Tufts

Class #10:
Feature Engineering

Machine Learning (COMP 135): M. Allen, 20 Feb. 20

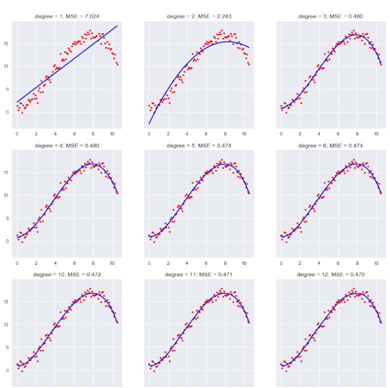
1

Feature Engineering

- As we saw with polynomial regression, we often want to ***transform*** our data in order to get better results from a machine learning algorithm
- We often get better results by:
  1. Changing how features are represented.
  2. Adding new features.
  3. Deleting/ignoring some features.

2

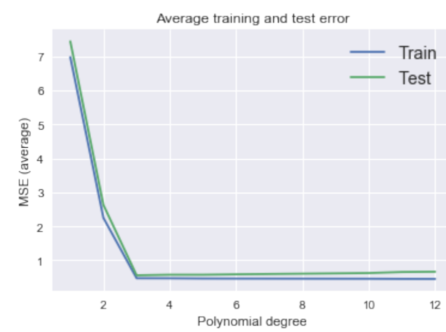
Example: Higher-Order Polynomial Features



- As seen in Assignment 02, transforming data by mapping to higher-degree polynomials, and then fitting a linear regression, can reduce error
  - Gains are most significant at first, and then error starts to level off

3

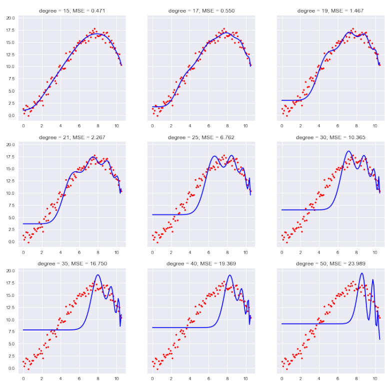
The Cost of Feature Transformation



- Not every transformation is as useful as others
- The polynomial degrees above 3 from previous slide also start to show some evidence of over-fitting, as revealed by cross-validation

4

The Cost of Feature Transformation



- Not every transformation is useful—at very high polynomials, some of the mathematics of the linear regression libraries in `sklearn` break down
  - Mathematically, we expect better and better fits
  - In practice, the method ceases working effectively, and models are generally useless

5

Feature Rescaling

Input: Each numeric feature has arbitrary min/max
- Some in [0, 1], Some in [-5, 5], Some [-3333, -2222]

Transformed feature vector
- Set each feature value f to have [0, 1] range

$$\phi(x_n)_f = \frac{x_{nf} - \min_f}{\max_f - \min_f}$$

- min_f = minimum observed in training set
- max_f = maximum observed in training set

6

Feature Standardization

Input: Each feature is numeric, has arbitrary scale

Transformed feature vector
- Set each feature value f to have zero mean, unit variance

$$\phi(x_n)_f = \frac{x_{nf} - \mu_f}{\sigma_f}$$

μ_f Empirical mean observed in training set

σ_f Empirical standard deviation observed in training set

7

Feature Standardization

$$\phi(x_n)_f = \frac{x_{nf} - \mu_f}{\sigma_f}$$

- Treats each feature as "Normal(0, 1)"
- Typical range will be -3 to +3
  - *If original data is approximately normal*
- Also called z-score transform

8

Best Subset Selection

Algorithm 6.1 *Best subset selection*

1. Let $\mathcal{M}_0$ denote the *null model*, which contains no predictors. This model simply predicts the sample mean for each observation.
2. For $k = 1, 2, \ldots p$:
   (a) Fit all $\binom{p}{k}$ models that contain exactly k predictors.
   (b) Pick the best among these $\binom{p}{k}$ models, and call it $\mathcal{M}_k$. Here *best* is defined as having the smallest RSS, or equivalently largest R^2.
3. Select a single best model from among $\mathcal{M}_0, \ldots, \mathcal{M}_p$ using cross-validated prediction error, C_p (AIC), BIC, or adjusted R^2.

- Main issue: too many subsets
  - There are $O(2^p)$ such collections of features
  - For problems with large feature-sets, this grows quickly infeasible

9

Forward Stepwise Selection

1. Start with zero feature model (guess mean)
   - Store as `M_0`
2. Add best scoring single feature (among all `F`)
   - Store as `M_1`
3. For each size `k = 2, ... F`
   - Try each possible not-included feature (`F – k + 1`)
   - Add best scoring feature to the model `M_k-1`
   - Store as `M_k`
4. Pick best among `M_0, M_1, ... M_F`, based upon the **validation** data

10

Best vs Forward Stepwise

# Variables	Best subset	Forward stepwise
One	rating	rating
Two	rating, income	rating, income
Three	rating, income, student	rating, income, student
Four	cards, income, student, limit	rating, income, student, limit

TABLE 6.1. *The first four selected models for best subset selection and forward stepwise selection on the* Credit *data set. The first three models are identical but the fourth models differ.*

Easy to find cases where forward stepwise 's greedy approach doesn't deliver best possible subset.

11

Backwards Stepwise Selection

The basic forward model can also be run backwards:

1. Start with all features
2. Gradually test all models with one feature removed from each
3. Repeat to remove 2, 3, … features, down to single-feature versions

12

Next Week

- **Special schedule**: Class Wednesday & Thursday
- **Topics**: Clustering methods
  - Readings linked from class schedule page
- **Assignments**:
  - Homework 03: due Wednesday, 26 Feb., 9:00 AM
    - Logistic regression & decision trees
  - Project 01: due Monday, 09 March, 5:00 PM
    - Feature engineering and classification for image data
  - Midterm Exam: Wednesday, 11 March
- **Office Hours**: 237 Halligan
  - Monday, 10:30 AM – Noon
  - Tuesday, 9:00 AM – 1:00 PM
  - TA hours can be found on class website