

Data Preparation and Pre-processing

Ryan Miller

Introduction

Last time we learned that KNN and decision trees estimate $f(\mathbf{X})$ in very different ways:

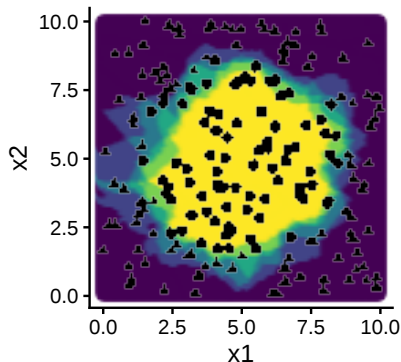
- ▶ KNN uses distances between observations.
- ▶ Decision trees use sequential splitting rules.

KNN's reliance on distance necessitates the use of data pre-processing steps in way that decision trees do not.

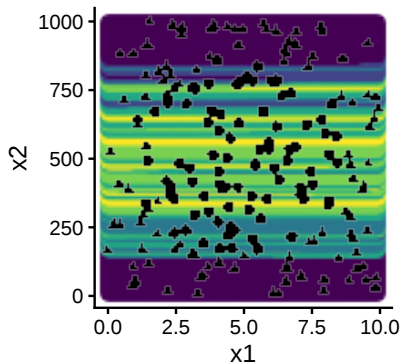
Introduction

In our toy example, x_1 and x_2 had similar scales (ie: similar standard deviations), but what if we multiplied all values of x_2 by 100?

Same scale



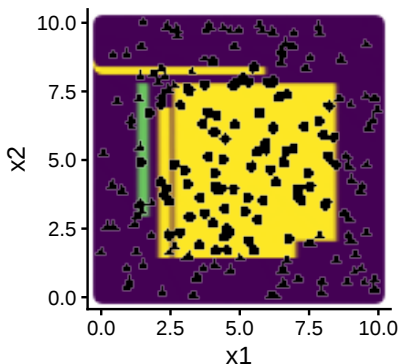
Larger scale for x_2



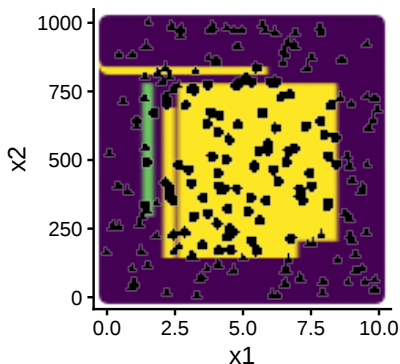
Introduction (cont.)

This same issue doesn't exist for decision trees, but why?

Same scale for



Larger scale for x_2



Re-scaling

Decision trees are considered *scale-invariant*, meaning they are not influenced by the scaling or normalizing the input features.

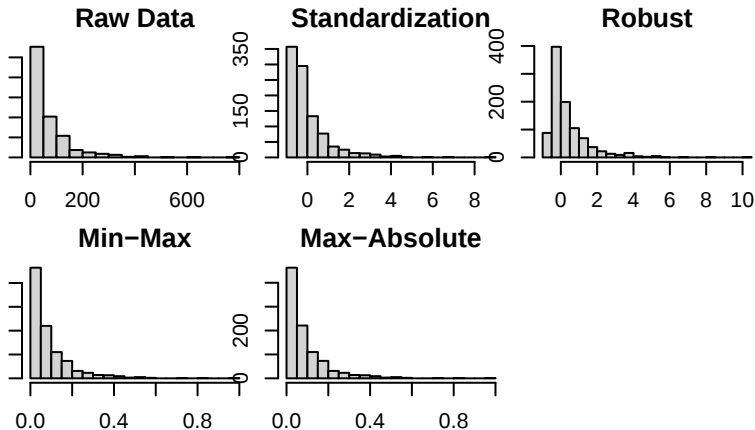
Conversely, KNN is sensitive to scale, so data must be *pre-processed* using a scaling step:

1. **Standardization:** $x_i^* = \frac{x_i - \text{mean}(x)}{\text{sd}(x)}$
2. **Robust scaling:** $x_i^* = \frac{x_i - \text{median}(x)}{\text{IQR}(x)}$
3. **Min-Max scaling:** $x_i^* = \frac{x_i - \min(x)}{\max(x) - \min(x)}$
4. **Max-Absolute scaling:** $x_i^* = \frac{x_i}{\max(|x|)}$

- ▶ Standardization forces features to have a *mean of zero* and a *standard deviation of one*
 - ▶ Robust scaling forces features to have a *median of zero*, and it can be beneficial for data with large outliers
- ▶ Min-Max scaling maps each feature onto a $[0,1]$ interval, which can have computational advantages
 - ▶ Max-Absolute scaling is similar to Min-Max scaling, but the output range is $[-1,1]$ and it will *preserve exact zeros* (important for sparse data)

Scaling vs. Normalization

Scaling changes the range of your data, it does not change the distributional shape:



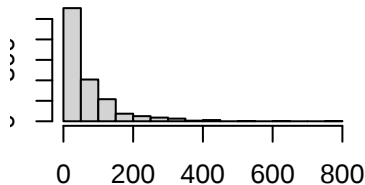
Normalization

Sometimes we might want to change the distributional shape of our inputs to reduce the impact of skew/outliers. Three possible pre-processing operations are:

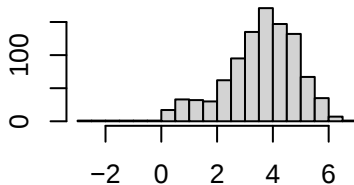
1. Log-transformation - simply taking the logarithm of each of the variable's values
2. Box-Cox transformation - $x_i^* = \frac{x_i^\lambda - 1}{\lambda}$ for $\lambda \neq 0$ and $x_i > 0$
3. Quantile mapping - map each quantile of the observed data to the corresponding quantile of a $Unif(0,1)$ distribution

Normalization

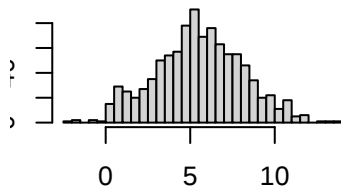
Raw Data



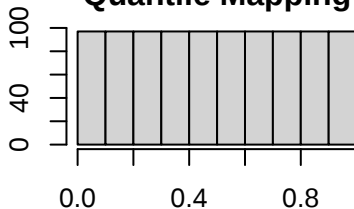
Log



Box-Cox (lam = 0.2)



Quantile Mapping



One-hot Encoding

Many algorithms require numerical inputs. **One-hot encoding** is a general way to represent categorical data numerically:

College	State
Grinnell College	IA
University of Iowa	IA
University of Minnesota	MN
Middlebury College	VT
Carlton College	MN



College	State = "IA"	State = "MN"	State = "VT"
Grinnell College	1	0	0
University of Iowa	1	0	0
University of Minnesota	0	1	0
Middlebury College	0	0	1
Carlton College	0	1	0

One-hot Encoding

Dropping the first dummy variable is sometimes done to prevent redundancy. In our example colleges in Iowa are still identifiable via having zeros in both dummy variables.

College	State
Grinnell College	IA
University of Iowa	IA
University of Minnesota	MN
Middlebury College	VT
Carlton College	MN



College	State = "MN"	State = "VT"
Grinnell College	0	0
University of Iowa	0	0
University of Minnesota	1	0
Middlebury College	0	1
Carlton College	1	0

Guidelines

- ▶ Be aware of algorithms that are sensitive to scale, such as KNN
 - ▶ There's rarely any harm introduced by re-scaling, so its sensible pre-processing step in most applications
- ▶ Use exploratory visualizations to identify features with highly skewed distributions or extreme outliers and consider normalizing transformations
- ▶ Represent categorical features using one-hot encoding
 - ▶ Be aware that dropping the first dummy variable is beneficial in models like linear regression where linear dependencies among predictors cause problems
- ▶ Next week we'll learn about *cross-validation*, which will provide us a data-driven tool for determine which pre-processing steps improve model performance

Preparing for Quiz #1

The following topics from these slides might appear on our first quiz:

1. Why scaling is important for KNN but not decision trees
2. Where pre-processing fits in the machine learning workflow (to prevent data leakage)
3. How categorical data can be represented numerically via one-hot encoding