BITS Pilani
Pilani Campus

BITS Pilani presentation

Tanmay Tulsidas Verlekar
CSIS

BITS Pilani
Pilani Campus

Applied Machine Learning SE ZG568 / SS ZG568
Lecture No.6

Naıve Bayes Classifier Revisited

To classify a test record, the naïve Bayes classifier computes the posterior probability for each class Y:

$$P(Y|\mathbf{X}) = \frac{P(Y)\prod_{i=1}^{d} P(X_i|Y)}{P(\mathbf{X})}. \tag{5.15}$$

Likelihood

Prior

Posterior

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

Normalizing constant

$$P(B) = \sum_{Y} P(B|A)P(A)$$

Example

Imagine that we have the following table detailing visits to a webpage:

This new case is the row vector (morning)12 and we want to know whether it is a ‘yes’ or a ‘no’.

Time	Buy
morning	no
afternoon	yes
evening	yes
morning	yes
morning	yes
afternoon	yes
evening	no
evening	yes
morning	no
afternoon	no
afternoon	yes
afternoon	yes
morning	yes

Logistic Regression

Logistic Regression (also called Logit Regression) is commonly used to estimate the probability that an instance belongs to a particular class,

Equation 4-1. Linear Regression model prediction

$$\hat{y} = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_n x_n$$

Equation 4-13. Logistic Regression model estimated probability (vectorized form)

$$\hat{p} = h_{\boldsymbol{\theta}}(\mathbf{x}) = \sigma\left(\mathbf{x}^T \boldsymbol{\theta}\right)$$

Logistic Regression

The logistic—noted $\sigma(\cdot)$—is a *sigmoid function* (i.e., *S*-shaped) that outputs a number between 0 and 1. It is defined as shown in Equation 4-14 and Figure 4-21.

Equation 4-14. Logistic function

$$\sigma(t) = \frac{1}{1 + \exp(-t)}$$

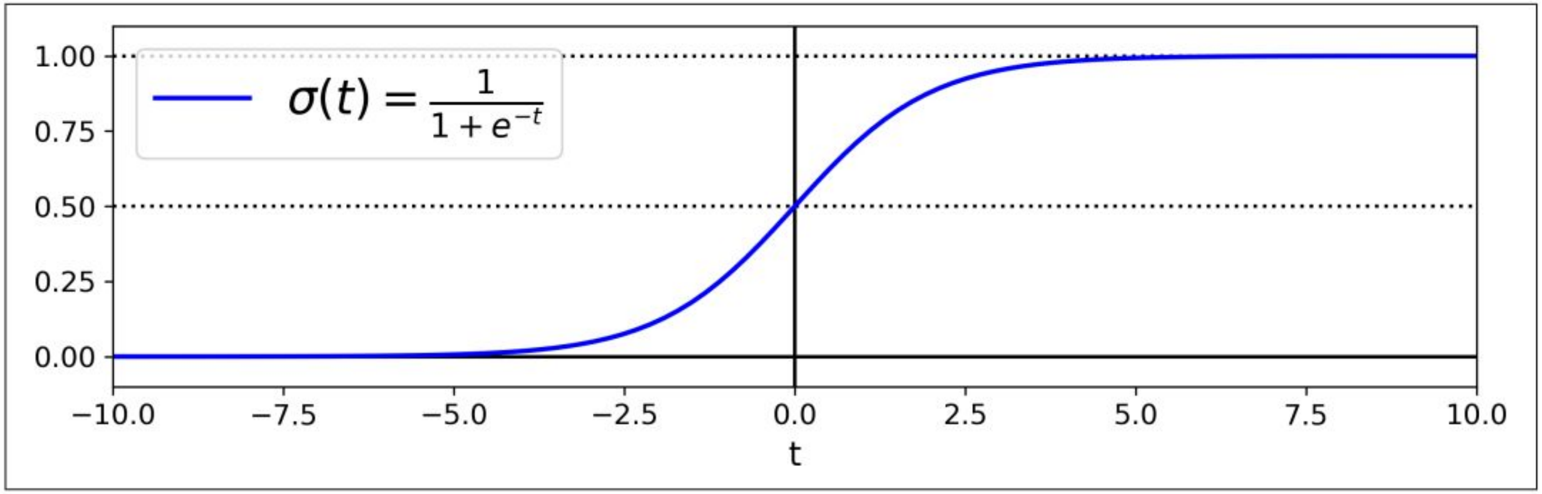

Figure 4-21. Logistic function

Model prediction

Equation 4-15. Logistic Regression model prediction

$$\hat{y} = \begin{cases} 0 & \text{if } \hat{p} < 0.5 \\ 1 & \text{if } \hat{p} \geq 0.5 \end{cases}$$

Training and Cost Function

The objective of training is to set the parameter vector θ so that the model estimates high probabilities for positive instances (y = 1) and low probabilities for negative instances (y = 0).

Equation 4-16. Cost function of a single training instance

$$c(\boldsymbol{\theta}) = \begin{cases} -\log(\hat{p}) & \text{if } y = 1 \\ -\log(1 - \hat{p}) & \text{if } y = 0 \end{cases}$$

Equation 4-17. Logistic Regression cost function (log loss)

$$J(\boldsymbol{\theta}) = -\frac{1}{m}\Sigma_{i=1}^{m}\left[y^{(i)}log\left(\hat{p}^{(i)}\right) + \left(1 - y^{(i)}\right)log\left(1 - \hat{p}^{(i)}\right)\right]$$

Equation 4-18. Logistic cost function partial derivatives

$$\frac{\partial}{\partial \theta_j} \mathrm{J}(\boldsymbol{\theta}) = \frac{1}{m} \sum_{i=1}^{m} \left(\sigma\left(\boldsymbol{\theta}^T \mathbf{x}^{(i)} \right) - y^{(i)} \right) x_j^{(i)}$$

Decision boundary

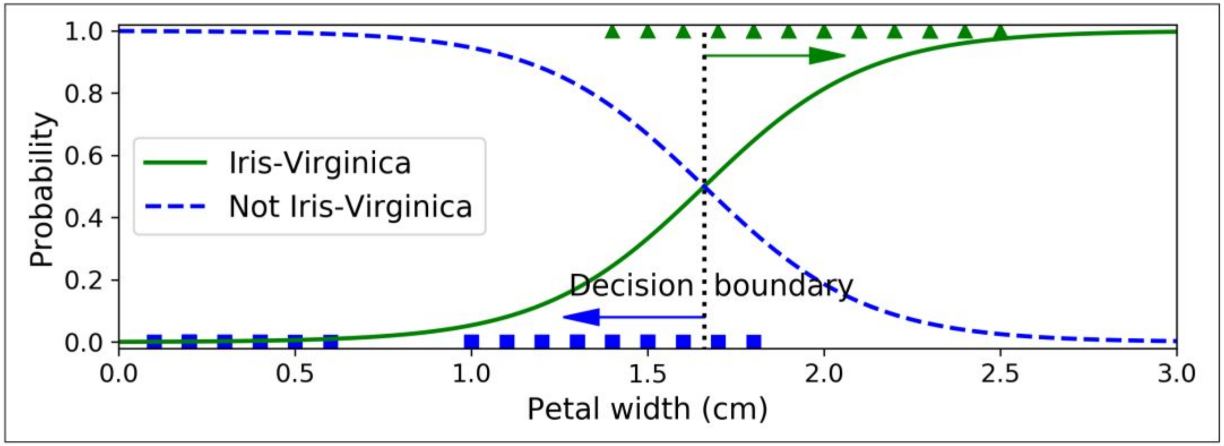

Figure 4-23. Estimated probabilities and decision boundary

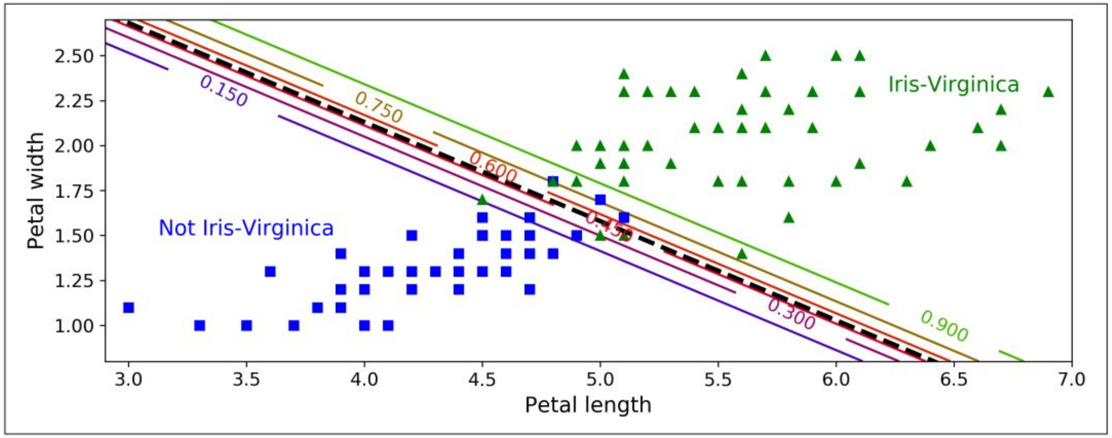

Figure 4-24. Linear decision boundary

Softmax Regression

Equation 4-20. Softmax function

$$\hat{p}_k = \sigma(\mathbf{s}(\mathbf{x}))_k = \frac{\exp\left(s_k(\mathbf{x})\right)}{\sum_{j=1}^{K} \exp\left(s_j(\mathbf{x})\right)}$$

- K is the number of classes.
- $\mathbf{s}(\mathbf{x})$ is a vector containing the scores of each class for the instance $\mathbf{x}$.
- $\sigma(\mathbf{s}(\mathbf{x}))_k$ is the estimated probability that the instance $\mathbf{x}$ belongs to class k given the scores of each class for that instance.

Cost function

Equation 4-22. Cross entropy cost function

$$J(\boldsymbol{\Theta}) = -\frac{1}{m}\sum_{i=1}^{m}\sum_{k=1}^{K} y_k^{(i)}\log\left(\hat{p}_k^{(i)}\right)$$

- $y_k^{(i)}$ is the target probability that the i$^{\text{th}}$ instance belongs to class *k*. In general, it is either equal to 1 or 0, depending on whether the instance belongs to the class or not.

Equation 4-23. Cross entropy gradient vector for class k

$$\nabla_{\boldsymbol{\theta}^{(k)}} J(\boldsymbol{\Theta}) = \frac{1}{m}\sum_{i=1}^{m}\left(\hat{p}_k^{(i)} - y_k^{(i)}\right)\mathbf{x}^{(i)}$$

Decision boundary

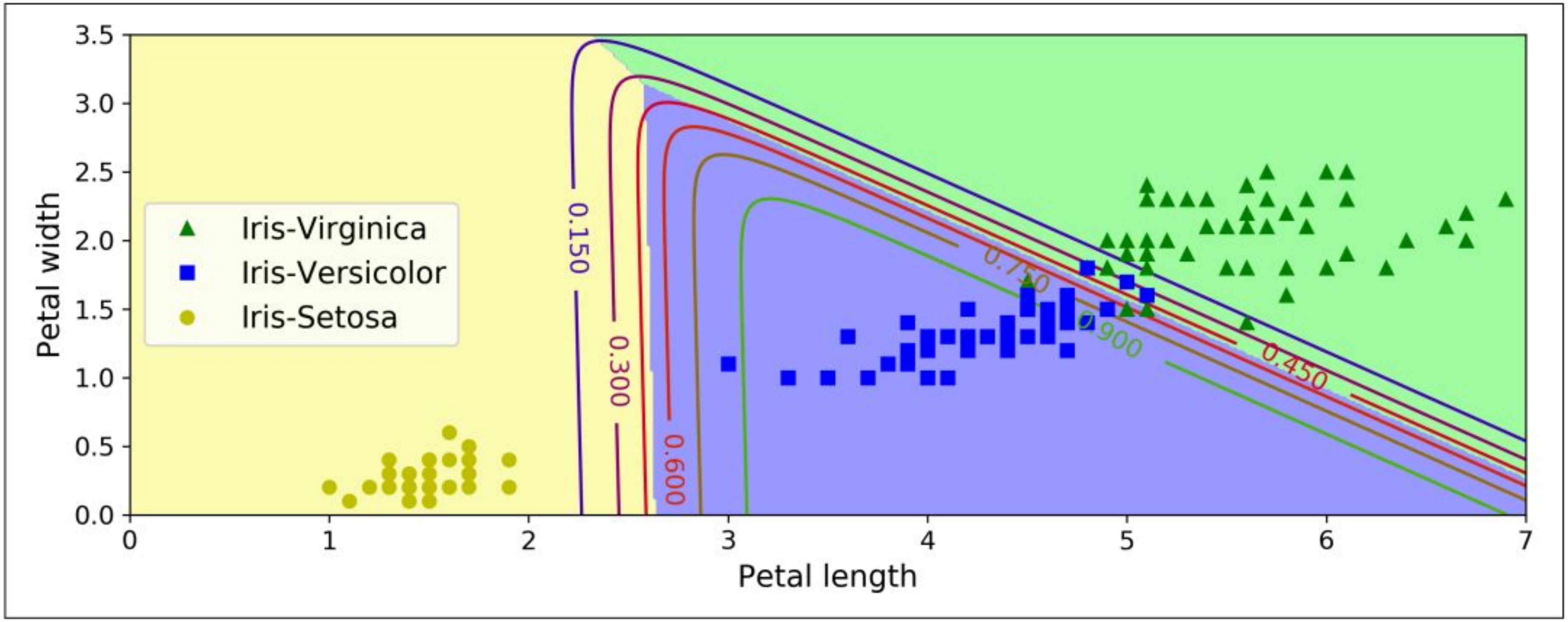

Figure 4-25. Softmax Regression decision boundaries