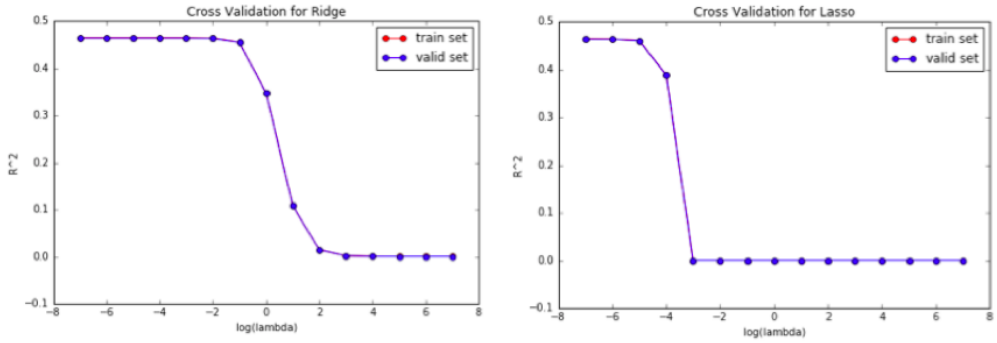

Figure 1: Ridge and Lasso R^2 from parameter tuning.

- **Random Forest**

The Random Forest Regression can be described as follows:

1) Bootstrap: Draw n datapoints with replacement from your training data
2) Feature randomization: Select m features of your total p features for the current node
3) Optimization: Choose the feature that can be split to minimize the sum of squared loss function
4) Split: Split the node into two daughter nodes
5) Continue steps 1-4 until a minimum node size or depth is reached
6) Continue steps 1-5 until the desired number of trees have been fitted
7) Prediction: Average the results of each of the decision trees

We used a 5 fold cross validation to tune our random forest regression for three parameters: `n_estimators` the number of trees, `max_depth` the maximum depth of each tree, and `max_features` the number of features to be considered at each node. Because of computation efficiency, we chose to train our model on 100,000 randomly sampled datapoints with replacement.

Because the number of trees in the random forest affect the variance (For number of tree B: if identical but not independent and pair correlation ρ is present, then the variance is: $\rho = \sigma^2 + \frac{1-\rho}{B}\sigma^2$ As B increases the second term disappears) and not the bias of the model, we only tuned over two sufficiently large values to confirm they produced similar testing errors. We tuned `max_depth` up to 80 to be certain we were seeing a full range of model flexibilities (as we increase depth, bias goes lower). Lastly, we tuned `max_features` from 5 to 65 in order to test all values of features (total 68).

We ultimately found that our random forest peaked at a CV-testing R^2 score of 0.87. We applied the principle of Occams Razor to choose the parameters of our final model: the smallest parameter values for the given testing R^2 of 0.87. These parameters were `n_estimators` = 40, `max_depth` = 20, `max_features` = 45.

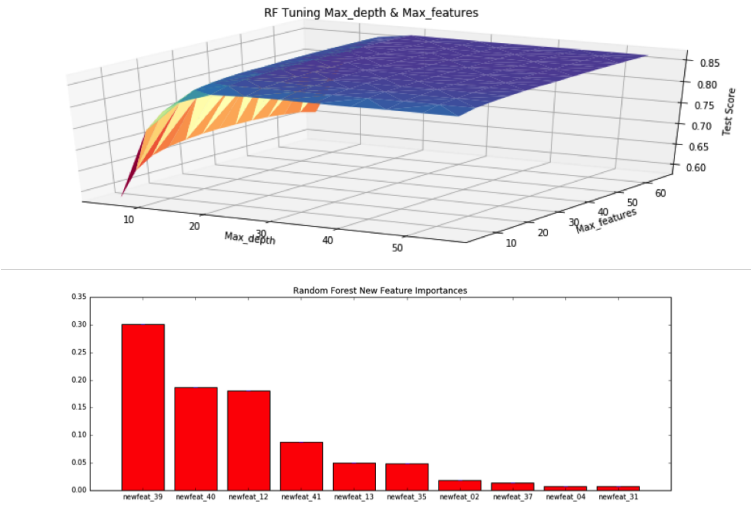

Figure 2: Random Forest tuning and top 10 important features with optimal parameters.

The 10 most important features are shown in Figure 2. It uses the gini importance which is the total decrease in node impurity weighted by the proportion of samples reaching that node averaged over all trees of the ensemble. At a high level, the feature importance values are the amount by which the feature on average decreases the gini coefficient so higher values indicate higher importance. Some of the top 10 features from random forest were seen in top 10 features from Lasso (data not shown).

In retrospect of our model building progress, we devoted large time on trying different models and tuning various parameters on original data, yet the accuracy plateaued around 0.5 for validation sets. However, after consulting experts in chemistry and including new features based on subject matter knowledge, the accuracy jumped to 0.9. It reveals to us that interdisciplinary communication is as important as mathematical reasoning and coding for data scientists.

3 Reference

[1] Brdas J., Norton J., Cornil J., and Coropceanu V. **Molecular Understanding of Organic Solar Cells: The Challenges**. *Accounts of Chemical Research* 2009 42 (11), 1691-1699
[2] Bakulin A., Rao A., et al. **The Role of Driving Energy and Delocalized States for Charge Separation in Organic Semiconductors**. *Science* 16 Mar 2012 : 1340-1344
[3]Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani. 2014. **An Introduction to Statistical Learning: With Applications in R**. *Springer Publishing Company, Incorporated.*