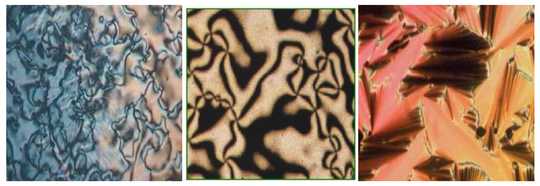


Statistics in Liquid Crystal Properties Prediction

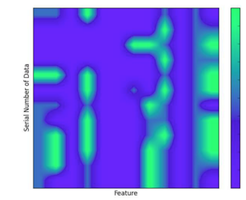
Xintian Han[1], Molei Liu[2] and Ruohan Zhan[1]
[1] Yuanpei College, Peking University, [2] School of Mathematical Sciences, Peking University

Introduction

A data-driven idea in liquid crystal production has drawn increasing attention in manufacturing. A precise beforehand judgement contributes much to the production efficiency. Numerical simulations select promising candidates for target products with desirable properties, thus largely shortening the procedure for trial and error, and reducing the production cycle.

To be specific, in this report, we need to predict eight regression properties: $TN-I, ne, no, \epsilon_{\parallel}, \epsilon_{\perp}, \eta, V90, V10$ and one classification property: $Ts-N$, given the liquid crystal recipe, that is, the proportion of monomers. For regression, our goal is to minimize the prediction error within *five times of production precision*. All shown results are tested on testing data. In the next, we will try statistics and machine learning methods we have learned so far, including linear methods, nonlinear methods, kernel methods, tree-based methods and graphic models, then followed a quantitative comparison for these models' performances and analysis of numerical results.

Data Preconditioning



We totally have 1347 raw data samples covering 295 monomers. For each data, the input is a vector denoting proportion of monomers in the recipe and the output is the corresponding eight regression properties and one classification property. It is worth noticing that the input vector is a very sparse one with averagely 13 nonzero entries, as it shows in the left graph.
Meanwhile, the usage frequency varies fiercely among monomers, leading to the consideration of input data dimension deduction based on the principle maximizing the ratio of sample number to monomer number. Besides, we delete the duplicate and incomplete data. As a result, 635 samples covering 119 monomers are chosen for regression properties and 404 samples for classification.

Classification

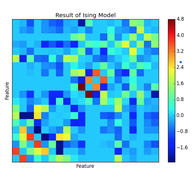
For classification, we test data on **logistic regression**(LR), **K-nearest neighbor**(KNN), **deep boltzmann neural network**(DBM), **support vector machine**(SVM, kernel=sigmoid), **classification tree**(CT), **random forest**(RF), **bagging**(BG), **boosting**(BST).

Algorithm	LR	KNN	DBM	SVM
Accuracy	89.6	90.9	88.1	88.1
Algorithm	CT	RF	BG	BST
Accuracy	88.1	94.0	89.6	90.9

It can be seen from the table that space distribution based methods(KNN, tree-based algorithms) perform relatively better than regression-based methods(LR, DBM, SVM), showing that this property is close to be piecewise constant in the space and the class patterns are so complicated that regression approximation does not work satisfactorily.

Further Work

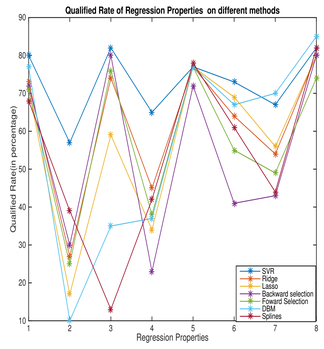
To our prior knowledge, there exist empirical combinations of monomers in the formula. In order to explore this data structure, we use Ising model to find out the probability of each two monomers to appear at the same time. The red patterns on the right graph show high probability to come together, while the blue ones mean the converse. We combine the monomers in red ones and use SVR to fit again, and it turns out that the qualified rate of Ne arises from 57 to 59, implying a potential data preconditioning method.

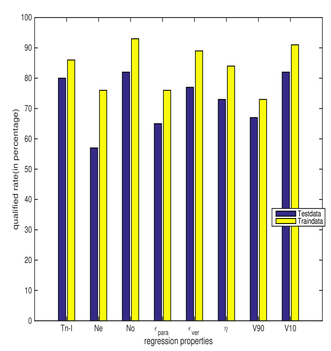


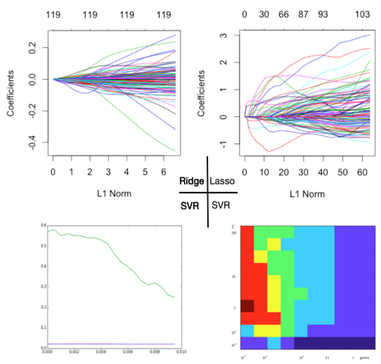
Besides, we notice that if we use $V10$ as a predictor, the qualified rate for others will improve notably, which suggests another promising exploration direction.

Regression-Linear, Nonlinear, Kernel Methods

We start from linear methods and use their result as a benchmark for other more complicated methods. Since the input vector has a dimension of 119 with only 13 or so nonzero elements, we test models on **subset selection**, **lasso** and **ridge** regression for feature selection and regularization. Then, kernel methods and nonlinear methods: **splines**, **support vector regression**(SVR) and **neural network** have been explored to approximate more complex functions. The performances are as follows:







The first graph shows that SVR performs best among all methods. It's interesting to notice that DBM performs well on some properties but performs worst on the others(Ne for example), while SVR performs almost best among all. In fact, the best choice of kernel transform in SVR varies from property to property, while in DBM all properties share the same approximation method, indicating that the functions are far from linear and hold different constructure. However, the second graph shows that SVR has the risk of overfitting, since the prediction on training data is notably better than on testing data. Parameter preferences and sensitivities can be found in third graph.

Regression-Tree based Methods

	$Tn-I$	Ne	No	$\epsilon_{\parallel}$	$\epsilon_{\perp}$	η	$V90$	$V10$
RT	64	12	78	27	70	54	48	78
BG	68	50	81	59	80	67	62	79
RF	72	44	82	58	78	67	62	79
BST	76	37	81	36	83	68	68	86
SVR	80	57	82	65	77	73	67	82

Motivated by the good performances of piecewise-constant based methods in classification, we explore this idea and use tree-based methods: **regression tree**(RT), **random forest**, **bagging**, **tree-based boosting** to implement regression prediction. Furthermore, we use Subset Selection to select the principle features for regression tree. Performances of tree based-methods are shown compared with SVR.
In the above table, we use colored figures to mark the best prediction and it's clear to see SVR has strong capability in prediction. Meanwhile, tree-based methods are good choices.

Contributions

Xintian Han: Model Fitting and Analysis
Molei Liu: Model Fitting and Analysis
Ruohan Zhan: Poster Making and Analysis

References

[1] The Elements of Statistical Learning, 2nd Ed. Hastie, Tibshirani, and Friedman.
[2] An Introduction to Statistical Learning, with applications in R. James, Witten, Hastie, and Tibshirani