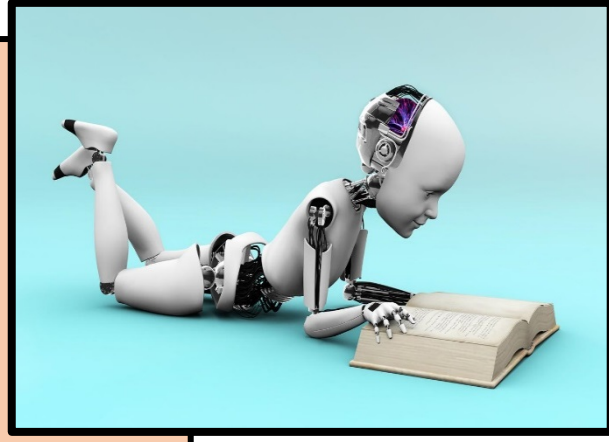
Machine learning from data
Class1: Linear Regression



Zohar Yakhini

IDC

IDC HERZLIYA

Example: House Pricing

- We want to know the price of a house as a function of its size (in sqft).
- We want to learn a function from the size of the house x to the price y=f(x) so we would be able to answer the above question
- Training set: 10 house "instances" with feature values and labels

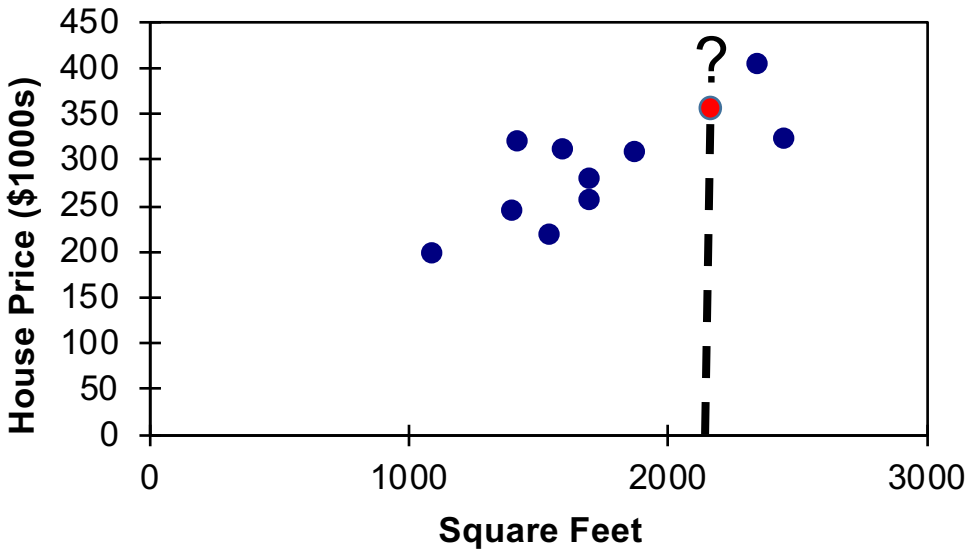
Square Feet (x)	House Price in $1000s (y)
1400	245
1600	312
1700	279
1875	308
1100	199
1550	219
2350	405
2450	324
1425	319
1700	255

IDC HERZLIYA

Statistics:
Dependent variable (y) = house price
Explaining variable (x) = square footage

Graphical Representation

Scatter plot of House Price (y) vs House Size (x)

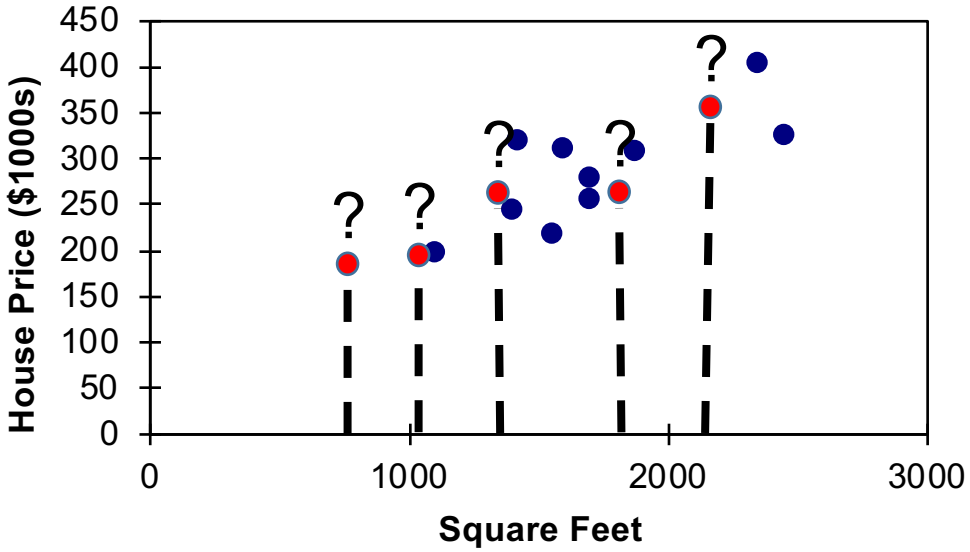


Prediction:

Given house of size x, what would be its price y = f(x)?

IDC HERZLIYA

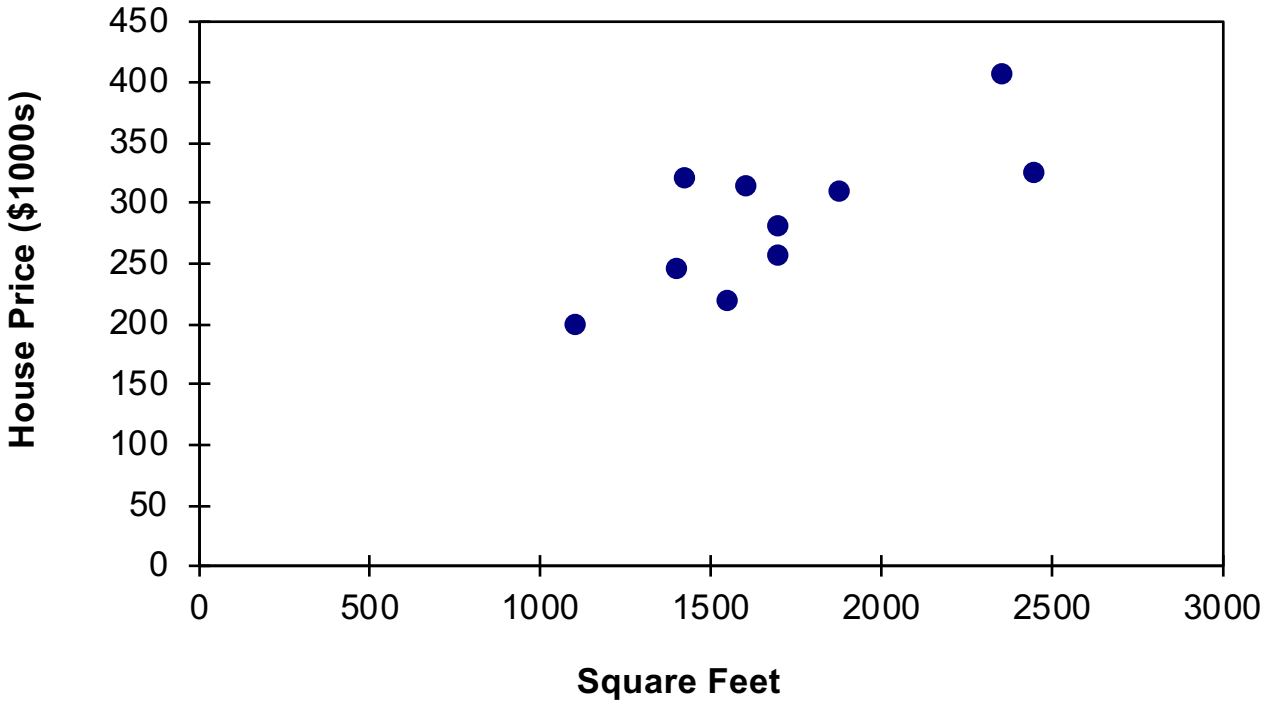
Memorization?



- Store all sizes?
- Our data doesn't cover all sizes … What shall we do w a house of size 1750 sqft?

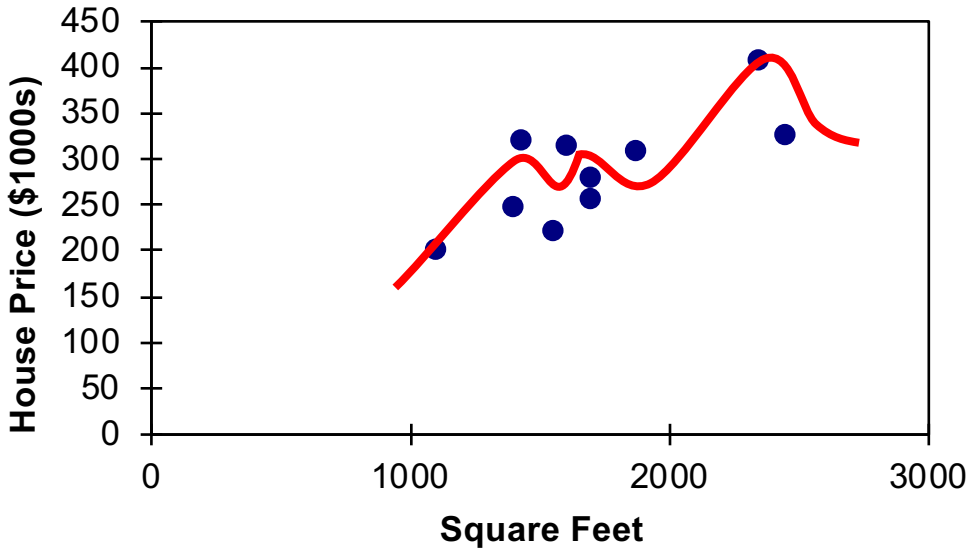
IDC HERZLIYA

OK … a function?



IDC HERZLIYA

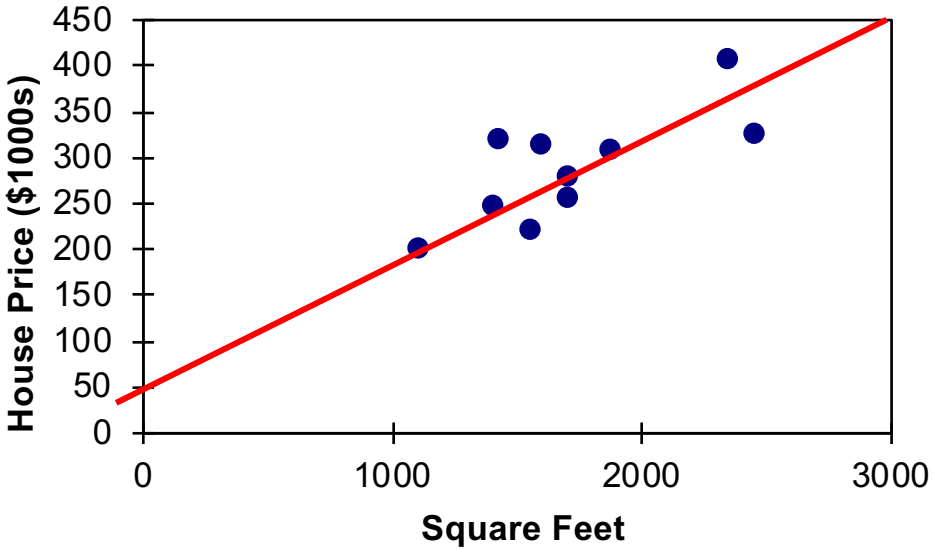
Generalization: Learn a Function



But which function y=f(x)?

IDC HERZLIYA

Simplest: Linear Model



- We assume/hypothesize that the relationship between the observed and the independent/explained variable is linear and thereby conduct our search
- This is our **Hypotheses Space –** all linear functions

IDC HERZLIYA

Linear Function Hypothesis

- How do we represent a hypothesis h in this space?

$$y = \theta_0 + \theta_1 x$$

- What if we have many features and not just house size?
- Such as:
  - Number of rooms
  - Distance to shopping center
  - Neighborhood crime rate
  - Distance to IDC
  - More…

IDC HERZLIYA

Multiple Features

- Let $\mathbf{x^{(i)}} = \left(\mathrm{x}_1^{(i)}, \mathrm{x}_2^{(i)}, \ldots, \mathrm{x}_n^{(i)}\right)$

  be the vector of feature values for each instance i
- For simplicity we add another "constant" feature for every i

$$\mathrm{x}_0^{(i)} = 1$$

- For n features our linear hypothesis will be represented by the parameters $\boldsymbol{\theta} = \left(\theta_0, \theta_1, \theta_2, \ldots, \theta_n\right)$

  for which we have:

$$\mathrm{y} = \theta_0 + \theta_1 \mathrm{x}_1 + \theta_2 \mathrm{x}_2 + \cdots + \theta_\mathrm{n} \mathrm{x}_\mathrm{n}$$

- We say that $\boldsymbol{\theta} = \left(\theta_0, \theta_1, \theta_2, \ldots, \theta_n\right)$

  is the vector of parameters that defines our function (or our **model**)

The Hypothesis (or model) is the execution algorithm

- How does a specific set of values,

  $\boldsymbol{\theta} = (\theta_0, \theta_1, \theta_2, \ldots, \theta_n)$,

  which defines a specific hypothesis (model) help us?
- How can we use it?
- Given a new house instance $\mathbf{x}' = (\mathrm{x}'_0, \mathrm{x}'_1, \ldots, \mathrm{x}'_n)$ we can estimate its price by an inner product with the vector $\boldsymbol{\theta}$:

$$\text{value of house} = \mathrm{y} = \theta_0 + \theta_1 \mathrm{x}'_1 + \theta_2 \mathrm{x}'_2 + \cdots + \theta_n \mathrm{x}'_n$$

IDC HERZLIYA

Finding the Best Hypothesis/Model

- There are many possible parameter vectors

$$\boldsymbol{\theta} = (\theta_0, \theta_1, \theta_2, \ldots, \theta_n)$$

and each one defines a different hypothesis in our hypotheses space

- How do we find the best one?
- What can we use to help us find it?

- The training set: m instances where, for each, we know the feature values $\mathbf{x^{(i)}} = \left(\mathrm{x}_0^{(i)}, \mathrm{x}_1^{(i)}, \ldots, \mathrm{x}_n^{(i)}\right)$

as well as the value $\mathrm{y}^{(i)}$

IDC HERZLIYA

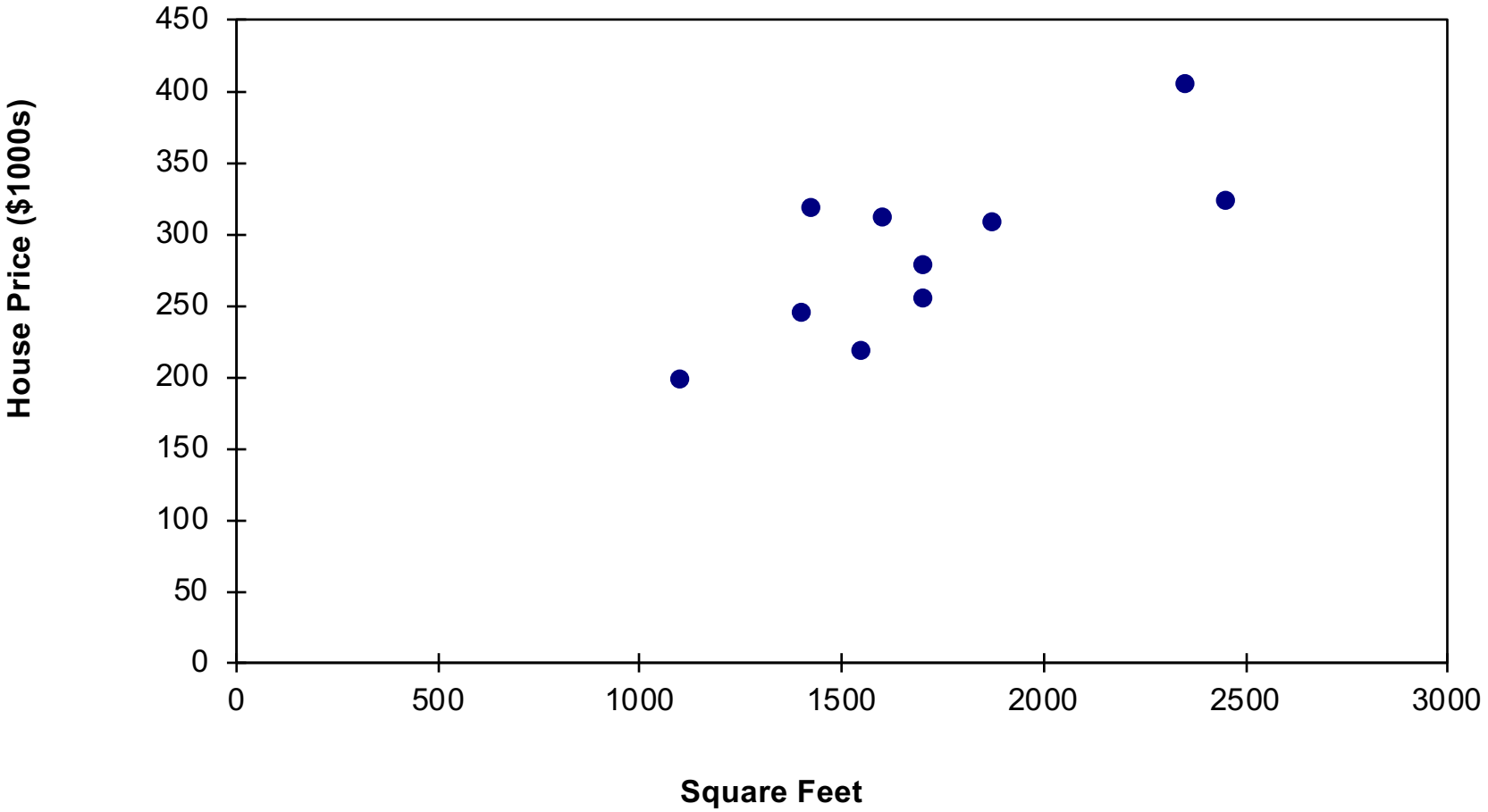
"Training" or "Learning"

- We require that on our training data the values of our prediction function (f) would be similar to the known value of the house.
- So, we want to find $\boldsymbol{\theta}$ such that <u>for all</u> instances i in the training set we will have:

$$y^{(i)} = \theta_0 x_0^{(i)} + \theta_1 x_1^{(i)} + \theta_2 x_2^{(i)} + \cdots + \theta_n x_n^{(i)} = \boldsymbol{\theta} \cdot \mathbf{x}^{(i)}$$

- However, this may not always be possible – (why?)

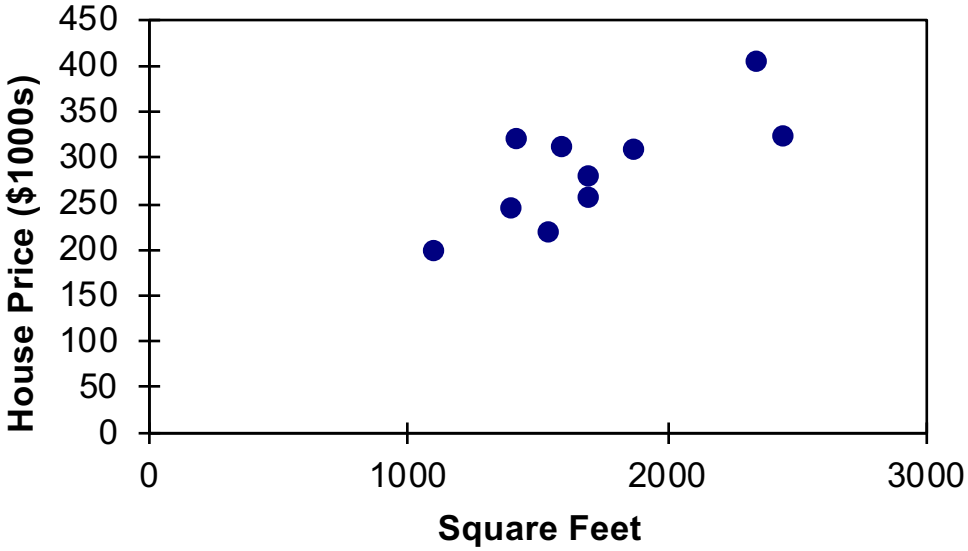
IDC HERZLIYA



IDC HERZLIYA

Consistent Learners

- A learning algorithm that can achieve 0 error on the training set is called a "consistent learner"
- This can not be done here.



IDC HERZLIYA

Cost Function of a Model θ

- Consistent learning may be impossible.
- Still, we can try to reduce the error.
  Per instance the error is:

Prediction *Actual value*

$$(\theta_0 x_0^{(i)} + \theta_1 x_1^{(i)} + \theta_2 x_2^{(i)} + \cdots + \theta_n x_n^{(i)} - y^{(i)}) = \boldsymbol{\theta} \cdot \mathbf{x}^{(i)} - y^{(i)}$$

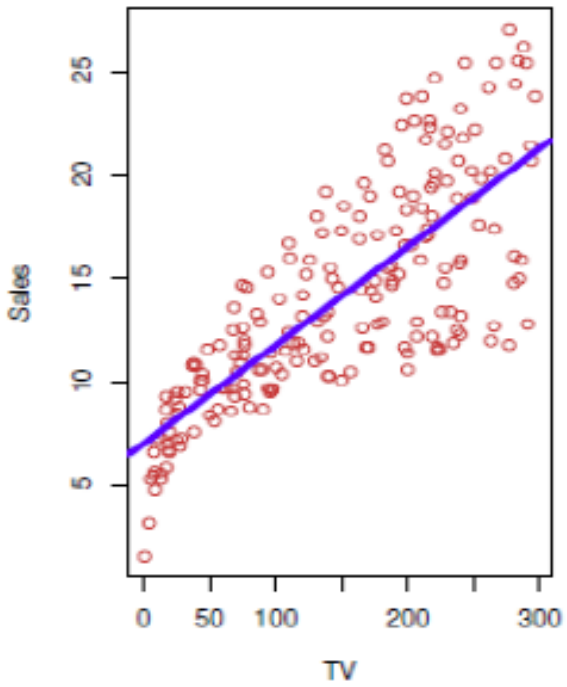
- And we now average on ALL m training instances to get our
  cost function:

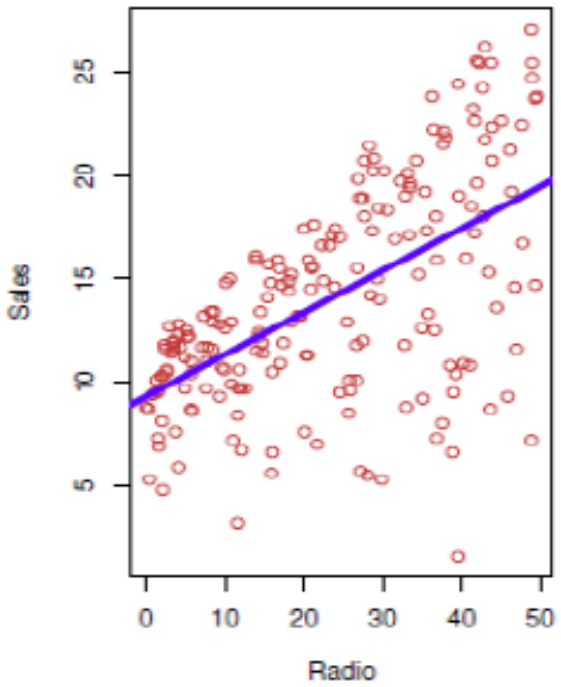
$$J(\boldsymbol{\theta}) = \frac{1}{2}\frac{1}{m}\sum_{i=1}^{m}\left(\boldsymbol{\theta} \cdot \mathbf{x}^{(i)} - y^{(i)}\right)^2$$

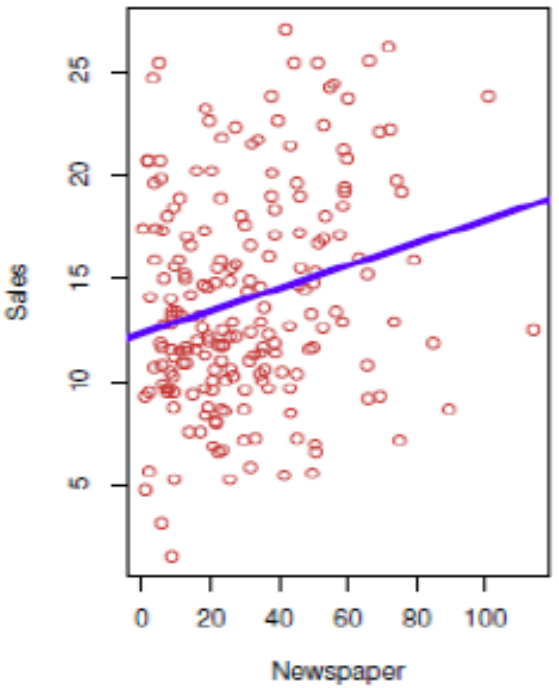
- Square errors are used so that errors in different directions don't cancel out …
  Its also a smoother function than $|x|$

IDC HERZLIYA

Example: advertising and sales

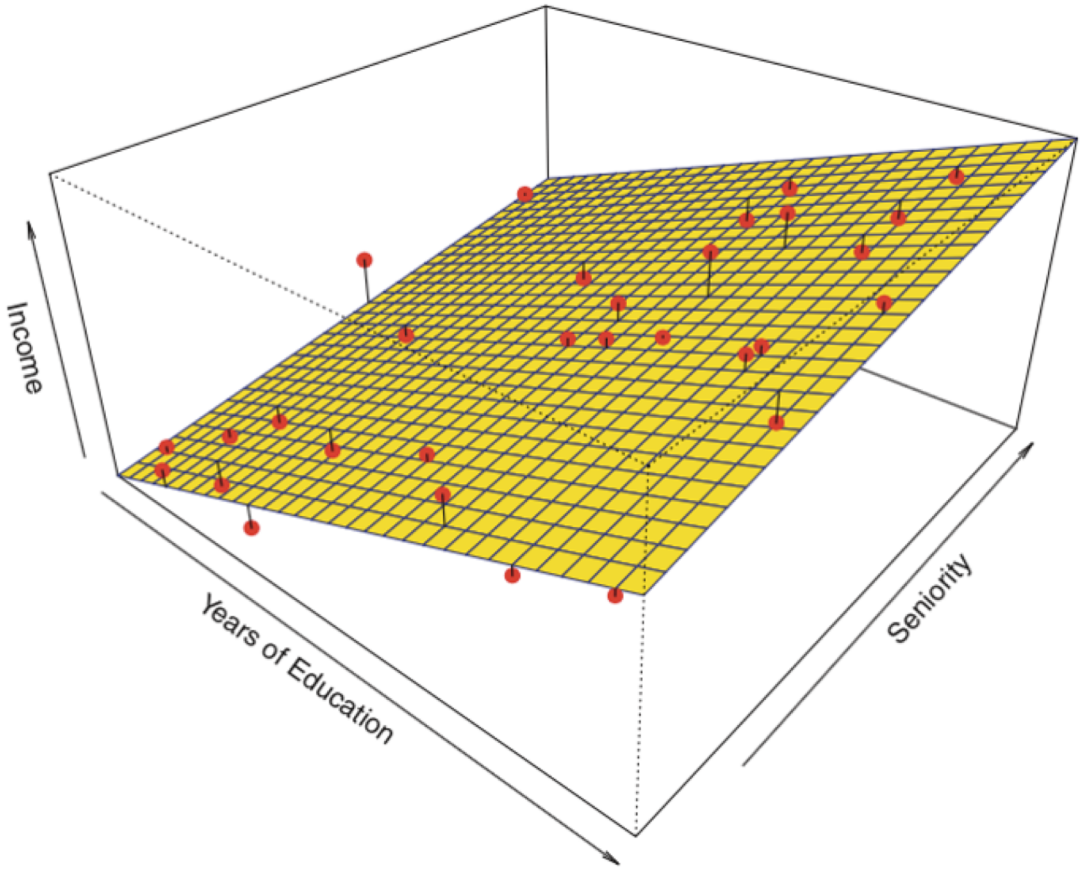






IDC HERZLIYA

In higher dimensions



IDC HERZLIYA

Minimizing the Cost Function

- Hence, our best hypothesis $\boldsymbol{\theta}^*$ would be the one that minimizes the cost function:

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}}\left[J(\boldsymbol{\theta})\right] = \arg\min_{\boldsymbol{\theta}}\left[\frac{1}{2m}\sum_{i=1}^{m}\left(\boldsymbol{\theta}\cdot\mathbf{x}^{(i)} - \mathrm{y}^{(i)}\right)^2\right]$$

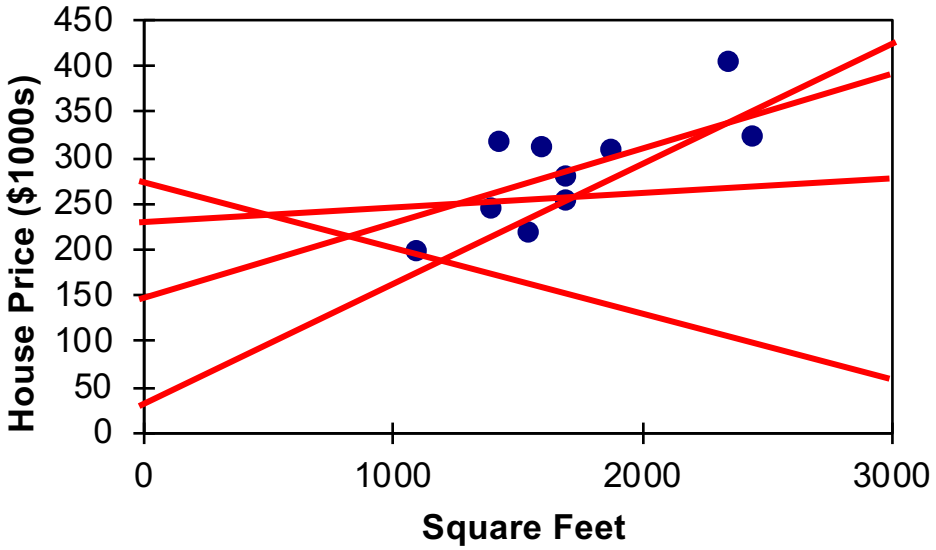
- How can we find it?

IDC HERZLIYA

The Hypotheses Space

- For the simple case of a single feature we get different possible straight lines when we change θ_0 and θ_1 in the hypothesis (model)

$$y = \theta_0 + \theta_1 x$$

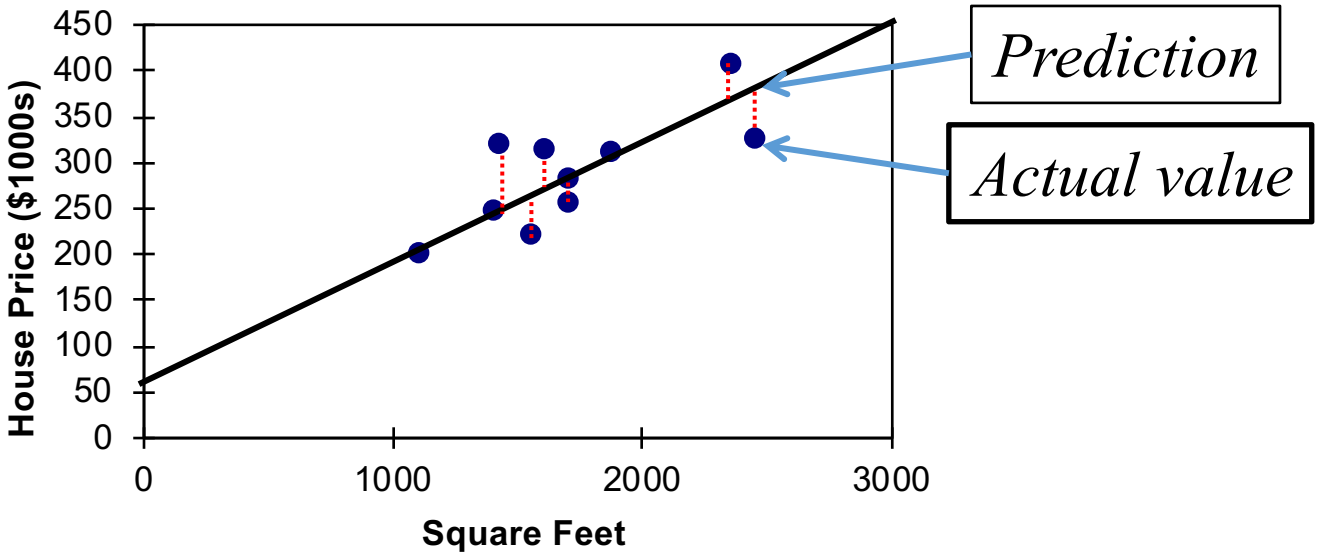


IDC HERZLIYA

Geometric interpretation of the error

We search for θ_0 and θ_1 that minimize the sum of (squared) errors:

$$J(\boldsymbol{\theta}) = \frac{1}{2m}\sum_{i=1}^{m}\left(\theta_0 + \theta_1 x^{(i)} - y^{(i)}\right)^2$$

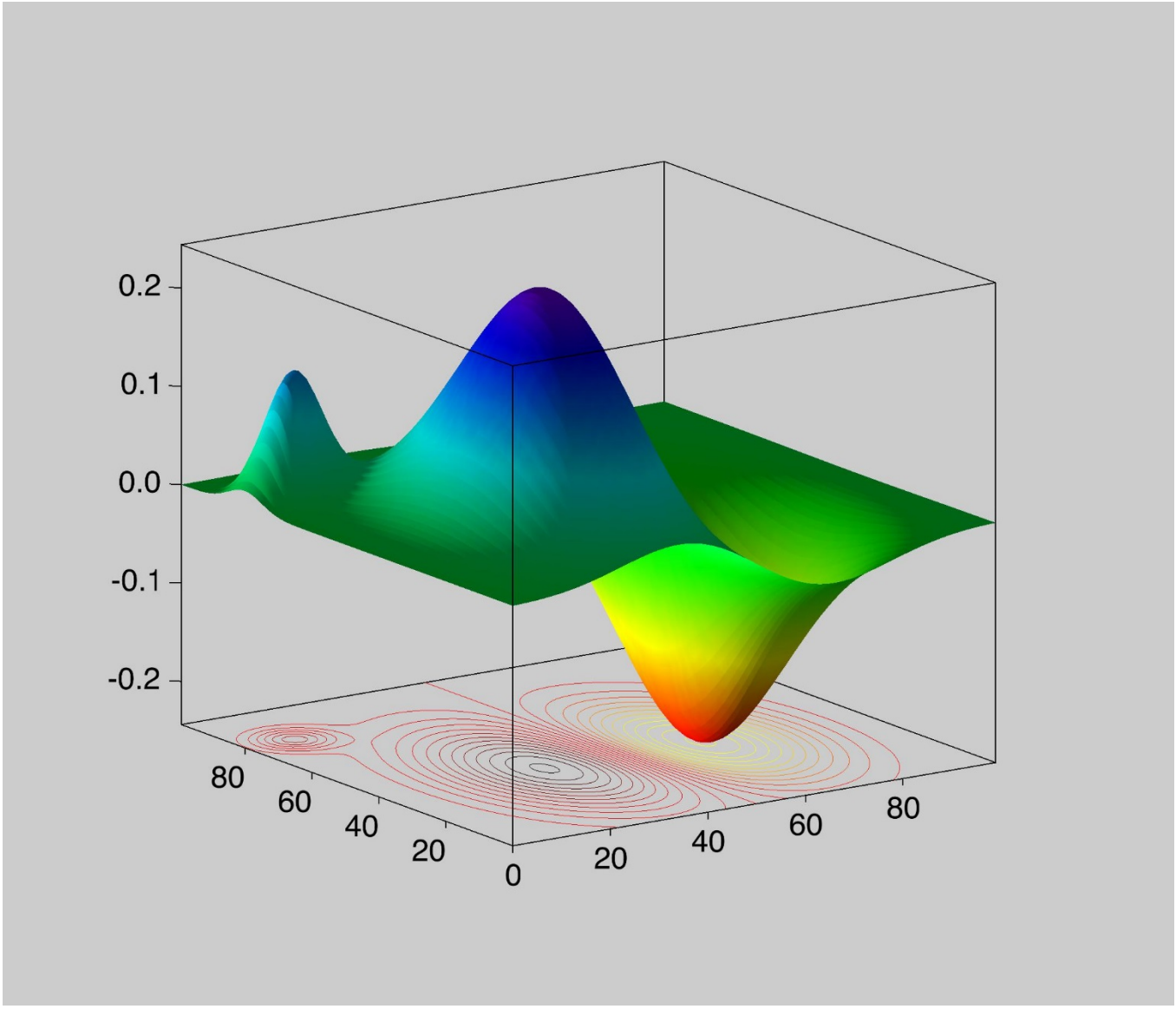


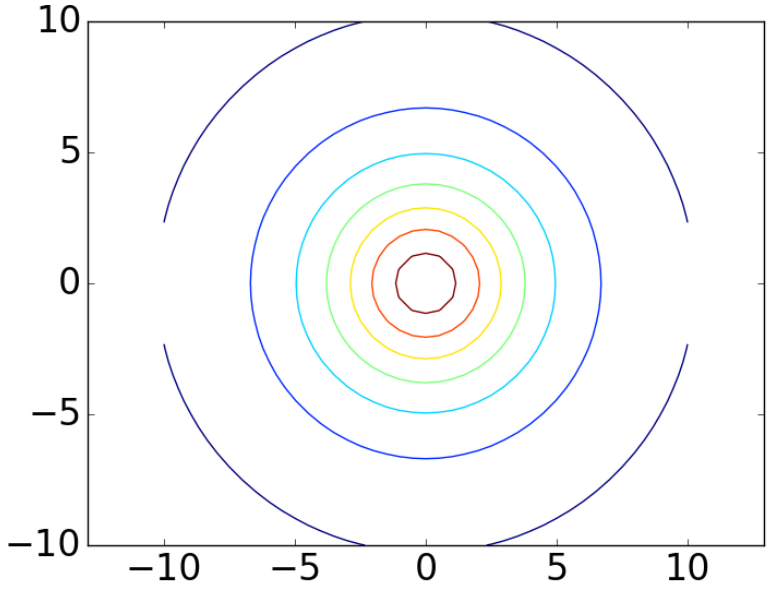
The Cost Function

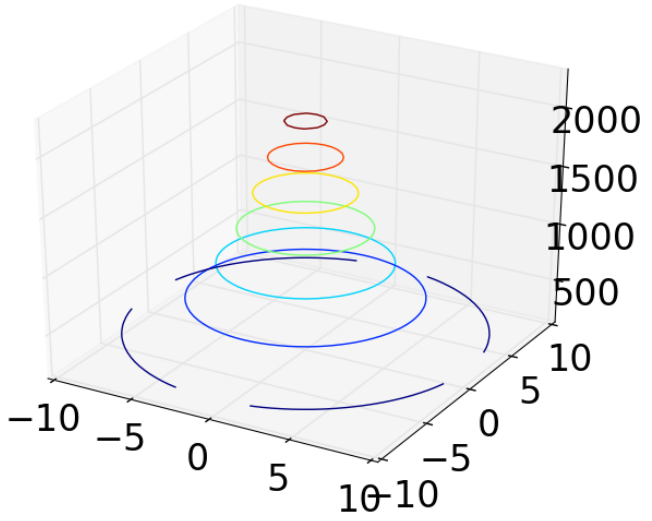
θ_0, θ_1

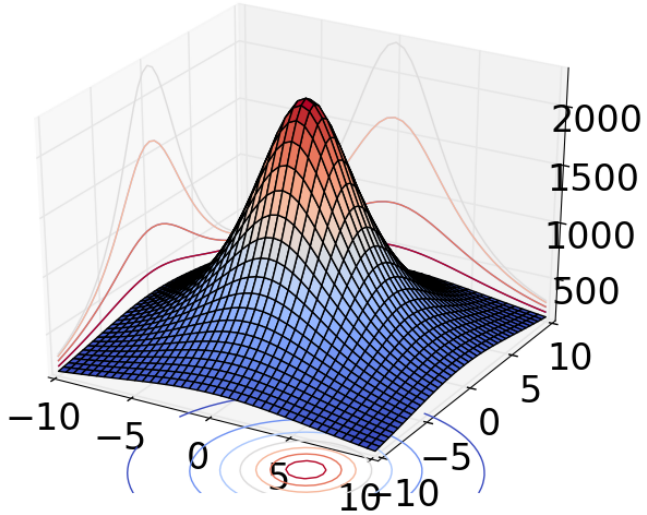
- In our simple case we have 2 parameters:
- For each value of these two parameters we can calculate the cost $J(\theta_0, \theta_1)$ (the training error).
- This is a function from $\mathbb{R}^2$ to $\mathbb{R}$.

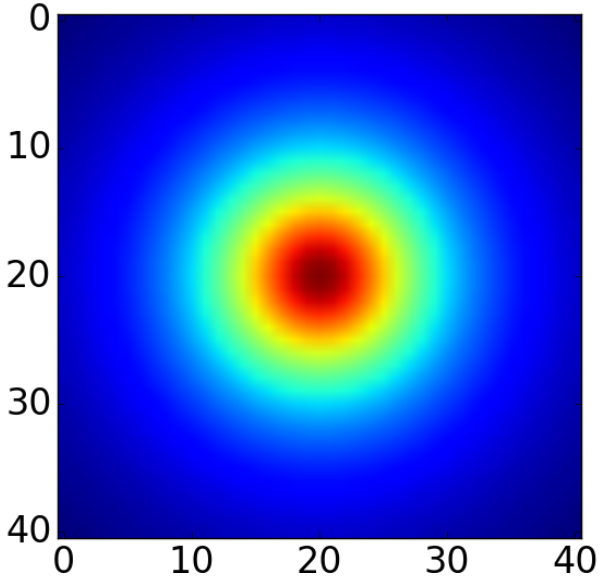
IDC HERZLIYA


0.2
0.1
0.0
-0.1
-0.2
80
60
40
20
0
20
40
60
80


10
5
0
−5
−10
−10
−5
0
5
10


2000
1500
1000
500
−10
−5
0
5
10
−10
−5
0
5
10

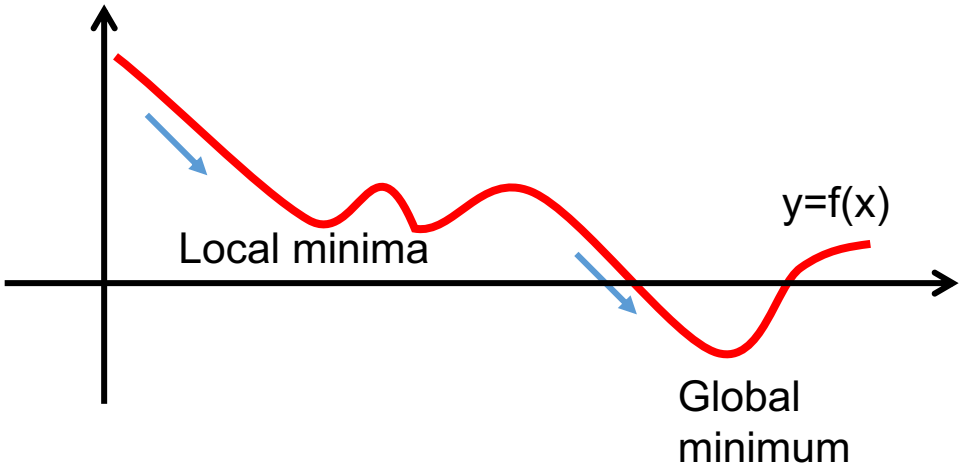

2000
1500
1000
500
−10
−5
0
5
10
−10
−5
0
5
10


0
10
20
30
40
0
10
20
30
40

How to Find the Minimum of a 1D function?

- In high school: derivative at the point where an extremum is attained should be 0
- We can also follow the “downward” direction.
- How is this done?
  1. Find the derivative
  2. Move in the negative direction

- Can be trapped in local minimum!



IDC HERZLIYA

Directional Derivatives

- For a differentiable 2D function $f(x_1, x_2)$ the partial derivatives in the directions x_i are

$$\frac{\partial}{\partial x_1} f, \frac{\partial}{\partial x_2} f$$

- For example …

$$f(x, y) = 2x + 13y$$
$$f(x, y) = y \exp(x)$$
$$f(x, y) = 2x + 3xy + 5y^2$$

https://mathinsight.org/directional_derivative_gradient_introduction

IDC HERZLIYA

$$f(x,y) = y\, e^{x}$$

$$\frac{\partial f}{\partial x}(x,y) = ye^{x}$$

$$\frac{\partial f}{\partial y}(x,y) = e^{x}$$

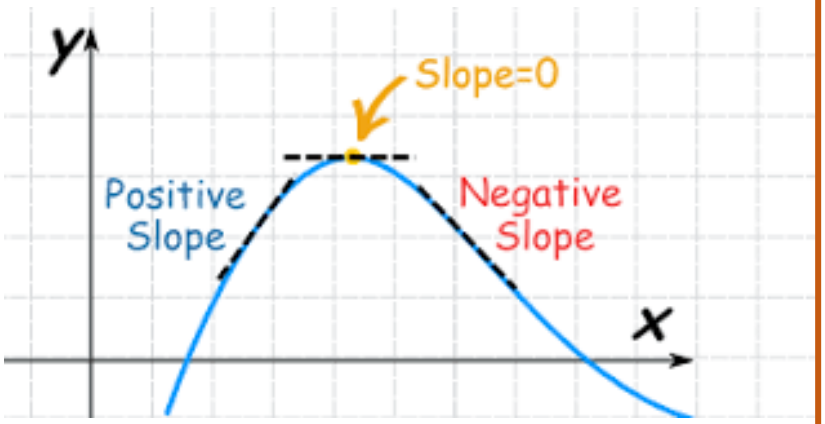
IDC HERZLIYA

$$f(x,y) = y^4 e^x$$

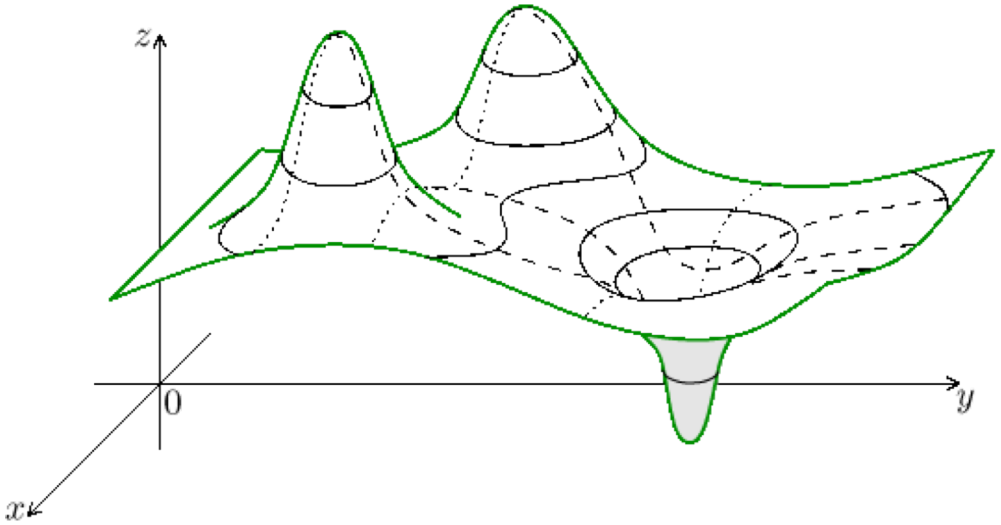
$$\frac{\partial f}{\partial x}(x,y) = y^4 e^x$$

$$\frac{\partial f}{\partial y}(x,y) = 4y^3 e^x$$

Extrema and zero derivatives



1D

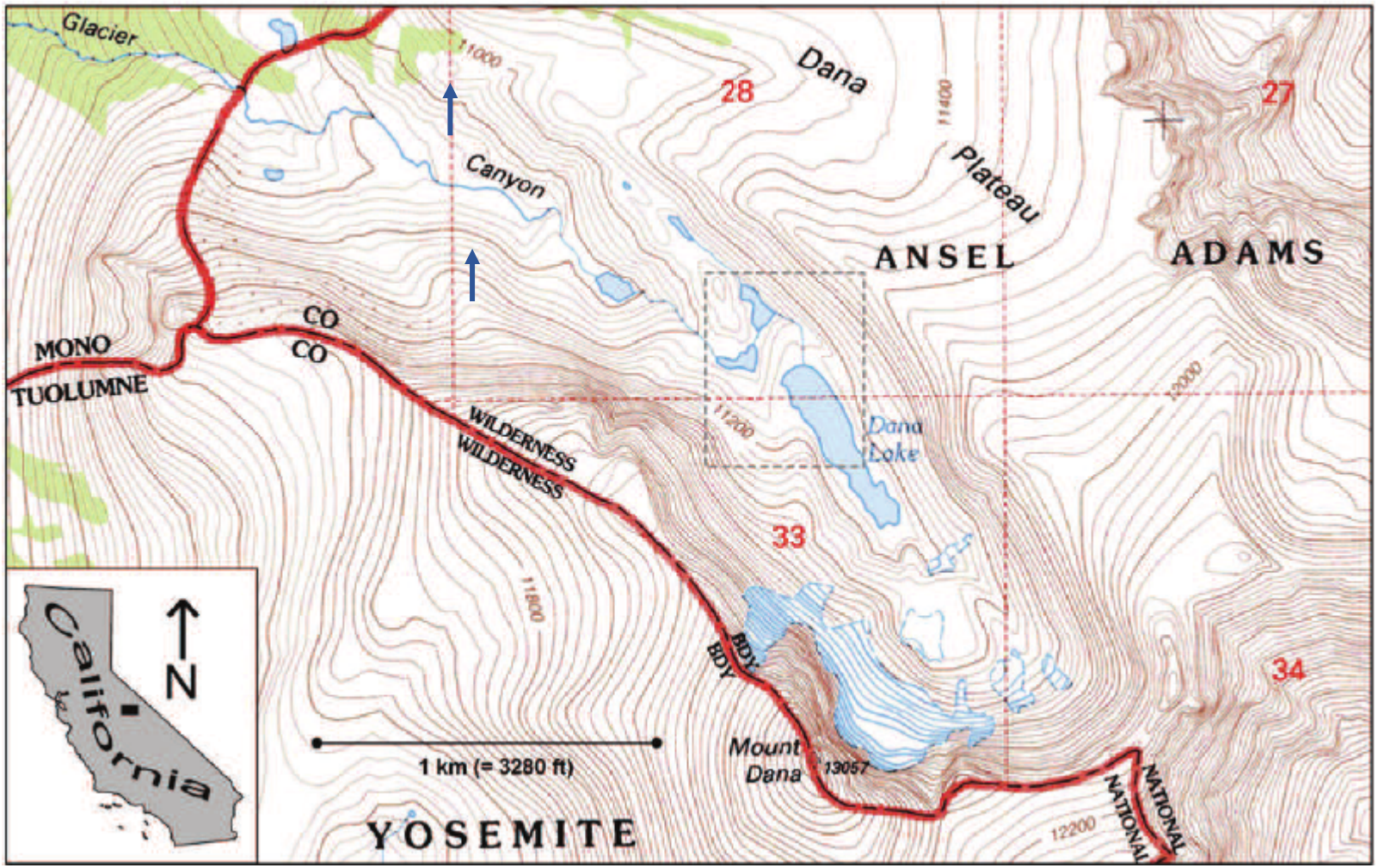


2D

IDC HERZLIYA

Example

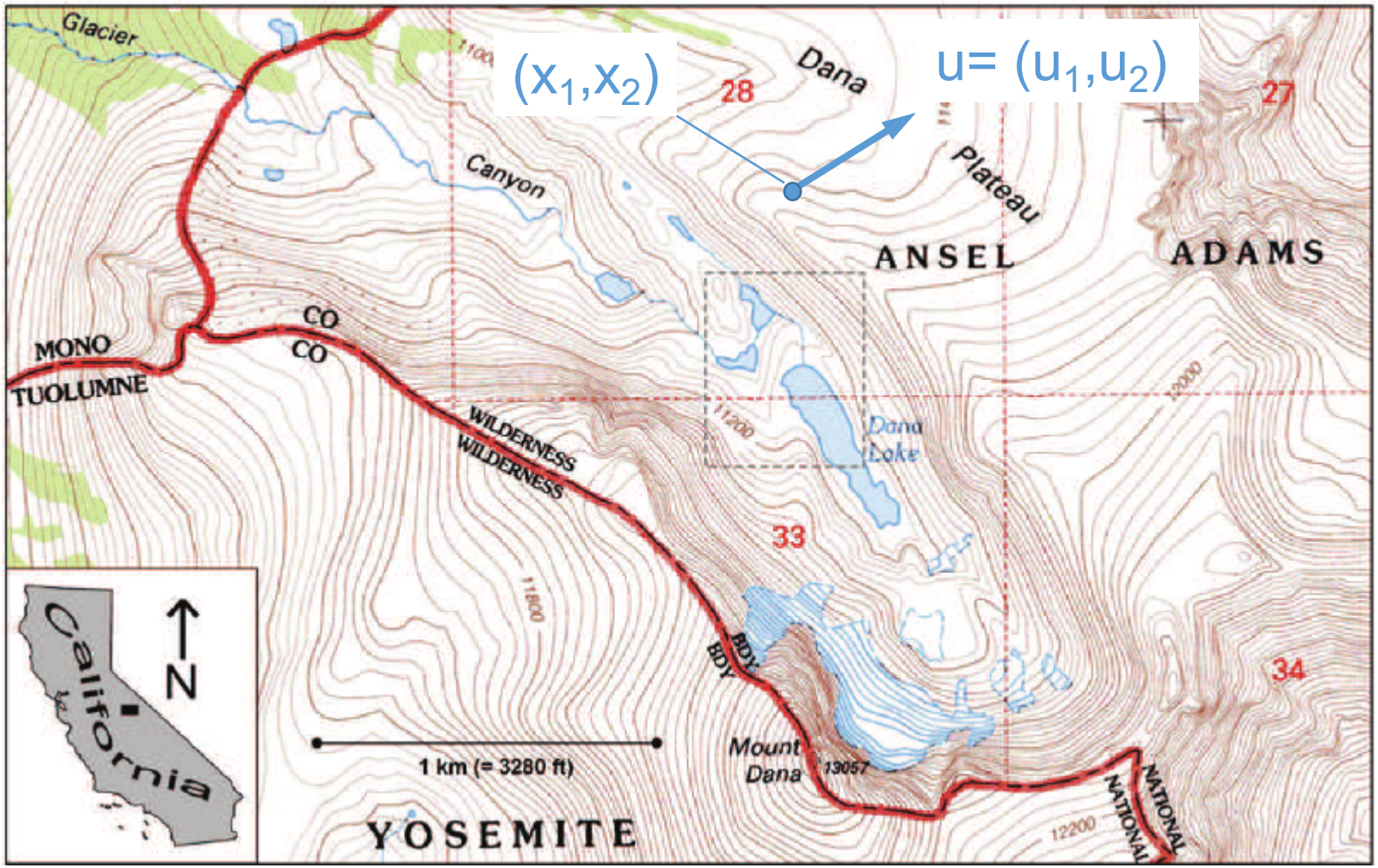
What is the sign of the y directional derivative?



IDC HERZLIYA

Example

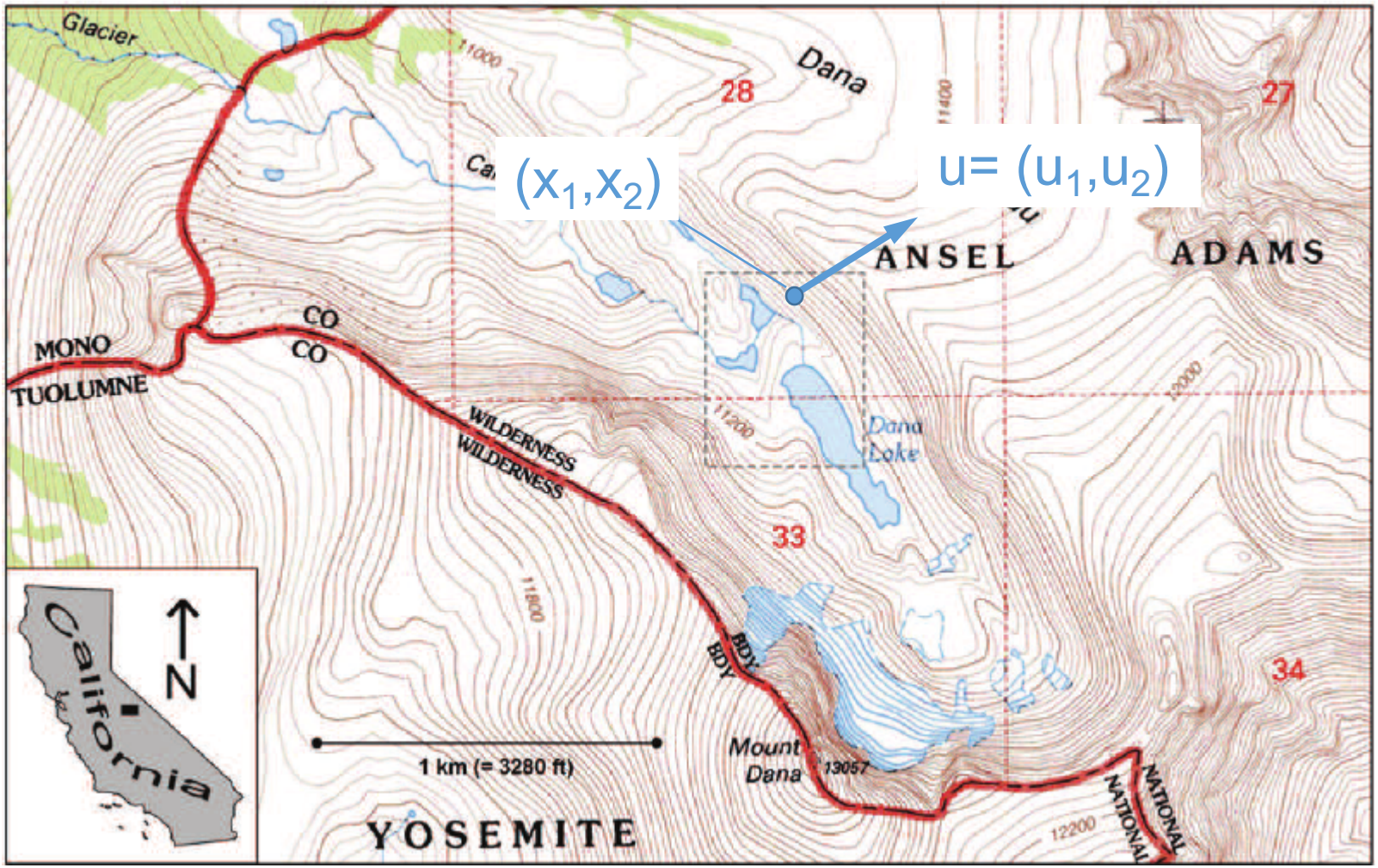
Up or down?



IDC HERZLIYA

Example

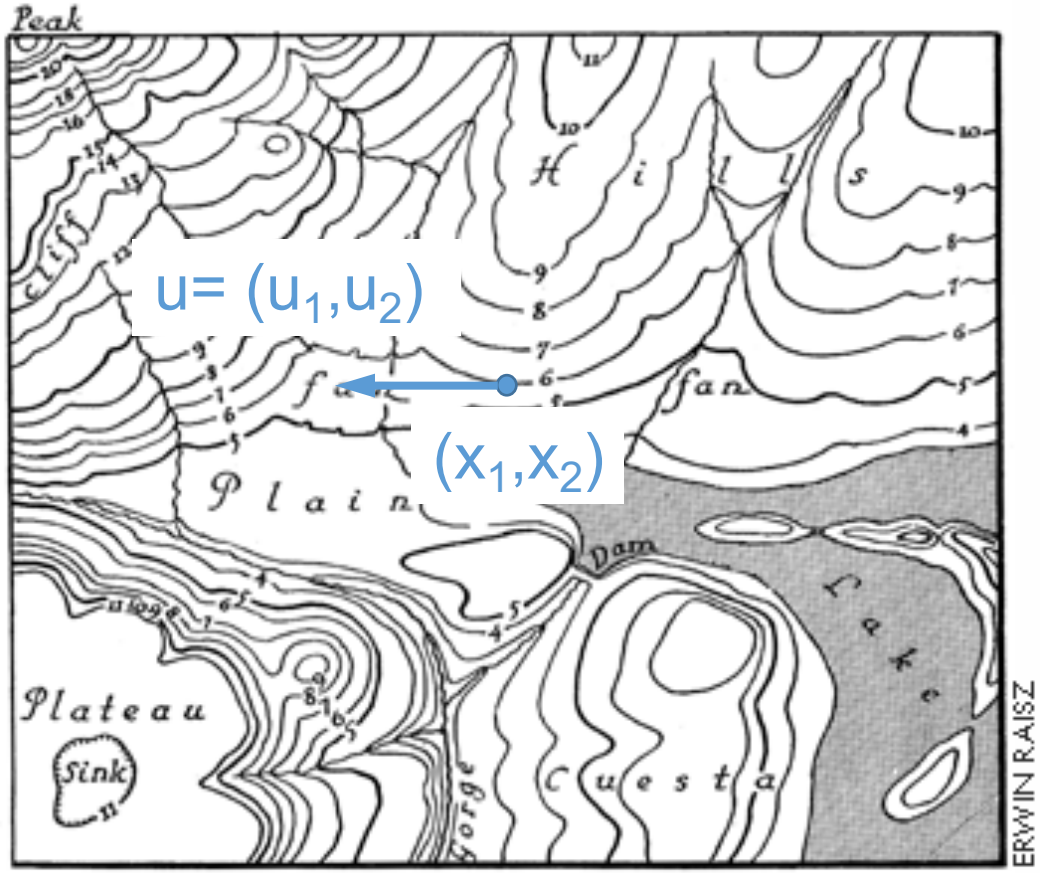
Up or down?



IDC HERZLIYA

Another …

Up or down?



IDC HERZLIYA

Directional Derivatives

- For a differentiable 2D function $f(x_1, x_2)$ the partial derivatives in direction x_i are $\frac{\partial}{\partial x_1} f, \frac{\partial}{\partial x_2} f$
- The derivative in a <u>general direction</u> $u = (u_1, u_2)$ (unit 2D vector) is called the directional derivative $D_u f$ and is defined as:

$$D_u f(x_1, x_2) = \lim_{s \to 0} \frac{f(x_1 + su_1, x_2 + su_2) - f(x_1, x_2)}{s} = \left(\frac{df}{ds} \right)_u$$

IDC HERZLIYA

The Gradient of a function

$$D_u f(x_1, x_2) = \left(\frac{df}{ds}\right)_u = \frac{\partial f}{\partial x_1}\frac{\partial x_1}{\partial s} + \frac{\partial f}{\partial x_2}\frac{\partial x_2}{\partial s} =$$

$$= \frac{\partial f}{\partial x_1}u_1 + \frac{\partial f}{\partial x_2}u_2 = \left(\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}\right) \cdot (u_1, u_2) = \nabla f(x_1, x_2) \cdot \mathbf{u}$$

This is the dot product of the direction u with a vector that is called the **GRADIENT** and consists of the principle directional derivatives evaluated at (x_1, x_2) :

$$\nabla f = \left(\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}\right)$$

IDC HERZLIYA

The Gradient of a function

Define the
GRADIENT of f:

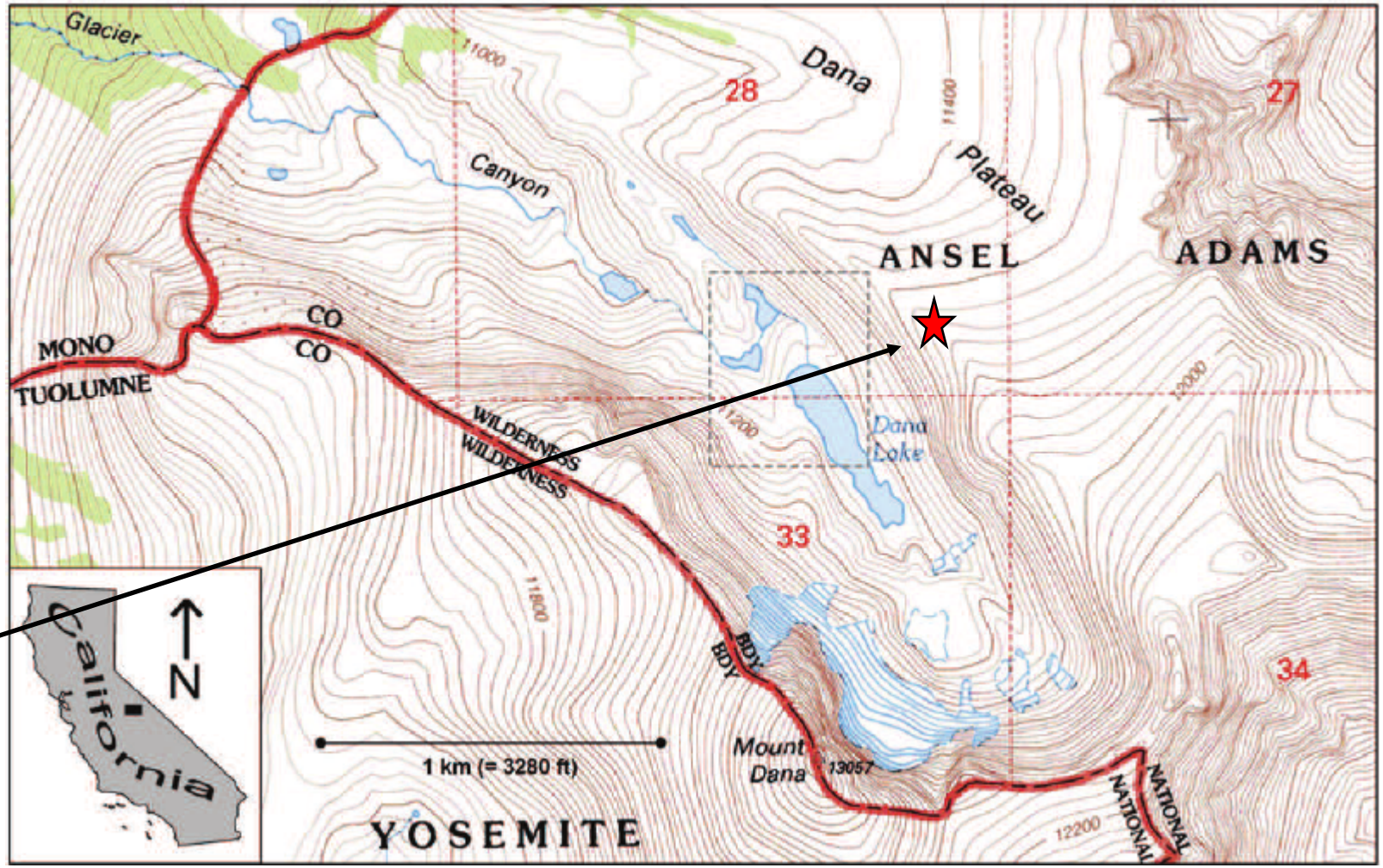
$$\nabla f = \left(\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2} \right)$$

Thm:
For any direction $u = (u_1, u_2)$ and any point $x = (x_1, x_2)$ we have:

$$D_u f(x) = \nabla f(x) \cdot u$$

IDC HERZLIYA

Most Rapid Increase at a point x



The pt x

IDC HERZLIYA

Most Rapid Increase at a point x

- The directional derivative in the direction of the vector u = (u_1,u_2) (a scalar!) can also be written as:

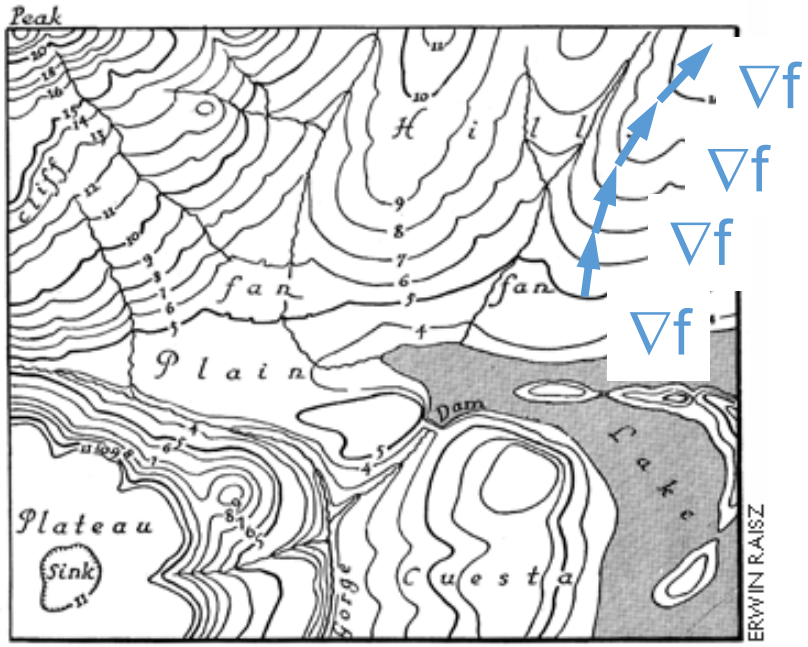
$$D_u f(x_1, x_2) = \nabla f \cdot \vec{\mathbf{u}} = \left|\nabla f\right|\left|\vec{\mathbf{u}}\right| \cos\beta = \left|\nabla f\right| \cos\beta$$

- Where β is the angle between $\boldsymbol{u}$ and ∇f
- However, $\cos\beta \leq 1$.
- Therefore:
  1. The greatest increase in the function happens in the direction of the gradient (i.e. β =0)
  2. The greatest decrease is in the direction $-\nabla f$ (i.e β =180^o)

IDC HERZLIYA

Steepest ascent and Iso-Contours

- Steepest ascent follows the gradient direction
- Using same argument, when the direction u is perpendicular to the gradient, cosβ =0 and there is no change in the function – this is exactly the direction of the iso-contours, walking with no change in altitude.



IDC HERZLIYA

The gradient of n-dimensional functions

- For an n-D function $y = f(x_1, x_2, \ldots, x_n)$ the gradient is defined as :

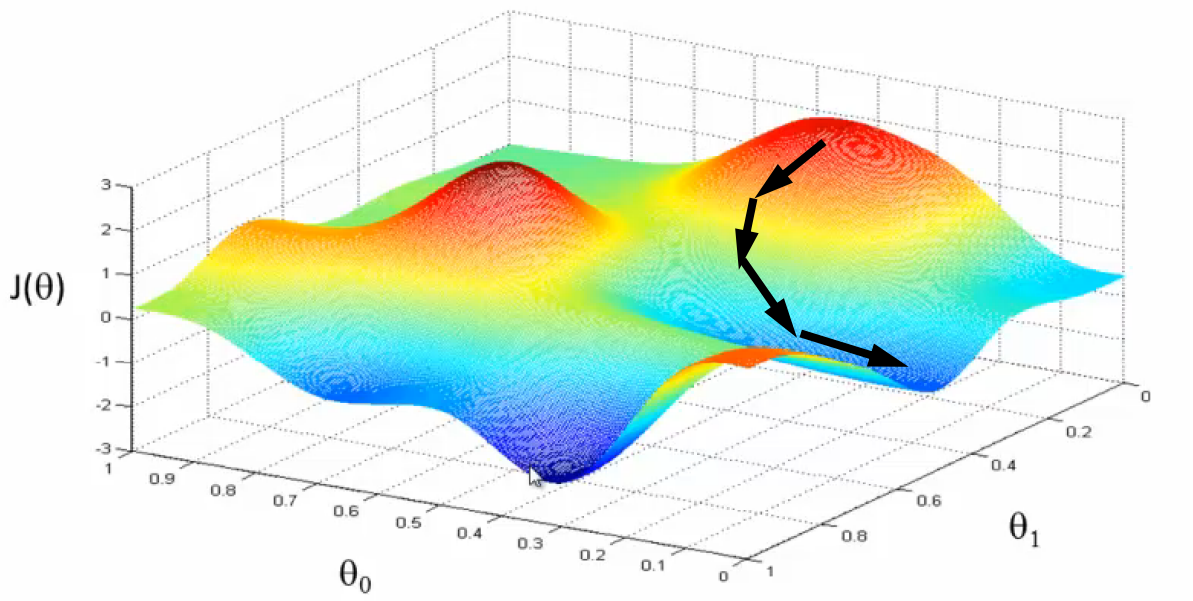
$$\nabla f = \left(\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \ldots, \frac{\partial f}{\partial x_n} \right)$$

- The gradient vector at any given point is the direction of greatest increase of f at this point.
- The direction of greatest decrease of f is the negative of the gradient: $-\nabla f(x_1, x_2 \ \ldots \ x_n)$

IDC HERZLIYA

How to find a minimum of a (reasonably smooth) n-dimensional function?

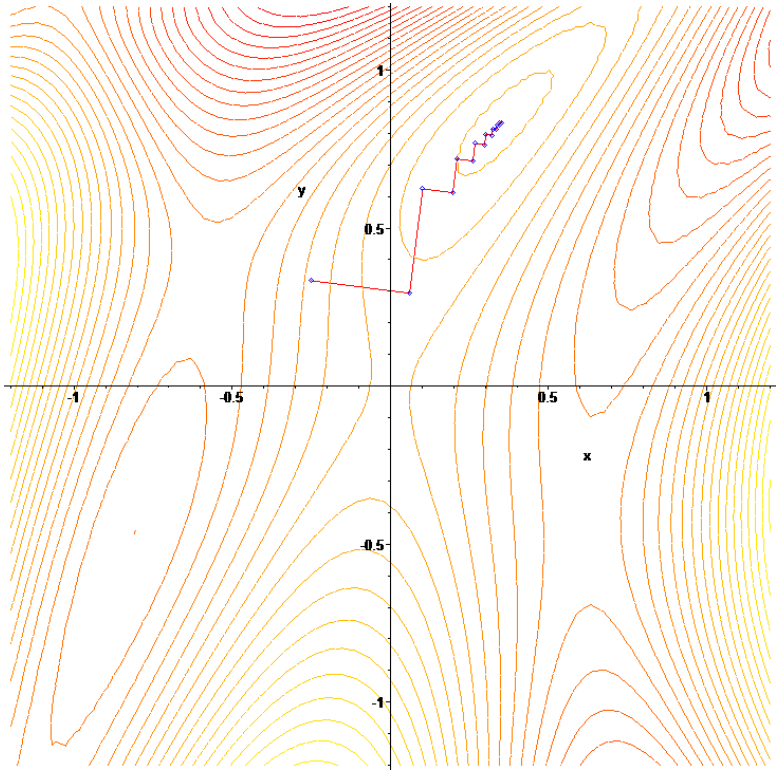
- Take steps in the direction of maximum decrease – i.e. the negative gradient direction!
- This is called **Gradient Descent**



IDC HERZLIYA

Gradient Descent Steps

- For a small enough α>0 we have $f(x) - \alpha\|\nabla f\| < f(x)$, so we take one step of size α in the opposite direction of the gradient
- Note:
  such steps can zig-zag



Back to Minimizing the Cost Function

- Recall that our best hypothesis $\boldsymbol{\theta}^*$ is the one that minimizes the cost function:

$$\boldsymbol{\theta}^* \; attains \; \min_{\boldsymbol{\theta}}\left[\mathrm{J}(\boldsymbol{\theta})\right] = \min_{\boldsymbol{\theta}}\left[\frac{1}{2m}\sum_{i=1}^{m}\left(\boldsymbol{\theta}\cdot\mathbf{x}^{(\mathbf{i})} - \mathrm{y}^{(\mathrm{i})}\right)^2\right]$$

- This is a real valued function of $\vec{\boldsymbol{\theta}} \in \mathbb{R}^n$
- So - we can start with some initial guess $\overrightarrow{\boldsymbol{\theta}_{\mathbf{0}}^*}$ and then use gradient descent

IDC HERZLIYA

Gradient Descent Algorithm

- Start with some value for $\boldsymbol{\theta} = (\theta_0, \theta_1, \theta_2, \ldots, \theta_n)$
- Repeat until you reach a minimum:

  For all j,

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

- $\alpha > 0$ is a parameter of the algorithm called the learning rate
- Updates are simultaneous (in all $n + 1$ directions)
- In the general case this process can still be trapped in local minima!

IDC HERZLIYA

Minimizing the Cost Function

- Hence, our best hypothesis $\boldsymbol{\theta}^*$ would be the one that minimizes the cost function:

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \left[J(\boldsymbol{\theta}) \right] = \arg\min_{\boldsymbol{\theta}} \left[\frac{1}{2m} \sum_{i=1}^{m} \left(\boldsymbol{\theta} \cdot \mathbf{x}^{(i)} - y^{(i)} \right)^2 \right]$$

- How can we find it?

Calculating the Partial Derivative

$$\frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1 \dots, \theta_1) = \frac{\partial}{\partial \theta_j} \frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2$$

$$= \frac{1}{2m} \sum_{i=1}^{m} \frac{\partial}{\partial \theta_j} (\theta_0 + \theta_1 x_1^{(i)} + \cdots + \theta_j x_j^{(i)} + \cdots \theta_n x_n^{(i)} - y^{(i)})^2$$

$$= \frac{1}{2m} \sum_{i=1}^{m} 2(\theta_0 + \theta_1 x_1^{(i)} + \cdots + \theta_j x_j^{(i)} + \cdots \theta_n x_n^{(i)} - y^{(i)}) \cdot x_j^{(i)}$$

$$= \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) \cdot x_j^{(i)}$$

IDC HERZLIYA

Gradient Descent for Linear Regression

- Initialize $\boldsymbol{\theta} = (\theta_0, \theta_1, \theta_2, \ldots, \theta_n)$
- Repeat until you reach a minimum (or stop cdn):
  - For all 0≤j ≤n,

$$\theta_j := \theta_j - \alpha \frac{1}{m} \sum_{i=1}^{m} \left(h_\theta\left(x^{(i)}\right) - y^{(i)} \right) \cdot x_j^{(i)}$$

- In words: set the new θ_j to the current θ_j minus the learning rate (α) times the partial derivative of the error function with respect to θ_j , computed at the current θ.
- Also remember that $x_0^{(i)}$=1

IDC HERZLIYA

Recitation

- Description of HWA1
- Feature scaling
- Learning rate
- Python

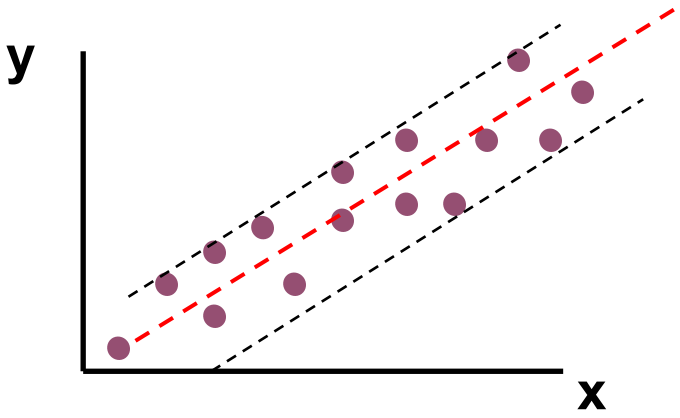
Extra slides/notes

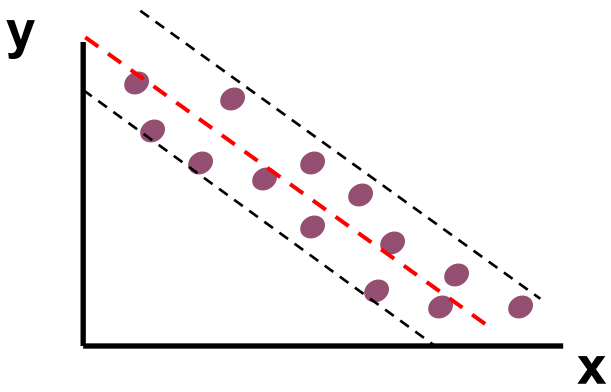
- Correlations
- Pseudo-inverse of a matrix
- Going beyond Linear

IDC HERZLIYA

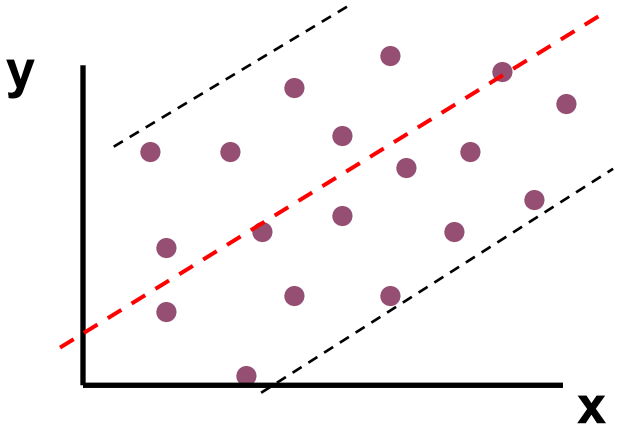
Strong vs. Weak Linear Relationship

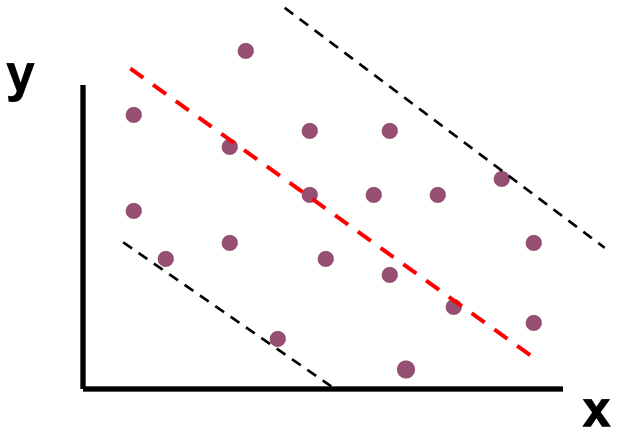
Strong relationships





Weak relationships

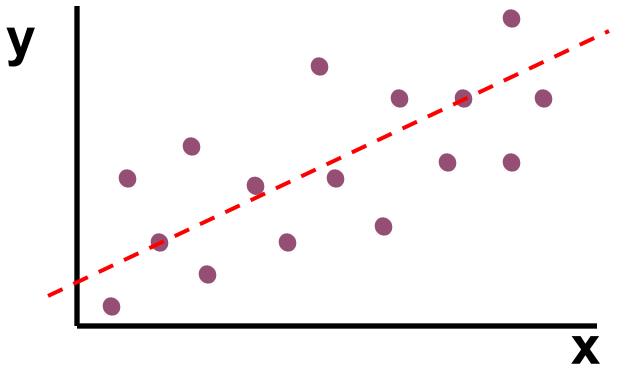


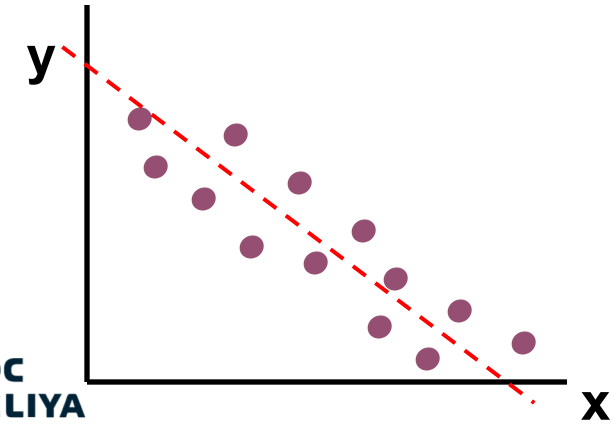


IDC HERZLIYA

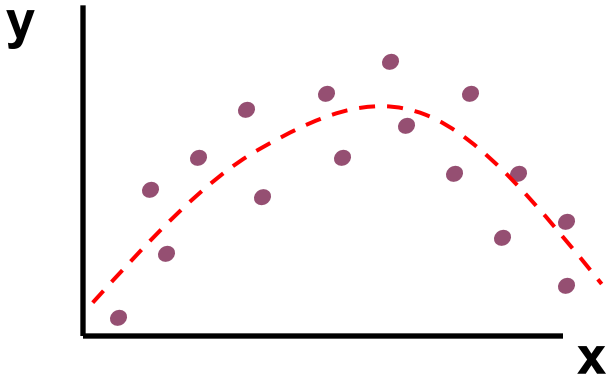
Not Everything is Linear!

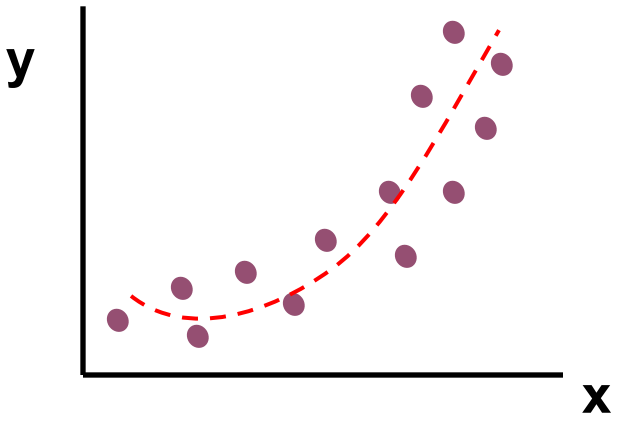
Linear relationships





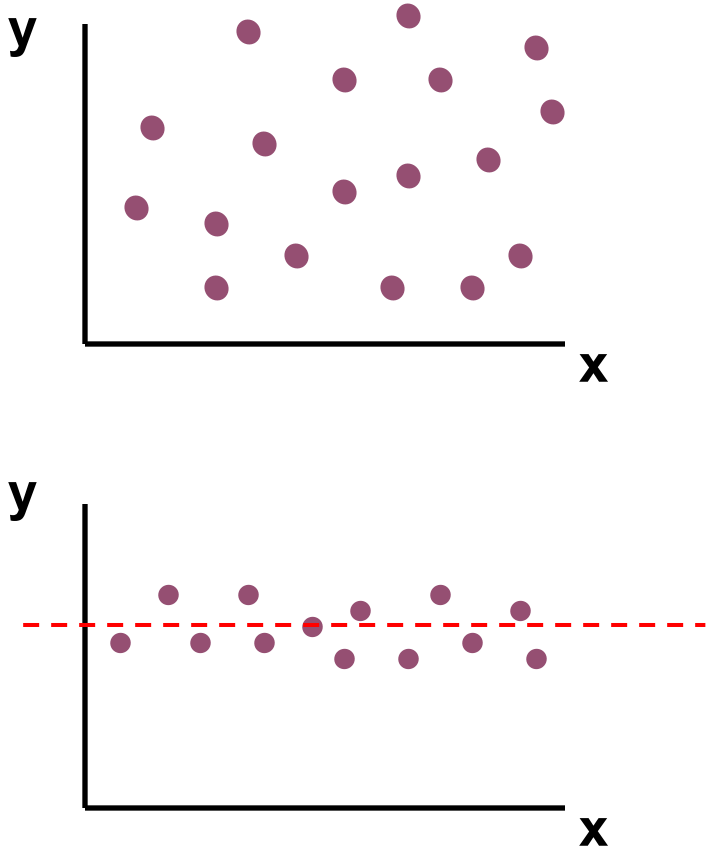
Curvilinear relationships





IDC HERZLIYA

No Evident Relationship



IDC HERZLIYA

Correlation Analysis

- Correlation analysis is used to measure strength of the association (**linear relationship**) between two variables
  - Only concerned with strength of the relationship
  - No causal effect is implied
- The sample (Pearson) correlation coefficient **r** is a measure of the strength of the linear relationship between two variables, based on sample observations

IDC HERZLIYA

The Pearson Correlation Coefficient

$$r = \frac{\sum_{i=1}^{m}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{m}(x_i - \bar{x})^2 \sum_{i=1}^{m}(y_i - \bar{y})^2}} = \frac{\sigma_{xy}}{\sigma_x \sigma_y}$$

Where:

r = Sample Pearson correlation coefficient

m = Number of samples

x = Value of an explaining variable

y = Value of the dependent variable

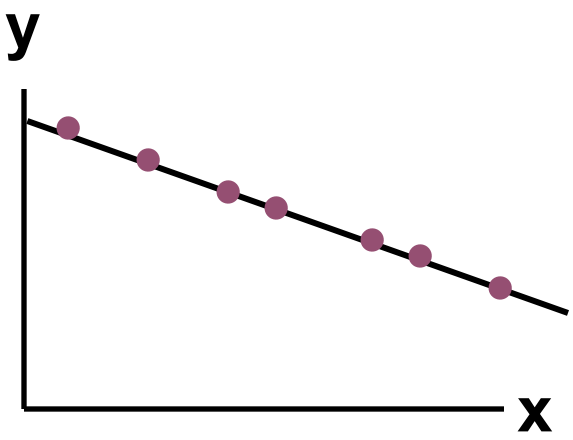
IDC HERZLIYA

Features of r

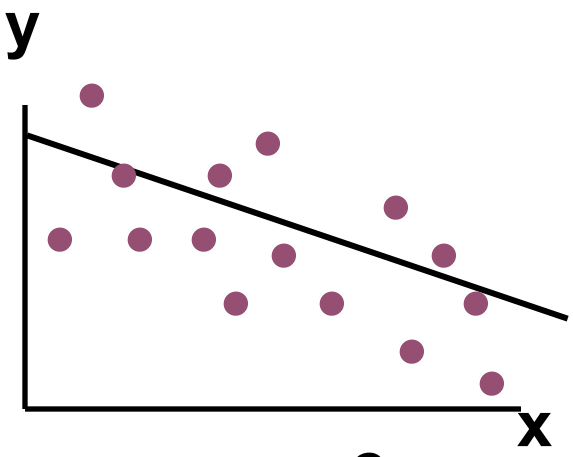
- Unit free
- Ranges between -1 and 1
- The closer to 0, the weaker the linear relationship
- The closer to -1, the stronger the negative linear relationship
- The closer to 1, the stronger the positive linear relationship

IDC HERZLIYA

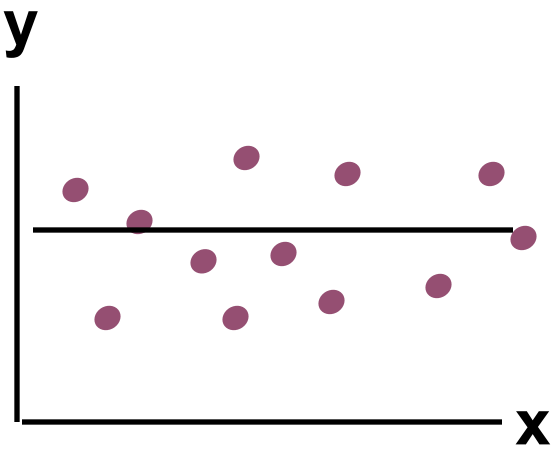
Examples of r Values



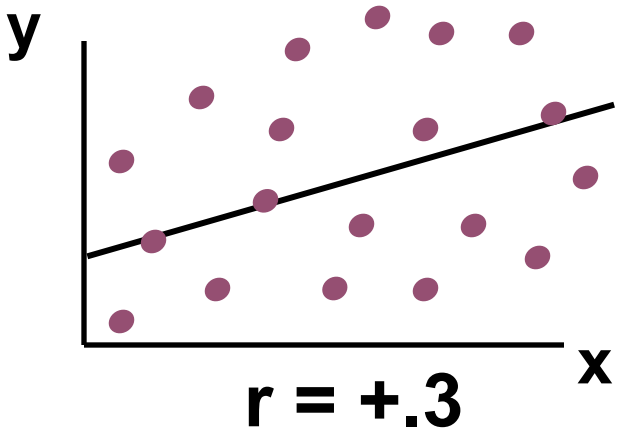
r = -1



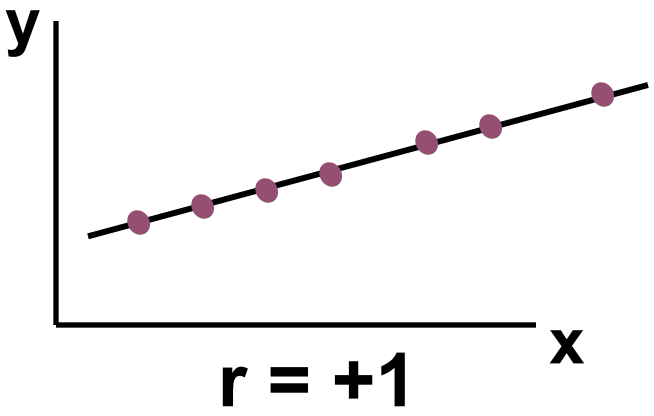
r = -.6



r = 0



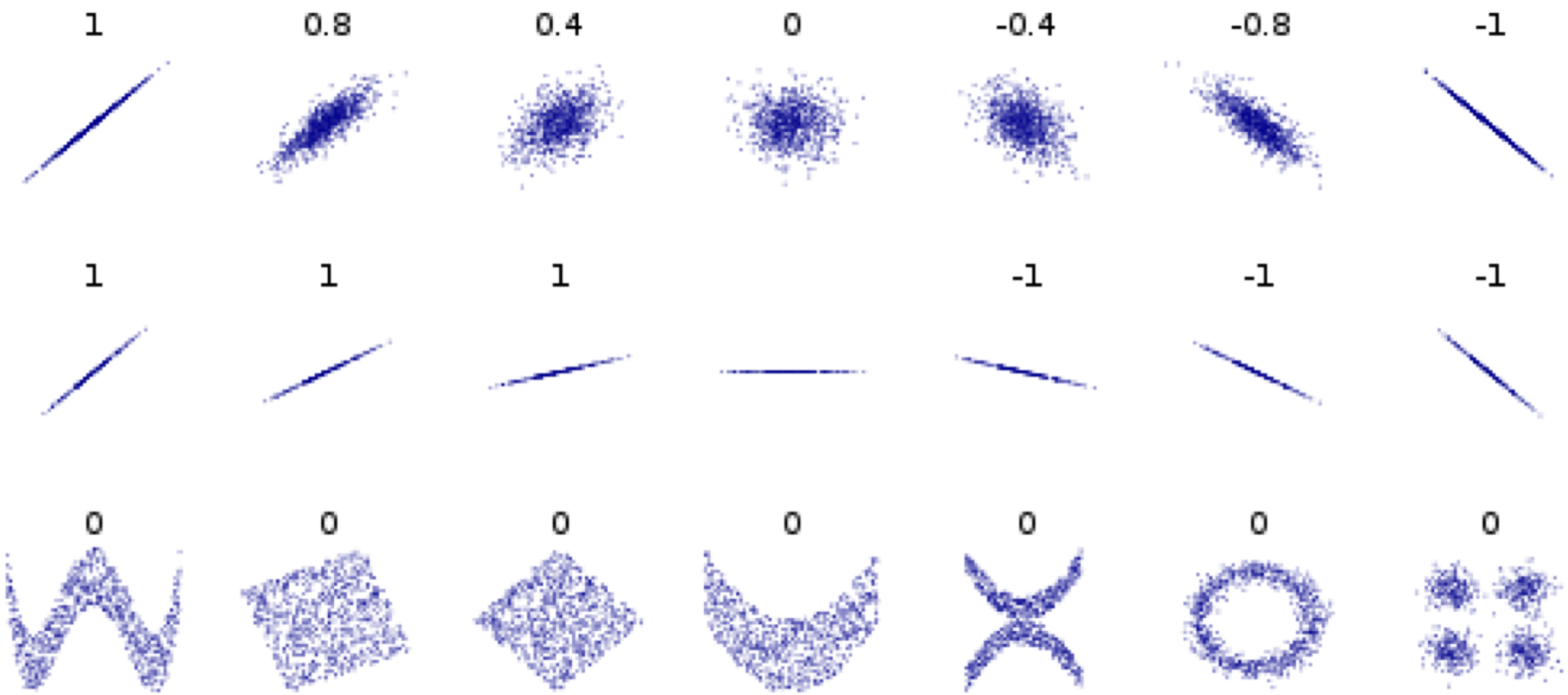
r = +.3



r = +1

IDC HERZLIYA

More Complex Relationships May Not Be Captured Using Pearson r



IDC HERZLIYA

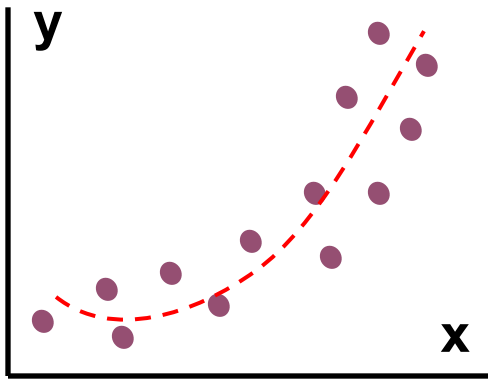
Polynomial Regression

- We can expand our feature space by using functions of the original features.
- For example, if we want to use a cubic function feature space we can define:

$$x_0 = 1, x_1 = x, x_2 = x^2, x_3 = x^3$$

then use regular regression and in essence we are learning the function

$$\theta_0 + \theta_1 x + \theta_1 x^2 + \theta_3 x^3$$



IDC HERZLIYA

Other Functions

- We can use other functions of features such as $x^{1/q}$
- We can define functions of sets of variables such as $x_1 x_5 x_7$ or $x_1^2 x_5 x_7^3$
- All of these can be represented as new features and increase the dimension of the entire calculation

IDC HERZLIYA

Closed form solution?

$$X \cdot \theta = y$$

$$\begin{pmatrix} x_0^{(1)} & x_1^{(1)} & \dots & x_n^{(1)} \\ x_0^{(2)} & x_1^{(2)} & \dots & x_n^{(2)} \\ \vdots & \vdots & & \vdots \\ \vdots & \vdots & & \vdots \\ x_0^{(m)} & x_1^{(m)} & \dots & x_n^{(m)} \end{pmatrix} \begin{pmatrix} \theta_0 \\ \vdots \\ \theta_n \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \\ \\ \\ y_m \end{pmatrix}$$

- If X is square and non singular we can write $\vec{\boldsymbol{\theta}} = X^{-1}y$
- Most of the time X is overdetermined ($m \gg n$).

IDC HERZLIYA

Minimum when Derivative is 0

- Lets look at the error vector e = X· $\vec{\boldsymbol{\theta}}$– y
- Minimizing the squared length of this error vector is equivalent to minimizing the sum of squared distance function:

$$E(\boldsymbol{\theta}) = \|\boldsymbol{X\theta} - \boldsymbol{y}\|^2 = \sum_{\boldsymbol{i=1}}^{\boldsymbol{m}} \left(\boldsymbol{x(i)\theta^T} - \boldsymbol{y(i)}\right)^2$$

- Forming the derivatives yields (see notes):

$$\nabla E(\boldsymbol{\theta}) = \sum_{\boldsymbol{i=1}}^{\boldsymbol{m}} \boldsymbol{2}\left(\boldsymbol{x(i)\theta^T} - \boldsymbol{y(i)}\right)\boldsymbol{x(i)} = \boldsymbol{2X^T(X\theta} - \boldsymbol{y})$$

IDC HERZLIYA

Pseudo-inverse & the Normal Equation

- Setting the gradient to zero yields the necessary condition for minimum (see notes):

$$\mathbf{X}^T\mathbf{X}\cdot\vec{\boldsymbol{\theta}} = \mathbf{X}^T\overrightarrow{\mathbf{Y}}$$

- Now, X^TX is square and often nonsingular and so we can solve for $\vec{\boldsymbol{\theta}}$ uniquely as:

$$\vec{\boldsymbol{\theta}} = pinv(\mathbf{X})\,\overrightarrow{\mathbf{Y}} \qquad \textbf{where} \qquad pinv(\mathbf{X}) = \left(\mathbf{X}^T\mathbf{X}\right)^{-1}\mathbf{X}^T$$

- The n×m matrix $\boldsymbol{pinv(X)} = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T$ is called the **pseudo inverse** of **X** (which is mxn)
- If **X** is square it is just its inverse.

IDC HERZLIYA

Gradient Descent and Pseudo Inverse

- $\mathbf{X^TX}$ can still be singular (or not full rank) and not have an inverse. This can be resolved with some more algebra.
- This pinv technique doesn't work for all error functions J. Gradient descent is more general.
- Gradinet descent allows parallelization.

IDC HERZLIYA

Summary

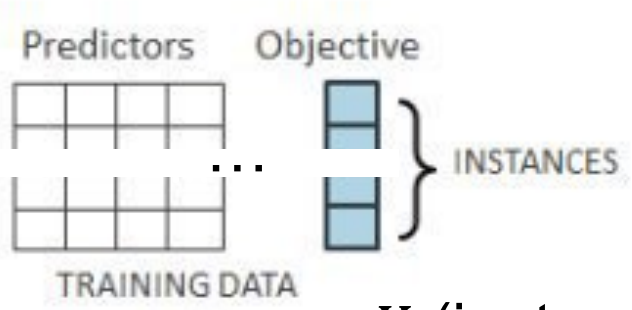
- Regression learns a function to predict values based on a vector of features
- Linear Regression uses a Gradient Descent Algorithm or a Pseudo-Inverse solution
- Gradient descent is a general minimization procedure
- There are other (non-linear) relations between variables
- The Pearson correlation coefficient is a measure of the strength of the linear relationship between two variables
- Pseudo inverse solution

IDC HERZLIYA

Summary - cont

The execution algorithm:

$y = f(x) = x\theta^T$



X (instances by features)

y (instances)

The learning algorithm (regression)