Linear Regression

Sujay Sanghavi

Supervised Learning

Regression

- Targets y real-valued.
- Make $f : \mathcal{X} \to \mathcal{Y}$ that outputs real values.
- Error metric:

$$\sum_i \left(y^{(i)} - f\left(x^{(i)}\right) \right)^2$$

  or similar
- Eg: Credit scoring: predict score for a person

Classification

- Targets y take finite # values.
- Make $f : \mathcal{X} \to \mathcal{Y}$, which are "decision regions"
- Error metric:

$$\sum_i \mathbf{1}\left\{ y^{(i)} \neq f\left(x^{(i)}\right) \right\}$$

- Eg: Credit scoring: accept or reject a loan application

Example

Credit scoring: Determining credit-worthiness based on:

Variable Name	Description	Type
SeriousDlqin2yrs	Person experienced 90 days past due delinquency or worse	Y/N
RevolvingUtilizationOfUnsecuredLines	Total balance on credit cards and personal lines of credit except real estate and no installment debt like car loans divided by the sum of credit limits	percentage
Age	Age of borrower in years	integer
NumberOfTime30-59DaysPastDueNotWorse	Number of times borrower has been 30-59 days past due but no worse in the last 2 years	integer
NumberOfTime60-89DaysPastDueNotWorse	Number of times borrower has been 60-89 days past due but no worse in the last 2 years	integer
NumberOfTime90DaysPastDueNotWorse	Number of times borrower has been 90 days or more past due	integer
DebtRatio	Monthly debt payments, alimony, living costs divided by monthly gross income	percentage
MonthlyIncome	Monthly income	real
NumberOfOpenCreditLinesAndLoans	Number of Open loans (installment like car loan or mortgage) and Lines of credit (e.g. credit cards)	integer
NumberRealEstateLoansOrLines	Number of mortgage and real estate loans including home equity lines of credit	integer
NumberOfDependents	Number of dependents in family excluding themselves (spouse, children etc.)	integer

`https://www.kaggle.com/c/home-credit-default-risk` $70,000 prize

`https://www.kaggle.com/c/GiveMeSomeCredit` $ 5000 prize

Credit scoring as Supervised Learning

Credit scoring: Determining credit-worthiness based on:

Variable Name	Description	Type
SeriousDlqin2yrs	Person experienced 90 days past due delinquency or worse	Y/N
RevolvingUtilizationOfUnsecuredLines	Total balance on credit cards and personal lines of credit except real estate and no installment debt like car loans divided by the sum of credit limits	percentage
Age	Age of borrower in years	integer
NumberOfTime30-59DaysPastDueNotWorse	Number of times borrower has been 30-59 days past due but no worse in the last 2 years	integer
NumberOfTime60-89DaysPastDueNotWorse	Number of times borrower has been 60-89 days past due but no worse in the last 2 years	integer
NumberOfTime90DaysPastDueNotWorse	Number of times borrower has been 90 days or more past due	integer
DebtRatio	Monthly debt payments, alimony, living costs divided by monthly gross income	percentage
MonthlyIncome	Monthly income	real
NumberOfOpenCreditLinesAndLoans	Number of Open loans (installment like car loan or mortgage) and Lines of credit (e.g. credit cards)	integer
NumberRealEstateLoansOrLines	Number of mortgage and real estate loans including home equity lines of credit	integer
NumberOfDependents	Number of dependents in family excluding themselves (spouse, children etc.)	integer

Make a feature vector for each person

$$x = (\text{SeriousDlquin2yrs}, \text{Age}, \text{DebtRatio}, \text{MonthlyIncome} \ldots)$$

$$x_{sujay} = (0, 40, 3.6, 5000, \ldots)$$

Credit scoring as Supervised Learning

Regression

Task: given a feature vector, find the **credit score**
Credit score: a number between 1 and 800
Higher score = lower interest rate

Output: a real number

Classification

Task: given a feature vector, **accept or reject a loan application**

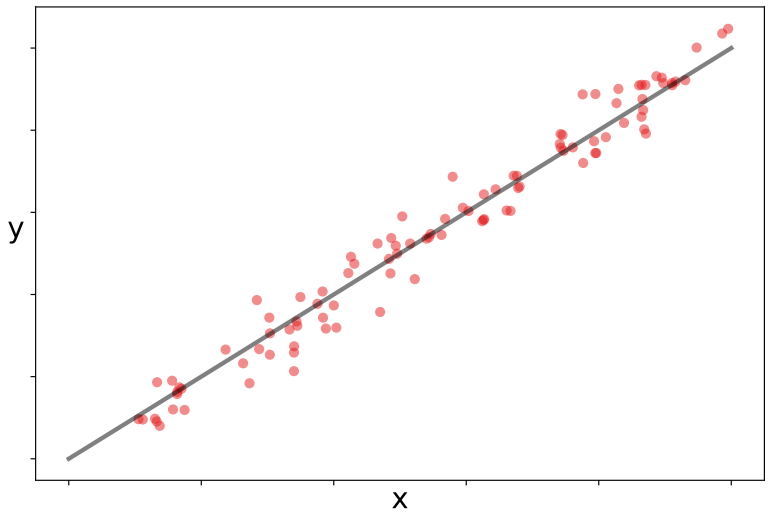
Output: a yes or no decision

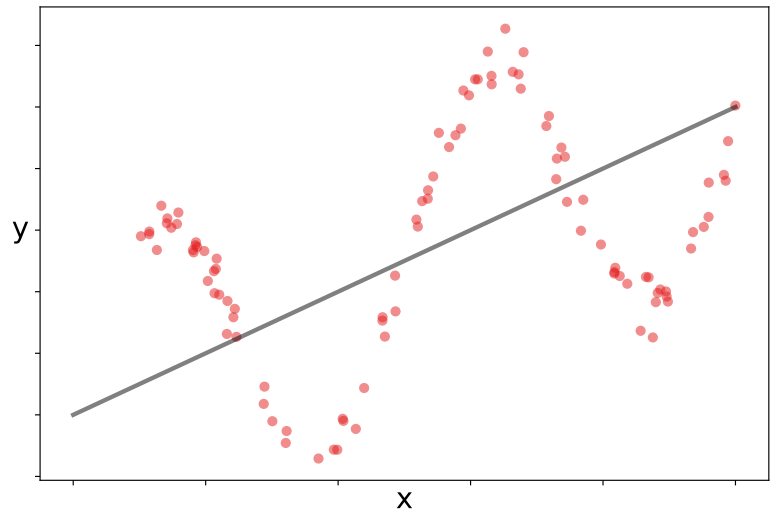
Supervised Learning

Make $f(x)$ a **linear function** of x

$$f(x) = \beta^\top x$$

Where β is a **parameter** of the model ...
.... that has to be found using the training data.





Linear Regression

$$f(x) \;=\; \beta_1 x_1 + \cdots + \beta_d x_d \;\triangleq\; \beta^\top x$$

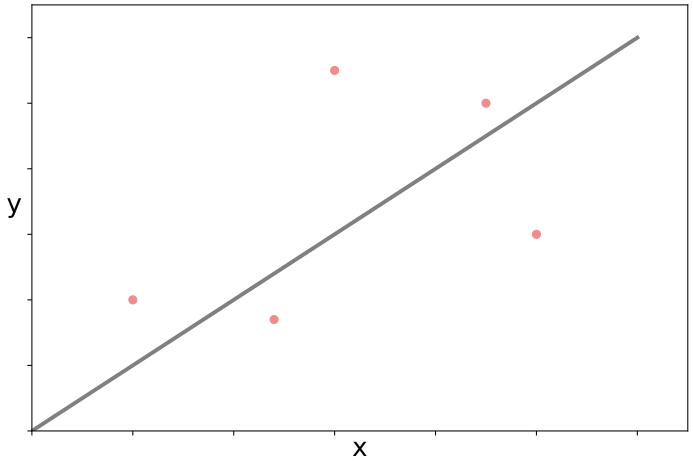
Step 1: Find β from samples $(x^{(i)}, y^{(i)})$ by **minimizing least square error**

$$\widehat{\beta} \;=\; \arg\min_{\beta} \sum_i \left(y^{(i)} - \beta^\top x^{(i)}\right)^2$$

Step 2: Validate & Evaluate

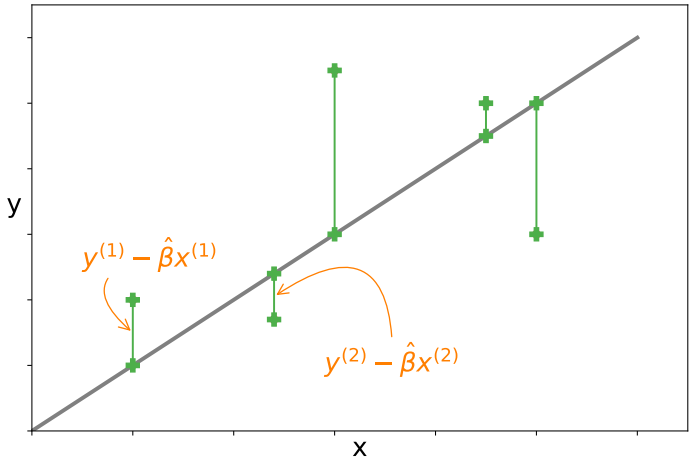
Is this $\widehat{\beta}$ "good" ? What is the error? etc.

Geometric Picture



$$\hat{\beta} = \arg\min_{\beta} \sum_{i} \left(y^{(i)} - \beta^{\top} x^{(i)} \right)^2$$

Geometric Picture



$$\hat{\beta} = \arg\min_{\beta} \sum_{i} \left(y^{(i)} - \beta^{\top} x^{(i)} \right)^2$$

Q: Why minimize sum of squared errors?

A1: Convenience: closed-form answer

A2: Corresponds to Gaussian noise ...

Probabilistic / Statistical View

Assume there is a "ground truth" β^*, such that

$$y = x^\top \beta^* + w$$

Probabilistic / Statistical View

Assume there is a "ground truth" β^*, such that

$$y = x^\top \beta^* + w$$

where noise w is Gaussian with mean 0 and variance σ^2, i.e. $w \sim \mathcal{N}(0, \sigma^2)$

$$\text{pdf of } w: \qquad p_W(w) = \frac{1}{\sqrt{2\pi\sigma^2}} exp\left(\frac{-w^2}{2\sigma^2}\right)$$

Recall: task: find $\widehat{\beta}$ given (x, y).

Likelihood:

$$\begin{aligned} L(\beta) &= p_W(y - \beta^\top x) \\ &= \frac{1}{\sqrt{2\pi\sigma^2}} exp\left(\frac{-(y - \beta^\top x)^2}{2\sigma^2}\right) \end{aligned}$$

Probabilistic / Statistical View

Maximum Likelihood: Find the β that makes the given x and y pair most likely

$$\widehat{\beta}_{ML} = \arg\max_{\beta} \frac{1}{\sqrt{2\pi\sigma^2}} exp\left(\frac{-(y-\beta^\top x)^2}{2\sigma^2}\right)$$

Probabilistic / Statistical View

Maximum Likelihood: Find the β that makes the given x and y pair most likely

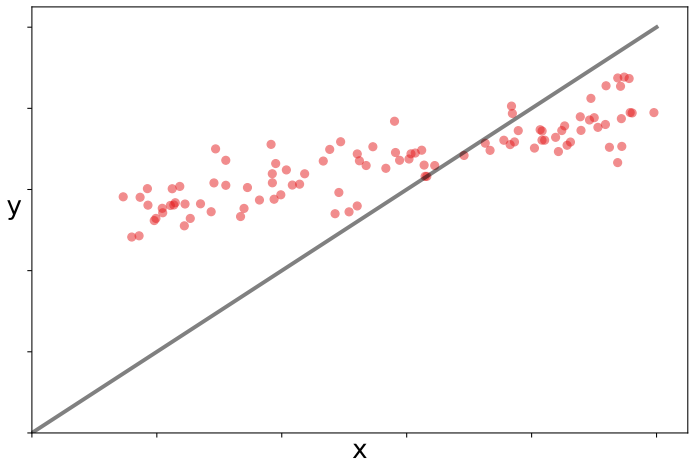
$$\begin{aligned}\widehat{\beta}_{ML} &= \arg\max_{\beta} \frac{1}{\sqrt{2\pi\sigma^2}} exp\left(\frac{-(y-\beta^\top x)^2}{2\sigma^2}\right) \\ &= \arg\min_{\beta} (y-\beta^\top x)^2\end{aligned}$$

i.e. the **least squares objective.**

This is the other "justification" for linear regression: it represents maximum likelihood (i.e. optimal estimator) when noise is Gaussian. Multiple samples:

$$\widehat{\beta}_{ML} = \arg\min_{\beta} \sum_i (y^{(i)}-\beta^\top x^{(i)})^2 \qquad \text{(Why?)}$$

(Removing) Intercept

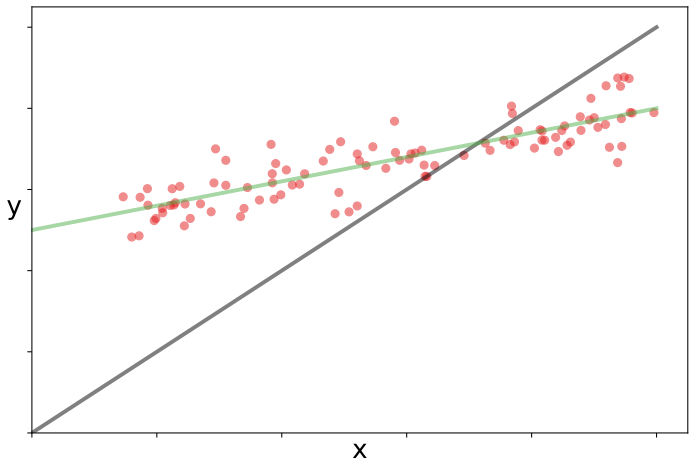


Best purely linear fit has to pass through $(0, 0)$

This is often a poor choice ...

... because y often has a *bias* term in practice ...

(Removing) Intercept



Fix: affine f

$$f(x) = \beta_0 + \beta_1 x_1 + \cdots + \beta_d x_d$$

$$\min_{\beta} \sum_i (y_i - \beta_0 - \beta_1 x_1 - \cdots - \beta_d x_d)^2$$

Common Trick:

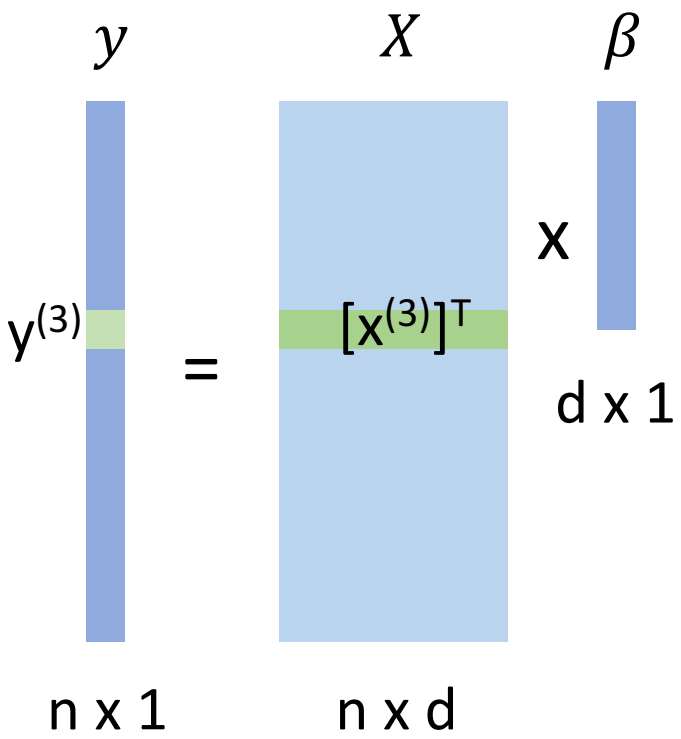
Pretend feature has extra "1" in 0-th coordinate

$$x = (1, x_1, x_2, \cdots, x_d)$$

$$\beta = (\beta_0, \beta_1, \ldots, \beta_d)$$

Solving Least Squares Problems

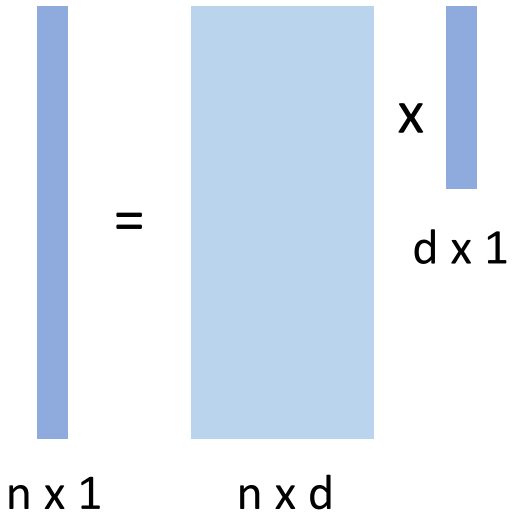
$$\min_{\beta} \sum_{i} \left(y^{(i)} - \beta^{\top} x^{(i)} \right)^2 \quad == \quad \min_{\beta} \|y - X\beta\|_2^2$$



i^{th} row of X is feature vector i^{th} sample, i.e. $[x^{(i)}]^{\top}$

Solving Least Squares Problems

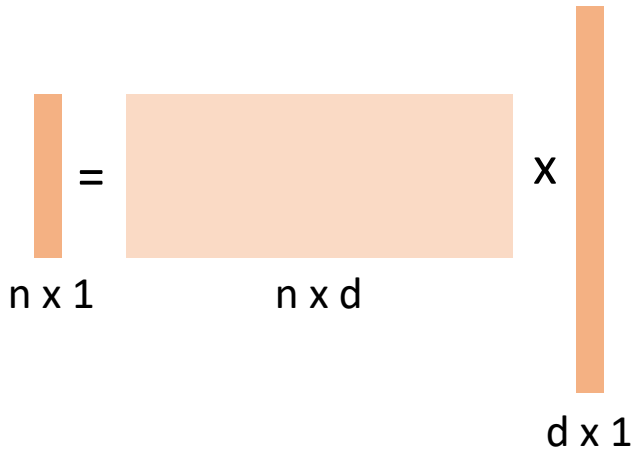
$n \geq d$



Over-determined

X is "tall / skinny" matrix

$n < d$

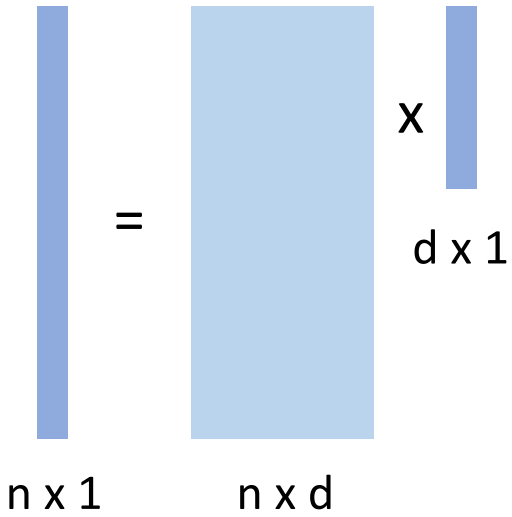


under-determined

X is "fat" matrix

Solving Least Squares Problems

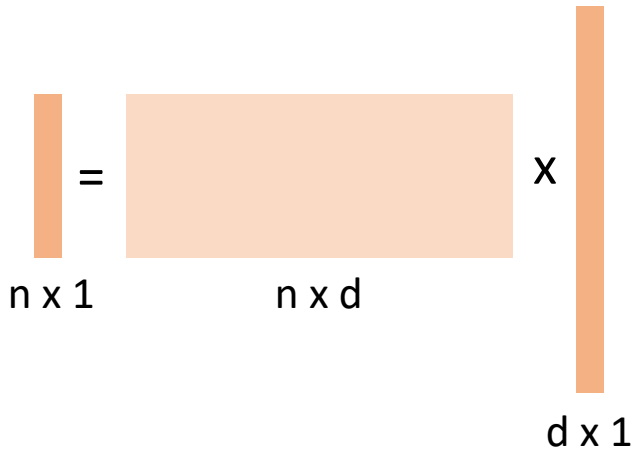
$n \geq d$



Unique best β:

$$\widehat{\beta} = (X^\top X)^{-1} X^\top y$$

$n < d$

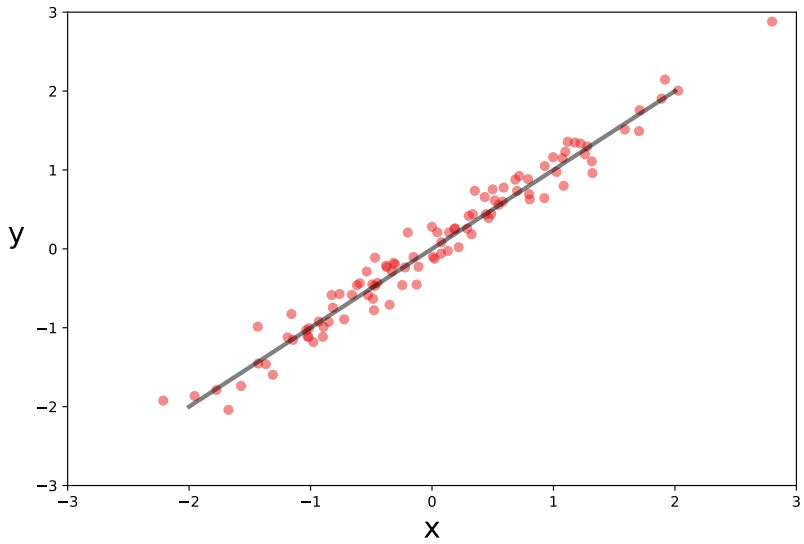


Many β can give 0 error. Common choice: the one with smallest $\|\beta\|$:

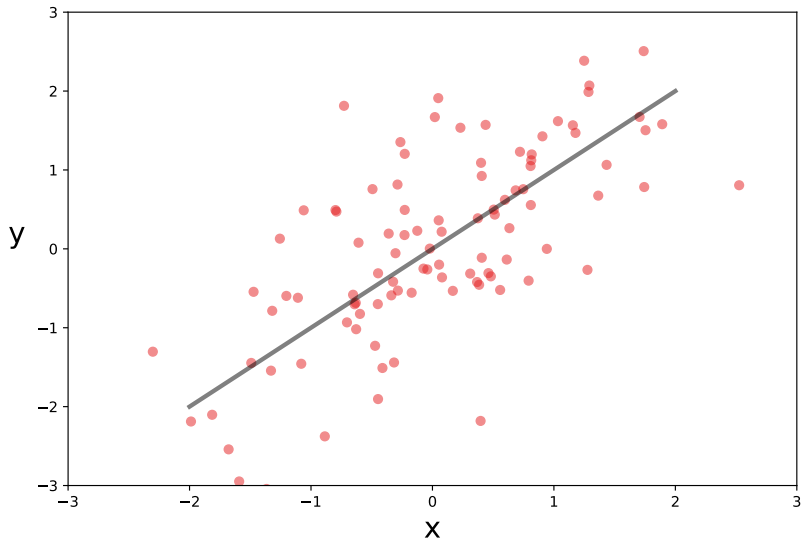
$$\widehat{\beta} = X^\top (X X^\top)^{-1} y$$

How Good is The Predictor ?

In both of the following: $x \sim \mathcal{N}(0, 1)$ and $y \sim \mathcal{N}(0, 1)$



Linear fit is "good"



Linear fit is "not so good"

Q: how do we evaluate quality of fit ?

Have we found a good predictor ?

Setting: we want to predict y from x

Idea: first look at how well we can "predict" y's if we **ignored** the x's

i.e. we want to find prediction $\widehat{y}$ given training samples $y^{(1)}, y^{(2)}, \ldots, y^{(n)}$

Minimizing mean squared error:

$$\begin{aligned} \widehat{y} &= \arg\min_{y} \sum_{i} \left(y^{(i)} - y \right)^2 \\ &= \frac{1}{n} \sum_{i} y^{(i)} \end{aligned}$$

Have we found a good predictor ?

Setting: we want to predict y from x

Idea: first look at how well we can "predict" y's if we **ignored** the x's

i.e. we want to find prediction $\widehat{y}$ given training samples $y^{(1)}, y^{(2)}, \ldots, y^{(n)}$

Minimizing mean squared error:

$$\begin{aligned} \widehat{y} &= \arg\min_{y} \sum_{i} \left(y^{(i)} - y \right)^2 \\ &= \frac{1}{n} \sum_{i} y^{(i)} \qquad \triangleq \bar{y} \quad \text{(the mean)} \end{aligned}$$

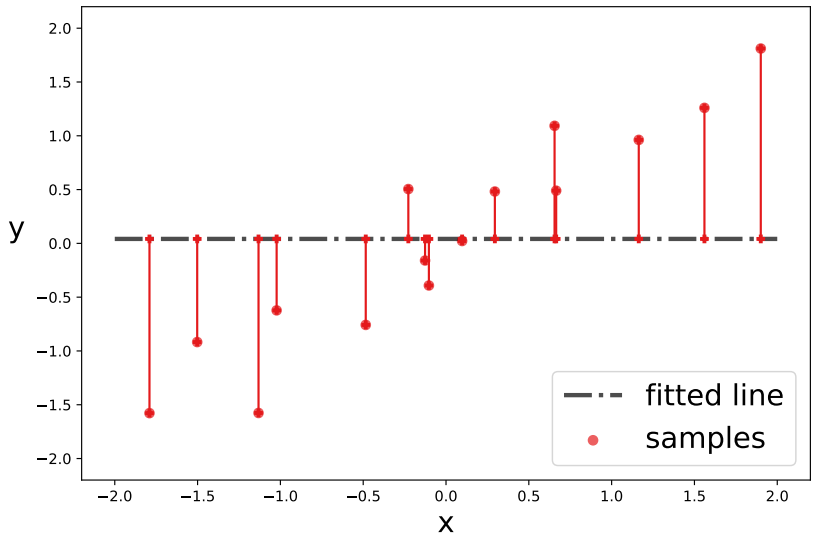
Note: same y value predicted for every sample

Have we found a good predictor ?

When features ignored, total error

$$\sum_i (y^{(i)} - \bar{y})^2$$

i.e. the variance of the y's



Have we found a good predictor ?

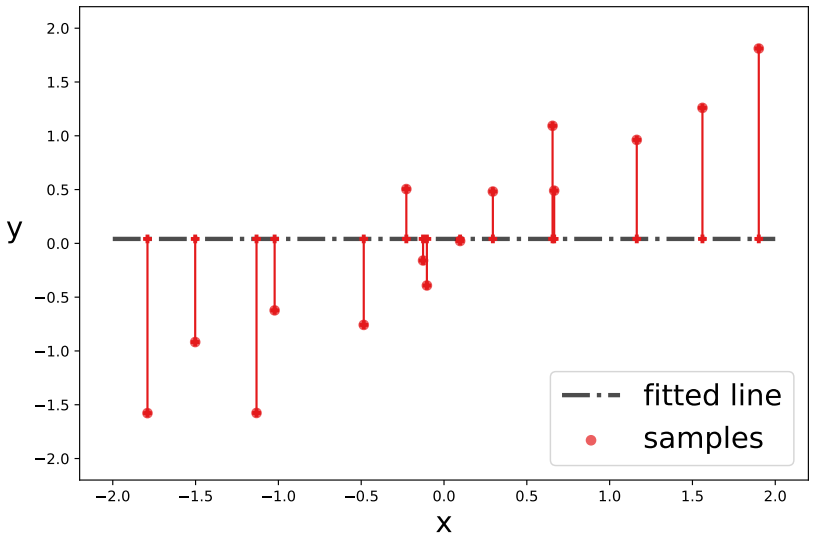
When features ignored, total error

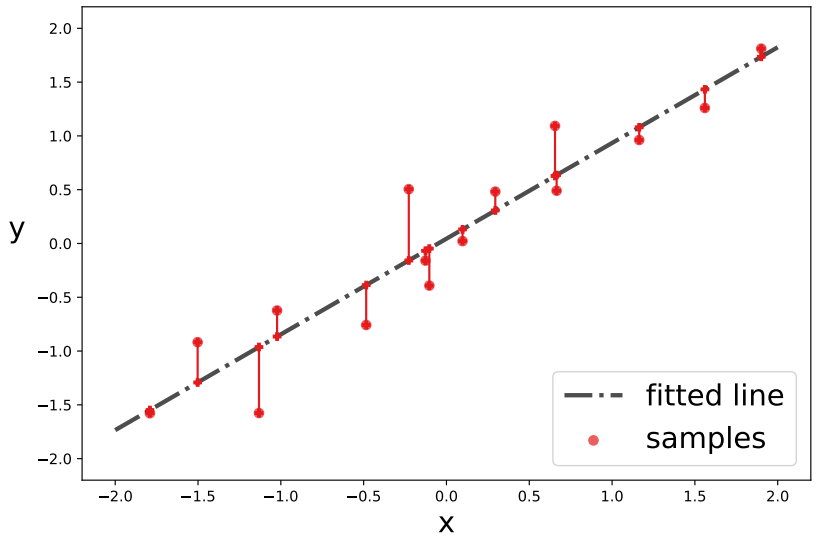
$$\sum_i (y^{(i)} - \bar{y})^2$$

i.e. the variance of the y's

Now suppose $\widehat{y}_i = \widehat{\beta} x^{(i)}$.
Total error is

$$\sum_i (y^{(i)} - \widehat{\beta}^\top x^{(i)})^2$$





Error of predictor is lower. (in this case)

R^2

The baseline in this case is $\widehat{y}_i = \bar{y}$, for which the mean square error is

$$\text{MSE(baseline)} \triangleq \sum_i (y^{(i)} - \bar{y})^2$$

The model in this case is $\widehat{y}_i = \widehat{\beta}^\top x^{(i)}$, for which mean squared error is

$$\text{MSE(model)} \triangleq \sum_i (y^{(i)} - \widehat{\beta}^\top x^{(i)})^2$$

R^2

The baseline in this case is $\widehat{y}_i = \bar{y}$, for which the mean square error is

$$\text{MSE(baseline)} \triangleq \sum_i (y^{(i)} - \bar{y})^2$$

The model in this case is $\widehat{y}_i = \widehat{\beta}^\top x^{(i)}$, for which mean squared error is

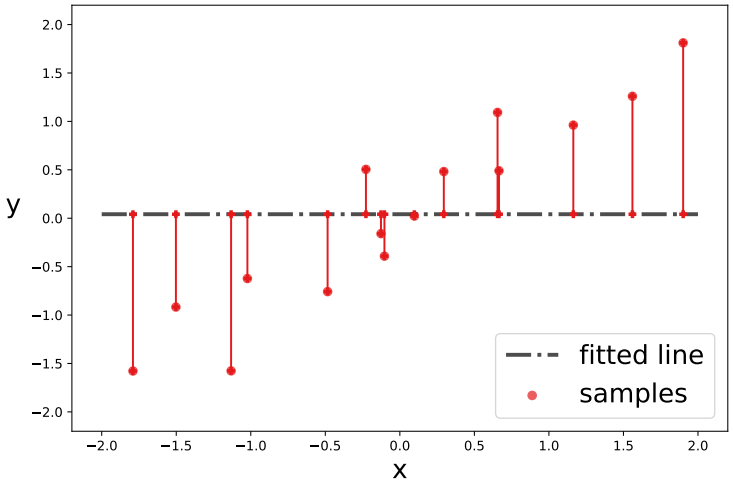
$$\text{MSE(model)} \triangleq \sum_i (y^{(i)} - \widehat{\beta}^\top x^{(i)})^2$$

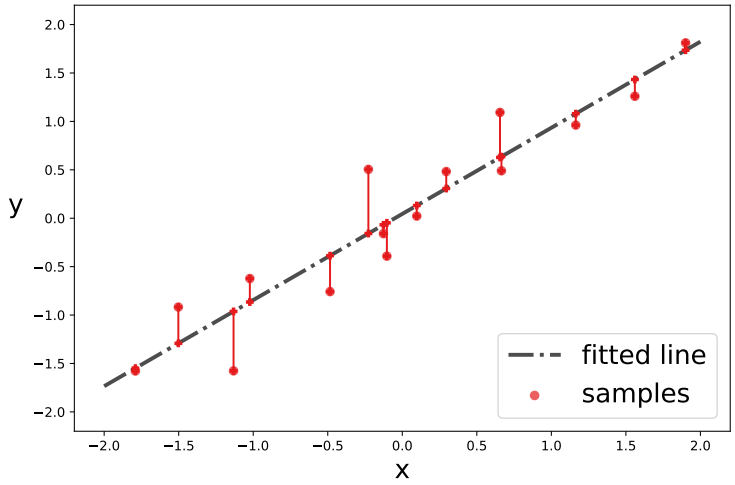
The $\mathbf{R^2}$ value, or **coefficient of determination**, or **explained variance**, is

$$R^2 \triangleq 1 - \frac{\text{MSE(model)}}{\text{MSE(baseline)}} = 1 - \frac{\sum_i (y^{(i)} - \widehat{\beta}^\top x^{(i)})^2}{\sum_i (y^{(i)} - \bar{y})^2}$$

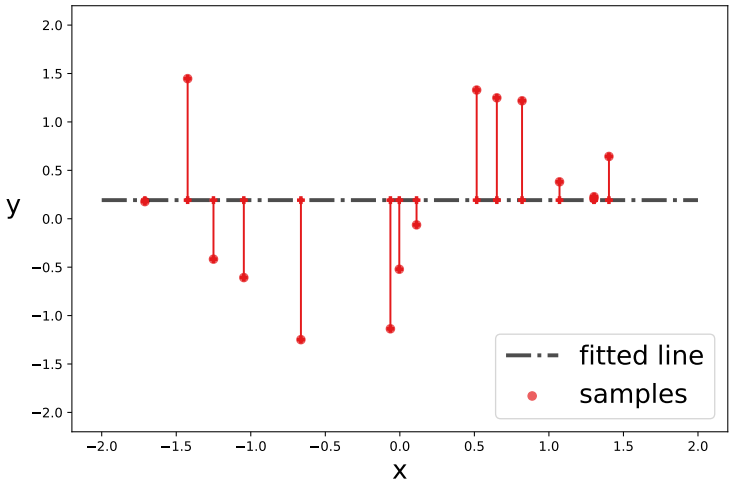
Higher $R^2 \Leftrightarrow$ better predictor.

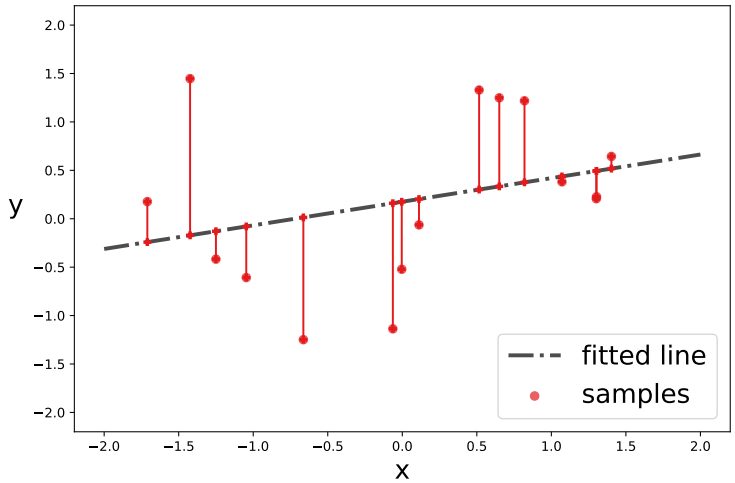
$R^2 = 0.8$





$R^2 = 0.2$





R^2 = gain in accuracy if x used to predict y (as compared to not using it)

R^2

$$R^2 = 1 - \frac{\text{MSE(model)}}{\text{MSE(baseline)}}$$

$R^2 = 0$ means model is no better than baseline.

Maximum value of R^2 is 1. What is an example of this ?

What is the minimum value ? What is an example of this ?

Adjusted R^2

What happens if we add an extra useless feature ?

Before: each sample is $x = (x_1, \ldots, x_d)$ and y
Now: each sample is $x = (x_1, \ldots, x_d, x_{d+1})$ and y

And suppose that the new $(d+1)^{th}$ feature has no explanatory power - say it is just noise.

We can go through the whole exercise again (i.e. do regression, calculate the R^2 etc.)

$$R^2_{before} = R^2_{after}$$

Adjusted R^2

Problem: R^2 does not penalize extraneous features.

This can lead to in-advertant over-fitting

Adjusted R^2 aims to correct for this

$$\bar{R}^2 \triangleq 1 - \left(1 - R^2\right) \frac{n-1}{n-d-1}$$

where n is the number of samples, and d is the number of features.

As the number of features d becomes larger, the model needs to perform better to get a better $\bar{R}^2$

Thought experiment: add features one at a time, starting with the most informative. What happens to R^2 ? And what happens to $\bar{R}^2$?