

特征缩放：

对于方程 $y = \theta_0 + \theta_1 * x_1 + \theta_2 * x_2$ ，如果 x_1 的规模远大于 x_2 ，那么 x_1 对应的参数 θ_1 的变化就会比较小；而 x_2 对应的参数 θ_2 的变化就会比较大。这是因为在更新参数的时候，公式为：

$$\begin{aligned}\theta_0 &:= \theta_0 - \alpha \frac{1}{m} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)}) x_0^{(i)} \\ \theta_1 &:= \theta_1 - \alpha \frac{1}{m} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)}) x_1^{(i)} \\ \theta_2 &:= \theta_2 - \alpha \frac{1}{m} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)}) x_2^{(i)}\end{aligned}$$

可以看到参数的更新幅度与对应数据的大小呈正比，但是对应的参数本身应该很小才对，这就导致了需要很长时间才可以对这个参数达到收敛的目的（当我们尽可能地将学习率降低）。对应于 x_2 而言，参数本身比较大，而收敛时候对应的 x_2 比较小，为使收敛速度增快，应当尽可能增大学习率。此处产生了矛盾，当选择某一个学习率的时候，往往可能出现两难问题，其中一个在等待另外一个收敛。因此进行特征缩放将问题规模在每个尺度上保持一致，使得收敛同步，减少迭代次数。

特征缩放的几种方法：

Rescaling (min-max normalization) [see]

Also known as min-max scaling or min-max normalization, is the simplest method and consists in rescaling the range of features to scale the range to $[0, 1]$ or $[-1, 1]$. Selecting the target range depends on the nature of the data. The general formula is given by:

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)}$$

where x is an original value, x' is the normalized value. For example, suppose that we have the students' weight data, and this attribute will also span $[100 \text{ pounds}, 200 \text{ pounds}]$. To normalize this data, we first subtract 100 from each student's weight and divide the result by 100 (the difference between the maximum and minimum weights).

Mean normalization [see]

$$x' = \frac{x - \text{average}(x)}{\max(x) - \min(x)}$$

where x is an original value, x' is the normalized value.

Standardization [see]

In machine learning, we can handle various types of data, e.g., audio signals and pixel values for image data, and this data can include multiple **dimensions**. Feature standardization makes the values of each feature in the data have zero-mean (when subtracting the mean in the numerator) and unit-variance. This method is widely used for normalization in many machine learning algorithms (e.g., **support vector machines**, **logistic regression**, and **artificial neural networks** [2] [Wolfe, 2016]). The general method of calculation is to determine the distribution **mean** and **standard deviation** for each feature. Next we subtract the mean from each feature. Then we divide the values (mean is already subtracted) of each feature by its standard deviation.

$$x' = \frac{x - \mu}{\sigma}$$

Where x is the original feature vector, μ is the mean of that feature vector, and σ is its standard deviation.

Scaling to unit length [see]

Another option that is widely used in machine-learning is to scale the components of a feature vector such that the complete vector has length one. This usually means dividing each component by the **Euclidean length** of the vector:

$$x' = \frac{x}{\|x\|}$$

In some applications (e.g., **Recommendation**) it can be more practical to use the L1 norm (i.e., **Manhattan Distance**, **City-Block Length** or **Taxicab Geometry**) of the feature vector. This is especially important if in the following learning steps the **Scalar Metric** is used as a distance measure.