Lecture-1: Advanced Regression

Regression Revisited, Regularization, Generalized linear Modelling

Introduction

- Linear Regression
- Predictor Interactions
- Polynomial Models
- Shrinkage / Regularisation
  - Ridge Regression
  - Lasso

General Linear Models

Introduction

- Recall

  - The Linear Regression Model

    $y_j = \beta_0 + \beta_1 x_{1j} + \beta_2 x_{2j} + \cdots + \beta_n x_{nj} + \varepsilon_j$ for $j = 1, 2, \dots, m$.

    - The ε_j's are independently and identically distributed as $\mathcal{N}(0, \sigma^2)$ and m is the number of data points.

    - The expected value of y_j can be written as

$$E(y_j) = \hat{y}_j = \beta_0 + \sum_{i=1}^{n} \beta_i x_{ij}$$

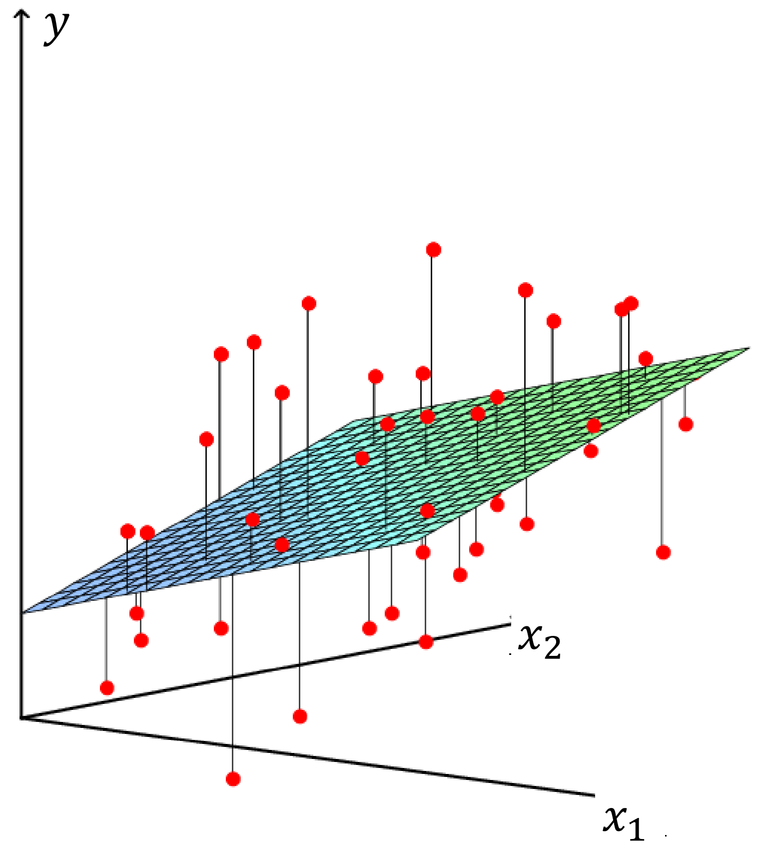
    - The maximum likelihood estimator of the β's are the same as the least square estimators

Introduction

- Least Squares Fit
  - We minimise the *residual sum of squares(RSS)*

$$RSS = \sum_{j=1}^{m}(y_j - \hat{y}_j)^2$$

$$= \sum_{j=1}^{m}(y_j - \beta_0 - \beta_1 x_{1j} - \cdots - \beta_n x_{nj})^2$$



Introduction

- Idea of a General Linear Model

  - There is a large class of regression problems that can be converted into the form of the *general linear model*

    - Polynomial relationships / interaction terms

$$E(y) = \hat{y} = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_1 x_2 + \beta_4 {x_1}^2 + \beta_5 {x_2}^2$$

    - We can consider this as a linear model by setting

$$\begin{gathered} x_3 = x_1 x_2 \\ x_4 = {x_1}^2 \\ x_5 = {x_2}^2 \end{gathered}$$

Introduction

- Advantages of the Linear Model

  - Even when the linear model is not strictly appropriate, we can often transform the output and/or the inputs so that a linear model can provide useful information

  - Possible to determine the statistical properties and perform statistical inference
    - Use hypothesis testing to compare different linear models and obtain interval estimates for predictions and for the β's

  - Interpretability

Regularization

- Penalising Models

  - Penalise complex models

    - Decrease variance / overfitting

  - Rather than solely minimizing the loss associated with a given model and data, the complexity of the model will also be taken into account

  - We minimize both the loss and complexity of the model

Empirical Risk Minimization ⟹ $\text{minimize}(\text{Loss}(\text{Data}|\text{Model}))$

Structural Risk Minimization ⟹ $\text{minimize}(\text{Loss}(\text{Data}|\text{Model}) + \text{complexity}(\text{Model}))$

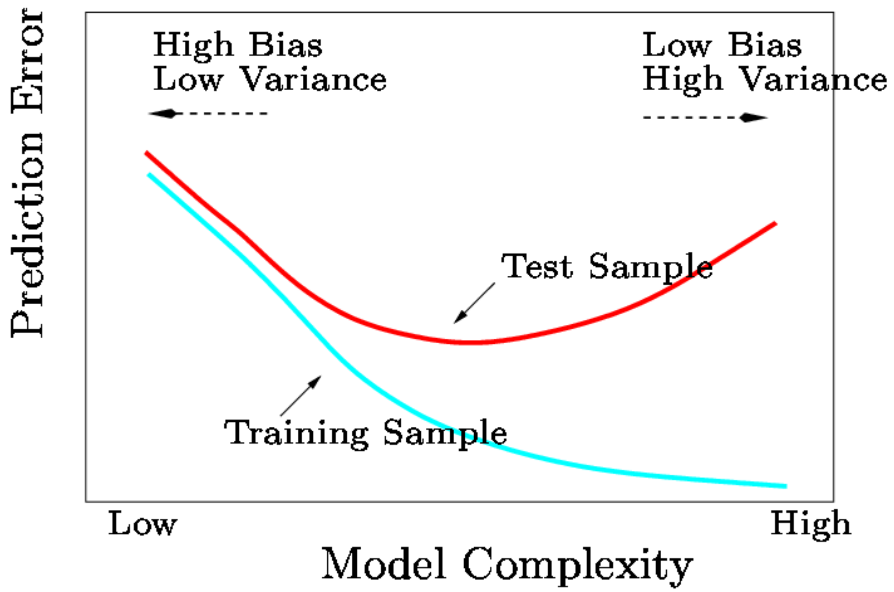
Regularization

- Penalising Models

> We want to reduce the test error, possibly at the expense of the training error

- How can we determine some measurement for the complexity of a model?
  - We could think of the model's complexity as being a function of the weights associated with the features of a model
  - We could think of the model's complexity as being a function of the total number of features with non-zero weights

Bias/Variance Trade-off



Regularization

- Penalising Models

We want to find the right balance between fitting the data and keeping the model simple enough to ensure that the model will generalize well.

- Regularization is the process of constraining a model to make it simpler and reduce the risk of overfitting

- Typically, controlled by learning algorithm hyperparameter λ that has to be tuned
  - Remember the *No Free Lunch Theorem*

- For a linear model, regularization is controlled by constraining the weights/coefficients of the model
  - Ridge Regression
  - Lasso Regression
  - Elastic Net

Each of these methods will control the weights in different ways by applying various shrinkage penalties

Regularization

- Ridge Regression
  - Also known as ℓ_2-Regularization
  - Here we minimize

- This has the effect of "shrinking" large values of β's towards zero
- Shrinking the coefficients can significantly reduce their variance
- Note: β_0 is not regularized
- Note: It is important to z-score scale data prior to ℓ_2-regularization

$$\underbrace{\sum_{j=1}^{m}(y_j - \beta_0 - \beta_1 x_{1j} - \cdots - \beta_n x_{nj})^2}_{RSS} + \underbrace{\lambda \sum_{i=1}^{n} {\beta_i}^2}_{Penalty}$$

for some $\lambda > 0$

Regularization

- Ridge Regression

  - Weights / coefficients close to zero have a small effect on model complexity as opposed to large weights that can have a considerably greater impact

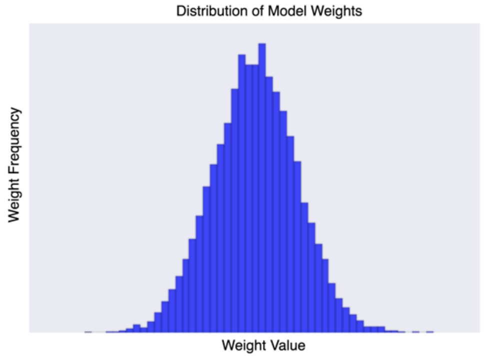
  - Regularization drives weights towards zero (but not exactly 0)

  - The regularization contribution to the gradient scales linearly with each β_i

  - The mean of the weights moves towards zero with the weights exhibiting a Gaussian distribution

  - A high λ value will produce a simpler model
    - Care needs to be taken that we don't underfit

  - Also known as Tikhonov Regularization



Regularization

- Heteroskedastic Ridge Regression

  - Don't z-score scale the coefficients
    - Variances of predictors can remain unequalised
  - Instead use the variances as weights in the penalty term
    - Penalise different coefficients with different strength

Set $w_i = \sigma_i$

- β_i of variables with small σ_i, thus little uncertainty in the estimate, are less penalized
- β_i of variables with large σ_i, thus much uncertainty in the estimate, are heavily penalized

  - Here we minimize

$$\underbrace{\sum_{j=1}^{m}(y_j - \beta_0 - \beta_1 x_{1j} - \cdots - \beta_n x_{nj})^2}_{RSS} + \underbrace{\lambda \sum_{i=1}^{n} w_i {\beta_i}^2}_{Penalty}$$

for some $\lambda > 0$ and $w_i > 0$

Regularization

- Lasso (**L**east **A**bsolute **S**hrinkage and **S**election **O**perator)

  - Similar to ridge regression but this time we use the ℓ_1 norm in the penalty term
    - ℓ_1-Regularization

$$\sum_{j=1}^{m}(y_j - \beta_0 - \beta_1 x_{1j} - \cdots - \beta_n\, x_{nj})^2 + \lambda \sum_{i=1}^{n} |\beta_i|$$

for some $\lambda > 0$

RSS

$Penalty$

Regularization

- Lasso (**L**east **A**bsolute **S**hrinkage and **S**election **O**perator)
  - Lasso will tend to completely eliminate the weights of the least important features
    - The weights are set to zero
  - Lasso automatically performs feature selection!
    - The output will be a *sparse* model
  - Computational savings when working with sparse vectors
  - The regularization contribution to the gradient no longer scales linearly with each β_i ; instead it is a constant factor with a sign equal to $\mathrm{sign}(\beta_i)$

Regularization

- Elastic Net

  - A middle-ground between ridge regression and lasso regression

  - Regularization penalty term is a mixture of penalty terms from ridge regression and lasso regression where the ratio of the mix is parameterised

$$\underbrace{\sum_{j=1}^{m}(y_j - \beta_0 - \beta_1 x_{1j} - \cdots - \beta_n x_{nj})^2}_{RSS} + \underbrace{r\lambda \sum_{i=1}^{n}|\beta_i| + \frac{1-r}{2}\lambda \sum_{i=1}^{n}\beta_i{}^2}_{Penalty}$$

for some $\lambda > 0$ and some $0 \leq r \leq 1$

Regularization

- Other Techniques

  - Early Stopping

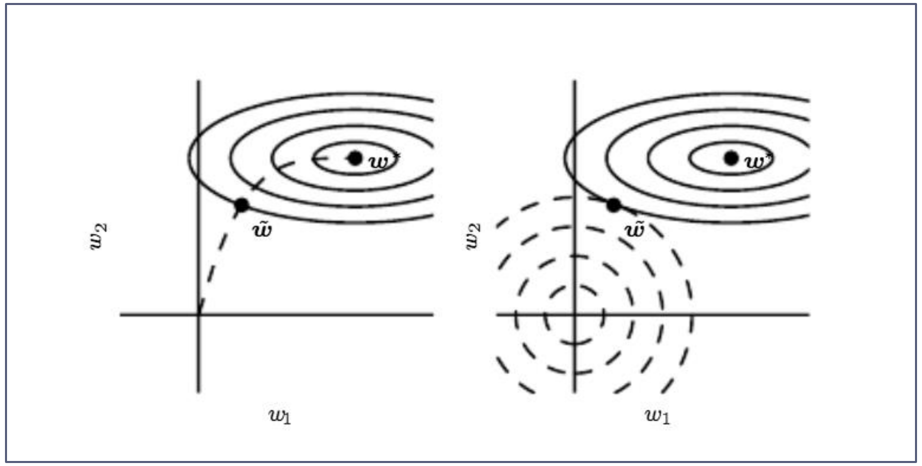


Figure 7.4: An illustration of the effect of early stopping. *(Left)*The solid contour lines indicate the contours of the negative log-likelihood. The dashed line indicates the trajectory taken by SGD beginning from the origin. Rather than stopping at the point $\boldsymbol{w}^*$ that minimizes the cost, early stopping results in the trajectory stopping at an earlier point $\tilde{\boldsymbol{w}}$. *(Right)*An illustration of the effect of L^2 regularization for comparison. The dashed circles indicate the contours of the L^2 penalty, which causes the minimum of the total cost to lie nearer the origin than the minimum of the unregularized cost.

https://www.deeplearningbook.org/contents/regularization.html

Generalised Linear Models

- Generalised Linear Models

  - The generalised linear model offers an expansion of the problems that can be addressed by regression

  - Generalisation is in two parts:

    - The distribution of the output does not have to be normal but can be any of the distributions in the exponential family (e.g. Poisson, Binomial, Gamma)

    - Instead of the expected value of the output being a linear function of the β's, we have

$$g\left(E(y_j)\right) = g(\hat{y}_j) = \beta_0 + \sum_{i=1}^{n} \beta_i x_{ij}$$

where $g(\cdot)$ is a monotone differentiable function called the **link function**.

Generalised Linear Models

- Generalised Linear Models

  - Assumptions
    - The data $y_1, y_2, \dots, y_3$ are independently distributed
    - The output variable does not need to be normally distributed
    - Generalised linear models does not assume a linear relationship between the output and the inputs; but a linear relationship between the transformed response in terms of the link function and the inputs
      - A function of the response varies linearly with the inputs
    - No assumption on the homogeneity of variance
    - Errors are independent but not necessarily normally distributed

Generalised Linear Models

- Generalised Linear Models

  - Some non-linear models can be framed as generalised linear models

  - There are general algorithms for fitting generalised linear models
    - Iterative application of least squares estimates

  - Examples

    - Log-linear Models

    - Logistic Regression

Generalised Linear Models

- Log Linear Models

  - Useful to investigate relationships between categorical variables

  - The output is assumed to have a Poisson distribution with expected value μ_j
  - The (natural) log of μ_j is assumed to be linear in the β's

$$y_j \sim \mathrm{Poi}(\mu_j)$$

$$\ln(\mu_j) = \beta_0 + \beta_1 x_{1j} + \beta_2 x_{2j} + \cdots + \beta_n x_{nj}$$

  - Categorical variables would require appropriate encoding

Generalised Linear Models

- Log Linear Models

  - Associations between inputs correspond to interaction terms in the model
    - We want to find out which β's are zero
      - Which of the inputs are actually associated with the output?

    - For the log linear model we consider a quantity called *deviance* when we are trying to compare two models

Deviance is analogous to the sum of squares in an ANOVA.

In fact, if the generalized linear model happens to be a linear model, the *deviance* and the sum of squares are the same thing.

The Analysis of Deviance works in a similar way to ANOVA.

Generalised Linear Models

- Logistic Regression

  - Model the number of successes out of a number of trials with each trial resulting in either success or failure

  - The output is assumed to have a Binomial distribution $\mathrm{Bin}(n_j, p_j)$ with expected value $\mu_j = p_j$ where p_j is the probability of success

  - The logit link function is assumed to be linear in the β's

$$y_j \sim \mathrm{Bi}(n_j, \mu_j)$$

$$\ln\left(\frac{p_j}{1-p_j}\right) = \ln\left(\frac{\mu_j}{1-\mu_j}\right) = \beta_0 + \beta_1 x_{1j} + \beta_2 x_{2j} + \cdots + \beta_n x_{nj}$$

Generalised Linear Models

- Generalised Additive Models (GAMs)

  - A generalisation of generalised linear models

  - The generalisation is that

$$g\left(E(y_j)\right) = g(\hat{y}_j) = \beta_0 + \sum_{i=1}^{n} s_i(x_{ij})$$

    where the s_i's are arbitrary (usually smooth) functions.

  - The smoothing functions (the s_i's) are estimated from the data
    - Can be computationally expensive
    - May use bagging / boosting techniques

Generalised Linear Models

- Python Example (using Pandas)

```
df = pd.read_csv('Data/Grade_Set_1.csv')

print('####### Linear Regression Model ########')
# Create linear regression object
lr = lm.LinearRegression()

x= df.Hours_Studied[:, np.newaxis] # independent variable
y= df.Test_Grade.values            # dependent variable

# Train the model using the training sets
lr.fit(x, y)

print "Intercept: ", lr.intercept_
print "Coefficient: ", lr.coef_

print('\n####### Generalized Linear Model ########')
import statsmodels.api as sm

# To be able to run GLM, we'll have to add the intercept constant to x
variable
x = sm.add_constant(x, prepend=False)

# Instantiate a gaussian family model with the default link function.
model = sm.GLM(y, x, family = sm.families.Gaussian())
model = model.fit()
print model.summary()
```

Generalised Linear Models

- Python Example (using Pandas)

```
#----output----

####### Linear Regression Model ########
Intercept:  49.6777777778
Coefficient:  [ 5.01666667]

####### Generalized Linear Model ########
                 Generalized Linear Model Regression Results
==========================================================================
Dep. Variable:                y   No. Observations:                    9
Model:                      GLM   Df Residuals:                        7
Model Family:          Gaussian   Df Model:                            1
Link Function:         identity   Scale:                    5.3626984127
Method:                    IRLS  Log-Likelihood:                 -19.197
Date:          Sun, 25 Dec 2016   Deviance:                       37.539
Time:                  21:27:42   Pearson chi2:                     37.5
No. Iterati                   4
=========================================================================
         coef    std err     z     P>|z|    [95.0% Conf. Int.]
-------------------------------------------------------------------------
x1       5.0167   0.299   16.780      0.000         4.431    5.603
const    49.6778  1.953     25.439      0.000         45.850  53.505
=========================================================================
```

Why are the coefficients the same for both the linear regression model and the generalized linear model?

Summary

- Summary
  - Linear Regression
  - Regularization
  - GLMs
- References / Bibliography

  - Dangeti P., (2017). *Statistics for Machine Learning*. Packt Publishing.
  - Swamynathan M., (2017). *Mastering Machine Learning with Python in Six Steps*. Packt Publishing.
  - Géron A., (2017). *Hands-On Machine Learning with Scikit-Learn and Tensorflow.* O'Reilly.
  - https://developers.google.com/machine-learning/crash-course/regularization-for-simplicity/l2-regularization
  - https://www.deeplearningbook.org/contents/regularization.html
  - https://www.datacamp.com/community/tutorials/tutorial-ridge-lasso-elastic-net