

DATA 3464: Fundamentals of Data Processing

Interaction Effects

Charlotte Curtis

February 12, 2026

Topic overview

- Definitions and description of interaction effects
- Detecting interaction effects
- A brief discussion on feature selection

Resources used:

- [Feature Engineering Chapter 7](#)
- [Feature Engineering Chapter 10](#)

Definition

Two or more predictors are said to interact if their combined effect is different (less or greater) than what we would expect if we were to add the impact of each of their effects when considered alone. -- Feature Engineering, Ch 7

- Interactions matter if they affect the *outcome*
- Features may have a relationship without having an interaction effect

Example: Stroke data from week 1

Mathematical representation

A linear model is trying to fit:

$$\hat{y} = w_0 + w_1x_1 + w_2x_2 + \dots + w_kx_k$$

To consider interaction effects, we add the **product** of features. For a model with only features x_1 and x_2 :

$$\hat{y} = w_0 + w_1x_1 + w_2x_2 + w_3x_1x_2$$

Similar to a **basis expansion** transformation that adds polynomial terms

Which interactions to include?

First, use your domain knowledge! Then consider **guiding principles**:

- **Hierarchy principle**: The higher the degree of interaction, the less likely the interaction will explain variation in the response
- **Effect sparsity**: only a fraction of the possible interactions are responsible for variation in the response
- **Heredity principle**: for interaction term x_1x_2 to be considered:
 - x_1 AND x_2 must be significant (strong heredity), or
 - x_1 OR x_2 must be significant (weak heredity)

Brute-force approach

- Try all second order interactions and see if they improve the model, e.g. using scikit-learn's [PolynomialFeatures](#)
- Based on guiding principles, this results in many extraneous features
- **Feature selection** can be used to prune them back

Feature selection

- Not just for interaction terms!
- Fields such as bioinformatics can end up afflicted by the [Curse of Dimensionality](#) or the $p \gg n$ problem
- Models do not like to have more features than samples:
 - Risk of overfitting
 - Multicollinearity issues
 - Can negatively impact performance

Goal of feature selection: Reduce the number of predictors as far as possible without compromising predictive performance

Performing feature selection

- Many different ways to choose which features to keep
 - **Intrinsic methods:** some models effectively ignore irrelevant features
 - **Filter methods:** filter features based on some criteria (e.g. correlation)
 - **Wrapper methods:** select subset based on model results, then iterate
- As usual, Scikit-learn can [help with this](#)

A subtle source of data leakage is in performing feature selection on the entire training set, then cross-validating

Summary

- Interactions between features can be considered by adding new features with their product
- This can cause a dimensionality explosion
- Particularly for small datasets, feature selection is then needed to avoid adverse effects of irrelevant features

As always, don't make feature selection decisions on test data!

Overall Processing Order

In general, the recommended order is:

Numeric Features

1. Impute any missing features
2. Compose interaction terms
3. Transform if necessary
4. Rescale

Categorical features

1. Encode + Impute
2. Compose interaction terms
3. Rescale if necessary (e.g. high cardinality ordinal)

Coming up next

- Reading week!!
- Midterm practice (both in lab and in class)
- I will post practice questions during reading week as well
- After midterm: text wrangling