ISL ch. 2: Statistical Learning

(2.1) Try to predict [output variable / response / dependent variable / Y] from [input variables / predictors / independent variables / features / X]

How? Develop an accurate model.

More formally: We observe a quantitative response Y and p predictors $X_1, \ldots, X_p$. We assume there is some relationship between Y and $X = (X_1, \ldots, X_p)$ which can be written as

$$Y = f(X) + \varepsilon$$

where f is some fixed but unknown function of X and ε is a random error term independent of X and with mean zero. f represents the systematic information that X provides about Y.

Statistical learning refers to a set of approaches for estimating f.

Why estimate f? **Prediction** and **Inference**.

1. Prediction — often we have X but can't get Y. So we estimate f as $\hat{f}$ and make a prediction $\hat{Y} = \hat{f}(X)$.
   → Treat $\hat{f}$ as a black box — all that matters is that it makes good predictions.

Accuracy of predictions depends on:

1. Reducible error: $\hat{f}$ won't be a perfect estimate, but we can use sophisticated methods to improve it.
2. Irreducible error: Y is also a function of ε, which is independent of X. (It may contain unmeasured variables, or variation, e.g. "patient feelings on given day")

Thus

$$E(Y-\hat{Y})^2 = E\left[f(X)+\varepsilon-\hat{f}(X)\right]^2$$

$$= \underbrace{\left[f(X)-\hat{f}(X)\right]^2}_{\text{reducible}} + \underbrace{\text{Var}(\varepsilon)}_{\text{irreducible}}$$

expected value of the squared diff.

Focus of statistical learning methods is estimating f while minimizing reducible error.

(Irreducible error provides upper bound, always unknown)

2. Inference — often we want to understand the relationship between X and Y. Here $\hat{f}$ isn't a black box — we want to know:

→ Which predictors are associated w/ the response?
→ What is the relationship between the response and each predictor?
→ Is the relationship linear or more complicated?

★ Some problems involve prediction, some inference, and some both. Depending on which will lead to different choices:

E.g. linear model may provide helpful understanding of relationships (i.e. inference) but worse predictions

vs.

deep learning may provide good predictions but worse interpretability (i.e. inference)

How to estimate f

Various ways, but common characteristics.

Always assume: we have training data of n observed data points, each with p predictors:

$\{(x_1, y_1), \ldots, (x_n, y_n)\}$ where $x_i = (x_{i1}, \ldots, x_{ip})^T$

and $x_{ij} = j^{th}$ predictor of observation i

and y_i = response for i^{th} observation

Parametric methods

Two steps:

1. Assume the functional form/shape of f (e.g. assume linear, $f(X) = \beta_0 + \beta_1 X_1 + \ldots + \beta_p X_p$)
2. After model selection, use training data to fit/train the model. (e.g. for linear, estimate the params $\beta_0, \beta_1, \ldots, \beta_p$)

Methods include least squares and others.

Tradeoffs:

Pro: Easier to estimate params than completely unknown fn.

Con: Usually won't match true form of f.

↳ Try to pick flexible models, but too complex can be overfit (i.e. they follow the noise/error).

Non-Parametric

No assumptions about functional form — just estimate as close to data points as possible without being "too rough or wiggly".

Tradeoffs:

Pro: Can flexibly fit wide range of shapes w/out limitations imposed by assumption

Con: Requires huge amt. of data to train (since the problem hasn't been reduced to estimating small # of params.)

Accuracy vs. Interpretability

For inference it might be worth it to choose a less flexible but highly interpretable model (such as linear reg.)! Whereas when prediction is most impt. and interpretability isn't (e.g. stock prices), we might want more flexible, fully nonlinear models — if they perform better...

↳ But sometimes simpler models avoid overfitting!

Supervised vs. Unsupervised

Supervised: for each observation there is a response, and we use these responses to fit a model in order to better predict on new observations. E.g. linear and logistic reg., SVMS, etc.

Unsupervised: No responses w/ the observations! Can still try to understand relationships between the input vars / observations. E.g. clustering.

Sometimes **semi-supervised**: n observations, $m < n$ responses. Want to consider these and learn more than w/out them. (Not in book.)

Regression vs. Classification

Variables are **quantitative** (numbers) or **qualitative** (categorical) (values from k classes).

Regression — quant. response

Classification — qual. response

Not always clear: logistic regression produces class probabilities (regression problem) but often used for two-class/binary qual. response (classification).

Some methods support either: K-NN, boosting, etc.

★ **Predictor** type is irrelevant! Just need to encode them.

2.2 ASSESSING MODEL ACCURACY

Why so many methods?

→ No free lunch in statistics: no one method dominates all others over all possible data sets.

→ Impt. to decide which is best for each problem. This is one of the biggest challenges in practice!

Quality of fit

Need to measure how close predictions are.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{f}(x_i))^2$$

the mean squared error is most common in regression.

Note that in training we try to minimize the training MSE, but the actual objective is to minimize the test MSE — the MSE on unseen observations

→ As model flexibility increases (eg. w/ more degrees of freedom), training MSE decreases, but the test MSE may not!

→ Flexibility can lead to overfitting, i.e. picking up patterns that are due to randomness (ε) instead of true f.

Hard to estimate test MSE! Cross-validation (ch. 5) can help.

Bias/Variance Trade-off

It can be shown that the expected test MSE for a value x_0 can be decomposed into the sum of ① the variance of $\hat{f}(x_0)$, ② the squared bias of $\hat{f}(x_0)$, and ③ the variance of the errors ε:

$$E(y_0 - \hat{f}(x_0))^2 = \text{Var}(\hat{f}(x_0)) + [\text{Bias}(\hat{f}(x_0))]^2 + \text{Var}(\varepsilon)$$

(Note: Var ≥ 0 always)

The overall expected test MSE is the avg. of this over all vals x_0 in test set.

So: to minimize expected test error, choose a method to achieve low variance and low bias. But the floor is always Var(ε)!

Variance is the amt. $\hat{f}$ changes with a different training set. More flexible methods have higher variance!

Bias is the error introduced by approximating a real-life problem w/ a much simpler model. More flexible methods have less bias!

This tradeoff is an important recurring theme.

- Can always get low bias — fit curve to every point, but will cause high variance
- Similarly, can always get low variance — always fit a horizontal line, but ⇒ high bias.

Always keep in mind: simpler model might be best!

Classification setting

Many of the concepts (incl. bias/variance trade-off) carry over:

We seek to estimate f given observations $\{(x_1, y_1), \ldots, (x_n, y_n)\}$ where $y_1, \ldots, y_n$ are qualitative.

Then (training) error rate is $\frac{1}{n}\sum_{i=1}^{n} I(y_i \neq \hat{y}_i)$ where

$\hat{y}_i$ is predicted class label for i^{th} observation using $\hat{f}$ and $I(y_i \neq \hat{y}_i) = \begin{cases} 1 & \text{if } y_i \neq \hat{y}_i \\ 0 & \text{else.} \end{cases}$ (indicator variable)

So a good classifier is one for which the test error rate $\text{Ave}(I(y_0 \neq \hat{y}_0))$ is smallest given test observations (x_0, y_0).

Bayes Classifier

It can be shown that the test error rate is minimized on average by a simple classifier that assigns each observation to the most likely class given its predictor values. I.e.

assign x_0 to j s.t. $P(Y=j \mid X=x_0)$ is highest.

this is the Bayes Classifier

For binary (two-class) problems: $\hat{f}(x_0) = \begin{cases} 1 & \text{if } P(Y=1 \mid X=x_0) > 0.5 \\ 0 & \text{if } \leq 0.5. \end{cases}$

The Bayes decision boundary is the line s.t. an observation falling one one side is assigned to one class and falling on the other to the second class. I.e. (x,y) s.t. $P(Y=y|X=x)=0.5$.

The Bayes error rate (the lowest possible test error rate) is

$$1-E\left(\max_j P(Y=j|X)\right)$$

↳ avg. over all possible values of X

This is analagous to the irreducible error. You can't always reduce it to zero! The classes may overlap, etc.

Note: Computing this requires knowledge of the conditional distribution $P(Y|X)$, and we don't for real data!

→ So we can't compute the Bayes classifier. We just use it to compare other methods.

Some methods estimate the conditional distribution and classify to the class w/highest estimated prob. E.g.:

K-nearest neighbors classifier (KNN)

Given $K \in \mathbb{N}$ and test observation x_0, find K nearest points in training data N_0 and estimate

$$P(Y=j|X=x_0)=\frac{1}{K}\sum_{i\in N_0} I(y_i=j).$$

Classify x_0 to j s.t. this is maximized.

(What fraction of the K nbrs belongs to each class — then maximize)

KNN can produce classifiers surprisingly close to Bayes classifier! But choice of K has huge impact:

Smaller K: flexible, low bias, high variance

Larger K: more rigid, high bias, low var.

Bias/variance tradeoff applies:

Increase flexibility $\Rightarrow$ training err. down
test err. may not!
(e.g. U-shape w/overfitting)