Introduction to Statistical Learning

November 21, 2020

Statistical Learning

Core concepts covered in this chapter:

- Reducible/irreducible errors
- Linear/non-linear models
- Parametric/non-parametric methods
- Supervised/unsupervised learning
- Regression/classification problems
- MSE and overfitting
- Bias-Variance Trade-Off

What is statistical learning?

We want to predict some output variable Y (also called response, dependent variable) based on some input variables X (also called predictors, independent variables, features). We suppose there exists a relationship of the form $Y = f(X) + \varepsilon$. f is some unknown function of X and ε is a random error term, independent from X and of mean zero. f represents the *systematic information* that X provides about Y. The ε are the difference between the observations and the true underlying relationship between X and Y, which is usually unknown. Statistical learning refers to a set of approaches for estimating f.

Why estimate f?

Problems of statistical learning fall into the prediction setting, the inference setting and sometimes in both. Some models lead to very good predictions (highly non-linear models for instance) while some other models do not yield good predictions but are highly interpretable (linear models for instance).

Prediction We can predict Y using $\hat{Y} = \hat{f}(X)$ where $\hat{f}$ represents an estimate for f. For prediction, $\hat{f}$ can be treated as a *black box*. Accuracy of the resulting prediction depends on the *reducible error* ($\hat{f}$ cannot be a perfect estimate of f) and the *irreducible error* (Y is also a function of some error term ε that cannot be predicted using X). The average of the squared distances between the observations Y and the predictions $\hat{Y}$ is thus given by: $E(Y - \hat{Y}) = \underbrace{E[f(X) - \hat{f}(X)]^2}_{Reducible} + \underbrace{Var[\varepsilon]}_{Irreducible}$. Statistical learning: estimating f and minimizing the reducible error.

Inference We are interested in understanding the relationship betwen X and Y, $\hat{f}$ cannot be treated as a *black box* here. We may wonder: what are the important predictors? what is the influence of each predictor on Y? can the relationship between the predictors and the response be summarized by a linear model?

How do we estimate f?

Linear and non-linear methods share some characteristics. In both cases, the goal is to used the training data $\{(x_i, y_i)\}$ to build some estimator $\hat{f}$ such that $Y \approx \hat{f}(X)$. These methods can be either *parametric* or *non-parametric*.

Parametric methods They involve a two-steps *model-based* approach. (1) Make modeling assumptions about the functional form of f (linear for instance). The problem is simplified, because we only have to estimate f from a subset of functions. (2) Pick procedure to fit/train the model. In linear setting for instance, we use the *ordinary least squares* method to estimate the parameters of $\hat{f}$. The issue is that these models usually do not match true relationship. One can increase the number of parameters to make the model more flexible, but there is then a risk of overfitting (following the noise).

Non-parametric methods No assumptions made about the functional form of f. They avoid the possible flaws of parametric methods but require a huge amount of observations to obtain a correct estimate of f as the problem is not restricted to a subset of functions. See *thin-plate spline* for instance.

Trade-Off Prediction Accuracy and Model Interpretability

Usually, as flexibility increases, interpretability decreases. For inference problem, one might prefer a more restrictive approach, that are often more interpretable. When only interested in predictions, interpretability is not a concerned. Counterintuitively, the less flexible approach sometimes lead to the best prediction because it avoids *overfitting*.

Supervised vs. unsupervised learning

Most problems fall into two paradigms: *supervised* or *unsupervised learning*. One can even sometimes encounter *semi-supervised learning* problems.

Supervised learning There is a response measurement y_i for each predictor measure x_i. Prediction or inference problems can be solved with supervised methods.

Unsupervised learning We observe measurements x_i but no response measurements associated. We are "working in the blind". We seek to understand relationships between variables (see *cluster analysis* for instance).

Regression vs. classification problems

These categories of problems are defined according to the type of response variables. *Regression methods* are used on *quantitative* response variables while *classification methods* are used to solve problems with *categorical* response variables. The predictors' variable types do not matter here.

Assessing Model Accuracy

There is no free lunch in statistics, everything comes at a cost and no method dominate all the other ones accross all possible data sets and problem types. Selecting the best model for the problem encountered is one of the most challenging task of statistical learning.

Measuring the Qualify of Fit

MSE and overfitting To estimate the quality of the estimator $\hat{f}$, we need to measure how close to the true observations and the predictions are. We use the *mean squared error* (MSE) to do so: $MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{f}(x_i))$. But this quantity is computed on our *training data set*, i.e. on previous observations. What about the quality of the model on future observations? We actually want to select the model with the *lowest test MSE*. We sometimes have some test observations to select the learning method whose test MSE is the smallest. When we don't, then the problem is that minimizing the training MSE does not mean minimizing the test MSE.

Overfitting In fact, as a model flexibility (or degrees of freedom) increases, training MSE will decrease, but test MSE may not. The model is then working too hard to find pattern in the training data: this is called *overfitting*. The test MSE is then higher because these these patterns cannot be found in the test data as they are due to the random noise ε. *Cross-validation* is a method that can be used to estimate the test MSE.

The Bias-Variance Trade-Off

In both regression and classification, choosing the appropriate model flexibility is a complicated task.

Test MSE Decomposition The U-shape of the test MSE we just described is the result of two competing properties of statistical learning methods. One can prove that the test MSE of a given value x_0 can decomposed into three quantities: the *variance of* $\hat{f}(x_0)$, the *squared bias of* $\hat{f}(x_0)$ and the *variance of the error terms* ε we described at the beginning of the chapter: $E[y_0 - \hat{f}(x_0)] = Var[\hat{f}(x_0)] + [Bias(\hat{f}(x_0)]^2 + Var[\varepsilon]$. We need a method that achieves both low variance and low bias.

Low variance The *variance* of a method refers to the amount by which $\hat{f}$ will vary when computed on different training data set. Usually, more flexible methods have higher variance, as they have more freedom to follow the data they are given.

Low bias The *bias* of a method refers to the error that occurs when trying to approximate complex real-life problems with simple models. More restrictive models like linear regression, that assumes that there is a linear relationship between Y and $X_1, \ldots, X_p$, have usually high bias because they can not fully reflect the complexity of the true f. More flexible methods have less bias.

Impact on Test MSE The variation of the test error is related to the relative rate of change of the variance and the bias. As the model flexibility increases, the reduction of bias is faster than the rise of variance and the test error decreases. But at some point, adding more degrees of freedom to the model significantly increases the variance and the test error increases.

The Trade-Off This relationship between bias, variance and test MSE is referred to as *bias-variance trade-off*. The challenge is to find a method with relative low variance and low squared bias. Always keep in mind that it's easy to find a some fancy highly flexible method that basically eliminate any bias. But it may well be outperformed by a simple and more restrictive model like a linear regression.

The Classification Setting

The same considerations apply in the classification setting but some tweaks must be made. The quality of fit is measured using the *error rate* $\frac{1}{n}\sum_{i=1}^{n} \mathbf{1}_{\{y_i \neq \hat{y}_i\}}$. The goal is to minimize $Ave[\mathbf{1}_{\{y_0 \neq \hat{y}_0\}}]$ for test observations (x_0, y_0).

The Bayes Classifier One can prove that test error we just described is minimized by a classifier that *assigns each new observation* x_0 *to the most likely class given its predictor values.* This classifier is called *Bayes Classifier*, as it relies on conditional probability and the Bayes' theorem. It assigns x_0 to the class j for which $P(Y = j|X = x_0)$ is the largest (> 0.5 actually when there are two classes). The *Bayes Error Rate* at x_0 is given by $1 - \max_j\{P(X = j|X = x_0)\}$ (analoguous to the irreducible error discussed earlier).

K-Nearest Neighbors In reality, the *Bayes classifier* is a nice gold standard against which to compare, but it often cannot be implemented on real data as we do not know the law of Y given X. Some more realistic approaches like the *K-Nearest Neighbors* try to estimate this distribution and classify the new x_0 based on this *estimated* probability. It computes the conditional probability that x_0 belongs to class j given its predictor values based the classes of K closest points ($\mathcal{N}_0$) of x_0 : $P(Y = j|X = x_0) = \frac{1}{K}\sum_{i \in \mathcal{N}_0} \mathbf{1}_{\{y_i = j\}}$. The choice of K has a huge impact here on the test error. The lower K, the lower the bias but the higher the variance.