Regression

Regression is a method for learning the relationship between a <u>response variable</u> Y and a <u>predictor variable</u> X. The relationship is summarized through the <u>regression function</u>, $\boxed{r(x) = \mathbb{E}(Y \mid X = x)}$

Goals

1.) Learn the regression function, $r(x)$, from data of the form: $(X_1, Y_1), (X_2, Y_2), \ldots, (X_n, Y_n)$

2.) Explain the relationship

3.) Use your learned regression function to predict the value of Y given $X = x$

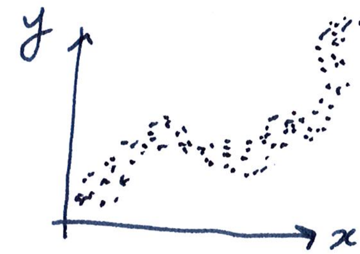
After gathering the data, we can first look at plots to decide the form of $r(x)$

$r(x) = \beta_0 + \beta_1 x$

"linear form"

$r(x) = \beta_0 + \beta_1 x + \beta_2 x^2$

"quadratic form"



"complicated form"

We could also use multiple predictors and have $r(x) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p$

"multiple linear regression"

we will focus on "simple linear regression"
(one predictor, $r(x)$ is assumed to have a linear form)

Model

we have data of the form $(X_1, Y_1), \dots (X_n, Y_n)$

$$Y_i = \underbrace{\beta_0 + \beta_1 X_i}_{\substack{r(X) \equiv \text{ the} \\ \text{Relationship} \\ \text{between } X \text{ \& } Y}} + \underbrace{\epsilon_i}_{\substack{\text{Random} \\ \text{error}}} \quad \text{where } \epsilon_i \sim N(0, \sigma^2)$$

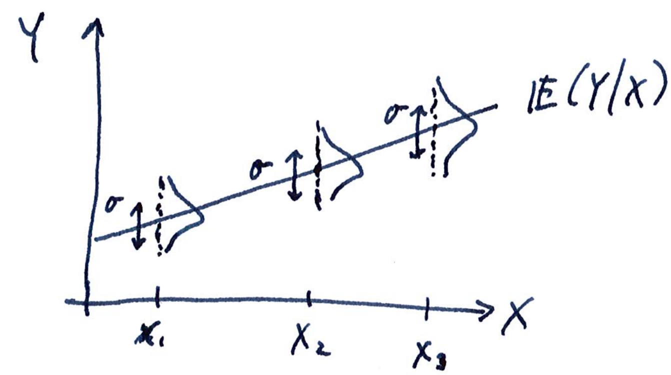
In other words we write

$$Y_i / X_i \sim N(\beta_0 + \beta_1 X_i, \sigma^2) \quad \text{where}$$

$$\mathbb{E}(Y_i/X_i) = \beta_0 + \beta_1 X_i$$

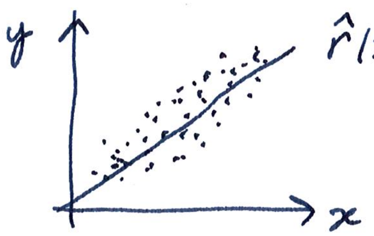
$$Var(Y_i/X_i) = \sigma^2$$

Picture



At a given X, there is a population of Y's that is normally distributed with mean $\beta_0 + \beta_1 X_i$ and Variance σ^2. Goal: learn $\mathbb{E}(Y/X) = \beta_0 + \beta_1 X$

In practice, we have a sample from the model and use the data to estimate the regression function



$\hat{r}(x) = \hat{\beta}_0 + \hat{\beta}_1 x$

we "fit" this line to the data using the method of least squares

For a given value x_i, we have

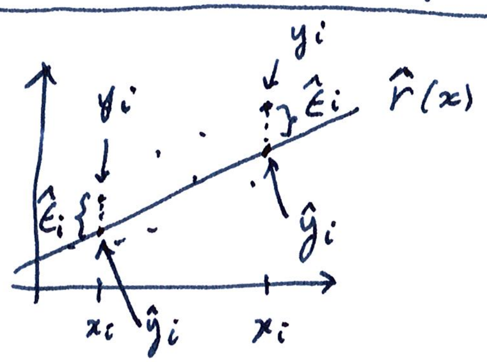
y_i = observed value (from the data pair (x_i, y_i))

$\hat{y}_i$ = predicted/fitted value ($\hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_i$)

define a residual as:

$$\hat{\varepsilon}_i = y_i - \hat{y}_i$$

Least squares finds $\hat{\beta}_0$ and $\hat{\beta}_1$ to minimize the residual sum of squares $\Rightarrow$ minimize $\sum_{i=1}^{n} (y_i - \hat{y}_i)^2$



Finding the line to minimize the residual sum of squares is a calculus problem. Given our data $(x_1, y_1) \ldots (x_n, y_n)$ it turns out:

$$\boxed{\hat{\beta}_1 = \frac{\sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n} (x_i - \bar{x})^2} \quad \text{and} \quad \hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x}}$$

yielding the least squares regression line:

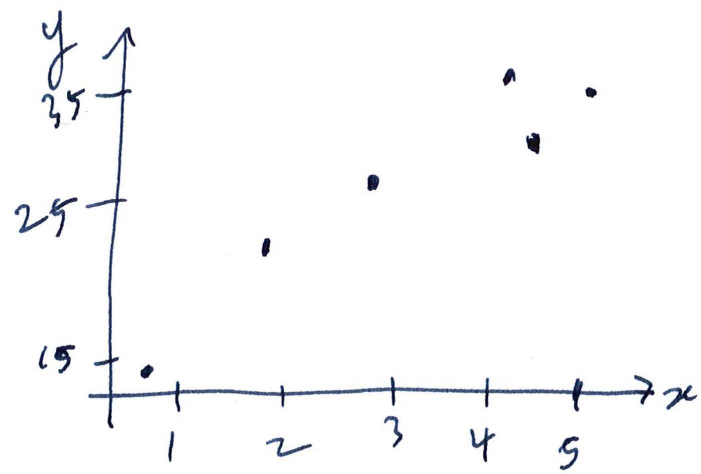
$$\boxed{\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x}$$

Example

For 6 fixed x values, I simulated 6 Y values from the model $Y = 10 + 5x + \epsilon$ $\epsilon \sim N(0,(1.5)^2)$

data

x	y
.66	14.36
4.36	34.34
2.88	25.54
4.85	34.08
4.42	29.68
1.96	20.54



Summary statistics

$$\bar{x} = \frac{\sum x_i}{6} = 3.188, \quad \bar{y} = \frac{\sum y_i}{6} = 26.09$$

$$\sum(x_i - \bar{x})^2 = 13.65, \quad \sum_i (x_i - \bar{x})(y_i - \bar{y}) = 64.626$$

$$\hat{\beta}_1 = \frac{64.626}{13.65} = 4.73, \quad \hat{\beta}_0 = 26.09 - (4.73)(3.188)$$
$$= 11.01$$

So the least squares equation is

$$\boxed{\hat{y} = 11.01 + 4.73x}$$

what can we do with our equation?

1.) Explain the relationship between X and Y

$\hat{\beta}_1$ (the slope of our line) tells us the expected change in Y for a unit change in X.

we could also make intervals or do Hyp tests for β_1

$H_0: \beta_1 = 0$ vs $H_1: \beta_1 \neq 0$

This tests ~~what~~ whether a line with slope predicts better than a flat line

If $\beta_1 = 0 \Rightarrow$ model is $Y = \beta_0 + \epsilon$

If $\beta_1 \neq 0 \Rightarrow$ model is $Y = \beta_0 + \beta_1 x + \epsilon$

In other words, is X a good predictor of Y?

2.) Prediction: Simply plug in values of x into our fitted equation to predict the value of Y.

$$\Rightarrow \hat{Y} = \hat{\beta}_0 + \hat{\beta}_1 x$$

plug in x

predicted Y

Example

universities have models from past student data. Suppose we want to predict the Freshman GPA of applicants based on their ACT score. From past data we have built a model say:

$\hat{Y} = .796 + .094x$ where x = ACT score, $\hat{Y}$ = predicted GPA

Two applicants have ACT scores of 32 and 27

$\hat{Y}_1 = .796 + .094(32) = 3.804$

$\hat{Y}_2 = .796 + .094(27) = 3.334$

How good are our predictions?

A common measure is the root mean square error (RMSE)

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

"Average squared distance our predictions are from the observed values"

RMSE is a (biased) estimator of σ in our model

For a data set and equation we could calculate RMSE.

$(x_1, y_1) \ldots (x_n, y_n)$ we could then plug the x's in are equation to get the $\hat{y}$'s and then compute RMSE.

observed y's "Reality

<u>Problem</u>

This is not the best approach, because by construction our line should match the data closely. Thats what least squares does, it minimizes $\sum(y_i - \hat{y}_i)^2$.

we Should test our predictions on a "Test set"
A set of data not used to build our prediction equation.

Data (full set)

x	y	
x_1	y_1	training data
$\vdots$	$\vdots$	
x_{100}	y_{100}	Test data
$\vdots$	$\vdots$	
x_{150}	y_{150}	

we can build our model using the training data and test it on the test data.

- plug in x's in test data into model to get $\hat{y}$'s.
- we have the true y values
- calculate RMSE

If we were comparing models we would prefer the model with the smallest "test RMSE"

Application

Netflix prize

Netflix Prize

From Wikipedia, the free encyclopedia

The **Netflix Prize** was an open competition for the best collaborative filtering algorithm to predict user ratings for films, based on previous ratings without any other information about the users or films, i.e. without the users or the films being identified except by numbers assigned for the contest.

Problem and data sets

Netflix provided a *training* data set of 100,480,507 ratings that 480,189 users gave to 17,770 movies. Each training rating is a quadruplet of the form `<user, movie, date of grade, grade>`. The user and movie fields are integer IDs, while grades are from 1 to 5 (integral) stars

In summary, the data used in the Netflix Prize looks as follows:

- Training set (99,072,112 ratings not including the probe set, 100,480,507 including the probe set)
  - Probe set (1,408,395 ratings)
- Qualifying set (2,817,131 ratings) consisting of:
  - Test set (1,408,789 ratings), used to determine winners
  - Quiz set (1,408,342 ratings), used to calculate leaderboard scores

Submitted predictions are scored against the true grades in terms of **root mean squared error (RMSE)**, and the goal is to reduce this error as much as possible.

There was some controversy as to the choice of RMSE as the defining metric. Would a reduction of the RMSE by 10% really benefit the users? It has been claimed that even as small an improvement as 1% RMSE results in a significant difference in the ranking of the "top-10" most recommended movies for a user.

Prizes

Prizes were based on improvement over Netflix's own algorithm, called *Cinematch*, or the previous year's score if a team has made improvement beyond a certain threshold. A trivial algorithm that predicts for each movie in the quiz set its average grade from the training data produces an RMSE of 1.0540. Cinematch uses "straightforward statistical linear models with a lot of data conditioning"

Using only the training data, Cinematch scores an RMSE of 0.9514 on the quiz data, roughly a 10% improvement over the trivial algorithm. Cinematch has a similar performance on the test set, 0.9525. **In order to win the grand prize of $1,000,000, a participating team had to improve this by another 10%, to achieve 0.8572 on the test set.[2] Such an improvement on the quiz set corresponds to an RMSE of 0.8563.**

Once one of the teams succeeded to **improve the RMSE by 10% or more**, the jury would issue a *last call*, giving all teams 30 days to send their submissions. Only then, the team with best submission was asked for the algorithm description, source code, and non-exclusive license, and, after successful verification; declared a grand prize winner

On September 18, 2009, Netflix announced team "BellKor's Pragmatic Chaos" as the prize winner **(a Test RMSE of 0.8567),** and the prize was awarded to the team in a ceremony on September 21, 2009.[24] "The Ensemble" team had matched BellKor's result, but since BellKor submitted their results 20 minutes earlier, the rules award the prize to BellKor.