4.5Regularized Linear Models

正则化是用来防止模型过拟合而采取的手段。我们对代价函数增加一个限制条件，**限制其较高次的参数大小不能过大**。所以如果我们能让这些高次的系数接近于0的话，我们就能很好的拟合了，因此，我们对代价函数 J(θ)进行修改

L2 正则化 ridge

L2正则化对于绝对值较大的权重予以很重的惩罚，对于绝对值很小的权重予以非常非常小的惩罚，当权重绝对值趋近于0时，基本不惩罚。这个L2和平方项有关系，即越大的数，其平方越大，越小的数，比如小于1的数，其平方反而越小。

L2正则化，通常与岭回归（Ridge Regression）相关联，是一种用于避免回归模型过拟合的技术。它通过在损失函数中添加一个与系数平方和成比例的惩罚项来实现正则化。这种方法的目的是限制模型参数的大小，使得模型更为简单，从而减少过拟合的风险。

$$J(\theta) = MSE(\theta) + \frac{\alpha}{2}\sum_{i=1}^{n}\theta_i^2$$

- $L(\theta)$ 是正则化后的损失函数。
- MSE(θ) 是模型的均方误差（Mean Squared Error），表示未正则化的损失。
- θ 是模型参数（或系数）的向量。
- λ 是正则化强度的参数，控制正则化项的影响程度。
- 求和从1到n的θ_i的平方是模型系数的平方和，即L2范数。

L2正则化的关键特征是它倾向于将模型的系数缩小，而不是将它们完全置为零，这与L1正则化形成对比。这种方法的结果是，岭回归倾向于包含所有特征，但以较小的系数来限制它们的影响，这有助于提高模型的稳定性和预测性能。

在实践中，L2正则化有助于处理特征之间高度相关（多重共线性）的问题，因为它会平衡系数，避免它们过度波动。此外，L2正则化通常用于解决那些特征数量大于样本数量的问题，因为它可以减少模型的复杂性，防止过拟合。

L1 正则化 lasso

L1正则化，通常关联于Lasso（Least Absolute Shrinkage and Selection Operator）回归，是一种用于增强统计模型的技术。它通过将模型系数的绝对值之和添加到损失函数中来工作。具体来说，在L1正则化中，损失函数是原始损失函数与系数绝对值之和的线性组合。

$$J(\theta) = MSE(\theta) + \alpha\sum_{i=1}^{n}|\theta_i|$$

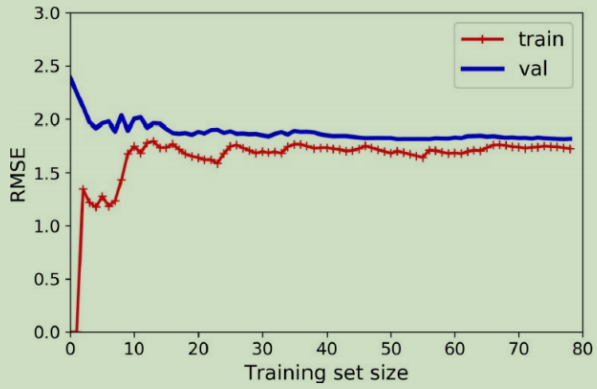
- $L(\theta)$ 是正则化后的损失函数。
- MSE(θ)MSE(θ) 是模型的均方误差（Mean Squared Error），代表未正则化的损失。
- θ 是模型参数（或系数）的向量。
- λ 是正则化强度的参数，控制正则化项的影响程度。

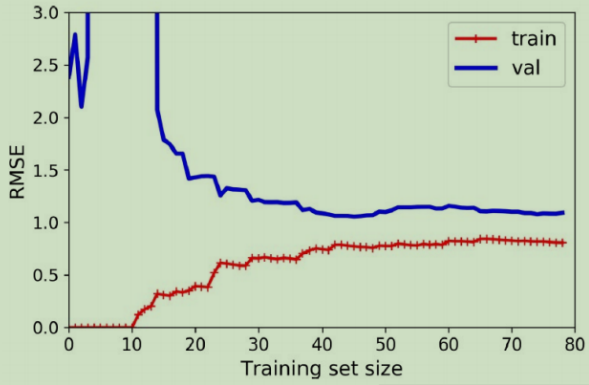
- 求和从1到n的θ_i的绝对值 是模型系数的绝对值之和，即L1范数。

L1正则化的关键特点在于它倾向于产生稀疏的系数。具体来说，一些系数可能会变为零，这意味着Lasso回归可以自动进行特征选择，从而有助于减少模型的复杂性和避免过拟合。这使得Lasso特别适用于具有大量特征的情况，其中只有少数特征是真正重要的。

Learning Curves

- Underfitting (1d) vs. Overfitting(300d)

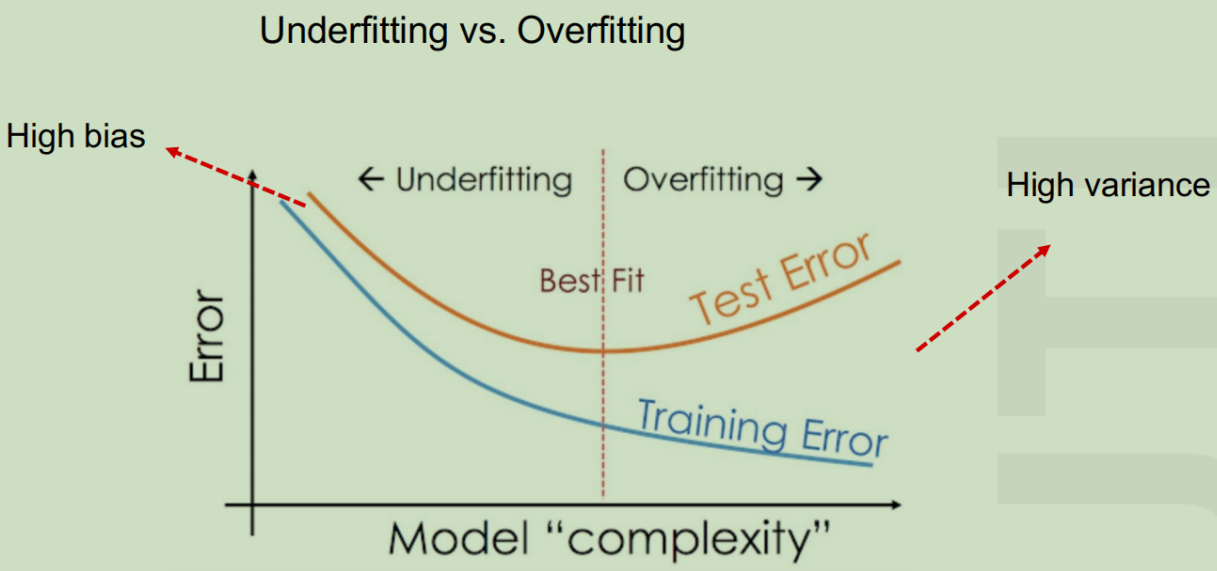




Poor performance on training data and poor performance on validation data.

Good performance on training data but poor performance on validation data

Learning Curves

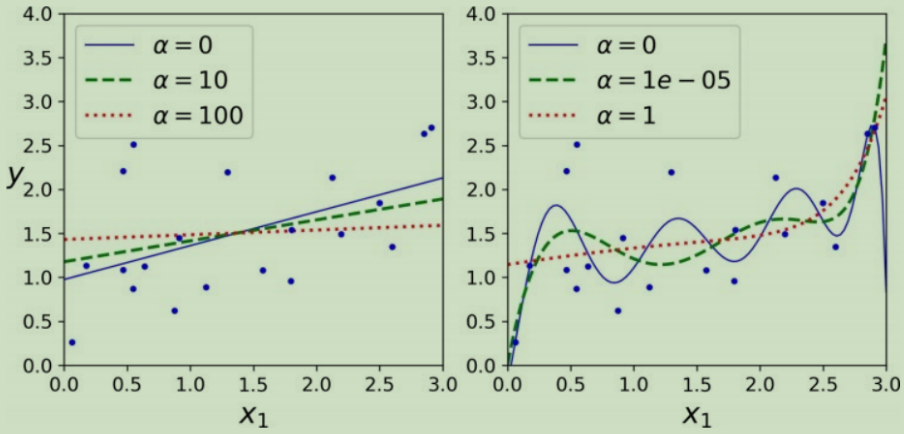


Regularized Linear Models

- **Ridge Regression(L2)**:

Cost function: $J(\boldsymbol{\theta}) = \text{MSE}(\boldsymbol{\theta}) + \alpha \frac{1}{2} \sum_{i=1}^{n} {\theta_i}^2$

This forces the learning algorithm to not only fit the data but also keep the model weights as small as possible.



Regularized Linear Models

- **Lasso Regression(L1)**:

Cost function: $J(\boldsymbol{\theta}) = \text{MSE}(\boldsymbol{\theta}) + \alpha \sum_{i=1}^{n} |\theta_i|$

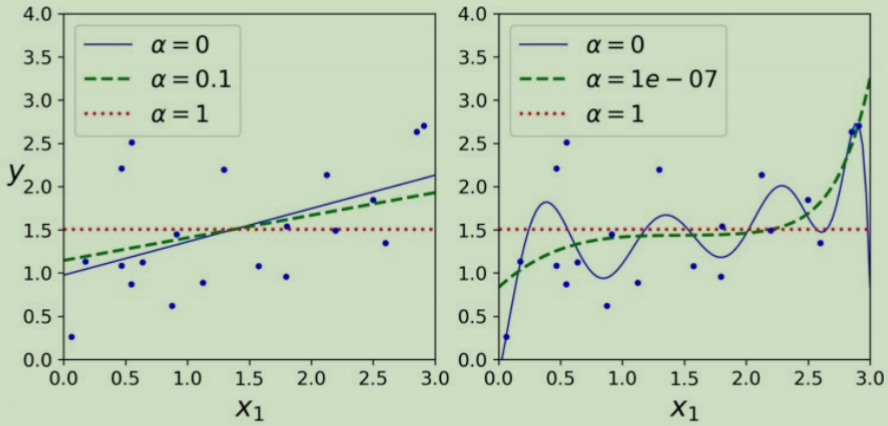


Figure 4-18. A linear model (left) and a polynomial model (right), both using various levels of Lasso regularization

Lasso Regression automatically performs **feature selection** and outputs a *sparse model*

逻辑回归是用来进行分类的。我们处理二分类问题。由于分成两类，我们便让其中一类标签为0，另一类为1。我们需要一个函数，对于输入的每组数据x(i)，都能映射成0~1之间的数。并且如果函数值大于0.5，就判定属于1，否则属于0。

Logistic Regression

Logistic Regression: is commonly used to **estimate the probability** that an instance belongs to a particular class.

Logistic Regression model

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

Sigmoid function:

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

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

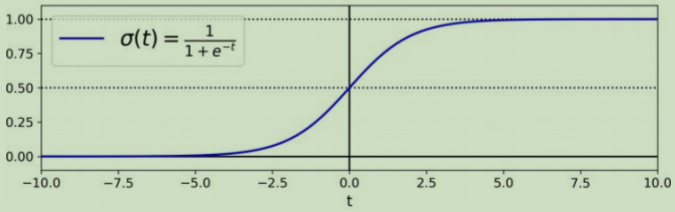


Figure 4-21. Logistic function

Logistic Regression (Softmax Regression)

Softmax Regression: for Multinomial Logistic Regression.

Softmax score for class k: $s_k(\mathbf{x}) = \mathbf{x}^{\intercal}\boldsymbol{\theta}^{(k)}$

Softmax function: $\hat{p}_k = \sigma(\mathbf{s}(\mathbf{x}))_k = \dfrac{\exp(s_k(\mathbf{x}))}{\sum_{j=1}^{K}\exp(s_j(\mathbf{x}))}$

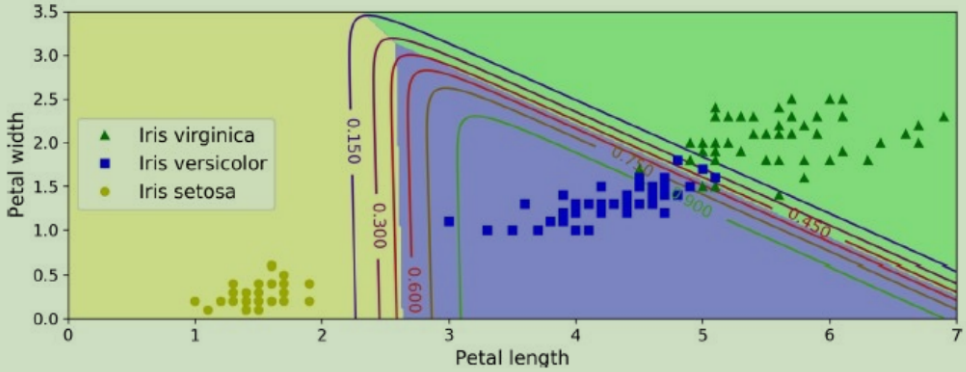


Figure 4-25. Softmax Regression decision boundaries