Chatper 3: Linear Models for Regression and Classification

ifding

Abstract

Linear Basis Function Models, The Bias-Variance Decomposition, Discriminant Functions, Probabilistic Generative Models

The goal of regression is to predict the value of one or more continuous *target* variables t given the value of a D-dimensional vector $\mathbf{x}$ of *input* variables. From a probabilistic perspective, we aim to model the predictive distribution $p(t|\mathbf{x})$ because this expresses our uncertainty about the value of t for each value of $\mathbf{x}$.

The goal in classification is to take an input vector $\boldsymbol{x}$ ans to assign it to one of K discrete classes $\mathcal{C}_k$ where $k = 1, \ldots, K$. The input space is divided into *decision regions* whose boundaries are called *decision boundaries* or *decision surfaces*. For the targe variable $\boldsymbol{t}$, it is convenient to use a 1-of-K coding scheme.

1. Linear Basis Function Models

Consider linear combinations of fixed nonlinear functions of the input variables,

$$y(\mathbf{x}, \mathbf{w}) = w_0 + \sum_{j=1}^{M-1} w_j \phi_j(\mathbf{x}) \tag{1}$$

where $\phi_j(\mathbf{x})$ are known as *basis functions*. The total number of parameters in this model will be M. It is often convenient to define an additional dummy 'basis function' $\phi_0(\mathbf{x}) = 1$.

$$y(\mathbf{x}, \mathbf{w}) = \sum_{j=0}^{M-1} w_j \phi_j(\mathbf{x}) = \mathbf{w}^{\mathrm{T}} \boldsymbol{\phi}(\mathbf{x}) \tag{2}$$

where $\mathbf{w} = (w_0, \ldots, w_{M-1})^{\mathrm{T}}$ and $\boldsymbol{\phi} = (\phi_0, \ldots, \phi_{M-1})^{\mathrm{T}}$. If the original variables comprise the vector $\mathbf{x}$, the nonlinear basis functions $\{\phi_j(\mathbf{x})\}$ express the extracted features.

1.1. Maximum likelihood and least squares

We assume that the target variable t is given by a deterministic function $y(\mathbf{x}, \mathbf{w})$ with additive Gaussian noise so that

$$t = y(\mathbf{x}, \mathbf{w}) + \epsilon \tag{3}$$

where ϵ is a zero mean Gaussian random variable with precision (inverse variance) β. Thus we can write

$$p(t|\mathbf{x}, \mathbf{w}, \beta) = \mathcal{N}\left(t|y(\mathbf{x}, \mathbf{w}), \beta^{-1}\right) \tag{4}$$

If we assume a squared loss function, then the optimal prediction, for a new value of $\mathbf{x}$, will be given by the conditional mean of the target variable. In the case of a Gaussian conditional distribution, the conditional mean will be simply

$$\mathbb{E}[t|\mathbf{x}] = \int t p(t|\mathbf{x}) \mathrm{d}t = y(\mathbf{x}, \mathbf{w}) \tag{5}$$

Note hat the Gaussian noise assumption implies that the conditional distribution of t given $\mathbf{x}$ is unimodal, which may be inappropriate for some applications.

2. The Bias-Variance Decomposition

Although the regularization terms can control overfitting for models, how to determine a suitable regularization coefficient λ? What's the *bias-variance* trade-off?

Given the conditional distribution $p(t|\mathbf{x})$, a popular choice is the squared loss function, for which the optimal

prediction is given by the conditional expectation, $h(x)$,

$$h(\mathbf{x}) = \mathbb{E}[t|\mathbf{x}] = \int t p(t|\mathbf{x}) \mathrm{d}t \tag{6}$$

The expected squared loss can be written in the form

$$\mathbb{E}[L] = \int \{y(\mathbf{x}) - h(\mathbf{x})\}^2 p(\mathbf{x}) \mathrm{d}\mathbf{x} + \int \{h(\mathbf{x}) - t\}^2 p(\mathbf{x}, t) \mathrm{d}\mathbf{x}\mathrm{d}t \tag{7}$$

The second term arises from the intrinsic noise on the data and represents the minimum achievable value of the expected loss. The first term depends on our choice for the function $y(\mathbf{x})$, and we will seek a solution for $y(\mathbf{x})$ which makes this term a minimum. For a particular data set

$$\mathbb{E}_{\mathcal{D}}\left[\{y(\mathbf{x};\mathcal{D}) - h(\mathbf{x})\}^2\right] = \underbrace{\{\mathbb{E}_{\mathcal{D}}[y(\mathbf{x};\mathcal{D})] - h(\mathbf{x})\}^2}_{(\text{bias})^2} + \underbrace{\mathbb{E}_{\mathcal{D}}\left[\{y(\mathbf{x};\mathcal{D}) - \mathbb{E}_{\mathcal{D}}[y(\mathbf{x};\mathcal{D})]\}^2\right]}_{\text{variance}} \tag{8}$$

The squared *bias* represents the extent to which the average prediction over all data sets differs from the desired regression function. The *variance* measures the extent to which the solutions for individual data sets vary around their average, and hence this measures the extent to which the function $y(\mathbf{x};\mathcal{D})$ is sensitive to the particular choice of data set.

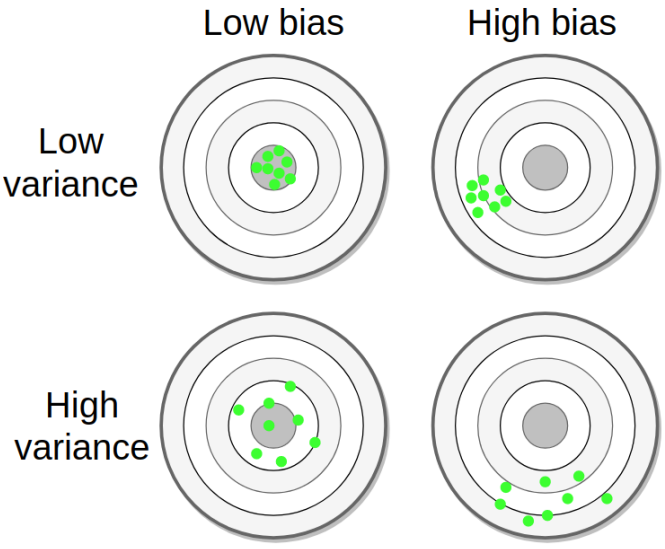

Figure 1: The bias-variance tradeoff.

The decomposition of the expected squared loss

$$\text{expected loss} = (\text{bias})^2 + \text{variance} + \text{noise} \tag{9}$$

where

$$\begin{aligned}(\text{bias})^2 &= \int \{\mathbb{E}_{\mathcal{D}}[y(\mathbf{x};\mathcal{D})] - h(\mathbf{x})\}^2 p(\mathbf{x}) \mathrm{d}\mathbf{x} \\ \text{variance} &= \int \mathbb{E}_{\mathcal{D}}\left[\{y(\mathbf{x};\mathcal{D}) - \mathbb{E}_{\mathcal{D}}[y(\mathbf{x};\mathcal{D})]\}^2\right] p(\mathbf{x}) \mathrm{d}\mathbf{x} \\ \text{noise} &= \int \{h(\mathbf{x}) - t\}^2 p(\mathbf{x}, t) \mathrm{d}\mathbf{x}\mathrm{d}t\end{aligned} \tag{10}$$

There is a trade-off between bias and variance, with very flexible models having low bias and high variance, and relatively rigid models having high bias and low variance, as shown in Figure 1.

3. Discriminant Functions

A discriminant is a function that takes an input vector $\boldsymbol{x}$ and assigns it to one of K classes, denoted $\mathcal{C}_k$. This section focuses on *linear discriminants* of two classes.

$$y(\mathbf{x}) = \mathbf{w}^{\mathrm{T}}\mathbf{x} + w_0 \tag{11}$$

where $\mathbf{w}$ is called a *weight vector*, and w_0 is a *bias*. An input vector $\mathbf{x}$ is assigned to class $\mathcal{C}_1$ if $y(\mathbf{x}) \leq 0$ and to class $\mathcal{C}_2$ otherwise. The decision boundary is $y(\mathbf{x}) = 0$. Consider two points $\mathbf{x}_A$ and $\mathbf{x}_B$, $y(\mathbf{x}_A) = y(\mathbf{x}_B) = 0$, we have $\mathbf{w}^{\mathrm{T}}(\mathbf{x}_A - \mathbf{x}_B) = 0$ and hence the vector $\mathbf{w}$ is orthogonal to every vector lying within the decision surface.

4. Probabilistic Generative Models

Consider first of all the classes. The posterior probability for class $\mathcal{C}_1$ can be written as

$$\begin{aligned}p(\mathcal{C}_1|\mathbf{x}) &= \frac{p(\mathbf{x}|\mathcal{C}_1)\, p(\mathcal{C}_1)}{p(\mathbf{x}|\mathcal{C}_1)\, p(\mathcal{C}_1) + p(\mathbf{x}|\mathcal{C}_2)\, p(\mathcal{C}_2)} \\ &= \frac{1}{1 + \exp(-a)} = \sigma(a)\end{aligned} \tag{12}$$

where we have defined

$$a = \ln \frac{p(\mathbf{x}|\mathcal{C}_1)\, p(\mathcal{C}_1)}{p(\mathbf{x}|\mathcal{C}_2)\, p(\mathcal{C}_2)} \tag{13}$$

For the case of $K > 2$ classes, we have

$$\begin{aligned}p(\mathcal{C}_k|\mathbf{x}) &= \frac{p(\mathbf{x}|\mathcal{C}_k)\, p(\mathcal{C}_k)}{\sum_j p(\mathbf{x}|\mathcal{C}_j)\, p(\mathcal{C}_j)} \\ &= \frac{\exp(a_k)}{\sum_j \exp(a_j)}\end{aligned} \tag{14}$$

which is known as the *normalized exponential* and is also known as the *softmax function*, as it represents a smoothed version of the 'max' function because if $a_k \gg a_j$ for all $j \neq k$, then $p(\mathcal{C}_k|\mathbf{x}) \simeq 1$, and $p(\mathcal{C}_j|\mathbf{x}) \simeq 0$. Here the quantities a_k are defined by

$$a_k = \ln p\left(\mathbf{x}|\mathcal{C}_k\right) p\left(\mathcal{C}_k\right) \tag{15}$$

4.1. Continuous inputs

Let us assume that the class-conditional densities are Gaussian and all classes share the same covariance matrix.

$$p\left(\mathbf{x}|\mathcal{C}_k\right) = \frac{1}{(2\pi)^{D/2}} \frac{1}{|\mathbf{\Sigma}|^{1/2}} \exp\left\{-\frac{1}{2}\left(\mathbf{x}-\boldsymbol{\mu}_k\right)^{\mathrm{T}} \mathbf{\Sigma}^{-1}\left(\mathbf{x}-\boldsymbol{\mu}_k\right)\right\} \tag{16}$$

Consider first the case of two classes.

$$p\left(\mathcal{C}_1|\mathbf{x}\right) = \sigma\left(\mathbf{w}^{\mathrm{T}}\mathbf{x} + w_0\right) \tag{17}$$

where we have defined

$$\mathbf{w} = \mathbf{\Sigma}^{-1}\left(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2\right) \tag{18}$$

$$w_0 = -\frac{1}{2}\boldsymbol{\mu}_1^{\mathrm{T}}\mathbf{\Sigma}^{-1}\boldsymbol{\mu}_1 + \frac{1}{2}\boldsymbol{\mu}_2^{\mathrm{T}}\mathbf{\Sigma}^{-1}\boldsymbol{\mu}_2 + \ln\frac{p\left(\mathcal{C}_1\right)}{p\left(\mathcal{C}_2\right)} \tag{19}$$

The decision boundaries are linear in input space. The prior probabilities $p(\mathcal{C}_k)$ enter only through the bias w_0 so that changes in the priors have the effect of making parallel shits of the decision boundary.

4.2. Maximum likelihood solution

Suppose we have a data set $\{\mathbf{x}_n, t_n\}$ where $n = 1, \ldots, N$. Here $t_n = 1$ denotes class $\mathcal{C}_1$ and $t_n = 0$ denotes class $\mathcal{C}_2$. We denote the prior class probability $p(\mathcal{C}_1) = \pi$, so that $p(\mathcal{C}_2) = 1 - \pi$. All classes share the same covariance matrix. For a data point $\mathbf{x}_n$ from class $\mathcal{C}_1$, we have $t_n = 1$ and hence

$$p\left(\mathbf{x}_n, \mathcal{C}_1\right) = p\left(\mathcal{C}_1\right) p\left(\mathbf{x}_n|\mathcal{C}_1\right) = \pi\mathcal{N}\left(\mathbf{x}_n|\boldsymbol{\mu}_1, \mathbf{\Sigma}\right) \tag{20}$$

Similarly for class $\mathcal{C}_2$, we have $t_n = 0$ and hence

$$p\left(\mathbf{x}_n, \mathcal{C}_2\right) = p\left(\mathcal{C}_2\right) p\left(\mathbf{x}_n|\mathcal{C}_2\right) = (1-\pi)\mathcal{N}\left(\mathbf{x}_n|\boldsymbol{\mu}_2, \mathbf{\Sigma}\right) \tag{21}$$

Thus the likelihood function is given by

$$p\left(\mathbf{t}|\pi, \boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \mathbf{\Sigma}\right) = \prod_{n=1}^{N}\left[\pi\mathcal{N}\left(\mathbf{x}_n|\boldsymbol{\mu}_1, \mathbf{\Sigma}\right)\right]^{t_n}\left[(1-\pi)\mathcal{N}\left(\mathbf{x}_n|\boldsymbol{\mu}_2, \mathbf{\Sigma}\right)\right]^{1-t_n} \tag{22}$$

where $\mathbf{t} = (t_1, \ldots, t_N)^{\mathrm{T}}$. It's convenient to maximize the log of the likelihood function. Consider first the maximization with respect to π. The terms in the log likelihood function that depend on π are

$$\sum_{n=1}^{N}\left\{t_n \ln\pi + (1-t_n)\ln(1-\pi)\right\} \tag{23}$$

Setting the derivative with respect to π equal to zero and rearranging, we obtain

$$\pi = \frac{1}{N}\sum_{n=1}^{N} t_n = \frac{N_1}{N} = \frac{N_1}{N_1 + N_2} \tag{24}$$

where N_1 denotes the total number of data points in class $\mathcal{C}_1$, and N_2 denotes the total number of data points in $\mathcal{C}_2$.

Now consider the maximization with respect to $\boldsymbol{\mu}_1$. Again we can pick out of the log likelihood function those terms that depend on $\boldsymbol{\mu}_1$ giving

$$\sum_{n=1}^{N} t_n \ln\mathcal{N}\left(\mathbf{x}_n|\boldsymbol{\mu}_1, \mathbf{\Sigma}\right) = -\frac{1}{2}\sum_{n=1}^{N} t_n\left(\mathbf{x}_n - \boldsymbol{\mu}_1\right)^{\mathrm{T}}\mathbf{\Sigma}^{-1}\left(\mathbf{x}_n - \boldsymbol{\mu}_1\right) + \text{ const.} \tag{25}$$

Setting the derivative with respect $\boldsymbol{\mu}_1$ to zero and rearranging, we obtain

$$\boldsymbol{\mu}_1 = \frac{1}{N_1}\sum_{n=1}^{N} t_n\mathbf{x}_n \tag{26}$$

which is simply the mean of all the input vectors $\mathbf{x}_n$ assigned to class $\mathcal{C}_1$. The corresponding result for $\boldsymbol{\mu}_2$ is given by

$$\boldsymbol{\mu}_2 = \frac{1}{N_2}\sum_{n=1}^{N}(1-t_n)\mathbf{x}_n \tag{27}$$

which again is the mean of all the input vectors $\mathbf{x}_n$ assigned to class $\mathcal{C}_2$.

Finally, consider the maximum likelihood solution for the shared covariance matrix $\mathbf{\Sigma}$, we have

$$-\frac{N}{2}\ln|\mathbf{\Sigma}| - \frac{N}{2}\mathrm{Tr}\left\{\mathbf{\Sigma}^{-1}\mathbf{S}\right\} \tag{28}$$

where we have defined

$$\mathbf{S} = \frac{N_1}{N}\mathbf{S}_1 + \frac{N_2}{N}\mathbf{S}_2 \tag{29}$$

$$\mathbf{S}_1 = \frac{1}{N_1}\sum_{n\in\mathcal{C}_1}\left(\mathbf{x}_n - \boldsymbol{\mu}_1\right)\left(\mathbf{x}_n - \boldsymbol{\mu}_1\right)^{\mathrm{T}} \tag{30}$$

$$\mathbf{S}_2 = \frac{1}{N_2}\sum_{n\in\mathcal{C}_2}\left(\mathbf{x}_n - \boldsymbol{\mu}_2\right)\left(\mathbf{x}_n - \boldsymbol{\mu}_2\right)^{\mathrm{T}} \tag{31}$$

Using the standard result for the maximum likelihood solution for a Gaussian distribution, we see that $\mathbf{\Sigma} = \mathbf{S}$, which represents a weighted average of the covariance matrices associated with each of the two classes separately.

References

[1] Bishop, Christopher M. Pattern recognition and machine learning. springer, 2006.

[2] https://www.machinelearningtutorial.net/2017/01/26/the-bias-variance-tradeoff/