Contents

Contents

Contents

Contents

Contents

Contents

Contents

Contents

Contents

Contents

Outline

1. Linear Models
   - Framework and notations
   - Toy example : Housing price prediction
   - Toy example (from the book) : Sales prediction
   - Linear model properties
   - Fitting the regression
   - Solving the regression analytically
   - Gradient descent principles
   - Three general notions/definitions
   - Other algorithms : LDA, Polynomial expansion
   - Summary

Starting point :

- Outcome measurement Y (also called dependent variable, response, target, label);
- Vector of p predictor measurements X_i (also called inputs, regressors, covariates, features, independent variables). X is a matrix of dimension (N,p), where N is the number of measurements;
- In the **regression problem**, Y is quantitative (e.g price, sales, categories, blood pressure);
- We have training data (x_1,y_1),..., (x_N,y_N). These are observations (examples, instances) of these measurements.

Outline

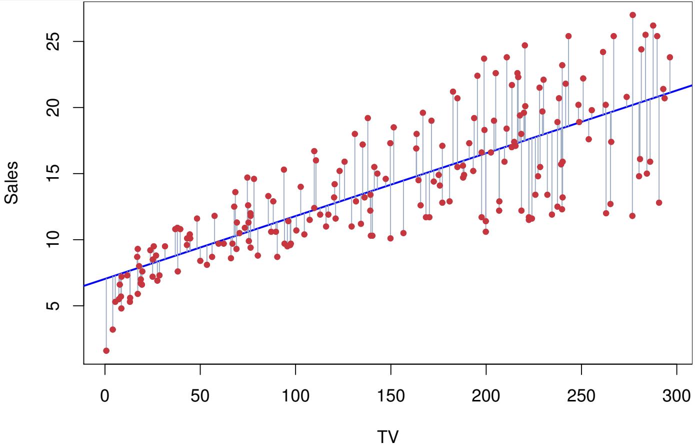
1. Linear Models
   - Framework and notations
   - Toy example : Housing price prediction
   - Toy example (from the book) : Sales prediction
   - Linear model properties
   - Fitting the regression
   - Solving the regression analytically
   - Gradient descent principles
   - Three general notions/definitions
   - Other algorithms : LDA, Polynomial expansion
   - Summary

Linear Regression model with one variable

- Pattern / Model : $Y_i = \beta_0 + \beta_1 X_i + \epsilon_i$ for $0 \leq i \leq N$
  $Price_{house30} = \beta_0 + \beta_1 Surface_{House30} + \epsilon$
- In fact, we could imagine the price depends from severals factors, so we come with Linear Regression with several variables :
  $Price_{house30} = K + \beta_1 Surface_{House30} + \beta_2 NbOfRooms_{House30} + \beta_3 Location_{House30} + \epsilon$
- In general :
  $Y_i = h(X^i) = \beta_0 + \beta_1 X_{i,1} + \beta_2 X_{i,2} + ... + \beta_i X_{i,p} + \epsilon$
- Traditionally p is called the number of features. We will use matrix notation, so there will be double indices.

Outline

1. Linear Models
   - Framework and notations
   - Toy example : Housing price prediction
   - Toy example (from the book) : Sales prediction
   - Linear model properties
   - Fitting the regression
   - Solving the regression analytically
   - Gradient descent principles
   - Three general notions/definitions
   - Other algorithms : LDA, Polynomial expansion
   - Summary



$$Sales \approx \beta_0 + \beta_1 TV$$

Outline

- The parameters in the linear regression model are very easy to interpret.
- $\beta_j, 1 \leq j \leq p$ is the average increase in Y when X_j is increased by one and all other X_i are held constant.
- Vocabulary : β_0 is the intercept (i.e. the average value for Y if all the Xs are zero), β_j is the slope for the jth variable X_j

Outline

- Historical method : Least Square Regression ;
- Modern method : numerical iterative process : gradient descent and a huge family of similar algorithms (Maths : (Numerical)(Convex or not) Optimization.

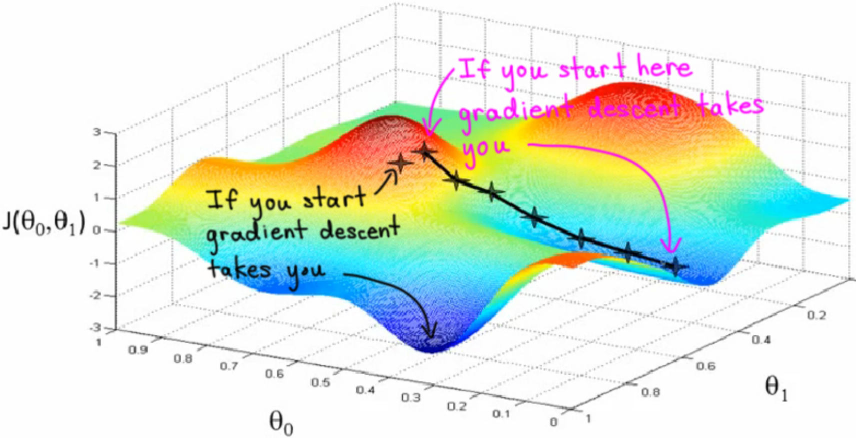
Cost function, traditionally noted $J(\beta)$ is given by :
$J(\beta) = \frac{1}{2N} \sum_{i=1}^{N} (h(X^i) - Y_i)^2$
Recall naming and indices are not universal, you will find sometimes n instead of N or p, or m instead of N, etc.

Outline

- We want to minimize the quantity $\sum_{i=1}^{N}(h(X^i) - Y_i)^2$ called MSE Mean Square Error
- Solution in one dimension : write partial derivatives in β_0 and β_1 of the cost function. To be done
- Solution in p dimension : Matrix

- In one dimension you will derive a 2x2 linear system :
$$\begin{cases} \beta_0 \sum_{i=1}^N x_i & +\beta_1 \sum_{i=1}^N x_i^2 = \sum_{i=1}^N x_i y_i, \\ N\beta_0 & +\beta_1 \sum_{i=1}^N x_i = \sum_{i=1}^N y_i. \end{cases}$$
- To recall : you can shift the variables (x_i, y_i) to be centred on the mean, new variables $(\bar{x_i}, \bar{y_i})$, verifies $\sum_i \bar{x_i} = 0$, $\sum_i \bar{y_i} = 0$ it gives directly the well-known slope coefficient $\beta_1 = \frac{\sum_{i=1}^N x_i \bar{y_i}}{\sum_{i=1}^N \bar{x_i}^2}$
- Solution in p dimensions : Matrix X including a column vector of 1, verify
$(X^T X)\beta = X^T Y$ gives $\beta = (X^T X)^{-1}(X^T Y)$ assuming that the square matrix $X^T X$ is invertible. This is the **Normal Equation**.

Outline


If you start here gradient descent takes you
If you start gradient descent takes you
$J(\theta_0,\theta_1)$
θ_0
θ_1
3
2
1
0
-1
-2
-3
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0.2
0.4
0.6
0.8
1

- Levels on a curve or surface, directions of steepest descent
- Stochastic approach steps :
  - Initialize β_0, β_1, ... β_p
  - Compute the new direction : $\beta_j := \beta_j - \alpha_j \frac{\partial J(\beta)}{\partial \beta_j}$, for $j = 0, ..., p$
  - Evaluate $J(\beta)$ and iterate
- Compare the complexity of the 2 methods

Outline

1. Linear Models
   - Framework and notations
   - Toy example : Housing price prediction
   - Toy example (from the book) : Sales prediction
   - Linear model properties
   - Fitting the regression
   - Solving the regression analytically
   - Gradient descent principles
   - **Three general notions/definitions**
   - Other algorithms : LDA, Polynomial expansion
   - Summary

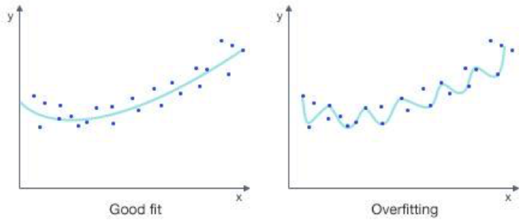
Bias-Variance definition

- Informal definition
- So-called Bias-Variance trade-off

Train vs Test set error

- Graphical model tuning

Overfitting



- Polynomial expansion (chapter Beyond Linearity)

Outline

1. Linear Models
   - Framework and notations
   - Toy example : Housing price prediction
   - Toy example (from the book) : Sales prediction
   - Linear model properties
   - Fitting the regression
   - Solving the regression analytically
   - Gradient descent principles
   - Three general notions/definitions
   - Other algorithms : LDA, Polynomial expansion
   - Summary

- LDA
- Polynomial expansion : beyond the linearity (another chapter)

Outline

- Linearity is a limitation but solving principles are more general ;
- Introducing complexity/flexibility of a model vs interpretability