Some advice on choosing a (set of) prediction methods

Marjolein Fokkema & Julian Karch

When it comes to applied data analyses, it is wise to carefully consider what method to apply to your data. There are no ground truths on what is the best method, but careful thinking about this before you start your analyses tends to save a lot of time or embarrassment later on.

This means carefully considering the costs and benefits of computation, information, predictive accuracy and interpretability. This can be an even more daunting task than fitting or interpreting a model!

The first distinction that must be made is whether you need to use supervised or unsupervised learning. Here, we assume that a decision has already been made in favor of supervised learning.

Two main aspects should drive your choice:

1. Context of Application

This centers around the question: What should the (ideal) end result of your analyses look like? More specifically:

Should it be human interpretable?

Single decision trees, GAM or lasso regression are well known for their relative ease of interpretation. But whether they are actually interpretable also depends on who will be the user of the final model and how it will be presented to them.

With the help of variable importances and partial dependence plots (or Shapley values, but these were not discussed in the course) black-box methods like random forests, gradient boosted ensembles or support vector machines can be made interpretable. But you have to be prepared to be able to explain how the values on the x- or y-axes have been computed!

Should the model allow for inference?

Support vector machines, random forests, gradient boosted ensembles, lasso regression, neural networks: They are notoriously good for prediction and notoriously bad for inference. They can *capture* non-linearities and interactions, but they do not explicitly model them, so you'll have no way of deciding whether the effects of the predictor variables could have likely occurred due to chance or not.

For inference, a good old GLM estimated by maximum likelihood or a Bayesian approach might fit your needs if its assumptions are approximately met. Word on the street used to that we can buy power by assumption. Nowadays, word on the street is that we can reduce variance by introducing a little bias. Note how these two statements essentially say the same thing!

Of course, when the assumptions are substantially incorrect (i.e., the bias incurred is large), this does not buy much power (i.e., will not reduce variance much) and might lead to misleading results. Thus, if you think the linearity assumption is overly strict or unrealistic, consider a GAM with smoothing splines to approximate the non-linear main effects; it still allows for inference! If you want to go beyond main effects

and want to perform inference on interactions, you can of course manually specify those. Then be warned that your sample size needs to (very) large and/or the interaction effect substantial to be able to reliably estimate them.

Is collecting the predictors for new observations costly?

Predictor variables derived from, for example (MRI) scans, blood or genetic testing, interviews or experiments might be costly to collect. In such cases, a method that performs variable selection would often be preferred, so that not all potential predictor variables will be included in the final model. A prime example of this is (relaxed) lasso regression.

Is computation costly?

If predictions have to be computed by a human decision maker, a (penalized) GLM or decision tree should likely be preferred. Of course, computations will often be done on a computer. Yet, even in completely digital environments, where data and computation come relatively cheaply, the predictive model might be queried very often and results must be returned very fast. A deep neural net may not be feasible in such cases, even though it might provide superior predictive performance. Support vector machines, random forests or gradient boosted ensembles are computationally less demanding and may present a feasible option. But the strikingly low computational load for prediction with a single decision tree or lasso regression model may still be preferred, even in high-tech applications, whenever gains in computational efficiency are of the essence.

2. Characteristics of the Data Problem

What is the sample size?

The lower the sample size, the higher the variance of your fitted model. So with smaller sample size, a method with higher bias and lower variance would be preferred. For example, GLMs and GAMs assume no interactions are present in the data. Generally speaking, interactions are always (much) weaker than the main effects. Thus, a method that only captures main effects might introduce only a little bias, but reduce variance by a lot.

Are there many predictor variables?

A GLM estimated by maximum likelihood does not allow for $p > N$. Single decision trees, support vector machines, random forests, gradient boosted ensembles, neural networks, penalized regression and Bayesian regression methods do allow for having $p > N$. Of course, none of these methods can really get rid of the curse of dimensionality, but they all have smart built-in regularization to lessen the curse.

Is this a noisy problem?

That is: Is irreducible error high? The higher the irreducible error, the higher the variance of your fitted model. So the more noisy the data problem, the more beneficial it will be to introduce a little bias (and to have larger sample size!). For example, by using only small trees or stumps in a decision tree ensemble, or by assuming linear associations. Using cross-validation to select the model-fitting parameters will likely reduce the chance of overfitting.

Are non-linearities and/or interactions sizable?

When they are not, a method assuming linearity (GLM, ridge and/or lasso penalized GLM, SVM with linear kernel) will likely do fine. When non-linear main effects are expected to be present and sizable, these can be captured well by a GAM; or by a decision tree, support vector machine with non-linear kernel, decision tree ensemble (especially when the number of splits is restricted to 1!). The same methods, except GAM, will work well when interactions are expected to be present and sizable. For tree ensembles, the tree size will then need to be > 1!

Of course, one often does not know in advance whether the non-linearities and interactions are sizable. It is then helpful to compare the performance of, for example: support vector machines with linear and non-linear kernels; gradient boosted ensembles with trees of size 1 and larger trees; elastic net regression with GAM and with random forest.

Do not try too many models!

Given our advice on comparing performance of some different models based on the training data (typically using cross-validation), you might ask yourself why you cannot simply try all models you know on a given data problem. The reason for this is again overfitting. Each time we fit and evaluate a model, we are essentially giving it a chance to get lucky and perform the best. After hundreds of models, we are very likely to find one that performs best on our data set purely by chance. This model would seem to be the best choice based on our evaluations, but its success is likely an illusion. It probably will not perform as well on future observations. This is why it is essential to follow a structured approach to model selection, which includes carefully choosing a handful of plausible models a-priori. This approach helps prevent overfitting and yields models that generalize well.