“All models are wrong, but some models are useful”--- George Box

Credit: wikipedia
Last time

- Linear regression
- The problem
- The least square solution
- The training and prediction
- The R-squared for the evaluation of the fit.

\[ y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p + \varepsilon \]

\[ y = X \beta + \varepsilon \]

\[ \hat{\beta} = (X^T X)^{-1} X^T y \]

\[ \hat{y} = X \hat{\beta} \]

\[ y = X \beta + e \]

\[ R^2 = \frac{\text{var}(X \hat{\beta})}{\text{var}(y)} \]
Objectives

- Linear regression (cont.)
- Modeling non-linear relationship with linear regression
- Outliers and over-fitting issues
- Regularized linear regression/Ridge regression
- Nearest neighbor regression
What if the relationship between variables is non-linear?

A linear model will not produce a good fit if the dependent variable is not linear combination of the explanatory variables.

\[
R^2 = 0.1
\]

\[
\text{corr} = \sqrt{0.1} \approx 0.3
\]
Transforming variables could allow linear model to model non-linear relationship.

In the word-frequency example, log-transforming both variables would allow a linear model to fit the data well.

$$\log f = \beta_0 + \beta_1 \log r$$

$${f = c \cdot \left(\frac{1}{r}\right)^s}$$

Zipf's law
More example: Data of fish in a Finland lake

- Perch (a kind of fish) in a lake in Finland, 56 data observations
- Variables include: Weight, Length, Height, Width
- In order to illustrate the point, let’s model Weight as the dependent variable and the Length as the explanatory variable.

Yellow Perch
Is the linear model fine for this data?

A. YES
B. NO
Is the linear model fine for this data?

- R-squared is 0.87 may suggest the model is OK.
- But the trend of the data suggests non-linear relationship.
- Intuition tells us length is not linear to weight given fish is 3-dimensional.
- We can do better!
Transforming the explanatory variables

- Weight vs length^3 in perch from Lake Laengelmavesi
- Weight predicted from length^3 in perch from Lake Laengelmavesi
Q. What are the matrix $X$ and $y$?

<table>
<thead>
<tr>
<th>1</th>
<th>Length$^3$</th>
<th>Weight</th>
</tr>
</thead>
</table>

$X^o = \begin{bmatrix} 1 & 2 \\ 1 & 3 \end{bmatrix}$

$\Rightarrow X^{\text{new}} = \begin{bmatrix} 1 & 8 \\ 1 & 27 \end{bmatrix}$

$y = y^o$ (as before)
Transforming the dependent variables

Weight^{1/3} vs length in perch from Lake Laengelmavesi

Weight^{1/3} predicted from length in perch from Lake Laengelmavesi
What is the model now?

\[ 3 \sqrt{y} = \beta_0 + \beta_1 x \]

\[ y = \beta_0 + \beta_1 x^3 \]

\[ y = \beta_0 + \beta_1 x + \beta_2 x^2 + \ldots + \beta_n x^n \]
What are the matrix $X$ and $y$?

<table>
<thead>
<tr>
<th></th>
<th>Length</th>
<th>$\sqrt[3]{w}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
General form of transformation in Linear Regression

\[ y = \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3 + \ldots \]
Effect of outliers on linear regression

- Linear regression is sensitive to outliers
Effect of outliers: body fat example

- Linear regression is sensitive to outliers
Over-fitting issue: example of using too many power transformations

\[Y = \beta_0 + \beta_1 x + \beta_2 x^2 + \ldots + \beta_n x^n\]
Avoiding over-fitting

- **Method 1: validation**
  - Use a validation set to choose the transformed explanatory variables.
  - The difficulty is the number of combination is exponential in the number of variables.

- **Method 2: regularization**
  - Impose a penalty on complexity of the model during the training.
  - Encourage smaller model coefficients.
  - We can use validation to select regularization parameter $\lambda$. 

$$
\text{cost} = \max(0, 1 - y_i \cdot a^T x) + \text{penalty} \\
\text{penalty} = \frac{1}{2} \| w \|^2
$$
In ordinary least squares, the cost function is $\|e\|^2$:

$$\|e\|^2 = \|y - X\beta\|^2 = (y - X\beta)^T(y - X\beta)$$

$L_2$

In regularized least squares, we add a penalty with a weight parameter $\lambda$ ($\lambda > 0$):

$$\|y - X\beta\|^2 + \lambda \frac{\|\beta\|^2}{2} = (y - X\beta)^T(y - X\beta) + \lambda \frac{\beta^T\beta}{2}$$
Training using regularized least squares

* Differentiating the cost function and setting it to zero, one gets:

\[(X^T X + \lambda I)\hat{\beta} - X^T y = 0\]

* \((X^T X + \lambda I)\) is always invertible, so the regularized least squares estimation of the coefficients is:

\[\hat{\beta} = (X^T X + \lambda I)^{-1} X^T y\]
Why is the regularized version always invertible?

Prove: \((X^T X + \lambda I)\) is invertible \((\lambda > 0, \lambda \text{ is not the eigenvalue})\).

- **Positive semi-definite:**
  \[ f^T Af \geq 0 \quad \Rightarrow \quad \text{eigenvalues } \lambda_i \geq 0 \]

- **Positive definite:**
  \[ f^T Af > 0 \quad \Rightarrow \quad f \text{ non-zero vector} \]

\[ f^T (X^T X + \lambda I) f > 0 \]
\[ = f^T X^T X f + f^T \lambda I f \]
\[ = f^T X^T X f + \lambda f^T f \]
\[ = f^T X^T X f + \lambda \|f\|^2 \]
\[ > 0 \quad \quad \lambda > 0 \]
Why is the regularized version always invertible?

Prove: \((X^T X + \lambda I)\) is invertible \((\lambda > 0\), \(\lambda\) is not the eigenvalue).

Energy based definition of semi-positive definite:
Given a matrix \(A\) and any nonzero vector \(f\), we have
\[
f^T Af \geq 0
\]
and positive definite means
\[
f^T Af > 0
\]

If \(A\) is positive definite, then all eigenvalues of \(A\) are positive, then it’s invertible.

for any nonzero vector \(f\)
consider \(f^T (X^T X + \lambda I) f\)
suppose \(A = X^T X + \lambda I\)
\[
f^T Af = f^T X^T X f + \lambda f^T f
\]
\[
= f^T X^T X f + \lambda \|f\|^2
\]
given \(X^T X\) is semi positive definite
\[
f^T X^T X f \geq 0
\]
given \(\lambda > 0\)
we know \(\lambda \|f\|^2 > 0\)
\[
\Rightarrow f^T Af > 0
\]
Over-fitting issue: example from using too many power transformations
Choosing lambda using cross-validation

\[ \hat{\beta} = (X^T X + \lambda I)^{-1} X^T y \]

Weight vs length in perch from Lake Laengelmavesi, all powers up to 10, regularized

Length: Coefficient of power
1: 6.72
2: 0.12
3: 2e-3
4: 4e-5
5: 7e-7
6: 1e-8
7: 1e-10
8: 7e-13
9: 3e-14
10: 2e-15

\[ \log \lambda = 4 \]
\[ \lambda = e^4 \]
\[ \lambda > 0 \]
Mean Square Error in this model

\[ \text{MSE} = \frac{e^T e}{N} = \frac{\| y - X\hat{\beta} \|^2}{N} \]

\[ \hat{\beta} = (X^T X + \lambda I)^{-1} X^T y \]

\[ y^p = X\hat{\beta} \]
Q. Can we use the R-squared to evaluate the regularized model correctly?

A. YES

B. NO

C. YES and NO
Q. Can we use the R-squared to evaluate the regularized model correctly?

A. YES
B. NO
C. YES and NO

\[ y = X\beta + e \]
\[ \text{var}(y) = \text{var}(X\beta) + \text{var}(e) + 2\text{cov}(X\beta, e) \]

\[ X^TX \beta = X^Ty \]
\[ e \perp X\beta \]
\[ \text{var}(y) = \text{var}(X\beta) + \text{var}(e) \]
\[ (X^TX + \lambda I)\beta = X^Ty \]
\[ e \perp X\beta \]
Nearest neighbor regression

- In addition to linear regression and generalize linear regression models, there are methods such as **Nearest neighbor regression** that do not need much training for the model parameters.

- When there is plenty of data, nearest neighbors regression can be used effectively.
The idea is very similar to k-nearest neighbor classifier, but the regression model predicts numbers. 

K=1 gives piecewise constant predictions.
The goal is to predict $y_0^p$ from $x_0$ using a training set $\{(x, y)\}$.

Let $\{(x_j, y_j)\}$ be the set of $k$ items in the training data set that are closest to $x_0$.

Prediction is the following:

$$y_0^p = \frac{\sum_j w_j y_j}{\sum_j w_j}$$

Where $w_j$ are weights that drop off as $x_j$ gets further away from $x_0$. 

$w_j \downarrow$ as $\text{dist} \uparrow$

$w_j \uparrow$ as $\text{dist} \downarrow$
Choose different weights functions for KNN regression

\[ y_0^p = \frac{\sum_j w_j y_j}{\sum_j w_j} \]

- Inverse distance
  \[ w_j = \frac{1}{\| x_0 - x_j \|} \]

- Exponential function
  \[ w_j = \exp\left(-\frac{\| x_0 - x_j \|^2}{2\sigma^2}\right) \]
Evaluation of KNN models

Which methods do you use to choose K and weight functions?

A. Cross validation
B. Evaluation of MSE
C. Both A and B
The Pros and Cons of K nearest neighbor regression

Pros:
- The method is very intuitive and simple
- You can predict more than numbers as long as you can define a similarity measure.

Cons
- The method doesn’t work well for very high dimensional data
- The model depends on the scale of the data
Assignments

- Finish Chapter 13 of the textbook
- Week 13 module including the quiz
- Next time: Curse of Dimension, clustering
Additional References

- Kelvin Murphy, “Machine learning, A Probabilistic perspective”
See you next time

See You!