Erasmus
School of
Economics

Supervised Machine Learning
Week 3

Patrick J.F. Groenen

2020-2021

Erasmus University Rotterdam

Erasmus

Table of Contents

1. Introduction
2. Basis Expansion 1: Interaction Effects
3. Basis Expansion 2: Polynomial
4. Basis Expansion 3: Categorical Predictors
5. Basis Expansion 4: Kernels
6. Thursday Meeting
7. Basis Expansion 5: Splines
8. Bias-Variance Trade-Off
9. Summary and Exercise

Table of Contents

1. Introduction
2. Basis Expansion 1: Interaction Effects
3. Basis Expansion 2: Polynomial
4. Basis Expansion 3: Categorical Predictors
5. Basis Expansion 4: Kernels
6. Thursday Meeting
7. Basis Expansion 5: Splines
8. Bias-Variance Trade-Off
9. Summary and Exercise

Summary

Summary:

Week	Topics	Material
1	Introduction; Introduction to R; Linear methods for regression, model selection, and assessment	3.1, 3.2, 3.3, Xiong (2014)
2	Regularized regression and k-fold cross validation	3.4.1-3.4.3, 3.8.4, 7.10
3	Basis function expansions, kernels, bias-variance trade-off	5.1-5.2.1, 5.8, 7.3
4	Support vector machines	Groenen, Nalbantov, Bioch (2009); 12.1-12.3
5	Classification and regression trees, random forests, bootstrap	7.11, 9.2, 15
6	Boosting	10

Introduction

Material this lecture:

Topic		To read
1.	Basis function expansions	5.1
2.	Kernels	5.8
3.	Splines	5.2-5.2.1
4.	Bias-variance trade-off	7.3

Introduction

- Key idea basis function expansions:
  map vector $\mathbf{x}_i \in \mathbb{R}^p$ to a higher dimensional q vector $\mathbf{b}_i \in \mathbb{R}^q$.
- Examples of basis function expansion:
  1. Interaction effects
  2. Polynomial basis expansion
  3. Categorical predictors
  4. Kernels (must have a ridge penalty)
  5. Splines (piecewise polynomials)
- Basis function expansion is linear in the space of the basis $\mathbf{B}$ and nonlinear in the original space of $\mathbf{X}$.
- Can be used to make any linear model nonlinear through a preprocessing step (except kernels that require a ridge penalty).

Table of Contents

1. Introduction
2. Basis Expansion 1: Interaction Effects
3. Basis Expansion 2: Polynomial
4. Basis Expansion 3: Categorical Predictors
5. Basis Expansion 4: Kernels
6. Thursday Meeting
7. Basis Expansion 5: Splines
8. Bias-Variance Trade-Off
9. Summary and Exercise

Basis Expansion 1: Interaction Effects

- Consider advertising data set and the model

$$\texttt{Sales} = \beta_0 + \beta_1 \texttt{TV} + \beta_2 \texttt{Radio} + \epsilon$$

- An interaction effect occurs when there is synergy on `Sales` when increasing both `TV` and `Radio` simultaneously.
- Formalization

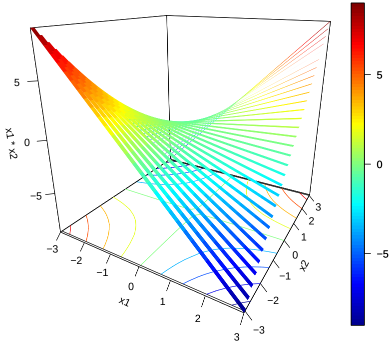
$$Y = \underbrace{\beta_0}_{\text{Intercept}} + \underbrace{\beta_1 X_1 + \beta_2 X_2}_{\text{Main}} + \underbrace{\beta_3 X_1 X_2}_{\text{Interaction}} + \epsilon$$

- Example

$$\texttt{Sales} = \beta_0 + \beta_1 \texttt{TV} + \beta_2 \texttt{Radio} + \beta_3 \texttt{TV} \times \texttt{Radio} + \epsilon$$

Basis Expansion 1: Interaction Effects

Example of interaction effect $X_1 \times X_2$:



Basis Expansion 1: Interaction Effects

- Interpretation interaction effect with $\beta_3 > 0$:
  - for larger TV and larger Radio budgets $\Longrightarrow$ more Sales
  - for smaller TV and smaller Radio budgets $\Longrightarrow$ more Sales
  - for larger TV and smaller Radio budgets $\Longrightarrow$ less Sales
  - for smaller TV and larger Radio budgets $\Longrightarrow$ less Sales
- Interpretation interaction effect with $\beta_3 < 0$:
  Interpretation on Sales reverses.
- Always also model main effects of the interaction effect variables.
- Even with interaction effects it remains a linear model:
  consider the TV $\times$ Radio just as a third predictor variable.

$$\texttt{Sales} = \beta_0 + \beta_1 \texttt{TV} + \beta_2 \texttt{Radio} + \beta_3 \texttt{TV} \times \texttt{Radio} + \epsilon$$

Basis Expansion 1: Interaction Effects

Example of interaction effect `TV` × `Radio` for advertising data:

```
R> ## Interaction effects
R> load("Advertising.Rdata")  # Load the Advertsing data set
R>
R> # head() shows the first 6 rows of a matrix
R> # model.matrix() constructs the design matrix from a formula
R>
R> head(model.matrix( ~ TV + Radio + TV*Radio, data = Advertising))
  (Intercept)    TV Radio TV:Radio
1           1 230.1  37.8     8698
2           1  44.5  39.3     1749
3           1  17.2  45.9      789
4           1 151.5  41.3     6257
5           1 180.8  10.8     1953
6           1   8.7  48.9      425
```

Basis Expansion 1: Interaction Effects

Example of interaction effect `TV` × `Radio` for advertising data:

```
R> ## Fit model with interaction
R> result <- lm(Sales ~ TV + Radio + TV*Radio, data = Advertising)
R> summary(result)

Call:
lm(formula = Sales ~ TV + Radio + TV * Radio, data = Advertising)

Residuals:
   Min     1Q Median     3Q    Max
-6.337 -0.403  0.183  0.595  1.525

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 6.75e+00   2.48e-01   27.23   <2e-16 ***
TV          1.91e-02   1.50e-03   12.70   <2e-16 ***
Radio       2.89e-02   8.91e-03    3.24   0.0014 **
TV:Radio    1.09e-03   5.24e-05   20.73   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.944 on 196 degrees of freedom
Multiple R-squared:  0.968,Adjusted R-squared:  0.967
F-statistic: 1.96e+03 on 3 and 196 DF,  p-value: <2e-16
```

Basis Expansion 1: Interaction Effects

Example of interaction effect `TV` $\times$ `Radio` for advertising data:

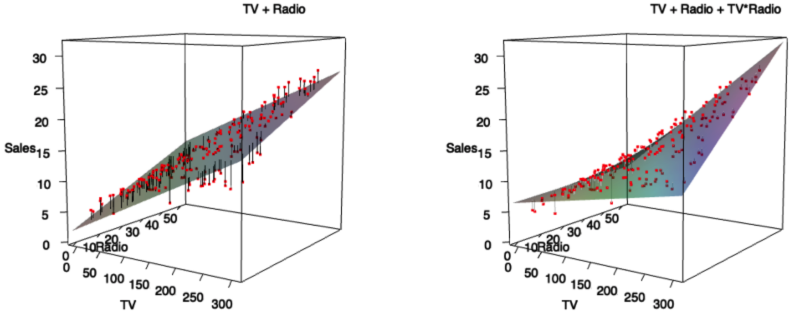


Table of Contents

1. Introduction
2. Basis Expansion 1: Interaction Effects
3. Basis Expansion 2: Polynomial
4. Basis Expansion 3: Categorical Predictors
5. Basis Expansion 4: Kernels
6. Thursday Meeting
7. Basis Expansion 5: Splines
8. Bias-Variance Trade-Off
9. Summary and Exercise

Basis Expansion 2: Polynomial

- Relations between response variable Y and predictors X may be nonlinear.
- Simple trick to fit nonlinear effects by polynomial regression: add powers of the a predictor X to the model

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_1^2 + \beta_3 X_1^3 + \ldots + \beta_p X_1^p + \epsilon$$

- Auto data example: predict miles per gallon `mpg` by `horsepower`

$$\texttt{mpg} = \beta_0 + \beta_1 \texttt{horsepower} + \beta_2 \texttt{horsepower}^2 + \epsilon$$

Basis Expansion 2: Polynomial

Example of polynomial regression of degree 2:

```
R> # The I() below means: include a new variable consisting of this function of th
R> head(model.matrix( ~ horsepower + I(horsepower^2), Auto))
  (Intercept) horsepower I(horsepower^2)
1           1        130           16900
2           1        165           27225
3           1        150           22500
4           1        150           22500
5           1        140           19600
6           1        198           39204
```

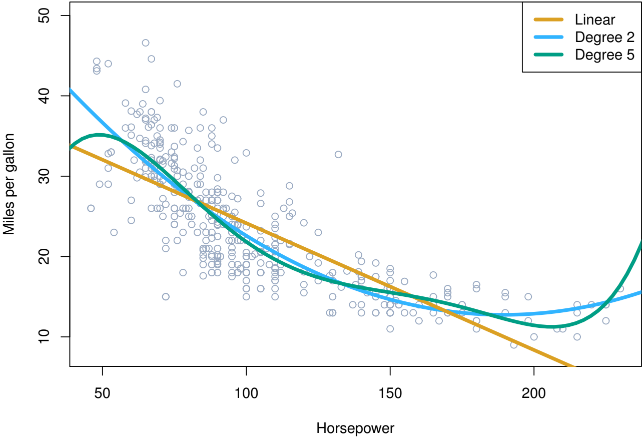
Basis Expansion 2: Polynomial

Example of polynomial regression of degree 2:

```
R> result <- lm(mpg ~ horsepower + I(horsepower^2), Auto) # Fit polynomial regres
R> summary(result)

Call:
lm(formula = mpg ~ horsepower + I(horsepower^2), data = Auto)

Residuals:
    Min      1Q  Median      3Q     Max
-14.714  -2.594  -0.086   2.287  15.896

Coefficients:
                 Estimate Std. Error t value Pr(>|t|)
(Intercept)     56.900100   1.800427    31.6   <2e-16 ***
horsepower      -0.466190   0.031125   -15.0   <2e-16 ***
I(horsepower^2)  0.001231   0.000122    10.1   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 4.37 on 389 degrees of freedom
Multiple R-squared:  0.688,Adjusted R-squared:  0.686
F-statistic:  428 on 2 and 389 DF,  p-value: <2e-16
```

Basis Expansion 2: Polynomial

Example of polynomial regression of degrees 2 and 5:



Basis Expansion 2: Polynomial

Example of polynomial regression of degrees 2 and 5:

```
R> x <- Auto[, "horsepower"]
R> y <- Auto[, "mpg"]
R> result1 <- lm(mpg ~ horsepower, Auto)
R> result2 <- lm(mpg ~ horsepower + I(horsepower^2), Auto)
R> result5 <- lm(mpg ~ poly(horsepower, 5), Auto)
R> yhat.1  <- result1$fitted.values
R> idx     <- order(Auto$horsepower)       # We need to reorder horsepower monotone increasin
R> yhat.2  <- result2$fitted.values
R> yhat.5  <- result5$fitted.values
R> plot(x, y, col = "grey",                # Make color of points grey
        xlab = "Horsepower", ylab = "MPG", # Labels of x-axis and y-axis
        las = 1)                            # Make vertical axis tick labels horizontal
R> # Add lines
R> lines(x, yhat.1,                        # Add the predicted line for simple regression
         col = "orange", lwd = 2)           # Color of line is orange with line width 2 point
R> lines(x[idx], yhat.2[idx],              # Add the predicted line for quadratic regression
         col = "blue", lwd = 2)             # Color of line is blue, line width is 2 points
R> lines(x[idx], yhat.5[idx],              # Add prediction line for pol. regr. of degree 5
         col = "green", lwd = 2)            # Color of line is green, line width is 2 points
R> legend("topright",                  # The position of the legend in the plot
          legend = c("Linear", "Quadratic","Degree 5"), # Text vector of labels
          col = c("orange", "blue", "green"),           # Colors of the lines
          lwd = c(2, 2, 2))                             # Line widths of the lines
```

Basis Expansion 2: Polynomial

Polynomial basis:

- For 4-th degree polynomial, instead of predictor variable x_1 introduce also new predictor variables x_{12}, x_{13}, x_{14} by the polynomial basis matrix

$$\mathbf{B}_1 = \begin{bmatrix} x_{11} & x_{11}^2 & x_{11}^3 & x_{11}^4 \\ x_{12} & x_{12}^2 & x_{12}^3 & x_{12}^4 \\ x_{13} & x_{13}^2 & x_{13}^3 & x_{13}^4 \\ x_{14} & x_{14}^2 & x_{14}^3 & x_{14}^4 \\ x_{15} & x_{15}^2 & x_{15}^3 & x_{15}^4 \end{bmatrix}$$

- Do this for each predictor variable and use the matrix $\mathbf{B} = [\mathbf{B}_1\ \mathbf{B}_2\ \mathbf{B}_3]$ (for the polynomial bases of three original predictor variables).

Table of Contents

1. Introduction
2. Basis Expansion 1: Interaction Effects
3. Basis Expansion 2: Polynomial
4. Basis Expansion 3: Categorical Predictors
5. Basis Expansion 4: Kernels
6. Thursday Meeting
7. Basis Expansion 5: Splines
8. Bias-Variance Trade-Off
9. Summary and Exercise

Basis Expansion 3: Categorical Predictors

- What to do with categorical predictors?
- Standard trick: replace by a set of dummy variables, e.g., predictor `price` (X_1) with levels 'low', 'medium', 'high'.

$$X = \begin{bmatrix} \text{high} \\ \text{high} \\ \text{high} \\ \text{low} \\ \text{low} \\ \text{medium} \end{bmatrix} \Longrightarrow \mathbf{G} = \begin{array}{c} \begin{array}{ccc} \text{high} & \text{low} & \text{medium} \end{array} \\ \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \end{array}$$

Basis Expansion 3: Categorical Predictors

- Problem: we cannot use **G** directly as predictor due to multicollinearity: $g_{i1} = 1 - g_{i2} - g_{i3}$
- Solution: use category 1 (or another) as reference category.
- Model becomes: $\beta_0 + \beta_1 \text{low} + \beta_2 \text{medium}$

$$\begin{array}{ccc} \text{intercept} & \text{low} & \text{medium} \end{array}$$
$$\begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}$$

- Interpretation: β_1 is the contrast effect of category low against high.

Basis Expansion 3: Categorical Predictors

- In R, categorical variables are called factors and the categories are levels:

```
R> # Make factor edu
R> price.ex <-factor(c("high", "high", "high", "low", "low",  "medium"))
R> model.matrix(~ 1 + price.ex)
  (Intercept) price.exlow price.exmedium
1           1           0              0
2           1           0              0
3           1           0              0
4           1           1              0
5           1           1              0
6           1           0              1
attr(,"assign")
[1] 0 1 1
attr(,"contrasts")
attr(,"contrasts")$price.ex
[1] "contr.treatment"
```

Basis Expansion 3: Categorical Predictors

- Analysis of Variance (ANOVA) is multiple regression with categorical predictors
- Useful to determine whether category means differ.
- F-tests are done to simultaneously test:
  H_0: all $\beta_j = 0$ with j referring to the categories of one factor.
  H_a: at least one of the $\beta_j \neq 0$.
- Multiple regression and ANOVA yield exactly the same results (but differ in presentation).
- Model:

$$Y = \beta_0 + \beta_1 \text{Var1.Level2} + \beta_2 \text{Var1.Level3} + \ldots$$

Basis Expansion 3: Categorical Predictors

```
R> load("Credit.RData")
R> # ANOVA:  testing for Gender difference on Balance.
R> result <- aov(Balance ~ Gender, Credit)
R> summary(result)

             Df   Sum Sq Mean Sq F value Pr(>F)
Gender        1    38892   38892    0.18   0.67
Residuals   398 84301020  211812

R> coef(result)

 (Intercept) GenderFemale
       509.8         19.7

R> # The same but now through lm()
R> result <- lm(Balance ~ Gender, Credit)
R> summary(result)


Call:
lm(formula = Balance ~ Gender, data = Credit)

Residuals:
   Min     1Q Median     3Q    Max
-529.5 -455.4  -60.2  334.7 1489.2

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)     509.8       33.1   15.39   <2e-16 ***
GenderFemale     19.7       46.1    0.43     0.67
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 460 on 398 degrees of freedom
Multiple R-squared:  0.000461,Adjusted R-squared:  -0.00205
F-statistic: 0.184 on 1 and 398 DF,  p-value: 0.669

R> coef(result)

 (Intercept) GenderFemale
       509.8         19.7
```

Basis Expansion 3: Categorical Predictors

- Interaction effects make one dummy variable for each combination of categories of two (or more) categorical predictors.
- Example of two-way interaction effects of predictors `education` (X_1) with levels m = middle, h = high, c = college and `gender` (X_2) with levels f = female, m = male.

$$
\begin{array}{cc}
X_1 & X_2
\end{array}
\quad
\begin{array}{ccc|cc|cccccc}
 & X_1 & & & X_2 & & & & X_1X_2 & & \\
c & h & m & f & m & cf & hf & mf & cm & hm & mm
\end{array}
$$

$$
\begin{bmatrix} c & m \\ c & m \\ c & f \\ h & m \\ h & f \\ m & m \end{bmatrix}
\Longrightarrow
\begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}
\begin{bmatrix} 0 & 1 \\ 0 & 1 \\ 1 & 0 \\ 0 & 1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}
\begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}
$$

Basis Expansion 3: Categorical Predictors

- Because of multicollinearity it is enough to fit $(K_1 - 1) \times (K_2 - 1)$ two-way interaction dummy variables.
- Example

$$
\begin{array}{cc}
X_1 & X_2
\end{array}
\quad\quad
\begin{array}{ccccc}
\multicolumn{2}{c}{X_1} & X_2 & \multicolumn{2}{c}{X_1X_2} \\
\mathrm{h} & \mathrm{m} & \mathrm{m} & \mathrm{hm} & \mathrm{mm}
\end{array}
$$

$$
\begin{bmatrix}
\mathrm{c} & \mathrm{m} \\
\mathrm{c} & \mathrm{m} \\
\mathrm{c} & \mathrm{f} \\
\mathrm{h} & \mathrm{m} \\
\mathrm{h} & \mathrm{f} \\
\mathrm{m} & \mathrm{m}
\end{bmatrix}
\Longrightarrow
\begin{bmatrix}
0 & 0 \\
0 & 0 \\
0 & 0 \\
1 & 0 \\
1 & 0 \\
0 & 1
\end{bmatrix}
\begin{bmatrix}
1 \\
1 \\
0 \\
1 \\
0 \\
1
\end{bmatrix}
\begin{bmatrix}
0 & 0 \\
0 & 0 \\
0 & 0 \\
1 & 0 \\
0 & 0 \\
0 & 1
\end{bmatrix}
$$

Basis Expansion 3: Categorical Predictors

- Multiple regression with categorical predictors is also called Analysis of Variance
-

```
R> ## ANOVA (= multiple regression with categorical predictors)
R> result <- aov(Balance ~ Student + Ethnicity + Ethnicity:Student, Credit)
R> summary(result)
                   Df   Sum Sq Mean Sq F value  Pr(>F)
Student             1  5658372 5658372   28.57 1.5e-07 ***
Ethnicity           2    50043   25021    0.13    0.88
Student:Ethnicity   2   599466  299733    1.51    0.22
Residuals         394 78032031  198051
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
R> coef(result)
                 (Intercept)                   StudentYes
                       480.7                        497.9
              EthnicityAsian           EthnicityCaucasian
                       -34.8                         16.4
    StudentYes:EthnicityAsian StudentYes:EthnicityCaucasian
                        23.5                       -247.1
```

Basis Expansion 3: Categorical Predictors

Interpretation can be done in terms of means per category:

```
R> ## Two-way Interaction Plot
R> interaction.plot(Credit$Ethnicity, Credit$Student, Credit$Balance, type = "b", col = c(1:3
      leg.bty = "o",  lwd = 2, pch = c(18, 24, 22),
      xlab = "Ethnicity",  ylab = "Balance", main = "Interaction Plot",
      trace.label = "Student")
```

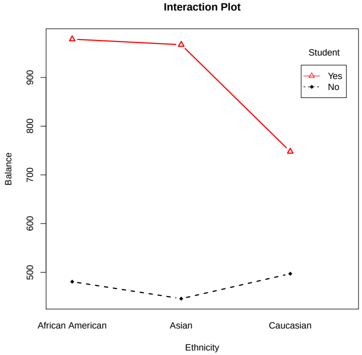


Table of Contents

1. Introduction
2. Basis Expansion 1: Interaction Effects
3. Basis Expansion 2: Polynomial
4. Basis Expansion 3: Categorical Predictors
5. Basis Expansion 4: Kernels
6. Thursday Meeting
7. Basis Expansion 5: Splines
8. Bias-Variance Trade-Off
9. Summary and Exercise

Basis Expansion 4: Kernels

Kernels

- Kernels make use of the same trick polynomial basis expansion and spline transformations.
- Requires a ridge penalty: $\lambda \mathbf{w}^\top \mathbf{w}$, e.g., in kernel ridge regression (KRR) or support vector machines (SVM).
- Maps $\mathbf{x}_i$ (row i of $\mathbf{X}$) to ϕ_i in some high dimensional space.
- Fit the model linearly in the high dimensional space.
- Then, at most $n+1$ parameters need to be optimized through a dual approach.

Basis Expansion 4: Kernels

Ridge regression

- Loss function ridge regression:

$$L_{\text{ridge}}(w_0, \mathbf{w}) \quad = \quad \|\mathbf{y} - (w_0\mathbf{1} + \mathbf{X}\mathbf{w})\|^2 + \lambda \mathbf{w}^\top \mathbf{w}$$

- The vector of predicted values is: $\hat{\mathbf{y}} = \mathbf{q} = w_0\mathbf{1} + \mathbf{X}\mathbf{w}$
- The intercept w_0 complicates things; therefore, we set $\tilde{\mathbf{q}} = \mathbf{X}\mathbf{w}$ so that $\mathbf{q} = w_0\mathbf{1} + \mathbf{X}\mathbf{w} = w_0\mathbf{1} + \tilde{\mathbf{q}}$

Basis Expansion 4: Kernels

A dual approach for KRR:

- Basic idea of the dual approach:
  If $p \gg n$ (and $\mathbf{X}$ has rank n), then switch to the minimization over $\mathbf{q}$ (n parameters) instead of w_0 and $\mathbf{w}$ ($p+1$ parameters)

Basis Expansion 4: Kernels

Towards a dual approach:

- Example of an $\mathbf{X}$ with $n < p$: $n = 2, p = 3$

$$\mathbf{X} = \begin{bmatrix} -.25 & .75 & .50 \\ .50 & .50 & .50 \end{bmatrix}$$

- Choose (e.g.)

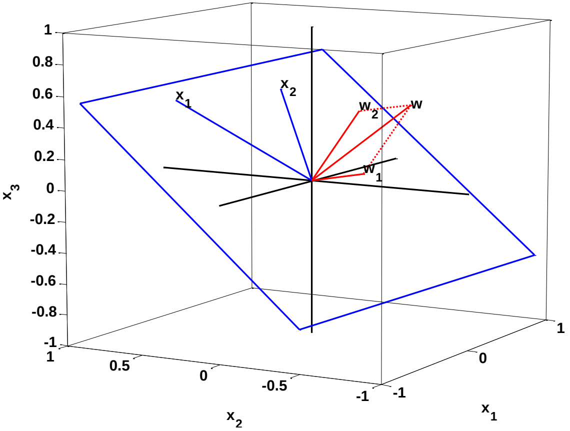
$$\mathbf{w} = \begin{bmatrix} .25 \\ -.50 \\ .50 \end{bmatrix}$$

- Then, the $n \times 1 = 2 \times 1$ vector $\tilde{\mathbf{q}}$ must be in the linear space spanned by $\mathbf{x}_1$ and $\mathbf{x}_2$

$$\tilde{\mathbf{q}} = \mathbf{X}\mathbf{w} = \mathbf{X}\mathbf{w}_1 = \begin{bmatrix} -.1875 \\ .1250 \end{bmatrix}$$

Basis Expansion 4: Kernelss

Towards a dual approach:



Basis Expansion 4: Kernels

Steps to arrive at a dual ridge regression formulation:

1. Decompose $\mathbf{w} = \mathbf{w}_1 + \mathbf{w}_2$ with a part that is in the linear space of $\mathbf{X}$ ($\mathbf{w}_1$) and a part that is orthogonal to the linear space of $\mathbf{X}$ ($\mathbf{w}_2$).
2. $\tilde{\mathbf{q}}$ depends only on $\mathbf{w}_1$ and not on $\mathbf{w}_2$.
3. Penalty term has $\lambda \mathbf{w}^\top \mathbf{w} = \lambda \mathbf{w}_1^\top \mathbf{w}_1$ because $\mathbf{w}_2^\top \mathbf{w}_2 = 0$.
4. Penalty term equals $\lambda \mathbf{w}^\top \mathbf{w} = \lambda \tilde{\mathbf{q}}^\top (\mathbf{X}\mathbf{X}^\top)^{-1} \tilde{\mathbf{q}}$ where the $n \times n$ matrix $\mathbf{X}\mathbf{X}^\top$ has elements $\mathbf{x}_i^\top \mathbf{x}_{i'}$.
5. Without loss of generality, we may optimize directly over the n parameters $\tilde{q}_i$ without any restriction.
6. $L_{\text{ridge}}(w_0, \tilde{\mathbf{q}})$ is now only a function of w_0 and $\tilde{q}_i$.

Basis Expansion 4: Kernels

- L_{ridge} is now only a function of w_0 and $\tilde{q}_i$:

$$L_{\text{ridge}}(w_0, \tilde{\mathbf{q}}) = \underbrace{\|\mathbf{y} - (w_0\mathbf{1} + \tilde{\mathbf{q}})\|^2}_{\boxed{\text{Regression term}}} + \underbrace{\lambda\tilde{\mathbf{q}}^\top(\mathbf{X}\mathbf{X}^\top)^{-1}\tilde{\mathbf{q}}}_{\boxed{\text{Penalty term}}}$$

- Proof that $\lambda\mathbf{w}^\top\mathbf{w} = \lambda\tilde{\mathbf{q}}^\top(\mathbf{X}\mathbf{X}^\top)^{-1}\tilde{\mathbf{q}}$:

Basis Expansion 4: Kernels

Computation of dual ridge regression minimizing $L_{\text{ridge}}(w_0, \tilde{\mathbf{q}})$:

- To be able to separate estimation of intercept w_0 and $\tilde{\mathbf{q}}$, we set $\tilde{\mathbf{X}} = \mathbf{J}\mathbf{X}$ with $\mathbf{J} = \mathbf{I} - n^{-1}\mathbf{1}\mathbf{1}^\top$.
- Then $L_{\text{ridge}}(w_0, \tilde{\mathbf{q}}) = \|\mathbf{y} - w_0\mathbf{1}\|^2 + \|\mathbf{J}\mathbf{y} - \tilde{\mathbf{q}}\|^2 + \lambda\tilde{\mathbf{q}}(\mathbf{X}\mathbf{X}^\top)^{-1}\tilde{\mathbf{q}}$.
- Optimal $w_0 = n^{-1}\mathbf{1}^\top\mathbf{y}$.
- Optimal $\tilde{\mathbf{q}} = \left(\mathbf{I} + \lambda(\mathbf{X}\mathbf{X}^\top)^{-1}\right)^{-1}\mathbf{J}\mathbf{y}$.
- If $p \gg n$, this update is quite fast.
- if n is not too large, then following computation is faster:
  - Compute the eigendecomposition $\mathbf{X}\mathbf{X}^\top = \mathbf{U}\mathbf{D}^2\mathbf{U}^\top$.
  - $\tilde{\mathbf{q}} = \mathbf{U}\left(\mathbf{I} + \lambda\mathbf{D}^{-2}\right)^{-1}\mathbf{U}^\top\mathbf{J}\mathbf{y}$

    where the diagonal matrix $\left(\mathbf{I} + \lambda(\mathbf{D})^{-2}\right)^{-1}$ has diagonal elements $d_{ii}^2/(d_{ii}^2 + \lambda)^{-1}$.

Basis Expansion 4: Kernels

Kernels for nonlinear prediction:

- Kernels make use of same dual trick for $p \gg n$.
- Replace the all the variables in $\mathbf{X}$ by their $n \times k$ kernel basis $\mathbf{\Phi}(\mathbf{X})$ or $\mathbf{\Phi}$ for short.
- The equivalent of matrix $\mathbf{X}\mathbf{X}^\top$ becomes the $n \times n$ kernel matrix $\mathbf{K} = \mathbf{\Phi}\mathbf{\Phi}^\top$ with elements $k_{ii'} = \phi_i^\top \phi_{i'}$
- Kernel trick: choose smart $\mathbf{\Phi}$ such that k_{ij} can be directly computed from rows $\mathbf{x}_i$ and $\mathbf{x}_{i'}$.
- Kernel ridge regression loss equals:

$$L_{\text{KRR}}(w_0, \tilde{\mathbf{q}}) = \|\mathbf{y} - (w_0 \mathbf{1} + \tilde{\mathbf{q}})\|^2 + \lambda \tilde{\mathbf{q}}^\top \mathbf{K}^{-1} \tilde{\mathbf{q}}$$

Basis Expansion 4: Kernels

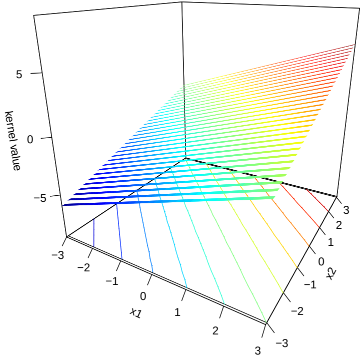
We discuss three kernels:

1. Linear kernel.
2. Radial basis function (RBF) or Gaussian kernel.
3. Inhomogeneous polynomial kernel.
4. Several other kernels exist.

Basis Expansion 4: Kernels

The linear kernel:

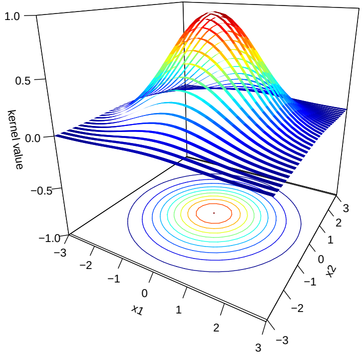
- Choose $\mathbf{K} = \mathbf{X}\mathbf{X}^\top$, thus $k_{ii'} = \mathbf{x}_i^\top \mathbf{x}_{i'}$.
- Exactly the same as linear ridge regression.



Basis Expansion 4: Kernels

The radial basis function (RBF) or Gaussian kernel:

- Choose $k_{ii'} = e^{-\gamma \|\mathbf{x}_i - \mathbf{x}_{i'}\|^2}$ for some $\gamma > 0$ (fixed).
- For $\gamma = (2\sigma)^{-1}$ the RBF and Gaussian kernels are the same.



Basis Expansion 4: Kernels

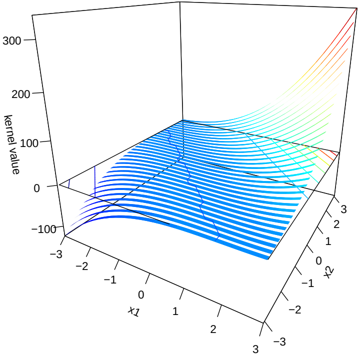
The radial basis function (RBF) or Gaussian kernel:

- For large γ, $k_{ii'} \to 0$ for $i \neq i'$ and $k_{ii} = 1$.
- For small $\gamma \downarrow 0$, $k_{ii'} \to 1$.
- Use K-fold cross validation to determine hyper parameters λ and possibly γ.
- Good choice for fixing: $\gamma = 1/m$ if predictors in $\mathbf{X}$ are z-scores.

Basis Expansion 4: Kernels

The inhomogeneous polynomial kernel:

- Choose $k_{ii'} = (1 + \mathbf{x}_i^\top \mathbf{x}_{i'})^d$ for some degree $d > 0$ (fixed).
- For $d = 1$ the inhomogeneous polynomial kernel is the same as the linear kernel.



Basis Expansion 4: Kernels

Kernel types:

- Not so clear what kernel to choose.
- Radial basis function seems powerful and often used.
- General strategy: try several kernels and choose the one with the best test set classification.
- Kernels are sensistive to standardisation of predictor variables:
  - Change all $\mathbf{x}_j$ to be z-scores.
  - Change all $\mathbf{x}_j$ to be have range between 0 and 1.
    (With many zeros, the $\mathbf{X}$ becomes sparse, computations can be accelerated, and big data are possible.)

Basis Expansion 4: Kernels

Final step needed with kernels for predicting the test data (unseen data) $\mathbf{X}_u$:

- Map $n \times p$ training data $\mathbf{X}$ to $\mathbf{\Phi}$ so that $\mathbf{K} = \mathbf{\Phi}\mathbf{\Phi}^\top$.
- Map $n_u \times p$ test data matrix $\mathbf{X}_u$ to $\mathbf{\Phi}_u$.
- The goal is to find $\mathbf{q}_u = w_0\mathbf{1} + \mathbf{\Phi}_u\mathbf{w}$.
- When using kernels, it is often not possible to compute $\mathbf{\Phi}_u$ and $\mathbf{w}$ but we do have

$$\tilde{\mathbf{q}} \quad = \quad \mathbf{\Phi}\mathbf{w}.$$

Basis Expansion 4: Kernels

Final step needed with kernels for predicting the test data $\mathbf{X}_u$:

- Let the SVD of $\mathbf{\Phi} = \mathbf{UDV}^\top$. Then $\mathbf{\Phi}^\top(\mathbf{\Phi\Phi}^\top)^{-1}\mathbf{\Phi} = \mathbf{VV}^\top$ because

$$\begin{aligned}\mathbf{\Phi}^\top(\mathbf{\Phi\Phi}^\top)^{-1}\mathbf{\Phi} &= \mathbf{VDU}^\top(\mathbf{UDV}^\top\mathbf{VDU}^\top)^{-1}\mathbf{UDV}^\top \\ &= \mathbf{VDU}^\top\mathbf{UD}^{-2}\mathbf{U}^\top\mathbf{UDV}^\top \\ &= \mathbf{VDD}^{-2}\mathbf{DV}^\top = \mathbf{VV}^\top\end{aligned}$$

- Then the predicted $\mathbf{q}_u$ for the test set $\mathbf{X}_u$ is

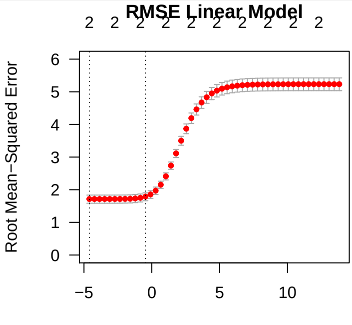
$$\begin{aligned}\mathbf{q}_u = w_0\mathbf{1} + \mathbf{\Phi}_u\mathbf{w} &= w_0\mathbf{1} + \mathbf{\Phi}_u\mathbf{VV}^\top\mathbf{w} \\ &= w_0\mathbf{1} + \mathbf{\Phi}_u\mathbf{\Phi}^\top(\mathbf{\Phi\Phi}^\top)^{-1}\mathbf{\Phi w} \\ &= w_0\mathbf{1} + (\mathbf{\Phi}_u\mathbf{\Phi}^\top)(\mathbf{\Phi\Phi}^\top)^{-1}(\mathbf{\Phi w}) \\ &= w_0\mathbf{1} + \mathbf{K}_u\mathbf{K}^{-1}\tilde{\mathbf{q}}\end{aligned}$$

  with $\mathbf{K}_u$ is the $n_u \times n$ kernel matrix with elements k_{ij} where i stands for row i of $\mathbf{X}_u$ and j for row j of $\mathbf{X}$.

Basis Expansion 4: Kernels

Comparing RMSE for Linear and RBF KRR models for Advertising data:

```
R> load("Advertising.RData")
R> y.resp <- y <- as.vector(Advertising$Sales)      # y variable
R> X <- model.matrix(Sales ~ TV + Radio, data = Advertising)  # Predictor variables (as a matrix, not dataframe)
R> X <- scale(X[, 2:3])                              # Make columns z-scores
R> ## Linear model
R> lin.cv <- cv.glmnet(X, y, alpha = 0, lambda = 10^seq(-2, 6, length.out = 50),
                           standardize = FALSE)      # Ridge regression (alpha must be 0 for ridge)
R> lin.cv$cvm  <- lin.cv$cvm^0.5; lin.cv$cvup <- lin.cv$cvup^0.5; lin.cv$cvlo <- lin.cv$cvlo^0.5
R>
R> ## Fit RBF KRR model through dsmle package
R> ker.cv <- cv.krr(y.resp, X, kernel.type = "nonhompolynom")
R> # Plot RMSE for Linear and RBF KRR models
R> op <- par(mfrow = c(1, 2))
R> plot(lin.cv, ylab = "Root Mean-Squared Error", ylim = c(0, 6), las = 1, main = "RMSE Linear Model")
R> plot(ker.cv, ylim = c(0, 6))
R> par(op)
```



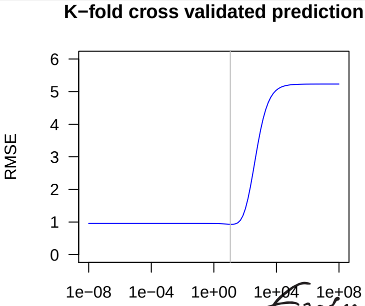


Table of Contents

1. Introduction
2. Basis Expansion 1: Interaction Effects
3. Basis Expansion 2: Polynomial
4. Basis Expansion 3: Categorical Predictors
5. Basis Expansion 4: Kernels
6. Thursday Meeting
7. Basis Expansion 5: Splines
8. Bias-Variance Trade-Off
9. Summary and Exercise

Thursday Meeting

Schedule for Thursday November 12, 2020, topic of Week 2

	Team				
Team Task	1	2	3	5	6
Presentation methods, results and interpretation		+			
Discussion methods,			+		
Discussion results and interpretation				+	
Presentation code					+
Discussion code	+				

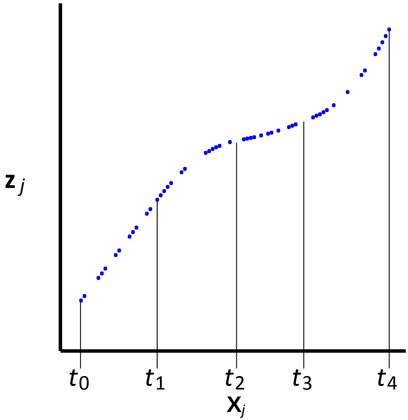
- Discussions address three items:
  - what you think was good;
  - possibly address issues that were unclear to you;
  - suggestions of issues that you think could be improved.

Table of Contents

1. Introduction
2. Basis Expansion 1: Interaction Effects
3. Basis Expansion 2: Polynomial
4. Basis Expansion 3: Categorical Predictors
5. Basis Expansion 4: Kernels
6. Thursday Meeting
7. Basis Expansion 5: Splines
8. Bias-Variance Trade-Off
9. Summary and Exercise

Basis Expansion 5: Splines

Example of an I-Spline transformation $\mathbf{z}_j$ of predictor variable $\mathbf{x}_j$:



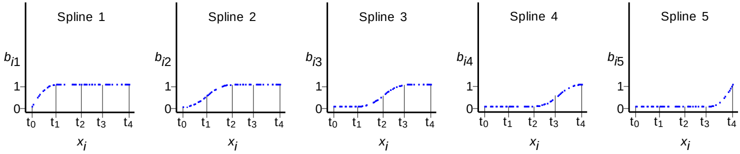
Basis Expansion 5: Splines

Properties of an I-Spline transformation of predictor variable $\mathbf{x}_j$:

- The range of the variable is partitioned in adjacent intervals.
- Each interval has approximately equal number of observations.
- Each interval as a polynomial transformation of degree d.
- Adjacent intervals are smoothly connected.
- Special case: with $k = 0$ interior knots, then I-Spline is equal to polynomial regression of order d.

Basis Expansion 5: Splines

- Given the variable $\mathbf{x}$, the degree d and the number of interior knots k, an explicit $n \times (d+k)$ matrix with the spline basis $\mathbf{B}$ can be computed.
- Example of columns of $\mathbf{B}$ for $k = 3$ interior knots and degree $d = 2$:



- Every linear combination gives a smooth transformation:

$$z_i = b_{i1}w_1 + b_{i2}w_2 + b_{i3}w_3 + b_{i4}w_4 + b_{i5}w_5$$

- For I-Splines only: if all $w_j \geq 0$ then the transformation from $\mathbf{x}$ to $\mathbf{z}$ is monotone increasing

Basis Expansion 5: Splines

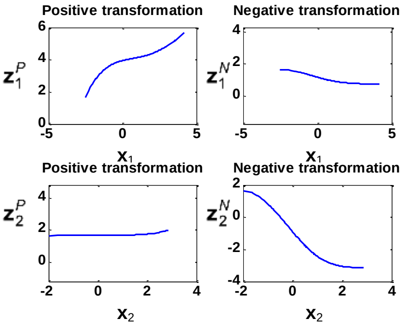
Thus, the steps for introducing nonlinearity using splines are:

- Map each variable into an $d + k$ dimensional space.
- The original m variables are mapped into an $m(d + k)$ dimensional (feature) space.
- Then, a linear model is applied with this high dimensional space.
- Easy handling of a test point x_{ti} for given w_js: each interval is a polynomial function of the original variable $\mathbf{x}$.

Basis Expansion 5: Splines

Interpreting the I-spline transformations:

- For each variable, make a transformation plot using only:
  - the positive weights in w_j (thus $z_i^P = \sum_j b_{ij} w_j^P$)
  - the negative weights in w_j (thus $z_i^N = \sum_j b_{ij} w_j^N$)
- $\mathbf{z}^P$ is monotone increasing with $\mathbf{x}$ and $\mathbf{z}^N$ is monotone decreasing with $\mathbf{x}$.



Basis Expansion 5: Splines

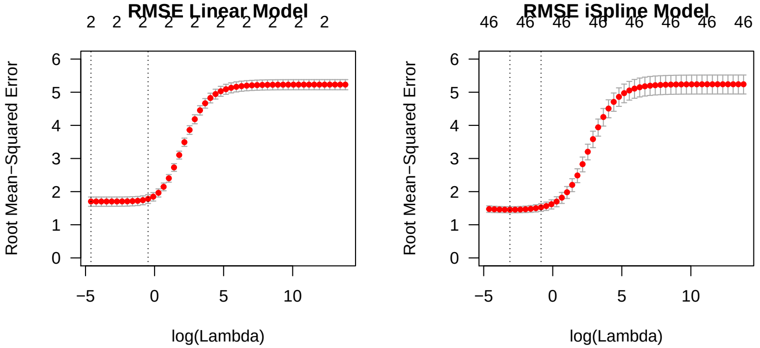
Comparing Linear with I-Spline model for Advertising data:

```
R> ## Set up iSpline with interior knots of based on deciles
R> nknots <- 19; degree <- 5
R> knots <- apply(X, 2, FUN = function(x)
     quantile(x, seq(1/(nknots + 1), 1 - 1/(nknots + 1), by = 1/(nknots + 1))) )
R> X.iSpline.list <- list()
R> X.iSpline.list[[1]] <- X.iSpline <- iSpline(X[, 1], knots = knots[, 1], degree = degree - 1)
R> for (j in 2:ncol(X)){
     X.iSpline.list[[j]] <- iSpline(X[, j], knots = knots[, j], degree = degree - 1)
     X.iSpline <- cbind(X.iSpline, X.iSpline.list[[j]])
   }
R> spl.cv <- cv.glmnet(X.iSpline, y, alpha = 0, lambda = 10^seq(-2, 6, length.out = 50),
                          standardize = FALSE)          # Ridge regression (alpha must be 0 for ridge)
R> spl.cv$cvm  <- spl.cv$cvm^0.5; spl.cv$cvup <- spl.cv$cvup^0.5; spl.cv$cvlo <- spl.cv$cvlo^0.5
```

Basis Expansion 5: Splines

Comparing RMSE for Linear and I-Spline models for Advertising data:

```
R> ## Plot RMSE for Linear and iSpline models
R> op <- par(mfrow = c(1, 2))
R> plot(lin.cv, ylab = "Root Mean-Squared Error", ylim = c(0, 6), las = 1, main = "RMSE Linear Model")
R> plot(spl.cv, ylab = "Root Mean-Squared Error", ylim = c(0, 6), las = 1, main = "RMSE iSpline Model")
R> par(op)
```



Basis Expansion 5: Splines

Comparing regression surfaces for Linear with I-Spline model for Advertising data:

```
R> ## Show regression surface for Linear and iSpline models
R> ## options(rgl.useNULL = TRUE, rgl.printRglwidget = TRUE)
R> source("plot.surface.R")
R> plot.surface.init()
R> mfrow3d(1, 2, sharedMouse = TRUE)
R> plot.surface(coef(lin.cv, s = "lambda.min"), X, y.resp)
R> next3d()
R> plot.surface(coef(spl.cv, s = "lambda.min"), X, y.resp, X.iSpline.list)
R> mfrow3d(1, 1)
```

Basis Expansion 5: Splines

Comparing regression surfaces for Linear with I-Spline model for Advertising data:

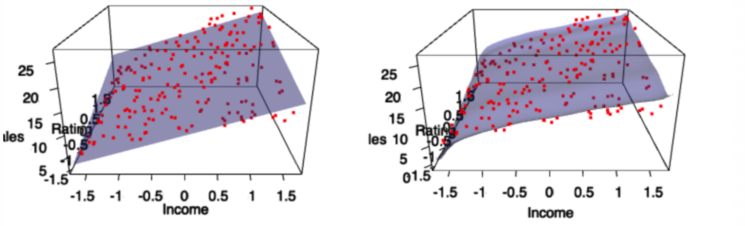


Table of Contents

1. Introduction
2. Basis Expansion 1: Interaction Effects
3. Basis Expansion 2: Polynomial
4. Basis Expansion 3: Categorical Predictors
5. Basis Expansion 4: Kernels
6. Thursday Meeting
7. Basis Expansion 5: Splines
8. Bias-Variance Trade-Off
9. Summary and Exercise

Bias-Variance

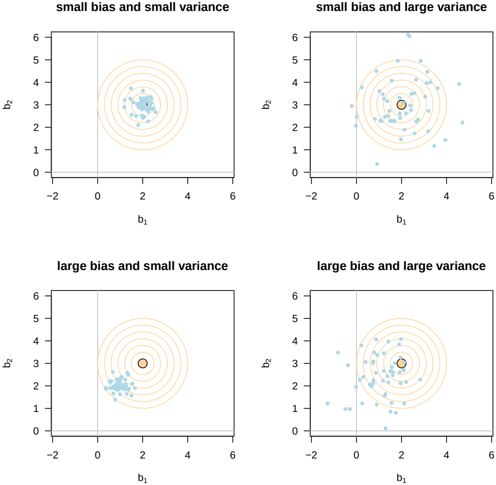
- Bias: systematic difference between the true population parameter β and the (expected) estimator b:

$$\text{Bias} = E(b) - \beta$$

- Variance: measure of spread (uncertainty) of an estimated parameter, for regression: $\text{Var}(b_j)$
- If p approaches n (and assuming that the variance of the error stays constant) then the variance for b_j becomes larger: overfitting.

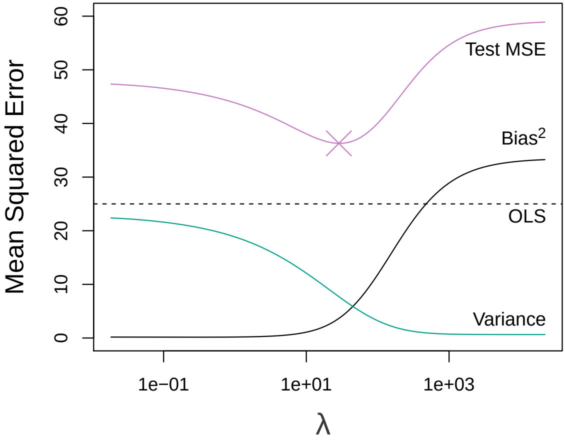
Bias-Variance

Four cases of bias and variance of two parameters (e.g., b_1 and b_2):



Bias-Variance

Effect on bias-variance trade-off by ridge regression (simulated data with known population weights β_j):



Bias-Variance

Goal penalty methods (such as ridge and lasso regression):

Introduce bias by shrinkage of the b_js to reduce the variance such that test MSE becomes as small as possible

Bias-Variance

Bias-variance trade-off:

- Unbiased estimators are expected to recover the true population parameter
- But unbiased estimators can have a huge variance
  - Example 1: in multiple regression almost multicollinearity.
  - Example 2: in multiple regression with many predictors.
- General solution: reduce variance at the cost of introducing bias.
- Effect of penalty term in Ridge and LASSO regression: shrinkage which causes bias and reduces variance.

Bias-Variance

Bias-variance trade-off:

- Assume $y = f(\mathbf{x}) + \epsilon$ with $E(\epsilon) = 0$ and $E(\epsilon^2) = \sigma_\epsilon^2$,
- where $E(.)$ stands for expectation (the value that would happen after many repetitions),
- the true predictor function $f = f(\mathbf{x})$,
- the estimated predictor function $\hat{f} = \hat{f}(\mathbf{x})$, for example, $\hat{f} = \mathbf{x}^\top \mathbf{b}$.

Bias-Variance

Bias-variance trade-off:

- The expected test mean squared error (test MSE) of a prediction for an observation i is:

$$\underbrace{E(y-\hat{f}(\mathbf{x}))^2}_{\text{Expected test MSE}} = \underbrace{\text{Bias}[\hat{f}(\mathbf{x})]^2}_{\text{Bias}^2} + \underbrace{\text{Var}[\hat{f}(\mathbf{x})]}_{\text{Variance}} + \underbrace{\sigma_\epsilon^2}_{\text{Irreducible error}}$$

- $\text{Bias}[\hat{f}(\mathbf{x})] = E[\hat{f}(\mathbf{x})] - E[f(\mathbf{x})]$,
- $\text{Var}[\hat{f}(\mathbf{x})] = E[\hat{f}(\mathbf{x})^2] - E[\hat{f}(\mathbf{x})]^2$.

Bias-Variance

Proof of Bias-variance trade-off:

$$E(y-\hat{f})^2 \quad = \quad \text{Bias}[\hat{f}]^2 \quad + \quad \text{Var}[\hat{f}] \quad + \quad \sigma_\epsilon^2$$

with $y = f + \epsilon$.

Table of Contents

1. Introduction
2. Basis Expansion 1: Interaction Effects
3. Basis Expansion 2: Polynomial
4. Basis Expansion 3: Categorical Predictors
5. Basis Expansion 4: Kernels
6. Thursday Meeting
7. Basis Expansion 5: Splines
8. Bias-Variance Trade-Off
9. Summary and Exercise

Summary and Exercise

Summary:

Week	Topics	Material
1	Introduction; Introduction to R; Linear methods for regression, model selection, and assessment	3.1, 3.2, 3.3, Xiong (2014)
2	Regularized regression and k-fold cross validation	3.4.1-3.4.3, 3.8.4, 7.10
3	Basis function expansions, kernels, bias-variance trade-off	5.1-5.2.1, 5.8, 7.3
4	Support vector machines	Groenen, Nalbantov, Bioch (2009); 12.1-12.3
5	Classification and regression trees, random forests, bootstrap	7.11, 9.2, 15
6	Boosting	10

To Do for Next Time

To Do for Next Time:

- This is an exercise that is not graded (but you will get feedback).
- Try to predict `output` in `Airline.RData` through kernel ridge regression using the other variables as predictors. An explanation of the variables is given in the `Ecdat`-package.
- Write your own R-function for KRR provided in the slides.
- Try at least the following two kernels: the radial basis function (RBF) and nonhomegenous polynomial kernels.
- You can compare your results with the `krr()` function of the `dsmle`-package (stand-alone package, see canvas) and explain briefly whether or not they are the same and why this is so.
- Upload your code as R file and as pdf.

Acknowledgement

Some of the figures in this presentation are taken from "An Introduction to Statistical Learning, with applications in R" (Springer, 2013) with permission from the authors: G. James, D. Witten, T. Hastie and R. Tibshirani