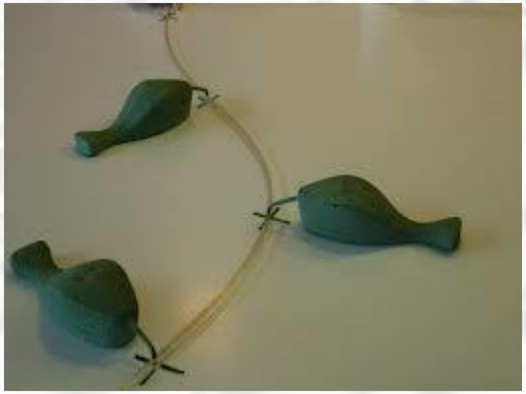
Linear Regression

When outcome is not binary

- Outcome can model trend, price, etc.
- Ability to both interpolate (make inference) and extrapolate (make prediction)
- A rich math area studied in many disciplines (e.g., spline theory)



igure 11c. The forward part of the bump is refined and drawn orward to create a snout.

Figure 11d. The tip of the snout is refined and pulled down into a beak.

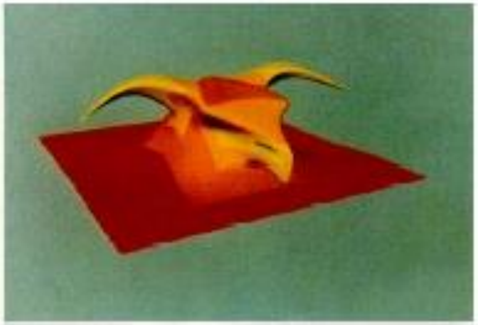
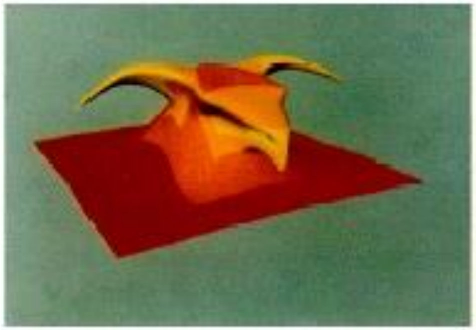


Figure 11e: Brow ridges are brought forward and down.

Figure 11f. The tip of the snout is refined twice and nostrils are constructed.



Basic Formulation

- Given (x_i, y_i), i=1,..,n, find $y = f(x)$
  - x can be a long vector (multi-dimensional features)
  - f can be many different types of functions and of many different orders

Linear Regression

- *f* is linear (hyper-plane) $$f(X) = \beta_0 + \sum_{j=1}^{p} X_j \beta_j.$$
  - *n*: # of training data $(x_1, y_1) \ldots (x_N, y_N)$
  - *p*: dimension of feature vectors $x_i = (x_{i1}, x_{i2}, \ldots, x_{ip})^T$
  - *p+1*: model variables $\beta = (\beta_0, \beta_1, \ldots, \beta_p)^T$
- Minimize RSS

$$\begin{aligned} \text{RSS}(\beta) &= \sum_{i=1}^{N} (y_i - f(x_i))^2 \\ &= \sum_{i=1}^{N} \Big(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j \Big)^2. \end{aligned}$$

Linear Regression (cont.)

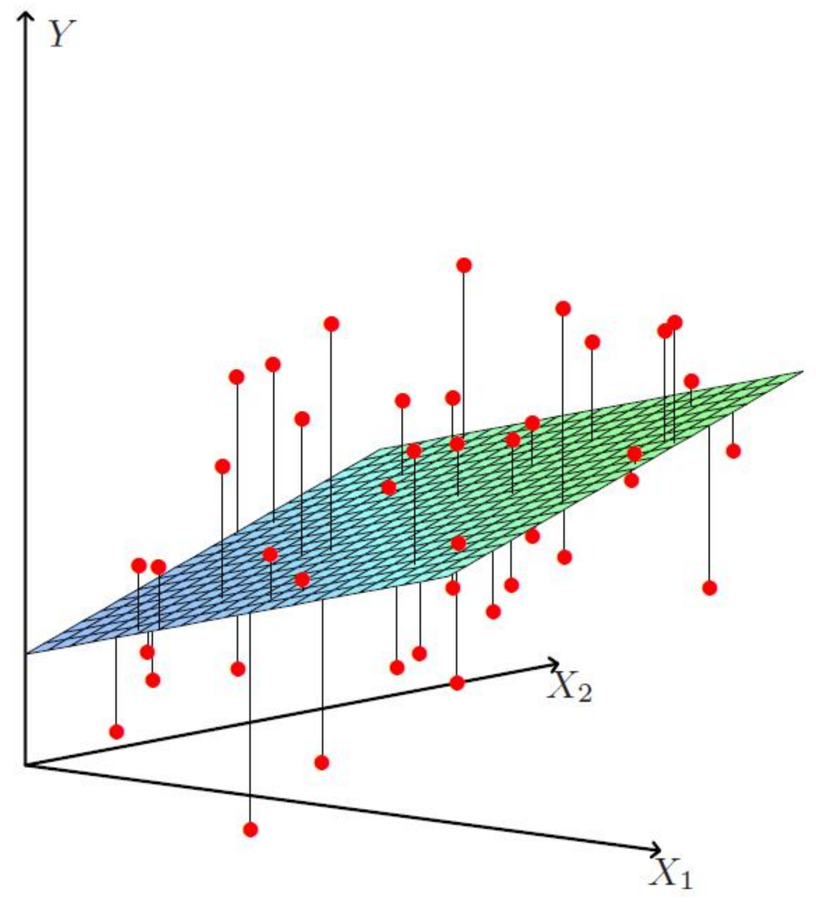
$$\mathrm{RSS}(\beta) = (\mathbf{y} - \mathbf{X}\beta)^T(\mathbf{y} - \mathbf{X}\beta).$$

$$\frac{\partial \mathrm{RSS}}{\partial \beta} = -2\mathbf{X}^T(\mathbf{y} - \mathbf{X}\beta)$$

$$\mathbf{X}^T(\mathbf{y} - \mathbf{X}\beta) = 0$$

$$\hat{\beta} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}.$$

Normal equation



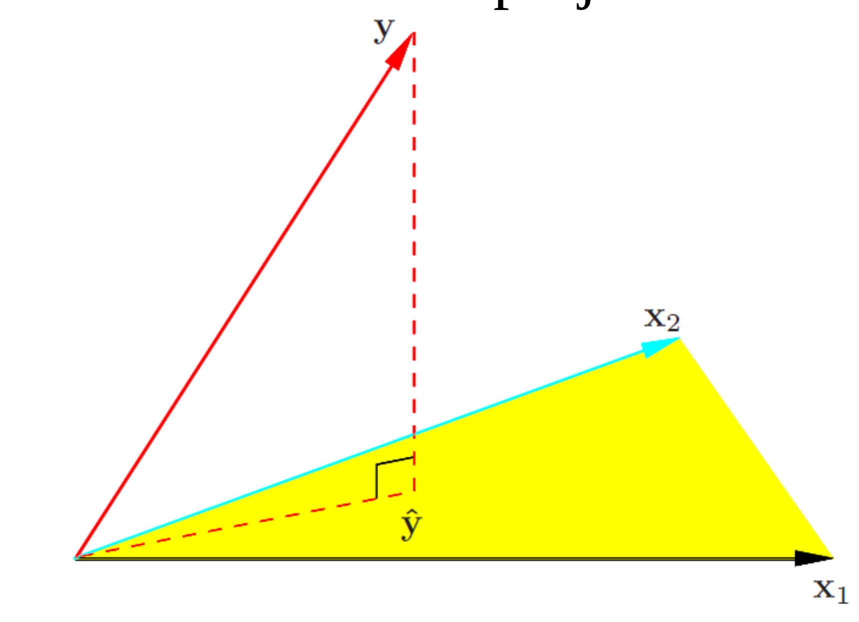
Why Linear Regression?

- Orthogonal projection onto the "known" space
- Minimum variance solutions
- Possible "massaging" *x* (feature vectors) to achieve nonlinearity

$$\text{RSS}(\beta) = \|\mathbf{y} - \mathbf{X}\beta\|^2$$

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

projection



Many Generalizations

- Polynomial models
- Basis (spline, Fourier, wavelet) expansion
- Regularization
- Outlier removals

Regularization

$$\hat{\beta} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}.$$

- When feature vectors are correlated ($x_2=3x_1$), the coefficient matrix become degenerate
- A large β_2 can be cancelled out by a equally large, negative β_1
- Think of regularization as controlling the magnitude of these coefficients

Ridge Regression

- λ is a "weighting" or "shrinking" term

$$\hat{\beta}^{\text{ridge}} = \underset{\beta}{\operatorname{argmin}} \left\{ \sum_{i=1}^{N} \left(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j \right)^2 + \lambda \sum_{j=1}^{p} \beta_j^2 \right\}.$$

- RSS is slightly different

$$\text{RSS}(\lambda) = (\mathbf{y} - \mathbf{X}\beta)^T(\mathbf{y} - \mathbf{X}\beta) + \lambda\beta^T\beta,$$

- Solution is

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

- See slides on RBF for details

General Curve Fitting

$y = f(x, a_1, a_2, \cdots, a_n)$	$y = ax^2 + bx + c$
n input points $(x_1, y_1), (x_2, y_2), \cdots, (x_n, y_n)$	3 input points $(1,1), (2,2), (3,1)$
n equations	3 equations
$y_1 = f(x_1, a_1, a_2, \cdots, a_n)$ $y_2 = f(x_2, a_1, a_2, \cdots, a_n)$ $\cdots$ $y_n = f(x_n, a_1, a_2, \cdots, a_n)$	$a + b + c = 1$ $4a + 2b + c = 2$ $9a + 3b + c = 1$
$\begin{bmatrix} f(x_1) \\ \cdots \\ f(x_n) \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \cdots \\ a_n \end{bmatrix} = \begin{bmatrix} y_1 \\ \cdots \\ y_n \end{bmatrix}$	$\begin{bmatrix} 1 & 1 & 1 \\ 4 & 2 & 1 \\ 9 & 3 & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix}$
solve for $a_1, \cdots, a_n$	$a = -1, b = 4, c = -2$

University of California Santa Barbara

General Least Square Regression

$$\min_{\theta=(a_0,a_1,\ldots,a_{n-1})} E$$

$$\text{where } E = \sum_{i=1}^{m} (y_i - \hat{y}_i)^2 =$$

$$\min_{\theta=(a_0,a_1,\ldots,a_{n-1})} \sum_{i=1}^{m} (y_i - (a_{n-1}x_i^{\,n-1} + a_{n-2}x_i^{\,n-2} + \ldots + a_1 x_i^{\,1} + a_0))^2$$

$$\frac{\partial E}{\partial a_j} = 0,\ j = 1,\ldots,n$$

$$\sum_{i=1}^{m} x_i^{\,j} (y_i - (a_{n-1}x_i^{\,n-1} + a_{n-2}x_i^{\,n-2} + \ldots + a_1 x_i^{\,1} + a_0)) = 0$$

General Least Square Regression

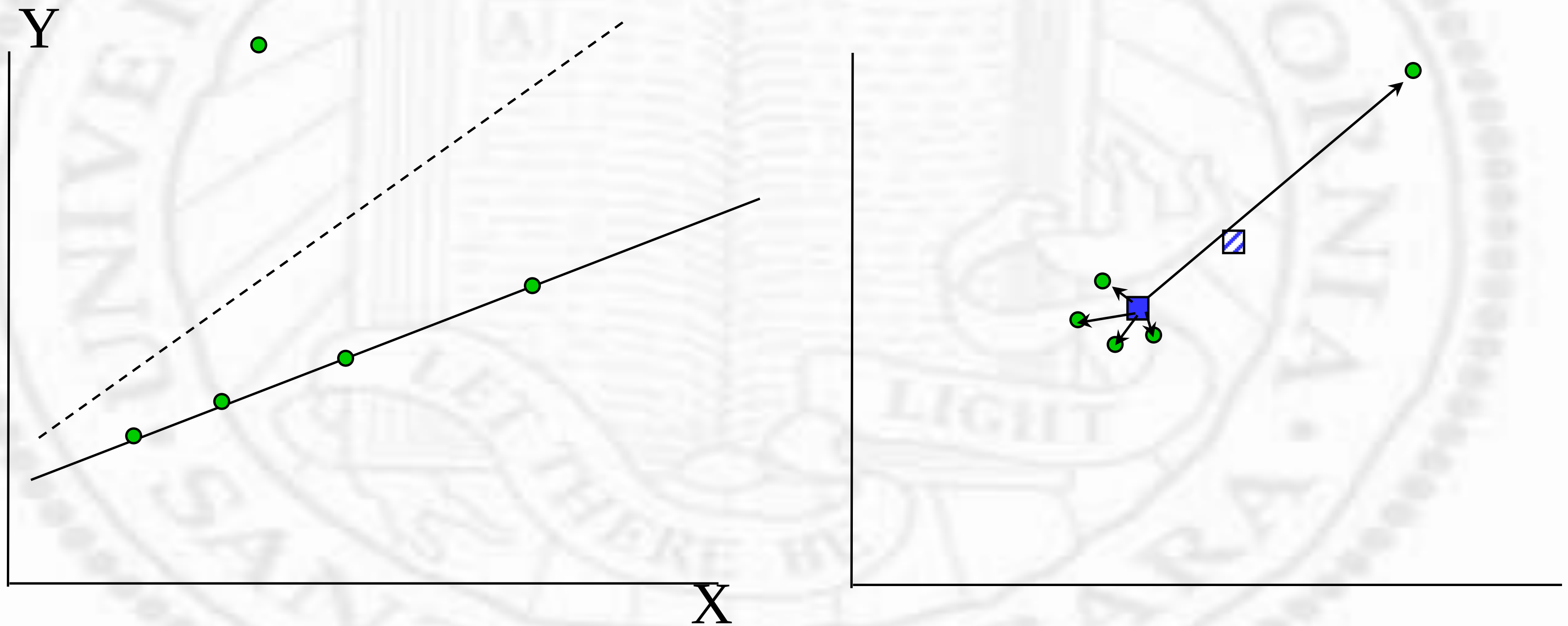
$$\sum_{i=1}^{m} x_i^{\,j}(y_i - (a_{n-1}x_i^{\,n-1} + a_{n-2}x_i^{\,n-2} + \ldots + a_1 x_i^{\,1} + a_0)) = 0$$

$$(\sum_{i=1}^{m} x_i^{\,j} x_i^{\,n-1})a_{n-1} + (\sum_{i=1}^{m} x_i^{\,j} x_i^{\,n-2})a_{n-2} + \ldots + (\sum_{i=1}^{m} x_i^{\,j} x_i^{\,1})a_1 + (\sum_{i=1}^{m} x_i^{\,j})a_0 = \sum_{i=1}^{m} x_i^{\,j} y_i$$

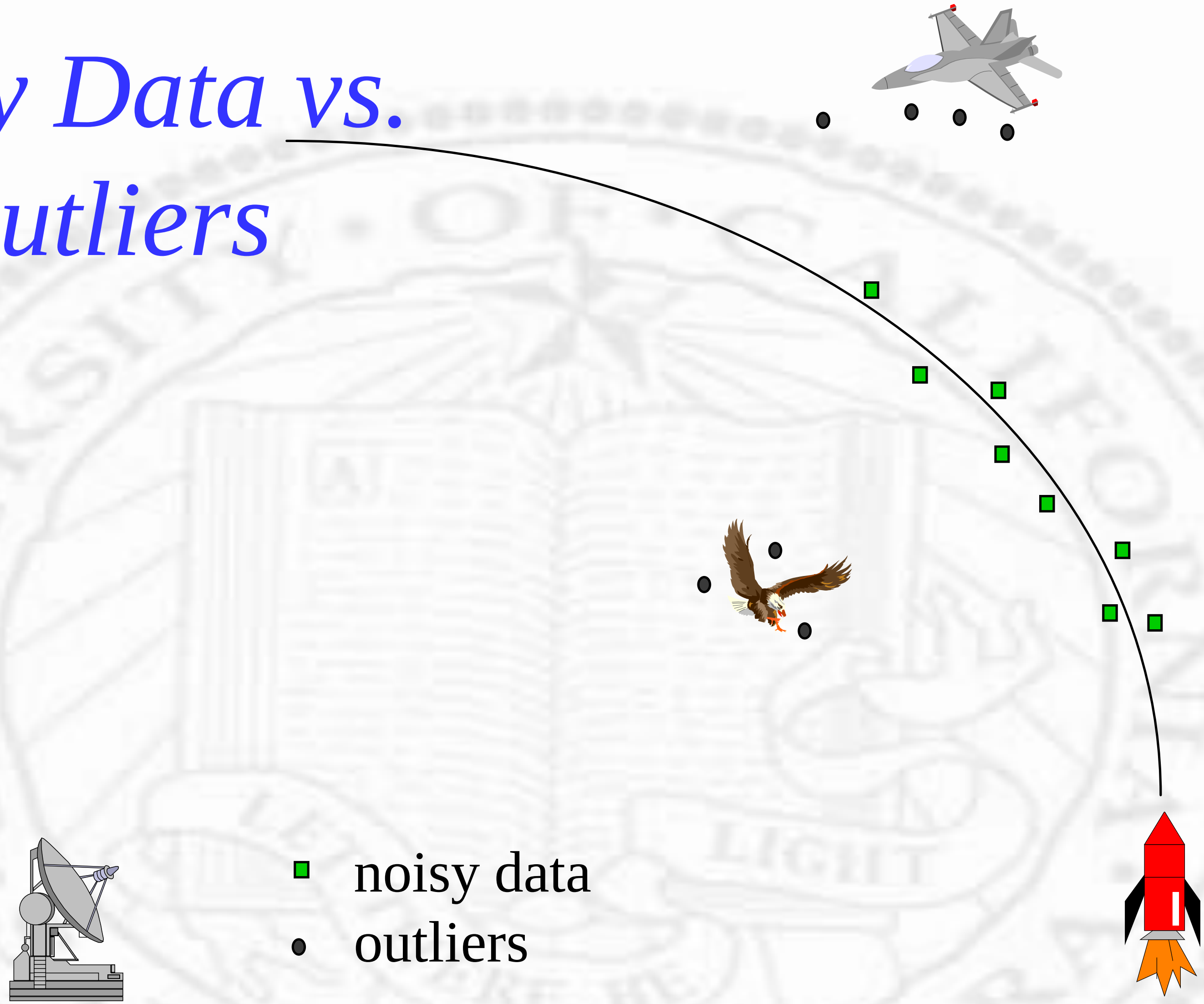
$$\begin{bmatrix} \sum_{i=1}^{m} x_i^{\,n-1} x_i^{\,n-1} & \sum_{i=1}^{m} x_i^{\,n-1} x_i^{\,n-2} & \cdots & \sum_{i=1}^{m} x_i^{\,n-1} \\ \sum_{i=1}^{m} x_i^{\,n-2} x_i^{\,n-1} & \sum_{i=1}^{m} x_i^{\,n-2} x_i^{\,n-2} & \cdots & \sum_{i=1}^{m} x_i^{\,n-2} \\ \cdots & \cdots & \cdots & \vdots \\ \sum_{i=1}^{m} x_i^{\,n-1} & \sum_{i=1}^{m} x_i^{\,n-2} & \cdots & \sum_{i=1}^{m} 1 \end{bmatrix} \begin{bmatrix} a_{n-1} \\ a_{n-2} \\ \vdots \\ a_o \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{m} x_i^{\,n-1} y_i \\ \sum_{i=1}^{m} x_i^{\,n-2} y_i \\ \vdots \\ \sum_{i=1}^{m} y_i \end{bmatrix}$$

Caveats

- LS
  - Democracy, everybody gets an equal say
  - Perform badly with “outliers”

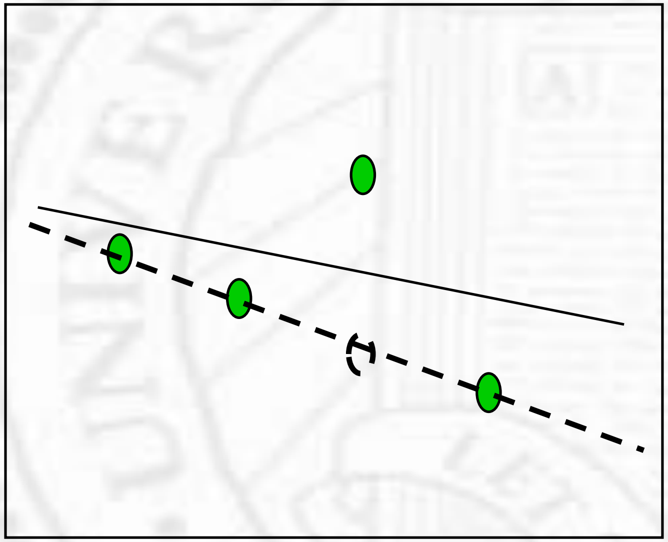


Noisy Data vs. Outliers

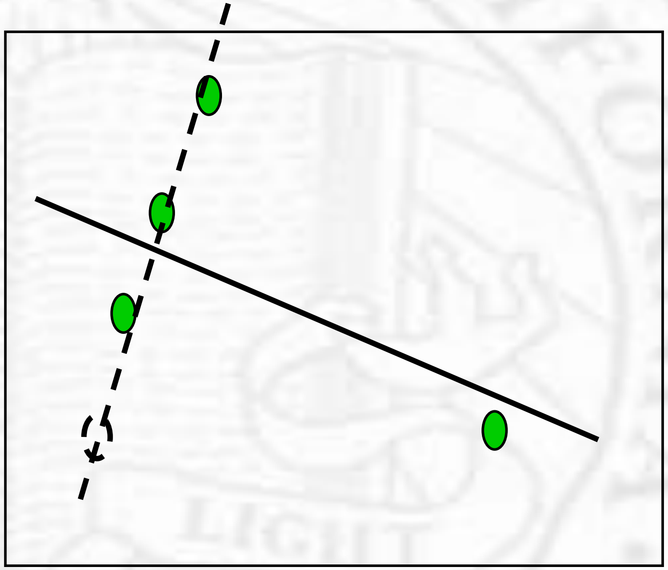


Outliers

- Outliers (y)



- Outliers (x, leverage points)



Randomized Algorithm

- Choose p points at random from the set of n data points
- Compute the fit of model to the p points
- Compute the median of the fitting error for the remaining n-p points
- The fitting procedure is repeated until a fit is found with sufficiently small median of squared residuals or up to some predetermined number of fitting steps (Monte Carlo Sampling)

How Many Trials?

- Well, theoretically it is *C(n,p)* to find all possible *p*-tuples
- Very expensive

$1 - (1 - (1 - \varepsilon)^p)^m$

ε : fraction of bad data

$(1 - \varepsilon)$: fraction of good data

$(1 - \varepsilon)^p$: all p samples are good

$1 - (1 - \varepsilon)^p$: at least one sample is bad

$(1 - (1 - \varepsilon)^p)^m$: got bad data in all m tries

$1 - (1 - (1 - \varepsilon)^p)^m$: got at least one good p set in m tries

How Many Trials (cont.)

- Make sure the probability is high (e.g. >95%)
- given p and epsilon, calculate m

p	5%	10 %	20 %	25 %	30 %	40 %	50 %
1	1	2	2	3	3	4	5
2	2	2	3	4	5	7	11
3	2	3	5	6	8	13	23
4	2	3	6	8	11	22	47
5	3	4	8	12	17	38	95

Best Practice

- Randomized selection can completely remove outliers
- "plutocratic"
- Results are based on a small set of features

- LS is most fair, everyone get an equal say
- "democratic"
- But can be seriously influenced by bad data

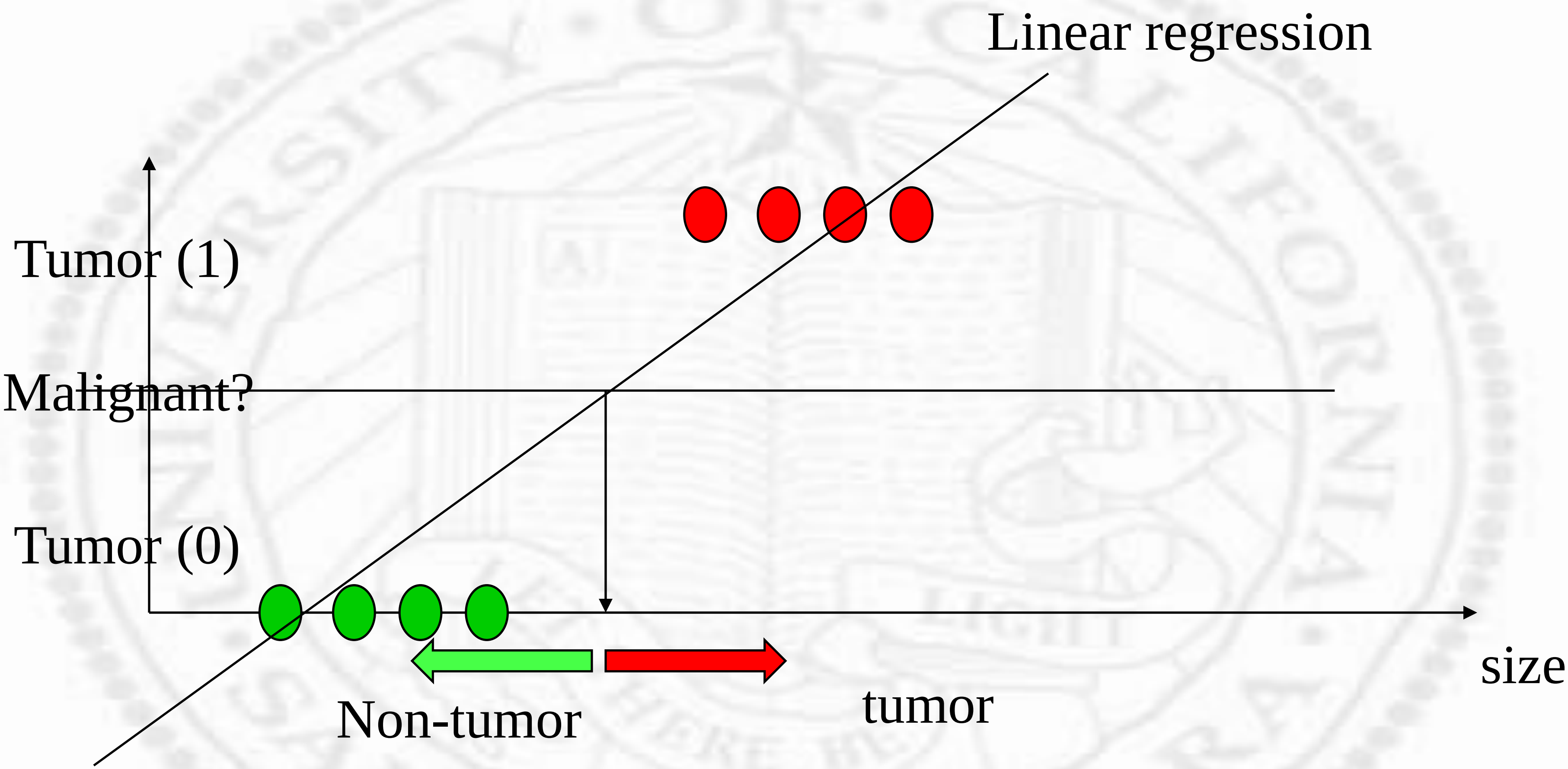
- Use randomized algorithm to remove outliers
- Use LS for final "polishing" of results (using all "good" data)
- Allow up to 50% outliers theoretically

Logistic Regression

Logistic Regression

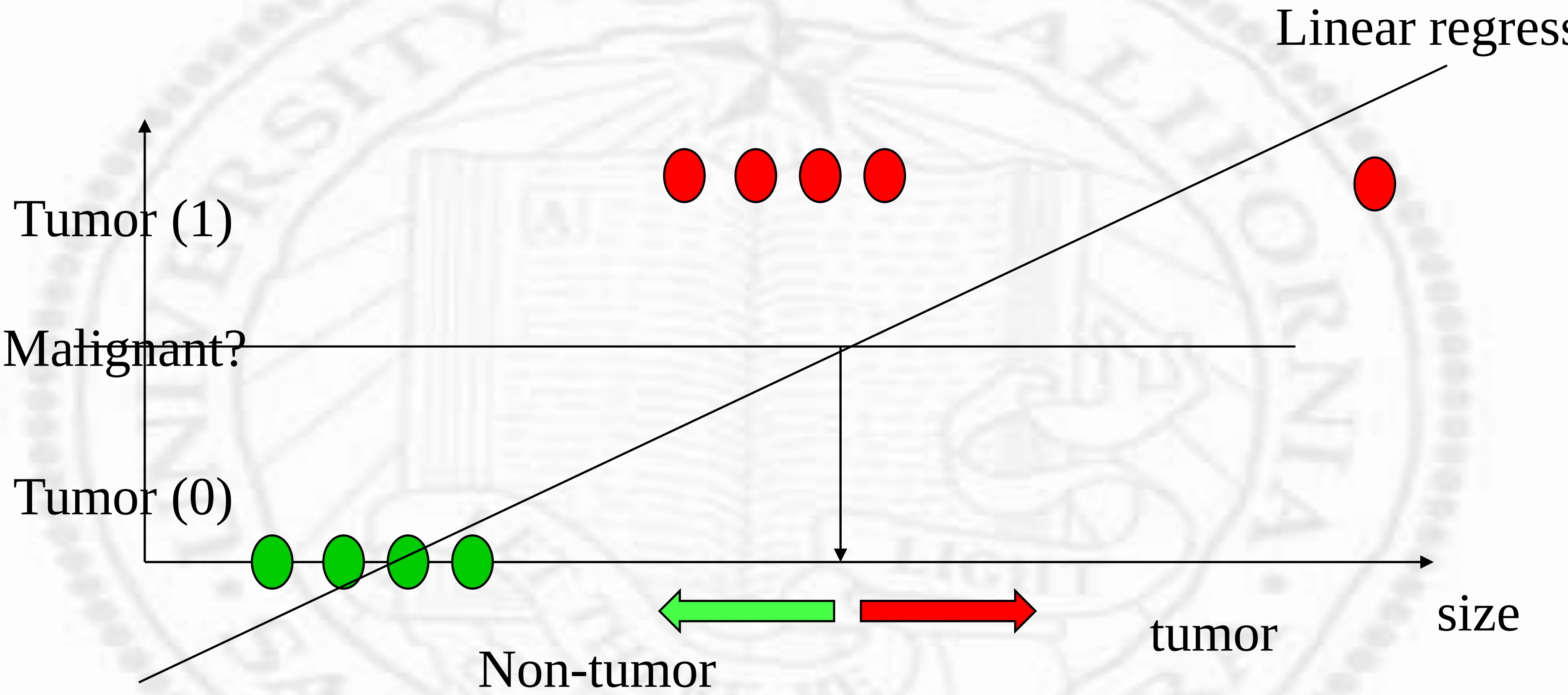
- Despite the name, LR is a classification scheme, not a “regression“ (curve or surface fitting routine)
- Considered more general than LDA, but formulated in a way to be solved efficiently using Gradient Descent
- Introduce the concept of margin

Motivation



y(tumor/not tumor) = f(size) = a*size +b

Motivation



y(tumor/not tumor) = f(size) = a*size +b

Problem: tumor (1) and not tumor (0) are class labels

Lesson Learned

- Regression should not be used for classification
  - Linear regression pays attention to all data equally, outliers can easily skew the results (hence, the concepts of “inliers” or “importance”)
  - Linear regression outputs a continuous range of values, while a classification scheme outputs [0..1]

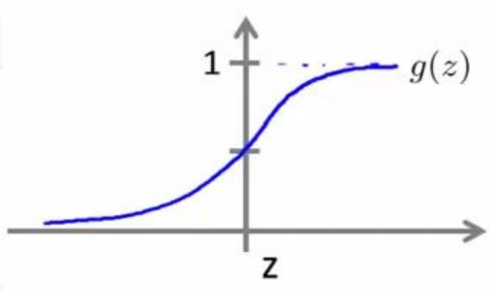
Details

$0 \le h_\theta(x) \le 1$

$\theta \leftrightarrow \omega$

$h_\theta(x) = g(\theta^T x)$

$g(z) = \frac{1}{1+e^{-z}}$



Compress the parameter range

y=$\omega^T x \leftrightarrow$ z=$\theta^T x$

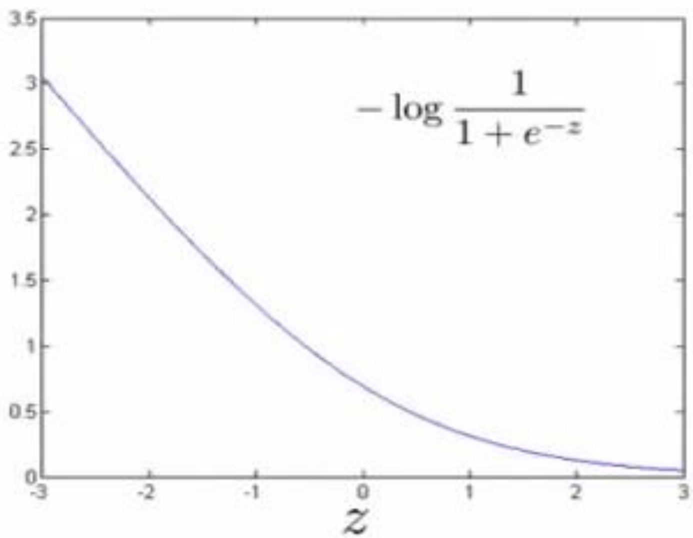
$$J(\theta) = \frac{1}{m} \sum_{i=1}^{m} \mathrm{Cost}(h_\theta(x^{(i)}), y^{(i)})$$

$$\mathrm{Cost}(h_\theta(x), y) = \begin{cases} -\log(h_\theta(x)) & \text{if } y = 1 \\ -\log(1 - h_\theta(x)) & \text{if } y = 0 \end{cases}$$

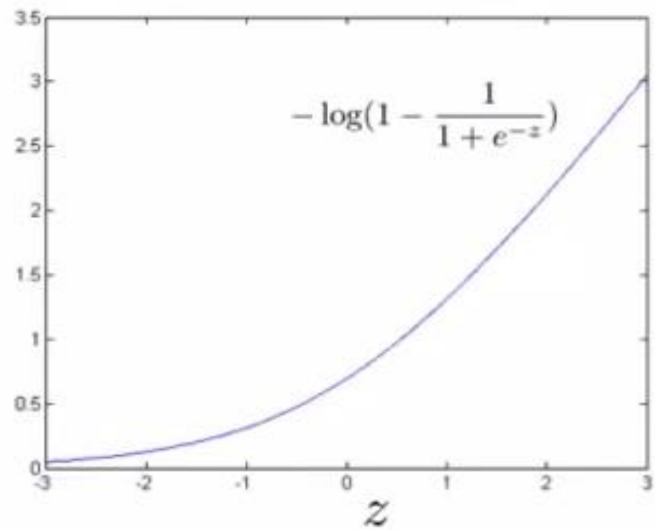
Note: $y = 0$ or 1 always

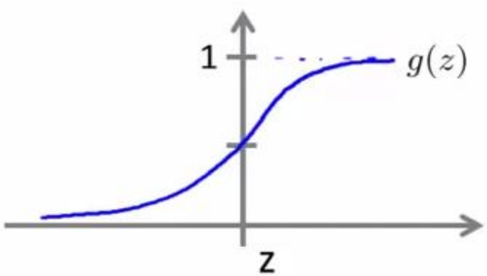
$$J(\theta) = \frac{1}{m} \sum_{i=1}^{m} \mathrm{Cost}(h_\theta(x^{(i)}), y^{(i)})$$
$$= -\frac{1}{m}\left[\sum_{i=1}^{m} y^{(i)} \log h_\theta(x^{(i)}) + (1 - y^{(i)}) \log\left(1 - h_\theta(x^{(i)})\right)\right] + \frac{\lambda}{2m} \sum_{j=1}^{n} \theta_j^2$$

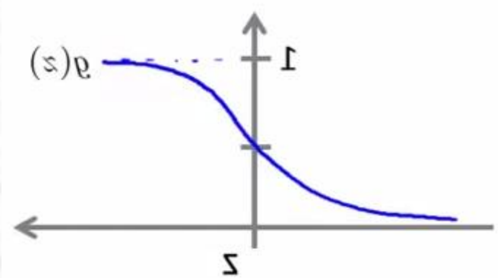
If $y = 1$ (want $\theta^T x \gg 0$):



If $y = 0$ (want $\theta^T x \ll 0$):







Repeat $\{$

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta)$$

$\}$ (simultaneously update all θ_j)

Multi-Classes

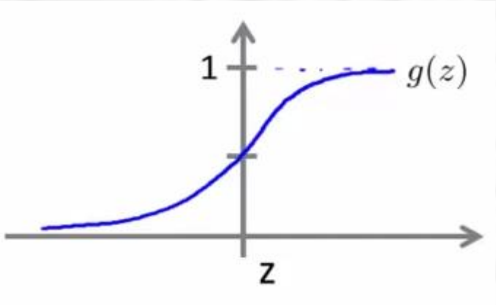
- Generalization of binary case
  - k: number of classes
  - m: number of samples
  - 1{.}: indicator function, true:1, false: 0

$$J(\theta) = -\left[\sum_{i=1}^{m}\sum_{k=1}^{K} 1\left\{y^{(i)} = k\right\} \log \frac{\exp(\theta^{(k)\top} x^{(i)})}{\sum_{j=1}^{K} \exp(\theta^{(j)\top} x^{(i)})}\right]$$

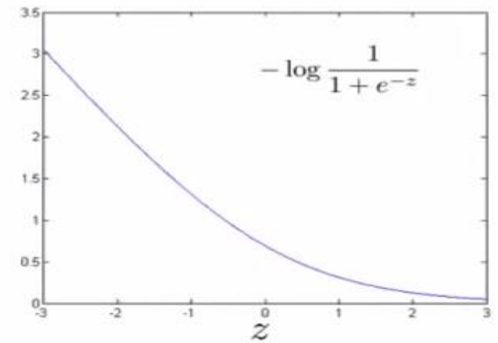
Multi-Classes

- $h_\theta(x)$ functions are probabilities
  - $h_\theta(x)$ in the range of 0 and 1
  - With correct class, $h_\theta(x)$ ->1, or small penalty (-log)

$$h_\theta(x) = \begin{bmatrix} P(y=1|x;\theta) \\ P(y=2|x;\theta) \\ \vdots \\ P(y=K|x;\theta) \end{bmatrix} = \frac{1}{\sum_{j=1}^{K} \exp(\theta^{(j)\top} x)} \begin{bmatrix} \exp(\theta^{(1)\top} x) \\ \exp(\theta^{(2)\top} x) \\ \vdots \\ \exp(\theta^{(K)\top} x) \end{bmatrix}$$



If $y = 1$ (want $\theta^T x \gg 0$):



Numerical Solutions

$$\nabla_{\theta^{(k)}} J(\theta) = -\sum_{i=1}^{m} \left[x^{(i)} \left(1\{y^{(i)} = k\} - P(y^{(i)} = k | x^{(i)}; \theta) \right) \right]$$