Aarhus University

Decision Support Systems

PROJECT REPORT

Author:

Lasse Lildholdt
(201507170)
Stinus Skovgaard
(201507170)
Daniel Tøttrup
(201507170)
Johan Vasegaard
(201507170)
Frederik Madsen
(201504477)

Supervisor:

Christian Fischer Pedersen

10. maj 2020

© 2020 - All Rights Reserved

Indhold

Figurer

Kapitel 1

INTRODUCTION

In this project report, the reader will be presented with problem solutions for the course Decision support systems. Throughout the solution the reader will achieve knowledge on several different subject within the main area. Each topic will be presented with the theory along with solutions for appropiate exercise to validate the presented theory.

The main topics which will be handled in this report will be as follows:

- Simple linear regression / Multiple linear regression
- Logisitic regression / Linear discriminant analysis
- Cross validation / Bootstrap
- Subset selection
- Shrinkage methods / DImension reduction methods

Kapitel 2

SIMPLE LINEAR REGRESSION / MULTIPLE LINEAR REGRESSION

The simple Linear Regression approach is a quick and simple method for predicting a response Y based on X. The linear model is used to give an idea of the relationship between to dataset. For this to be true an assumption that the two variables have a linear relationship is needed. The mathematical representation of this can be seen below.

$$Y \approx \beta_0 + \beta_1 * X \tag{2.1}$$

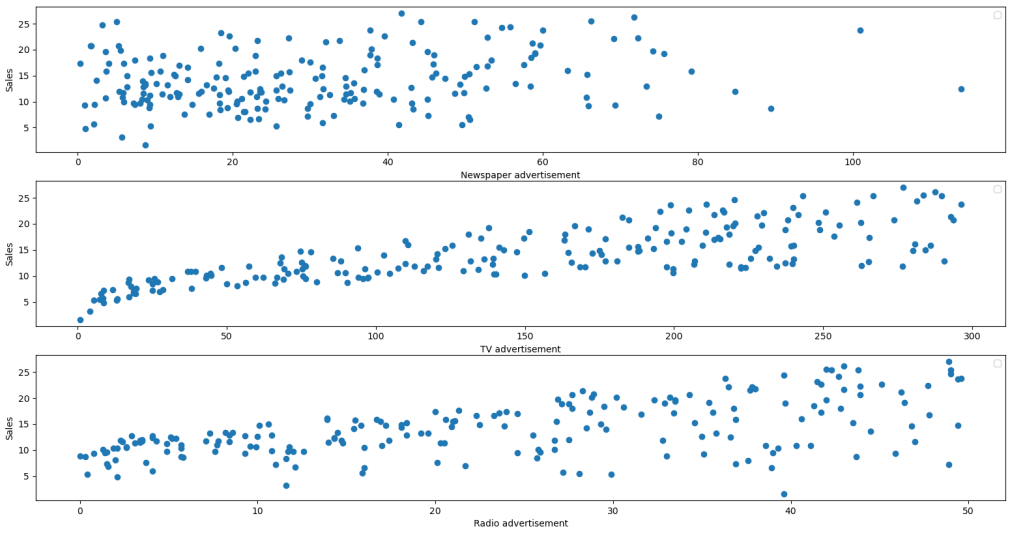
eq. (2.1) can also be seen as "Regressing Y onto X". As an example the dataset Advertising.csv contains sales on a certain product and advertisement money spent on certain media platforms. X represents TV advertising and Y represents sales. It is the possible to regress sales onto TV. This is expressed as:

$$sales \approx \beta_0 + \beta_1 * TV \tag{2.2}$$

The two constants β_0 and β_1 represents the intercept (Where it intercepts the y-axis) and slope of the linear model. These constant needs to be predicted and when they have we will end out with a linear model that fits our data.

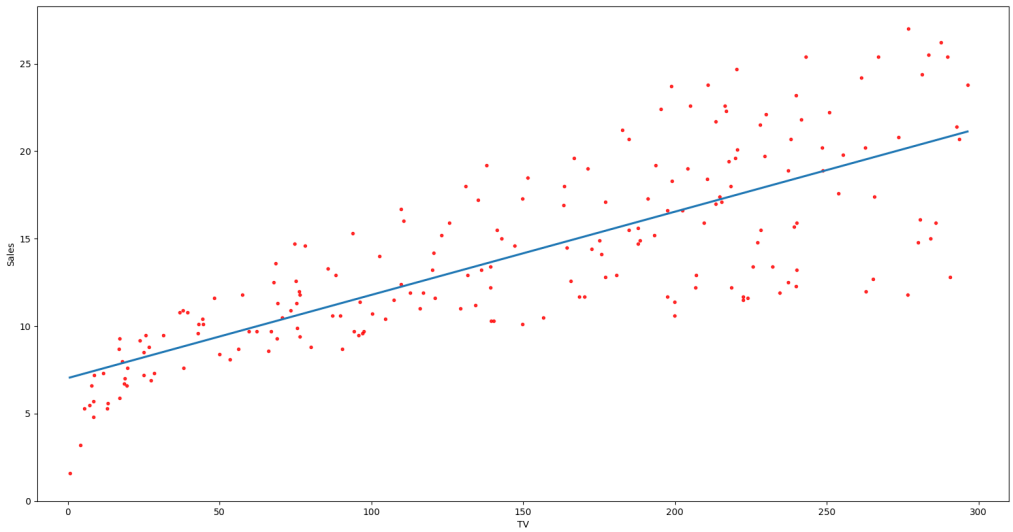
2.1 Predict β_0 and β_1

In the Advertising.csv dataset a number of observations (n = 200) have been made on amount of advertisements for tv, radio, newspaper and the corresponding sales. The advertisement data can be seen on fig. 2.1.



Figur 2.1: Advertisement data

To estimate β_0 and β_1 we need to introduce RSS which is the *residual sum of squares*. RSS is the difference or error between observed response value and predicted response value that is predicted by our linear model. On fig. 2.2 a linear model is put on top of the TV advertisement data. The RSS is the summed value of the difference between the red points to the blue line. We want this value to b as low as possible to get the most accurate model.



Figur 2.2: Linear regression on TV advertisement

We can define the RSS value as:

$$\begin{aligned} RSS &= e_1^2 + e_2^2 ... e_n^2 \\ &where \\ e &= y_i - \hat{y} \end{aligned} \tag{2.3}$$

This means that to get to a final method we need to minimize RSS. Some calculus shows that these minimizers are:

$$\begin{aligned} \hat{\beta}_1 &= \frac{\sum_{i=1}^{n}(x_i - \overline{x})(y_i - \overline{y})}{\sum_{i=1}^{n}(x_i - \overline{x})^2} \\ \beta_0 &= \overline{y} - \beta_1 \overline{x} \\ &\textbf{where} \\ \overline{y} &= \frac{1}{n}\sum_{i=1}^{n} y_i \\ &\textbf{and} \\ \overline{x} &= \frac{1}{n}\sum_{i=1}^{n} x_i \end{aligned} \tag{2.4}$$

This method of estimating the constants have been used on fig. 2.2. How this is done can be seen on the code snippet below.

```
x_mean = get_mean(x_data)
y_mean = get_mean(y_data)

num = 0
den = 0

for i in range(len(x_data)):
num += (x_data[i]-x_mean)*(y_data[i]-y_mean)
den += (x_data[i] - x_mean) ** 2

b1_hat = num/den
b0_hat = y_mean-b1_hat*x_mean

return b0_hat, b1_hat
```

β_0 and β_1 is estimated to 7.03 and 0.04 respectively for the TV advertisement data.

2.2 Multiple linear regression

Simple linear regression is a great to see the response of a single predictor variable. However there are times where it would be beneficial to have multiple predictor variables.

In the advertisement data we have three predictor variables; TV, Radio and Newspaper. We can define this as the following:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + ... + \beta_p X_p + \epsilon \tag{2.5}$$

If we fit this equation to out advertisement example we would get:

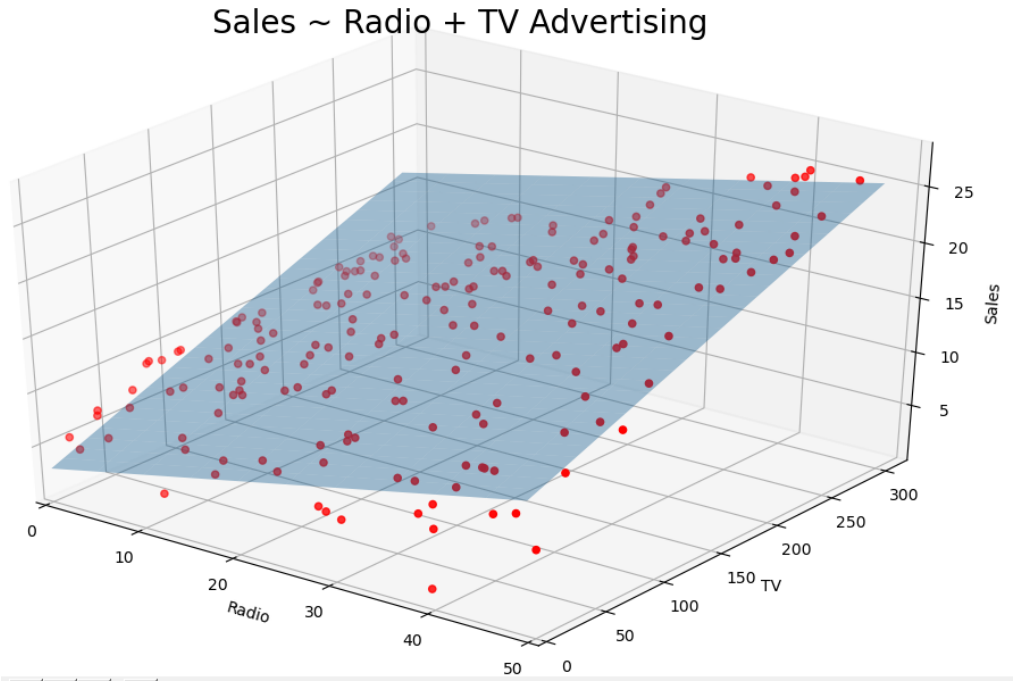
$$\textbf{sales} = \beta_0 + \beta_1 \textbf{TV} + \beta_2 \textbf{Radio} + \beta_2 \textbf{Newspaper} + \epsilon \tag{2.6}$$

2.2.1 Estimating the regression constants

To estimate the constants we do same as in the simple regression method with some changes. We still try to minimize the RSS value, but since there are more constants the RSS value is calculated the following way:

$$\begin{aligned} RSS &= \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \\ &= \sum_{i=1}^{n} (y_i - \hat{\beta}_0 - \hat{\beta}_1 x_{i1} - \hat{\beta}_2 x_{i2} - ... - \beta_p x_{ip})^2 \end{aligned} \tag{2.7}$$

When using two predictor variables the regression line becomes a plane. On fig. 2.3 the regression pane when using TV and Radio as predictor variables is shown. The plane is placed in a way such that the RSS value is as low as possible.



Figur 2.3: The regression pane with TV and Radio as predictor variables

The code for making the regression plane is listed below:

```
 # Multiple Regression
regr = skl_lm.LinearRegression()

X = advertising[['Radio', 'TV']].values
y = advertising.Sales

regr.fit(X, y)

Radio = np.arange(0, 50)
TV = np.arange(0, 300)

B1, B2 = np.meshgrid(Radio, TV, indexing='xy')
Z = np.zeros((TV.size, Radio.size))

for (i, j), v in np.ndenumerate(Z):
# The response on TV and Radio
Z[i, j] = (regr.intercept_ + B1[i, j] * regr.coef_[0] + B2[i, j] * regr.coef_[1])
```

Kapitel 3

LOGISITIC REGRESSION / LINEAR DISCRIMINANT ANALYSIS

Kapitel 4

CROSS VALIDATION / BOOTSTRAP

Kapitel 5

SUBSET SELECTION

Kapitel 6

SHRINKAGE AND DIMENSION REDUCTION METHODS

Contrary to subset selection methods, which uses least squares to fit a linear model containing a subset with n of all p predictors, shrinkage methods fits a model containing all p predictors. This is done by a penalty that regularizes the coefficient estimates and thereby shrinks them towards zero. By shrinking the coefficient estimates their variance can be significantly reduced. In this section the two shrinkage methods Ridge regression and The Lasso will be covered.
Both shrinking methods tries to control variance by either using a subset of the original variables or by shrinking their coefficients towards zero and uses all of the original predictors. Another class of approaches, dimension reduction, is one that transforms the predictors first and then fits a least squares model using the transformed predictors. The reduction comes from the fact that the methods reduces the problem of estimating $p + 1$ coefficients to estimating $M + 1$ where $M < p$. In this section the two dimension reduction methods Principal Component Regression and Partial Least Squares will be covered.

6.1 Ridge regression

Ridge regression is very similar to least square fitting in that it seeks to minimize RSS, but it adds a second term called the shrinkage penalty.
The equation 6.1 shows the full equation for ridge regression. Here the first term is the RSS where $\beta_0, ..., \beta_p$ is to be estimated such that it is minimized, but the second term introduces a penalty to β_j which effectively shrinks it towards zero. This penalty is scaled with λ such that as it moves towards zero, the penalty moves towards zero and the equation produces the least squares estimates. Moving the tuning parameter towards infinity will in turn drive the coefficient estimates towards zero.
Thus ridge regression will produce a different set of coefficient estimates, $\hat{\beta}^R_\lambda$, for each value of λ hence making it a tuning parameter. It is then critical to choose a good value for λ, which can be done e.g. by the cross-validation method.

$$\sum_{i=1}^{n}(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_i j)^2 + \lambda \sum_{j=1}^{p}\beta_j^2 \tag{6.1}$$

The advantage of ridge regression over standard least squares comes from the bias-variance trade-off. By increasing the value of λ and thereby the effect of the penalty term, it is possible to decrease the variance of the predictor however the bias increases. As the test MSE (mean squared error) is a function of the variance plus the squared

bias, finding a λ value which decreases the variance more than it increases the bias can result in a lower test MSE. Thus ridge regression will be superior whenever the least squares estimates have high variance.

6.2 The Lasso

The Lasso improves upon a disadvantage of ridge regression, namely that it includes all p predictors in the final model, because it only reduces the magnitudes of the coefficients but never actually excludes any of them. Where ridge regression uses the ℓ_2-norm in its regularization term (equation 6.1) , the Lasso uses ℓ_1-norm (equation 6.2). This has the effect of forcing some of the coefficients to be zero, depending on te value of λ, instead of only driving them towards zero. Therefore the Lasso acts like subset selection, as it effectively performs variable selection yielding a sparse model which exactly is its improvement over ridge regression.
Selecting a good value for λ is critical just as it is for ridge regression and can be done by cross-validation.

$$\sum_{i=1}^{n}(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_i j)^2 + \lambda \sum_{j=1}^{p}|\beta_j| \tag{6.2}$$

6.3 Principal Component Regression

As with all dimension reduction methods Principal Component Regression (PCR) works in two steps: first transformed predictors are obtained and then a model is fit using the transformed predictors.
The underlying method for obtaining the transformed predictors is in this case PCA (Principal Component Analysis). PCA derives a low-dimensional set of features from a large set of variables by acquiring basis vectors, *principal components*, that forms an orthogonal basis. This is done by finding vectors with values that minimizes the sum of squared perpendicular distances between each point and the vector. Finding the next principal component is done as a linear combination of the variables that is uncorrelated with the principal component before it and has the largest variance. Up to $M <= p$ principal components can be constructed this way, with p being the number of predictors. By this construction the first principal component will contain the most information of the data-set with each following principal component containing less and less information.
Fitting a least square model to $Z_1, ..., Z_M$ principal components instead of $X_1, ..., X_p$ data points with $p << M$, can in theory lead to better results as it can mitigate overfitting. This stems from the notion that most or all of the information in the data relating to the response if contained in the $Z_1, ..., Z_M$ principal component. If however $p == M$ PCR will perform the same as doing least square fitting on all of the original predictors.
In PCR the number of principal components M used for least square fitting becomes a hyper parameter that needs to be chosen carefully. This is typically done using the cross-validation method.

6.4 Partial Least Squares

In PCR the M principal components is guaranteed to best explain the predictors, however they are not guaranteed to best explain the response. This is due to the unsupervised nature of PCR, where the response does not supervise the identification of the transformed features (principal components).
Unlike PCR, Partial Least Squares (PLS) is supervised in the sense that the response is used to identify the transformed features that not only approximates the original predictors but also relates to the response.
PLS first identifies a new set of features $Z_1, ..., Z_M$ that represents $M < p$ linear combinations of the original p predictors. This is done by first standardizing the p predictors. Then the first translated feature Z_1 is calculated as in equation 6.3, with the difference that each θ_{j1} is set equal to the coefficient from the simple linear regression of the response Y onto X_j. In doing this tweaked version PLS places a higher weight on variables that are strongly related to the response. To compute Z_2 the residuals from regressing each variable on Z_1 are then used following the same calculations as for determining Z_1. Lastly least squares as in equation 6.4 is used to fit a linear model to predict Y using $Z_1, ..., Z_M$ translated feature instead of the original p features just like in PCR.

$$Z_m = \sum_{j=1}^{p} \phi_{jm} X_j \tag{6.3}$$

$$y_i = \theta_0 + \sum_{m=1}^{M} \theta_m Z_{im} + \epsilon_i \tag{6.4}$$