Dimension Reduction with Derived Features

PCA, SVM, Ridge Regression

DS 6030 | Fall 2021

penalized.pdf

Contents

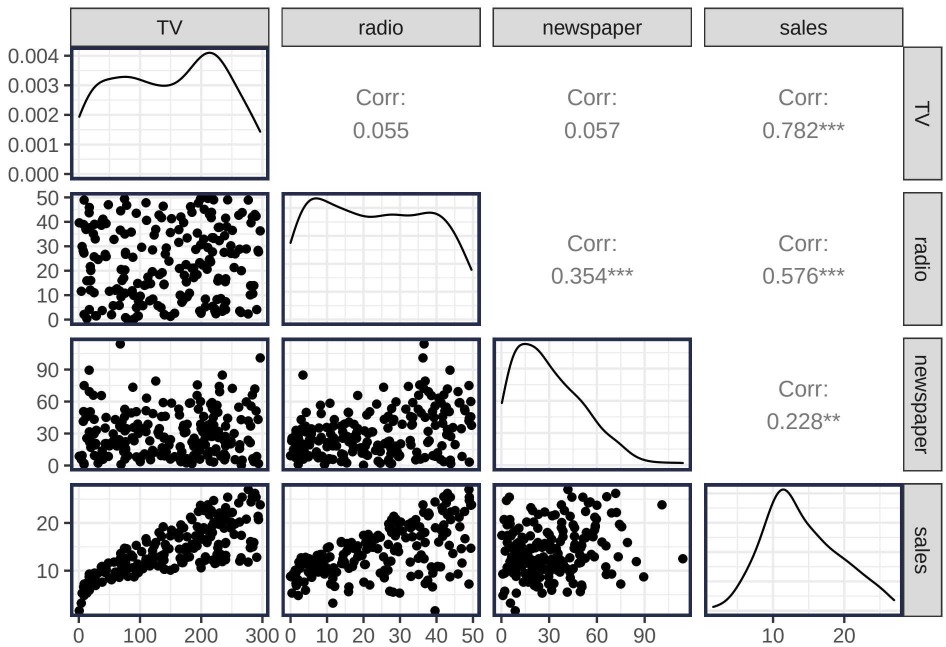
1 Data

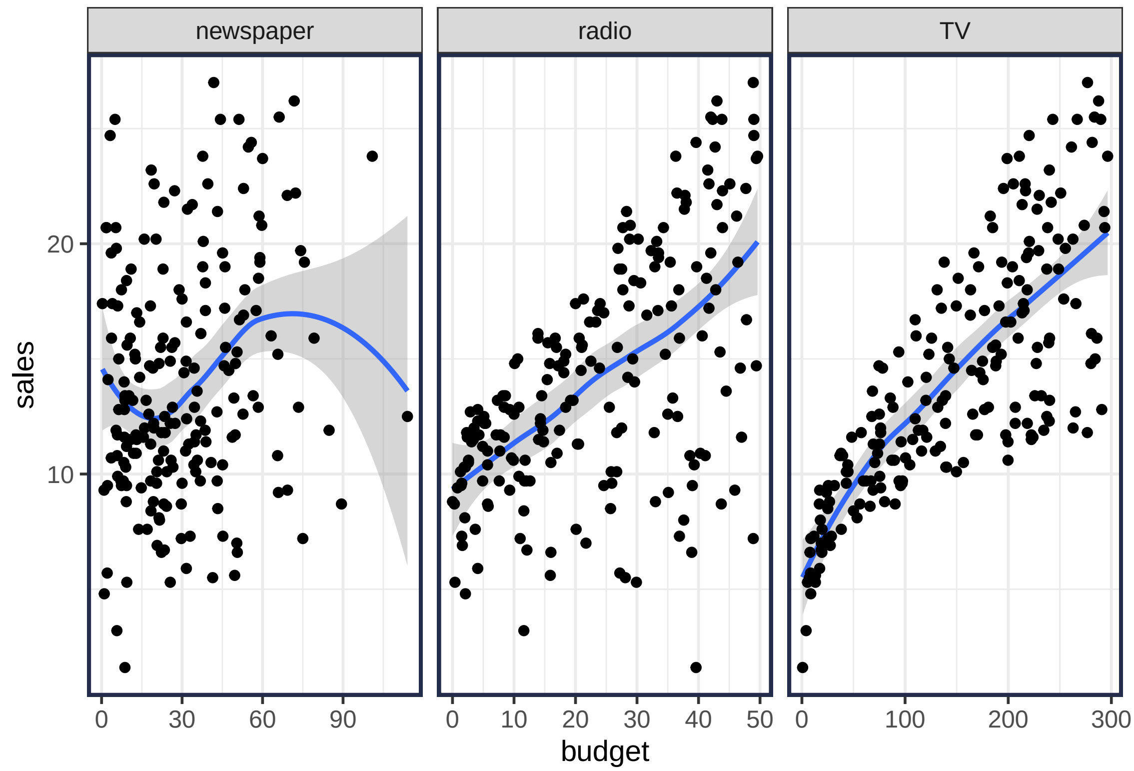
1.1 Advertising Data

The Introduction to Statistical Learning (ISL) text has some data on advertising.

These data give the sales of a product (in thousands of units) under advertising budgets (in thousands of dollars) of TV, radio, and newspaper.

The goal is to predict `sales` for a given `TV`, `radio`, and `newspaper` budget.





1.2 Linear Regression (OLS)

The standard generic form for a linear regression model is

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots, + \beta_p X_p + \epsilon$$

- Y is the response or dependent variable
- $X_1, X_2, \ldots, X_p$ are called the p explanatory, independent, or predictor variables
- the greek letter ϵ (epsilon) is the random error variable
- For example:

$$\text{sales} = \beta_0 + \beta_1 \times (\text{TV}) + \beta_2 \times (\text{radio}) + \beta_3 \times (\text{newspaper}) + \text{error}$$

Training data is used to estimate the *model parameters* or *coefficients*.

$$\left[\begin{array}{cccccc|c} x_{11} & x_{12} & \cdot & \cdot & \cdot & x_{1p} & y_1 \\ x_{21} & x_{22} & \cdot & \cdot & \cdot & x_{2p} & y_2 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ x_{n1} & x_{n2} & \cdot & \cdot & \cdot & x_{np} & y_n \end{array}\right]$$

Producing the predictive model:

$$\hat{y}(x_1, x_2, \ldots, x_p) = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2 + \ldots, + \hat{\beta}_p x_p$$

- where $\hat{\beta}_j$ are the weights assigned to each variable

- these weights are the values that minimize the residual sum of squares (RSS) for predicting the training data
- For example:

$$\widehat{\text{sales}} = 2.939 + 0.046 \times (\text{TV}) + 0.189 \times (\text{radio}) \times -0.001 \times (\text{newspaper})$$

- The *complexity* of an OLS regression model is the *number of estimated parameters*
  - it is $p + 1$ (using the notation above), where the $+1$ is added for the intercept.

1.3 Estimation

- The weights/coefficients (β) are the *model parameters*
- OLS uses the weights/coefficients that minimize the RSS loss function over the training data

$$\begin{aligned} \hat{\beta} &= \arg\min_{\beta}\ \text{RSS}(\beta) \qquad \text{Note: } \beta \text{ is a } \textit{vector} \\ &= \arg\min_{\beta}\ \sum_{i=1}^{n}(y_i - f(\mathbf{x}_i;\beta))^2 \\ &= \arg\min_{\beta}\ \sum_{i=1}^{n}(y_i - \beta_0 - \beta_1 x_{i1} - \beta_2 x_{i2} + \ldots + \beta_p x_{ip})^2 \end{aligned}$$

OLS equivalently minimizes the MSE since MSE $=$ RSS$/n$.

1.3.1 Matrix notation

$$f(\mathbf{x};\beta) = \mathbf{x}^{\mathsf{T}}\beta$$

$$Y = \begin{bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_n \end{bmatrix} \qquad X = \begin{bmatrix} 1 & X_{11} & X_{12} & X_{13} & \ldots & X_{1p} \\ 1 & X_{21} & X_{22} & X_{23} & \ldots & X_{2p} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & X_{n1} & X_{n2} & X_{n3} & \ldots & X_{np} \end{bmatrix} \qquad \beta = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_p \end{bmatrix}$$

$$\text{RSS}(\beta) = (\text{Y} - \text{X}\beta)^{\mathsf{T}}(\text{Y} - \text{X}\beta)$$

$$\begin{aligned} \frac{\partial \text{RSS}(\beta)}{\partial \beta} &= 2X^{\mathsf{T}}(Y - X\beta) \\ &\implies X^{\mathsf{T}}Y = X^{\mathsf{T}}X\beta \\ &\implies \boxed{\hat{\beta} = (X^{\mathsf{T}}X)^{-1}X^{\mathsf{T}}Y} \end{aligned}$$

1.3.2 OLS in R with `lm()`

```
#-- Fit OLS
prostate.lm = lm(lpsa~., data=prostate.train)
#> Error in is.data.frame(data): object 'prostate.train' not found
prostate.lm %>% broom::tidy()
#> Error in broom::tidy(.): object 'prostate.lm' not found
```

1.4 Some Problems with least squares estimates

There are a few problems with using least squares estimation (OLS) to estimate the regression parameters (coefficients)

- *Prediction Accuracy*
  - the least squares estimates in high dimensional data may have low bias but can suffer from large variance.
  - Prediction accuracy can sometimes be improved by shrinking or setting some coefficients to zero.
  - By doing so we sacrifice a little bit of bias to reduce the variance of the predicted values, and hence may improve the overall prediction accuracy.
  - Some predictors may not have any predictive value and only increase noise
- *Interpretation*: With a large number of predictors, we often would like to determine a smaller subset that exhibit the strongest effects. In order to get the "big picture", we are willing to sacrifice some of the small details
  - When $p > n$ least squares won't work at all

1.5 Improving Least squares

We will examine 3 standard approaches to improve on least squares estimates

1. Subset Selection
   - Only use a subset of predictors, but estimate with OLS
   - Examples: *best subsets*, *forward step-wise*
2. Shrinkage/Penalized/Regularized Regression
   - Instead of an "all or nothing" approach, shrinkage methods force the coefficients closer toward 0.
   - Examples: *ridge*, *lasso*, *elastic net*

3. Dimension Reduction with Derived Inputs

- Use a subset of linearly transformed predictors
  - Examples: *PCA*, *PLS*

All three methods introduce some additional bias in order to reduce variance and *hopefully* improve prediction.

2 Derived Linear Features

Instead of using the raw features as predictors, it can sometimes be helpful to use derived features (e.g., new features as transformations of the raw features).

- X is the $(n \times p)$ raw predictor matrix
  - p predictors
- Z is the $(n \times r)$ derived predictor matrix
  - r predictors
  - r could be less than (*dimension reduction*) , equal to, or greater than p (*feature expansion*)
- We saw *feature expansion* (i.e., basis expansion) when we used splines to allow non-linear relationship between outcome and single predictor

- Today's material is more focused on *dimension reduction* ($r < p$) as a way to introduce some bias to reduce variance
  - Just like we did with penalized regression (e.g., ridge, lasso, elasticnet)

2.1 Linear Transformations

- Let $Z = XA$ be the $(n \times r)$ transformed model matrix
  - X is the $(n \times p)$ original features
  - A is the $(p \times r)$ linear transformation matrix
  - A_j is the *j*th column of A
  - a_{jm} is the (j, m) element of A

$$\begin{aligned} Z &= XA \\ Z_m &= XA_j \\ &= \sum_{j=1}^{p} X_j a_{jm} \\ Z_{im} &= \sum_{j=1}^{p} X_{ij} a_{jm} \end{aligned}$$

2.2 OLS with derived feature model

- Once we have the new feature matrix Z, we can estimate parameters like usual. For example, with OLS:

$$\boxed{\hat{\theta} = (Z^{\mathsf{T}} Z)^{-1} Z^{\mathsf{T}} Y}$$

This gives predictions:

$$\hat{y}_i = \hat{\theta}_0 + \sum_{m=1}^{r} Z_{im} \hat{\theta}_m$$

Plugging in $Z_{im} = \sum_{j=1}^{p} X_{ij} a_{jm}$:

Estimated Derived Beta Coefficients

2.3 Dimension Reduction vs. Feature Selection

If $r < p$, then fewer model parameters need to be estimated. This is called *dimension reduction* since we have less parameters to estimate.

- Hence, edf is decreased (lower variance, higher bias)

However, because we still use all original features we haven't actually done *feature selection*, so all raw features must still be collected.

3 Principal Component Regression (PCR)

3.1 Eigen Decomposition (Spectral Analysis)

3.2 Principal Component Analysis (PCA)

3.3 Dimension Reduction with PCR

4 Singular Value Decomposition (SVD)

5 PCA with SVD

6 Ridge Regression with SVD

7 Comparison