Chapter 6

Linear Classification

In a classification problem, we have an input vector $\boldsymbol{x}$ together with a corresponding label y. Based on a data set $\{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \ldots, (\boldsymbol{x}_n, y_n)\}$, our goal is to predict y given a new value for $\boldsymbol{x}$. If y is a continuous variable the problem is that of regression, whereas in classification problems, y will represent a set of discrete class labels. For example, we may wish to classify images of handwritten digits. In this case, $\boldsymbol{x}$ is a vector providing the values of pixels of the image and $y \in \{0, 1, \ldots, 9\}$ is the label indicating what digit the image represents.

6.1 Overview of probabilistic models

The probabilistic approach to classification requires us to learn the distribution $p(y|\boldsymbol{x})$, which for any given $\boldsymbol{x}$ provides the probability of belonging to different classes. We can identify the class for a given $\boldsymbol{x}$ as the class that has the maximum probability,

$$\hat{y}(\boldsymbol{x}) = \arg\max_j p(y = j|\boldsymbol{x}).$$

This choice *minimizes the probability of predicting the wrong class*

$$\mathcal{L} = \Pr(\hat{y}(\boldsymbol{x}) \neq y) = \mathbb{E}[I(y \neq \hat{y}(\boldsymbol{x}))].$$

To find the distribution $p(y|\boldsymbol{x})$, our first step is developing a model that relates $\boldsymbol{x}$ and y. There are two possible approaches.

We may develop a **generative model**, i.e., a model that is capable of *generating* data and also helping us predict y for a given $\boldsymbol{x}$. A generative model has two components, both of which must be learned from data:

- Prior class probabilities: $p(y)$
- Class-conditional probabilities: $p(\boldsymbol{x}|y)$

From these, using Bayes' theorem we can find $p(y|\boldsymbol{x})$ as

$$p(y|\boldsymbol{x}) = \frac{p(y)p(\boldsymbol{x}|y)}{p(\boldsymbol{x})} \propto p(y)p(\boldsymbol{x}|y).$$

We can often estimate $p(y = j)$ simply by computing the fraction of class j in our training data. For $p(\boldsymbol{x}|y)$, a common approach is to represent it parametrically and then learn the parameters from the data. For example, we may assume that given class j, $\boldsymbol{x}$ is distributed normally with mean $\boldsymbol{\mu}_j$ and covariance matrix K_j and then learn these parameters from data.

Alternatively, we can develop a **discriminative model**. In this case, we directly model $p(y|\boldsymbol{x})$ since this is the distribution that we need to decide which class $\boldsymbol{x}$ belongs to.

6.2 Generative Probabilistic Models

6.2.1 Gaussian Class-Conditionals

Let us denote

$$p(y = j) = \pi_j.$$

We further assume $p(\boldsymbol{x}|y = j)$ is Gaussian with mean $\boldsymbol{\mu}_j$ and covariance matrix Σ_j,

$$p(\boldsymbol{x}|y = j) = \frac{1}{\sqrt{2\pi|\Sigma_j|}} \exp\left(-\frac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu}_j)^T \Sigma_j^{-1} (\boldsymbol{x} - \boldsymbol{\mu}_j)\right)$$

For our purpose, it suffices to consider $\ln p(\boldsymbol{x}|y = j)$, which, after dropping the constant terms, becomes

$$\ln p(\boldsymbol{x}|y = j) \doteq -\frac{1}{2}\ln|\Sigma_j| - \frac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu}_j)^T \Sigma_j^{-1} (\boldsymbol{x} - \boldsymbol{\mu}_j).$$

From these, we can find $\ln p(\boldsymbol{x}|y = j)$ and then decide $\hat{y}(\boldsymbol{x})$ as

$$\hat{y}(\boldsymbol{x}) = \arg\max_j \ln p(y = j|\boldsymbol{x}).$$

More specifically,

$$\begin{aligned} \ln p(y = j|\boldsymbol{x}) &\doteq \ln \pi_j - \frac{1}{2}\ln|\Sigma_j| - \frac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu}_j)^T \Sigma_j^{-1} (\boldsymbol{x} - \boldsymbol{\mu}_j) \\ &= \boldsymbol{x}^T \left(-\frac{1}{2}\Sigma_j^{-1}\right)\boldsymbol{x} + \left(\boldsymbol{\mu}_j^T \Sigma_j^{-1}\right)\boldsymbol{x} + \left(-\frac{1}{2}\boldsymbol{\mu}_j^T \Sigma_j^{-1}\boldsymbol{\mu}_j - \frac{1}{2}\ln|\Sigma_j| + \ln \pi_j\right) \\ &= \boldsymbol{x}^T A_j \boldsymbol{x} + \boldsymbol{\beta}_j^T \boldsymbol{x} + \gamma_j, \end{aligned} \tag{6.1}$$

For an appropriately defined symmetric matrix A_j, vector $\boldsymbol{\beta}_j$, and scalar γ_j. In the next two sections, we will consider two cases based on whether the covariance matrix depends on the class.

6.2.2 Linear Discriminant Analysis

First, let us suppose all classes have the same covariance matrix $\Sigma_j = \Sigma$. Then, the terms $\boldsymbol{x}^T\left(-\frac{1}{2}\Sigma^{-1}\right)\boldsymbol{x}$ and $-\frac{1}{2}\ln|\Sigma_j|$ in (6.1) become independent of the class and we thus have

$$\ln p(y = j|\boldsymbol{x}) \doteq \boldsymbol{\beta}_j^T \boldsymbol{x} + \gamma_j, \tag{6.2}$$

where

$$\boldsymbol{\beta}_j^T = \boldsymbol{\mu}_j^T \Sigma^{-1}, \qquad \gamma_j = -\frac{1}{2}\boldsymbol{\mu}_j^T \Sigma^{-1}\boldsymbol{\mu}_j + \ln \pi_j,$$

Suppose we have only two classes, $y = 0$ and $y = 1$, with $p(y = 1) = \pi = 1 - p(y = 0)$. Equivalent to finding $\arg\max_j \ln p(y = j|\boldsymbol{x})$ for each $\boldsymbol{x}$, we can divide the space into two regions,

$$\ln p(y = 1|\boldsymbol{x}) \underset{\hat{y}=0}{\overset{\hat{y}=1}{\gtrless}} \ln p(y = 0|\boldsymbol{x}).$$

What is the decision boundary between them? We can find it by solving $\ln p(y = 1|\boldsymbol{x}) = \ln p(y = 0|\boldsymbol{x})$,

$$\boldsymbol{\beta}_1^T\boldsymbol{x} + \gamma_1 = \boldsymbol{\beta}_0^T\boldsymbol{x} + \gamma_0 \iff (\boldsymbol{\beta}_1 - \boldsymbol{\beta}_0)^T\boldsymbol{x} + \gamma_1 - \gamma_0 = 0 \iff \boldsymbol{\beta}^T\boldsymbol{x} + \gamma = 0,$$

where

$$\boldsymbol{\beta}^T = (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)^T\Sigma^{-1}, \qquad \gamma = \ln\frac{\pi}{1-\pi} - \frac{1}{2}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)^T\Sigma^{-1}(\boldsymbol{\mu}_1 + \boldsymbol{\mu}_0). \tag{6.3}$$

Hence, the decision boundary is the hyperplane $\boldsymbol{\beta}^T\boldsymbol{x} + \gamma = 0$. On one side of this plane, we predict class 1 (we let $\hat{y}(\boldsymbol{x}) = 1$) and on the other side, we declare class 0:

$$\hat{y}(\boldsymbol{x}) = \begin{cases} 1, & \boldsymbol{\beta}^T\boldsymbol{x} + \gamma > 0 \\ 0, & \boldsymbol{\beta}^T\boldsymbol{x} + \gamma < 0 \end{cases}$$

Since the boundary is linear (i.e., a hyperplane such as a line, 2-D plane, etc), this method is called **Linear Discriminant Analysis** (LDA).

As a special case, consider, $\pi = \frac{1}{2}, \Sigma = I$. The the boundary becomes

$$(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)^T\left(\boldsymbol{x} - \frac{\boldsymbol{\mu}_1 + \boldsymbol{\mu}_0}{2}\right) = 0,$$

which implies that the boundary is the plane that passes through the midpoint of the line connecting $\boldsymbol{\mu}_1$ and $\boldsymbol{\mu}_0$ and is perpendicular to it.

What about the probability $p(y|\boldsymbol{x})$ of each class for a given $\boldsymbol{x}$, which can tell us about the certainty of belonging to each class? From (6.2), we have $p(y = j|\boldsymbol{x}) \propto e^{\boldsymbol{\beta}_j^T\boldsymbol{x}+\gamma_j}$ and so for two classes

$$p(y = 1|\boldsymbol{x}) = \frac{e^{\boldsymbol{\beta}_1^T\boldsymbol{x}+\gamma_1}}{e^{\boldsymbol{\beta}_1^T\boldsymbol{x}+\gamma_1} + e^{\boldsymbol{\beta}_0^T\boldsymbol{x}+\gamma_0}} = \frac{1}{1 + e^{-(\boldsymbol{\beta}^T\boldsymbol{x}+\gamma)}} = \sigma(\boldsymbol{\beta}^T\boldsymbol{x} + \gamma),$$

where $\boldsymbol{\beta}$ and γ are given in (6.3), and $\sigma(u) = \frac{1}{1+e^{-u}}$ is the sigmoid (logistic) function.

If there are $c > 2$ classes, decision hyperplanes between pairs of classes will divide the space into c regions. And the conditional probability of class j is given by

$$p(y = j|\boldsymbol{x}) = \frac{e^{\boldsymbol{\beta}_j^T\boldsymbol{x}+\gamma_j}}{\sum_{k=1}^{c} e^{\boldsymbol{\beta}_k^T\boldsymbol{x}+\gamma_k}} = \sigma_j(\boldsymbol{\beta}_1^T\boldsymbol{x} + \gamma_1, \ldots, \boldsymbol{\beta}_c^T\boldsymbol{x} + \gamma_c),$$

where $\sigma_j(\boldsymbol{v}) = \frac{e^{v_j}}{\sum_k e^{v_k}}$ is the softmax function.

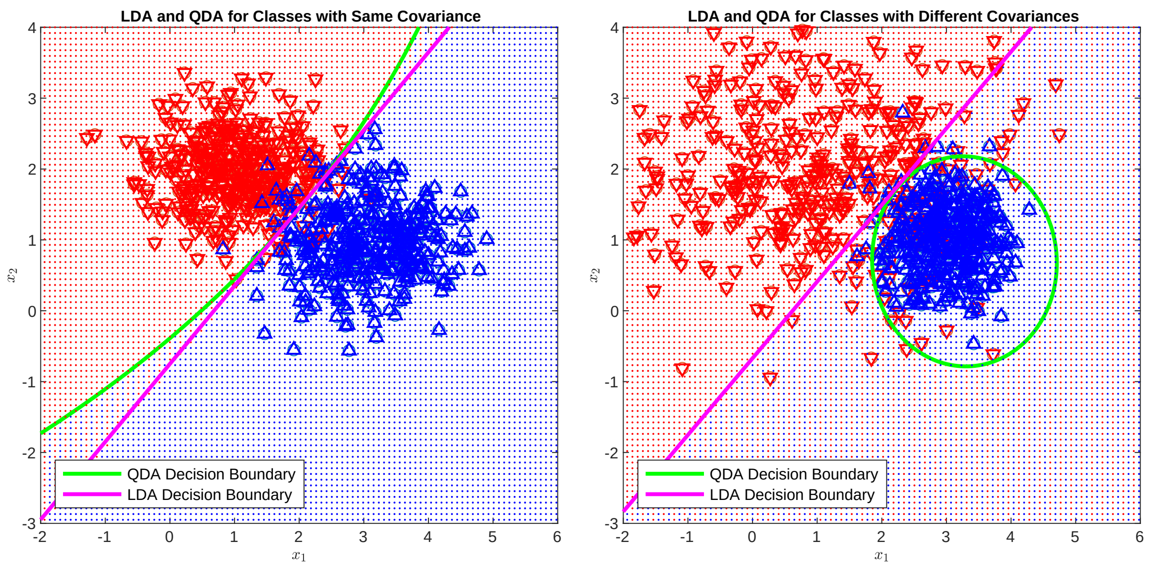


Figure 6.1: LDA vs QDA for when $\Sigma_1 = \Sigma_2$ (left) and when $\Sigma_1 \neq \Sigma_2$ (right).

6.2.3 Quadratic Discriminant Analysis

Let us now assume that each class has a different covariance matrix Σ_j. To decide between two classes, say $y = 0$ and $y = 1$, the decision boundary is given by $\ln p(y = 1|\boldsymbol{x}) = \ln p(y = 0|\boldsymbol{x})$. This will lead to a quadratic equation of the form $\boldsymbol{x}^T A\boldsymbol{x} + \boldsymbol{\beta}^T\boldsymbol{x} + \gamma = 0$, which leads to a nonlinear decision boundary. As a result, this method is called **Quadratic Discriminant Analysis** (QDA).

Figure 6.1 demonstrates LDA and QDA when $\Sigma_1 = \Sigma_2$ (left) and when $\Sigma_1 \neq \Sigma_2$ (right). Here the boundaries are learned from data (see Section 6.2.2). On the left the data is generated by distributions that match the assumption made by LDA and so LDA and QDA perform similarly. However, on the right the covariances are different and so, as expected, QDA performs better. Note however that we could augment our feature vectors as $(x_1, x_2, x_1x_2, x_1^2, x_2^2)$ instead of just (x_1, x_2) and then apply LDA, allowing a decision boundary that is not linear in x_1, x_2. In that case, the performance of LDA would generally be similar to that of QDA (Hastie et al,. Elements of Statistical Learning).

6.2.4 Maximum Likelihood Solution to LDA

Once we specified a parametric form for the class-conditional densities $p(\boldsymbol{x}|y = j)$, we can determine the values of the parameters, together with the prior class probabilities $p(y = j)$, using maximum likelihood.

Data: Our data set comprises of observations of $\boldsymbol{x}$ along with their corresponding class labels. Let the n independent samples be denoted by $\mathcal{D} = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \ldots, (\boldsymbol{x}_n, y_n)\}$, where $\boldsymbol{x}_i \in \mathbb{R}^m$ and $y_i \in \{0, 1\}$ for all i.

Model:

$$p(y=j) = \begin{cases} \pi, & j=1 \\ 1-\pi, & j=0 \end{cases}$$

$$p(\boldsymbol{x}|y=0) \sim \mathcal{N}(\boldsymbol{\mu}_0, \Sigma),$$
$$p(\boldsymbol{x}|y=1) \sim \mathcal{N}(\boldsymbol{\mu}_1, \Sigma),$$

for some $\pi, \boldsymbol{\mu}_0, \boldsymbol{\mu}_1$ and diagonal matrix $\Sigma = \text{diag}(\sigma_1^2, \sigma_2^2, \ldots, \sigma_m^2)$. Note that since we assume both classes have the same covariance matrix, the decision boundary will be linear (i.e., LDA). Also, we have assumed given the class, features are independent (since Σ is diagonal); this is called the **Naive Bayes** model.

Likelihood:

$$\begin{aligned} p(\mathcal{D}|\pi, \boldsymbol{\mu}_0, \boldsymbol{\mu}_1, \Sigma) &= \prod_{i=1}^n p(y_i)p(\boldsymbol{x}_i|y_i) = \left(\prod_{i:y_i=0} p(y_i)p(\boldsymbol{x}_i|y_i)\right)\left(\prod_{i:y_i=1} p(y_i)p(\boldsymbol{x}_i|y_i)\right) \\ &= \prod_{i:y_i=0}\left((1-\pi)\prod_{j=1}^m \frac{1}{\sqrt{2\pi\sigma_j^2}} \exp(-(x_{i,j}-\mu_{0,j})^2/2\sigma_j^2)\right) \times \\ &\quad \prod_{i:y_i=1}\left(\pi\prod_{j=1}^m \frac{1}{\sqrt{2\pi\sigma_j^2}} \exp(-(x_{i,j}-\mu_{1,j})^2/2\sigma_j^2)\right), \end{aligned}$$

where $x_{i,j}$ is the j^{th} component of $\boldsymbol{x}_i$ and $\mu_{0,j}, \mu_{1,j}$ are the j^{th} components of $\boldsymbol{\mu}_0$ and $\boldsymbol{\mu}_1$, respectively. The maximum likelihood solution is (exercise)

$$\begin{aligned} \hat{\pi}_{ML} &= \frac{\sum y_i}{n}, \\ (\hat{\mu}_{0,j})_{ML} &= \frac{\sum_{i:y_i=0} x_{i,j}}{\sum(1-y_i)}, \quad (\hat{\mu}_{1,j})_{ML} = \frac{\sum_{i:y_i=1} x_{i,j}}{\sum y_i}, \\ (\hat{\sigma_j^2})_{ML} &= \frac{1}{n}\left(\sum_{i:y_i=0}(x_{i,j}-\hat{\mu}_{0,j})^2 + \sum_{i:y_i=1}(x_{i,j}-\hat{\mu}_{1,j})^2\right) \end{aligned}$$

6.2.5 Generative Model for Discrete Features **

If a features is categorical, for example, type of a vehicle or genre of a movie, we can encode them as binary vectors. For example, if there are three categories, with the vector $(1,0,0)$ we can indicate belonging to the first category. This is called *one-hot* or *dummy encoding.* In this case, our data is still denoted by $(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_n, y_n)$, where each $\boldsymbol{x}_i$ is composed of vectors, that is[1]

$$\boldsymbol{x}_i = (\boldsymbol{x}_{i1}, \ldots, \boldsymbol{x}_{im}),$$

[1]All vectors in this section are column vectors and all concatenations are also along the vertical dimension. However, for simplicity of notation, we write $\boldsymbol{x}_i = (\boldsymbol{x}_{i1}, \ldots, \boldsymbol{x}_{im})$ instead of $\boldsymbol{x}_i^T = (\boldsymbol{x}_{i1}^T, \ldots, \boldsymbol{x}_{im}^T)^T$

and each $\boldsymbol{x}_{ij} = (x_{ij1}, \ldots, x_{ijl}, \ldots)$ is a binary vector of finite length which represents a one-hot encoding of a feature.

Example 6.1 (One-hot encoding)**.** Suppose $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_m$ provide information about a set of movies, where $\boldsymbol{x}_1 = (\boldsymbol{x}_{11}, \ldots, \boldsymbol{x}_{1m}), \boldsymbol{x}_2 = (\boldsymbol{x}_{21}, \ldots, \boldsymbol{x}_{2m}), \ldots$, with $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_m$ denoting in order the genre of the movie, the director of the movie, etc. Explicitly, for the genre, if we order them as (comedy, horror, drama, scifi, action), and for five directors A, B, C, D, E, order them as (A, B, C, D, E), then $\boldsymbol{x}_{11} = (0, 0, 1, 0, 0), \boldsymbol{x}_{12} = (0, 0, 0, 1, 0)$ means movie 1 is a drama directed by director D, and $\boldsymbol{x}_{21} = (1, 0, 0, 0, 0), \boldsymbol{x}_{22} = (1, 0, 0, 0, 0)$ means that movie 2 is a comedy directed by director A. △

Model: We model this classification problem in the following way:

$$p(x_{ijl} = 1 | y_i = k) = \eta_{kjl}, \qquad \sum_l \eta_{kjl} = 1,$$

and all x_{ijl} are independent from one another. For two vectors $\boldsymbol{a}, \boldsymbol{b}$ with the same length, we define $\boldsymbol{a}^{\boldsymbol{b}} = \prod_{i=1}^{|\boldsymbol{a}|} a_i^{b_i}$. Let $\boldsymbol{\eta}_{kj} = (\eta_{kj1}, \ldots)$. We have

$$p(\boldsymbol{x}_{ij} | y_i = k) = \boldsymbol{\eta}_{kj}^{\boldsymbol{x}_{ij}},$$

and then

$$p(\boldsymbol{x}_i | y_i = k) = \prod_{j=1}^m p(\boldsymbol{x}_{ij} | y_i = k) = \prod_{j=1}^m \boldsymbol{\eta}_{kj}^{\boldsymbol{x}_{ij}} = \boldsymbol{\eta}_k^{\boldsymbol{x}_i},$$

where $\boldsymbol{\eta}_k = (\boldsymbol{\eta}_{k1}, \ldots, \boldsymbol{\eta}_{km})$. It follows that

$$p(y_i = k | \boldsymbol{x}_i) \propto p(\boldsymbol{x}_i | y_i = k) p(y_i = k) = \pi_k \boldsymbol{\eta}_k^{\boldsymbol{x}_i} \propto \exp(\ln \pi_k + \boldsymbol{x}_i^T \ln \boldsymbol{\eta}_k).$$

For a new data point $(\boldsymbol{x}, y)$, we similarly have

$$\ln p(y = k | \boldsymbol{x}) \doteq \boldsymbol{\beta}_k^T \boldsymbol{x} + \gamma_k, \tag{6.4}$$

where $\boldsymbol{\beta}_k = \ln \boldsymbol{\eta}_k$ and $\gamma_k = \ln \pi_k$. The log-probabilities are again linear in $\boldsymbol{x}$, an fact that as we will see contributes to the motivation for logistic regression.

6.2.6 Class-conditionals from the exponential family

The exponential family of distributions includes common distributions such as Gaussian, exponential, gamma, beta, Dirichlet, Bernoulli, Poisson, and geometric. Distributions from this family have the following form

$$p(\boldsymbol{x} | \boldsymbol{\theta}) = \exp[\boldsymbol{b}(\boldsymbol{\theta})^T \boldsymbol{a}(\boldsymbol{x}) + f(\boldsymbol{x}) + g(\boldsymbol{\theta})].$$

Let us consider the case in which $\boldsymbol{a}(\boldsymbol{x}) = \boldsymbol{x}$, and parameters are functions of class y. So instead of $\boldsymbol{\theta}$ we write $\boldsymbol{\theta}_j$, when considering the jth class. Then the class-conditional distribution will become

$$p(\boldsymbol{x} | y = j) = \exp[\boldsymbol{b}(\boldsymbol{\theta}_j)^T \boldsymbol{x} + f(\boldsymbol{x}) + g(\boldsymbol{\theta}_j)].$$

Furthermore, let $p(y = j) = \pi_j$. Given $\boldsymbol{x}$, the log-probability of each class is given as

$$\ln p(y = j | \boldsymbol{x}) \doteq \ln \pi_j + \ln p(\boldsymbol{x} | y = j) \doteq \ln \pi_j + \boldsymbol{b}(\boldsymbol{\theta}_j)^T \boldsymbol{x} + g(\boldsymbol{\theta}_j) \doteq \boldsymbol{\beta}_j^T \boldsymbol{x} + \gamma_j, \tag{6.5}$$

where $\boldsymbol{\beta}_j = \boldsymbol{b}(\boldsymbol{\theta}_j)$ and $\gamma_j = \ln \pi_j + g(\boldsymbol{\theta}_j)$. So for a large class of class-conditional probabilities, the log-probabilities of classes given the feature vector $\boldsymbol{x}$ is linear in $\boldsymbol{x}$.

6.3 Discriminative Models and Logistic Regression

In the discriminative approach, we model $p(y = j|\boldsymbol{x})$ directly. But what is a good model? As we have seen in (6.2), (6.4) and (6.5), in many generative cases, the log-probabilities of classes given data is linear in $\boldsymbol{x}$,

$$\ln p(y = j|\boldsymbol{x}) \doteq \boldsymbol{\beta}_j^T \boldsymbol{x} + \gamma_j.$$

And based on Section 6.2.2, this form leads to linear class boundaries and posterior class probabilities of the logistic form for two classes,

$$p(y = 1|\boldsymbol{x}) = \frac{1}{1 + e^{-(\boldsymbol{\beta}^T \boldsymbol{x} + \gamma)}}, \qquad p(y = 0|\boldsymbol{x}) = \frac{e^{-(\boldsymbol{\beta}^T \boldsymbol{x} + \gamma)}}{1 + e^{-(\boldsymbol{\beta}^T \boldsymbol{x} + \gamma)}},$$

where $\boldsymbol{\beta} = \boldsymbol{\beta}_1 - \boldsymbol{\beta}_2$ and $\gamma = \gamma_1 - \gamma_2$.

Limiting ourselves to two classes, this observation raises the following question: 'Why not assume from the beginning that $p(y|\boldsymbol{x})$ is of the logistic form and learn this distribution instead of learning first $p(\boldsymbol{x}|y)$ and $p(y)$?' Doing so leads to a *discriminative model* resulting in *logistic regression.*

Let $h(\boldsymbol{x}) = p(y = 1|\boldsymbol{x})$ and assume that the data consists of n iid samples, $\mathcal{D} = \{(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_n, y_n)\}$. We have

$$p(\mathcal{D}; \boldsymbol{\beta}, \gamma) = \prod_{i=1}^{n} h(\boldsymbol{x}_i)^{y_i} (1 - h(\boldsymbol{x}_i))^{1-y_i}$$

and the negative log-likelihood loss is given by

$$-\mathcal{L}(\boldsymbol{\beta}, \gamma) = \sum_{i=1}^{n} \left(y_i \ln \frac{1}{h(\boldsymbol{x}_i)} + (1 - y_i) \ln \frac{1}{1 - h(\boldsymbol{x}_i)} \right). \tag{6.6}$$

We can use gradient descent to minimize this loss (maximize the likelihood). For simplicity, let $\boldsymbol{\theta} = \begin{pmatrix} \boldsymbol{\beta} \\ \gamma \end{pmatrix}$, $\tilde{\boldsymbol{x}} = \begin{pmatrix} \boldsymbol{x} \\ 1 \end{pmatrix}$, and $h_{\boldsymbol{\theta}} = p(y = 1|\boldsymbol{x}) = \frac{1}{1+e^{-\boldsymbol{\theta}^T \tilde{\boldsymbol{x}}}}$. Then

$$\boldsymbol{\theta}^{(t+1)} = \boldsymbol{\theta}^{(t)} + \rho_t \nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}),$$

where

$$\nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}) = \sum_{i=1}^{n} (y_i - h_{\boldsymbol{\theta}}(\tilde{\boldsymbol{x}}_i)) \tilde{\boldsymbol{x}}_i.$$

When we find $\boldsymbol{\theta}$ and thus $\boldsymbol{\beta}, \gamma$ we have the decision boundary as $\boldsymbol{\beta}^T \boldsymbol{x} + \gamma = 0$. Points $\boldsymbol{x}$ for which $\boldsymbol{\beta}^T \boldsymbol{x} + \gamma > 0$ are classified as class $y = 1$.

6.4 Risk minimization and loss functions for classification

An alternative approach to generative models and logistic regression we discussed before is directly minimizing an empirical loss,

$$\frac{1}{n} \sum_{i=1}^{n} L(y_i, \boldsymbol{x}_i, \hat{y}(\boldsymbol{x}_i)),$$

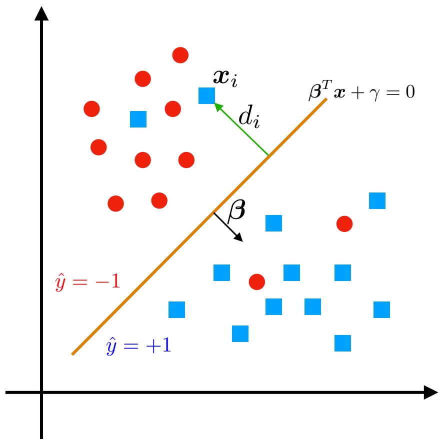


Figure 6.2: A linear classifier defined by the vector $\boldsymbol{\beta}$ and scalar γ. Squares represents points with $y=+1$ and circles $y=-1$. For a point $\boldsymbol{x}_i$, many loss functions can be viewed as a function of the signed distance d_i of $\boldsymbol{x}_i$ to the decision hyperplane.

where $\hat{y}(\boldsymbol{x})$ is the predictor of the class for input vector $\boldsymbol{x}$. For ease of exposition, instead of assuming $y \in \{0,1\}$, we assume $y \in \{-1,1\}$.

Our attention will be limited to linear classifiers, determined by a vector $\boldsymbol{\beta}$ and a constant γ, which define the hyperplane $\boldsymbol{\beta}^T\boldsymbol{x}+\gamma=0$. On one side of the hyperplane, we decide class 1 and the other side class -1,

$$\hat{y}(\boldsymbol{x}) = \operatorname{sign}(\boldsymbol{\beta}^T\boldsymbol{x}+\gamma) = \begin{cases} 1, & \text{if } \boldsymbol{\beta}^T\boldsymbol{x}+\gamma > 0, \\ -1, & \text{if } \boldsymbol{\beta}^T\boldsymbol{x}+\gamma < 0, \end{cases}$$

where the dependence of $\hat{y}$ on $\boldsymbol{\beta}, \gamma$ is implicit.

One such linear classifier is shown in Figure 6.2. Below, we will use the fact that for any point $\boldsymbol{x}_i$ with label y_i and prediction $\hat{y}(\boldsymbol{x}_i)$, the loss contributed by it can often be viewed as a function of its signed distance d_i to the decision hyperplane. Without loss of generality, assume that $y_i = 1$ and $\boldsymbol{x}_i = \boldsymbol{x}_0 + d_i\boldsymbol{\beta}/\|\boldsymbol{\beta}\|$ for some $\boldsymbol{x}_0$ on the decision boundary. If d_i is positive, then this point is classified correctly, since $\boldsymbol{\beta}^T\boldsymbol{x}_i+\gamma > 0$. The distance between $\boldsymbol{x}_i$ and the decision boundary equals $|d_i|$.

6.4.1 Zero-one loss

The most natural loss function for classification is the **0-1 loss**,

$$L_{01}(y,\hat{y}(\boldsymbol{x})) = \left\{\begin{matrix} 1, & \text{if } y \neq \hat{y}(\boldsymbol{x}) \\ 0, & \text{if } y = \hat{y}(\boldsymbol{x}) \end{matrix}\right\} = \begin{cases} 1, & \text{if } y(\boldsymbol{\beta}^T\boldsymbol{x}+\gamma) < 0, \\ 0, & \text{if } y(\boldsymbol{\beta}^T\boldsymbol{x}+\gamma) > 0. \end{cases}$$

Figure 6.3 shows the 0-1 loss for a point in the positive class. Note that how far the point is from the boundary does not affect how much it contributes to the loss.

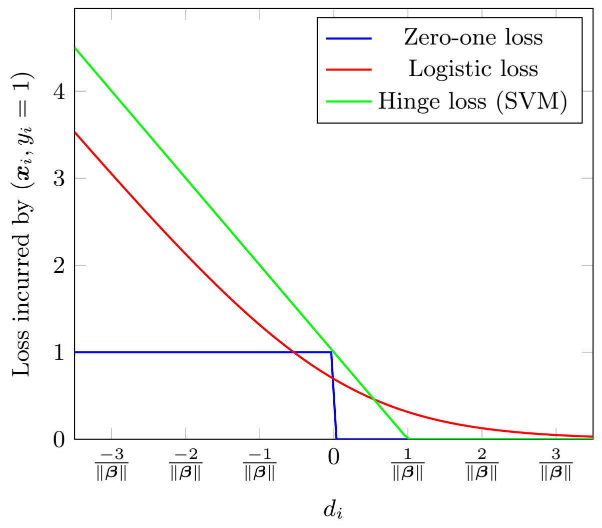


Figure 6.3: Loss functions for a data point $(\boldsymbol{x}_i, y_i = 1)$ as a function of the distance of $\boldsymbol{x}_i$ from the boundary (in terms of the length of $\boldsymbol{\beta}$).

Unfortunately, minimizing this loss function is computationally difficult (NP-hard) [1]. So in practice, we use differentiable loss-functions for which efficient algorithms exist. Here we will consider two such loss functions. First, we view logistic regression in terms of empirical risk minimization and then we will consider the hinge loss in the context of support vector machine (SVM) classifiers.

6.4.2 Logistic regression

Let us re-examine the logistic regression loss function (6.6). The loss incurred by a data point $\boldsymbol{x}_i$ at signed distance d_i from the decision hyperplane (i.e., $\boldsymbol{x}_i = \boldsymbol{x}_0 + d_i\boldsymbol{\beta}/\|\boldsymbol{\beta}\|$) is

$$\ln(1 + e^{-(\boldsymbol{\beta}^T\boldsymbol{x}_i+\gamma)}) = \ln(1 + e^{-d_i\|\boldsymbol{\beta}\|}).$$

The figure below shows this loss: For $d_i < 0$, where the input is misclassified, the loss is larger, and it increases as the point gets farther from the boundary. But even for points that are classified correctly, there is a loss, which decreases as we get farther from the boundary.

6.4.3 Hinge loss (SVM)

Hinge loss results from penalizing misclassified points as well as those that are classified correctly, but are within a certain margin close to the decision boundary. The expression for hinge loss is

$$\max(0, 1 - y_i(\boldsymbol{\beta}^T\boldsymbol{x}_i + \gamma)).$$

Letting $y_i = 1$ and $\boldsymbol{x}_i = \boldsymbol{x}_0 + d_i\boldsymbol{\beta}/\|\boldsymbol{\beta}\|$ as before, results in

$$\max(0, 1 - d_i\|\boldsymbol{\beta}\|).$$

which is shown in Figure 6.3. So the penalty for misclassified points is larger the farther away they are from the boundary. In addition, even points classified correctly are penalized if they are within a **margin** of width $1/\|\boldsymbol{\beta}\|$ of the decision boundary.

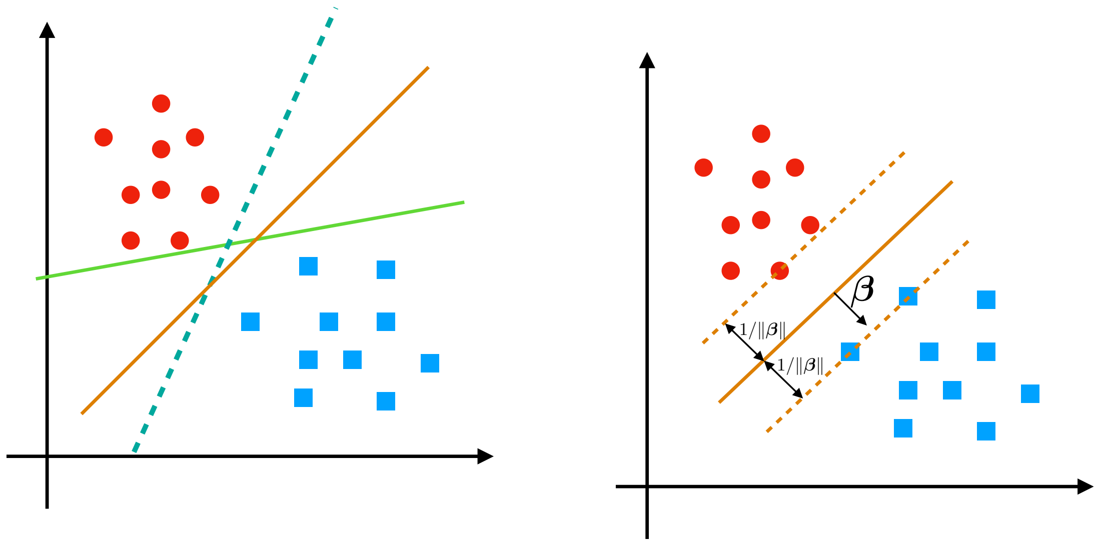
In addition to penalizing points within the margin, we would like to ensure that the margin is not very small. This can be done by ensuring $1/\|\boldsymbol{\beta}\|$ is large or equivalently $\|\boldsymbol{\beta}\|^2$ is small. Both of these goals can be achieved with the loss

$$\frac{1}{n}\sum_{i=1}^{n}\max(0, 1 - y_i(\boldsymbol{\beta}^T\boldsymbol{x}_i + \gamma)) + \lambda\|\boldsymbol{\beta}\|^2, \tag{6.7}$$

where λ is a constant that balances the two components of the loss. This results in the so called **support vector machine classifier** (SVM).

SVM as maximum-margin classifier. Let us consider the case in which the data is linearly separable, i.e., there exists a hyperplane that correctly classifies all data points. In such a case, as shown in Figure 6.4a, there are typically an infinite number of separating hyperplanes. This leads to the question of which one should be chosen. The SVM loss given in (6.7) provides a solution. Assume λ is positive but very small. So we are primarily concerned about the first term in the loss, i.e., the hinge loss. Between choices that incur the same hinge loss, we must pick the one that maximizes the margin, i.e., minimizes $\|\boldsymbol{\beta}\|^2$. Thus:

- We can make the hinge loss term zero by choosing any separating hyperplane that makes no mistakes and choosing any margin (length of $\|\boldsymbol{\beta}\|$) that is small enough such that there no points within the margin.
- Now the second term ensures that among the hyperplanes that perfectly separate the data, we should pick the one that has the maximum margin, as shown in Fig. 6.4b.



(a) For two linearly separable classes, there are an infinite number of classifiers that perfectly separate the training data. Which one should we pick?

(b) The maximum-margin classifier is the classifier that maximizes the distance between the decision boundary and the closest points to it.

Figure 6.4: SVM for separable data

Bibliography

[1] T. T. Nguyen and S. Sanner, “Algorithms for direct 0-1 loss optimization in binary classification,” in *Proceedings of the 30th International Conference on International Conference on Machine Learning - Volume 28*, ICML’13, (Atlanta, GA, USA), pp. III–1085–III–1093, JMLR.org, June 2013.