STAT3612 Lecture 4

Feature Engineering

Dr. Aijun Zhang

22 September 2020

Department of 統計及精算學系
Statistics & Actuarial Science

Table of Contents

Feature Engineering

- **Feature engineering** refers to the process of creating new input features to improve model performance.
- **Data preprocessing** usually refers to data cleaning, vector representation, missing value imputation, feature scaling (normalization/standardization), data reduction and splitting. It may also include feature engineering as a key procedure.
- "Coming up with features is difficult, time-consuming, requires expert knowledge. *Applied machine learning* is basically feature engineering." — Dr. Andrew Ng
- An interesting machine learning jargon in Chinese: "特徵沒選好，調參調到老"。

Andrew Ng (born 1976)
Chinese: 吳恩達
Wikipedia

Feature Engineering

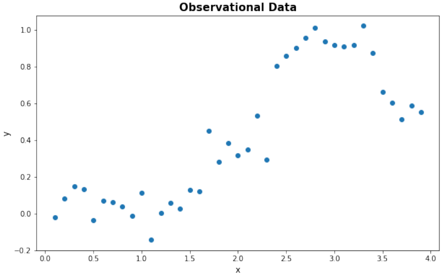
In this lecture, we focus on the feature engineering methods that transform a continuous variable to multiple bases in order to better capture the **nonlinear** patterns. In particular, we study the following two scenarios:

- Nonparametric regression for curve fitting problem
  - Polynomial bases (also log, polar, etc.)
  - Piecewise polynomials and B-Splines
- Binning techniques for logistic regression
  - Top-down splitting by FICO Information Value
  - Bottom-up merging by ChiMerge Algorithm

Feature engineering would increase the signal strengths and allow for more sophisticated modeling, e.g. in the generalized additive models (GAM).

Curve Fitting Problem

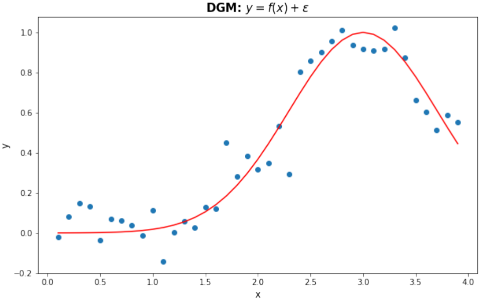
- Suppose we are given a dataset with (x_i, y_i) observed from a signal plus noise model
$$y_i = f(x_i) + \varepsilon_i$$
where $f(x)$ is the underlying true function and the noise $\varepsilon_i \sim N(0, \sigma^2)$.



- We want to estimate $f(x)$ by data modeling. This is an inverse problem.

Data Generating Mechanism

- Assume the true signal $f(x) = e^{-(x-3)^2}$, add random noise $N(0, 0.1^2)$ to generate the data; use a random seed for ensuring reproducibility.



- Such ground truth $f(x)$ is unknown while we conduct the data modeling.

Basis Expansion

- **Basis expansion** is a popular approach to feature engineering. It is a simple extension of linear models to capture nonlinearity.
- It is to transform the raw features with new representations $\{\phi_j(x)\}_{[p]}$ through certain basis functions. Then, predict the response by

$$f(x) \approx \sum_{j=1}^{p} \beta_j \phi_j(x) = \boldsymbol{\phi}(x)^T \boldsymbol{\beta}$$

- This reduces to the linear modeling with least squares solution:

$$\min_{\boldsymbol{\beta}} \sum_{i=1}^{n} \left[y_i - \boldsymbol{\phi}(x)^T \boldsymbol{\beta}\right]^2 = (\boldsymbol{y} - \boldsymbol{\Phi}\boldsymbol{\beta})^T (\boldsymbol{y} - \boldsymbol{\Phi}\boldsymbol{\beta})$$

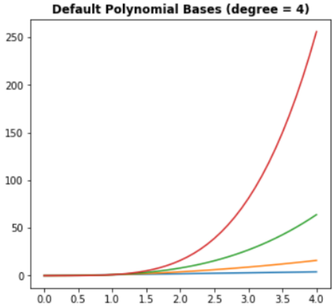
$$\Rightarrow \hat{\boldsymbol{\beta}} = (\boldsymbol{\Phi}^T \boldsymbol{\Phi})^{-1} \boldsymbol{\Phi}^T \boldsymbol{y}$$

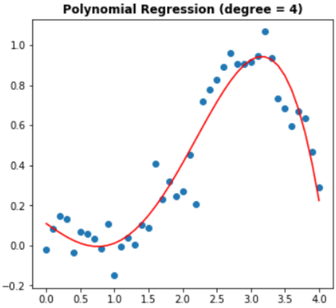
- **Questions:** a) What type of basis functions? b) How many of them?

Table of Contents

Polynomial Regression

- Use the default polynomials of different degrees as the basis functions, then substitute them as the design matrix for linear modeling

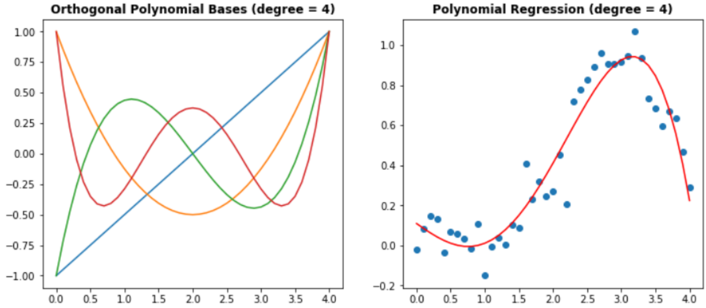





```
Correlation Matrix:
 [[1.     0.9668 0.9141 0.8633]
 [0.9668 1.     0.9859 0.9581]
 [0.9141 0.9859 1.     0.9921]
 [0.8633 0.9581 0.9921 1.    ]]
```

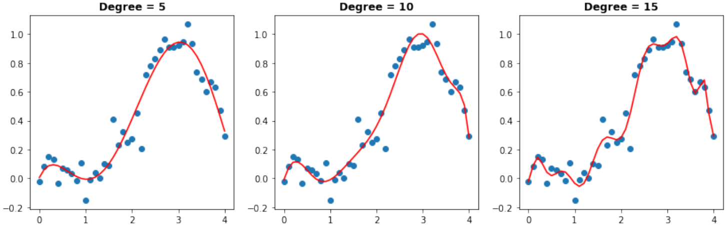
Polynomial Regression

- Use the **orthogonal polynomials** as the basis functions, in order to reduce the feature correlation (Wikipedia:Legendre Polynomials)




```
Correlation Matrix:
 [[1.     0.     0.107  0.    ]
 [0.     1.     0.     0.1523]
 [0.107  0.     1.     0.    ]
 [0.     0.1523 0.     1.    ]]
```

Polynomial Regression



- Lower order polynomials capture the global behavior (low-frequency, or long-term trends)
- Higher order polynomials capture the local behavior (high-frequency, or short-term trends)
- Polynomial regression usually fits poorly near the endpoints (so-called boundary effect)

Table of Contents

Piecewise Linear Bases

- Divide the interval of interest $[a, b]$ into $K + 1$ disjoint subintervals:

$$a = \tau_0 < \tau_1 < \cdots < \tau_K < \tau_{K+1} = b$$

based on the knots $\{\tau_k\}$ for $k = 1, \ldots, K$.

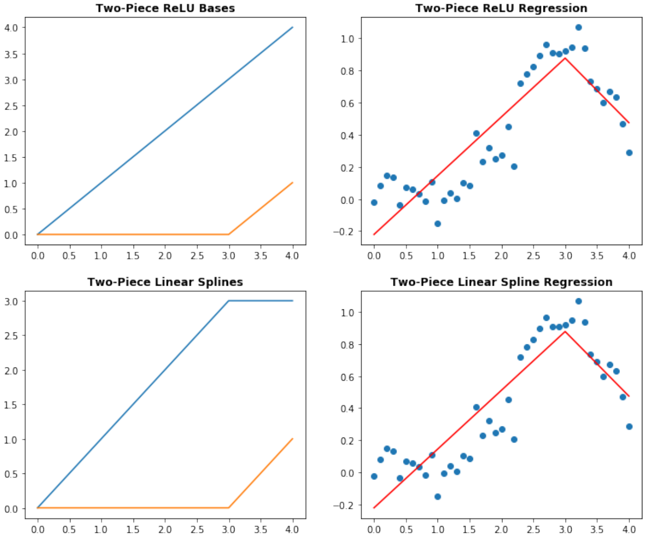
- For each subinterval define a piecewise basis function of the ReLU type,

$$\phi_k(x) = \begin{cases} x - \tau_k, & \text{if } x \geq \tau_k \\ 0, & \text{o.w.} \end{cases}$$

or the flattened type

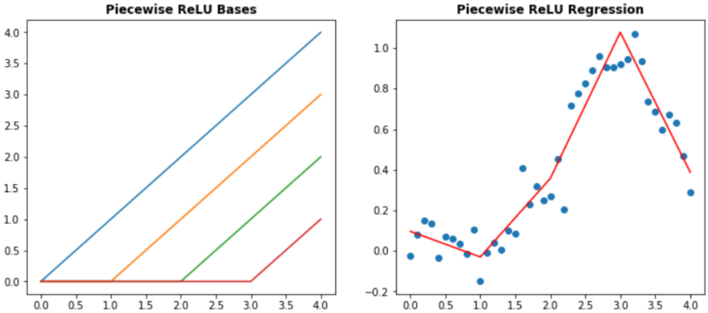
$$\phi_k(x) = \begin{cases} \tau_{k+1} - \tau_k & \text{if } x \geq \tau_{k+1} \\ x - \tau_k, & \text{if } \tau_k < x \leq \tau_{k+1} \\ 0, & \text{o.w.} \end{cases}$$

Two-piece Linear Regression



Piecewise Linear Regression

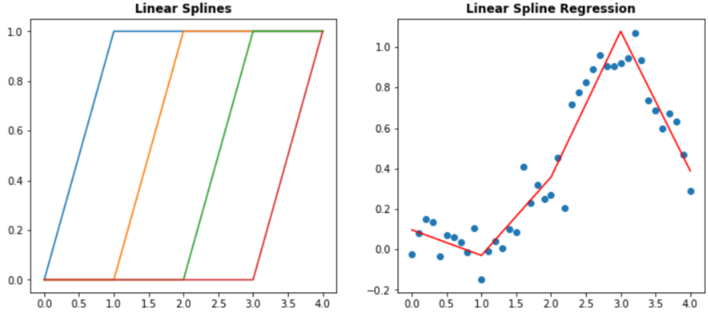
- Use the ReLU type of piecewise linear bases:




```
Correlation Matrix:
 [[1.     0.9808 0.8943 0.6996]
 [0.9808 1.     0.9449 0.755 ]
 [0.8943 0.9449 1.     0.8742]
 [0.6996 0.755  0.8742 1.    ]]
```

Piecewise Linear Regression

- Use the flattened type of piecewise linear bases:




```
Correlation Matrix:
 [[1.     0.686  0.417  0.2371]
 [0.686  1.     0.7334 0.417 ]
 [0.417  0.7334 1.     0.686 ]
 [0.2371 0.417  0.686  1.    ]]
```

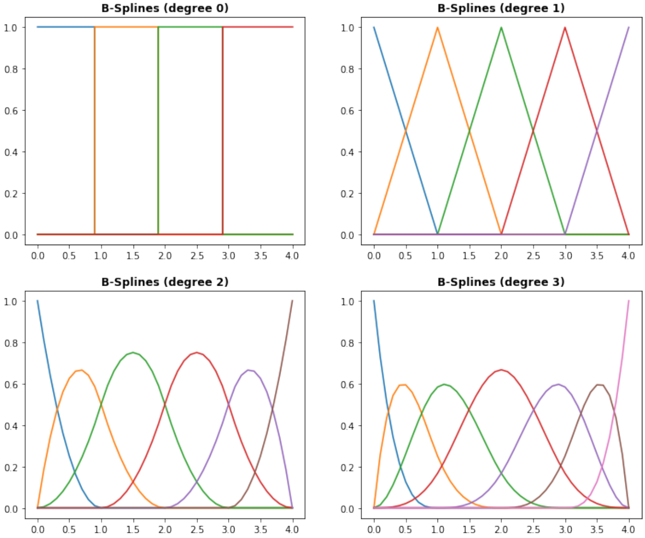
B-Splines

- **B-splines** extend from piecewise linear bases to higher order piecewise polynomials (De Boor,1978)
- B-spline basis functions of degree q are defined for $k = 1, \ldots, K + q + 1$ recursively by

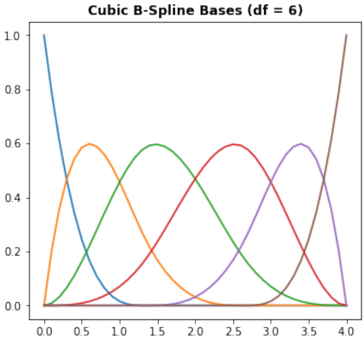
$$B_{k,q}(x) = \frac{x - \tau_k}{\tau_{k+q} - \tau_k} B_{k,q-1}(x) + \frac{\tau_{k+q+1} - x}{\tau_{k+q+1} - \tau_{k+1}} B_{k+1,q-1}(x),$$

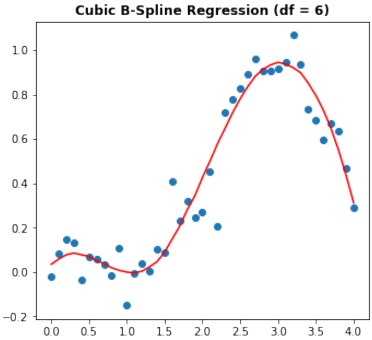
  with the initialized Haar basis functions for $q = 0$: $B_{k,1} = 1_{\{\tau_k \leq x < \tau_{k+1}\}}$.
- Nice localized property: $B_{k,q}(x)$ is non-zero over $[\tau_k, \tau_{k+q+1}]$.
- Check more details at Wikipedia: B-spline

B-Spline Bases



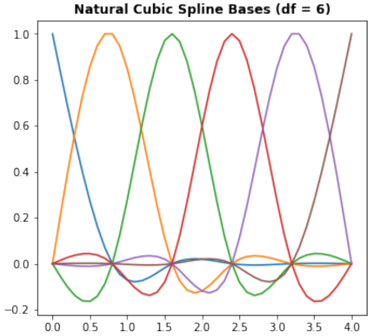
Cubic B-Spline Regression





Natural Cubic Spline Regression

- Impose natural boundary conditions to force the linear polynomial functions beyond the boundary (roughly speaking).
- Natural cubic splines often have superior smoothing performances.



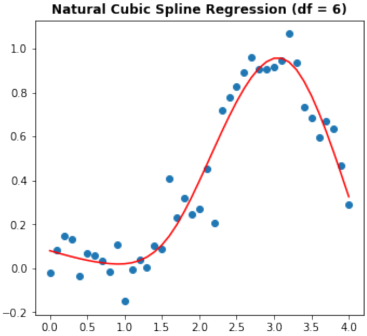
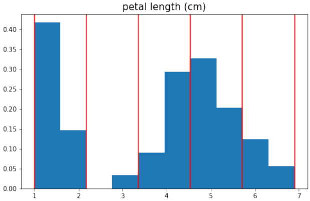
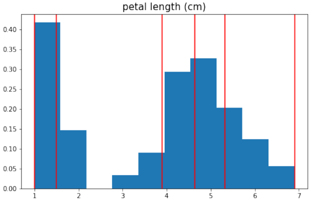


Table of Contents

Unsupervised Binning

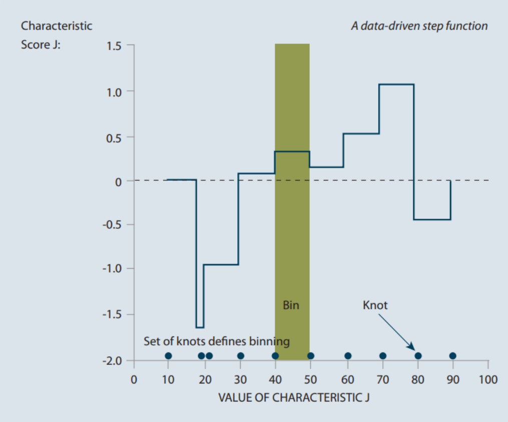
- **Equal Width Binning**: Each bin has identical width.
- **Equal Frequency Binning**: Each bin has the same number of samples (i.e. percentile binning).





- More interested: supervised binning for logistic regression in particular.

Binning Technique by FICO ScoreCard



Source: FICO Model Builder (White Paper)

FICO white paper: "Building Powerful, Predictive Scorecards"

IV Binning for Binary Responses

- Suppose the feature vector is partitioned into K bins. Let p_{1k} and p_{0k} denote the event and non-event percentages in the kth bin.
- **Weight of Evidence (WOE)**: The WOE of kth bin is given by

$$\mathrm{WOE}_k = \log\left(\frac{p_{0k}}{p_{1k}}\right), \quad k = 1, \ldots, K.$$

- **Information Value (IV)**: the feature importance is measured by

$$\mathrm{IV} = \sum_{k=1}^{K}(p_{0k} - p_{1k})\mathrm{WOE}_k = \sum_{k=1}^{K}(p_{0k} - p_{1k})\log\left(\frac{p_{0k}}{p_{1k}}\right).$$

- This is a top-down splitting procedure.

IV Binning for Binary Reponses

- Rules of thumb:

IV	Feature Predictiveness
< 0.02	Not useful for prediction
0.02 to 0.1	Weak predictive power
0.1 to 0.3	Medium predictive power
> 0.3	Strong predictive power

- **IV binning** for partitioning a variable is performed by building a decision tree through maximizing the IV gain.
- Refer to this blog and this package for more details.

ChiMerge Binning for Binary Responses

- **ChiMerge Algorithm**:
  1. Partition the input range into several initial intervals such that each sample finds its own interval.
  2. Compute χ^2 value for every pair of adjacent intervals.
  3. Merge the pair with the smallest χ^2.
  4. Repeat steps 1 – 3 until the χ^2 values of all adjacent pairs exceed a certain threshold. That is, all adjacent pairs are significantly different in terms of χ^2 independence test.
- The threshold is typically chosen as the $\chi^2_{1,1-\alpha}$ with significance level α for binary labeled data.
- This is a bottom-up merging procedure.

ChiMerge Binning for Binary Responses

- The formula of χ^2 value is given by

$$\chi^2 = \sum_{i=1}^{m} \sum_{j=1}^{k} \frac{(A_{ij} - E_{ij})^2}{E_{ij}},$$

where

- $m = 2$ as adjacent intervals are considered.
- $k = 2$ for the binary labeled data.
- A_{ij}: The number of samples in the ith interval and the jth class.
- E_{ij}: Let C_j denote total number of samples in the jth class, $N = \sum_{j=1}^{K} C_j$, and N_i denote the number of samples in the ith interval. $E_{ij} = N_i \frac{C_j}{N}$.

Thank You!

Q&A or Email ajzhang@umich.edu