Methods for Scaling and Cleaning Data

Before we talk about using linear algebra on matrices and vectors representing data, the data first needs to be put in a form that we can use. Typical things to consider:

- Missing or inconsistent data

  If some data is missing information, you may have to remove those entries. There are methods we can use as an "educated guess" for these values (we'll discuss this in the SVD section).

  We may need to clean up inconsistent entries (for example, we may see some entries with "United States" and some entries with "US"). Especially with categorical data (below), it's good to look at all the different possible selections.

- Categorical data

  Categorical data is data given in classes that are not numerical. For example, you might see "Yes" or "No", or "Low", "Neutral", "High". We need to convert these to numerical data.

- Data statistics

  Mean, max, min, standard deviation. Think of outliers as at least three standard deviations from the mean (this is just a rule of thumb- look at your data). The scaling of the data itself may be problematic when comparing one column of data to another.

1 Scaling Data

Scaling ensures that features have a comparable range, preventing certain variables from dominating others in distance-based algorithms. If one feature (or column) of data is very large in comparison to another, in order to minimize the least squares error, our model will focus on minimizing the error in that largest column first (which may not be what we're looking for).

Here are some common ways of scaling a column of data (or a single "feature").

1.1 Min-Max Scaling

Given a set of data, $\{x_1, x_2, \ldots, x_p\}$, we can scale it so that the data has a minimum of 0 and a maximum of 1. This is also referred to as Min-Max scaling.

$$x_i \Rightarrow \frac{x_i - x_{\min}}{x_{\max} - x_{\min}} \tag{1}$$

where $x_{\min}$ and $x_{\max}$ are the minimum and maximum values. Notice that if $x_i = x_{\min}$, then the result is 0, and if $x_i = x_{\max}$, the result is 1.

One might do this if the data is somewhat uniformly distributed, without outliers. This method also gives all non-negative numbers, which may or may not be desirable (it would be something to keep in mind).

If you want to keep negative numbers, and if you may have a few numbers that are larger/smaller than the others, a better scaling would be the next scaling method.

1.2 Standardization (Z-score Normalization)

Again, given data $\{x_1, x_2, \ldots, x_p\}$, this method rescales data to have zero mean and unit variance.

Let $\bar{x}$ and s be the sample mean of variance. Then

$$x_i \to \frac{x_i - \bar{x}}{s} \tag{2}$$

This is probably the most widely used scaling.

If you have large/small data that you can't remove, then instead of using the mean and standard deviation, you might use the **sample median** and the interquartile range (IQR) to do the scaling. The interquartile range (IQR) is a measure of the spread of the middle 50% of a dataset. It is calculated as the difference between the third quartile (Q3) and the first quartile (Q1), where the second quartile is the median.

1.3 Robust Scaling

Robust scaling uses the median and interquartile range (IQR) to reduce the influence of outliers:

$$x_i \to \frac{x_i - \text{median}}{\text{IQR}} \tag{3}$$

2 Transform Categorical Data Into Numerical Data

Categorical data needs to be transformed into numerical representations for machine learning.

We'll take a couple of examples. The first is called "ordinal data" in that the data has a natural ranked order. For example, coming in "First", "Second", "Third" in a competition.

For the second example suppose we have three categories, like "Electric","Wood", or Gas".

2.1 Label Encoding

Label encoding assigns a unique integer to each category. In the first example set, we would naturally assign

$$\text{First} \Rightarrow 1, \qquad \text{Second} \Rightarrow 2, \qquad \text{Third} \Rightarrow 3$$

Since there was an ordering (third is farthest from first, for example), then this substitution should maintain that.

In the second example, we could make the substitutions:

$$\text{Electric} \Rightarrow 1, \qquad \text{Wood} \Rightarrow 2, \qquad \text{Gas} \Rightarrow 3.$$

But in this case, order did not matter, and we're actually inducing a metric ("Gas" is closer to "Wood" than "Electric", for example). This may not be the best choice for this example- It may be better suited for data where order does matter.

For these cases, it's better to try to use something where the k choices would all be equidistant to each other (or close to it). That's the so-called "one hot encoding" below.

2.2 One-Hot Encoding

One-hot encoding converts categorical variables into binary column (with 0's and 1's). If we have k categories, then we create a binary column with seven entries. The presence of a category is marked with 1, and the absence is marked with 0. In our previous example,

$$\text{Electric} \Rightarrow \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \qquad \text{Wood} \Rightarrow \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \qquad \text{Gas} \Rightarrow \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}.$$

2.3 Frequency Encoding

This method encodes categories based on their frequency in the dataset: For example, consider a dataset with a categorical feature "City":

City	Frequency Encoding
New York	0.4
Los Angeles	0.3
Chicago	0.2
Houston	0.1

Table 1: Example of Frequency Encoding

2.4 Target Encoding

If you have a target variable that you're modeling (in the example below, "house price"), then one method for encoding a categorical variable like "City" might be to replace the city with the mean of the target variable (using that choice of categorical value). Here's an example of doing that:

City	Target Encoding (Mean House Price)
New York	500,000
Los Angeles	450,000
Chicago	350,000
Houston	300,000

Table 2: Example of Target Encoding

Concluding Thoughts

Clean data with no missing values will seldom be available. In fact, many practioners of data science will tell you that most of their time is spent cleaning data. It isn't the most exciting aspect of data science, but it can be critical for successfully modeling the data.