Machine Learning Notes

Giacomo Belleri

July 2024

Contents

Bibliography

These notes are produced using the following resources:

[1] P. Mehta *et al.*, "A high-bias, low-variance introduction to Machine Learning for physicists," *Physics Reports*, vol. 810, pp. 1–124, May 2019, arXiv:1803.08823, ISSN: 03701573. DOI: 10.1016/j.physrep.2019.03.001. [Online]. Available: http://arxiv.org/abs/1803.08823 (visited on 08/02/2024).

[2] A. Ng and T. Ma, *CS229 Lecture Notes*, en. Nov. 2023. [Online]. Available: https://cs229.stanford.edu/main_notes.pdf (visited on 08/02/2024).

[3] G. James, D. Witten, T. Hastie, R. Tibshirani, and J. Taylor, *An Introduction to Statistical Learning: with Applications in Python* (Springer Texts in Statistics), en. Cham: Springer International Publishing, 2023, ISBN: 978-3-031-38746-3 978-3-031-38747-0. DOI: 10.1007/978-3-031-38747-0. [Online]. Available: https://link.springer.com/10.1007/978-3-031-38747-0 (visited on 08/02/2024).

Notation

1 Introduction To Statistical Learning

1.1 Basis and Goals Of Statistical Learning

The statistical distribution of a particular quantity y (*response* or *output/dependent variable*) is determined by the relationship between itself and the quantities that affect its values. These quantities $\vec{\mathbf{x}} = (x_1, x_2, \ldots, x_n)^T$ are known as *input/independent variables* or *predictors*. This relationship is modeled mathematically by the function $f(\vec{\mathbf{x}})$ and a *random error term* ε as follows:

$$y = f(\vec{\mathbf{x}}) + \varepsilon \tag{1}$$

In Eq,1, the function f contains the systematic information that $\vec{\mathbf{x}}$ provides about y. The ε term quantifies the *irreducible error* arising from unmeasured variables or unmeasured variation in the dataset. It is independent of $\vec{\mathbf{x}}$ and it has a mean of zero i.e. $\langle \varepsilon \rangle = 0$.

Statistical learning refers to a set of methods that can be used to *estimate the function f with $\hat{f}$*. This estimation can be used for two purposes:

1. *Prediction*: Use estimated function $\hat{f}$ to predict the associated response $\hat{y} = \hat{f}(\vec{\mathbf{x}})$
2. *Inference*: Use estimated function $\hat{f}$ to analyse the relationship between $\vec{\mathbf{x}}$ and y e.g. determine parameters of fit.

1.1.1 Basis of Statistical Learning in Matrix Notation

A more mathematical approach summarises the dataset as $\mathcal{D} = (\mathbf{X}, \vec{\mathbf{y}})$. The matrix $\vec{\mathbf{X}}$ contains all observations (column) vectors $\vec{\mathbf{x}}^{(i)}$ as a rows[1]. The matrix has thus dimension $n \times p$ where n is the number of observations and p is the number of predictors for each observation. On the other hand, the vector $\vec{\mathbf{y}}$ is a $n \times 1$ vector containing the response values to each independent variable observation. These vectors are reported in Eq.2.

$$\vec{\mathbf{x}}^{(i)} = \begin{bmatrix} x_1^{(i)} \\ x_2^{(i)} \\ \vdots \\ x_p^{(i)} \end{bmatrix} \qquad \mathbf{X} = \begin{bmatrix} -- & (\vec{\mathbf{x}}^{(1)})^T & -- \\ -- & (\vec{\mathbf{x}}^{(2)})^T & -- \\ & \vdots & \\ -- & (\vec{\mathbf{x}}^{(n)})^T & -- \end{bmatrix} \qquad \vec{\mathbf{y}} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} \tag{2}$$

Each observation in $\vec{\mathbf{y}}$ satisfies Eq.1 such that $f : \mathcal{C}^{p\times 1} \to \mathcal{C}$. Then, to estimate f, a function $\hat{f}(\vec{\mathbf{x}}, \vec{\theta})$ that depends on observation $\vec{\mathbf{x}}$ and parameters $\vec{\theta}$ can be used. Using the matrices and vectors of Eq.2, f can be interpreted as map $f : \mathbf{X} \to \vec{\mathbf{y}}$ such that $f : \mathcal{C}^{n\times p} \to \mathcal{C}^{n\times 1}$.

[1] The reason for this switch is to be found in the dimensions of the matrices and vectors involved. This will become clearer when linear regression is analysed in Sec.2.

1.1.2 Prediction

Accurately predicting $\hat{Y}$ is useful in many applications and a big part of data analysis is determining the most accurate approach to estimate $\hat{Y}$. The accuracy of the estimate is determined by two types of error:

1. *Reducible*: Error produced by the model - Can be reduced by tweaking or changing the model
2. *Irreducible*: Built into the dataset by ε - Cannot be reduced as it is entirely random and independent of X

The difference between the two errors can be made clearer by considering n responses Y_i measured for the same set of predictors X. These different responses are determined by the function $f(X)$ and the random error ε, which ensures variability in the responses Y_i. The average squared difference between the measured and predicted responses is given by the following equation:

$$\begin{aligned}
\langle (Y-\hat{Y})^2 \rangle &= \frac{1}{n}\sum_{i=1}^{n}(Y_i-\hat{Y})^2 = \frac{1}{n}\sum_{i=1}^{n}[f(X)+\varepsilon-\hat{f}(X)]^2 = \\
&= \frac{1}{n}\sum_{i=1}^{n}\left\{\left[f(X)-\hat{f}(X)\right]^2+\varepsilon^2+2[f(X)-\hat{f}(X)]\varepsilon\right\} = \\
&= \left[f(X)-\hat{f}(X)\right]^2+\langle\varepsilon^2\rangle+2[f(X)-\hat{f}(X)]\langle\varepsilon\rangle = \\
&= \underbrace{\left[f(X)-\hat{f}(X)\right]^2}_{Reducible}+\underbrace{\mathrm{Var}(\varepsilon)}_{Irreducible}
\end{aligned} \tag{3}$$

Note that the irreducible error is truly irreducible only for the situation mentioned above. In most projects, more sets of predictors (with different responses) are available. This allows the estimation of $\hat{f}$. However, if the model is overfitted, the noise associated with ε actively contributes to $\hat{f}$. As a result, by tweaking the model, it is possible to reduce the error associated with ε. This is discussed in more detail in later sections.

1.1.3 Inference

In many scientific contexts, it is interesting to determine the model's shape and the value of the associated parameters. This can be done through an inference-focused application of statistical learning.

Inference-based approaches can answer the following questions:

1. *Is there a relationship between the response Y and the predictor X_i?*
   Determines which predictors are associated with the response Y.
2. *What type of relationship is there between Y and X_i?*
   Determines whether the relationship is linear, polynomial, or more complex.
3. *How large is the association between the predictor(s) and the response?*
   Determines the influence on the response of changes in the predictors.
4. *Is there synergy/interaction between predictors?*
   Determines whether multiple predictors can influence each other's response. For example, $X_1 \cdot X_2$ constitutes an interaction between two predictors as changes in X_1 affect X_2.

1.2 Methods of Estimation of f

To estimate a model, a *training dataset* of n observations (Y_i for $i \in [1,n]$) with p predictors (x_{ij} for $j \in [1,p]$) is used. This dataset can be summarised as:

$$\{(X_1,Y_1),(X_2,Y_2),\ldots,(X_n,Y_n)\} \qquad \text{with} \qquad X_i = (x_{i1},x_{i2},\ldots,x_{ip})^T \tag{4}$$

There are various methods to estimate f. A few categories are described in the following sections.

1.2.1 Parametric Vs Non-Parametric

- *Parametric*: Reduces estimation of f to estimation of parameters (e.g. Linear regression). Parametric methods are based onto steps:

  1. Guessing of functional form of the model from data distribution (and theory?)
  2. Fitting (Training) model according to some error minimization procedure

  Note that the method is highly dependent on chosen functional form. If the guessed functional form is not close to the true functional form the fit will be poor. This can be solved by using more flexible methods. However, this leads to losses of inference capabilities (*loss of interpretability*) and can lead to considerable *overfitting*.

- *Non-Parametric*: Seek estimate of f without fitting parameters.
  Avoids issue of guessed function being different from true functional form but require large number of observations for accurate estimate.

1.2.2 Supervised Vs Unsupervised

- *Supervised*: For each observation of the predictor measurements there is an associated response measurement. Examples include *linear regression*.

- *Unsupervised*: There is no associated response measurement to predictor vector measurements. *Clustering* algorithms are the main example.

- *Semi-supervised*: Mixture of supervised and unsupervised

1.2.3 Regression Vs Classification

Variables can be classified into two main categories:

1. *Quantitative*: The variables are numerical. Quantitative responses are often analyzed through *Regression* algorithms.

2. *Categorical*: Qualitative variables that take a value of one out of K different classes. Categorical responses are often analyzed through *Classification* algorithms.

Note: Often whether a response is quantitative or qualitative is more important than the type of the predictors, which can often be coded to move from categorical to quantitative.

1.3 The Quality of Fit

To analyze whether a fit is appropriate and precise, it is possible to use a cost function, which is often minimized on the test data set. An example of cost functions is proposed in Sec.1.3.2 and further analyzed in later chapters. The minimization criteria of *Bias-Variance trade-off* is then treated in Sec.1.3.3.

1.3.1 The Cost Function

In addition to the dataset $\mathcal{D}$ and the model $\hat{f}$, a fundamental ingredient of of statistical learning is the *cost function* $C(\vec{\mathbf{y}},\ \hat{f}(\mathbf{X},\vec{\theta}))$. This function is evaluated for various models to select the most appropriate and best performing for the dataset $\mathcal{D}$.

The cost function is often combined with the tool of *Cross Validation*. Using this method, the dataset is split into a *training subset* $\mathcal{D}_{\text{train}}$ and a *testing subset* $\mathcal{D}_{\text{test}}$. For each model option, the cost function is evaluated in both subsets. This leads to the "*in-sample error*" $E_{in} = C(\vec{\mathbf{y}}_{\mathbf{train}},\ \hat{f}(\mathbf{X}_{\mathbf{train}},\vec{\theta}))$ and the "*out-of-sample error*" $E_{out} = C(\vec{\mathbf{y}}_{\mathbf{test}},\ \hat{f}(\mathbf{X}_{\mathbf{test}},\vec{\theta}))$. Generally, a model is selected by choosing parameters that produce a low E_{in}. However, the minimization of the in-sample error may be due to overfitting to the training data set as it is not uncommon for E_{in} to decrease as model complexity increases. Therefore, to evaluate the true performance of the dataset, the appropriate model is chosen based on the performance on previously unseen data. Therefore, the model that minimizes the cost function E_{out} is selected[2].

Takeaway: To learn the parameters $\vec{\theta}$, the model(s) are trained on a *training dataset* and their performance is evaluated on a different dataset previously unseen i.e. *test dataset*. The model is selected to minimize the *out-of-sample error* $E_{out} = C(\vec{\mathbf{y}}_{\mathbf{test}},\ \hat{f}(\mathbf{X}_{\mathbf{test}},\vec{\theta}))$.

1.3.2 An Example: Mean Squared Error

One of the most common procedures to measure the quality of fit of a certain model to data is to analyze the *Mean Squared Error (MSE)* cost-function. To appropriately fit a model, this cost function is often minimized in a procedure known as *Least-Squares Fitting*.

While computing the training MSE is often useful, the test function is calculated and minimized on the test dataset, which is unseen by the trained model. This ensures that the model is properly fitted for all data and does not pick up noise from the training dataset (overfitting). This is related to the bias-variance trade-off treated in the next section.

To compute the MSE, consider the dataset D to which the model f is fitted. It follows that the average squared deviation is given by:

$$MSE = \frac{1}{k}\sum_{i=1}^{k}(y_i - \hat{f}(x_i))^2 = \frac{1}{dim(\vec{\mathbf{y}})}\left(\vec{\mathbf{y}} - \hat{f}(\mathbf{X},\ \vec{\theta})\right)^T\left(\vec{\mathbf{y}} - \hat{f}(\mathbf{X},\ \vec{\theta})\right) \tag{5}$$

[2] As the test dataset is used in model selection, the test dataset becomes an implicit part of the training and selection process. Therefore, after model selection, the dataset is not "unseen" anymore. Because of this, real-world performance is expected to be slightly worse.

1.3.3 Bias-Variance Trade-Off

The training of a model can be a challenging task as the choice of models and parameters is governed by the "*Bias-Variance Trade-Off*".

To more accurately represent the behavior of the training data, a more complex model can be chosen. However, if the model is too complex, the fitting procedure might incorporate "*noise*", the random errors of the analyzed response. Therefore, high-complexity models tend to *overfit* the training dataset and perform poorly on unseen data. Overfitting can also be achieved in low-complexity models by the fine-tuning of parameters. In both cases, overfitting is associated with the variance of the dataset.

In addition to overfitting by fine-tuning, simpler models tend to present a different issue. Due to their lower complexity, these models are unable to represent more complex patterns in the dataset. For example, a linear fit is unable to represent a quadratic curve for large predictor intervals. The inability of the model response to fit and predict the true response effect is due to the *bias* of the chosen model.

2 Linear Regression