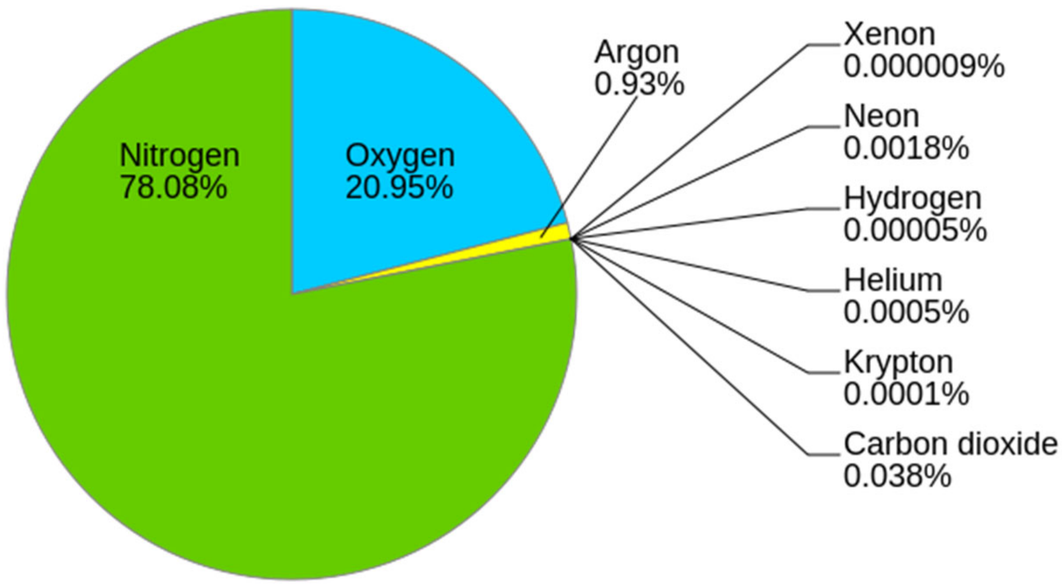
Reminder

Two variable samples



- Oxygen can be distilled from the air
- Hydrocarbons need to be filtered out or the whole thing would go kaboom!!!
- When more hydrocarbons were removed, the remaining oxygen stays cleaner
- Except we don't know how dirty was the air to begin with

Table 11-1 Oxygen and Hydrocarbon Levels

Observation Number	Hydrocarbon Level x (%)	Purity y (%)
1	0.99	90.01
2	1.02	89.05
3	1.15	91.43
4	1.29	93.74
5	1.46	96.73
6	1.36	94.45
7	0.87	87.59
8	1.23	91.77
9	1.55	99.42
10	1.40	93.65
11	1.19	93.54
12	1.15	92.52
13	0.98	90.56
14	1.01	89.54
15	1.11	89.85
16	1.20	90.39
17	1.26	93.25
18	1.32	93.41
19	1.43	94.98
20	0.95	87.33

Linear regression

The **simple linear regression model** is given by

$Y = \beta_0 + \beta_1 X + \varepsilon = \hat{Y} + \varepsilon$

ε is the **random error term**

slope β_1 and intercept β_0 of the line are called **regression coefficients**

Note: Y , $\hat{Y}$, X and ε are random variables
The minimal assumption: E(ε | x)=0 →
E(Y| x) =$\beta_0 + \beta_1 x$ + E(ε | x) =$\beta_0 + \beta_1 x$

$$Y = \beta_0 + \beta_1 X + \epsilon$$

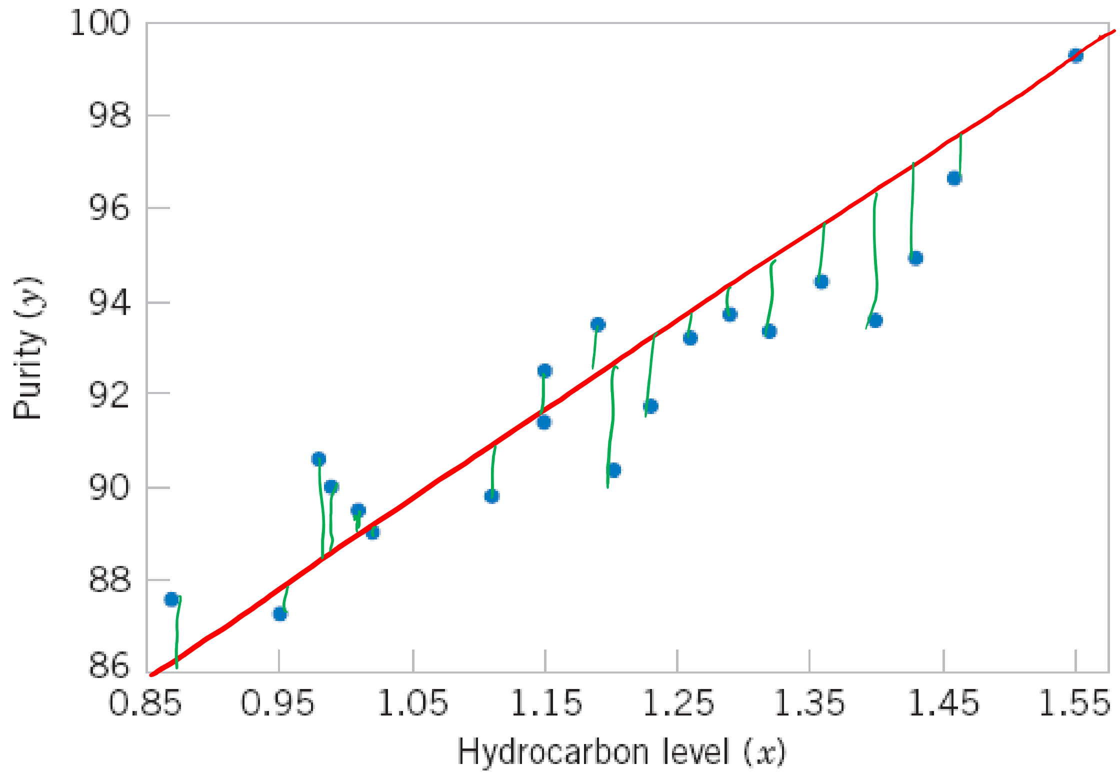


Figure 11-1 Scatter diagram of oxygen purity versus hydrocarbon level from Table 11-1.

$$Y = 75 + 15 \cdot X + \epsilon$$

$$Y = \beta_0 + \beta_1 X + \epsilon \; ; \quad E(\epsilon \mid x) = 0 \quad \forall x$$

How does one find β_0 & β_1?

$$Cov(Y, X) = Cov((\beta_0 + \beta_1 X + \epsilon), X) =$$

$$= Cov(\beta_0, X) + \beta_1 Cov(X, X) + Cov(\epsilon, X)$$

$Cov(\beta_0, X) = 0$ since β_0 is constant

$$Cov(X, X) = E(X^2) - E(X)^2 = Var(X)$$

$$Cov(\epsilon, X) = E(\epsilon \cdot X) - E(\epsilon) \cdot E(X) =$$

$$= E(\epsilon \cdot X) = \sum_{\text{all } x} x \cdot E(\epsilon \mid x) = 0$$

Thus

$$\boxed{\beta_1 = \frac{Cov(X, Y)}{Var(X)}} ; \quad \boxed{\beta_0 = E(Y) - \beta_1 E(X)}$$

Method of least squares

- The **method of least squares** is used to estimate the parameters, β_0 and β_1 by minimizing the sum of the squares of the vertical deviations in Figure 11-3.

Figure 11-3 Deviations of the data from the estimated regression model.

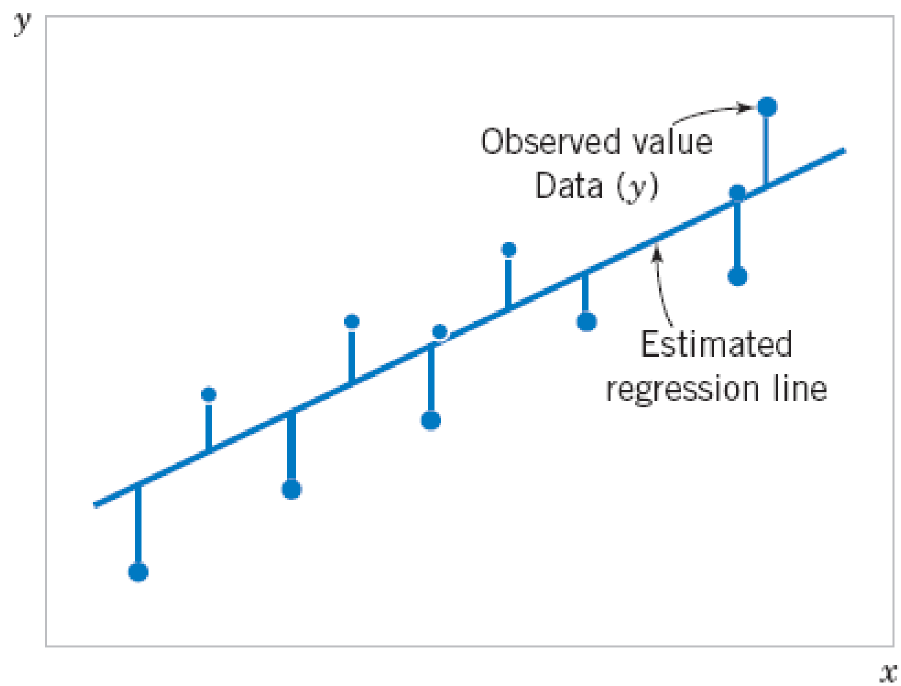


Figure 11-3 Deviations of the data from the estimated regression model.

Traditional notation

Definition

The **least squares estimates** of the intercept and slope in the simple linear regression model are

$$\hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x} \qquad (11\text{-}7)$$

$$\hat{\beta}_1 = \frac{\sum_{i=1}^{n} y_i x_i - \frac{\left(\sum_{i=1}^{n} y_i\right)\left(\sum_{i=1}^{n} x_i\right)}{n}}{\sum_{i=1}^{n} x_i^2 - \frac{\left(\sum_{i=1}^{n} x_i\right)^2}{n}} = \frac{S_{xy}}{S_{xx}} \qquad (11\text{-}8)$$

where $\bar{y} = (1/n)\sum_{i=1}^{n} y_i$ and $\bar{x} = (1/n)\sum_{i=1}^{n} x_i$.

Connection to Cov(X,Y)/Var(X) result

Definition

The **least squares estimates** of the intercept and slope in the simple linear regression model are

$$\hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x} \qquad (11\text{-}7)$$

$$\hat{\beta}_1 = \frac{\dfrac{\sum_{i=1}^{n} y_i x_i}{n} - \dfrac{\left(\sum_{i=1}^{n} y_i\right)\left(\sum_{i=1}^{n} x_i\right)}{n^2}}{\dfrac{\sum_{i=1}^{n} x_i^2}{n} - \dfrac{\left(\sum_{i=1}^{n} x_i\right)^2}{n^2}} = \frac{Cov(X,Y)}{Var(X)} \qquad (11\text{-}8)$$

where $\bar{y} = (1/n)\sum_{i=1}^{n} y_i$ and $\bar{x} = (1/n)\sum_{i=1}^{n} x_i$.

Different types of y

The **least squares estimates** of the intercept and slope in the simple linear regression model are

$$\hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x} \qquad (11\text{-}7)$$

$$\hat{\beta}_1 = \frac{\sum_{i=1}^{n} y_i x_i - \frac{\left(\sum_{i=1}^{n} y_i\right)\left(\sum_{i=1}^{n} x_i\right)}{n}}{\sum_{i=1}^{n} x_i^2 - \frac{\left(\sum_{i=1}^{n} x_i\right)^2}{n}} \qquad (11\text{-}8)$$

where $\bar{y} = (1/n)\sum_{i=1}^{n} y_i$ and $\bar{x} = (1/n)\sum_{i=1}^{n} x_i$.

$$= \frac{Cov(X, Y)}{Var(X)}$$

$$\bar{y} = \sum y_i / n$$

$$\hat{y}_i = \hat{\beta}_1 x_i + \hat{\beta}_0$$

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

The analysis of variance identity is

$$\sum_{i=1}^{n}(y_i - \bar{y})^2 = \sum_{i=1}^{n}(\hat{y}_i - \bar{y})^2 + \sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \qquad (11\text{-}24)$$

Symbolically,

$$SS_T = SS_R + SS_E \qquad (11\text{-}25)$$

11-7: Adequacy of the Regression Model

11-7.2 Coefficient of Determination (R^2)

VERY COMMONLY USED

- The quantity

$$R^2 = \frac{SS_R}{SS_T} = 1 - \frac{SS_E}{SS_T}$$

is called the **coefficient of determination** and is often used to judge the adequacy of a regression model.

- $0 \le R^2 \le 1$;
- We often refer (loosely) to R^2 as the amount of variability in the data explained or accounted for by the regression model.

11-2: Simple Linear Regression

Estimating σ_ε^2

An **unbiased estimator** of σ_ε^2 is

$$\hat{\sigma}_\varepsilon^2 = \frac{SS_E}{n-2} \qquad (11\text{-}13)$$

where SS_E can be easily computed using

$$SS_E = SS_T - \hat{\beta}_1 S_{xy} \qquad (11\text{-}14)$$

Multiple Linear Regression

(Chapters 12-13 in
Montgomery, Runger)

12-1: Multiple Linear Regression Model

12-1.1 Introduction

- Many applications of regression analysis involve situations in which there are more than one regressor variable X_k used to predict Y.

- A regression model then is called a **multiple regression model**.

Multiple Linear Regression Model

$$Y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \ldots \beta_k x_k + \varepsilon$$

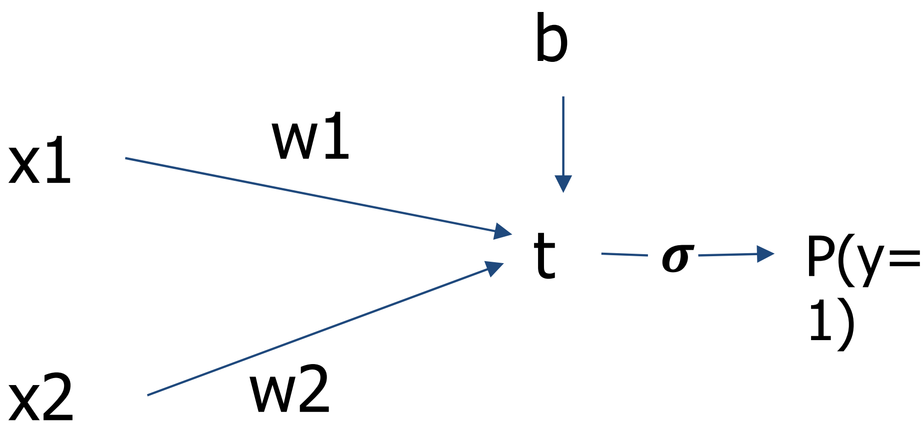
One can also use powers and products of other variables or even non-linear functions like $\exp(x_i)$ or $\log(x_i)$ instead of $x_3, \ldots x_k$.

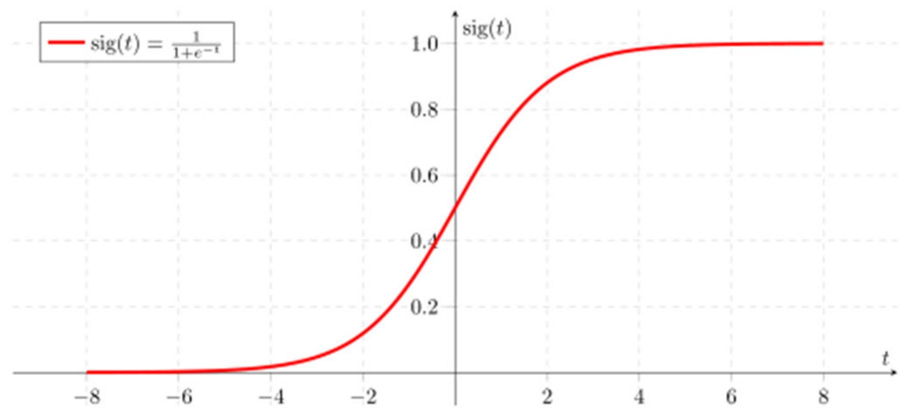
Example: the general two-variable quadratic regression has 6 constants:

$$Y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 (x_1)^2 + \beta_4 (x_2)^2 + \beta_5 (x_1 x_2) + \varepsilon$$

Logistic Regression

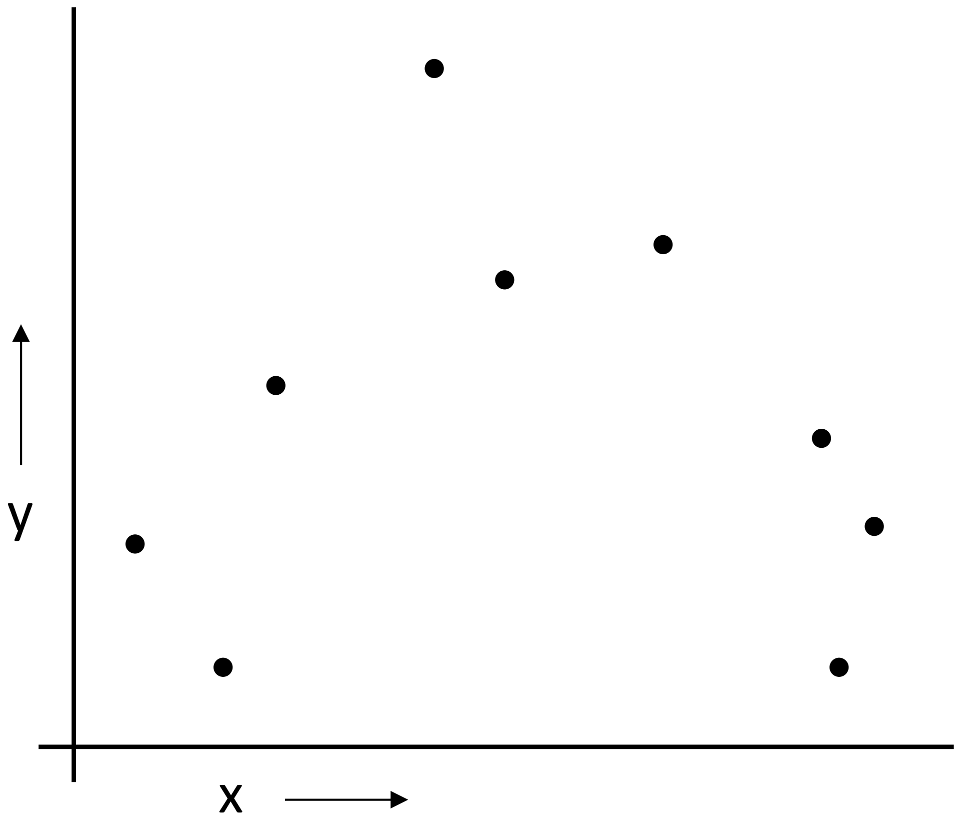
P(y=1) = $\boldsymbol{\sigma}$(x1*w1 + x2*w2 + b)





How to know where to stop adding new variables or powers of old variables?

A Regression Problem

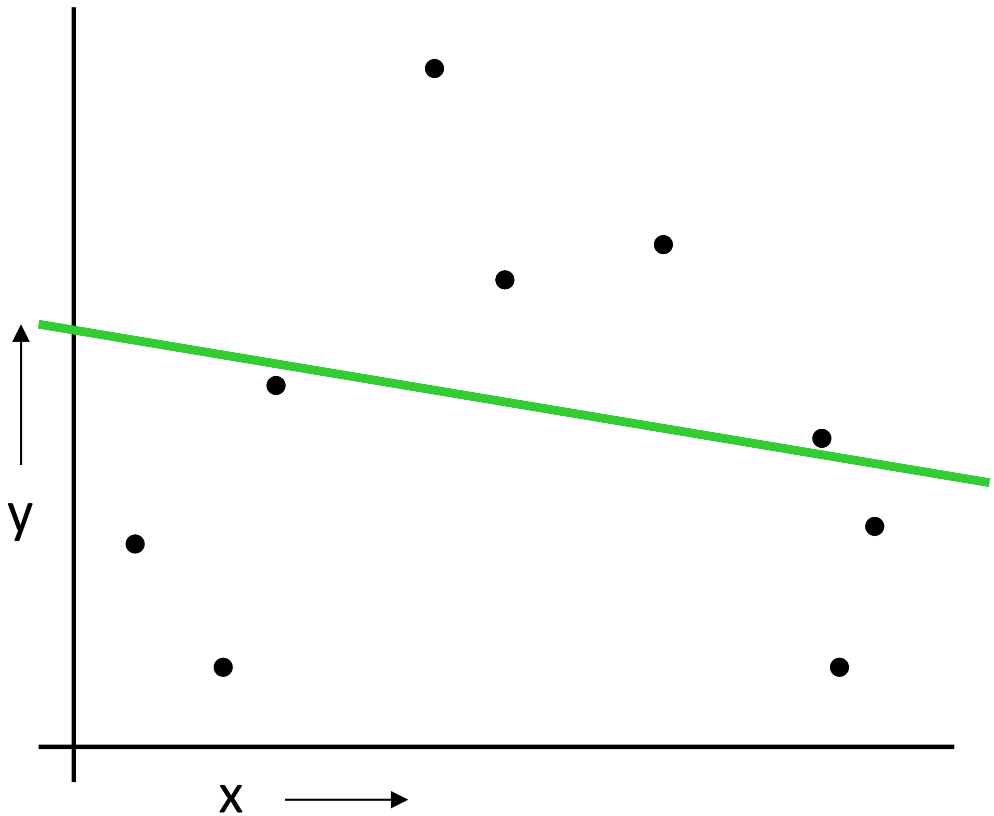


y = f(x) + noise

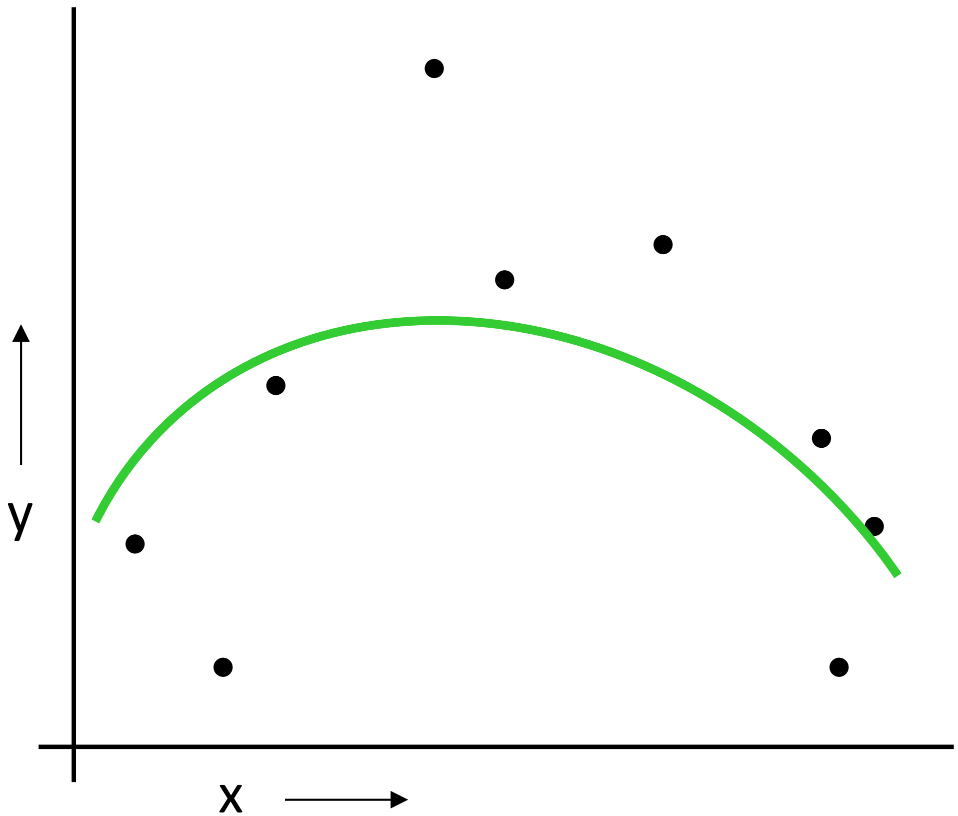
Can we learn f from this data?

Let’s consider three methods…

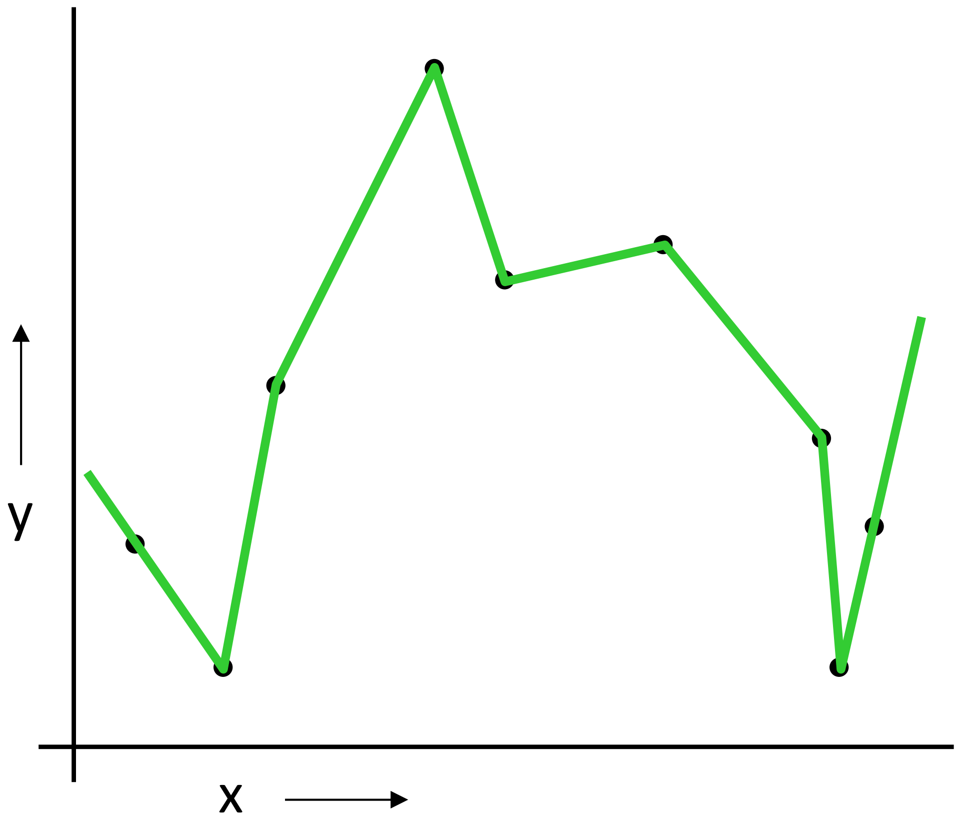
Linear Regression



Quadratic Regression

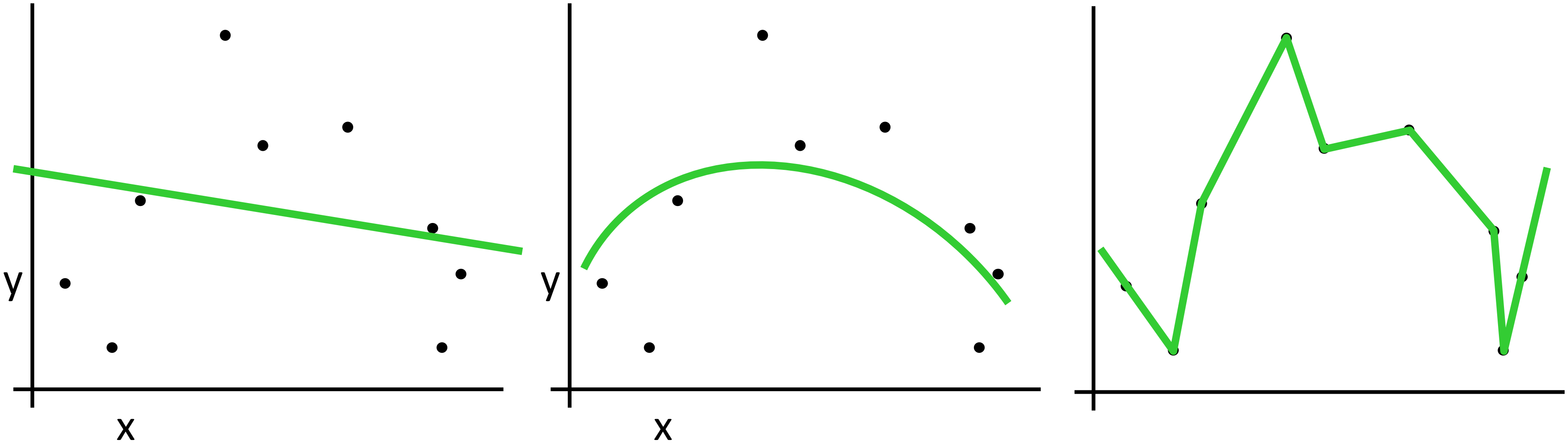


Join-the-dots



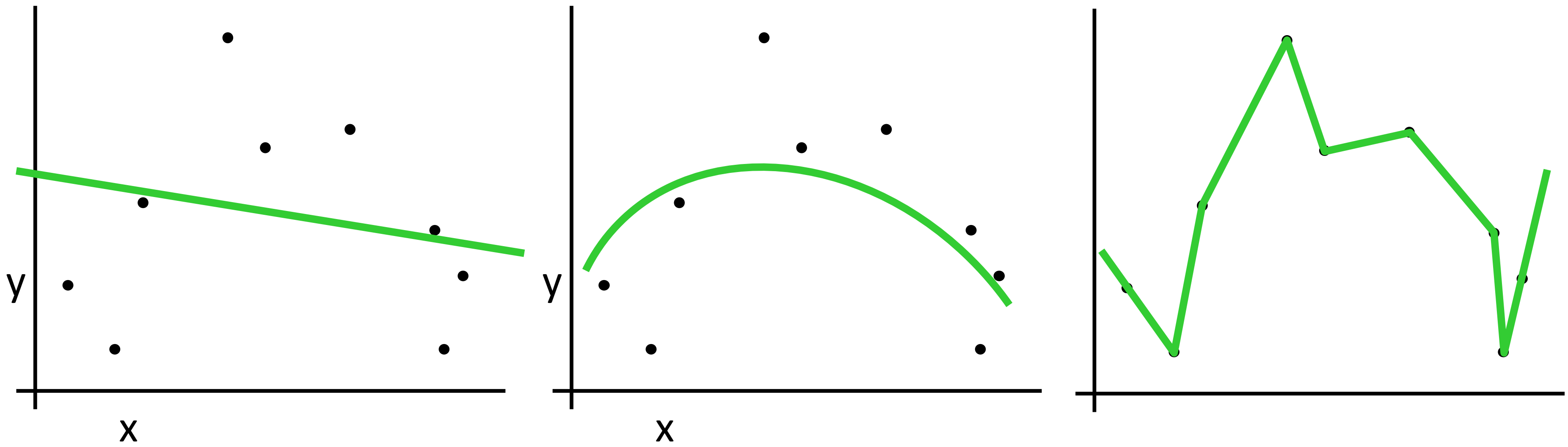
Also known as piecewise linear nonparametric regression if that makes you feel better

Which is best?



Why not choose the method with the best fit to the data?

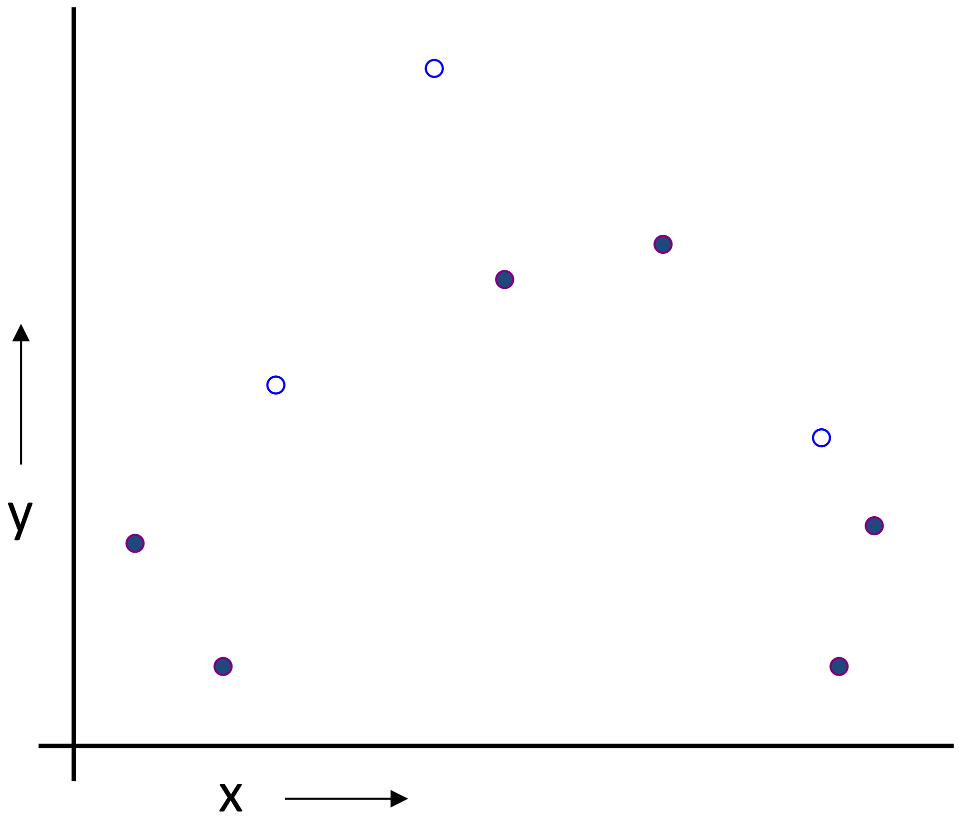
What do we really want?



Why not choose the method with the best fit to the data?

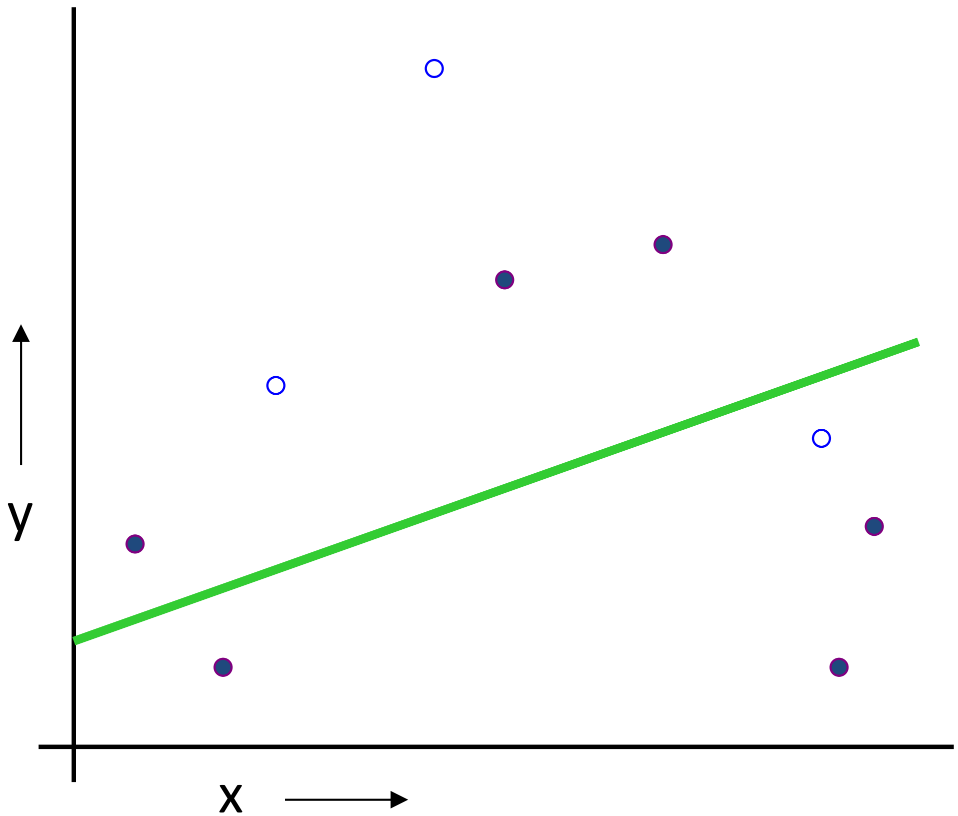
"How well are you going to predict future data drawn from the same distribution?"

The test set method



1. Randomly choose 30% of the data to be in a test set
2. The remainder is a training set

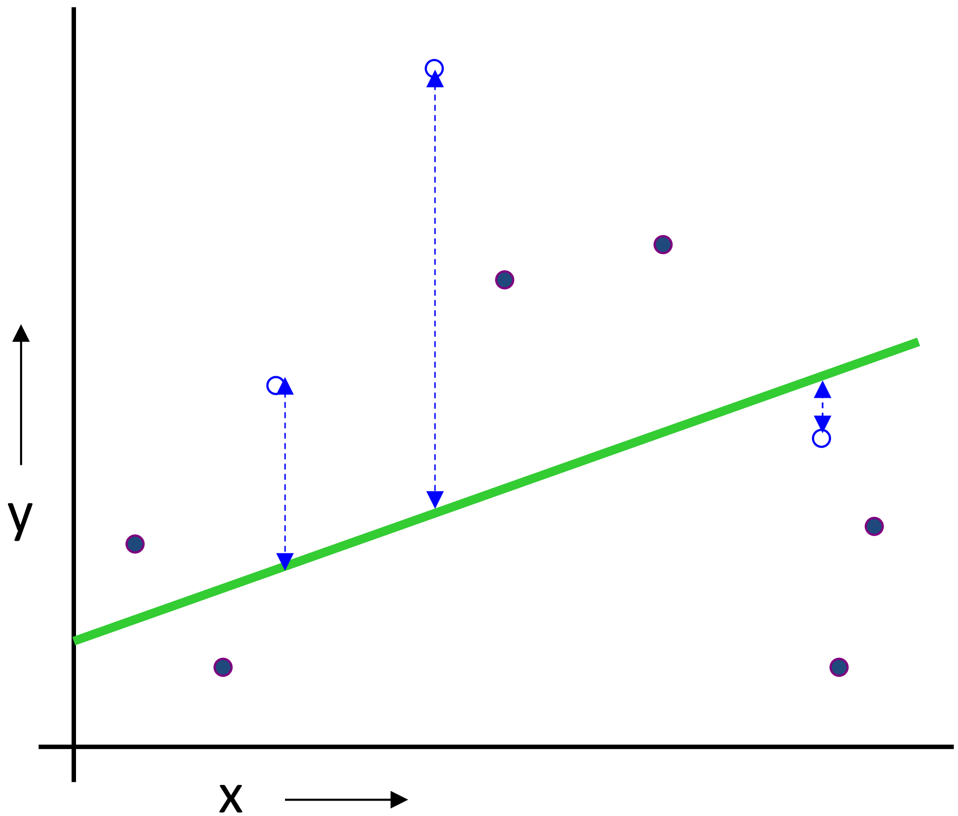
The test set method



(Linear regression example)

1. Randomly choose 30% of the data to be in a test set
2. The remainder is a training set
3. Perform your regression on the training set

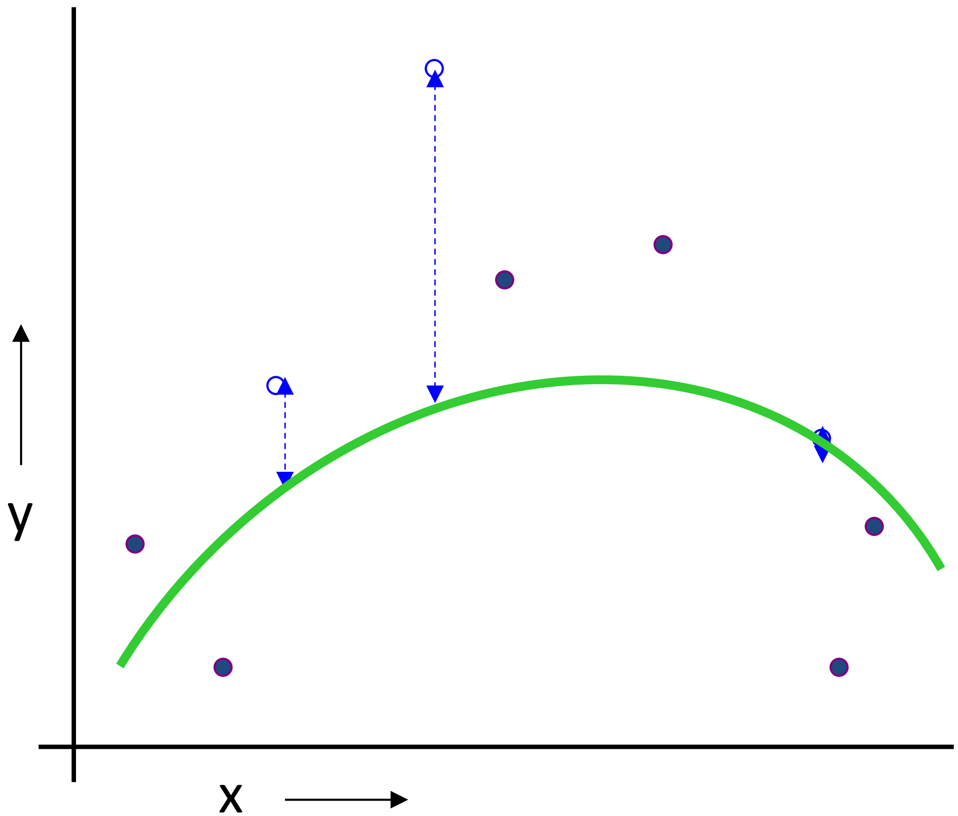
The test set method



(Linear regression example)
Mean Squared Error = 2.4

1. Randomly choose 30% of the data to be in a test set
2. The remainder is a training set
3. Perform your regression on the training set
4. Estimate your future performance with the test set

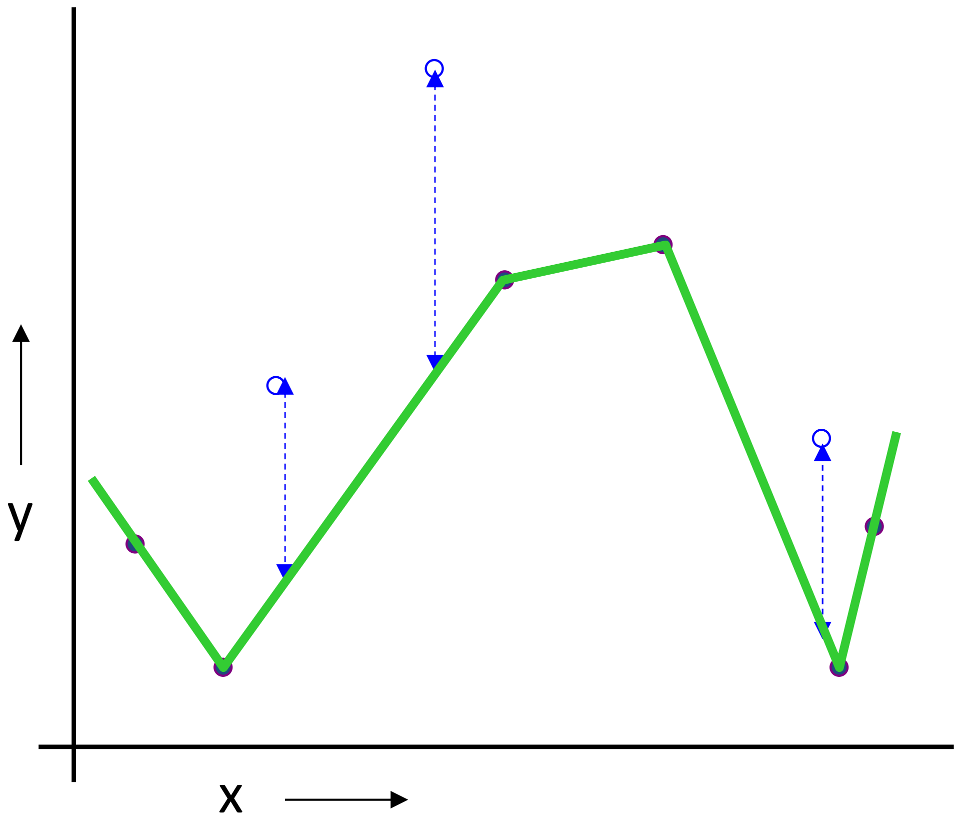
The test set method



(Quadratic regression example)
Mean Squared Error = 0.9

1. Randomly choose 30% of the data to be in a test set
2. The remainder is a training set
3. Perform your regression on the training set
4. Estimate your future performance with the test set

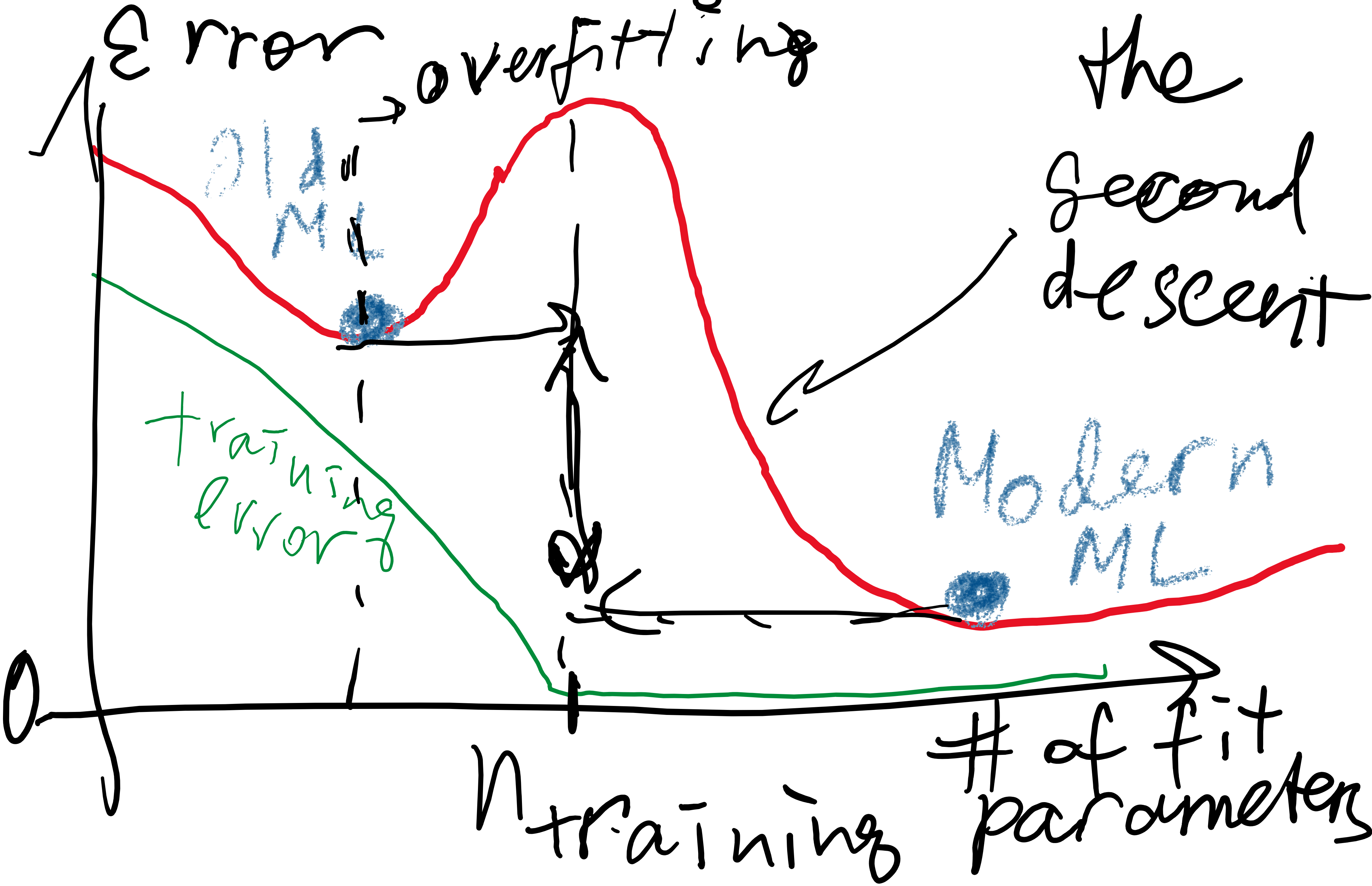
The test set method



(Join the dots example)
Mean Squared Error = 2.2

1. Randomly choose 30% of the data to be in a test set
2. The remainder is a training set
3. Perform your regression on the training set
4. Estimate your future performance with the test set

Double descend- the main reason modern Machine Learning works so well



12-1: Multiple Linear Regression Model

12-1.3 Matrix Approach to Multiple Linear Regression

Suppose the model relating the regressors to the response is

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_k x_{ik} + \varepsilon_i \qquad i = 1, 2, \ldots, n$$

In matrix notation this model can be written as

$$\mathbf{y} = \mathbf{X}\beta + \varepsilon \qquad (12\text{-}6)$$

12-1: Multiple Linear Regression Model

12-1.3 Matrix Approach to Multiple Linear Regression

where

$$\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} \quad \mathbf{X} = \begin{bmatrix} 1 & x_{11} & x_{12} & \cdots & x_{1k} \\ 1 & x_{21} & x_{22} & \cdots & x_{2k} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & x_{n1} & x_{n2} & \cdots & x_{nk} \end{bmatrix} \quad \boldsymbol{\beta} = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_k \end{bmatrix} \text{ and } \boldsymbol{\varepsilon} = \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_n \end{bmatrix}$$

12-1.3 Matrix Approach to Multiple Linear Regression

We wish to find the vector $\hat{\beta}$ that minimizes the sum of squares of error terms:

$$L = \sum_{i=1}^{n} \varepsilon_i^2 = \varepsilon'\varepsilon = (y - \mathbf{X}\beta)'(\mathbf{y} - \mathbf{X}\beta)$$

$$0 = \frac{\partial L}{2\partial \beta} = -\boldsymbol{X}'(\mathbf{y} - \mathbf{X}\beta) = -\boldsymbol{X}'\boldsymbol{y} + (\boldsymbol{X}'\boldsymbol{X})\boldsymbol{\beta}$$

The resulting least squares estimate is

$$\hat{\beta} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y} \qquad (12\text{-}7)$$

Analog of $\frac{1}{Var(X)}$

Analog of $Cov(X,Y)$

Multiple Linear Regression Model

$$\hat{\beta} = (\mathbf{X}'\mathbf{X})^{-1}\,\mathbf{X}'\mathbf{y}$$

H is an idempotent matrix

$$\hat{y} = X\hat{\beta} = X(X'X)^{-1}X'y,$$

$$\hat{y} = Hy, \quad \text{and} \quad e = (I - H)y.$$

$H = H'$; $H^2 = X(X'X)^{-1}X'X(X'X)^{-1}X' = X(X'X)^{-1}X' = H$

($X'X(X'X)^{-1}$ = I)

Vectors $\hat{y}$ & e are orthogonal since

$\hat{y}' \cdot e = y'H(I-H)y = 0$ since

$$H(I - H) = H - H^2 = H - H = 0.$$

12-1: Multiple Linear Regression Models

12-1.4 Properties of the Least Squares Estimators

Unbiased estimators:

$$\begin{aligned} E(\hat{\boldsymbol{\beta}}) &= E[(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{Y}] \\ &= E[(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'(\mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon})] \\ &= E[(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{X}\boldsymbol{\beta} + (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\boldsymbol{\epsilon}] \\ &= \boldsymbol{\beta} \end{aligned}$$

Covariance Matrix of Estimators:

$$\mathbf{C} = (\mathbf{X}'\mathbf{X})^{-1} = \begin{bmatrix} C_{00} & C_{01} & C_{02} \\ C_{10} & C_{11} & C_{12} \\ C_{20} & C_{21} & C_{22} \end{bmatrix}$$

12-1: Multiple Linear Regression Models

12-1.4 Properties of the Least Squares Estimators

Individual variances and covariances:

$$V(\hat{\beta}_j) = \sigma^2 C_{jj}, \qquad j = 0, 1, 2$$

$$\text{cov}(\hat{\beta}_i, \hat{\beta}_j) = \sigma^2 C_{ij}, \qquad i \neq j$$

In general,

$$\text{cov}(\hat{\boldsymbol{\beta}}) = \sigma^2(\mathbf{X}'\mathbf{X})^{-1} = \sigma^2\mathbf{C}$$

12-1: Multiple Linear Regression Models

Estimating error variance σ_ε^2

An unbiased estimator of error variance σ_ε^2 is

$$\hat{\sigma}_\varepsilon^2 = \frac{\sum_{i=1}^{n} e_i^2}{n-p} = \frac{SS_E}{n-p} \qquad (12\text{-}16)$$

Here p=k+1 for k-variable multiple linear regression

R^2 and Adjusted R^2

The **coefficient of multiple determination R^2**

$$R^2 = \frac{SS_R}{SS_T} = 1 - \frac{SS_E}{SS_T}$$

The **adjusted R^2** is

$$R^2_{\text{adj}} = 1 - \frac{SS_E/(n-p)}{SS_T/(n-1)} \qquad (12\text{-}23)$$

- The adjusted R^2 statistic penalizes adding terms to the MLR model.
- It can help guard against overfitting (including regressors that are not really useful)

How to know where to stop adding variables?

- Adding new variables x_i to MLR
  watch the adjusted R^2

- Once the adjusted R^2
  no longer increases = stop.
  Now you did the best you can.

Matlab exercise

- Every group works with
  g0=2907; g1=1527; g2=2629; g3=2881;
  g4=1144; g5=1066;
- Compute Multiple Linear Regression (MLR):
  where
  y=exp_t (g0); x1= exp_t (g1); x2= exp_t (g2);
- How much better the MLR did compared to the Single Linear Regression (SLR)?
- Continue increasing the number of genes in x until R_adj starts to decrease

How I did it

- `g0=2907; g1=1527; g2=2629; g3=2881;g4=1144; g5=1066;`
- `y=exp_t(g0,:)';`
- `%% first use one x to predict y`
- `x=exp_t(g1,:)';`
- `figure; plot(x,y,'ko')`
- `lm=fitlm(x,y)`
- `y_fit=lm.Fitted;`
- `hold on;`
- `plot(x,lm.Fitted,'r-');`
- `%% now use 2 x's to predict y`
- `x=[exp_t(g1,:)', exp_t(g2,:)'];`
- `lm2=fitlm(x,y)`
- `y_fit=lm2.Fitted;`
- `hold on; plot(x(:,1),y_fit,'gd');`
- `%% now use m x's to predict y`
- `corr_matrix=corr(exp_t');`
- `g0=2907;`
- `[u v]=sort(corr_matrix(g0,:),'descend');`
- `x=[exp_t(v(2:m+1),:)'];`
- `lm3=fitlm(x,y)`
- `y_fit=lm3.Fitted;`
- `plot(x(:,1),y_fit,'s');`

Credit: XKCD comics