Linear Classification and Logistic Regression [*]

February 27, 2025

1 Linear Classification

1.1 Background

In the previous lectures, we explored a class of linear regression models that have particularly simple analytical and computational properties. We now discuss an analogous class of linear models for solving classification problems. The goal in classification is to take an input vector $\mathbf{x}$ and to assign it to one of the multiple discrete classes Y. In the most common scenario, the classes are taken to be disjoint, so that each input is assigned to one and only one class. The input space is thereby divided into decision regions whose boundaries are called *decision boundaries* or *decision surfaces*.

In this lecture, we consider linear models for classification, by which we mean that the decision boundaries are linear functions of the input vector $\mathbf{x}$ and hence are defined by $D-1$ dimensional hyperplanes within the D dimensional input space. Datasets whose classes can be separated exactly by linear decision surfaces are said to be **linearly separable**.

Our course includes two types of classification approaches:

- Generative approach. It first solves the inference problem of determining the class-conditional densities $p(\mathbf{x}|y_i)$ for each class y_i individually. Then, it uses Bayes' theorem in the form:
$$P(y_i|\mathbf{x}) = \frac{p(\mathbf{x}|y_i)P(y_i)}{p(\mathbf{x})}$$
to find the posterior class probabilities $P(y_i|\mathbf{x})$. Equivalently, we can model the joint distribution $p(\mathbf{x}, y_i)$ directly and then normalize to obtain the posterior probabilities. Having found the posterior probabilities, we use decision theory to determine class membership for each new input $\mathbf{x}$. Approaches that explicitly or implicitly model the distribution of inputs, as well as outputs, are known as generative models.

- Discriminant function: Find a function $f(\mathbf{x})$, called a discriminant function, which maps each input $\mathbf{x}$ directly onto a class label. For instance, in the case of two-class problems, f might be binary valued and such that $f = 1$ represents class y_1 and $f = -1$ represents class y_2. In this case, probabilities play no role.

[*] References
- Christopher Bishop. Pattern Recognition and Machine Learning. 2006

1.2 Standard Linear (Fisher) Classification

Linear classification (also known in some textbooks as Fisher classification) is one of the simplest representations of a discriminant function, obtained by taking a linear function of the input vector so that

$$y(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + b, \tag{1}$$

where **w** is called a weight vector, and b is a bias. An input vector **x** is assigned to class y_1 if $y(\mathbf{x}) > 0$ and to class y_2 otherwise. The corresponding decision boundary is therefore defined by the relation

$$y(\mathbf{x}) == 0, \tag{2}$$

which corresponds to a $D-1$ dimensional hyperplane within the D dimensional input space.

Given a set of input **x** with corresponding class labels y, there are several approaches for learning the parameters **w** and b in this linear discriminant function (i.e., least squares, Fisher's linear discriminant, perceptron algorithm). In this lecture, we focus only on the first approach. Recall that in the lecture on linear regression, we saw that the minimization of a sum-of-squares error function led to a simple closed-form solution for the parameter values. Similarly, we can use least squares for solving a classification problem. This is done by treating discrete class labels as the output values. Figure 1 gives an illustration.

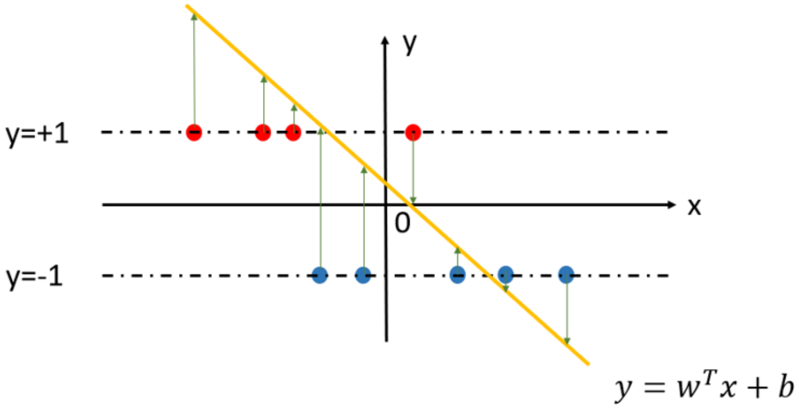

Figure 1: Least squares for classification.

In this example, the input feature vector x is a scalar. Class y_1 is assigned with the value +1, and class y_2 is assigned with the value -1. In such a way, the classification problem is approximated to a regression problem and could be solved using least-squares introduced in previous lectures.

The least-squares approach gives an exact closed-form solution for the discriminant function parameters. However, it lacks robustness to outliers. As illustrated in Figure 2, we see that a single outlier data point in the right-hand figure will cause a significant change in the location of the decision boundary, even though these points would be correctly classified by the original decision boundary in the left-hand figure.

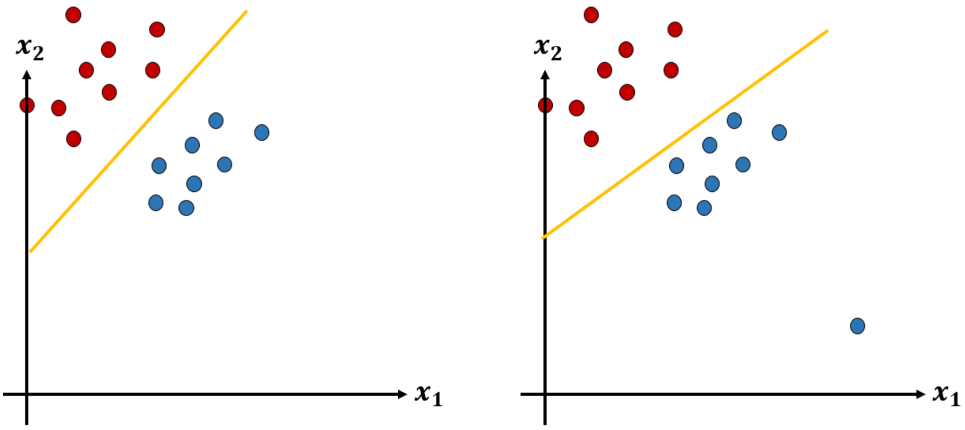

Figure 2: Linear classification problem in a 2D feature space. Different colors of dots indicate different classes. The left figure shows the original decision boundary obtained from least squares. The right figure shows that adding a single outlier point will cause a non-trivial change to the classifier.

2 Logistic Regression

Logistic regression, despite its name, is a classification model rather than a regression model. Logistic regression is a simple and more efficient method for binary and linear classification problems. It is a classification model that is very easy to realize and achieves very good performance with linearly separable classes. The primary difference between linear regression and logistic regression is that logistic regression's range is bounded between 0 and 1. In addition, as opposed to linear regression, logistic regression does not require a linear relationship between inputs and output variables [1].

2.1 Logistic Model

In contrast to Section 1, we turn next to a probabilistic view of classification and show how models with linear decision boundaries arise from simple assumptions about the distribution of the data. Here we adopt a generative approach in which we model the posterior probabilities $P(y|\mathbf{x})$. Again, consider a two-case classification problem where we aim to assign the input vector $\mathbf{x}$ to either class y_1 or class y_2. This is achieved by computing the posterior probabilities $P(y_1|\mathbf{x})$ and $P(y_2|\mathbf{x})$ using Bayes theorem, and comparing their quantities. One common criterion for classification is given by

$$l = \ln(P(y_1|\mathbf{x})) - \ln(P(y_2|\mathbf{x})),$$

where $\ln(P(y|\mathbf{x}))$ is also known as log-likelihood. It measures given $\mathbf{x}$ present, how much likely the vector will belong to a certain class. Log-likelihood is widely used. One possible reason is that many ML generative models have Gaussian assumptions, and using the log operation will facilitate the computation to a large extent.

Constructing a logistic model assumes that the differences of the log-likelihood of the two classes can be modeled with a linear function

$$\ln(P(y_1|\mathbf{x})) - \ln(P(y_2|\mathbf{x})) = \mathbf{w}^T\mathbf{x} + b = f(\mathbf{x}).$$

[1] https://www.sciencedirect.com/topics/computer-science/logistic-regression

Based on this assumption, we have

$$\frac{P(y_1|\mathbf{x})}{P(y_2|\mathbf{x})} = e^{f(\mathbf{x})}.$$

Meanwhile, due to the two-case classification problem, we also have

$$P(y_1|\mathbf{x}) + P(y_2|\mathbf{x}) = 1.$$

From all the formulas above we can obtain (the derivation steps are omitted, but you can also derive on your own)

$$P(y_1|\mathbf{x}) = \frac{1}{1+e^{-f(\mathbf{x})}} = \sigma(f(\mathbf{x})), \tag{3}$$

where $\sigma(a) = 1/(1+e^{-a})$ is known as the logistic sigmoid function which is plotted in Figure 3. The term "sigmoid" means S-shaped. This type of function is sometimes also called a "squashing function" because it maps the whole real axis into a finite interval. The logistic sigmoid plays an important role in many classification algorithms.

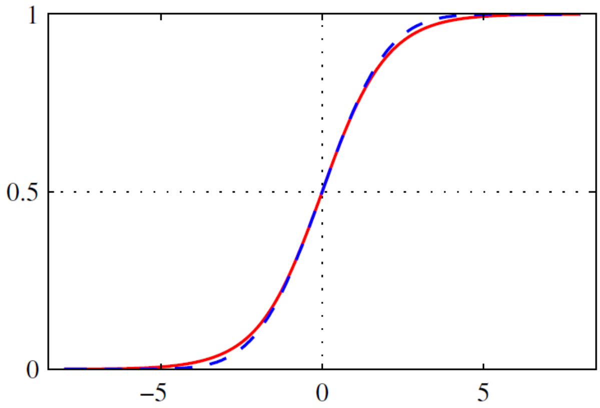

Figure 3: The logistic sigmoid function.

2.2 Maximizing the Likelihood

We use maximum likelihood to determine the parameters involved in a logistic model, which is extensively used in many generative ML models. Given a set of input vectors together with the class labels

$$(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), (\mathbf{x}_3, y_3), ..., (\mathbf{x}_n, y_n),$$

where $y_i (i \in n)$ is the class label of the i-th sample $\mathbf{x}_i$, i.e.,

$$y_i = \begin{cases} +1 & \text{if } y_i \text{ is in class 1} \\ -1 & \text{if } y_i \text{ is in class 2} \end{cases}.$$

Assume we draw the sample vectors $\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n$ independently and identically from the same input data distribution, according to the independent rule introduced in the probability basic, we have

$$P(\mathbf{y}|\mathbf{x}) = P(y_1|\mathbf{x}_1)P(y_2|\mathbf{x}_2)...P(y_n|\mathbf{x}_n).$$

Applying log operation on both ends of the equation we have:

$$\ln P(\mathbf{y}|\mathbf{x}) = \sum_{i=1}^{n} \ln P(y_i|\mathbf{x}_i)),$$

where $P(y_i|\mathbf{x}_i)$ can be modeled using the logistic sigmoid function

- if $y_i = +1$, $P(y_i|\mathbf{x}_i)) = \frac{1}{1+e^{-f(\mathbf{x}_i)}}$
- if $y_i = -1$, $P(y_i|\mathbf{x}_i)) = 1 - \frac{1}{1+e^{-f(\mathbf{x}_i)}} = \frac{1}{1+e^{f(\mathbf{x}_i)}}$

Therefore, we have:

$$\ln P(\mathbf{y}|\mathbf{x}) = \sum_{i=1}^{n} \ln \frac{1}{1+e^{-y_i f(\mathbf{x}_i)}} = -\sum_{i=1}^{n} \ln(1+e^{-y_i f(\mathbf{x}_i)})$$

The problem becomes minimizing the log-likelihood term: $\sum_{i=1}^{n} \ln(1+e^{-y_i f(\mathbf{x}_i)})$.
It is worth noting that maximum likelihood can prohibit severe over-fitting for data sets that are linearly separable. However, the maximum likelihood doesn't provide a closed-form solution as least squares. Moreover, it provides no way to favor one such solution over another, and which solution is found in practice will depend on the choice of optimization algorithm and on the parameter initialization.