Regression and Classification

Part 1: Regression

Sourish Das

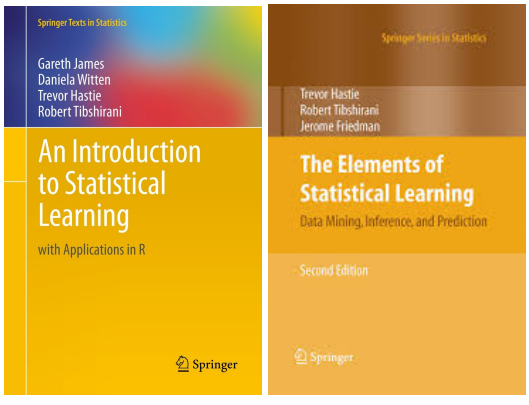
Chennai Mathematical Institute

Aug-Nov, 2019

Outline

Introduction

Reference



Reading material

- *Data Mining; Concepts and Techniques*, Jiawei Han and Micheline Kamber, Morgan Kaufman (2006).
- *Web Data Mining*, Bing Liu, Springer Verlag (2007).

For a good introduction to text mining and information retrieval, please see.

- *An Introduction to Information Retrieval*, Christopher D Manning, Prabhakar Raghavan and Hinrich Schütze, Cambridge University Press (2009). (Available online at `http://www-nlp.stanford.edu/IR-book`).

Introduction

- Researchers interested in **artificial intelligence** wanted to see if computers could learn from data.
- The iterative aspect of machine learning is important
- As models are exposed to new data, it should be able to independently adapt.
- learn from previous computations to produce reliable, repeatable decisions and results.

Popular Applications of ML

- The heavily hyped, self-driving Google car? The essence of machine learning.
- Machine learning applications for everyday life: Recommendation offers on Amazon or flipKart !!
- What customers are saying about you on Twitter? Machine learning combined with NLP.
- Credit Score to Probability of Default !! One of the extremely well known application of ML.

Popular Applications of ML

- Is Machine Learning (ML) different from Predictive Modeling?
- **Example**: Microsoft's chatbot "Tay" is a good example of machine learning.
- The chat bot was exposed to the real world and was given a chance to learn on it's own.

Popular Applications of ML

Popular Applications of ML

Popular Applications of ML

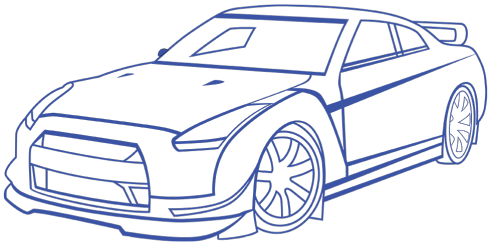
Popular Applications of ML

Microsoft took out this bot in less than 24 hours.

Supervised Learning

Motivating Examples of Supervised Learning

Ex 1 Given the different features of a new prototype car, can you predict the mileage or 'miles per gallon' of the car?



Motivating Examples of Supervised Learning

Ex 1 Given the different features of a new prototype car, can you predict the mileage or 'miles per gallon' of the car?

	mpg	cyl	disp	hp
Mazda RX4	21.0	6	160	110
Mazda RX4 Wag	21.0	6	160	110
Datsun 710	22.8	4	108	93
Hornet 4 Drive	21.4	6	258	110
.....				
Prototype	?	4	120	100

- Note that your objective is to predict the variable `mpg`.
- We are going to use `mtcars` data set in `R`.

Motivating Examples of Supervised Learning

Ex 2 Given the credit history and other features of a loan applicant, a bank manager want to predict if loan application would become good or bad loan!!

- Note that your objective is to predict the label of the loan good or bad!

Motivating Examples of Supervised Learning

Ex 3 Can you predict the Air Pressure Failue of Scania Truck?

- We are going to use `aps_failure_training_set.csv` and `aps_failure_test_set.csv` in `Data` folder.

Motivating Examples of Supervised Learning

Ex 4 Can you identify the correlation pathways between gene expression and the disease?

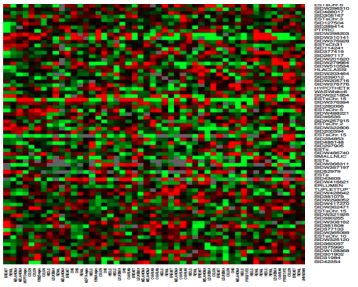


FIGURE 1.3. *DNA microarray data: expression matrix of 6830 genes (rows) and 64 samples (columns), for the human tumor data. Only a random sample of 100 rows are shown. The display is a heat map, ranging from bright green (negative, under expressed) to bright red (positive, over expressed). Missing values are gray. The rows and columns are displayed in a randomly chosen order.*

Supervised learning

- Supervised learning algorithms are trained using labeled data.
- For example, a piece of equipment could have data points labeled either "F" (failed) or "R" (runs).
- Typically,
$$y = f(X),$$
where y is target variable and X is feature matrix
- Objective: Learn $f(.)$

Supervised learning

- Supervised learning
$$y = f(X)$$
typically are of two types:
  1. Regression : target variable y is continuous variable - e.g., income, blood pressure, distance etc.
  2. Classification: target variable y is categorical or label variable - e.g., species type, color, class etc.

Data : Quantitative Response

x_{11}	x_{12}	$\dots$	x_{1p}	y_1
x_{21}	x_{22}	$\dots$	x_{2p}	y_2
$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
x_{n1}	x_{n2}	$\dots$	x_{np}	y_n
x^*_{11}	x^*_{12}	$\dots$	x^*_{1p}	$y^*_1 =?$
$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
x^*_{m1}	x^*_{m2}	$\dots$	x^*_{mp}	$y^*_m =?$

- $D_{train} = (X, y)$, is the traing dataset, where X is the matrix of predictors or features, y is the dependent or target variable.
- $D_{test} = (X^*, y^* =?)$ is the test dataset, where X^* is the matrix of predictors or features, and y^* is missing and we want to forecast or predict y^*

Data : Qualitative Response

x_{11}	x_{12}	$\dots$	x_{1p}	G_1
x_{21}	x_{22}	$\dots$	x_{2p}	G_2
$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
x_{n1}	x_{n2}	$\dots$	x_{np}	G_n
x^*_{11}	x^*_{12}	$\dots$	x^*_{1p}	$G^*_1 =?$
$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
x^*_{m1}	x^*_{m2}	$\dots$	x^*_{mp}	$G^*_m =?$

- Qualitative variables are also referred to as *categorical* or *discrete* variables as well as *factors*.

Practical Session with R

- Open `mtcars_analysis.R`
- Check out how we split the data into `train_data` and `test_data`.

Regression

Regression

- Given a vector of inputs $X^T = (X_1, X_2, \ldots, X_p)$, we predict the output Y via model

$$\hat{Y} = \hat{\beta}_0 + \sum_{j=1}^{P} X_j \hat{\beta}_j.$$

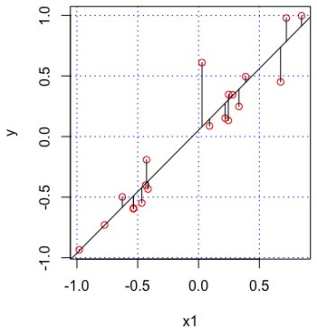
The term $\hat{\beta}_0$ is the intercept, also known as the *bias* in machine learning.

- Often it is convenient to include the constant variable 1 in X, include $\hat{\beta}_0$ in the vector of coefficients $\hat{\beta} = (\hat{\beta}_1, \ldots, \hat{\beta}_p)$
- write the linear model

$$\hat{Y} = X^T \hat{\beta}$$

Regression

$y = \hat{\beta}_0 + \hat{\beta}_1 x_1$



Regression

$y = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2$

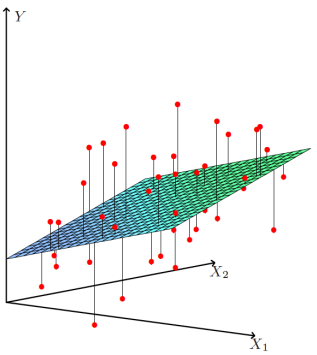


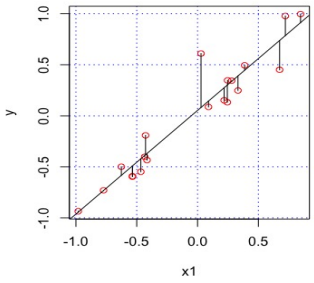
FIGURE 3.1. *Linear least squares fitting with $X \in \mathbb{R}^2$. We seek the linear function of X that minimizes the sum of squared residuals from Y.*

Regression

- Viewed as a function over the p-dimensional input space X
- $f(X) = X^T \beta$
- $f'(X) = \beta$: gradient is a vector in input space that points in the steepest uphill direction.

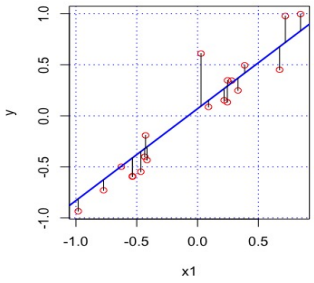
Regression

$y = \hat{\beta}_0 + \hat{\beta}_1 x_1 = 0.05 + 1.01\ x_1$



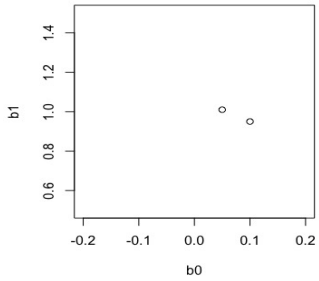
Regression

$y = \hat{\beta}_0 + \hat{\beta}_1 x_1 = 0.07 + 0.9\ x_1$



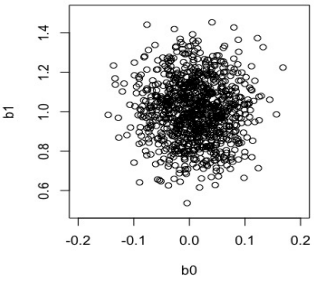
Choice of β

$(\beta_0 = 0.07,\ \beta_1 = 0.9)$ and $(\beta_0 = 0.05,\ \beta_1 = 1.01)$



Choice of β

However, thousands of choices are there, which one is best?

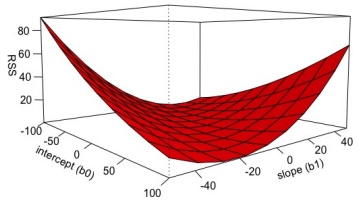


How do we fit Linear Regression Models?

- Many different methods, most popular is *least squares*.
- minimize the residual sum of squares

$$\begin{aligned} RSS(\beta) &= (y - X\beta)^T(y - X\beta) \\ &= \sum_{i=1}^{N}(y_i - x_i^T\beta)^2 \end{aligned}$$

Residual Sum of Square : Surface



- $RSS(\beta)$ is a quadratic function of the parameters
- Its minimum always exists, *but may not be unique*.

How do we fit Regression models?

- Differentiate $RSS(\beta)$ with respect to β and equate to 0

$$\begin{aligned} & \frac{\partial RSS(\beta)}{\partial \beta} = 0 \\ \implies & \frac{\partial}{\partial \beta}(y - X\beta)^T(y - X\beta) = 0 \\ \implies & -2X^T(y - X\beta) = 0 \\ \implies & X^TX\beta = X^Ty \quad \text{Normal Equations} \end{aligned}$$

- X^TX is $p \times p$ matrix,
- So normal equations have p unknown and p equations.

System of Equation

- Suppose that for a known matrix $A_{p\times p}$ and vector $b_{p\times 1}$, we wish to find a vector $x_{p\times 1}$ such that

$$Ax = b$$

- The standard approach is ordinary least squares linear regression.

$$\underset{x}{\text{minimize}} \quad ||Ax - b||^2$$

where $||.||$ is the Euclidean norm.

- Solution for x is

$$\hat{x} = A^{-1}b$$

- What happened A is not invertible?

Solution to System of Equation

- If $rank(A|b) > rank(A)$ then solution does not exists.
- If $rank(A|b) = rank(A)$ then at least one solution exists.
- If $rank(A|b) = rank(A) = p$, that is A is a full-rank matrix, then A^{-1} uniquely exists and the solution $\hat{x} = A^{-1}b$ is unique.
- If $rank(A|b) = rank(A) < p$, that is A is a less than full-rank matrix, then x has infinitely many solutions. This is considered as ill-posed problem. Which solution to choose and how to choose?

How do we fit Regression models?

Theorem
For normal equations,

$$rank(X^TX|X^Ty) = rank(X^TX)$$

- Whatever may be your data, irrespective of that, normal equations gurantees at least one solution.
- Atleast one solution always exists - if you adopt least squares method.
- If X^TX is nonsingular, i.e., $rank(X^TX) = p$, then the unique solution is given by

$$\hat{\beta} = (X^TX)^{-1}X^Ty$$

Practical Session with R

- Implement the OLS method for `mtcars` data
- Suppose we want to model the `mpg` as a function of `hp`
- the model we consider :
  $\texttt{mpg} = \beta_0 + \beta_1\ \texttt{hp} + \epsilon$

How do we fit Regression models?

- What happened if

$$rank(X^TX|X^Ty) = rank(X^TX) < p?$$

- No unique solution - but infinitely many solutions.
- It happens if $n < p$, i.e., sample size is less than the number of predictors
- Intuitively, it seems that we do not need a very large dataset to fit such a model.
- Microarray data in genomics and proteomics often you might have cases where $n < 100$ and $p > 1000$.
- These problems often known as high-dimension problem or "Large p - small n" problem.

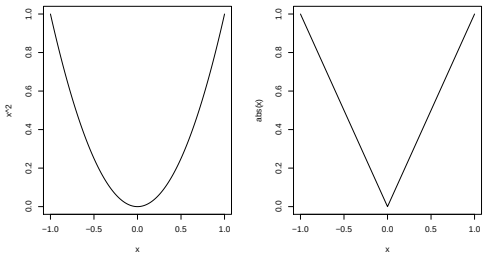
How do we fit Linear Models?

- What about mean absolute deviation?

$$\Delta(\beta) = \sum_{i=1}^{N} ||y_i - x_i^T \beta||$$

- Conceptually no problem - certainly you can do that.

How do we fit Regression models?

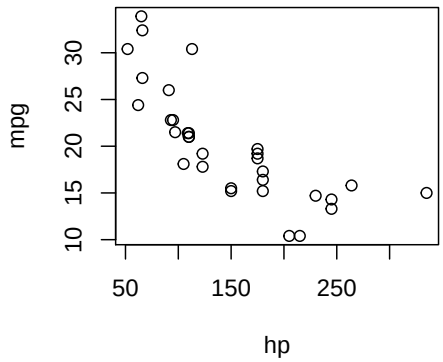


Practical Session with R

- Small data analysis to understand the consepts we consider the `mtcars_analysis.R`
- Large n small p data –`PRSA_data_analysis.R`
- Small n large p data – `colonCA.R`

Quadratic Regression

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



Quadratic Regression

- $\texttt{mpg} = \beta_0 + \beta_1\ \texttt{hp} + \beta_2\ \texttt{hp}^2 + \epsilon$
- We write the model in terms of linear models

$$Y = X\beta + \epsilon$$

where $Y = (\texttt{mpg}_1\ ,\ \texttt{mpg}_2\ ,\ \ldots\ ,\ \texttt{mpg}_n)^T$;

$$X = \begin{bmatrix} 1 & \texttt{hp}_1 & \texttt{hp}_1^2 \\ 1 & \texttt{hp}_2 & \texttt{hp}_2^2 \\ \vdots & \vdots & \vdots \\ 1 & \texttt{hp}_n & \texttt{hp}_n^2 \end{bmatrix},$$

$\beta = (\beta_0, \beta_1, \beta_2)^T$ and $\epsilon = (\epsilon_1, \epsilon_2, \ldots, \epsilon_n)^T$

- The linear model is linear in parameter.

Non-linear Regression Basis Functions

- Consider i^{th} record

$$y_i = f(t_i) + \epsilon_i$$

represents $f(t)$ as

$$f(t) = \sum_{j=1}^{K} \beta_j \phi_j(t) = \phi\beta$$

we say ϕ is a basis system for $f(t)$.

Representing Functions with Basis Functions

- Terms for curvature in linear regression

$$y_i = \beta_1 + \beta_2 t_i + \beta_3 t_i^2 + \ldots + \epsilon_i$$

implies

$$f(t) = \beta_1 + \beta_2 t + \beta_3 t^2 + \ldots$$

Fourier Basis

- sine cosine functions of incresing frequencies

$$y_i = \beta_1 + \beta_2 \sin(\omega t) + \beta_3 \cos(\omega t) + \beta_4 \sin(2\omega t) + \beta_5 \cos(2\omega t) \ldots + \epsilon_i$$

- constant $\omega = 2\pi/P$ defines the period P of oscillation of the first sine/cosine pair.

- $\phi = \{1, \sin(\omega t), \cos(\omega t), \sin(2\omega t), \cos(2\omega t) \ldots\}$
- $\beta^T = \{\beta_1, \beta_2, \beta_3, \ldots\}$

$$y = \phi\beta + \epsilon$$

Other Basis

- **Exponential Basis** $\phi = \{1, e^{\lambda_1 t}, e^{\lambda_2 t} ...\}$
- **Gaussian Basis**
  $\phi = \{1, \exp(-\lambda(t_1 - c)^2), \exp(-\lambda(t_2 - c)^2) ...\}$
- **Basis corresponds to Spline Regression**

$$y = \beta_0 + \sum_{k=1}^{K} \beta_k (t - \xi_k)_+^D + \ldots \epsilon$$

$\phi = \{1, (t - \xi_1)_+^D, (t - \xi_2)_+^D ...\}$

Functional Estimation/Learning

- We are writing the function with its basis expansion

$$y = \phi\boldsymbol{\beta} + \epsilon$$

- Lets assume basis ϕ is fully known
- Problem is $\boldsymbol{\beta}$ is unknown - hence we estimate $\boldsymbol{\beta}$.

Statistical Decision Theory

- $X \in \mathbb{R}^p$ denote a real valued random input vector
- $Y \in \mathbb{R}$ a real valued random output variable
- $\mathbb{P}(X, Y)$: Joint probability of X and Y
- $f(X)$ seek a function for predicting Y given values of the input X.
- $L(Y, f(X)) = (Y - f(X))^2$ is a squared error loss function or L_2 loss function

Statistical Decision Theory

- This leads us to a criterion for choosing f

$$\begin{aligned} EPE(f) &= \mathbb{E}(Y - f(X))^2 \\ &= \int (y - f(x))^2 \mathbb{P}(dx, dy) \end{aligned}$$

  EPE: Expected (squared) prediction error
- By conditioning on X, we can write EPE as

$$EPE(f) = \mathbb{E}_X \mathbb{E}_{Y|X}([Y - f(X)]^2 | X)$$

we would like to choose f, such that EPE is minimum.

Statistical Decision Theory

- Suffices to minimize EPE pointwise:

$$f(x) = argmin_g \mathbb{E}_{Y|X}([Y - g(x)]^2 | X = x)$$

- The solution is

$$f(X) = \mathbb{E}(Y|X = x)$$

the conditional expectation, also known as the regression function.

- The nearest-neighbor methods attempt to directly implement this recipe using the training data.

Statistical Decision Theory

- How does linear regression fit into this framework?
- The simplest explanation is that one assumes that the regression function $f(x)$ is approximately linear is its arguements:

$$f(x) \approx x^T \beta$$

- Should we be happy with L_2 loss function?
- What about L_1 loss function? That is

$$L_1 = |Y - f(X)|$$

The solution in this case is conditional median:

$$f(x) = median(Y|X = x)$$

Bias-Variance Tradeoff

- Suppose $\hat{y}_0 = x_0\hat{\beta}$ is the estimate of $f(x_0)$
- The expected prediction error at x_0 under L_2 is mean mean squared error (MSE) for estimating $f(x_0)$

$$\begin{aligned} MSE(x_0) &= \mathbb{E}(\hat{y}_0 - f(x_0))^2 \\ &= \mathbb{E}(\hat{y}_0 - \mathbb{E}(\hat{y}_0))^2 + [\mathbb{E}(\hat{y}_0) - f(x_0)]^2 \\ &= \mathbb{V}ar(\hat{y}_0) + Bias^2(\hat{y}_0) \end{aligned}$$

Feature Extraction/Transformed Predictors

- **Feature extraction** starts from an initial set of measured data and builds derived values (features)
- Suppose $X = \{X_1, X_2, \ldots, X_p\}$ is the available data
- Variable transformation such as X_j^k, or $\log(X_j)$ etc. can be adapted and feature space can be extended.
- Feature extraction extends the feature space into higher-dimension space.
- Often this may lead to overfitting and hence poor performance of the model.

Feature Extraction

- **Feature extraction** starts from an initial set of measured data and builds derived values (features)
- Suppose $X = \{X_1, X_2, \ldots, X_p\}$ is the available data
- Variable transformation such as X_j^k, or $\log(X_j)$ etc. can be adapted and feature space can be extended.
- Feature extraction extends the feature space into higher-dimension space.
- Often this may lead to overfitting and hence poor performance of the model.

Advanced Technique for Feature Extraction

- Apply unsepervised learning on **X** and discover new features X_{new} and add them into the design matrix **X**.
- One can apply k-means clustering on **X** and the cluster levels as new features in the model
- One can apply PCA to **X** and identify new extracted features

Principal Component Analysis

Principal Component Analysis

- Principal Components are linear combination of random statistical variables which have special properties in terms of variance.
- The first principal component is the normalized linear combination (the sum of squares of coefficients being one) with maximum variance.
- In effect, transforming the original vecctor variable to the vector of principal components amounts to a rotation of coordinate axes to a new coordinate system that has inherent statistical properties.

Principal Component Analysis

- Suppose $\mathbf{X}$ is random vector with p components
- Covariance matrix of $\mathbf{X}$ is Σ
- Assume $\mathbf{X}$ is already scaled - so mean vector is $\mathbf{0}$
- Lete β be p component column vector such the $\beta^T\beta = 1$
- The variance of $\beta^T\mathbf{X}$ is

$$\mathbb{E}(\beta^T\mathbf{X})^2 = \mathbb{E}(\beta^T\mathbf{X}\mathbf{X}^T\beta) = \beta^T\Sigma\beta.$$

Principal Component Analysis

- **Determine the normalized linear combination $\beta^T\mathbf{X}$ with maximum variance.**
- We must find a vector β satisfying $\beta^T\beta = 1$ which maximizes $\beta^T\Sigma\beta$, i.e.,

$$\max_{\beta} \beta^T\Sigma\beta$$
$$\text{such that } \beta^T\beta = 1$$

- We can write it as

$$\phi = \beta^T\Sigma\beta - \lambda(\beta^T\beta - 1),$$

where λ is a Lagrangian multiplier. The vector of partial derivative is

$$\frac{\partial\phi}{\partial\beta} = 2\Sigma\beta - 2\lambda\beta = \mathbf{0}$$

Principal Component Analysis

- A vector β maximizing $\beta^T\Sigma\beta$ must satisfy

$$\frac{\partial\phi}{\partial\beta} = 2\Sigma\beta - 2\lambda\beta = \mathbf{0}, \textbf{i.e.},$$
$$(\Sigma - \lambda\mathbf{I})\beta = \mathbf{0}. \quad (1)$$

- In order to have the solution with condition $\beta^T\beta = 1$ we must have $(\Sigma - \lambda\mathbf{I})$ singular, i.e.,

$$|\Sigma - \lambda\mathbf{I}| = \mathbf{0}.$$

- The function $|\Sigma - \lambda\mathbf{I}|$ is a polynomial in λ of degree p, which has p roots - $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_p$.
- If we multiply (1) on the left by β^T, we obtain

$$\beta^T\Sigma\beta = \lambda\beta^T\beta = \lambda.$$

- If β satisfies (1) and $\beta^T\beta = 1$ then the $\mathbb{V}ar(\beta^T\mathbf{X}) = \lambda$.

Principal Component Analysis

- As β satisfies (1), i.e.,

$$(\Sigma - \lambda\mathbf{I})\beta = \mathbf{0},$$

and $\beta^T\beta = 1$ then the $\mathbb{V}ar(\beta^T\mathbf{X}) = \lambda$.

- Thus for the maximum variance we should use in (1) the largest root λ_1.

Principal Component Analysis

- Let $\beta_{(1)}$ be normalized solution of $(\Sigma - \lambda_1 \mathbf{I})\beta = \mathbf{0}$.
- Then $U_1 = \beta_{(1)}^T \mathbf{X}$ is a normalized linear combination with maximum variance, aka., major or first principal component.
- If $\Sigma - \lambda_1 \mathbf{I}$ is of rank $p-1$, then there is only one solution to $(\Sigma - \lambda_1 \mathbf{I})\beta = \mathbf{0}$ and $\beta^T\beta = 1$, i.e.,

$$\max_{\beta} \beta^T \Sigma \beta$$
$$\text{such that} \quad \beta^T\beta = 1$$

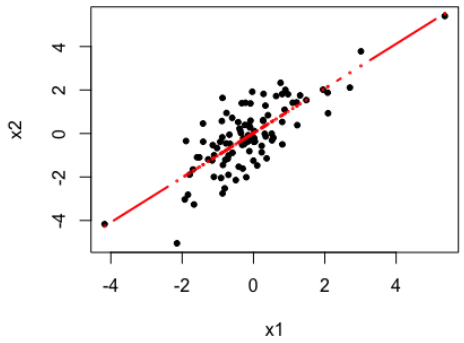
- We can write it as

$$\phi = \beta^T \Sigma \beta - \lambda(\beta^T\beta - 1),$$

where λ is a Lagrangian multiplier.

Principal Component Analysis

$U_1 = \beta_{11}x_1 + \beta_{12}x_2$



Principal Component Analysis

- Now let us find second major or principal component (PC). Second PC is a normalized combination $\beta^T\mathbf{X}$ that has maximum variance of all linear combinations uncorrelated with $U_1 = \beta_{(1)}^T$
- That is

$$\mathbf{0} = \mathbb{E}(\beta^T\mathbf{X}U_1) = \mathbb{E}(\beta^T\mathbf{X}\mathbf{X}^T\beta_{(1)}) = \beta^T\Sigma\beta_{(1)} = \lambda_1\beta^T\beta_{(1)},$$

since $\Sigma\beta_{(1)} = \lambda_1\beta_{(1)}$.

- The $\beta^T\mathbf{X}$ is orthogonal to U in both Statistical sense and Geometric sense.

Principal Component Analysis

- In statistical sense $\beta^T\mathbf{X}$ is orthogonal to U_1 means correlation between $\beta^T\mathbf{X}$ and U_1 is 0.
- In geometric sense the inner product of β and $\beta_{(1)}$ is 0. That is $\lambda_1\beta^T\beta_{(1)} = 0$ only if $\beta^T\beta_{(1)} = 0$ when $\lambda_1 \neq 0$.
- We must find a vector β satisfying $\beta^T\beta = 1$ and $\beta^T\Sigma\beta_{(1)} = 0$ which maximizes $\beta^T\Sigma\beta$, i.e.,

$$\max_{\beta} \beta^T\Sigma\beta$$
$$\text{such that} \quad \beta^T\beta = 1 \ \text{ and } \ \beta^T\Sigma\beta_{(1)} = 0$$

- We can write it as

$$\phi_2 = \beta^T\Sigma\beta - \lambda(\beta^T\beta - 1) - 2\nu_1\beta^T\Sigma\beta_{(1)},$$

where λ and ν_1 are Lagrange multipliers.

Principal Component Analysis

- The vector of partial derivative is

$$\frac{\partial \phi_2}{\partial \boldsymbol{\beta}} = 2\Sigma\boldsymbol{\beta} - 2\lambda\boldsymbol{\beta} - 2\nu_1\Sigma\boldsymbol{\beta}_{(1)} = \mathbf{0}.$$

- Multiply on left by $\boldsymbol{\beta}_{(1)}$,

$$\mathbf{0} = 2\boldsymbol{\beta}_{(1)}^T\Sigma\boldsymbol{\beta} - 2\lambda\boldsymbol{\beta}_{(1)}^T\boldsymbol{\beta} - 2\nu_1\boldsymbol{\beta}_{(1)}^T\Sigma\boldsymbol{\beta}_{(1)} = -2\nu_1\lambda_1,$$

- Let $\lambda_{(2)} = \max(\lambda_1, \lambda_2, \cdots, \lambda_p)$ such that there is a vector $\boldsymbol{\beta}_{(2)}$ satisfying

$$(\Sigma - \lambda_{(2)}\mathbf{I})\boldsymbol{\beta} = \mathbf{0}, \quad \boldsymbol{\beta}^T\boldsymbol{\beta} = 1, \quad \textit{and} \quad 0 = \boldsymbol{\beta}^T\Sigma\boldsymbol{\beta}_{(1)} = \lambda_1\boldsymbol{\beta}^T\boldsymbol{\beta}_{(1)}$$

- The corresponding linear combination $U_2 = \boldsymbol{\beta}_{(2)}\mathbf{X}$ is second major or principal component (PC).
- Eventually we can show $\lambda_{(1)} = \lambda_1$ and $\lambda_{(2)} = \lambda_2$

Principal Component Analysis

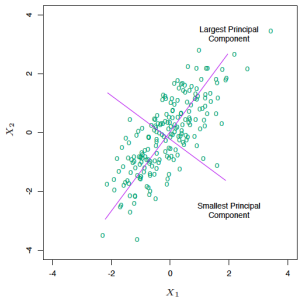


FIGURE 3.9. *Principal components of some input data points. The largest principal component is the direction that maximizes the variance of the projected data, and the smallest principal component minimizes that variance. Ridge regression projects y onto these components, and then shrinks the coefficients of the low–variance components more than the high-variance components.*

Principal Component Analysis

- The procedure continues for other PCs
- In general, let p-components random vector $\mathbf{X}$ have $\mathbb{E}(\mathbf{X}) = \mathbf{0}$ and $\mathbb{E}(\mathbf{X}\mathbf{X}^T) = \Sigma$, then $\exists$ an orthogonal linear transformation

$$\mathbf{U} = \mathbf{B}^T\mathbf{X}$$

such that the covariance matrix of $\mathbf{U}$ is $\mathbb{E}(\mathbf{U}\mathbf{U}^T) = \Lambda$ and

$$\Lambda = \begin{pmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_1 & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & \lambda_p \end{pmatrix}$$

where $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_p \geq 0$ are roots of $|\Sigma - \lambda\mathbf{I}| = 0$.

- The r^{th} column of $\mathbf{B}$, $\boldsymbol{\beta}_{(r)}$, satisfies $(\Sigma - \lambda_{(r)}\mathbf{I})\boldsymbol{\beta}_{(r)} = \mathbf{0}$.
- The r^{th} component of $\mathbf{U}$ is $U_r = \boldsymbol{\beta}_{(r)}^T\mathbf{X}$ has maximum variance of all normalized linear combinationwith $U_1, U_2, \cdots, U_{r-1}$.
- **The vector U is defined as vector of principal components of X.**

Practical Session with R

- R session on Principal Component Analysis

Feature/Variable Selection

- Suppose the feature space has p many features, i.e.,

$$X = \{X_1, X_2, \ldots, X_p\}$$

p is very large.

- We would like to drop features which have no impact on y

$$y = f(X_1, \ldots, X_q)$$

where $q << p$

- Ex: $p = 2000$ and $q = 15$

Best Subset Selection

- To perform *best subset selection*, we fit a separate least squares regression best subset for each possible combination of the p predictors.
- That is, we fit all p models that contain exactly one predictor, all ${}^pC_2 = p(p-1)/2$ models that contain exactly two predictors, and so forth. We then look at all of the resulting models, with the goal of identifying the one that is best.
- The size of the model space is $2^p - 1$.

Best Subset Selection

Algorithm:

1. Let $\mathcal{M}_0$ denote the null model , which contains no predictors. This model simply predicts the sample mean for each observation.
2. For $k = 1, 2, \cdots, p$;
   2.1 Fit all ${}^pC_k$ models that contain exactly k predictors.
   2.2 Pick the best among these ${}^pC_k$ models, and call it $\mathcal{M}_k$. Here best is defined as having the smallest RSS, or equivalently largest R^2.
3. Select a single best model from among $\mathcal{M}_0, \mathcal{M}_1, \cdots, \mathcal{M}_p$, using crossvalidated prediction error, C_p, AIC, BIC, or adjusted R^2.

Best Subset Selection

1. Though the step 2 identifies the best model (on the training data) for each subset size, in order to reduce the problem from one of 2^p possible models to one of the $p+1$ possible models.
2. The best subset selection involves fitting of 2^p models.
3. When $p = 20$, the best subset selection requires fitting $1,048,576$ models.
4. This means best subset selection is almost not possible, unless it is a toy/small dataset.

Forward stepwise selection

Algorithm:

1. Let $\mathcal{M}_0$ denote the null model , which contains no predictors.
2. For $k = 0, 1, \cdots, p-1$;
   - 2.1 Consider all $p-k$ models that augment the predictors in $\mathcal{M}_k$ with one additional predictor.
   - 2.2 Pick the best among these $p-k$ models, and call it $\mathcal{M}_{k+1}$. Here best is defined as having the smallest *RSS*, or equivalently largest R^2.
3. Select a single best model from among $\mathcal{M}_0, \mathcal{M}_1, \cdots, \mathcal{M}_p$, using crossvalidated prediction error, C_p, *AIC*, *BIC*, or adjusted R^2.

Forward stepwise selection

- Unlike best subset selection, which involved fitting 2^p models, *forward stepwise selection* involves fitting one null model, along with $p-k$ models in the k^{th} iteration, for $k = 0, \cdots, p-1$.
- This amounts to a total of $1 + \sum_{k=0}^{p-1}(p-k) = 1 + p(p+1)/2$ models.
- This is a substantial difference: when $p = 20$, *best subset selection* requires fitting $1,048,576$ models, whereas forward stepwise selection requires fitting only 211 models.
- Forward stepwise selection can be applied even in the high-dimensional setting where $n < p$; however, in this case, it is possible to construct submodels $\mathcal{M}_0, \mathcal{M}_1, \cdots, \mathcal{M}_{n-1}$ only, since each submodel is fit using least squares, which will not yield a unique solution if $p \geq n$.

Backward stepwise selection

Algorithm:

1. Let $\mathcal{M}_p$ denote the fukk model , which contains all predictors.
2. For $k = p, p-1, \cdots, 1$;
   - 2.1 Consider all k models that contain all but one predictors in $\mathcal{M}_k$, for a total of $k-1$ predictors.
   - 2.2 Pick the best among these k models, and call it $\mathcal{M}_{k-1}$. Here best is defined as having the smallest *RSS*, or equivalently largest R^2.
3. Select a single best model from among $\mathcal{M}_0, \mathcal{M}_1, \cdots, \mathcal{M}_p$, using crossvalidated prediction error, C_p, *AIC*, *BIC*, or adjusted R^2.

Backward stepwise selection

- Like forward stepwise selection, the backward selection approach searches through only $1 + p(p+1)/2$ models, and so can be applied in settings where p is too large to apply best subset selection.
- Also like *forward* stepwise selection, *backward* stepwise selection is not guaranteed to yield the *best model* containing a subset of the p predictors.

Practical Session with R

- Stepwise selection in `mtcars` dataset
- Large n small p data –`PRSA_data_analysis.R`

Why multicollinearity is a problem?

- Consider the standard linear model

$$Y = X\beta + \epsilon,$$

where $\epsilon \sim N(0, \sigma^2 I_n)$ and $n > p$

- This implies $Y \sim N(X\beta, \sigma^2 I_n)$
- The least square estimator of β is $\hat{\beta} = (X^T X)^{-1} X^T y$
- The sampling distribution of $\hat{\beta}$ is

$$\hat{\beta} \sim N(\beta, \sigma^2 (X^T X)^{-1})$$

Why multicollinearity is a problem?

- If correlation between two predictors of X is 1, that means one column is exactly dependent on other, that will result $det(X^T X) = 0$
- Heance $X^T X$ will not be invertible, (because $(X^T X)^{-1} = \frac{Adj(X^T X)}{det(X^T X)}$
- In such case unique solution does not exists.

Why multicollinearity is a problem?

- If correlation between two predictors of X is nearly 1 or -1, but not exactly 1.
- For example $cor(X_i, X_j) = 0.99$ - what happens then?
- $det(X^T X) = \delta > 0$, where δ is a very small value.
- $X^T X$ is invertible - but every element of $(X^T X)^{-1}$ will be very large.
- Unique solution $\hat{\beta}$ exists but $Cov(\hat{\beta}) = \sigma^2 (X^T X)^{-1}$ will be extremely large - so standard error will be very large.
  Hence valid statistical inference cannot be implemented.

Identify multicollinearity

- variance inflation factor (VIF) is an index which indicates how much a prdictor is contributing towards the multicollinearity problem
- Analyze the magnitude of multicollinearity by considering the size of the VIF($\hat{\beta}_i$) A rule of thumb is that if VIF($\hat{\beta}_i$) >10 then multicollinearity is high.
- A cutoff of 5 is also commonly used.

Practical Session with R

- Checkout multicollinarity with `vif`

Class of Ill-Posed Problems

- A class of problem is known as ill-posed problem - if either of the following feature exits
  1. Solution does not exist
  2. Solution exists - but computationally not feasable
  3. Solution exists - but unreliable

1 Problem of variable selection in "large p, small n" setup considered as ill-posed problems

2 Problem of variable selection in large p is considered as ill-posed problems for model complexity.

3 Problem of multicollinearity also considered ill-posed problems

Different Methods for multicollinearity and feature selection

- Ridge Regression takes care of multicollinearity (Hoerl and Kennard (1970))
- LASSO Regression takes care of feature selection (Tibshirani, 1996)
- ElasticNet Regression takes care of feature selection (Zou and Hastie, 2006)

Penalizing Objective Function

- The class of functions is controlled by explicitly penalizing $RSS(f)$ with a roughness penalty

$$PL_2 = PRSS(f;\lambda) = RSS(f) + \lambda P(f)$$

- The amount of penalty is controlled by $\lambda \geq 0$.
- $\lambda = 0$ means no-penalty
- Typically λ is estimated from data.

  As we take $f(X) = X\beta$

$$\begin{aligned} PL_2 = PRSS(\beta;\lambda) &= RSS(\beta) + \lambda P(\beta) \\ &= (y - X\beta)^T(y - X\beta) + \lambda P(\beta) \\ &= ||y - X\beta||^2 + \lambda P(\beta) \\ &= L_2(y, X\beta) + \lambda P(\beta) \end{aligned}$$

Penalizing Objective Function

- What about penalizing L_1-norm error? Can we penalize L_1-norm error?
- Yes we can. The model is:

$$PL_1 = ||y - X\beta||_1 + \lambda P(f)$$

- For now we focus on L_2-norm error.

What penalty to choose?

- For the model,

$$PL_2^2(\beta) = (y - X\beta)^T(y - X\beta) + \lambda P(\beta),$$

  one possible choice is L_2-norm penalty.
- That is

$$P(\beta) = (\beta - \beta_0)^T(\beta - \beta_0)$$

- Typical case $\beta_0 = 0$ and the penalty looks like

$$P(\beta) = \beta^T\beta$$

Analysis with L_2-penalty

- We want to minimize the L_2-penalized loss

$$PL_2^2(\beta) = (y - X\beta)^T(y - X\beta) + \lambda\beta^T\beta,$$

  and we can obtained the Ridge solution as,

$$\hat{\beta}_{Ridge} = \text{argmin}_\beta \left[(y - X\beta)^T(y - X\beta) + \lambda\beta^T\beta\right]$$

- An equivalent way to write the ridge problem is

$$\begin{aligned} \hat{\beta}_{Ridge} &= \text{argmin}_\beta \left[(y - X\beta)^T(y - X\beta)\right] \\ &\quad \text{subject to } \beta^T\beta \leq t, \end{aligned}$$

  which makes explicit the size constraint on the parameters.
- There is a one-to-one correspondence between the parameters λ and t.

Ridge Regression

- Solving the following minimization problem,

$$\hat{\beta}_{Ridge} = \text{argmin}_{\beta} \left[(y - X\beta)^T (y - X\beta) + \lambda \beta^T \beta \right],$$

we have the Ridge solution as

$$\hat{\beta}_{Ridge} = (X^T X + \lambda \mathbf{I})^{-1} X^T y,$$

where $\mathbf{I}$ is the $p \times p$ identity matrix.

- Ridge solution is a special case of Tikohonov solution.

Bayesian Interpretation of Ridge Regression

- The $PL_2^2(\beta)$ can be presented as

$$-\frac{1}{2\sigma^2} PL_2(\beta) = -\frac{1}{2\sigma^2}(y - X\beta)^T (y - X\beta) - \frac{1}{2\sigma^2}\beta^T \beta$$

If we take exponent, then we have

$$\exp\left[-\frac{1}{2\sigma^2} PL_2^2(\beta) \right] = \underbrace{\exp\left[-\frac{1}{2\sigma^2}(y - X\beta)^T (y - X\beta) \right]}_{L_2 \text{ norm part}} \cdot \underbrace{\times \exp\left[-\frac{1}{2\sigma^2}\beta^T \beta \right]}_{\text{penalty part}}$$

Bayesian Interpretation of Ridge Regression

- As we have

$$y = X\beta + \epsilon, \quad \epsilon \sim N(0, \sigma^2 \mathbf{I})$$

That is

$$y \sim N(X\beta, \sigma^2 \mathbf{I})$$

- Negative log-likelihood of $\beta|\sigma^2$ is

$$L = \frac{1}{2\sigma^2}(y - X\beta)^T (y - X\beta) \propto \frac{1}{2\sigma^2}||y - X\beta||^2$$

Bayesian Interpretation of Ridge Regression

- Consider prior on β

$$\beta|\sigma^2 \sim N(0, \sigma^2 \mathbf{I})$$

- The negative log-posterior of β is

$$L(\beta|y, X, \sigma^2) \propto \frac{1}{2\sigma^2}\left\{ ||y - X\beta||^2 + ||\beta||^2 \right\} \propto PL_2(\beta)$$

Bayesian Interpretation of Ridge Regression

- Exponent of L_2-norm error is the likelihood model of data, i.e.,

$$y|X, \sigma^2 \sim N(X\beta, \sigma^2 \mathbf{I}_n)$$

- Exponent of the L_2-penalty is equivalent to Gaussian prior on β, i.e.,

$$\beta|\sigma^2 \sim N(0, \lambda^{-2})$$

- The posterior distribution over β is

$$\beta|\sigma^2, y, X \sim N\big(\hat{\beta}_{Ridge}, \sigma^2(X^T X + \lambda \mathbf{I})^{-1}\big),$$

where

$$\hat{\beta}_{Ridge} = (X^T X + \lambda \mathbf{I})^{-1} X^T y,$$

- The solution $\hat{\beta}_{Ridge}$ is the **posterior mode**

Practical Session with R

- R session on Ridge Regression

LASSO

- Least Absolute Shrinkage and Selection Operator (LASSO)
- The lasso is a shrinkage method like ridge, with subtle but important differences.
- The lasso estimate is defined as

$$\hat{\beta}_{lasso} = \text{argmin}_\beta \left[(y - X\beta)^T (y - X\beta) + \lambda ||\beta||_1\right]$$

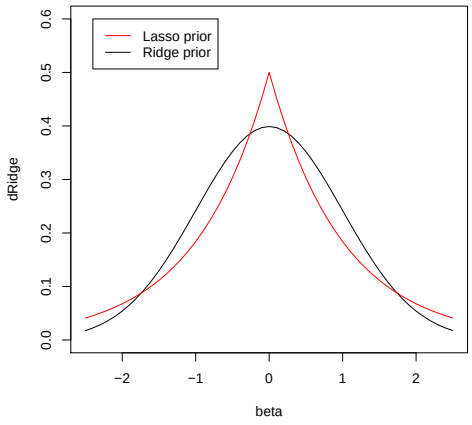
- Equivalently can be expressed as

$$\begin{aligned} \hat{\beta}_{lasso} &= \text{argmin}_\beta (y - X\beta)^T (y - X\beta) \\ &\text{subject to} \sum_{j=1}^{p} |\beta_j| \leq t \end{aligned}$$

Remark

- L_2 penalty or Ridge penalty on β corresponds to Gaussian prior (aka. Ridge prior)
- L_1 penalty or LASSO penalty on β corresponds to Laplace prior (aka. LASSO prior)
- LASSO doenot have closed form solution like Ridge.
- Computing the lasso solution is a quadratic programming problem.
- Efficient algorithms are available for computing the entire path of solutions as λ is varied, with the same computational cost as for ridge regression.

Remark



Remark

- Because of the nature of the constraint, making t sufficiently small will cause some of the coefficients to be exactly zero.
- Thus the lasso does a kind of continuous subset selection.
- Ridge takes care of multicollinearity kind of issues.
- compromise between ridge and lasso was give Zou and Hastie (2005), known as Elastic Net penalty

$$P_{EN}(\beta) = \lambda \sum_{j=1}^{p} (\alpha \beta_j^2 + (1-\alpha)|\beta_j|)$$

LASSO, Ridge and Elastic Net

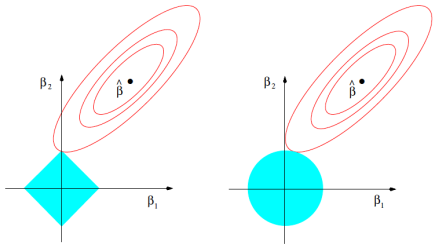
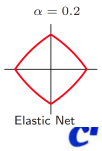


FIGURE 3.11. *Estimation picture for the lasso (left) and ridge regression (right). Shown are contours of the error and constraint functions. The solid blue areas are the constraint regions* $|\beta_1| + |\beta_2| \le t$ *and* $\beta_1^2 + \beta_2^2 \le t^2$*, respectively, while the red ellipses are the contours of the least squares error function.*



Practical Session with R

- R session on LASSO and ElastiNet

Non-linear Regression Basis Functions

- Consider i^{th} record
$$y_i = f(t_i) + \epsilon_i$$
represents $f(t)$ as
$$f(t) = \sum_{j=1}^{K} \beta_j \phi_j(t) = \boldsymbol{\phi\beta}$$
we say ϕ is a basis system for $f(t)$.

cmi

Representing Functions with Basis Functions

- Terms for curvature in linear regression
$$y_i = \beta_1 + \beta_2 t_i + \beta_3 t_i^2 + \ldots + \epsilon_i$$
implies
$$f(t) = \beta_1 + \beta_2 t + \beta_3 t^2 + \ldots$$

cmi

Fourier Basis

- sine cosine functions of incresing frequencies

  $y_i = \beta_1 + \beta_2 \sin(\omega t) + \beta_3 \cos(\omega t) + \beta_4 \sin(2\omega t) + \beta_5 \cos(2\omega t) \ldots + \epsilon_i$
- constant $\omega = 2\pi/P$ defines the period P of oscillation of the first sine/cosine pair.
- $\boldsymbol{\phi} = \{1, \sin(\omega t), \cos(\omega t), \sin(2\omega t), \cos(2\omega t) \ldots\}$
- $\beta^T = \{\beta_1, \beta_2, \beta_3, \ldots\}$

$$y = \boldsymbol{\phi\beta} + \epsilon$$

cmi

Other Basis

- **Exponential Basis** $\phi = \{1, e^{\lambda_1 t}, e^{\lambda_2 t} \ldots\}$
- **Gaussian Basis**
  $\phi = \{1, \exp(-\lambda(t_1 - c)^2), \exp(-\lambda(t_2 - c)^2) \ldots\}$
- **Basis corresponds to Spline Regression**
$$y = \beta_0 + \sum_{k=1}^{K} \beta_k (t - \xi_k)_+^D + \ldots \epsilon$$
  $\phi = \{1, (t - \xi_1)_+^D, (t - \xi_2)_+^D \ldots\}$

cmi

Functional Estimation/Learning

- We are writing the function with its basis expansion

$$y = \phi\beta + \epsilon$$

- Lets assume basis ϕ is fully known
- Problem is β is unknown - hence we estimate β.

Bayesian method

- Model:

$$y = f(t) + \epsilon$$

  - $\epsilon \sim \mathbf{N}(0, \sigma^2\mathbf{I}) \implies y \sim \mathbf{N}(f(t), \sigma^2\mathbf{I})$

$$f(t) = \phi\beta = \sum_{k=1}^{\infty} \phi_k(t)\beta_k$$

,

- β is unknown and want to estimate

  Assuming β's are uncorrelated random variable and $\phi_k(t)$ are known deterministic real-valued functions.
- Then due to **Kosambi-Karhunen-Loeve** theorem, we can say that $f(t)$ is a stochastic process.

Gaussian Process Prior

- As $f(t)$ is a stochastic process if we assume $\beta \sim \mathbf{N}(0, \sigma^2\mathbf{I})$ then $f(t) = \phi\beta$ follow Gaussian process.
- Since $f(t)$ is unknown function; therefore induced process on $f(t)$ is known as '**Gaussian Process Prior**'.

  Prior on β:

$$p(\beta) \propto \exp\left(-\frac{1}{2\sigma^2}\beta^T\beta\right)$$

  Induced Prior on $f = \phi\beta$:

$$p(f) \propto \exp\left(-\frac{1}{2\sigma^2}\beta^T\phi^T\mathbf{K}^{-1}\phi\beta\right)$$

Gaussian Process Prior

- The prior mean and covariance of $f(t)$ are given by

$$\mathbf{E}[f(t)] = \phi(t)E[\beta] = \phi\beta_0$$

$$\begin{aligned}\mathbf{cov}[f(t)] &= \mathbf{E}[f(t).f(t')^T] = \phi(t).\mathbf{E}[\beta.\beta^\mathbf{T}]\phi(t')^T \\ &= \sigma^2\phi(t).\phi(t')^T = \mathbf{K}(t,t')\end{aligned}$$

$$f(t) \sim \mathbf{N}_n(\phi(t)\beta_0, \mathbf{K}(t,t')), \quad \epsilon \sim \mathbf{N}_p(0, \sigma^2\mathbf{I})$$

$$y(t) \sim \mathbf{N}_n\Big(\phi(t)\beta_0, \mathbf{K}(t,t') + \sigma^2\mathbf{I}\Big)$$

Gaussian Process Regression

- The estimated value of y for a given t_* is the mean (expected) value of the functions sampled from from the posterior at that value of t_*.
- Suppose $\mu(t) = \phi(t)\beta_0$, then expected value of the estimate at a given t_* is given by

$$\begin{aligned}\hat{f}(t_*) &= \mathbf{E}(f(t_*)|t, Y) \\ &= \mu(t) + \mathbf{K}(t_*, t).\underbrace{[\mathbf{K}(t,t) + \sigma^2.\mathbf{I}]^{-1}}_{\text{Matrix of order } n}.(\mathbf{y} - \mu(t))\end{aligned}$$

- The time complexity of the matrix inversion is $\mathcal{O}(n^3)$

Likelihood Method: Gaussian Process Prior Model

- Data model:

$$y(t) \sim \mathbf{N}_n\Big(\phi(t)\beta_0, \mathbf{K}_{\alpha,\rho}(t,t') + \sigma^2\mathbf{I}\Big)$$

- Static or Hyperparameters: $\boldsymbol{\theta} = \{\beta_0, \alpha, \rho, \sigma^2\}$
- Likelihood function:

$$f(\beta|y, \phi, \sigma^2) \propto (\sigma^2)^{-p/2} \exp\left(-\frac{1}{2\sigma^2}(y-f)^T[\mathbf{K}+\sigma^2\mathbf{I}]^{-1}(y-f)\right)$$

- Negative Log-likelihood function:

$$l(\beta) \propto \frac{1}{2\sigma^2}(y - \phi\beta)^T[\mathbf{K} + \sigma^2\mathbf{I}]^{-1}(y - \phi\beta)$$

Gaussian Process Prior Model

- Negative log-posterior:

$$p(\beta) \propto \frac{1}{2\sigma^2}\Big((y-\phi\beta)^T[\mathbf{K}+\sigma^2\mathbf{I}]^{-1}(y-\phi\beta)+\beta^T\phi^T\mathbf{K}^{-1}\phi\beta\Big)$$

- Hence the induced penalty matrix in the Gaussian process prior is identity matrix
- It looks like weighted least square method with L_2 penalty on β
- Still hyperparameters: $\boldsymbol{\theta} = \{\beta_0, \alpha, \rho, \sigma^2\}$ are unknown.

Gaussian Process Prior Model

- One can use **optimization** routine to estimate the MLE
- However, often divergence of **optimization** routine for MLE is being reported.
- Berger *et al.* (2001) *JASA* showed that the improper prior for GP prior model would lead to improper posterior distribution
- This result implies posterior mode does not exist.
- We know under improper prior, the posterior mode is exactly same as MLE.
- Perhaps this is a Bayesian interpretation of why **optimization** routine for MLE faces convergence issues.

What prior to choose? And Why?

- Popular choice

$$\begin{aligned} \boldsymbol{\beta_0} &\sim N_p(0, K), \quad K \text{ is large} \\ (\alpha, \rho, \sigma) &\sim InvGamma(\epsilon, \epsilon), \quad 0 < \epsilon < 1. \end{aligned}$$

- Robust Choice

$$\begin{aligned} \boldsymbol{\beta_0} &\sim N_p(0, \tau), \\ \tau &\sim Half - Cauchy(0, 1), \\ (\alpha, \rho, \sigma) &\sim InvGamma(\epsilon, \epsilon), \quad 0 < \epsilon < 1. \end{aligned}$$

Ref: Gelman etal. (2006), Carvalho (2008), Dautta and Ghosh (2012) studied the robustness of this prior in regular regression model, but not in fda

cmi

Experiment with GP Regression

- Model:

$$y = \frac{\sin(x)}{x} + \epsilon,$$

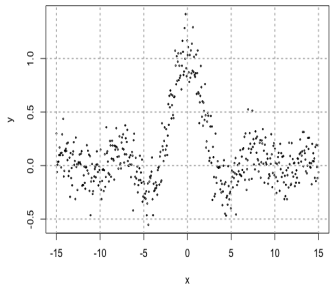
where $\epsilon \sim N(0, \tau)$.

- Simulate data from the above model and pretend we don't know the true function.
- **Objective** is to estimate/learn the function.

cmi

Experiment with GP Regression

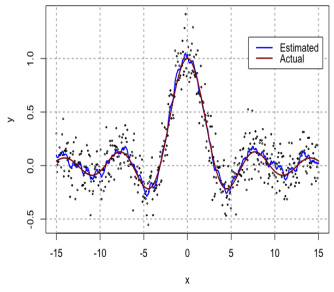
Objective is to estimate/learn the function.



cmi

Experiment with GP Regression

Objective is to estimate/learn the function.



cmi

Practical Session with R

- Open the `GP_reg_ex1.R` to implement the GP Regression.

Performance Measure

Overfitting

- Model is too specific
  - Tailored to fit anomalies in training data
  - Performs suboptimally on general data
- Variable Selection
  - Forward selection:
    - In this approach, one adds variables to the model one at a time.
    - The most significant of these variables is added to the model, so long as it's P-value is below some pre-set level.
  - Backward selection:
    - one starts with fitting a model with all the variables of interest. Then the least significant variable is dropped
    - continue by successively re-fitting reduced models and applying the same rule until all remaining variables are statistically significant.

Evaluating a Regression Model . . .

- Who provides the "oracle" to validate answers?
- *Holdout sets (aka. Test Set)*
  - Exclude a random sample of training data
  - Build classifier on remaining data, check answers on holdout set
  - Suitable if we have a large volume of training data
- *Cross validation*
  - Systematically exclude $1/n$ of training data
  - Build classifier on remaining data and check answers on excluded set
  - Repeat n times to span entire training data
  - Aggregate the scores obtained

Evaluating Regression Performance . . .

- **Correlation Coefficient**: This is how well the predictions are correlated or change with the actual output value. A value of 0 is the worst and a value of 1 is a perfectly correlated set of predictions.
- **Root Mean Squared Error**: This is the average amount of error made on the test set in the units of the output variable. This measure helps you get an idea on the amount a given prediction may be wrong on average.
- **Information Criterion**:
  - Akaike information criterion (AIC)
  - Bayesian information criterion

Thank You

sourish@cmi.ac.in