Lecture Notes: Statistical Learning and Classification

May 2, 2025

Contents

1 Bayesian Logic

In a general setup, the problem can be formalized as follows:

- **H** is a proposition to be checked
- **E** is the empirical evidence

We assign to each hypothesis **H** a degree of uncertainty, mathematically described by a probability. In a Bayesian setup, there are two different probabilities to summarize uncertainty:

- **prior probability**: gives the amount of uncertainty of the proposition **H**, independently on the empirical evidence provided by **E**.
- **likelihood**: gives the level of uncertainty referring to the reliability of the empirical evidence **E**. It summarizes how the empirical evidence **E** is consistent with the hypothesis **H**.

Combining prior and likelihood, we get the **posterior probability** through the **Bayes' formula**

$$P(H_i|E_j) = \frac{P(H_i, E_j)}{P(E_j)} = \frac{P(E_j|H_i)P(H_i)}{\sum_k P(H_k)P(E_j|H_k)} \quad \forall\ i, j\ . \tag{1}$$

This probability describes (and updates) the uncertainty on H_i once the event E_j has been obeserved.

1.1 A general framework

Suppose now we get two random variables X and Y and we aim at checking how information on X can help predicting the behaviour of Y. If X is not observed, the best we can do is to evaluate $P(Y = y)$. However, knowing some extra information, say $X = x$, we might obtain improved estimates on the probability to observe $Y = y$, in the sense the knowledge of X can help on removing uncertainty regarding Y. Therefore, we might be interested to compute

$$P(Y|X) = \frac{P(X, Y)}{P(X)} = \frac{P(X|Y)P(Y)}{\int P(y)P(X|y)dy}\ . \tag{2}$$

Without knowing X, the average behaviour of Y is summarized by $E[Y]$, that can be estimated through a sampling average $\bar{Y}$. We can improve our knowledge on the expected Y once we get information about X. We aim at computing the conditional expectation $E[Y|X]$, which is the average of Y computed from the conditional probability $P(Y|X)$. The conditional expectation is often approximated by an analytical function as $E[Y|X] \approx f(X)$. The choice of f sometimes is called **Statistical Learning**.

The conditional probability $E[Y]$ is the best predictor of Y since it minimizes the MSE (mean squared error) $E[(Y - f(X))^2]$, i.e.

$$E[(Y - f(X))^2] \geq E[(Y - E[Y|X])^2]\ . \tag{3}$$

Then, $f(X) = E[Y|X]$ is the **optimal predictor** under the MSE rule.

2 Statistical Learning

Suppose we observe a quantitative response Y and p different predictors, $X_1, X_2, \ldots, X_p$. We assume that there is some relationship between Y and $X = (X_1, X_2, \ldots, X_p)$, which can be written in the very general form

$$Y = f(X) + \epsilon\ , \tag{4}$$

where f is some fixed but unknown function of X and ϵ is a random error term, which is independent on X and has mean zero.

When the variable Y is *quantitative*, its prediction through f is referred as **regression**, whereas, if Y is *qualitative* or *categorical* prediction refers to **classification**.

There are two main reasons why we want to estimate f:

- **prediction**: estimate Y given X
- **inference**: understand the form of f

2.1 Predictions

In many situations, a set of inputs X are available, but the output Y cannot be easily obtained. In this setting, since the error term averages to zero, we can predict Y using $\hat{Y} = \hat{f}(X)$, where $\hat{f}$ is our estimate for f and $\hat{Y}$ is the resulting prediction for Y (using $\hat{f}$).

The prediction error can be decomposed into a *reducible part* (due to the accuracy of the estimate f versus f) and an *irreducible part* (due to the variance of the error term ϵ)

$$\begin{aligned} E[(Y - \hat{Y})^2|X] &= E[(f(X) + \epsilon - \hat{f}(X))^2|X] \\ &= [f(X) - \hat{f}(X)]^2 + Var(\epsilon|X) \ , \end{aligned}$$

where the first term is the reducible one and the second is the irreducible one.

Methods for estimating f divides in

- **parametric**
- **non-parametric**

In **parametric methods** we make an assumption about the functional form, or shape, of f. For example, one very simple assumption is that f is linear in X

$$f(X) = \beta_0 + \beta_1 X_1 + \ldots \beta_p X_p \ . \tag{5}$$

With parametric models, even nonlinearities are possible:

$$f(X) = \frac{e^{\beta_0 + \beta_1 X_1 + \ldots \beta_p X_p}}{1 + e^{\beta_0 + \beta_1 X_1 + \ldots \beta_p X_p}} \ . \tag{6}$$

Full knowledge of f is obtained once we find estimates for the parameters $\beta_0, \beta_1 \ldots, \beta_p$.

On the contrary, **non-parametric models** do not explicity assume a parametric form for f. An example is provided by the **K-Nearest Neighbors** (KNN). KNN works as follow: given a value for K and a prediction point x_0, KNN regression first identifies the K training observations that are closest to x_0 , represented by N_0 . It then estimates $f(x_0)$ using the average of all the training responses in $\mathcal{N}_0$, or

$$\hat{f}(x_0) = \frac{1}{K} \sum_{x_i \in \mathcal{N}_0} y_i \ . \tag{7}$$

Another example of non-parametric model is provided by **spline functions**, which are regressions with predictiors $b_j(x_i)$ such as

$$y_i = \beta_0 + \beta_1 b_1(x_i) + \ldots + \beta_k b_k(x_i) + \epsilon_i \ . \tag{8}$$

They involve dividing the range of X into K distinct regions. Within each region, a polynomial function is fit to the data. However, these polynomials are constrained so that they join smoothly at the region boundaries, or knots. Provided that the interval is divided into enough regions, this can produce an extremely flexible fit. Note that the functions $b_1, b_2, \ldots b_K$ fixed and known.

2.2 Model accuracy

In order to evaluate the performance of a statistical learning method on a given data set, we need some way to measure how well its predictions actually match the observed data.

In the regression setting, the most commonly used measure is the mean squared error (MSE), given by

$$MSE = \frac{1}{n}\sum_{i=1}^{n}\Big(y_i - \hat{f}(x_i)\Big)^2 \ . \tag{9}$$

The MSE is computed using the training data and should be called *training MSE*. But in general, we do not really care how well the method works on the training data. Rather, we are interested in the accuracy of the predictions that we obtain when we apply our method to previously unseen test data.

We want to choose the method that gives the lowest *test MSE*. In other words, if we had a large number of test observations (x_0, y_0), we could compute

$$E\left[\Big(y_0 - \hat{f}(x_0)\Big)^2\right] \tag{10}$$

that is, the average squared prediction error for these test observations (x_0, y_0).

2.2.1 bias-variance tradeoff

The expected *test MSE*, for a given value x_0 , can always be decomposed into the sum of three fundamental quantities:

- the variance of $\hat{f}(x_0)$,
- the squared bias of $f(x_0)$,
- the variance of the error terms ϵ,

that is

$$E\left[y_0 - \hat{f}(x_0)|x_0\right] = \underbrace{(f(x_0) - E[\hat{f}(x_0)])^2}_{\text{bias}^2} + \text{Var}(\varepsilon) + \underbrace{E\left[(\hat{f}(x_0) - E[\hat{f}(x_0)])^2\right]}_{\text{variance}} \ .$$

- **Variance** refers to the amount by which $\hat{f}$ would change if we estimated it using a different training data set. Indeed, different training data sets will result in a different $\hat{f}$. But ideally, the estimate for f should not vary too much between training sets. However, if a method has high variance, then small changes in the training data can result in large changes in $\hat{f}$.
- **Bias** refers to the error that is introduced by approximating a real-life problem, which may be extremely complicated, by a much simpler model. For example, linear regression assumes that there is a linear relationship between Y and $X_1, X_2, \ldots, X_p$. It is unlikely that any real-life problem truly has such a simple linear relationship, and so performing linear regression will undoubtedly result in some bias in the estimate of f.

As a general rule, as we use more flexible methods, the variance will increase and the bias will decrease. As we increase the flexibility of a class of methods, the bias tends to initially decrease faster than the variance increases. Consequently, the expected *test MSE* declines. However, at some point, increasing flexibility has little impact on the bias but starts to significantly increase the variance. When this happens the test MSE increases.

2.2.2 Overfitting

When a given method yields a **small training MSE** but a **large test MSE**, we are said to be **overfitting** the data.

In general, a sample (or training set) is just a limited random representation of a population. Different samples differ because they are random. If the model fit too well the training set, it means it works very well on a unique sample, that is not the population of interest. Results from an *overfitted model* can't be generalized to different samples and not even to the population.

Cross-validation is a method to estimate test error by holding out a subset of training data from the fitting process. **k-fold Cross-Validation** divides the training data into k folds; each fold is used as a validation set once, while the model is fit on the remaining $k-1$ folds. This process results in k estimates of the test error, $MSE_1, MSE_2, \ldots, MSE_k$. The test error is estimated by averaging the k resulting MSE estimates. This provides a more reliable estimate of how the model will perform on unseen data compared to training MSE.

2.3 Classification

In **classification**, the goal is to predict a qualitative response (assigning an observation to a category or class). Classification methods often predict the probability of belonging to each class (the posterior probability) as a basis for classification.

2.3.1 Logit models

For a qualitative variable with two levels (e.g., Success/Failure, represented as 1/0), a **Logit model** is a common parametric strategy to approximate the probability $P(Y = 1|X)$. The Logit function

$$f(\beta_0, \beta_1, \ldots, \beta_k, X) = \frac{e^{(\beta_0+\beta_1 X_1+\ldots\beta_k X_k)}}{(1 + e^{(\beta_0+\beta_1 X_1+\ldots\beta_k X_k)})} \tag{11}$$

is appropriate to describe probability because it takes values is between 0 and 1. Once the parameters are estimated, predictions are typically made by assigning the observation, $\hat{y}_i$, to class 1 if the predicted probability $\hat{f}(x_i) > 0.5$, and to class 0 otherwise.

2.3.2 Classification: model accuracy

The most common approach for quantifying the accuracy of our estimate $\hat{f}$ is the training error rate, the proportion of mistakes that are made if we apply our estimate $\hat{f}$ to the training observations:

$$\frac{1}{n}\sum_{i=1}^{n} I(y_i \neq \hat{y}_i) \ .$$

This is the fraction of incorrect classifications, since $I = 1$ if $y_i \neq \hat{y}_i$, and zero otherwise.

2.3.3 Naive Bayes classifier

Suppose we got a number $k \geq 2$ of classes. How to define a classification strategy based on the *posterior probability*? Here posterior means we got evidences from some predictors

X. The Naive Bayes classifier is based directly on Bayes' Theorem

$$P(Y = k|X) = \frac{\pi_k f_k(x)}{\sum_j \pi_j f_j(x)} \ ,$$

where $f_k(x) = P(X = x|Y = y)$ and π_k represents the overall or prior probability that a randomly chosen observation comes from the k-th class. We assign an observation to a class $k \in K$ if the posterior probability $P(Y = k|X)$ is largest.

If we assume that $f_k(x)$ is normal or Gaussian, in one dimension, we have

$$f_k(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\Big\{-\frac{1}{2\sigma^2}(x - \mu_k)^2\Big\} \ .$$

The **linear discriminant analysis** (LDA) method approximates the Bayes classifier by plugging estimates for π_k, μ_k and σ_k based on a training set. In particular, the following estimates are used

- $\hat{\mu}_k = \frac{1}{n_k} \sum_{i:y_i=k} x_i \ \ ,$
- $\hat{\sigma}^2 = \frac{1}{n-K} \sum_{k=1}^{K} \sum_{i:y_i=k} (x_i - \mu_k)^2 \ \ ,$
- $\hat{\pi}_k = \frac{n_k}{n} \ \ .$