HOWARD UNIVERSITY

Department of Mathematics

Moussa Doumbia, Ph.D.

Data Standardization Methods

1. Z-Score Standardization (Standard Scaling)

Description: This method transforms the data to have a mean of 0 and a standard deviation of 1. It's calculated by subtracting the mean of the data from each data point and then dividing by the standard deviation.

Formula:

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

Use Cases: Particularly useful in algorithms that assume the data is normally distributed and in methods where the scale of input features matters, such as k-nearest neighbors, neural networks, and principal component analysis (PCA).

2. Min-Max Scaling (Normalization)

Description: Scales and shifts the data so that it lies within a given range, typically 0 to 1. It's done by subtracting the minimum value of the feature and then dividing by the range of the feature.

Formula:

$$x_{norm} = \frac{x - x_{min}}{x_{max} - x_{min}}$$

Use Cases: Useful when you need to maintain the shape of the original distribution and do not want to assume a normal distribution. It is also good for algorithms that compute distances between data points and require a bounded range, such as neural networks.

3. Max Abs Scaling

Description: Scales each feature by its maximum absolute value. This method scales the data within the range [-1, 1] by dividing each value by the maximum absolute value in the feature.

Formula:

$$x_{scaled} = \frac{x}{\max(x)}$$

Use Cases: It is useful for data that is already centered at zero or sparse data. It is beneficial for algorithms that are sensitive to the scale of the data but do not require the data to have a normal distribution.

4. Robust Scaling

Description: Similar to Z-score standardization but uses the median and the interquartile range (IQR) instead of mean and standard deviation. This method reduces the effects of outliers.

Formula:

$$x_{robust} = \frac{x - Median(x)}{IQR(x)}$$

Use Cases: Ideal for datasets with outliers. It is useful in algorithms that assume data is centered but can be negatively impacted by outliers, such as support vector machines (SVMs) and k-nearest neighbors.

Quantile Transformation (Rank Scaling) Process

Process of Quantile Transformation (Rank Scaling)

1. **Sort Data:** Arrange the data points in ascending order.
2. **Calculate Quantiles:** Determine the quantiles of the data. The q-th quantile of a dataset is a value such that at most $q\%$ of the data is less than or equal to that value and at most $(100 - q)\%$ is greater.
3. **Map to Desired Distribution:** For each data point, calculate its empirical cumulative distribution function (CDF) value. This is essentially the rank of the data point normalized by the total number of points.
4. **Apply the Inverse CDF of the Target Distribution:** Transform the empirical CDF values using the inverse CDF (quantile function) of the target distribution (e.g., uniform or normal). This step maps the original data distribution to the desired distribution.

In practice, for a data point x_i, the transformed value x_i' in a quantile transformation to a normal distribution would be:

$$x_i' = \Phi^{-1}\left(\frac{rank(x_i)}{n+1}\right)$$

where:

- Φ^{-1} is the inverse CDF of the normal distribution.
- $rank(x_i)$ is the rank of x_i in the dataset.
- n is the total number of observations in the dataset.
- The division by $n + 1$ instead of n is a common practice to avoid issues with the highest and lowest ranks.

For a uniform distribution, the transformed value would simply be the normalized rank itself, as the inverse CDF (quantile function) of a uniform distribution on the interval $[0, 1]$ is the identity function.

This process ensures that the distribution of the transformed data matches the desired target distribution, making it a powerful technique for normalizing data with arbitrary distributions.

6. Unit Vector Scaling

Description: Scales the component of a feature vector such that the complete vector has a length of one. It is often done by dividing each component by the Euclidean length of the vector.

Formula:

$$x_{unit} = \frac{x}{x}$$

Use Cases: Commonly used in text classification or clustering when using models like cosine similarity, where the magnitude of the feature vector should not affect the computation.

Choosing the Right Method

The choice of standardization method depends on the specific requirements of your dataset and the machine learning algorithms you plan to use. Consider the presence of outliers, the distribution of the data, the scale of the features, and the algorithm's assumptions about the data distribution when selecting a standardization technique.