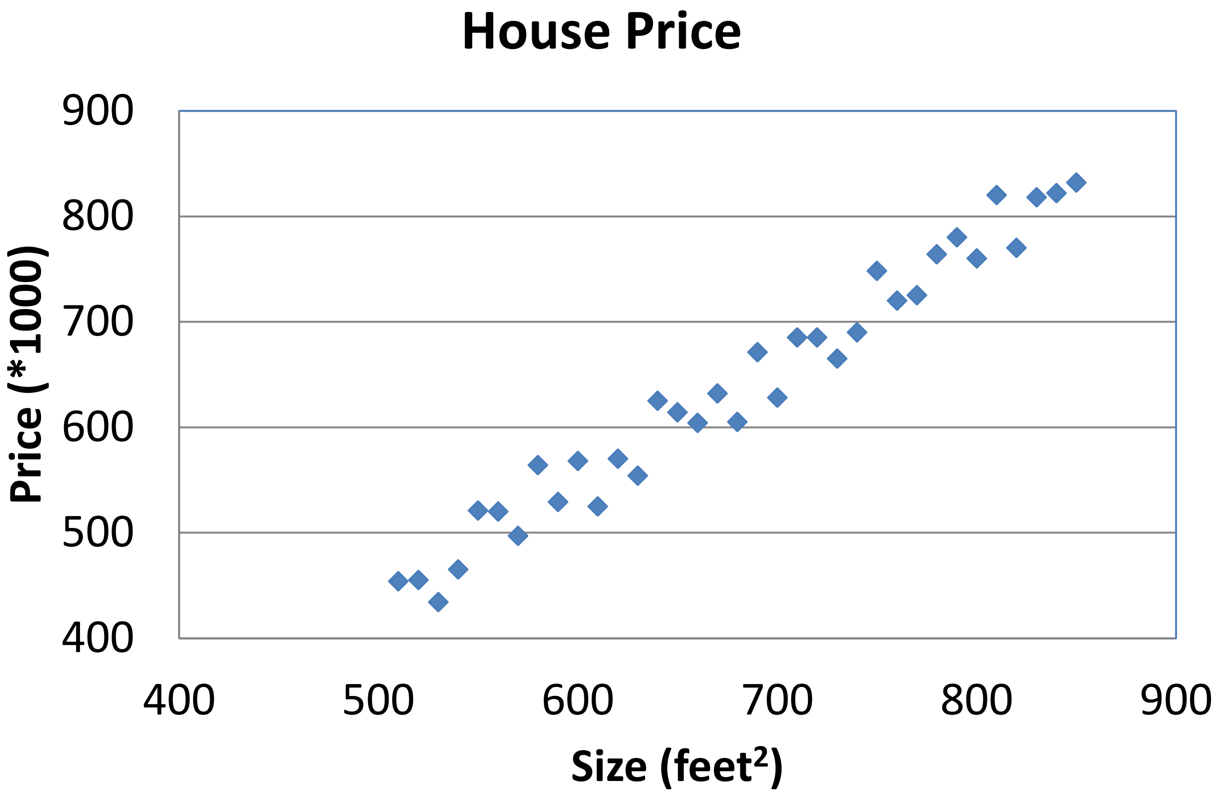
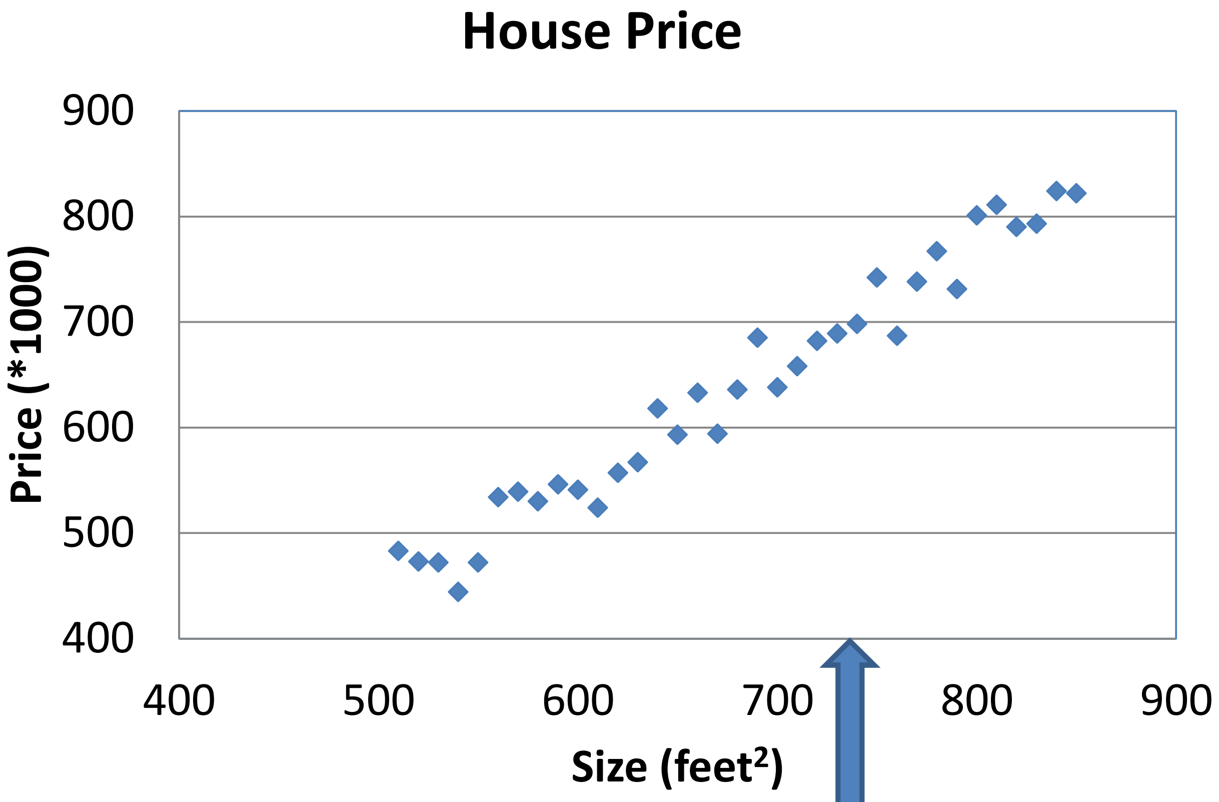
Linear Regression

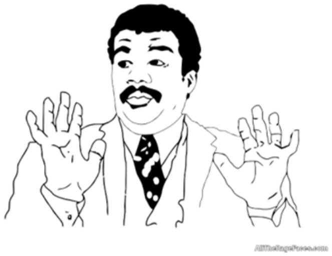
Avinava Dubey

Typical Example



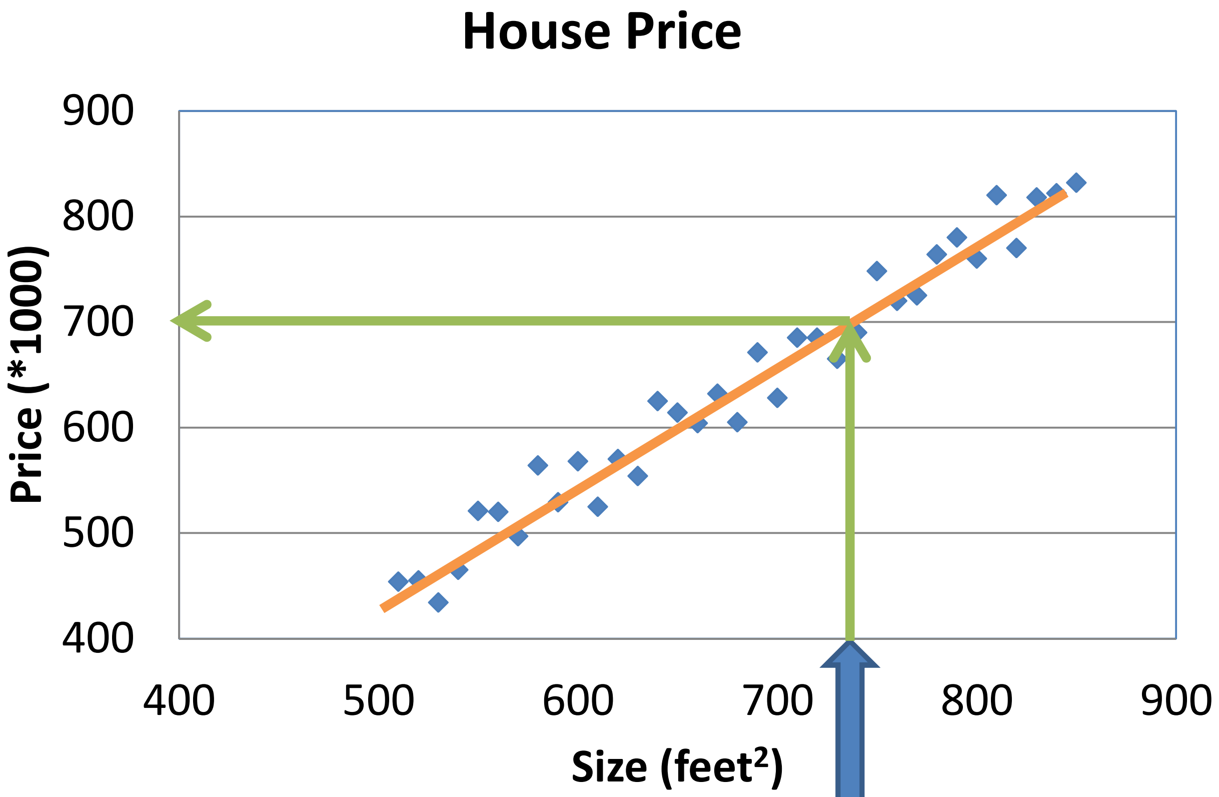
Typical Example

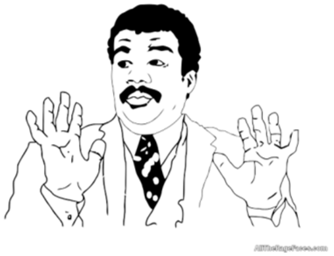




What is the price for 725 sq feet

Typical Example

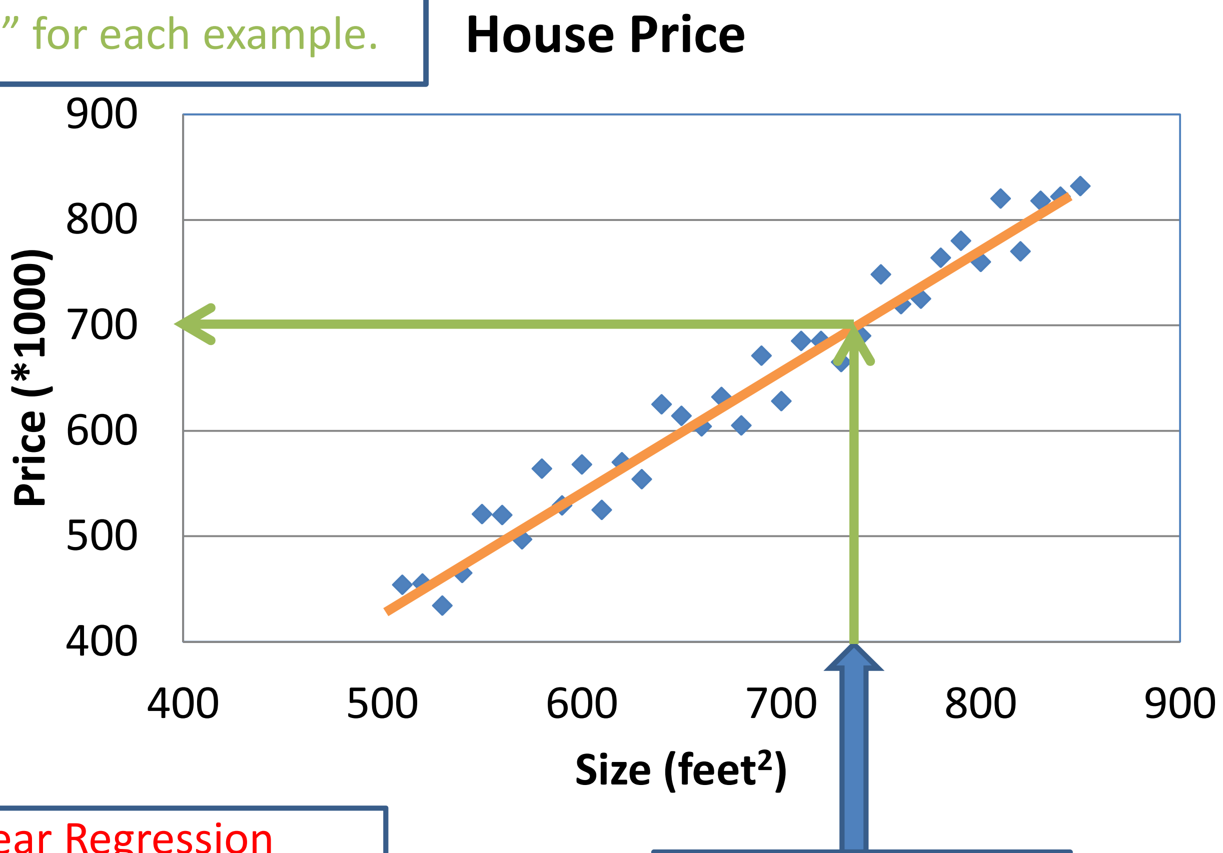




What is the price for 725 sq feet

Typical Example

Supervised Learning: Given the "right answer" for each example.



Linear Regression problem: Predict real valued output (What is the other type?)

What is the price for 725 sq feet

Regression

- Training Dataset

Size (feets^2)	Price (*1000)
510	413
650	629
810	840

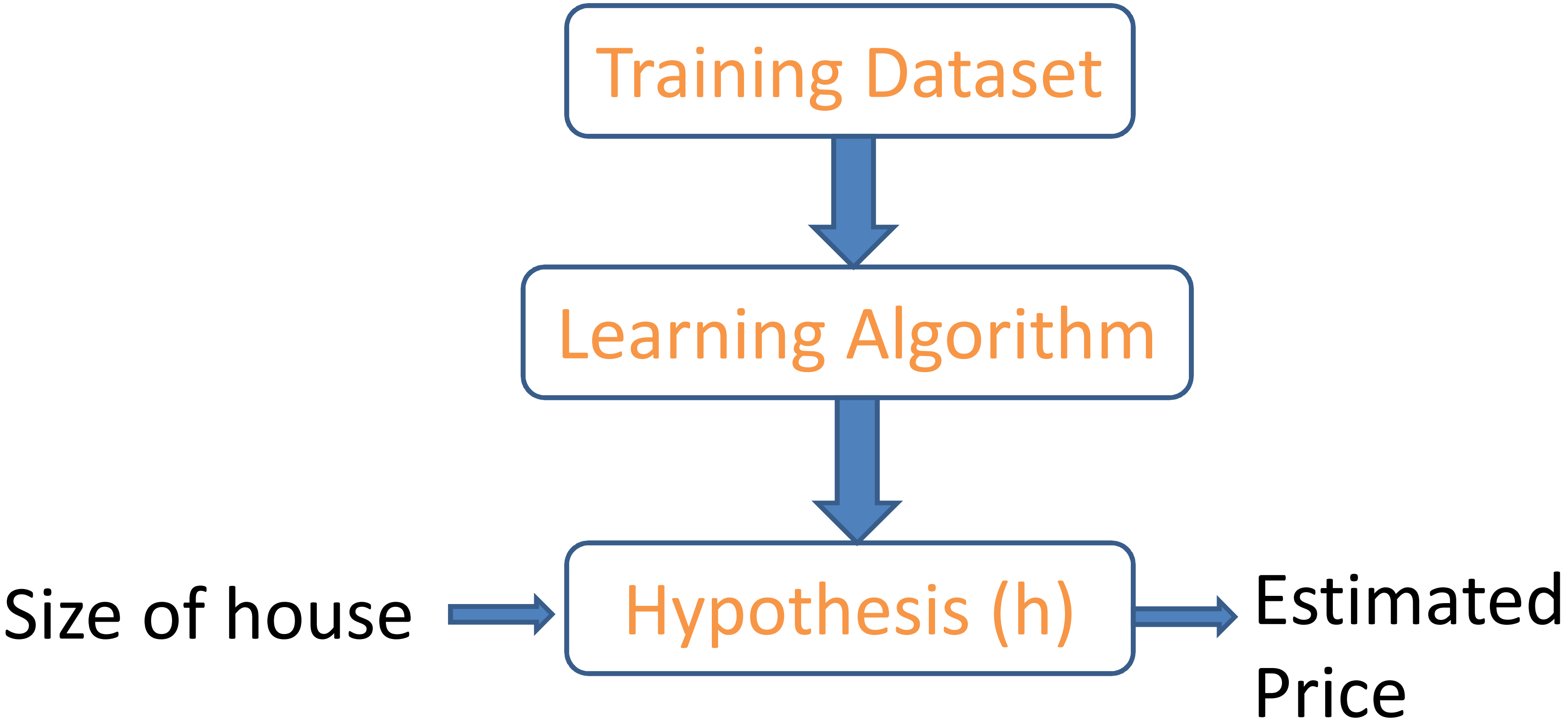
$X^{(i)}$ → 650

629 ← $Y^{(i)}$

Notation:

- m is the number of training examples
- x input features
- y output variable

Supervised Learning



Training Dataset

Learning Algorithm

Size of house

Hypothesis (h)

Estimated Price

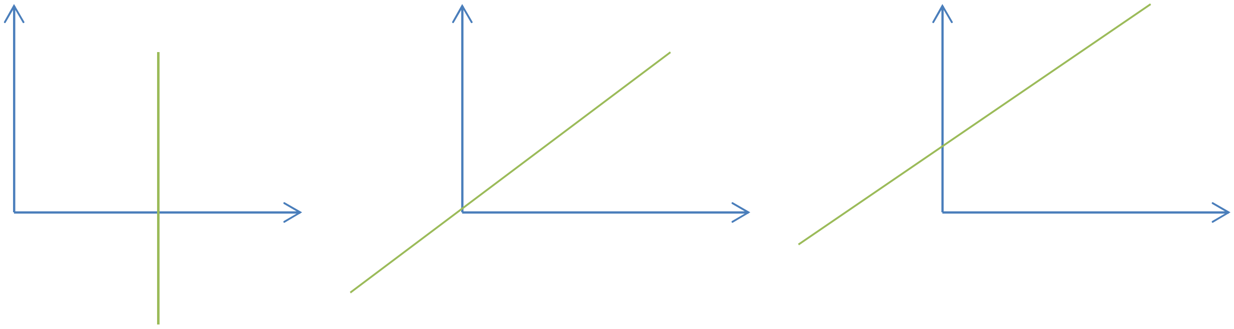
h maps input data x to output y

Linear Regression

- Hypothesis Set: Let output be a linear function of input data ie

$$h_a(x) = a_1x + a_0$$

- Parameters: a_1 ,a_0



Which h to choose

- Choose an h so that the prediction of the hypothesis is same as that of Y

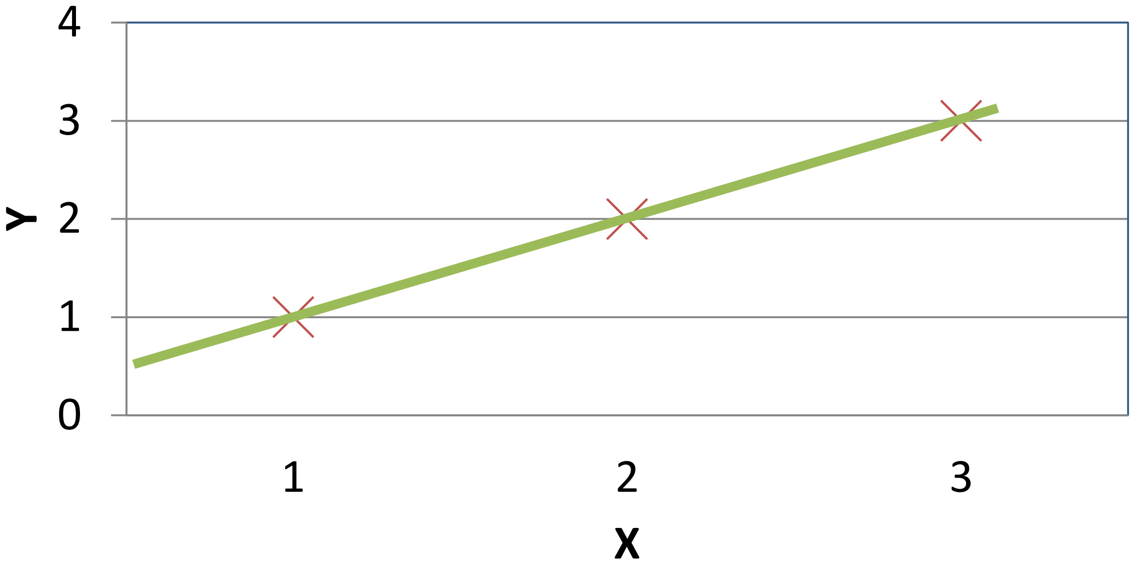
$$\boxed{\text{mimimize } J(a_1,a_0) = \frac{1}{2m}\sum_i (h(x^{(i)}) - Y^{(i)}\ )\text{\textasciicircum}2}$$

# of training samples Prediction actual

- J also known as cost function, loss function etc.

Simpler hypothesis

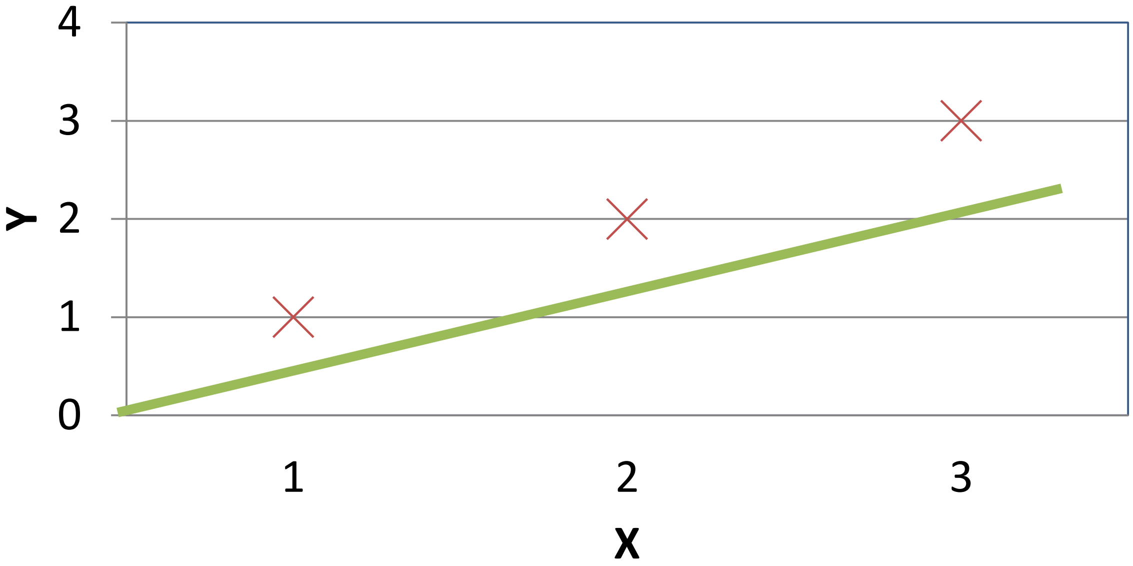
- $h(a_1) = a_1x$
- $a_1 = 1$



- $J(a_1) = \frac{1}{2m} \sum_i (h(x^{(i)}) - Y^{(i)})\text{^}2 = 0$

Simpler hypothesis

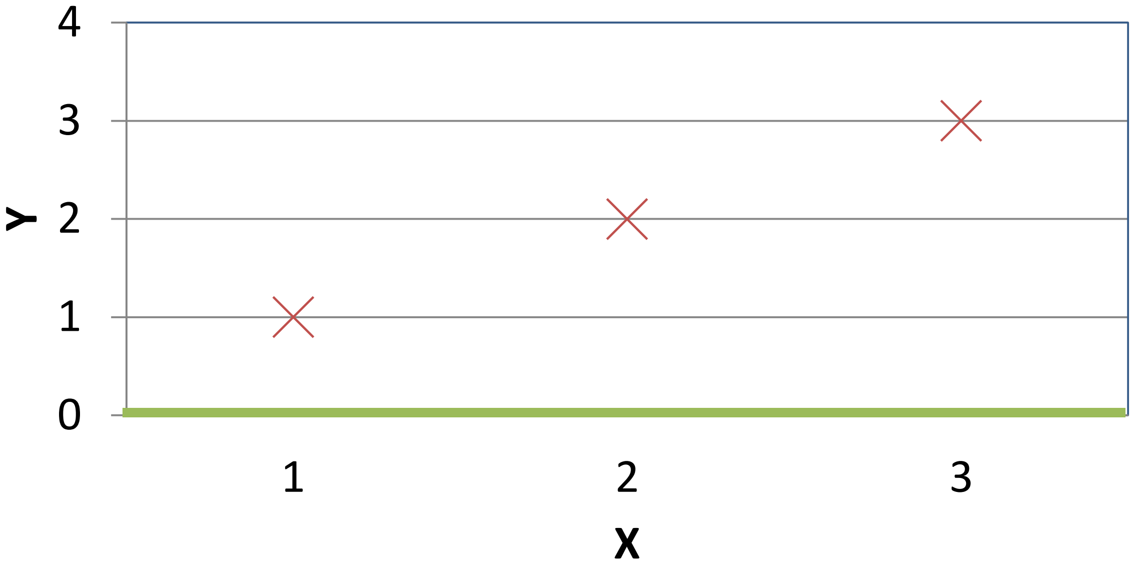
- $h(a_1) = a_1x$
- $a_1 = 0.5$



- $J(a_1) = \frac{1}{2m}\sum_i (h(x^{(i)}) - Y^{(i)})\^{}2 = 0.68$

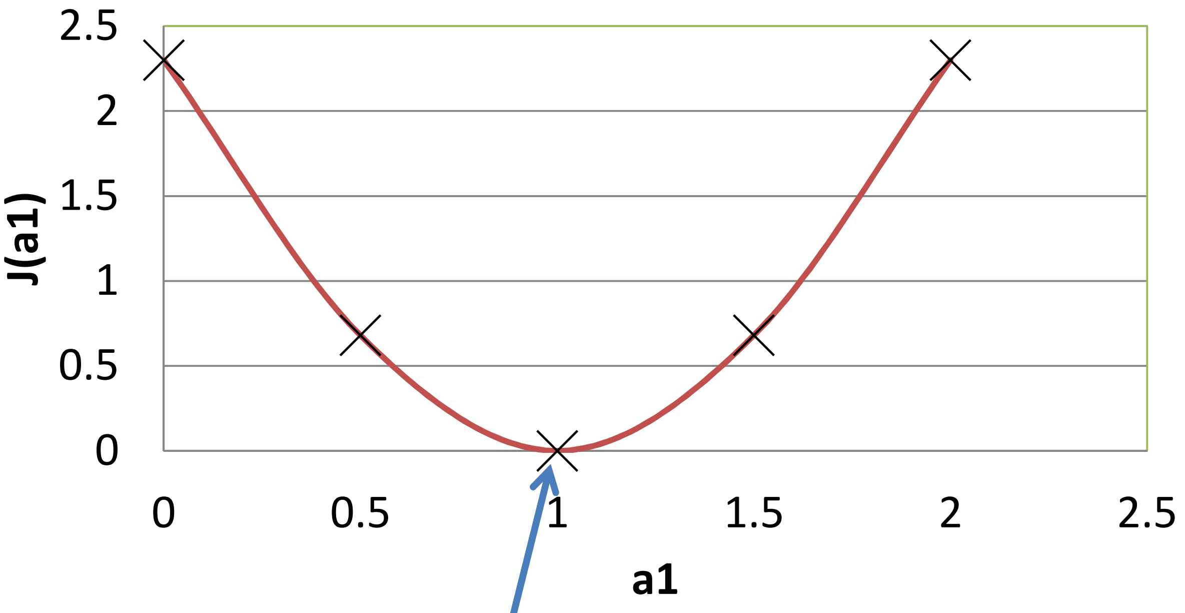
Simpler hypothesis

- $h(a_1) = a_1x$
- $a_1 = 0$



- $J(a_1) = \frac{1}{2m}\sum_i (h(x^{(i)}) - Y^{(i)})\text{^}2 = 2.3$

Simpler Hypothesis



a_1 = 1 minimizes J and corresponds to finding a straight line that fits the data well

Finding optimal parameter

- Analytical Solution:-

$$J(a_1) = \sum_i (y^{(i)2} - 2a_1 y^{(i)} x^{(i)} + a_1^2 x^{(i)2})$$

- Differentiate wrt a_1 and substitute as zero

$$\sum_i \left(-2y^{(i)} x^{(i)} + 2a_1 x^{(i)2}\right) = 0$$

$$a_1 = \frac{\sum_i y^{(i)} x^{(i)}}{\sum_i x^{(i)2}}$$

Vector Algebra/Calculus

- Let $Y = \begin{bmatrix} y^{(1)} \\ y^{(2)} \\ \ddots \\ y^{(m)} \end{bmatrix}$ and $X = \begin{bmatrix} 1 & {x_1}^{(1)} & {x_k}^{(d)} \\ 1 & {x_1}^{(2)} & {x_k}^{(d)} \\ \vdots & \ddots & \vdots \\ 1 & {x_1}^{(m)} & {x_k}^{(d)} \end{bmatrix}$

- The objective can be written as:

$$\boxed{J(\boldsymbol{a}) = ||Y \ - X\boldsymbol{a}||^2}$$

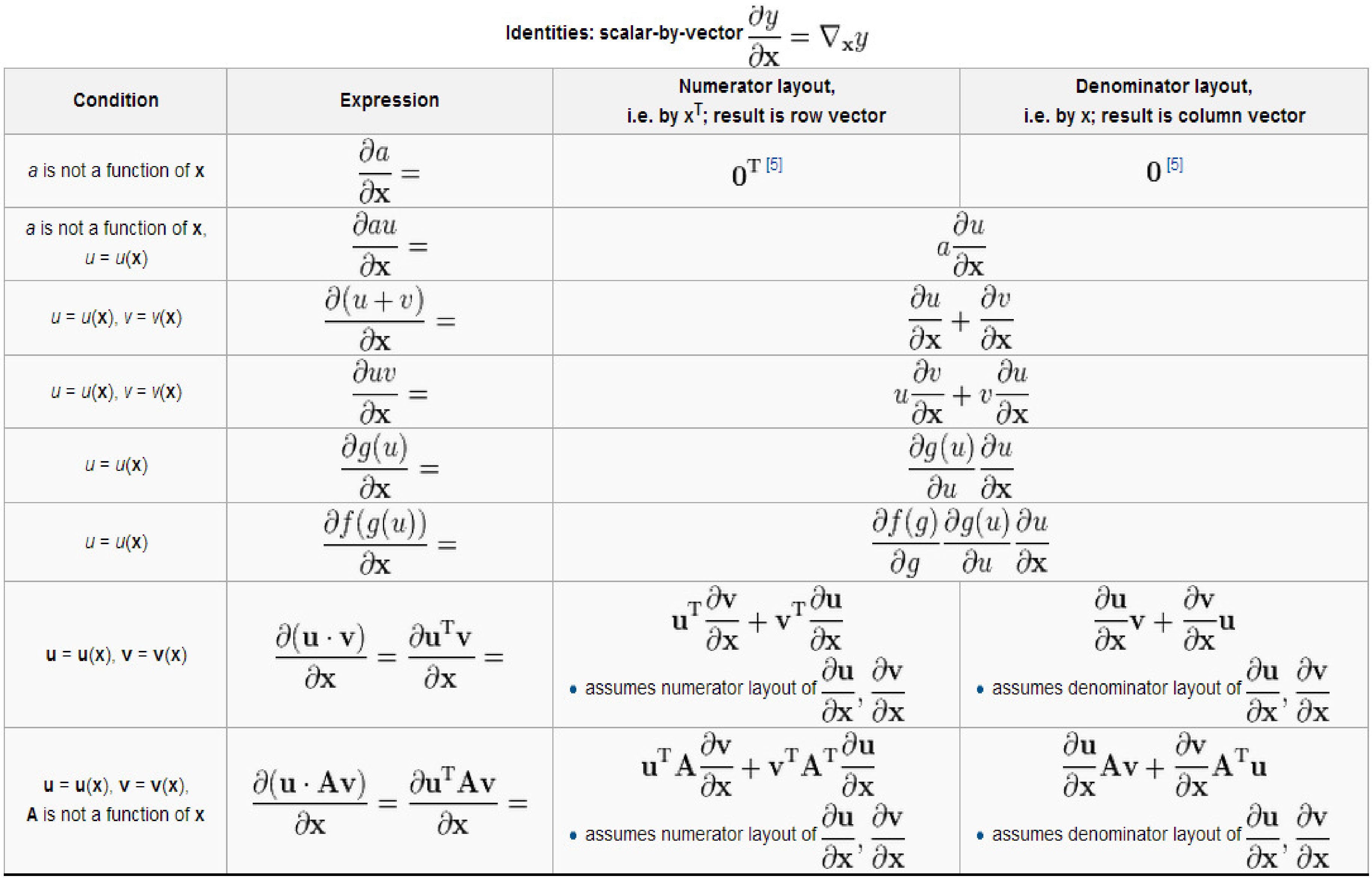
Vector Algebra

- Square of a vector: $\boxed{||\boldsymbol{a}||^2 = \boldsymbol{a}^T\boldsymbol{a}}$

- Diff. wrt a vector:

$$\boxed{\frac{\delta J}{\delta \boldsymbol{a}} = \begin{bmatrix} \frac{\delta J}{\delta a_1} \\ \frac{\delta J}{\delta a_2} \\ \vdots \\ \frac{\delta J}{\delta a_d} \end{bmatrix}}$$

Vector Calculus



Identities: scalar-by-vector $\frac{\partial y}{\partial \mathbf{x}} = \nabla_{\mathbf{x}} y$

Condition	Expression	Numerator layout, i.e. by $\mathbf{x}^T$; result is row vector	Denominator layout, i.e. by **x**; result is column vector
a is not a function of **x**	$\frac{\partial a}{\partial \mathbf{x}} =$	$\mathbf{0}^T$ [5]	$\mathbf{0}$ [5]
a is not a function of **x**, $u = u(\mathbf{x})$	$\frac{\partial au}{\partial \mathbf{x}} =$	$a\frac{\partial u}{\partial \mathbf{x}}$	
$u = u(\mathbf{x}), v = v(\mathbf{x})$	$\frac{\partial (u+v)}{\partial \mathbf{x}} =$	$\frac{\partial u}{\partial \mathbf{x}} + \frac{\partial v}{\partial \mathbf{x}}$	
$u = u(\mathbf{x}), v = v(\mathbf{x})$	$\frac{\partial uv}{\partial \mathbf{x}} =$	$u\frac{\partial v}{\partial \mathbf{x}} + v\frac{\partial u}{\partial \mathbf{x}}$	
$u = u(\mathbf{x})$	$\frac{\partial g(u)}{\partial \mathbf{x}} =$	$\frac{\partial g(u)}{\partial u}\frac{\partial u}{\partial \mathbf{x}}$	
$u = u(\mathbf{x})$	$\frac{\partial f(g(u))}{\partial \mathbf{x}} =$	$\frac{\partial f(g)}{\partial g}\frac{\partial g(u)}{\partial u}\frac{\partial u}{\partial \mathbf{x}}$	
$\mathbf{u} = \mathbf{u}(\mathbf{x}), \mathbf{v} = \mathbf{v}(\mathbf{x})$	$\frac{\partial (\mathbf{u}\cdot\mathbf{v})}{\partial \mathbf{x}} = \frac{\partial \mathbf{u}^T\mathbf{v}}{\partial \mathbf{x}} =$	$\mathbf{u}^T\frac{\partial \mathbf{v}}{\partial \mathbf{x}} + \mathbf{v}^T\frac{\partial \mathbf{u}}{\partial \mathbf{x}}$ • assumes numerator layout of $\frac{\partial \mathbf{u}}{\partial \mathbf{x}}, \frac{\partial \mathbf{v}}{\partial \mathbf{x}}$	$\frac{\partial \mathbf{u}}{\partial \mathbf{x}}\mathbf{v} + \frac{\partial \mathbf{v}}{\partial \mathbf{x}}\mathbf{u}$ • assumes denominator layout of $\frac{\partial \mathbf{u}}{\partial \mathbf{x}}, \frac{\partial \mathbf{v}}{\partial \mathbf{x}}$
$\mathbf{u} = \mathbf{u}(\mathbf{x}), \mathbf{v} = \mathbf{v}(\mathbf{x})$, **A** is not a function of **x**	$\frac{\partial (\mathbf{u}\cdot\mathbf{A}\mathbf{v})}{\partial \mathbf{x}} = \frac{\partial \mathbf{u}^T\mathbf{A}\mathbf{v}}{\partial \mathbf{x}} =$	$\mathbf{u}^T\mathbf{A}\frac{\partial \mathbf{v}}{\partial \mathbf{x}} + \mathbf{v}^T\mathbf{A}^T\frac{\partial \mathbf{u}}{\partial \mathbf{x}}$ • assumes numerator layout of $\frac{\partial \mathbf{u}}{\partial \mathbf{x}}, \frac{\partial \mathbf{v}}{\partial \mathbf{x}}$	$\frac{\partial \mathbf{u}}{\partial \mathbf{x}}\mathbf{A}\mathbf{v} + \frac{\partial \mathbf{v}}{\partial \mathbf{x}}\mathbf{A}^T\mathbf{u}$ • assumes denominator layout of $\frac{\partial \mathbf{u}}{\partial \mathbf{x}}, \frac{\partial \mathbf{v}}{\partial \mathbf{x}}$

Linear Regression

- Input: X of dim m*(d+1), output Y of dim m*1
- Objective:- $\boxed{\text{Maximize } J(\boldsymbol{a}) = ||Y \; - X\boldsymbol{a}||^2}$
- Parameter:- **a**

Finding optimal parameter

- Analytical Solution:-

$$J(a_1) = \sum_i (y^{(i)2} - 2a_1 y^{(i)} x^{(i)} + a_1^2 x^{(i)2})$$

- Differentiate wrt a_1 and substitute as zero

$$\sum_i \left(-2y^{(i)} x^{(i)} + 2a_1 x^{(i)2}\right) = 0$$

$$a_1 = \frac{\sum_i y^{(i)} x^{(i)}}{\sum_i x^{(i)2}}$$

Analytical Solution

$$J = (Y \; - X\boldsymbol{a})^T(Y \; - X\boldsymbol{a})$$
$$= \; Y^TY \; - 2Y^TX\boldsymbol{a} + \boldsymbol{a}^TX^TX\boldsymbol{a}$$

$$\frac{\delta J}{\delta \boldsymbol{a}} = -2X^TY + 2\,X^TX\boldsymbol{a}$$

$$\boldsymbol{a} = (X^TX)^{-1}X^TY$$

Newton Update

- If we consider Taylor's approximation at a point a_0 we have:-

$$\boxed{J(a) = J(a_0) + J'(a_0)(\Delta_a) + \frac{1}{2}J''(a_0)(\Delta_a)^2}$$

- Diff wrt Δ_a and putting to zero we get:-

$$J'(a_0) + J''(a_0)\Delta_a = 0$$

$$\Delta_a = \frac{J'(a_0)}{J''(a_0)}$$

Newton Update

- If we consider Taylor’s approximation at a point **a**$_0$ we have:-

$$\boxed{\boldsymbol{a} = \boldsymbol{a}_0 + \boldsymbol{\Delta}_a}$$

$$\boxed{\begin{gathered} J(\boldsymbol{a}) = J(\boldsymbol{a}_0) + J'(\boldsymbol{a}_0)(\boldsymbol{\Delta}_a) \\ + \frac{1}{2} H(\boldsymbol{a}_0)(\boldsymbol{\Delta}_a)^2 \end{gathered}}$$

- Diff wrt Δ_a and putting to zero we get:-

$$\boxed{\begin{gathered} J'(\boldsymbol{a}_0) + H(\boldsymbol{a}_0)\boldsymbol{\Delta}_a = 0 \\ \boldsymbol{\Delta}_a = -H^{-1} J'(\boldsymbol{a}_0) \end{gathered}}$$

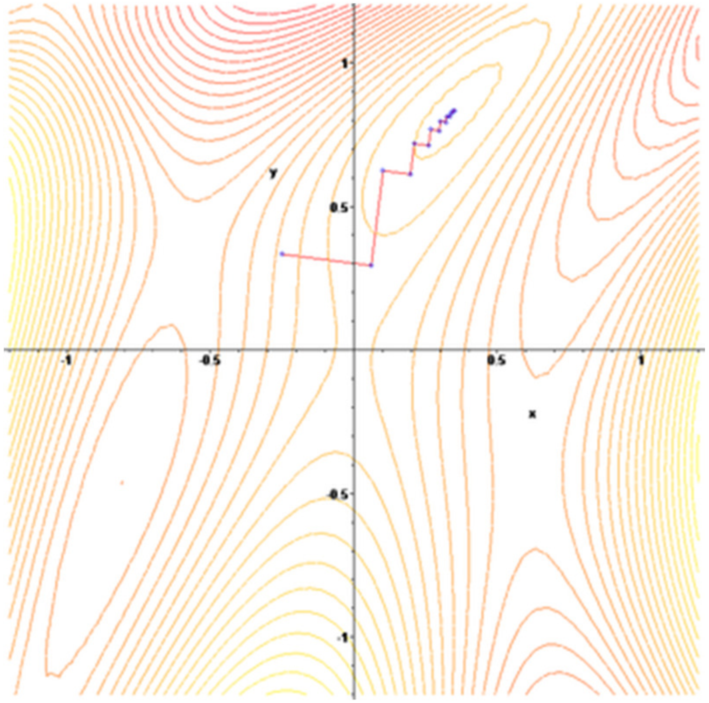
Gradient Descent

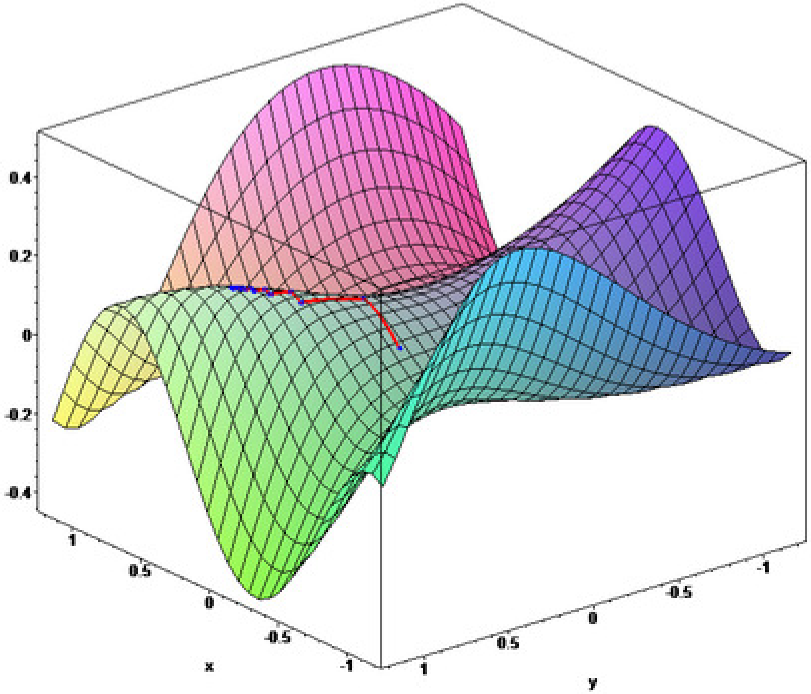
$$F(x,y) = \sin\left(\frac{1}{2}x^2 - \frac{1}{4}y^2 + 3\right)\cos(2x + 1 - e^y)$$

$$\boldsymbol{x} = \boldsymbol{x}_0 + \boldsymbol{\Delta}_x$$

$$\boldsymbol{\Delta}_x = -\gamma \nabla(x_0)$$

$$\boldsymbol{y} = \boldsymbol{y}_0 + \boldsymbol{\Delta}_y$$



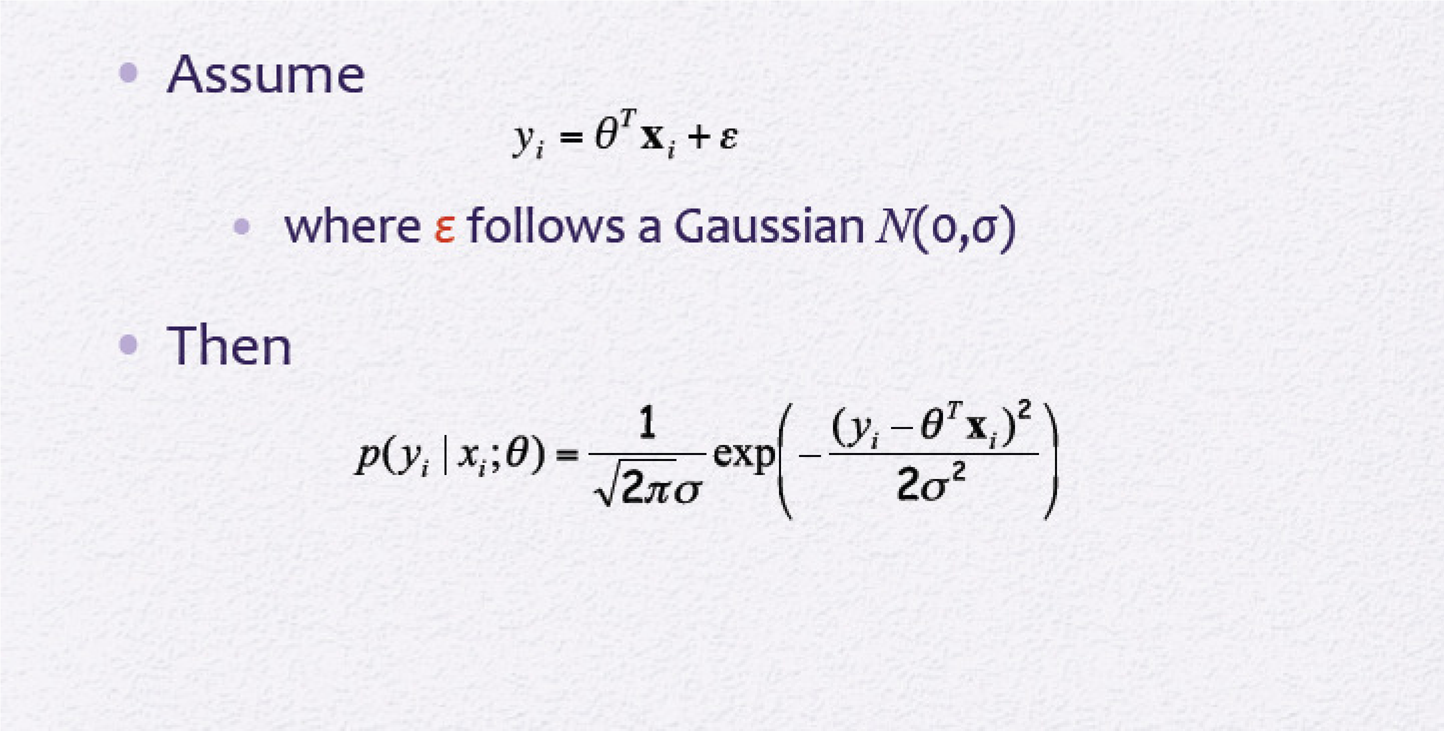


Gradient Descent

- Find the gradient $\nabla_{a^{\{t\}}}$
- Find an optimal step in the direction of the gradient α
  - Eg: Back-tracking, grid search etc.
- Iterate till the update is small enough

$$a^{\{t+1\}} = a^{\{t\}} - \alpha \nabla_{a^{\{t\}}}$$

Equivalence of LMS and MLE



- Assume

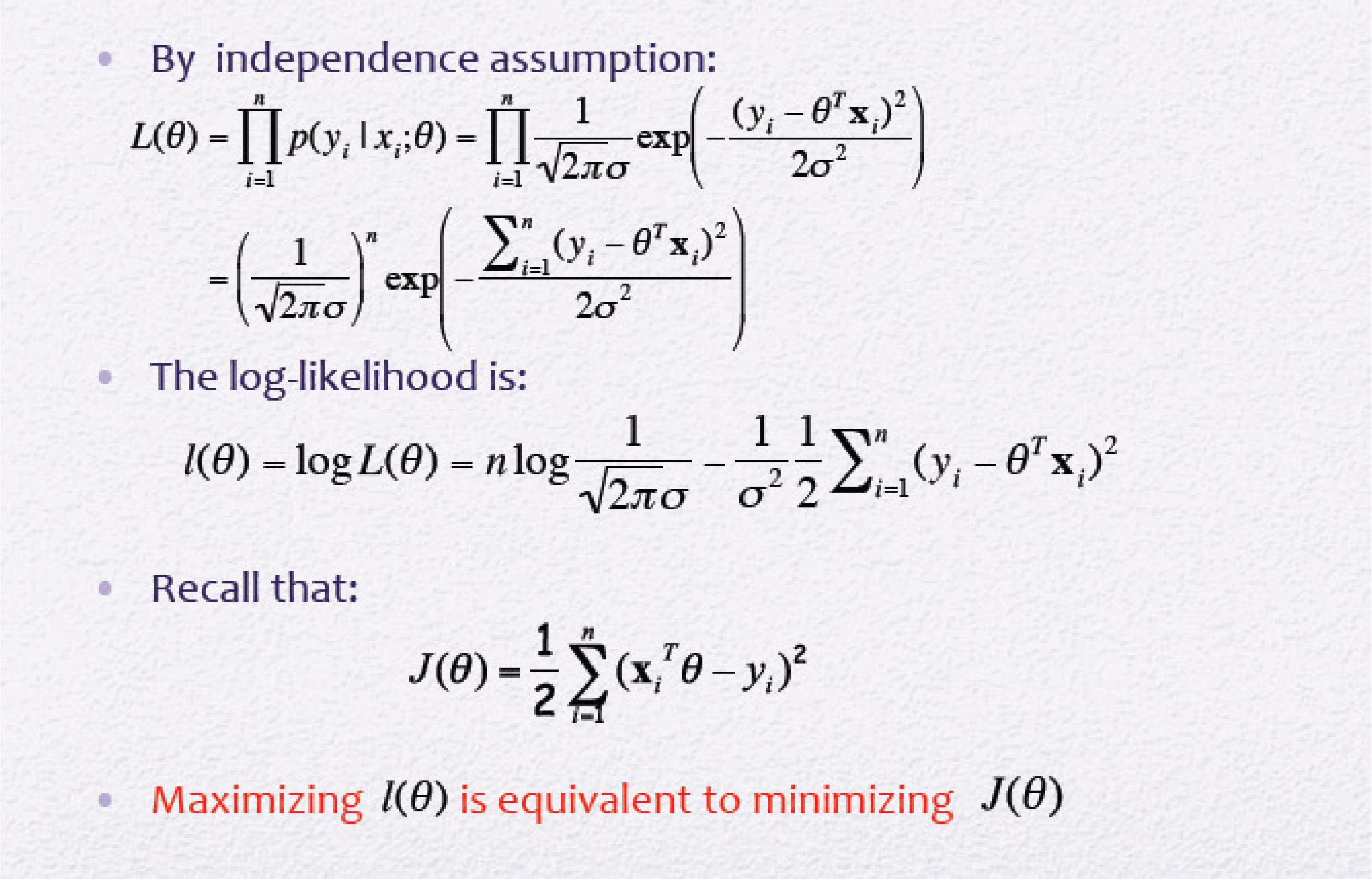
$$y_i = \theta^T \mathbf{x}_i + \varepsilon$$

  - where ε follows a Gaussian N(0,σ)

- Then

$$p(y_i \mid x_i;\theta) = \frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{(y_i - \theta^T \mathbf{x}_i)^2}{2\sigma^2}\right)$$

Equivalence of LMS and MLE



- By independence assumption:

$$L(\theta)=\prod_{i=1}^{n}p(y_i\,|\,x_i;\theta)=\prod_{i=1}^{n}\frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{(y_i-\theta^T\mathbf{x}_i)^2}{2\sigma^2}\right)$$

$$=\left(\frac{1}{\sqrt{2\pi}\sigma}\right)^n\exp\left(-\frac{\sum_{i=1}^{n}(y_i-\theta^T\mathbf{x}_i)^2}{2\sigma^2}\right)$$

- The log-likelihood is:

$$l(\theta)=\log L(\theta)=n\log\frac{1}{\sqrt{2\pi}\sigma}-\frac{1}{\sigma^2}\frac{1}{2}\sum_{i=1}^{n}(y_i-\theta^T\mathbf{x}_i)^2$$

- Recall that:

$$J(\theta)=\frac{1}{2}\sum_{i=1}^{n}(\mathbf{x}_i^T\theta-y_i)^2$$

- Maximizing $l(\theta)$ is equivalent to minimizing $J(\theta)$

Ridge & Lasso

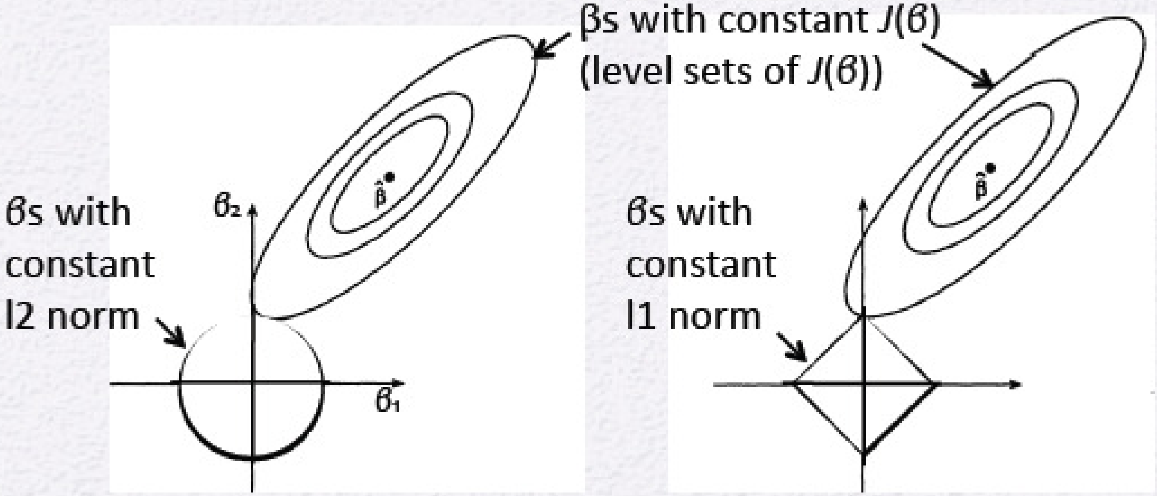
$$\min_{\beta}(\mathbf{X}\beta - \mathbf{Y})^T(\mathbf{X}\beta - \mathbf{Y}) + \lambda \mathrm{pen}(\beta) = \min_{\beta} J(\beta) + \lambda \mathrm{pen}(\beta)$$

Ridge Regression:

$$\mathrm{pen}(\beta) = \|\beta\|_2^2$$

Lasso:

$$\mathrm{pen}(\beta) = \|\beta\|_1$$



βs with constant $J(\beta)$ (level sets of $J(\beta)$)

βs with constant l2 norm

βs with constant l1 norm

Lasso (l1 penalty) results in sparse solutions – vector with more zero coordinates

Good for high-dimensional problems – don't have to store all coordinates!

What did we learn

- Vector Calculus
- A bunch of optimization schemes
  - Analytical, Newton update, Gradient descent
- Linear Regression
- Ridge & Lasso

Logistic Regression

- In Naïve Bayes, we learnt $P(X|Y)$ and $P(Y)$ in order to compute $P(Y|X)$

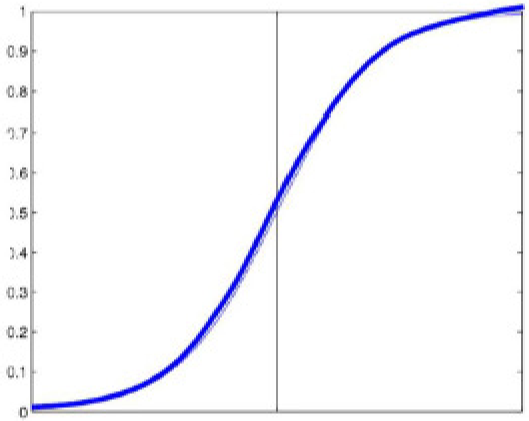
- Logistic regression learns $P(Y|X)$ directly for binary Y and real-valued X
  - LR is an example of a discriminative model
  - NB is a generative model

Logistic Regression

$$P(Y = 0|\mathbf{X}, \mathbf{w}) = \frac{1}{1 + exp(w_0 + \sum_i w_i X_i)}$$

- LR has a linear decision boundary
  - P(Y = 1|X,w) > 0.5 when $w_0 + \sum_i w_i X_i > 0$

- Logistic function $\frac{1}{1+exp(-z)}$ is sigmoid



Learning Parameter w

- Goal: Maximize conditional likelihood P(Y|X,w) w.r.t w

$$\hat{\mathbf{w}}_{MCLE} = \arg\max_{\mathbf{w}} \prod_{j=1}^{L} P(Y^{(j)} \mid X^{(j)}, \mathbf{w})$$

- Maximizing this is difficult, so we maximize log(P(Y|X,w)) instead:

$$\begin{aligned}\max_{\mathbf{w}} \; l(\mathbf{w}) \;&\equiv\; \ln \prod_{j}^{L} P(y^j|\mathbf{x}^j, \mathbf{w}) \\ &= \sum_{j}^{L} y^j(w_0 + \sum_{i}^{n} w_i x_i^j) - \ln(1 + exp(w_0 + \sum_{i}^{n} w_i x_i^j))\end{aligned}$$

Learning

$$\max_{\mathbf{w}} \; l(\mathbf{w}) \;\; \equiv \;\; \ln \prod_{j}^{L} P(y^j | \mathbf{x}^j, \mathbf{w})$$

$$= \;\; \sum_{j}^{L} y^j (w_0 + \sum_{i}^{n} w_i x_i^j) - \ln(1 + exp(w_0 + \sum_{i}^{n} w_i x_i^j))$$

- This function has no closed-form solution for its maximum

- But it is concave, so we can use gradient ascent to converge on the maximum

$$w_i^{(t+1)} \leftarrow w_i^{(t)} + \eta \, \frac{\partial l(\mathbf{w})}{\partial w_i^{(t)}}$$

Multi-Class

- What if Y takes on K > 2 values?

- One solution: K-class classification
  - For each class k < K:

$$P(Y = y_k|X) = \frac{\text{exp}(w_{k0} + \sum_{i=1}^{d} w_{ki}X_i)}{1 + \sum_{j=1}^{K-1} \text{exp}(w_{j0} + \sum_{i=1}^{d} w_{ji}X_i)}$$

  - For class K

$$P(Y = y_K|X) = \frac{1}{1 + \sum_{j=1}^{K-1} \text{exp}(w_{j0} + \sum_{i=1}^{d} w_{ji}X_i)}$$