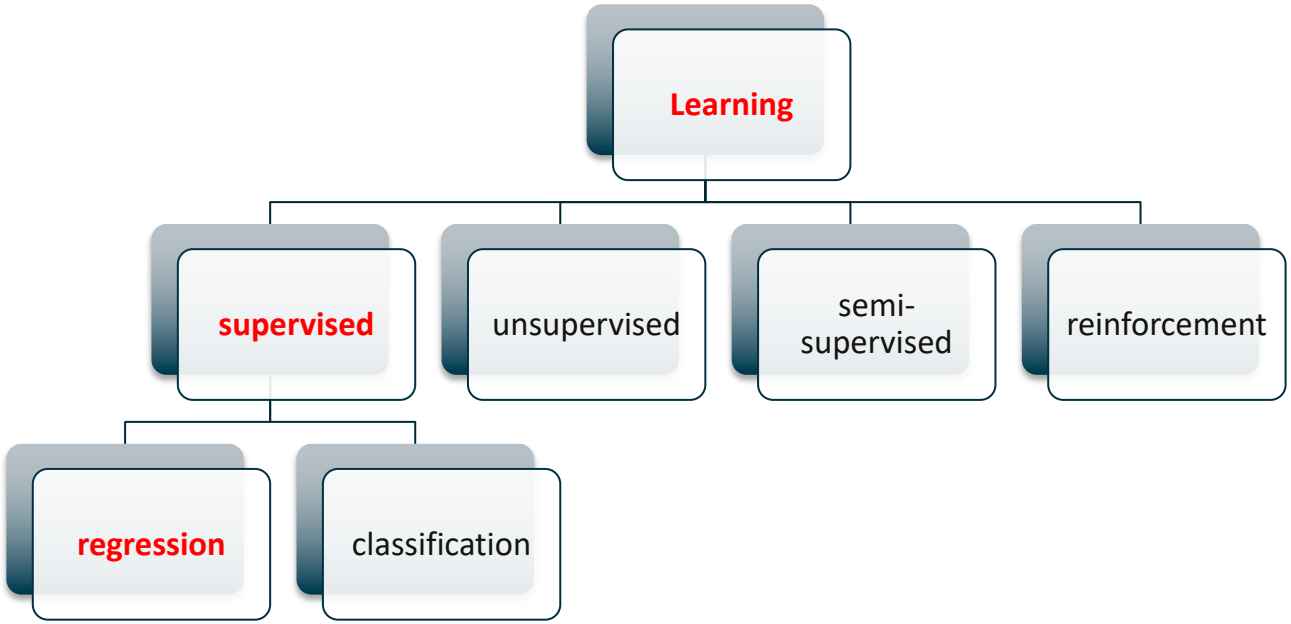
DAI-Labor
Distributed Artificial Intelligence Laboratory

L02 – Linear Regression

IV Foundations of Data Science

David Schultz

Content


Learning
supervised
unsupervised
semi-supervised
reinforcement
regression
classification

Content

- Problem and motivation of curve fitting
- Empirical risk minimization
- Simple linear regression
- Gradient descent for SLR

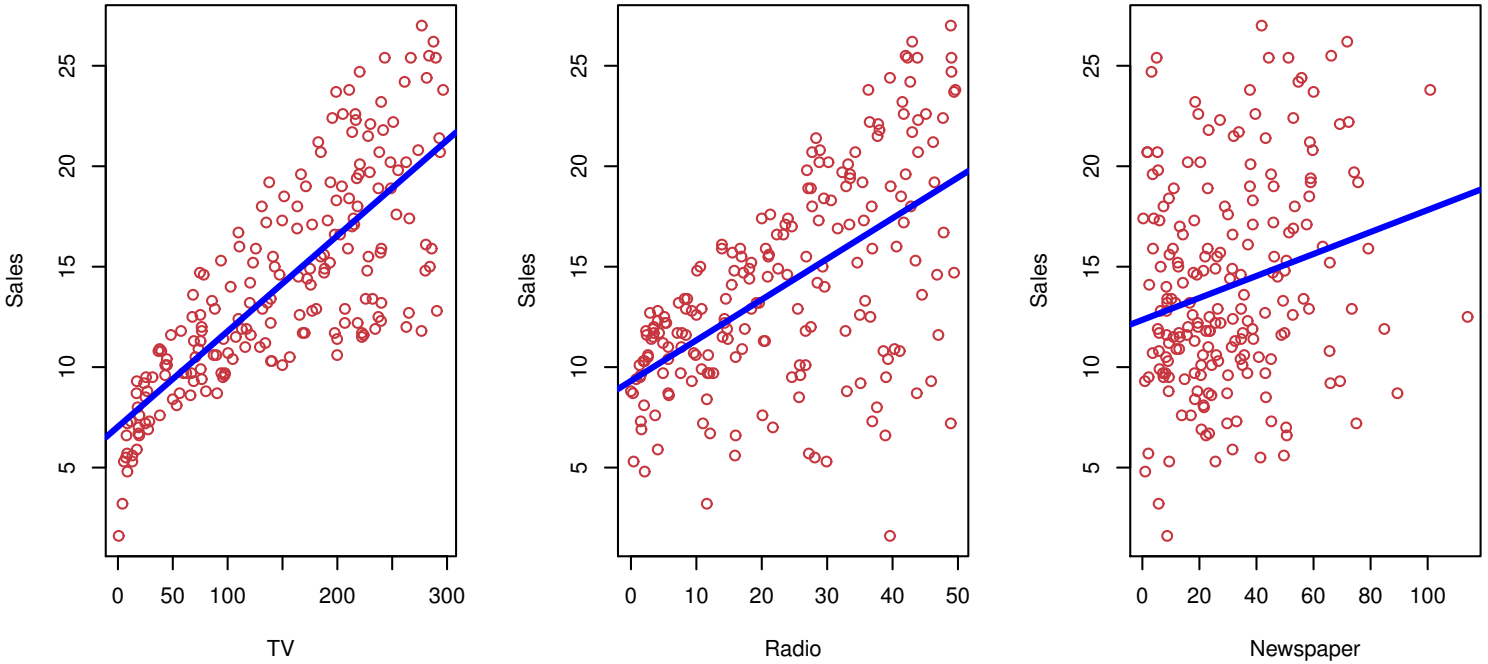
Problem Setting

Example: Advertisement

ID	TV	radio	newspaper	sales
1	230.1	37.8	69.2	22.1
2	44.5	39.3	45.1	10.4
3	17.2	45.9	69.3	9.3
4	151.5	41.3	58.5	18.5
5	180.8	10.8	58.4	12.9
6	8.7	48.9	75	7.2
7	57.5	32.8	23.5	11.8
8	120.2	19.6	11.6	13.2
9	8.6	2.1	1	4.8
10	199.8	2.6	21.2	10.6
…	…	…	…	…
199	283.6	42	66.2	25.5
200	232.1	8.6	8.7	13.4

Sales in 1000 units of a product in 200 markets as a function of advertisement budgets in 1000$ for TV, radio, and newspaper media.

Example: Advertisement



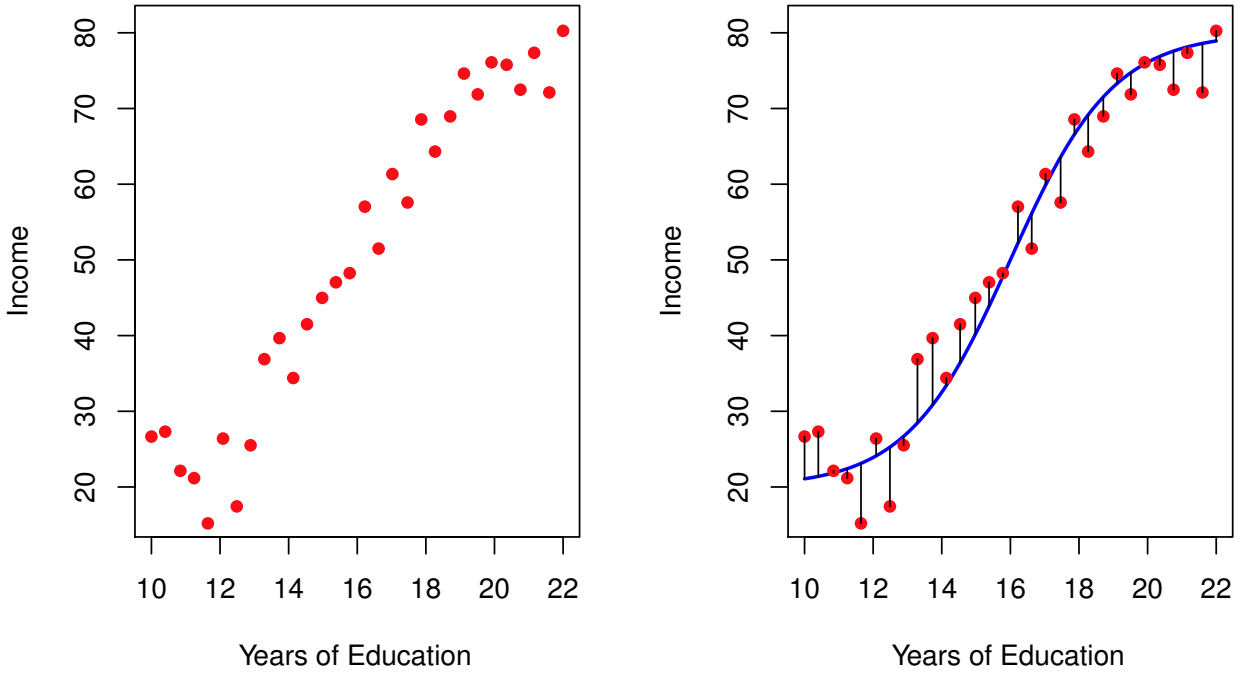
Sales in 1000 units of a product in 200 markets as a function of advertisement budgets in 1000$ for TV, radio, and newspaper media. Blue lines represent the "*best*" linear fit of the data.

Example: Income (simulated data)

ID	education	income
1	10.00	26.66
2	10.40	27.31
3	10.84	22.13
4	11.24	21.17
5	11.65	15.19
6	12.09	26.40
7	12.49	17.44
8	12.89	25.51
9	13.29	36.88
10	13.73	39.67
…	…	…
30	22.00	80.26

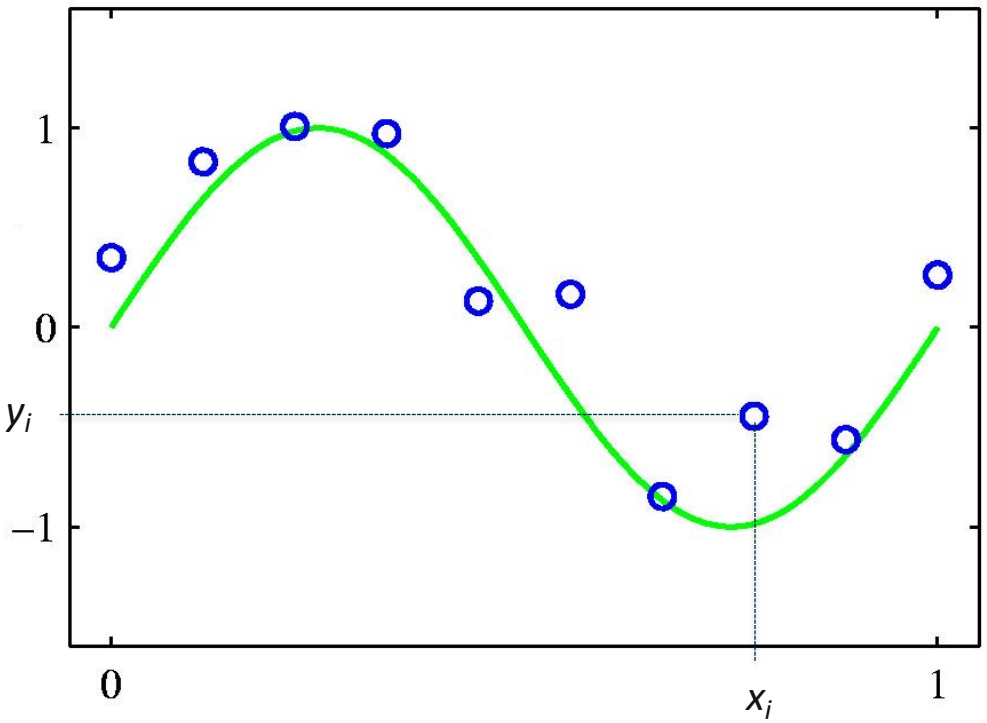
Simulated data: Years of education for 30 individuals and their income (in tens of thousands of dollars) .

Example: Income (simulated data)



Income data. *Left:* The red dots are the observed values of income and years of education. *Right:* The blue curve represents the true relationship between income and years of education. The black lines represent the error associated with each observation.

Example: Sine-Wave (simulated data)



Sinus-wave with measurement errors (noise). Input values x_i generated uniformly from (0,1). Corresponding output values y_i are of the form $y_i = \sin 2\pi x_i + \varepsilon$, where ε is random noise modelled by a Gaussian distribution with zero mean and std 0.3.

Data Representation

Advertisement data

ID	Input			Output
	TV	radio	news	sales
	x_1	x_2	x_3	y
1	230.1	37.8	69.2	22.1
2	44.5	39.3	45.1	10.4
…	…	…	…	…
200	232.1	8.6	8.7	13.4

Input (feature vector)	Output
$x_1 = (x_{11}, x_{12}, x_{13}) = (230.1, 37.8, 69.2)$	$y_1 = 22.1$
$x_2 = (x_{21}, x_{22}, x_{23}) = (44.5, 39.3, 45.1)$	$y_2 = 10.4$
⋮	⋮
$x_m = (x_{m1}, x_{m2}, x_{m3}) = (232.1, 8.6, 8.7)$	$y_m = 13.4$

Data Representation

- Training set $(x_1, y_1), (x_2, y_2), \ldots, (x_m, y_m) \in \mathbb{R}^n \times \mathbb{R}$

m	number of training examples
n	number of features
$x_i = (x_{i1}, \ldots, x_{in})$	feature vector (input) of i-th training example
x_{ij}	j-th feature of i-th training example
y_i	output of i-th training example

Ambiguity in Notation

x_i represents *i*-th input example

x_j represents *j*-th feature

$x_1 = (x_{11}, x_{12}, x_{13}) = (230.1, 37.8, 69.2)$

$x_2 = (x_{21}, x_{22}, x_{23}) = (44.5, 39.3, 45.1)$

TV	radio	news	sales
x_1	x_2	x_3	y
...	...	...	...

Problem of Curve Fitting

Assumption

Let $(x_1, y_1), \ldots, (x_m, y_m)$ be a training set such that

$$y_i = f(x_i) + \varepsilon,$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is an unknown function and $\varepsilon \sim \mathcal{N}(0, \sigma^2)$ is a random error term.

Goal

Estimate unknown function $f : \mathbb{R}^n \to \mathbb{R}$ *as good as possible.*

Notations:

m	number of training examples
n	number of features
x_{ij}	j-th feature of i-th training example
$x_i = (x_{i1}, \ldots, x_{in})$	feature vector i-th training example
y_i	output of i-th training example x_i

Why Estimating *f* ?

- Prediction
  - new input examples *x* are easy but output *y* is hard to obtain
  - predict output by hypothesis $\hat{y} = h(x)$ derived from training set
  - example: traffic volume at a certain time
  - example: risk of patient for an adverse reaction to a particular drug.

- Inference
  - understand relationship between input *x* and output *y*
  - which features are associated with the output?
  - what is the relationship between output and each feature?
  - is output linearly related to input?

Curve Fitting

Empirical Risk Minimization

Expected Risk

Goal

Estimate unknown function $f : \mathbb{R}^n \rightarrow \mathbb{R}$ *as good as possible.*

Let $P(x, y)$ be the probability distribution on the input-output space $\mathcal{Z} = \mathbb{R}^n \times \mathbb{R}$, let $\ell(\hat{y}, y)$ be a loss function, and let $\mathcal{H} = \{h : \mathbb{R}^n \rightarrow \mathbb{R}\}$ be a hypothesis space. Then

$$E[h] = \int_{\mathcal{Z}} \ell(h(x), y) dP(x, y).$$

is the expected risk of hypothesis $h \in \mathcal{H}$.

Goal

Find hypothesis $h \in \mathcal{H}$ that minimizes the expected risk $E[h]$

Problem: In general, $E[h]$ can not be evaluated, because $P(x, y)$ is unknown

Empirical Risk Minimization

- Suppose a training set $(x_1, y_1), (x_2, y_2), \ldots, (x_m, y_m)$ is given

Empirical Risk Minimization Principle

1. Determine hypothesis space $\mathcal{H} = \{h\colon \mathbb{R}^n \longrightarrow \mathbb{R}\}$
2. Determine loss function $\ell(h(x), y)$
3. Minimize cost function (empirical risk) over all hypotheses $h \in \mathcal{H}$

$$E_m[h] = \frac{1}{m} \sum_{i=1}^{m} \ell(h(x_i), y_i)$$

4. Use minimizer h_m^* of $E_m[h]$ as estimate of unknown function f

Components

- Hypothesis space $\mathcal{H} = \{h: \mathbb{R}^n \longrightarrow \mathbb{R}\}$
  - set of functions from which we pick our estimate of function *f*
  - manually pre-specified by data analyst
  - examples:
    - $\mathcal{H}$ = set of linear functions
    - $\mathcal{H}$ = set of polynomials

- Loss function $\ell(h(x), y)$
  - cost of predicting $h(x)$ when actual output is *y*
  - manually pre-specified by data analyst
  - **squared loss** is common choice for curve fitting

$$\ell(h(x), y) = \frac{1}{2}(h(x) - y)^2$$

Components

- Cost function $E_m[h]$
  - aka empirical risk, average loss, mean error, …
  - induced by loss $\ell(h(x), y)$
  - squared loss induces mean squared error (MSE) cost

$$E_m[h] = \frac{1}{m}\sum_{i=1}^{m} \ell(h(x_i)\ ,\ y_i) = \frac{1}{2m}\sum_{i=1}^{m}(h(x_i) - y_i)^2$$

  - MSE arises from Maximum Likelihood Estimation method

Maximum Likelihood Estimation

- Maximum Likelihood Estimation (MLE) is a method belonging to statistical inference
- The goal of statistical inference is to deduce information about a population based on a sample (i.e. observed data) collected from it.
- Assume the distribution of a population is parameterized by $\theta \in \mathbb{R}^d$.
  - For example $\mathcal{F} = \{\mathrm{P}(x;\theta) = \theta^x(1-\theta)^{1-x} | \theta \in (0,1)\}$ models the probability mass function of a Bernoulli(θ)-distribution with success probability θ.
- The goal of MLE is to estimate θ on the basis of an iid. sample.

Maximum Likelihood Estimation

Motivation in case of discrete random variables

- Let $x_1, \dots, x_m$ be an iid. sample of a probability distribution with probability mass function $\mathrm{P}(x; \theta)$ with **unknown** parameter θ.

- The **Likelihood** can be interpreted as the probability(*) of observing the data, when the parameter is θ:

$$L(\theta) = \mathrm{P}(x_1, \dots, x_m; \theta)$$

- Since the sample is iid., we have

$$L(\theta) = \prod_{i=1}^{m} \mathrm{P}(x_i; \theta)$$

(*) $L(\theta)$ is not a probability measure, because $\int_\Theta L(\theta) d\theta \neq 1$ in general!

Maximum Likelihood Estimation

General case

- Let $x_1, \dots, x_m$ be an iid. sample of a probability distribution with probability mass function or probability density function $p(x; \theta)$ with **unknown** parameter θ. The **Likelihood function** is defined by

$$L(\theta) = \prod_{i=1}^{m} p(x_i; \theta)$$

- A **Maximum Likelihood Estimator** $\hat{\theta}$ is a parameter which maximizes the Likelihood

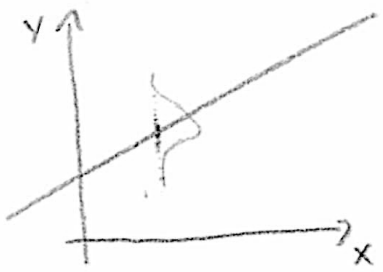
$$\hat{\theta} = \operatorname*{argmax}_{\theta \in \Theta} L(\theta)$$

- Often it is more convenient to consider the **Log-Likelihood**

$$l(\theta) = \ln(L(\theta)) = \sum_{i=1}^{m} \ln\big(p(x_i; \theta)\big)$$

Whiteboard: MSE via Maximum Likelihood

Model $y = f(x) + \varepsilon \;,\; \varepsilon \sim \mathcal{N}(0, \sigma^2)$



$$\Rightarrow p(y \mid x, f, \sigma^2) = \mathcal{N}(y \mid f(x), \sigma^2)$$

Consider training examples $(x_1, y_1), \ldots, (x_m, y_m) \in \mathbb{R}^m \times \mathbb{R}$ drawn independently from $P(x, y)$.

MLE Choose $h^* \in H$ that maximizes

$$L(h) = \prod_{i=1}^{m} \mathcal{N}(y_i \mid h(x_i), \sigma^2)$$

Maximizing L is equivalent to maximizing $\ln(L) = \ell(h)$

$$\ell(h) = \sum_{i=1}^{m} \ln \mathcal{N}(y_i \mid h(x_i), \sigma^2)$$

recall $\mathcal{N}(y \mid \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2}\frac{(y-\mu)^2}{\sigma^2}}$

$$\Rightarrow \ell(h) = \underbrace{\sum_{i=1}^{m} \ln\left(\frac{1}{\sqrt{2\pi\sigma^2}}\right)}_{\text{const.}} - \sum_{i=1}^{m} \frac{(y_i - h(x_i))^2}{2\sigma^2}$$

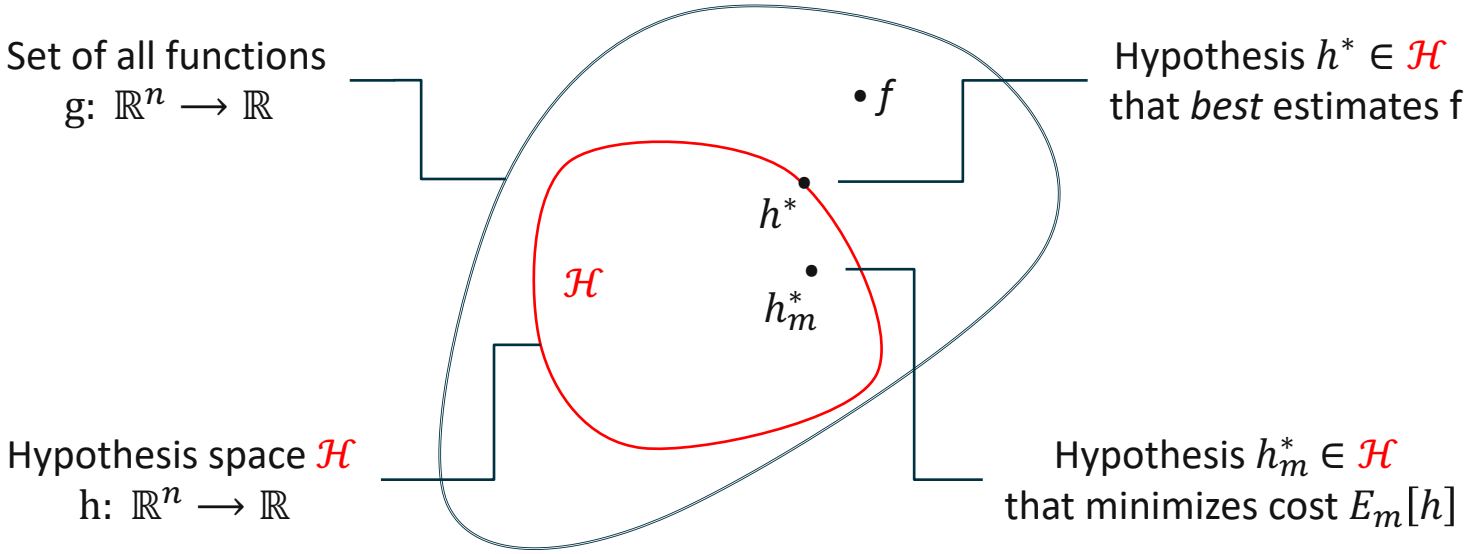
$$\Rightarrow \operatorname*{argmax}_{h \in H} \ell(h) = \operatorname*{argmin}_{h \in H} \sum_{i=1}^{m} \frac{(y_i - h(x_i))^2}{2\sigma^2}$$

$$= \operatorname*{argmin}_{h \in H} \frac{1}{2m} \sum_{i=1}^{m} (h(x_i) - y_i)^2$$

$$= \operatorname*{argmin}_{h \in H} E_m[h] \quad \text{(Mean Squared Error)}$$

Empirical Risk Minimization

Hypothesis space:
Set of functions from which we pick our estimate of unknown function f



Theory: Study necessary and sufficient conditions for consistency:

- $E_m[h]$ converges probabilistically to $\mathrm{E}[h]$ for fixed h with increasing m
- $E_m[\,h_m^*]$ converges probabilistically to $\mathrm{E}[\,h_\infty^*] = \mathrm{E}[h^*]$ with increasing m

When Empirical Risk Minimization fails (example)

- Assumptions
  - $\mathcal{H}$ is the set of all functions
  - Let $f: [0,1] \longrightarrow \mathbb{R}$ be a function of the form

$$f(x) = \begin{cases} 0 & : \quad 0 \leq x < 0.5 \\ 1 & : \quad 0.5 \leq x \leq 1 \end{cases}$$

  - $(x_1, y_1), (x_2, y_2), \ldots, (x_m, y_m) \in [0,1] \times [0,1]$ is a training set with $y_i = f(x_i)$ for all i.
  - Consider hypothesis $h_m \in \mathcal{H}$

$$h_m(x) = \begin{cases} y_i & : \quad x = x_i \text{ for some } i \in \{1, \ldots, m\} \\ 1 & : \quad \text{otherwise} \end{cases}$$

  - Uniform distribution on $[0,1]$ for x values
- Implications
  - We have $E_m[h_m] = 0$ but $E[h_m] = 0.5$ for every m
  - Hence, h_m is inconsistent
  - Hence, ERM only works when we restrict the hypothesis space.

Curve Fitting

Simple Linear Regression

$(n = 1)$

Examples

ID	TV x	sales y
1	230.1	22.1
2	44.5	10.4
3	17.2	9.3
4	151.5	18.5
5	180.8	12.9
6	8.7	7.2
7	57.5	11.8
8	120.2	13.2
9	8.6	4.8
10	199.8	10.6
…	…	…
199	283.6	25.5
200	232.1	13.4

Advertisement data

ID	education x	income y
1	10.00	26.66
2	10.40	27.31
3	10.84	22.13
4	11.24	21.17
5	11.65	15.19
6	12.09	26.40
7	12.49	17.44
8	12.89	25.51
9	13.29	36.88
10	13.73	39.67
…	…	…
30	22.00	80.26

Income data

Hypothesis Space $\mathcal{H}$

- Hypothesis space of linear functions in one variable $x \in \mathbb{R}$ (i.e. $n = 1$).

$$\mathcal{H} = \{h_w : \mathbb{R} \to \mathbb{R},\ h_w(x) = w_0 + w_1 x \mid w = (w_0, w_1) \in \mathbb{R}^2\}$$

- Hypothesis: linear function

$$h_w(x) = w_0 + w_1 x$$

- Examples:

$$income = w_0 + w_1 \times education$$

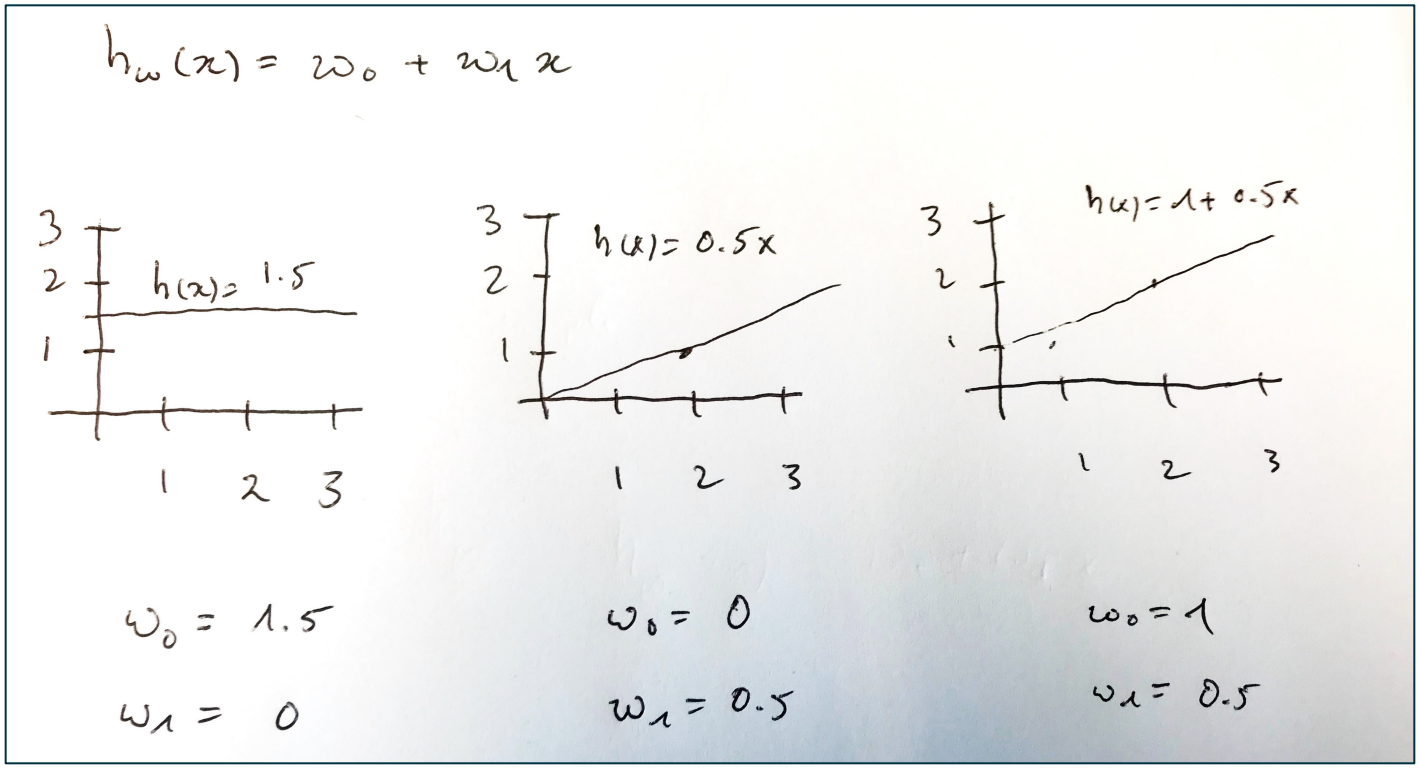
$$sales = w_0 + w_1 \times TV$$

- Parameters $w = (w_0, w_1)$:
  - w_0 : bias, intercept
  - w_1 : weight

Hypothesis Space $\mathcal{H}$

Role of parameters

- Bias w_0 determines position along y-axis
- Weight w_1 determines slope



Loss Function $\ell(h(x), y)$

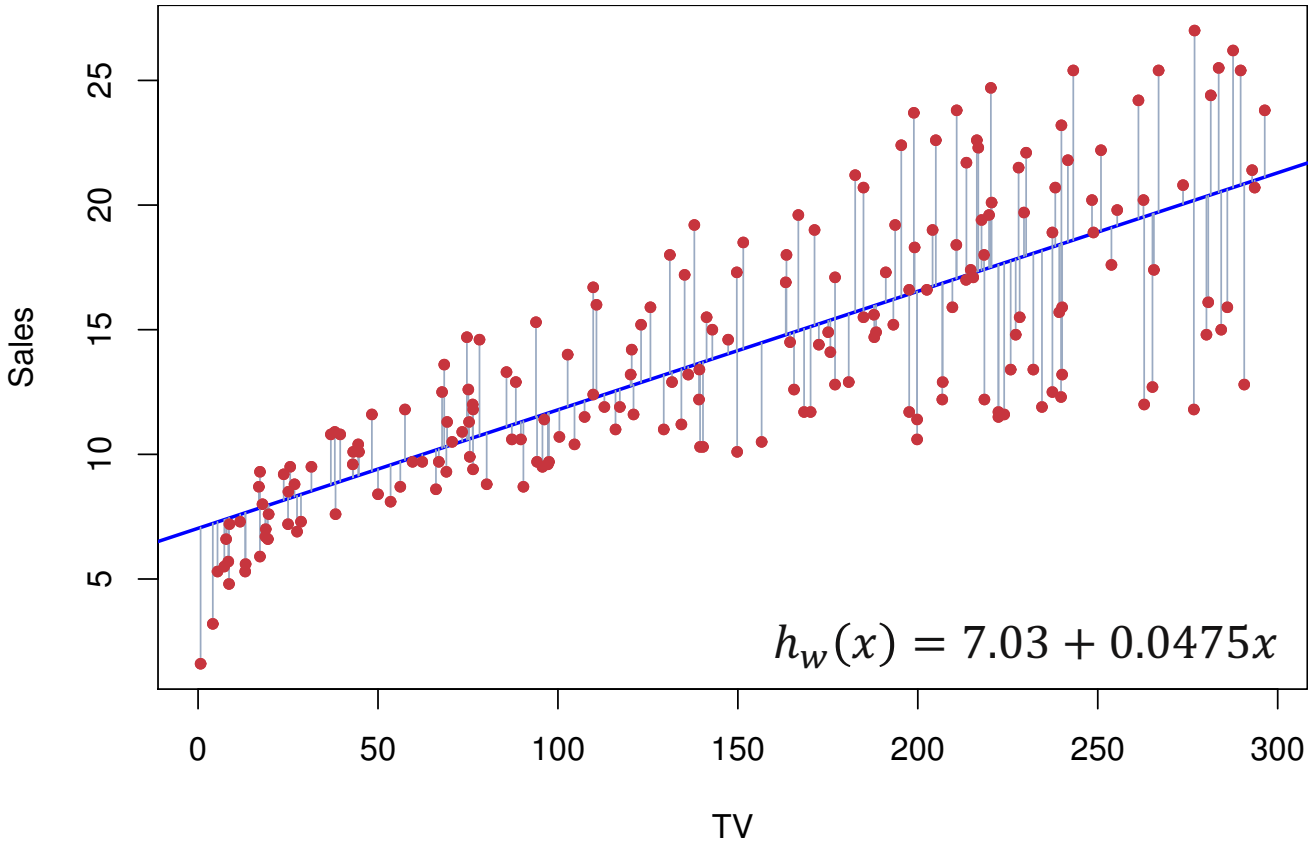
- Squared loss function

$$\ell(h_w(x), y) = \frac{1}{2}(h_w(x) - y)^2 = \frac{1}{2}(w_0 + w_1 x - y)^2$$

- Mean squared error function

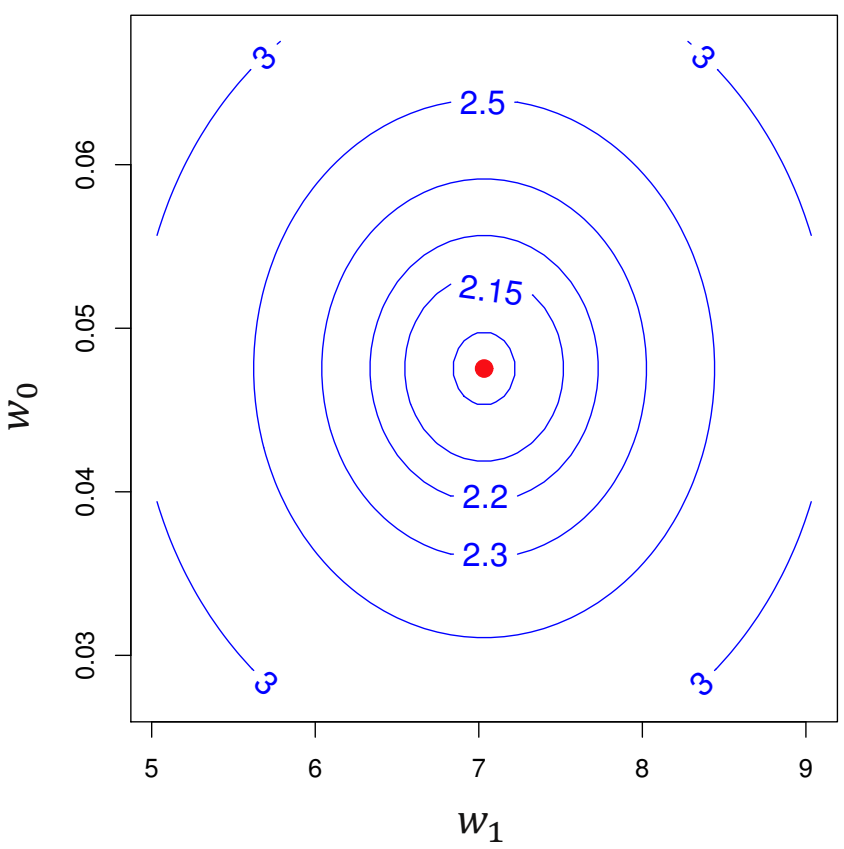
$$E_m[h_w] = \frac{1}{2m}\sum_{i=1}^{m}(h_w(x_i) - y_i)^2 = \frac{1}{2m}\sum_{i=1}^{m}(w_0 + w_1 x_i - y_i)^2$$

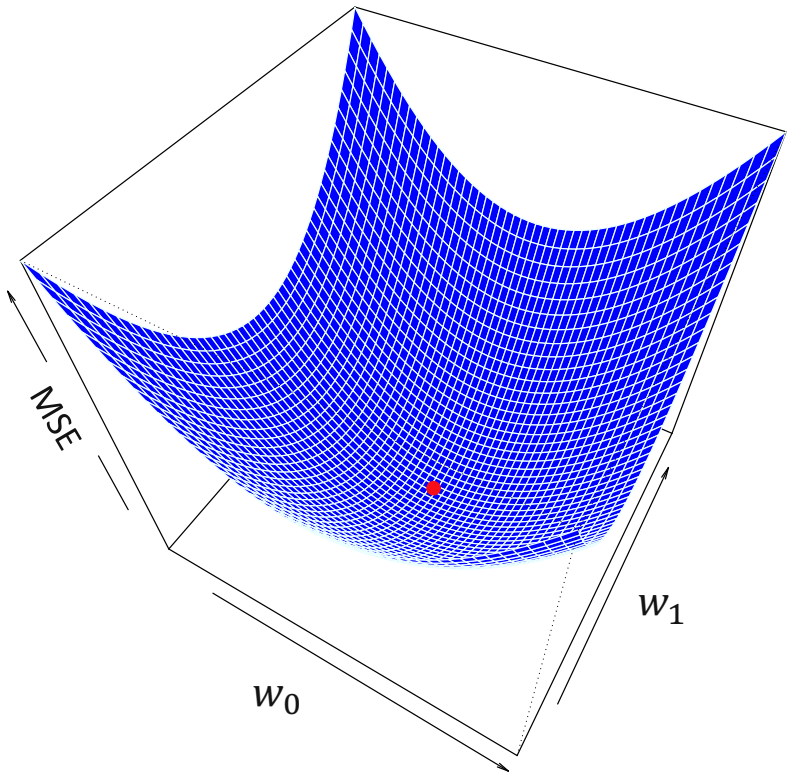
Simple Linear Regression



Least squares fit of sales onto TV.

Mean Squared Error Function





Contour and three-dimensional plots of the MSE on the advertising data, using sales as output and TV as input.

Optimal Model Parameters

- Closed-form solution for the minimizer h_w of $E_m[h_w]$

$$w_0 = \overline{y} - w_1 \overline{x} \ ,$$

$$w_1 = \frac{\sum_{i=1}^{m} (x_i - \overline{x})(y_i - \overline{y})}{\sum_{i=1}^{m} (x_i - \overline{x})^2} \ ,$$

where $\bar{x} = \frac{1}{m} \sum_{i=1}^{m} x_i$, $\bar{y} = \frac{1}{m} \sum_{i=1}^{m} y_i$

Curve Fitting

Gradient Descent

Motivation

- Curve fitting amounts in minimizing the MSE criterion
- Linear regression:
  - MSE can be minimized analytically
- General case:
  - closed-form solution is unknown or even not possible
  - gradient-based methods for minimizing smooth MSE functions
- Gradient-based methods
  - common technique for majority of ML methods

Gradient Descent for Simple Linear Regression

- Notation: to make gradient descent more explicit, we write

$$J(w_0, w_1) = E_m[h_w]$$

- Important property:
  - $J(w_0, w_1)$ is differentiable as a function of $w = (w_0, w_1)$
  - Hence, gradient of $J(w_0, w_1)$ at w exists
  - Hence, gradient descent can be applied

- Basic idea:
  - start with some initial values for w_0 and w_1
  - iteratively adjust w_0 and w_1 to reduce $J(w_0, w_1)$ until termination

Gradients

Gradient:

Let $f: \mathbb{R}^n \to \mathbb{R}$ be differentiable. Then the gradient $\nabla f(x)$ of f at point x is of the form

$$\nabla f(x) = \left(\frac{\partial f}{\partial x_1}(x), \dots, \frac{\partial f}{\partial x_n}(x) \right)^T$$

Properties:

(a) $\nabla f(x)$ points in direction of steepest ascent and $-\nabla f(x)$ in direction of steepest descent

(b) Necessary condition of optimality: if x^* is a local minimum of f, then we have $\nabla f(x^*) = 0$.

Example:

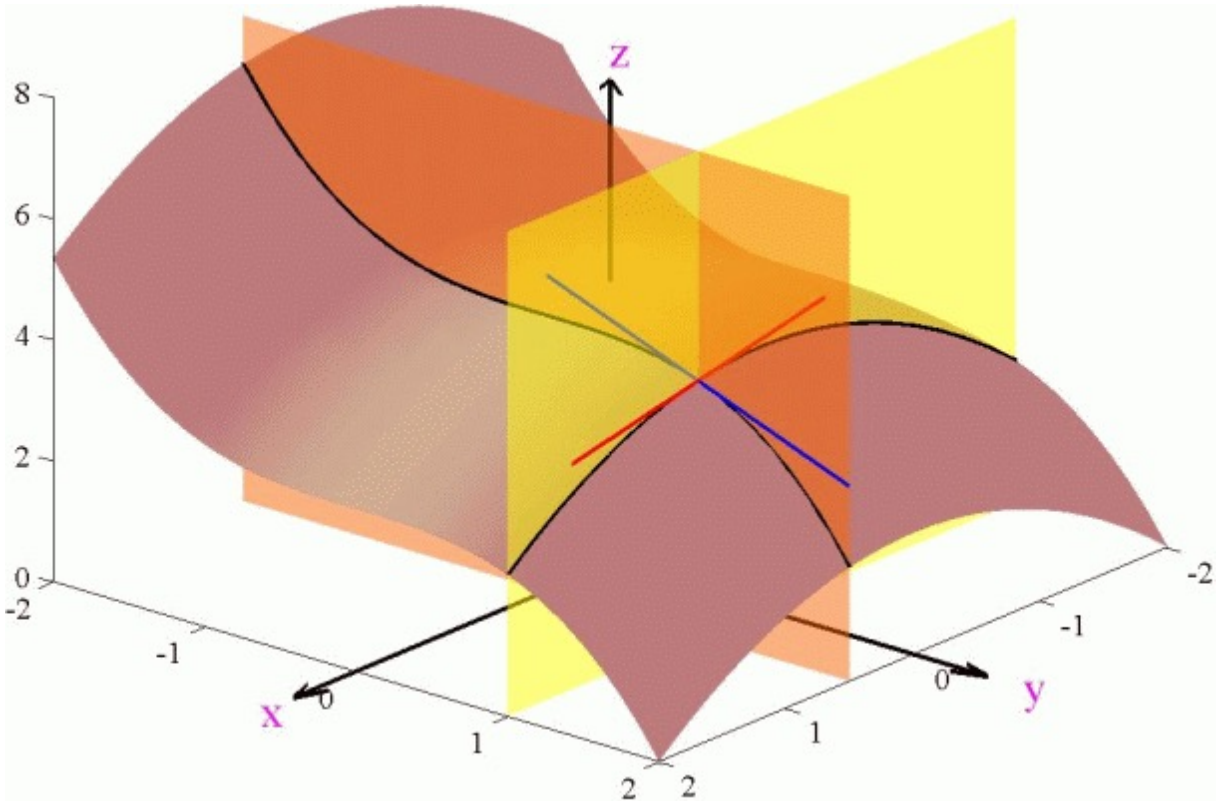
1. Let $f(x) = 2x + 3$. Then

$$\nabla f(x) = f'(x) = 2$$

2. Let $f(x_1, x_2) = x_1^2 + x_2^2$. Then

$$\nabla f(x_1, x_2) = (2x_1, 2x_2)$$

Partial Derivatives



Gradient Descent

- Basic procedure of gradient descent

Repeat

$$w_j \leftarrow w_j - \eta \frac{\partial}{w_j} J(w_0, w_1) \quad \text{for } j = 0, 1$$

until termination

- Notation

η	learning rate / step size
$\frac{\partial}{\partial w_j} J(w_0, w_1)$	partial derivative of $J(w_0, w_1)$

Gradient Descent

- Basic procedure of gradient descent

Repeat

$$w_j \leftarrow w_j - \eta \frac{\partial}{w_j} J(w_0, w_1) \quad \text{for } j = 0, 1$$

until termination

Caution!
Update all w_j simultaneously: First evaluate the right hand side for all j and then update all w_j.

Simultaneous updating (correct)

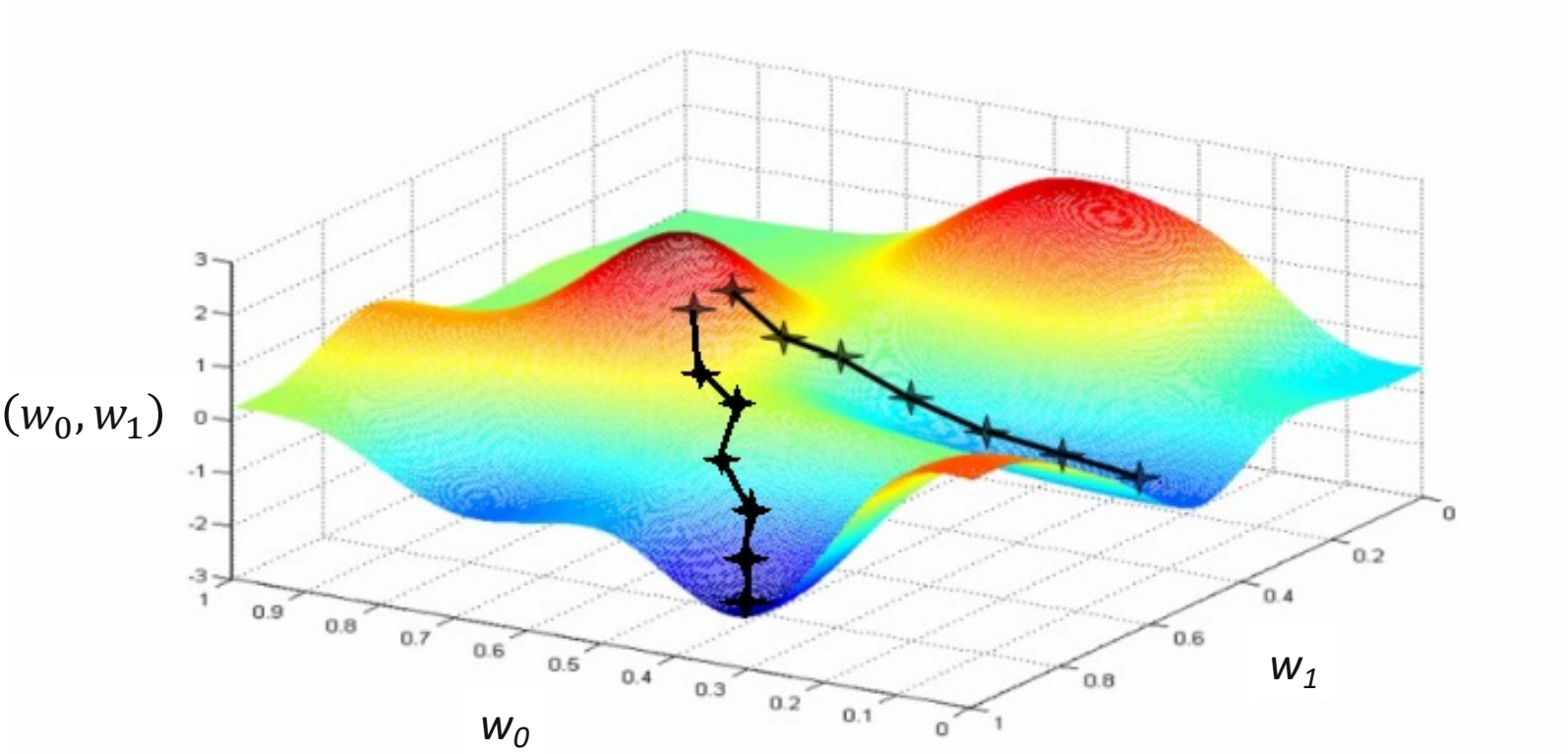
$$d0 := \frac{\partial}{w_0} J(w_0, w_1)$$
$$d1 := \frac{\partial}{w_1} J(w_0, w_1)$$
$$w0 := w0 - eta*d0$$
$$w1 := w1 - eta*d1$$

Sequential updating

$$d := \frac{\partial}{w_0} J(w_0, w_1)$$
$$\mathbf{w0} := w0 - eta*d$$
$$d := \frac{\partial}{w_1} J(\mathbf{w_0}, w_1)$$
$$w1 := w1 - eta*d$$

Incorrect! (not GD)

Gradient Descent



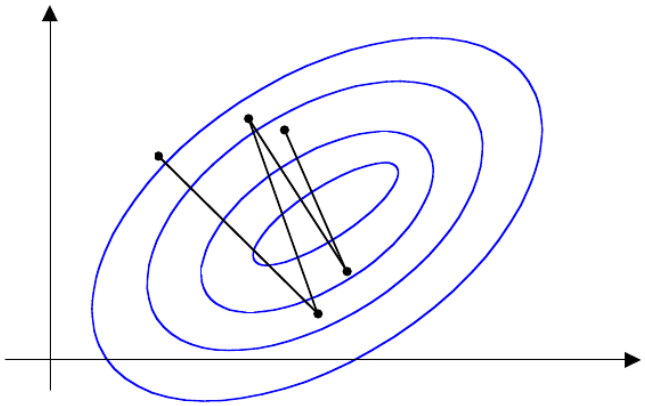
Gradient descent for minimizing the cost $J(w_0, w_1)$. In every iteration, the point (w_0, w_1) is adjusted in direction of steepest descent. Different initializations may lead to different solutions.

© Andrew Ng: *Machine Learning*, MOOC at Coursera.

Gradient Descent

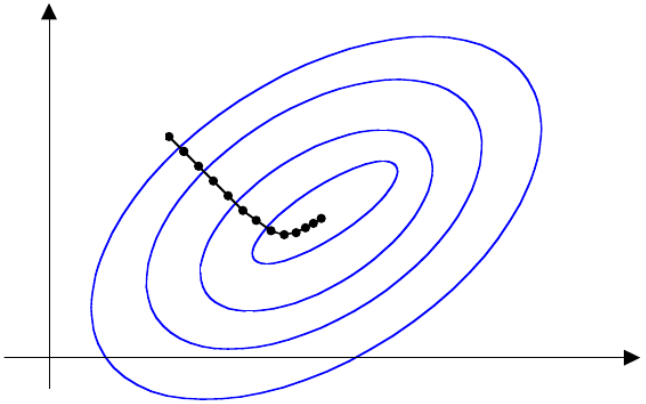
Choice of learning rate η matters

Learning rate η is too large



overshoots, zig-zags, and possibly diverges

Learning rate η is too small



η is too small: slow convergence

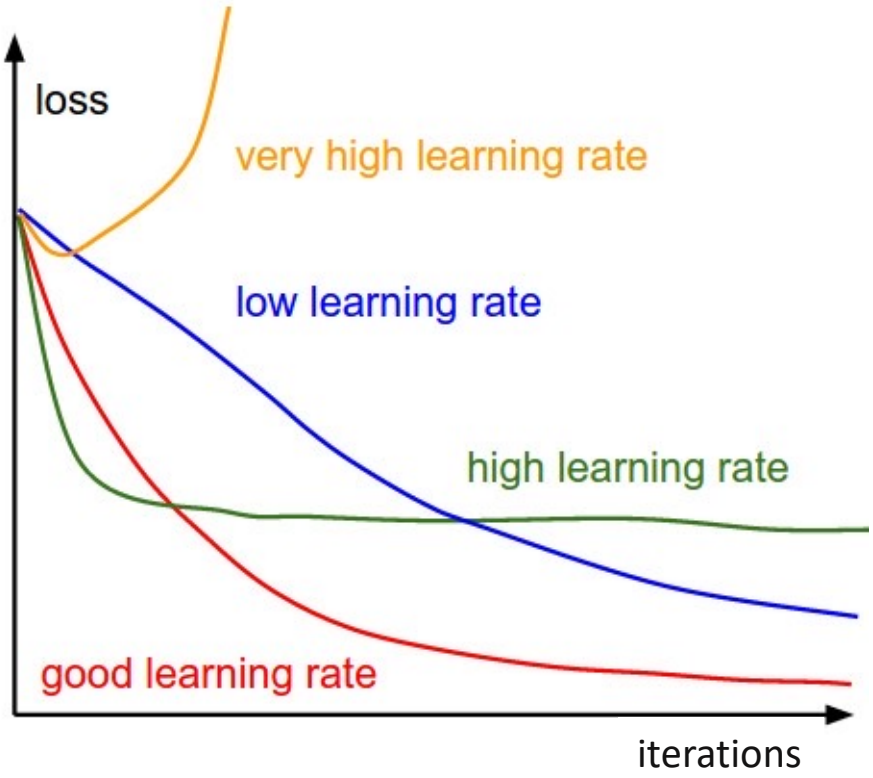
Gradient Descent

Choice of learning rate η matter

Two options:
- use fixed η
- decrease η in each iteration

Finding a suitable fixed η
1. Let $\eta_1 < \eta_2 < \cdots < \eta_k$
2. set $t = 1$
3. set $\eta \leftarrow \eta_t$
4. test GD with η
5. if η too large, set $t \leftarrow t + 1$
6. else use η for full GD

Example for η_t:
$0.3 < 0.1 < 0.03 < 0.01 \cdots$



Idealized plot depicting the MSE of a non-convex loss as a function of the number of iterations for different learning rates.

Gradient Descent

- Basic procedure of gradient descent

Repeat

$$w_j \leftarrow w_j - \eta \frac{\partial}{w_j} J(w_0, w_1) \quad \text{for } j = 0, 1$$

until termination

- Termination
  - terminate after maximum number of simultaneous updates
  - terminate if $\|\nabla J(w_0, w_1)\| < \varepsilon$

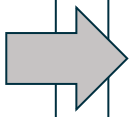
Gradient Descent for Simple Linear Regression

Repeat

$$w_0 \leftarrow w_0 - \eta \frac{\partial}{w_0} J(w_0, w_1)$$

$$w_1 \leftarrow w_1 - \eta \frac{\partial}{w_1} J(w_0, w_1)$$

until termination



Repeat

$$w_0 \leftarrow w_0 - \eta \frac{1}{m} \sum_{i=1}^{m} (h_w(x_i) - y_i)$$

$$w_1 \leftarrow w_1 - \eta \frac{1}{m} \sum_{i=1}^{m} (h_w(x_i) - y_i) x_i$$

until termination

Whiteboard: Partial Derivatives of $J(w_0, w_1)$

Partial derivatives of $J(w_0, w_1)$

$$J(w_0, w_1) = \frac{1}{2m} \sum_{i=1}^{m} (w_0 + w_1 x_i - y_i)^2$$

Consider

$$g_i : \mathbb{R}^2 \xrightarrow{\alpha_i} \mathbb{R} \xrightarrow{\beta_i} \mathbb{R}$$

$$(w_0, w_1) \mapsto \underbrace{w_0 + w_1 x_i - y_i}_{= z_i} \mapsto z_i^2$$

Apply chain rule:

$$\frac{\partial}{\partial w_0} g_i(w_0, w_1) = 2 z_i \cdot 1 = 2(w_0 + w_1 x_i - y_i)$$

$$\frac{\partial}{\partial w_1} g_i(w_0, w_1) = 2 z_i \cdot x_i = 2(w_0 + w_1 x_i - y_i) x_i$$

Apply sum rule:

$$\frac{\partial}{\partial w_0} J(w_0, w_1) = \frac{1}{m} \sum_{i=1}^{m} w_0 + w_1 x_i - y_i$$

$$\frac{\partial}{\partial w_1} J(w_0, w_1) = \frac{1}{m} \sum_{i=1}^{m} (w_0 + w_1 x_i - y_i) x_i$$

Simulation

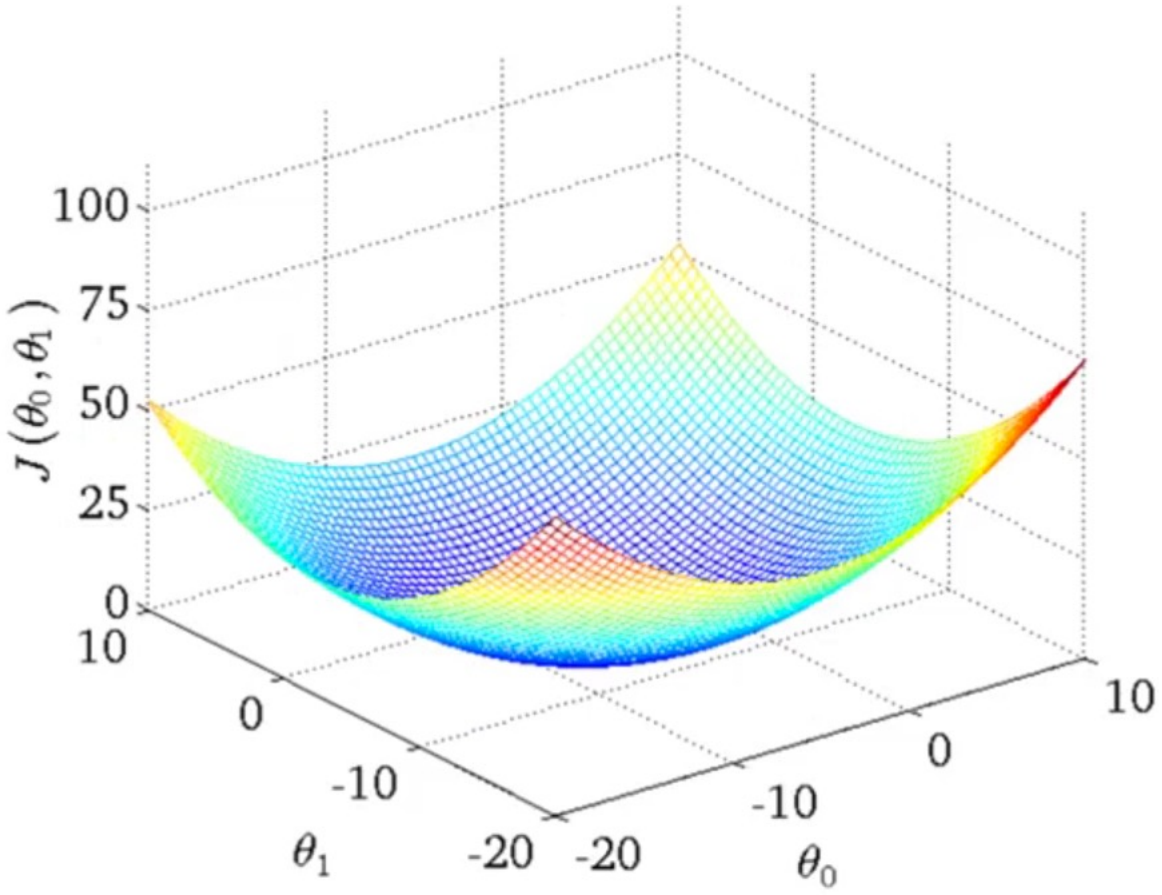
- Data:
  - x = living size (in square feet)
  - y = house prices (in USD)
- Goal:
  - Predict house price when size is given
- Task:
  - Simple linear regression with gradient descent

size	price
820	30105
1050	58448
1550	85911
1200	87967
1600	73722
1117	54630
550	42441
1162	79596

House prices

Notation differs in the following plots:
- θ_j instead of w_j

Simulation: Plot of MSE

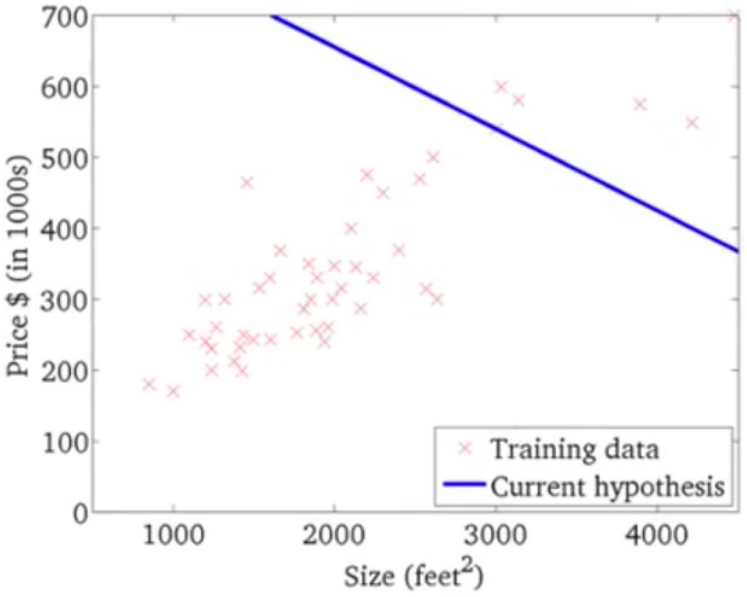


© Andrew Ng: *Machine Learning*, Coursera.

Simulation: Iteration 1

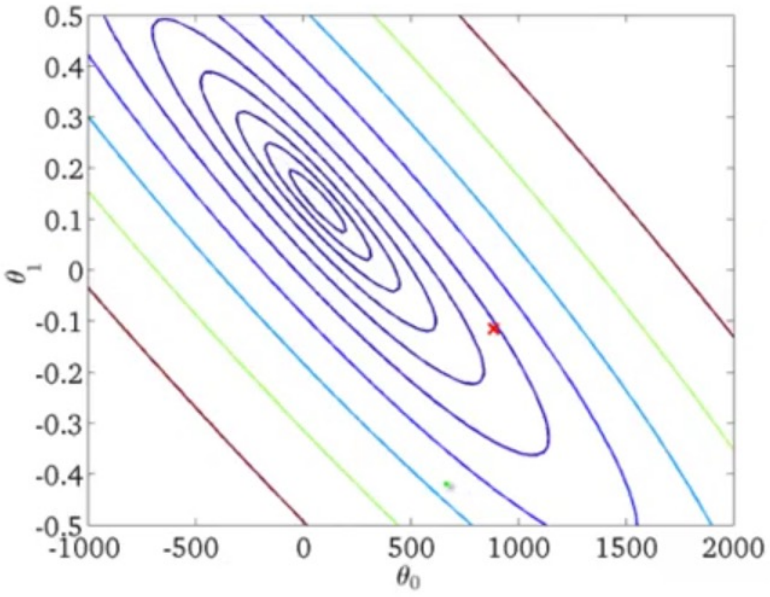
$h_\theta(x)$

(for fixed θ_0, θ_1, this is a function of x)



$J(\theta_0, \theta_1)$

(function of the parameters θ_0, θ_1)

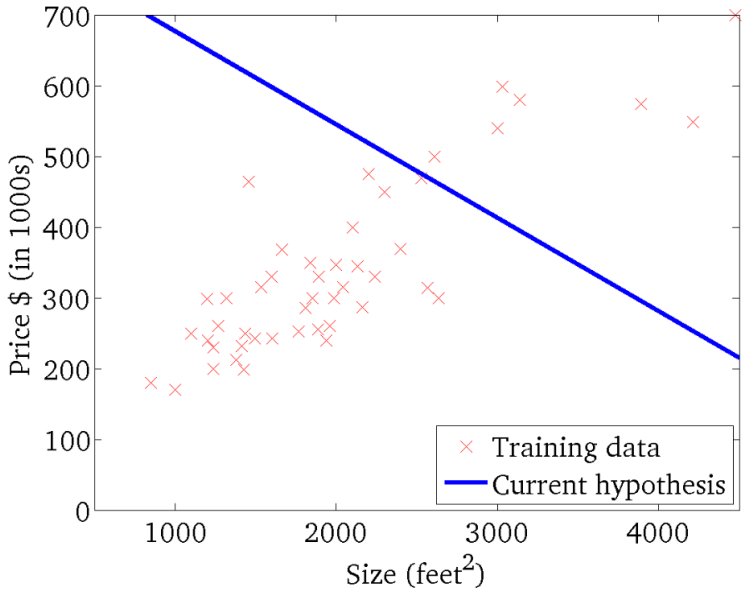


© Andrew Ng: *Machine Learning*, Coursera.

Simulation: Iteration 2

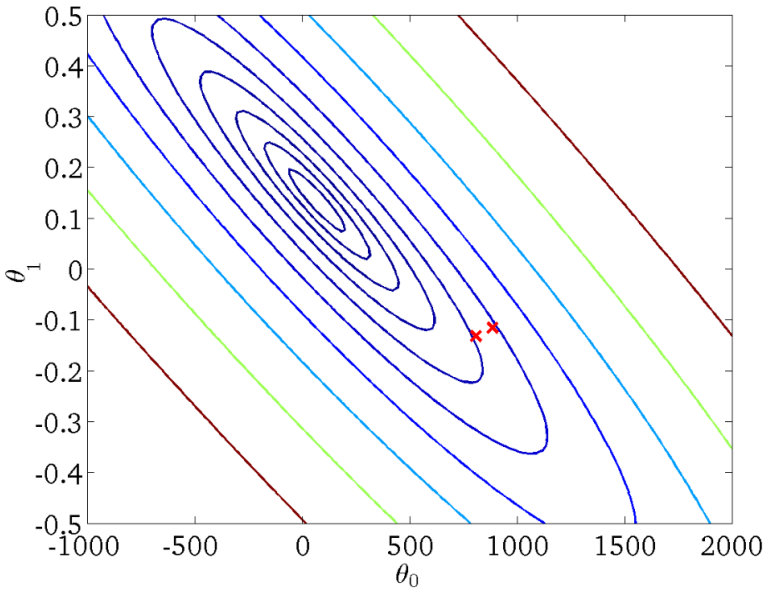
$h_\theta(x)$

(for fixed θ_0, θ_1, this is a function of x)



$J(\theta_0, \theta_1)$

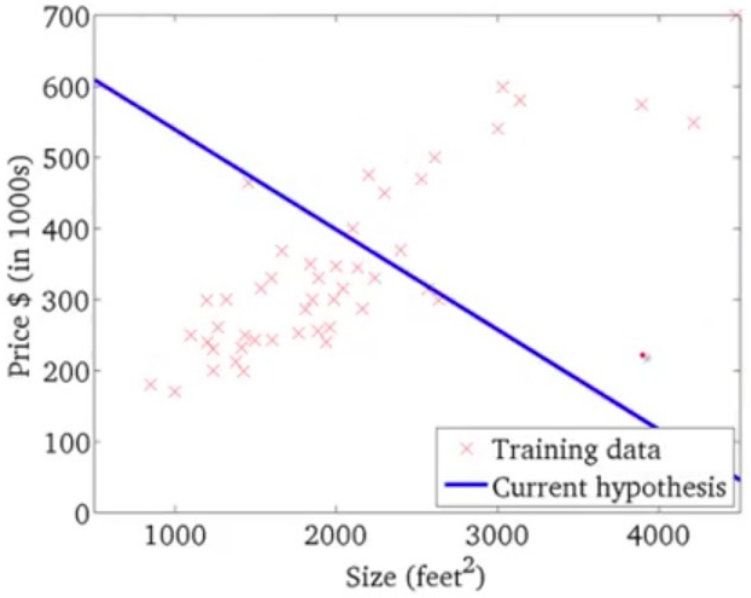
(function of the parameters θ_0, θ_1)



Simulation: Iteration 3

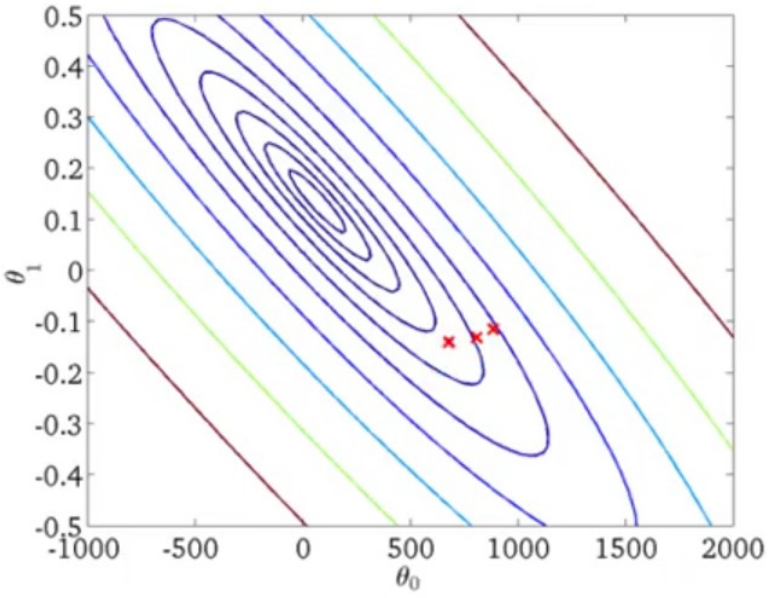
$h_\theta(x)$

(for fixed θ_0, θ_1, this is a function of x)



$J(\theta_0, \theta_1)$

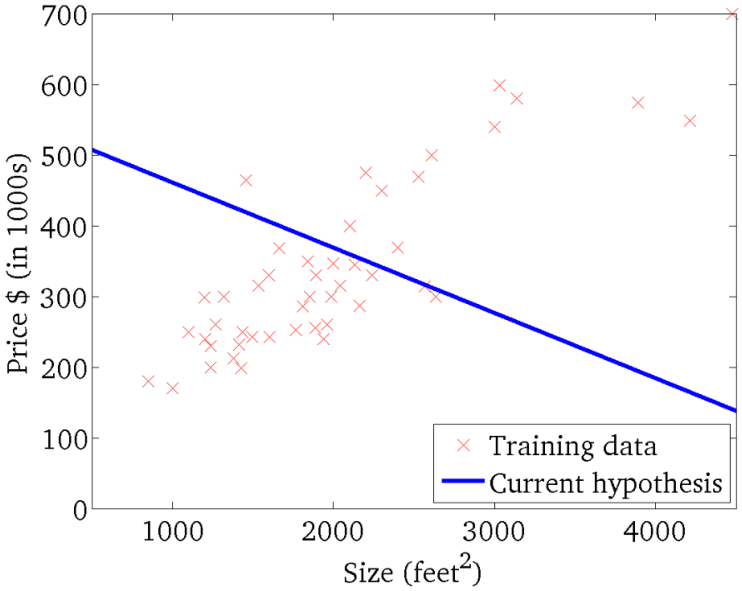
(function of the parameters θ_0, θ_1)



Simulation: Iteration 4

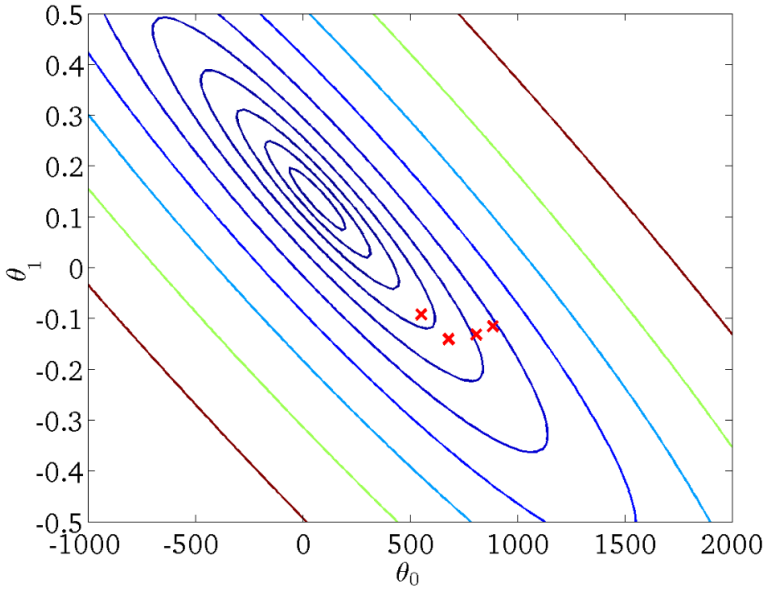
$h_\theta(x)$

(for fixed θ_0, θ_1, this is a function of x)



$J(\theta_0, \theta_1)$

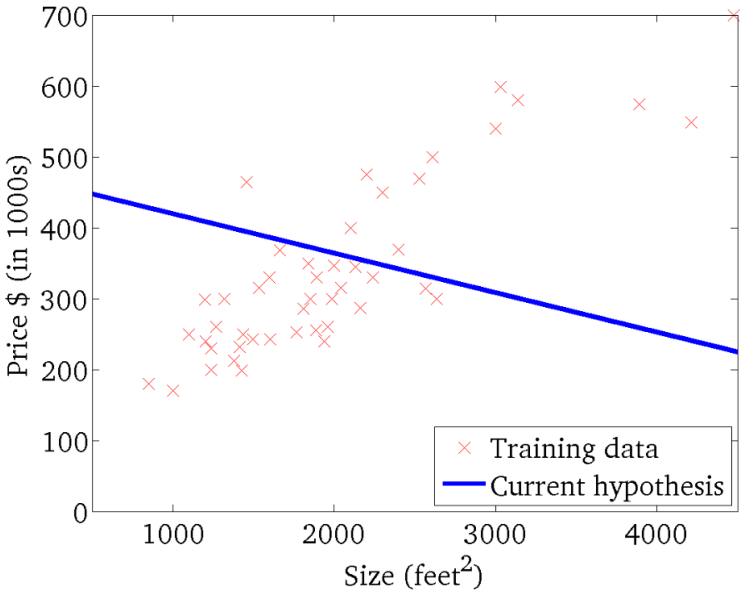
(function of the parameters θ_0, θ_1)



Simulation: Iteration 5

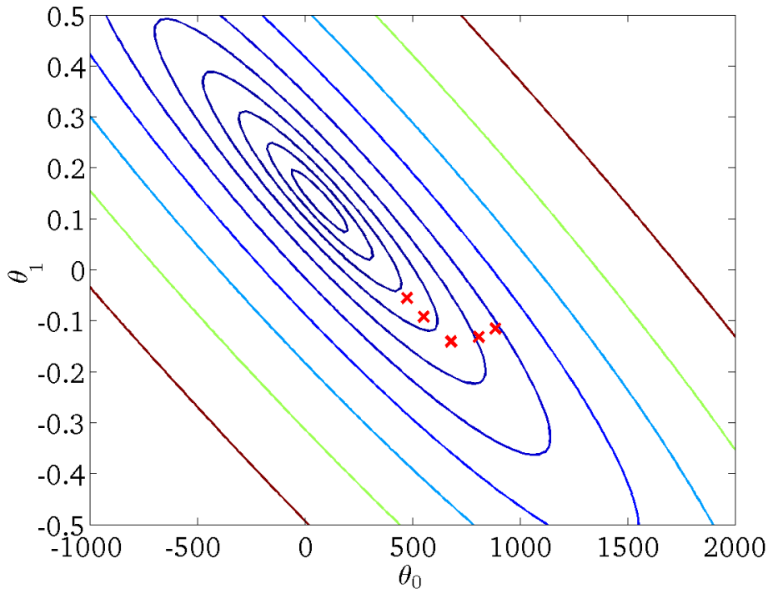
$h_\theta(x)$

(for fixed θ_0, θ_1, this is a function of x)



$J(\theta_0, \theta_1)$

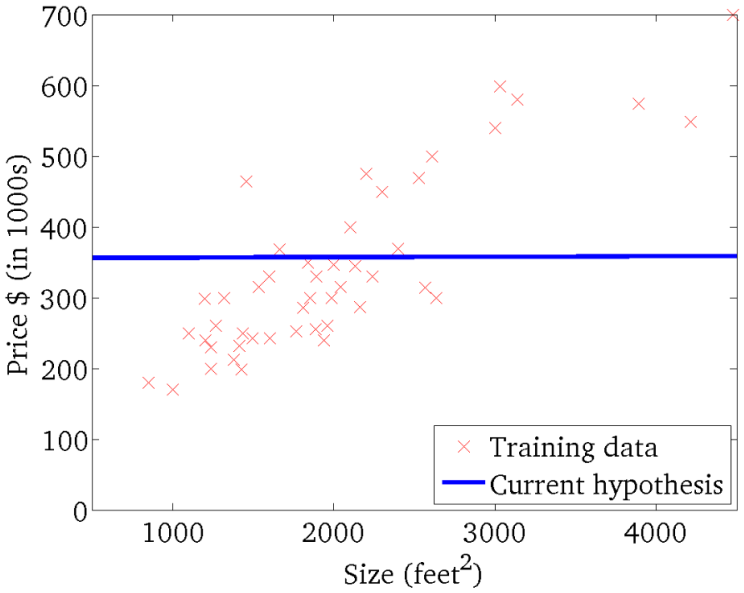
(function of the parameters θ_0, θ_1)



Simulation: Iteration 6

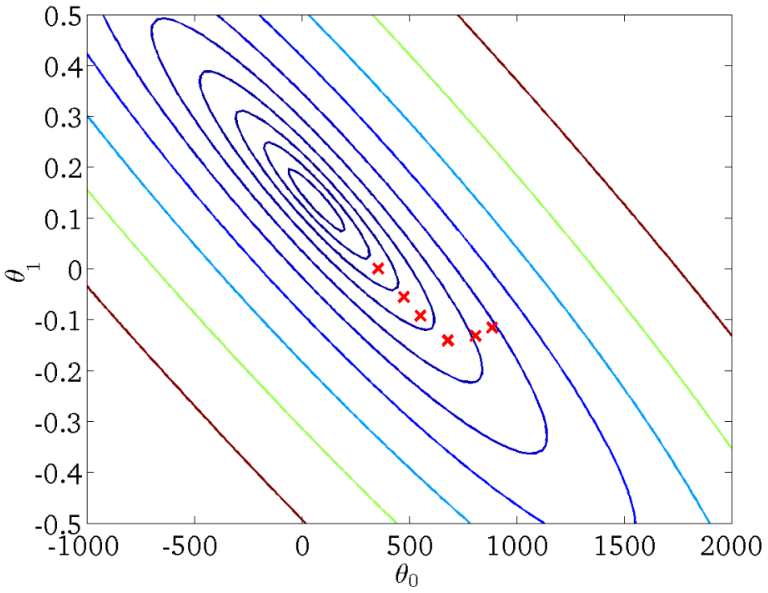
$h_\theta(x)$

(for fixed θ_0, θ_1, this is a function of x)



$J(\theta_0, \theta_1)$

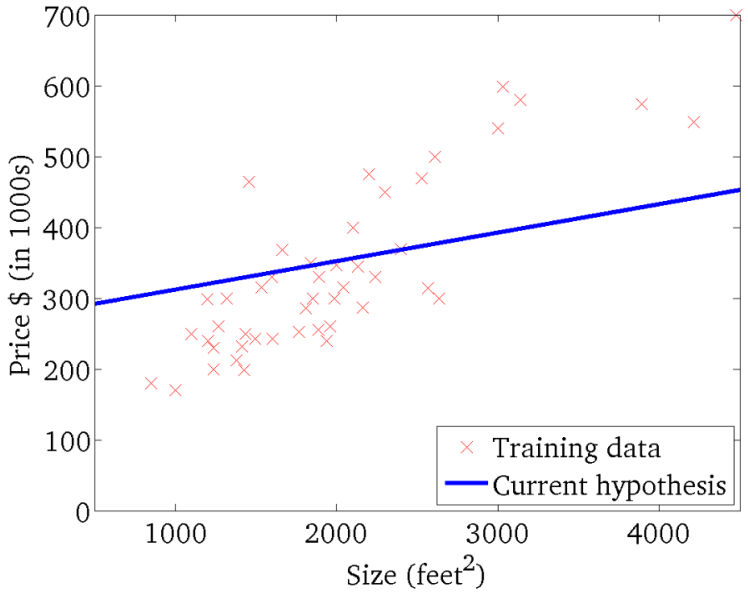
(function of the parameters θ_0, θ_1)



Simulation: Iteration 7

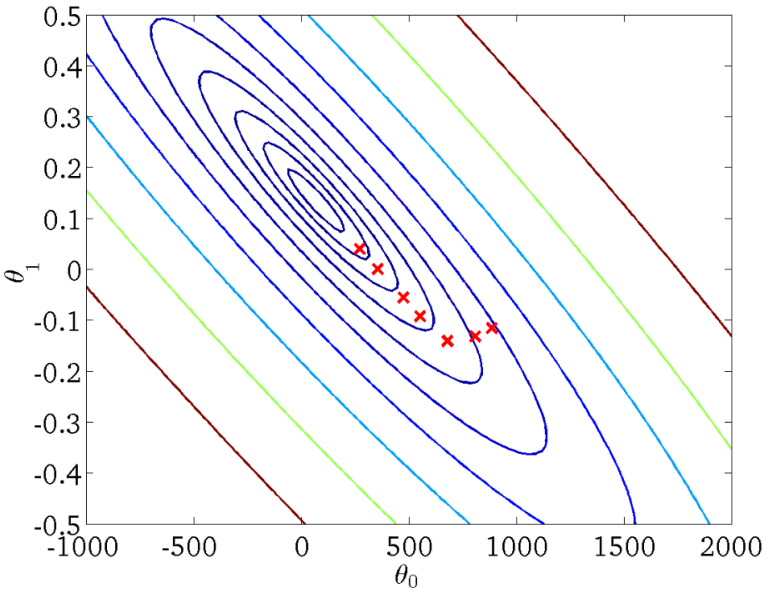
$h_\theta(x)$

(for fixed θ_0, θ_1, this is a function of x)



$J(\theta_0, \theta_1)$

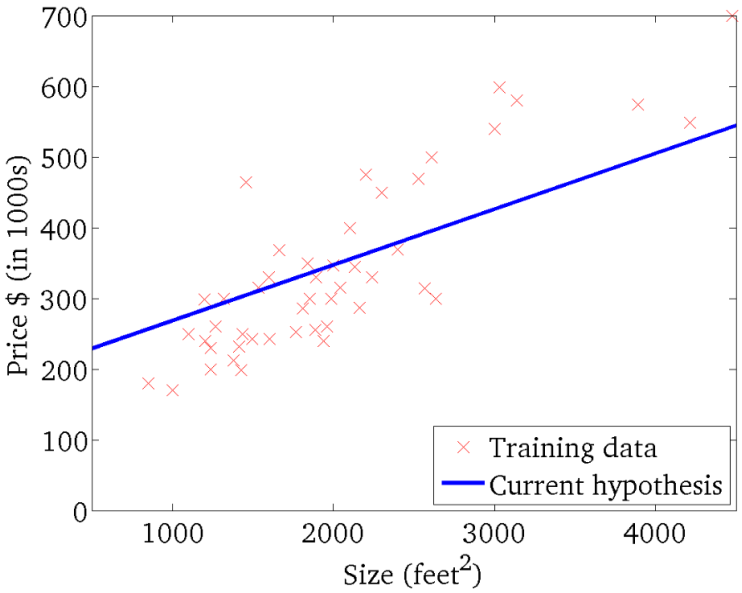
(function of the parameters θ_0, θ_1)



Simulation: Iteration 8

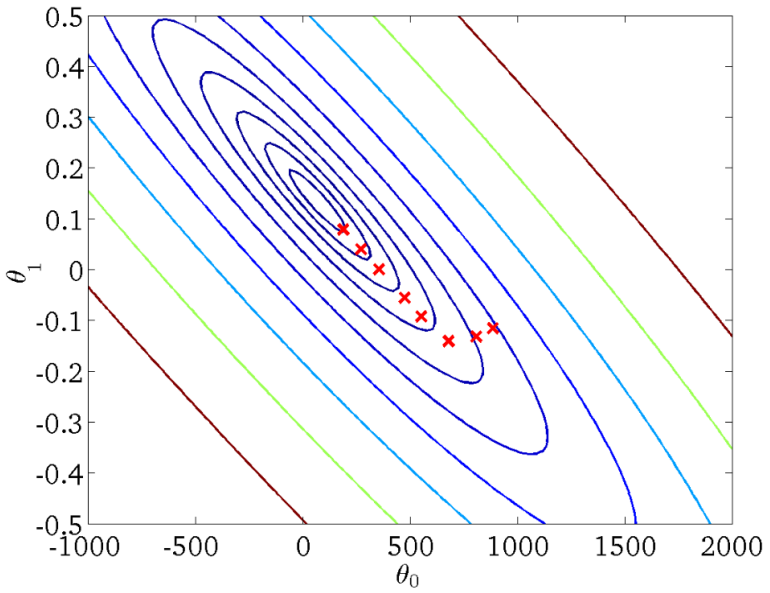
$h_\theta(x)$

(for fixed θ_0, θ_1, this is a function of x)



$J(\theta_0, \theta_1)$

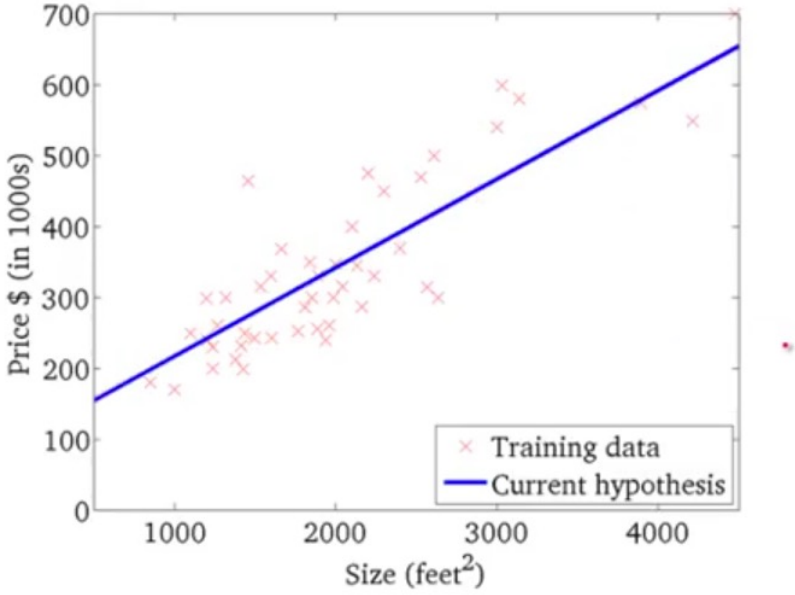
(function of the parameters θ_0, θ_1)



Simulation: Iteration 9

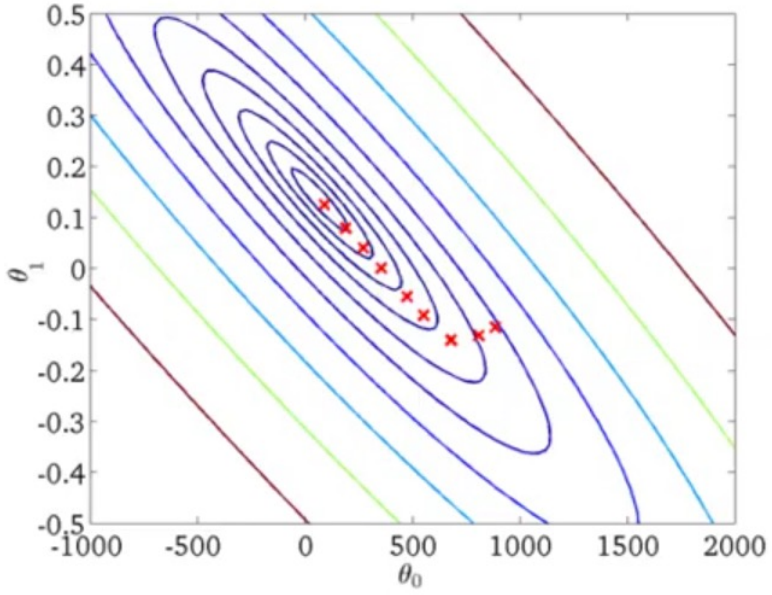
$h_\theta(x)$

(for fixed θ_0, θ_1, this is a function of x)

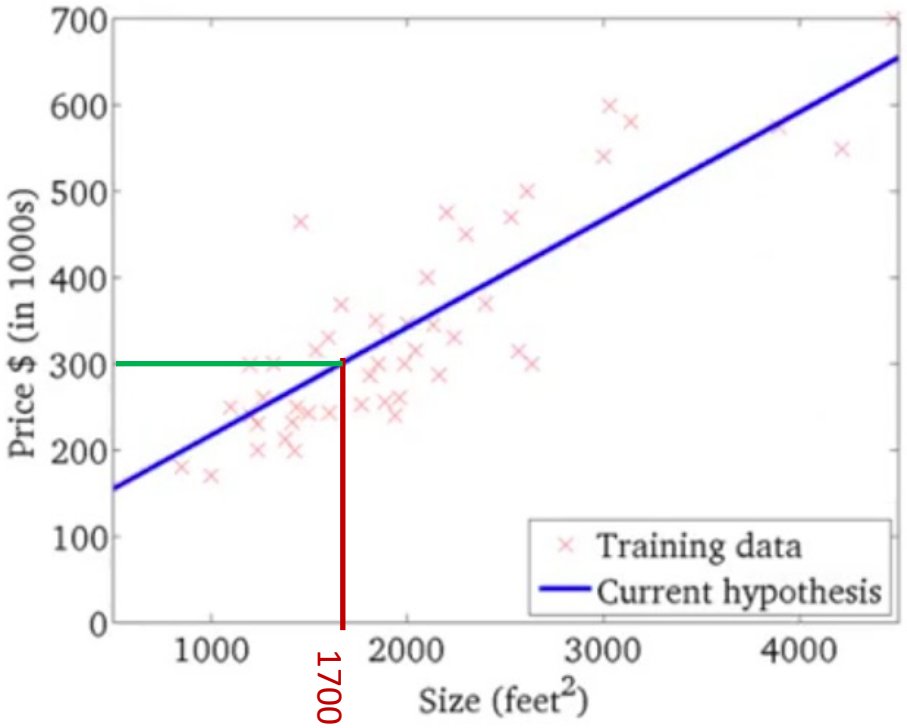


$J(\theta_0, \theta_1)$

(function of the parameters θ_0, θ_1)



Simulation: Prediction



Summary

- Problem of supervised learning
  - given: finite set of input-output examples
  - goal: estimate unknown functional relationship
- Inductive principle
  - empirical risk minimization
- Common technique
  - gradient descent methods

Get in Touch

David Schultz

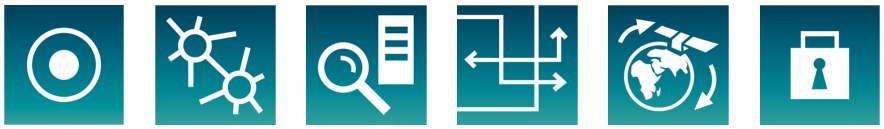
CC AIM

schultz@tu-berlin.de
Fon +49 (0) 30 / 314 - 74159

DAI-Labor
Technische Universität Berlin

Fakultät IV –
Elektrotechnik & Informatik

Sekretariat TEL 14
Ernst Reuter Platz 7
10587 Berlin



www.dai-labor.de