Prediction of perceived pleasantness of an odour based on some molecular features

Benoit DELECOURT, Julia HEINIGER
Introduction to Machine Learning for Bioengineers

I. Introduction

The pleasantness of on odour is defined as its quality of being enjoyable, attractive, friendly, or easy to like. An odour is based on the smell of molecules, each of them composed of a series of physical and chemical features. This project aims to predict the pleasantness of on odor based on its molecule features. A database containing smelled molecules, their physical and chemical features and corresponding pleasantness "scores" is given. By training both linear and non-linear machine learning models on the data, we'll try to make the most accurate predictions of pleasantness for unseen molecules.

II. Exploration of raw data

A. Description of variables

In a first step, the raw data was explored. The training data contains 708 molecules for which the subject has smelled and given a pleasantness score. The output variable is named *VALENCE.PLEASANTNESS*. The score can go from 0 to 100, although in this data-set, a minimum and maximum value of 0 and 98 respectively were found. The response variable is based on 4070 predictors, which includes the intensity of the odor and physicochemical features. Variables have different types: doubles, and integers. Note that the feature *SWEETORSOUR* is a boolean variable which is not present in the test set, so it was decided to exclude it from the training set. Concerning the response variable, it was found to be more or less normally distributed (Gaussian shape). This project can be without a doubt classified as a high dimensional regression task, as the number of predictors exceeds by far the number of observations.

B. Feature Reduction

To avoid the high dimensional risk of over-fitting the data, the dimension needs to be reduced by some data cleaning. After having removed all the missing values ("NA" in R), all input variables with zero variance were excluded, as they do not have meaningful predictive power and can result in unstable models. The dimension could be reduced from 4870 to 3028 variables (38% of data).

A key aspect of data visualisation is the Principal Component Analysis (PCA). The PCA did not result in definable patterns or clusters with the first two principal components. A large part of the proportion of variance is explained by about ten predictors. A t-SNE analysis was also performed to capture potential neighbourhood-relationships yet did not let us a better understanding of the data.

C. First Regression

With the 3028 predictors remaining of this data cleaning, a first multiple linear regression was performed. As expected, a highly overfit of the training data (training Root Mean Squared Error (RMSE): ~ 11.70863) and a very large validation RMSE of about 10^7 (see Table I) were obtained. These results are combined with really high variance, meaning that this method models the noise in the training set instead of the true function. Moreover, the application of linear models to correlated data may not be appropriate, with nonlinear functional relationships potentially going undetected and creating instabilities in the predictive model. Therefore, a correlation analysis consisting in removing the most correlated variables was performed and reduced the dimension. In fact, the idea is to prevent the risk of multicollinearity between variables, which undermines the statistical significance of an independent variable. No perfect correlation between predictors was found, but 1831 predictors were correlated with an absolute index value above 0.95. After removing them, 1195 predictor variables were remaining in total. We could reduce the validation RMSE by a factor 1000 and the variance by a factor 10^6, yet being still very large (see Table I). During this exploration of the data, it was confirmed that high dimensionality problems make it really favorable to over-fitting. Some more effort needs to be done in the reduction of both dimension and variance. The easiest and most common way to do it is by applying techniques that limit its effective capacity, namely by regularization. Hence, in the next section, some regularizing linear methods will be used to penalize the model's high flexibility and get better predictive performances.

III. Linear Methods

A. Regularization

For each of these methods, we compared the performance between the full data set (3028 predictors) and the data set after having removed correlated variables (1195 predictors). First, a Lasso Regularization with a 5-Fold Cross-Validation was performed. Thereby both the validation RMSE and variance decreased significantly (see Table I). The best results were obtained for a subset of predictors of size around 30 (compared to more than three thousand), showing that reduction of dimension was paramount. Another method, called the Ridge

Regularization, was used as a comparison. Instead of shrinking the number of predictors, this method keeps all predictors but penalizes the irrelevant ones. It resulted both in a higher mean validation RMSE and variance than the Lasso Regularization. Removing correlated variables (correlation index of 0.95 or above) did not end in a significant decrease in the mean of validation RMSE for the Lasso Regularization, in contrary to the Ridge one.

B. Feature selection

Instead of reducing the predictor's coefficients towards zero, another approach which aims to select variables with the largest correlation with the dependent variable was performed: forward selection. It was run with 5-fold cross-validation to find the top-performing model. A validation RMSE of 21.264 corresponding to a subset of nine predictors was found, which is a significant reduction of dimension. This result is better than the regularization methods, therefore we will perform some feature engineering on these predictors in the next section, which aims to detect a non-linear relationship between the features and the response.

IV. Non-Linear Methods

A. Feature Engineering

The subset of nine predictors obtained by the forward selection was used for polynomial feature engineering in the aim of improving the linear model. Indeed, transforms like raising input variables to a power can help to better expose the important relationships between input variables and the target variable. Therefore, a polynomial regression was performed by raising all the predictors to a power 1,2 or 3. By 5-fold Cross-Validation, each possible combination was evaluated and we selected the best one.

B. Random Forest, Bagging & Boosting

At the expense of some loss in interpretation, supervised learning decision tree methods (Random Forest, Bagging and Boosting) often result in improvements in prediction accuracy. First, the Bagging method was performed. To select the optimal parameters, we set them firstly to conventional values: the number of predictors for each split (*mtry*) is fixed to the total number of predictors (with correlation index below 0.9) and the number of trees (*ntree*) is set to 100 so that we keep the computational costs within a limit while reducing the variance enough. The number of leaf nodes (*maxnodes*) is set to 30, to catch the complexity but not to overfit the training data. Then, we cross-validated *ntree* and *maxnodes* parameters one after the other for different values around their start value and fixed them to the best one. As previously said, this problem is high dimensional, and including all the variables in the model may cause correlation problems. To alleviate it and improve variable selection, it was decided to perform the Random Forest method. Instead of using all the predictors, we set initially the *mtry* parameter to $1000 \sim \frac{predictors}{3}$. Again, we cross-validated the three parameters (*mtry*, then *ntree* and finally *maxnodes*), in the same way as for Bagging. Finally, we thought that having a sequential approach, i.e growing trees using information from previously grown trees could improve the previous models: that is why we performed Boosting. However, even after carefully tuned the number of trees (*nrounds*), the number of splits (*max.depth*) in each tree and the shrinkage parameter (*eta*), it did not result in better performance. (Table I).

C. Artificial Neural Network

In order to capture non-linear effects including high-dimensional interactions, we decided to construct a Neuronal network. We used three layers (one input, one hidden and one output layer) so that we do not over-fit the data. Afterwards, we cross-validated sequentially the parameters of the network (number of nodes, drop-out rates) and the learning function as we did for the random forest. Additionally, we also evaluated the seed. Indeed, it would be possible that with a specific seed a local minimum is attained instead of the global minimum. We chose the common "Relu" as activation function for the 2 first layers and and a "Linear" for the output layer.

V. Results

To compare the different selected models, we computed for each one 50 validation root mean square errors (RMSE) by splitting the training data 50 times in a random training set and validation set. We calculated the mean of the RMSEs and the variance out of it.

TABLE I
*Predictors with a correlation index under 0.95. **Predictors with a correlation index under 0.9

Comparison of different models		
Model	$\overline{RMSE}$	Variance
Linear methods		
Full Multiple Linear Regression	3.94×10^{7}	7.47×10^{16}
Multiple Linear Regression less predictors*	7.66×10^{4}	9.21×10^{10}
Full Lasso Regularization	21.954	0.717
Lasso with less predictors*	21.944	0.711
Full Ridge Regularization	22.576	6.064
Ridge with less predictors*	21.854	0.759
Forward selection	21.264	0.648
Non Linear methods		
Feature Engineering on Forward Selection	**21.183**	**0.614**
Random Forest	21.725	0.734
Bagging**	21.676	0.711
Neural Network	21.997	0.882
Boosting	23.666	0.878

By looking at table I, it is the linear regression with the nine predictors (Forward selection) that resulted, on average, in the smallest RMSE and variance for the Linear methods. This result was only improved by feature-engineering those predictors. All other non-linear methods have a slightly higher RMSE and a bigger variance. Nevertheless, Random Forest, Bagging and Neuronal Network are all better than the other multiple linear regression methods. However, we can see that regularized linear models performed nearly as well as random forest–based ones. At the end, it was shown that predicting the pleasantness of on odour does not imply thousands of molecular features, but only relevant ones that share both linear and non linear relationships with the response.