Natural Language Processing
Lecture 02: Machine Learning Basics; Text Classification

Valentin Malykh, MTS AI

Autumn 2024

The course is delivered at ITMO University, Saint-Petersburg, Bauman University, Moscow, IITU, Almaty

Content

Content

Content

What is machine learning?

— Wikipedia definition

- *Machine learning (ML)* is the scientific study of algorithms and statistical models that computer systems use to perform a specific task without using explicit instructions, relying on patterns and inference instead.
- It is seen as a subset of artificial intelligence.
- Machine learning algorithms build a mathematical model based on sample data, known as "training data", in order to make predictions or decisions without being explicitly programmed to perform the task.

- Machine learning algorithms are used in a wide variety of applications, such as email filtering and computer vision, where it is difficult or infeasible to develop a conventional algorithm for effectively performing the task.
- Machine learning is closely related to *computational statistics*, which focuses on making predictions using computers.
- The study of *mathematical optimization* delivers methods, theory and application domains to the field of machine learning.
- *Data mining* is a field of study within machine learning, and focuses on exploratory data analysis through unsupervised learning.
- In its application across business problems, machine learning is also referred to as *predictive analytics*.

Supervised machine learning

- (Supervised) Machine Learning techniques automatically learn a model of the relationship between a set of **descriptive features** and a **target feature** from a set of historical examples.

John Kelleher and Brian Mac Namee and Aoife D'Arcy, Fundamentals of Machine Learning for Predictive Data Analytics

Supervised machine learning

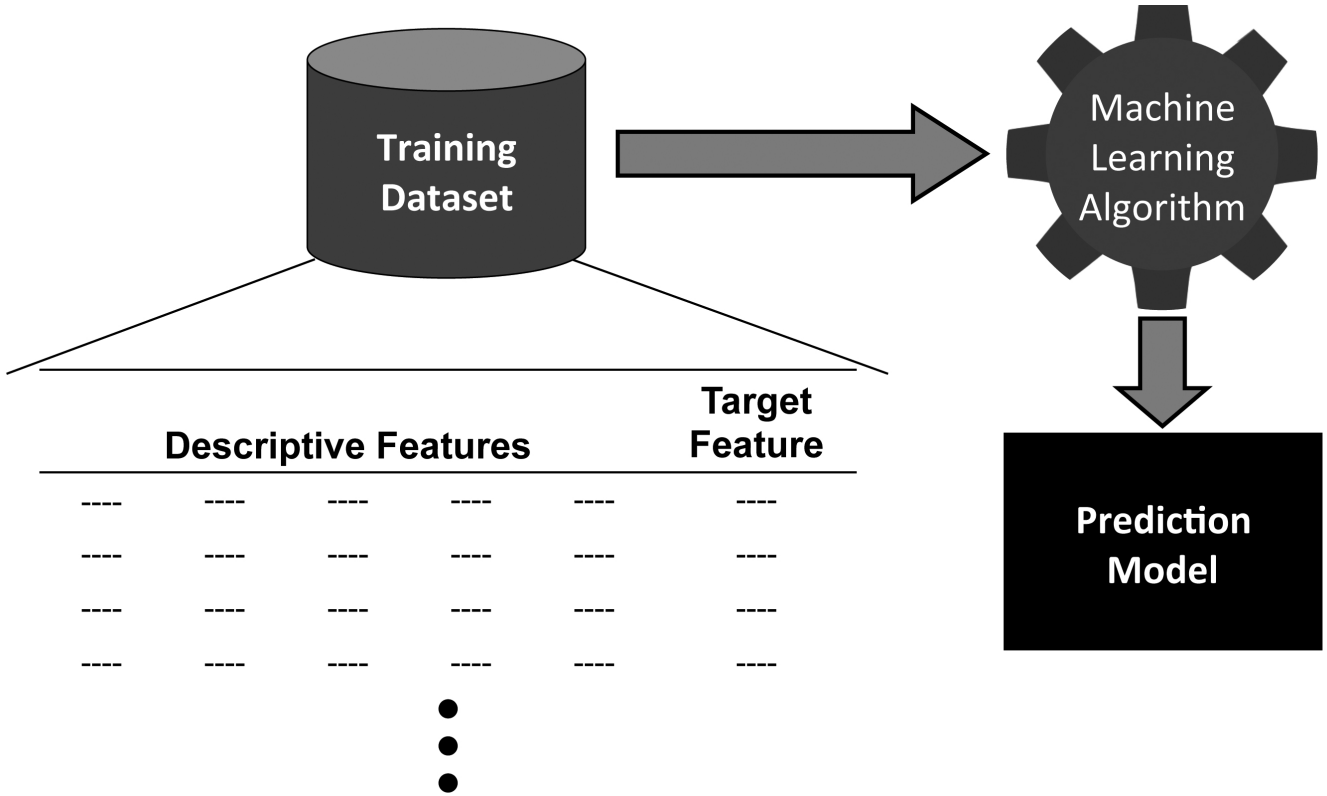


Figure: Using machine learning to induce a prediction model from a training dataset.

John Kelleher and Brian Mac Namee and Aoife D'Arcy, Fundamentals of Machine Learning for Predictive Data Analytics

Supervised machine learning

Figure: Using the model to make predictions for new query instances.

John Kelleher and Brian Mac Namee and Aoife D'Arcy, Fundamentals of Machine Learning for Predictive Data Analytics

ID	Occupation	Age	Loan-Salary Ratio	Outcome
1	industrial	34	2.96	repaid
2	professional	41	4.64	default
3	professional	36	3.22	default
4	professional	41	3.11	default
5	industrial	48	3.80	default
6	industrial	61	2.52	repaid
7	professional	37	1.50	repaid
8	professional	40	1.93	repaid
9	industrial	33	5.25	default
10	industrial	32	4.15	default

Input
Input Features
Descriptive Features
Query Instance

Output
Output Features
Target Features
Prediction

John Kelleher and Brian Mac Namee and Aoife D'Arcy, Fundamentals of Machine Learning for Predictive Data Analytics

Content

ID	Occupation	Age	Loan-Salary Ratio	Outcome
1	industrial	34	2.96	repaid
2	professional	41	4.64	default
3	professional	36	3.22	default
4	professional	41	3.11	default
5	industrial	48	3.80	default
6	industrial	61	2.52	repaid
7	professional	37	1.50	repaid
8	professional	40	1.93	repaid
9	industrial	33	5.25	default
10	industrial	32	4.15	default

- What is the relationship between the **descriptive features** (Occupation, Age, Loan-Salary Ratio) and the **target**

John Kelleher and Brian Mac Namee and Aoife D'Arcy, Fundamentals of Machine Learning for Predictive Data Analytics

Machine learning – an example

if LOAN-SALARY RATIO $>$ 3 **then**
  OUTCOME=*'default'*
else
  OUTCOME=*'repay'*
end if

John Kelleher and Brian Mac Namee and Aoife D'Arcy, Fundamentals of Machine Learning for Predictive Data Analytics

Machine learning – an example

if LOAN-SALARY RATIO $>$ 3 **then**
  OUTCOME=*'default'*
else
  OUTCOME=*'repay'*
end if

- This is an example of a **prediction model**

John Kelleher and Brian Mac Namee and Aoife D'Arcy, Fundamentals of Machine Learning for Predictive Data Analytics

Machine learning – an example

if LOAN-SALARY RATIO $>$ 3 **then**
  OUTCOME=*'default'*
else
  OUTCOME=*'repay'*
end if

- This is an example of a **prediction model**
- This is also an example of a **consistent** prediction model

John Kelleher and Brian Mac Namee and Aoife D'Arcy, Fundamentals of Machine Learning for Predictive Data Analytics

Machine learning – an example

if LOAN-SALARY RATIO $>$ 3 **then**
  OUTCOME=*'default'*
else
  OUTCOME=*'repay'*
end if

- This is an example of a **prediction model**
- This is also an example of a **consistent** prediction model
- Notice that this model does not use all the features and the feature that it uses is a derived feature (in this case a ratio): **feature design** and **feature selection** are two

John Kelleher and Brian Mac Namee and Aoife D'Arcy, Fundamentals of Machine Learning for Predictive Data Analytics

Machine learning – an example

- What is the relationship between the **descriptive features** and the **target feature** (OUTCOME) in the following dataset?

John Kelleher and Brian Mac Namee and Aoife D'Arcy, Fundamentals of Machine Learning for Predictive Data Analytics

ID	Amount	Salary	Loan-Salary Ratio	Age	Occupation	House	Type	Outcome
1	245,100	66,400	3.69	44	industrial	farm	stb	repaid
2	90,600	75,300	1.2	41	industrial	farm	stb	repaid
3	195,600	52,100	3.75	37	industrial	farm	ftb	default
4	157,800	67,600	2.33	44	industrial	apartment	ftb	repaid
5	150,800	35,800	4.21	39	professional	apartment	stb	default
6	133,000	45,300	2.94	29	industrial	farm	ftb	default
7	193,100	73,200	2.64	38	professional	house	ftb	repaid
8	215,000	77,600	2.77	17	professional	farm	ftb	repaid
9	83,000	62,500	1.33	30	professional	house	ftb	repaid
10	186,100	49,200	3.78	30	industrial	house	ftb	default
11	161,500	53,300	3.03	28	professional	apartment	stb	repaid
12	157,400	63,900	2.46	30	professional	farm	stb	repaid
13	210,000	54,200	3.87	43	professional	apartment	ftb	repaid
14	209,700	53,000	3.96	39	industrial	farm	ftb	default
15	143,200	65,300	2.19	32	industrial	apartment	ftb	default
16	203,000	64,400	3.15	44	industrial	farm	ftb	repaid
17	247,800	63,800	3.88	46	industrial	house	stb	repaid
18	162,700	77,400	2.1	37	professional	house	ftb	repaid
19	213,300	61,100	3.49	21	industrial	apartment	ftb	default
20	284,100	32,300	8.8	51	industrial	farm	ftb	default
21	154,000	48,900	3.15	49	professional	house	stb	repaid
22	112,800	79,700	1.42	41	professional	house	ftb	repaid
23	252,000	59,700	4.22	27	professional	house	stb	default
24	175,200	39,900	4.39	37	professional	apartment	stb	default
25	149,700	58,600	2.55	35	industrial	farm	stb	default

John Kelleher and Brian Mac Namee and Aoife D'Arcy, Fundamentals of Machine Learning for Predictive Data Analytics

Machine learning – an example

if LOAN-SALARY RATIO $<$ 1.5 **then**
  OUTCOME=*'repay'*
else if LOAN-SALARY RATIO $>$ 4 **then**
  OUTCOME=*'default'*
else if AGE $<$ 40 **and** OCCUPATION $=$*'industrial'* **then**
  OUTCOME=*'default'*
else
  OUTCOME=*'repay'*
end if

John Kelleher and Brian Mac Namee and Aoife D'Arcy, Fundamentals of Machine Learning for Predictive Data Analytics

Machine learning – an example

if LOAN-SALARY RATIO $<$ 1.5 **then**
  OUTCOME=*'repay'*
else if LOAN-SALARY RATIO $>$ 4 **then**
  OUTCOME=*'default'*
else if AGE $<$ 40 **and** OCCUPATION $=$*'industrial'* **then**
  OUTCOME=*'default'*
else
  OUTCOME=*'repay'*
end if

- The real value of machine learning becomes apparent in situations like this when we want to build prediction models

John Kelleher and Brian Mac Namee and Aoife D'Arcy, Fundamentals of Machine Learning for Predictive Data Analytics

Content

Model spaces and inductive bias

- Machine learning algorithms work by searching through a set of possible prediction models for the model that best captures the relationship between the descriptive features and the target feature.

John Kelleher and Brian Mac Namee and Aoife D'Arcy, Fundamentals of Machine Learning for Predictive Data Analytics

Model spaces and inductive bias

- Machine learning algorithms work by searching through a set of possible prediction models for the model that best captures the relationship between the descriptive features and the target feature.
- An obvious search criteria to drive this search is to look for models that are consistent with the data.

John Kelleher and Brian Mac Namee and Aoife D'Arcy, Fundamentals of Machine Learning for Predictive Data Analytics

Model spaces and inductive bias

- Machine learning algorithms work by searching through a set of possible prediction models for the model that best captures the relationship between the descriptive features and the target feature.
- An obvious search criteria to drive this search is to look for models that are consistent with the data.
- However, because a training dataset is only a sample ML is an ill-posed problem.

John Kelleher and Brian Mac Namee and Aoife D'Arcy, Fundamentals of Machine Learning for Predictive Data Analytics

Model spaces and inductive bias

Table: A simple retail dataset

ID	BBY	ALC	ORG	GRP
1	no	no	no	couple
2	yes	no	yes	family
3	yes	yes	no	family
4	no	no	yes	couple
5	no	yes	yes	single

John Kelleher and Brian Mac Namee and Aoife D'Arcy, Fundamentals of Machine Learning for Predictive Data Analytics

Model spaces and inductive bias

Table: A full set of potential prediction models before any training data becomes available.

BBY	ALC	ORG	GRP	$\mathbb{M}_1$	$\mathbb{M}_2$	$\mathbb{M}_3$	$\mathbb{M}_4$	$\mathbb{M}_5$	...	$\mathbb{M}_{6\,561}$
no	no	no	?	couple	couple	single	couple	couple		couple
no	no	yes	?	single	couple	single	couple	couple		single
no	yes	no	?	family	family	single	single	single		family
no	yes	yes	?	single	single	single	single	single		couple
yes	no	no	?	couple	couple	family	family	family	...	family
yes	no	yes	?	couple	family	family	family	family		couple
yes	yes	no	?	single	family	family	family	family		single
yes	yes	yes	?	single	single	family	family	couple		family

John Kelleher and Brian Mac Namee and Aoife D'Arcy, Fundamentals of Machine Learning for Predictive Data Analytics

Model spaces and inductive bias

Table: A sample of the models that are consistent with the training data

BBY	ALC	ORG	GRP	$\mathbb{M}_1$	$\mathbb{M}_2$	$\mathbb{M}_3$	$\mathbb{M}_4$	$\mathbb{M}_5$	...	$\mathbb{M}_{6\,561}$
no	no	no	couple	couple	couple	single	couple	couple		couple
no	no	yes	couple	single	couple	single	couple	couple		single
no	yes	no	?	family	family	single	single	single		family
no	yes	yes	single	single	single	single	single	single		couple
yes	no	no	?	couple	couple	family	family	family	...	family
yes	no	yes	family	couple	family	family	family	family		couple
yes	yes	no	family	single	family	family	family	family		single
yes	yes	yes	?	single	single	family	family	couple		family

John Kelleher and Brian Mac Namee and Aoife D'Arcy, Fundamentals of Machine Learning for Predictive Data Analytics

Model spaces and inductive bias

Table: A sample of the models that are consistent with the training data

BBY	ALC	ORG	GRP	$\mathbb{M}_1$	$\mathbb{M}_2$	$\mathbb{M}_3$	$\mathbb{M}_4$	$\mathbb{M}_5$	...	$\mathbb{M}_{6\,561}$
no	no	no	couple	couple	couple	single	couple	couple		couple
no	no	yes	couple	single	couple	single	couple	couple		single
no	yes	no	?	family	family	single	single	single		family
no	yes	yes	single	single	single	single	single	single		couple
yes	no	no	?	couple	couple	family	family	family	...	family
yes	no	yes	family	couple	family	family	family	family		couple
yes	yes	no	family	single	family	family	family	family		single
yes	yes	yes	?	single	single	family	family	couple		family

- Notice that there is more than one candidate model left! It is because a single consistent model cannot be found

John Kelleher and Brian Mac Namee and Aoife D'Arcy, Fundamentals of Machine Learning for Predictive Data Analytics

Model spaces and inductive bias

- Consistency $\approx$ memorizing the dataset.
- Consistency with noise in the data isn't desirable.
- Goal: a model that **generalises** beyond the dataset and that isn't influenced by the noise in the dataset.
- So what criteria should we use for choosing between models?

John Kelleher and Brian Mac Namee and Aoife D'Arcy, Fundamentals of Machine Learning for Predictive Data Analytics

Model spaces and inductive bias

- Inductive bias the set of assumptions that define the model selection criteria of an ML algorithm.
- There are two types of bias that we can use:
  1. restriction bias
  2. preference bias
- Inductive bias is necessary for learning (beyond the dataset).

John Kelleher and Brian Mac Namee and Aoife D'Arcy, Fundamentals of Machine Learning for Predictive Data Analytics

Content

Classification

Table: A simple retail dataset

ID	BBY	ALC	ORG	GRP
1	no	no	no	couple
2	yes	no	yes	family
3	yes	yes	no	family
4	no	no	yes	couple
5	no	yes	yes	single

To predict a target feature with categorical values.

John Kelleher and Brian Mac Namee and Aoife D'Arcy, Fundamentals of Machine Learning for Predictive Data Analytics

Regression

Table: The age-income dataset.

ID	AGE	INCOME
1	21	24,000
2	32	48,000
3	62	83,000
4	72	61,000
5	84	52,000

To predict a target feature with numerical values.

John Kelleher and Brian Mac Namee and Aoife D'Arcy, Fundamentals of Machine Learning for Predictive Data Analytics

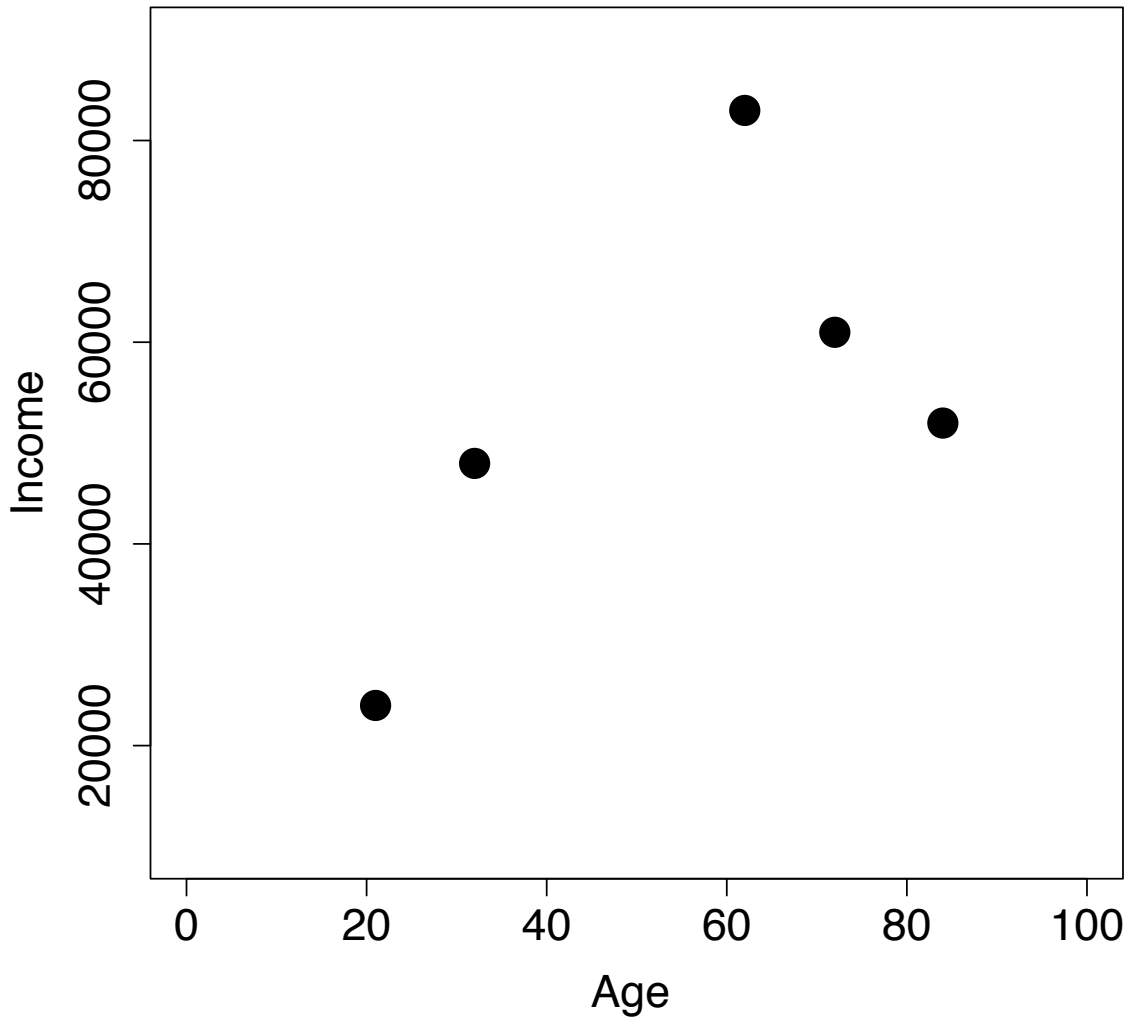
Content

Overfitting and underfitting

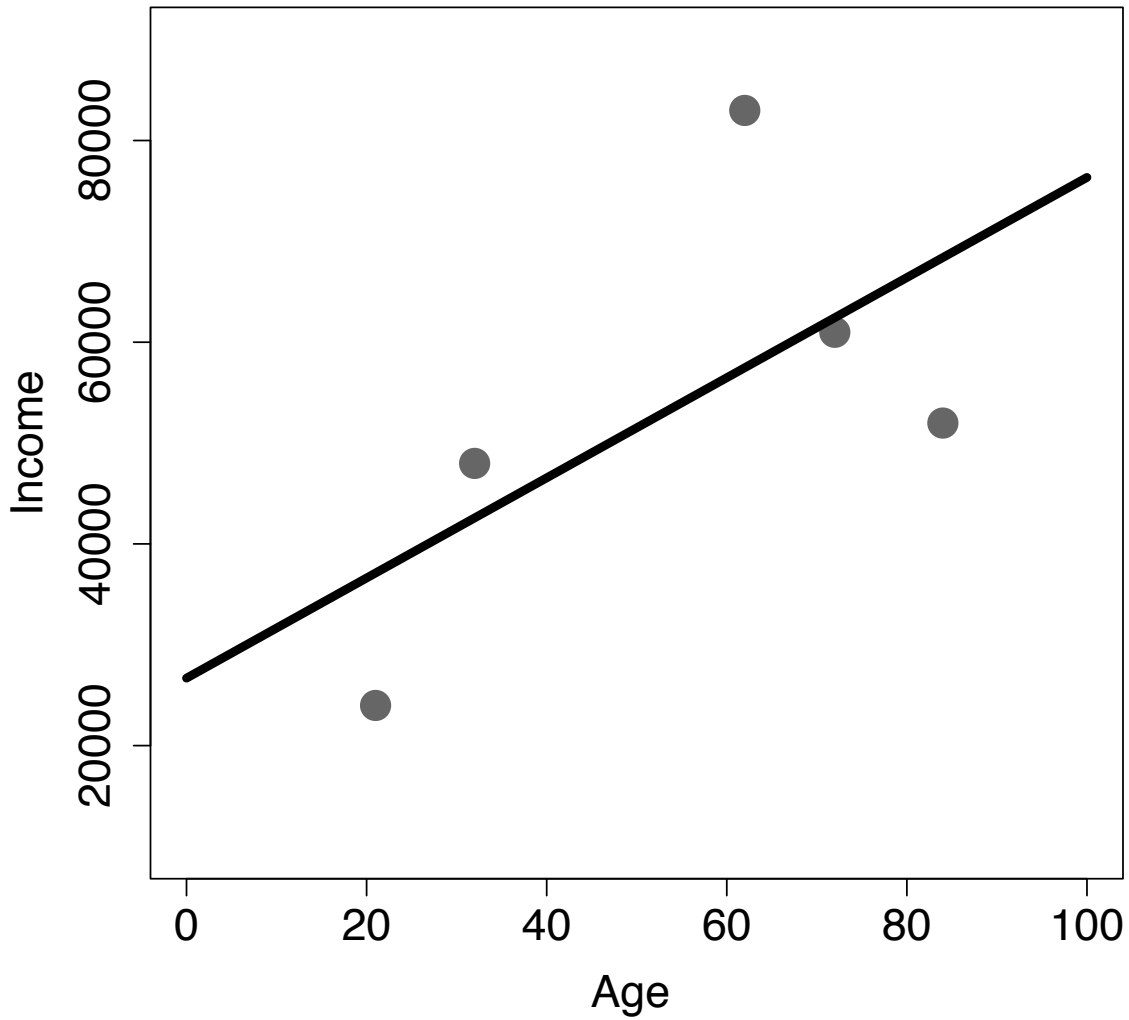
Table: The age-income dataset.

ID	AGE	INCOME
1	21	24,000
2	32	48,000
3	62	83,000
4	72	61,000
5	84	52,000

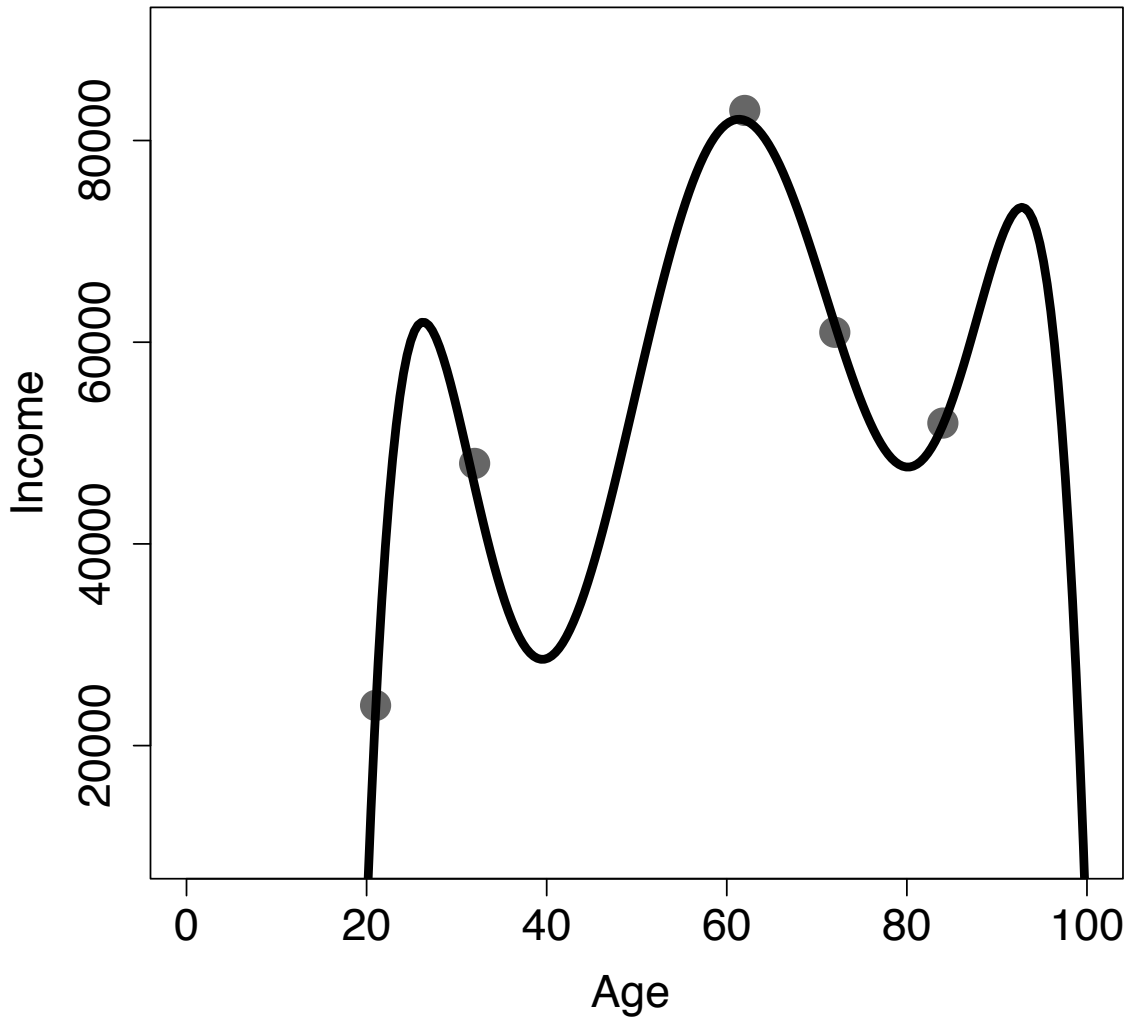
John Kelleher and Brian Mac Namee and Aoife D'Arcy, Fundamentals of Machine Learning for Predictive Data Analytics



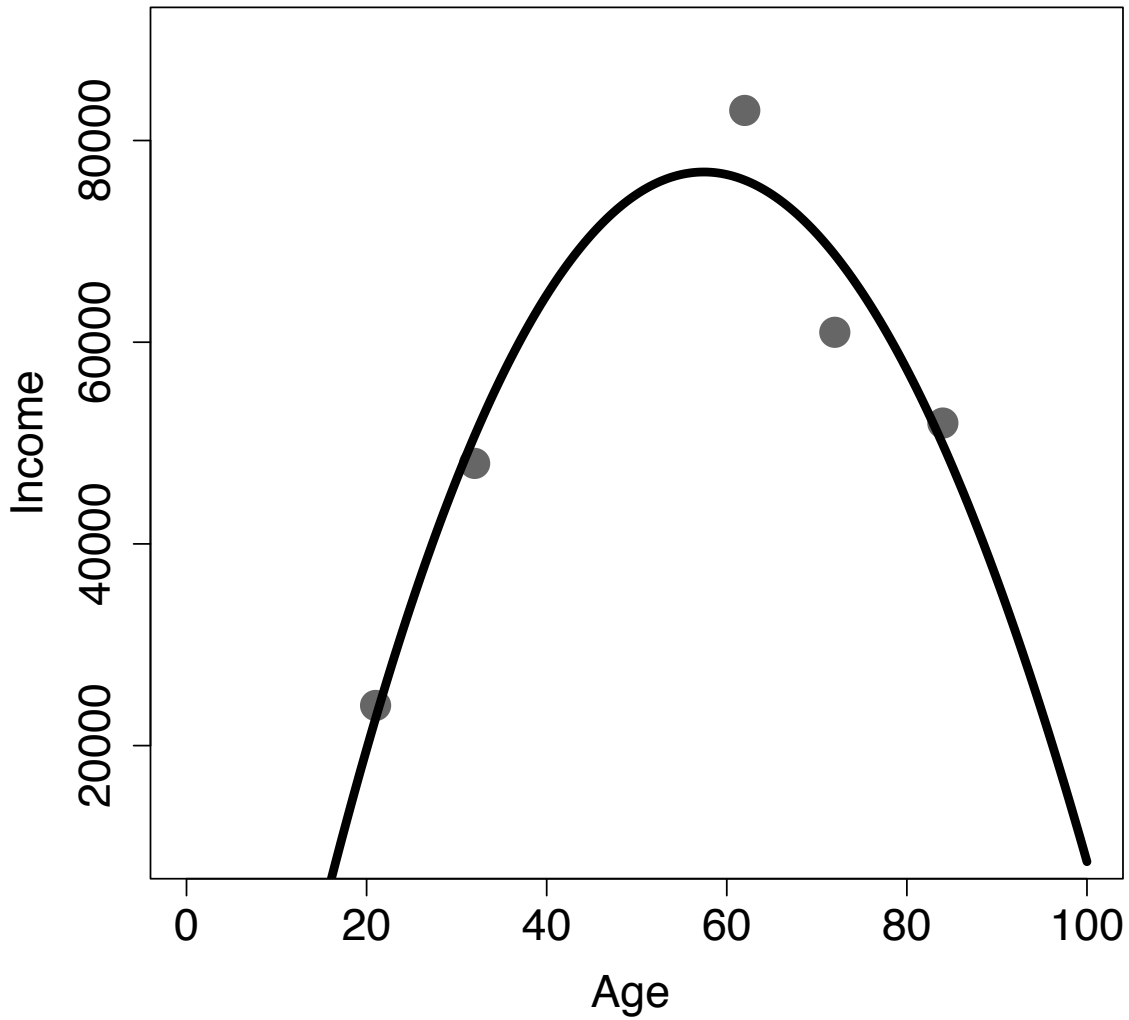
John Kelleher and Brian Mac Namee and Aoife D'Arcy, Fundamentals of Machine Learning for Predictive Data Analytics



John Kelleher and Brian Mac Namee and Aoife D'Arcy, Fundamentals of Machine Learning for Predictive Data Analytics

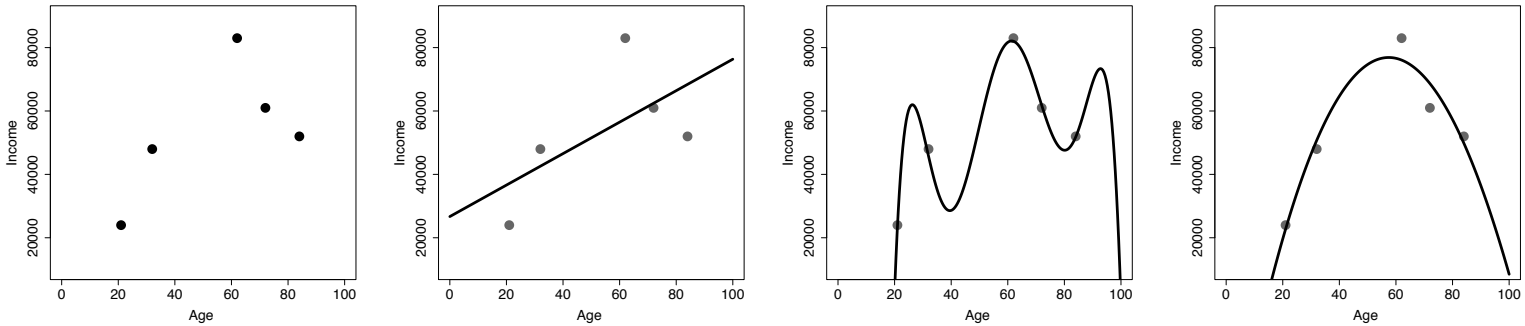


John Kelleher and Brian Mac Namee and Aoife D'Arcy, Fundamentals of Machine Learning for Predictive Data Analytics



John Kelleher and Brian Mac Namee and Aoife D'Arcy, Fundamentals of Machine Learning for Predictive Data Analytics

Overfitting and underfitting



(a) Dataset (b) Underfitting (c) Overfitting (d) Just right

Figure: Striking a balance between overfitting and underfitting when trying to predict age from income.

John Kelleher and Brian Mac Namee and Aoife D'Arcy, Fundamentals of Machine Learning for Predictive Data Analytics

Content

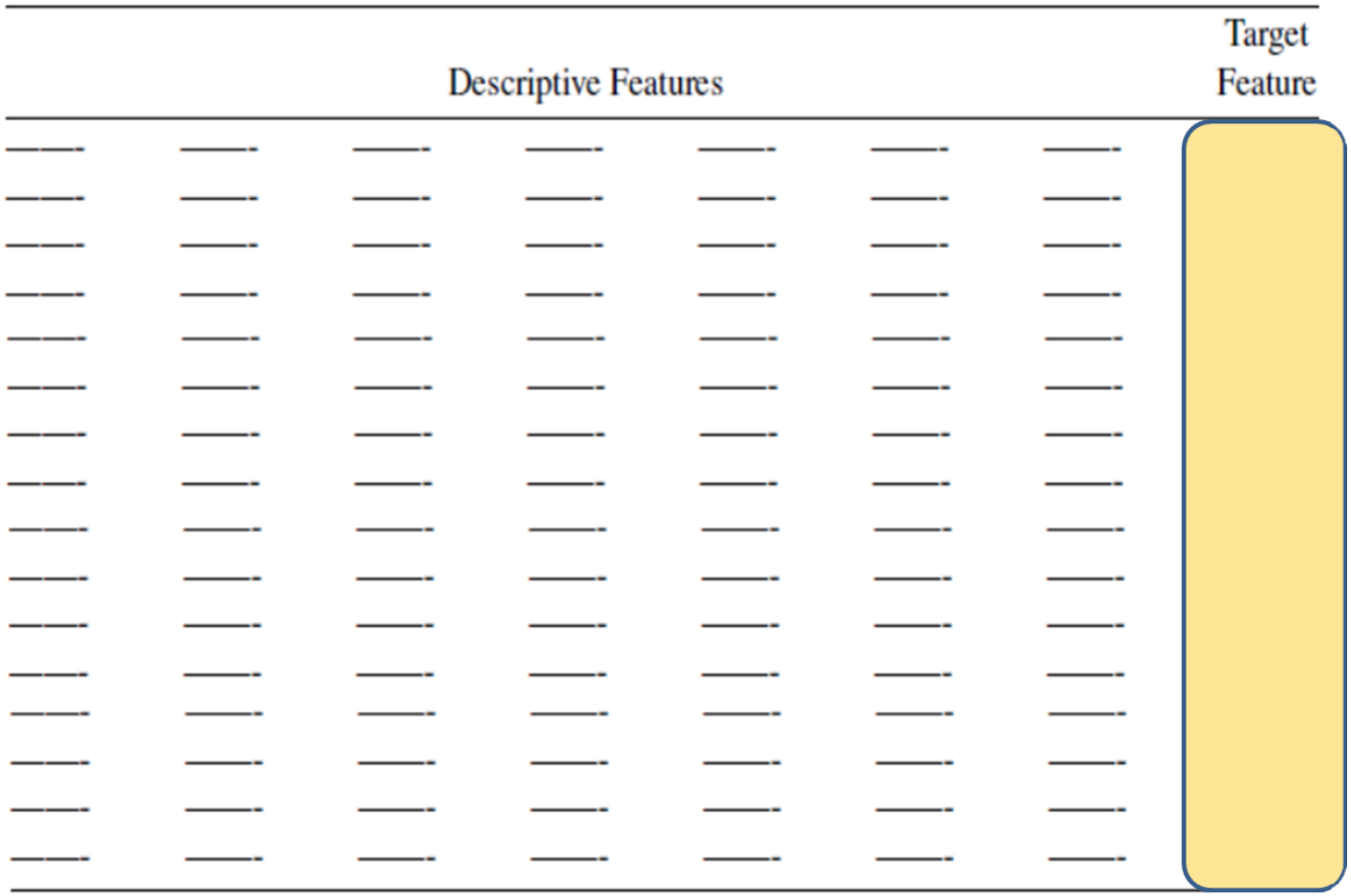
Unsupervised learning

- *Unsupervised learning* is the *machine learning* task of inferring a function to describe hidden structure from unlabeled data.
- Since the examples given to the learner are unlabeled, there is no error or reward signal to evaluate a potential solution.
- This distinguishes unsupervised learning from supervised learning.

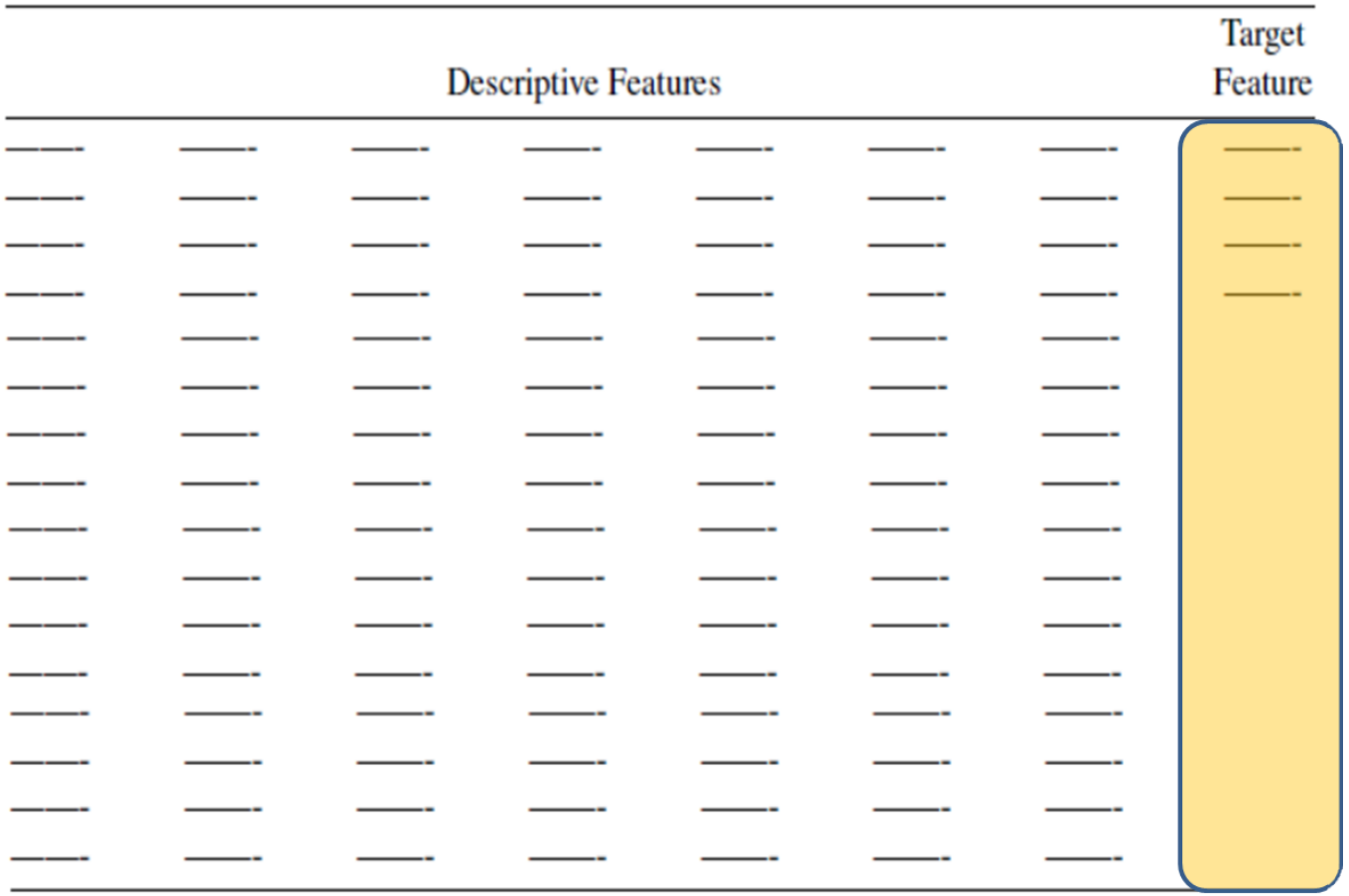
Supervised learning

Descriptive Features							Target Feature

Unsupervised learning



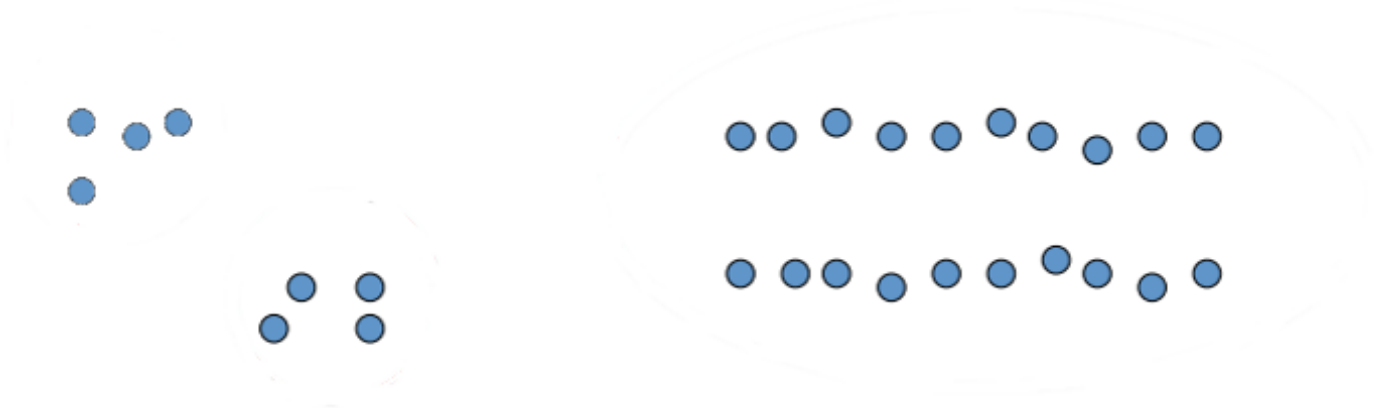
Semi-supervised learning



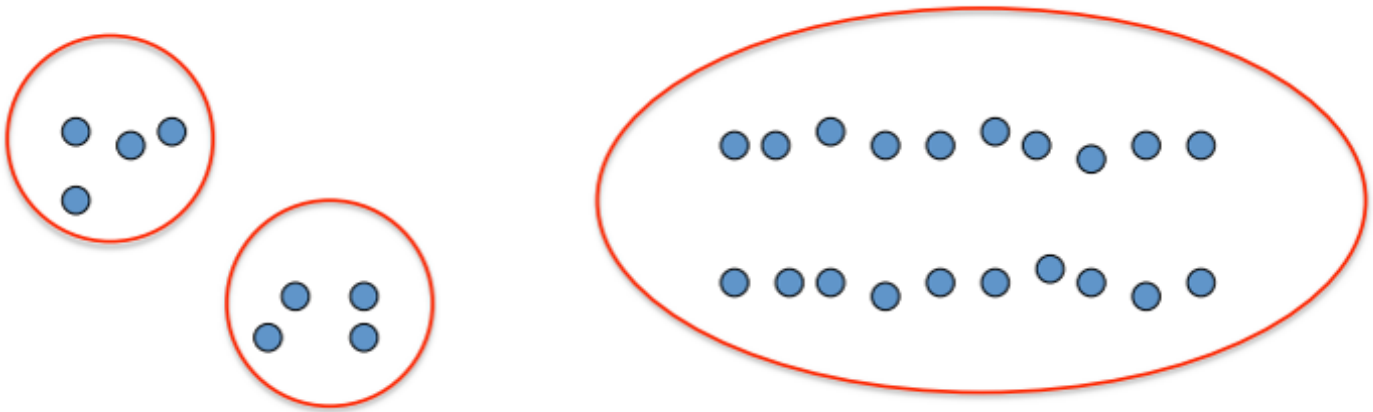
Clustering

- *Cluster analysis* or *clustering* is the task of grouping a set of objects in such a way that objects in the same group (called a cluster) are more similar (in some sense or another) to each other than to those in other groups (clusters).
- Clustering is a typical unsupervised learning task.

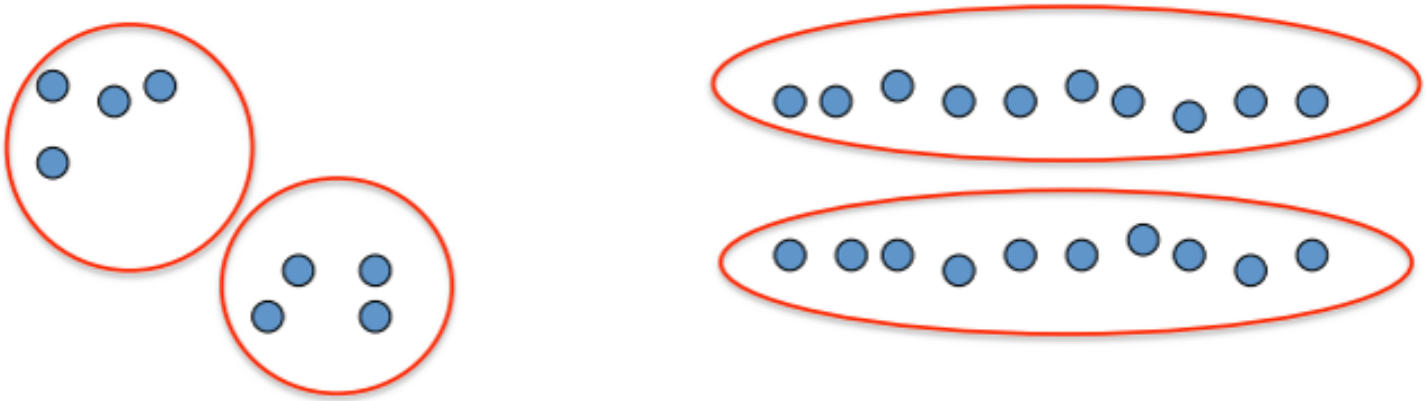
Clustering – An example



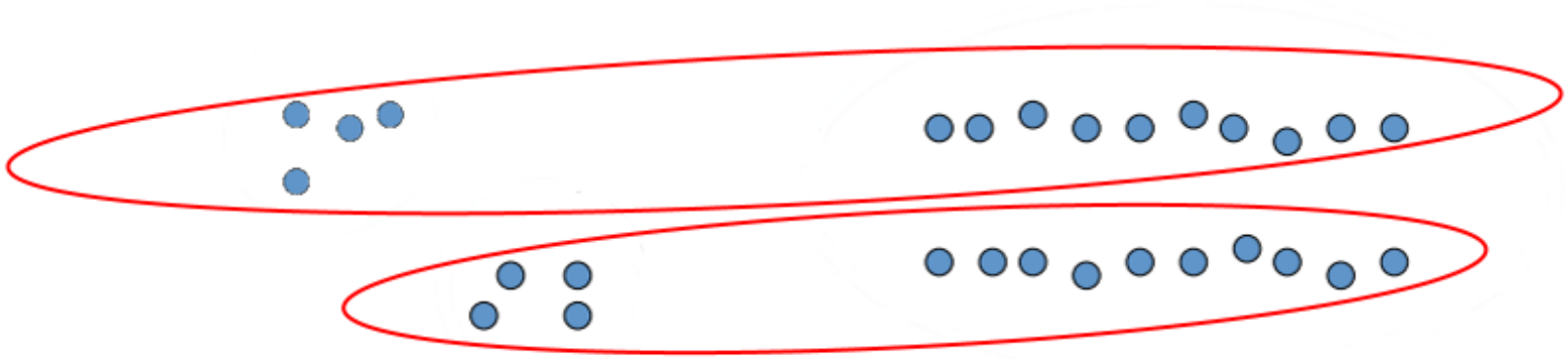
Clustering – An example



Clustering – An example



Clustering – An example



Content

Content

Classification - an example

A power generator

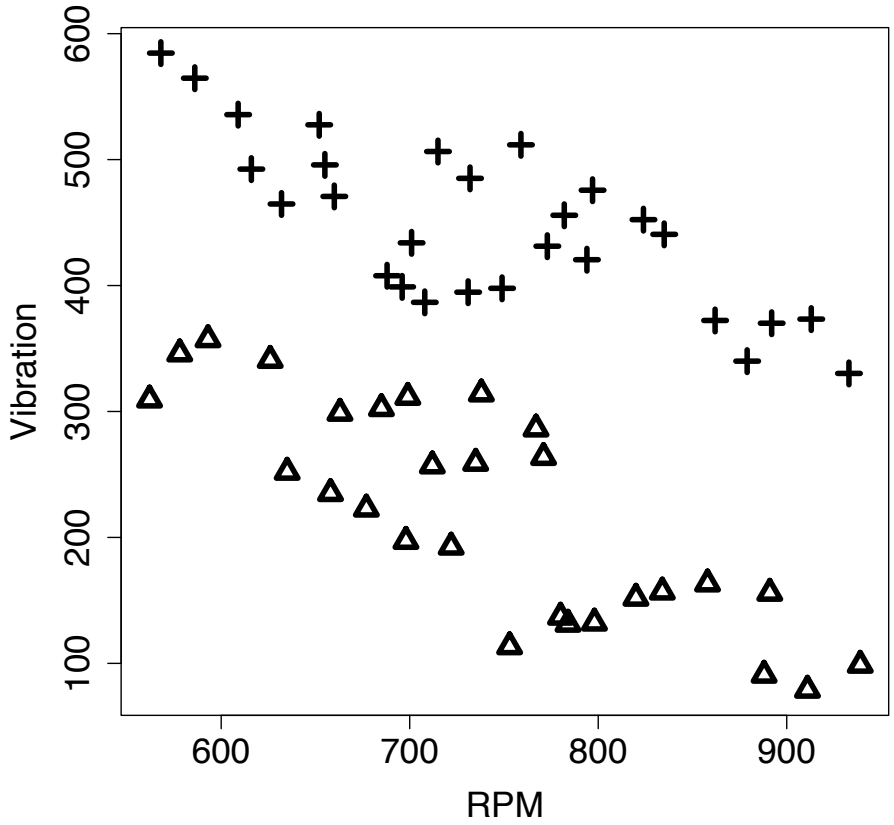
Table: A dataset listing features for a number of generators.

ID	RPM	VIBRATION	STATUS
1	568	585	good
2	586	565	good
3	609	536	good
4	616	492	good
5	632	465	good
6	652	528	good
7	655	496	good
8	660	471	good
9	688	408	good
10	696	399	good
11	708	387	good
12	701	434	good
13	715	506	good
14	732	485	good
15	731	395	good
16	749	398	good
17	759	512	good
18	773	431	good
19	782	456	good
20	797	476	good
21	794	421	good
22	824	452	good
23	835	441	good
24	862	372	good
25	879	340	good
26	892	370	good
27	913	373	good
28	933	330	good

ID	RPM	VIBRATION	STATUS
29	562	309	faulty
30	578	346	faulty
31	593	357	faulty
32	626	341	faulty
33	635	252	faulty
34	658	235	faulty
35	663	299	faulty
36	677	223	faulty
37	685	303	faulty
38	698	197	faulty
39	699	311	faulty
40	712	257	faulty
41	722	193	faulty
42	735	259	faulty
43	738	314	faulty
44	753	113	faulty
45	767	286	faulty
46	771	264	faulty
47	780	137	faulty
48	784	131	faulty
49	798	132	faulty
50	820	152	faulty
51	834	157	faulty
52	858	163	faulty
53	888	91	faulty
54	891	156	faulty
55	911	79	faulty
56	939	99	faulty

John Kelleher and Brian Mac Namee and Aoife D'Arcy, Fundamentals of Machine Learning for Predictive Data Analytics

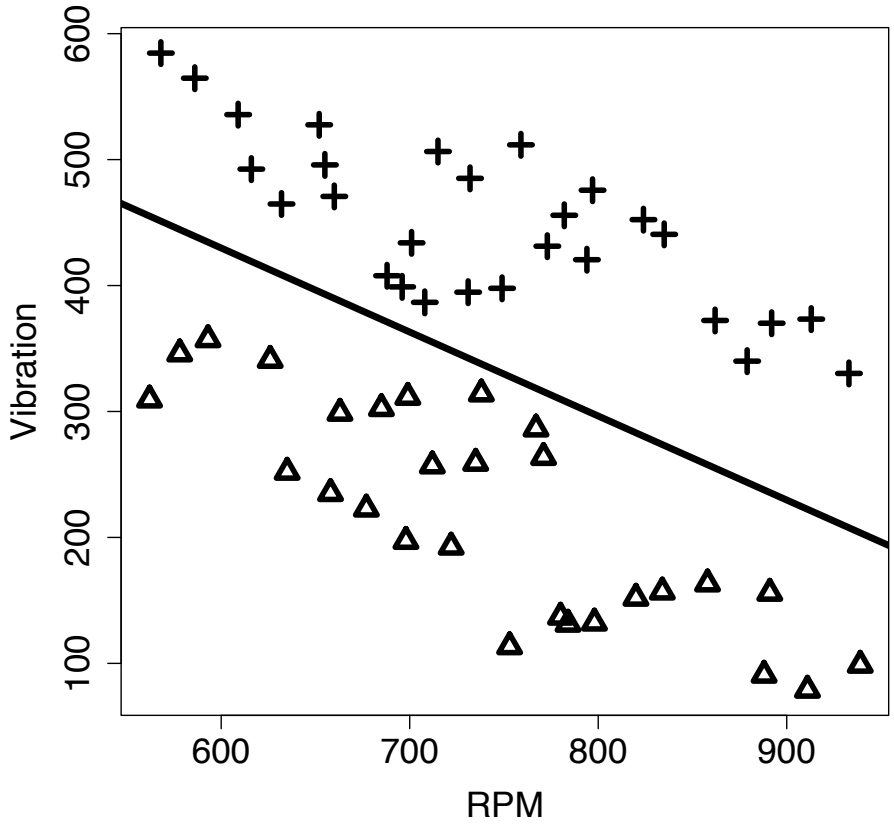
Classification - an example



John Kelleher and Brian Mac Namee and Aoife D'Arcy, Fundamentals of Machine Learning for Predictive Data Analytics

Content

Decision boundary



John Kelleher and Brian Mac Namee and Aoife D'Arcy, Fundamentals of Machine Learning for Predictive Data Analytics

Decision boundary

$$830 - 0.667 \times \text{RPM} - \text{Vibration} = 0$$

$$\theta_0 + \theta_1 x_1 + x_2 = 0$$

- For all "good" generators, we have: $\theta_0 + \theta_1 x_1 + x_2 >= 0$
- For all "faulty" generators, we have: $\theta_0 + \theta_1 x_1 + x_2 < 0$

Decision boundary

$$830 - 0.667 \times \text{RPM} - \text{Vibration} = 0$$

$$\theta_0 + \theta_1 x_1 + x_2 = 0$$

- For all “good” generators, we have: $\theta_0 + \theta_1 x_1 + x_2 >= 0$
- For all “faulty” generators, we have: $\theta_0 + \theta_1 x_1 + x_2 < 0$

Decision boundary

$$830 - 0.667 \times \text{RPM} - \text{Vibration} = 0$$

$$\theta_0 + \theta_1 x_1 + x_2 = 0$$

- For all “good” generators, we have: $\theta_0 + \theta_1 x_1 + x_2 >= 0$
- For all “faulty” generators, we have: $\theta_0 + \theta_1 x_1 + x_2 < 0$

Notation

$$\text{Let: } \theta = \begin{bmatrix} 1 \\ \theta_1 \\ \theta_0 \end{bmatrix}, x = \begin{bmatrix} x_2 \\ x_1 \\ 1 \end{bmatrix}$$

Then we have the decision boundary:

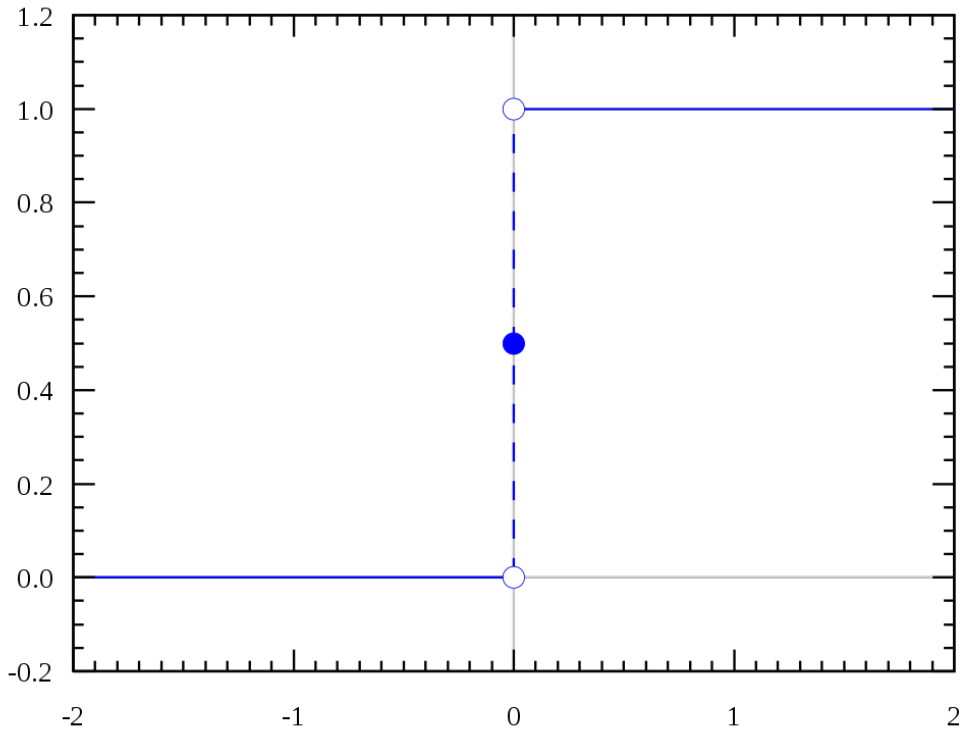
$$d_\theta(x) = \theta^T x = 0$$

Content

Heaviside step function

$$Heaviside(x) = \begin{cases} 1 & \text{if: } x >= 0 \\ 0 & \text{if: } x < 0 \end{cases}$$

Heaviside step function



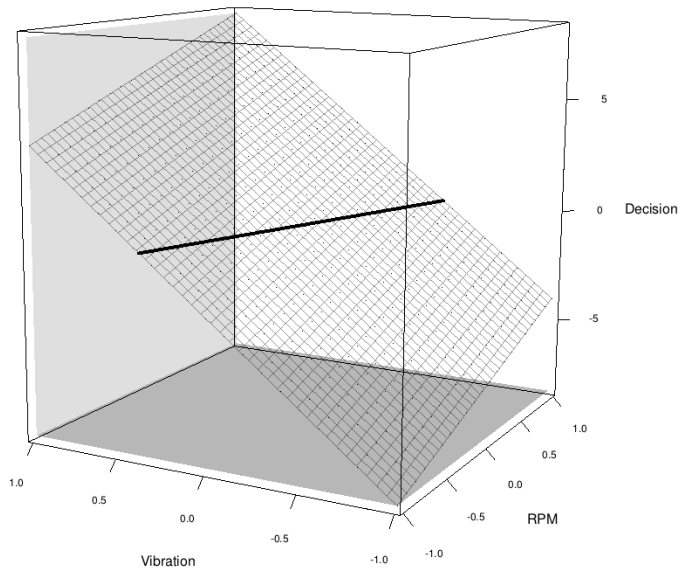
By Omegatron - Own work, CC BY-SA 3.0, https://commons.wikimedia.org/w/index.php?curid=801382

Model as a Heaviside step function

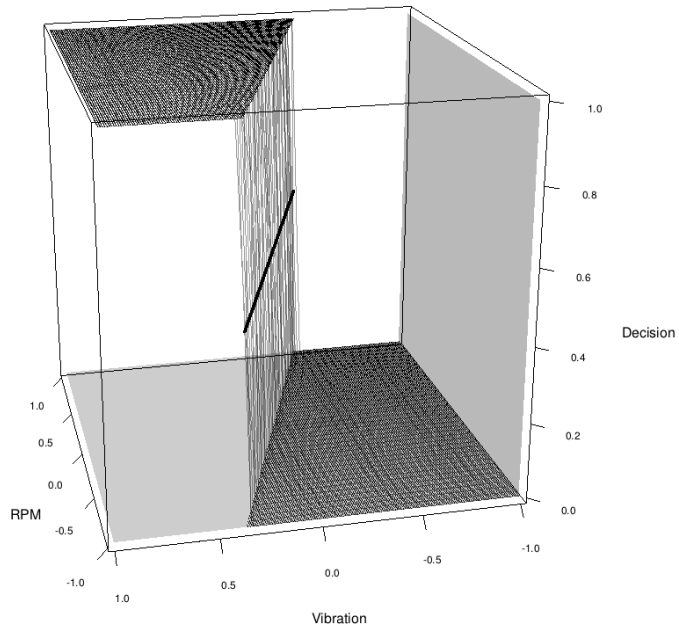
$$h_\theta(x) = Heaviside(d_\theta(x))$$

$$= \begin{cases} 1 & \text{if: } d_\theta(x) = \theta^T x >= 0; \text{ for } good \text{ generators} \\ 0 & \text{if: } d_\theta(x) = \theta^T x < 0; \text{ for } faulty \text{ generators} \end{cases}$$

Model as a Heaviside step function



(a)



(b)

$d_\theta(x)$

$h_\theta(x) = Heaviside(d_\theta(x))$

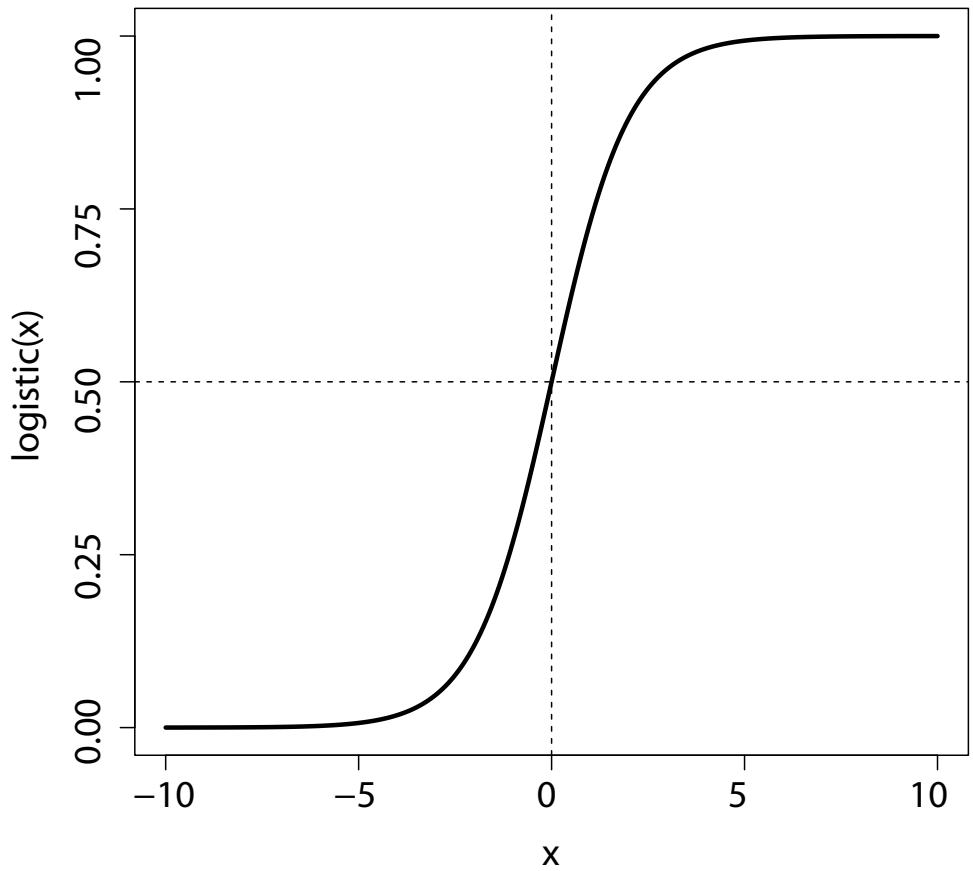
John Kelleher and Brian Mac Namee and Aoife D'Arcy, Fundamentals of Machine Learning for Predictive Data Analytics

Problem of the Heaviside step function

- Heaviside step function is not derivable and hard to optimize.
- An alternative is using a Logistic function (sigmoid function):

$$Logistic(x) = \frac{1}{1 + e^{-x}}$$

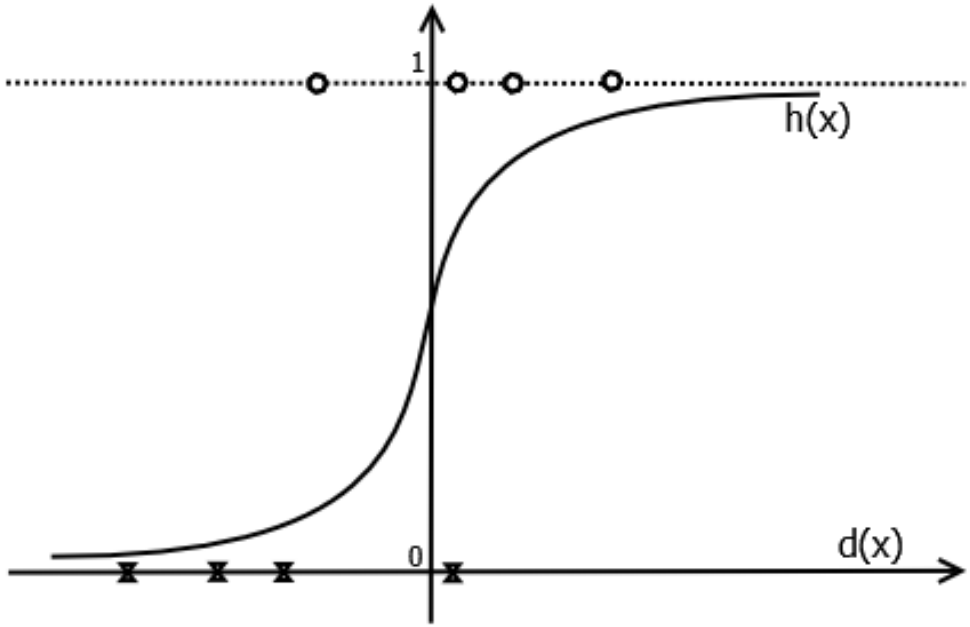
Logistic function



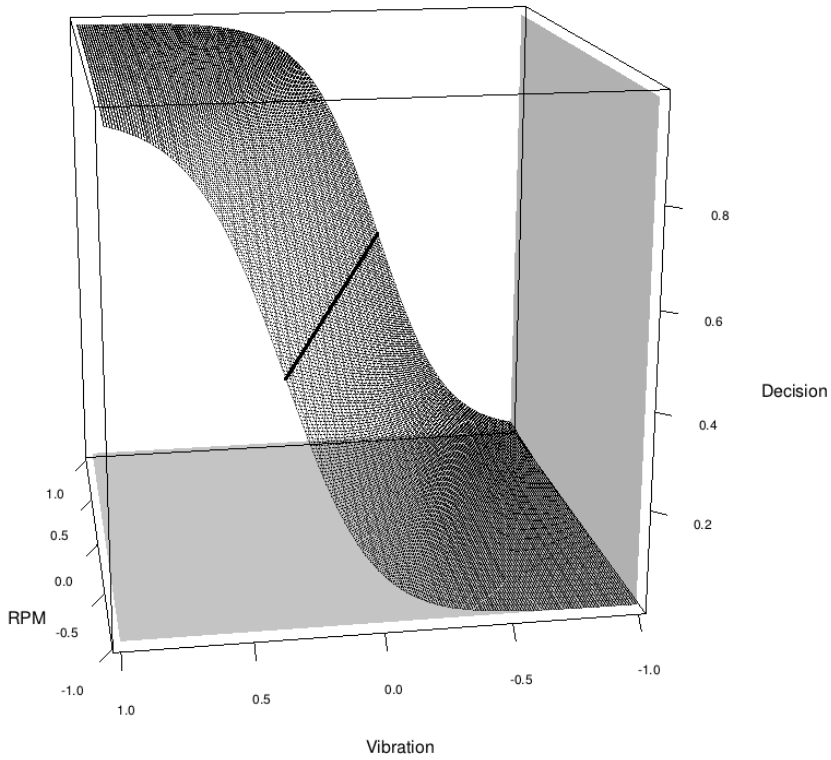
John Kelleher and Brian Mac Namee and Aoife D'Arcy, Fundamentals of Machine Learning for Predictive Data Analytics

Model as a logistic function

$$h_\theta(x) = Logistic(d_\theta(x)) = \frac{1}{1 + e^{-d_\theta(x)}} = \frac{1}{1 + e^{-\theta^T x}}$$



Model as a logistic function



John Kelleher and Brian Mac Namee and Aoife D'Arcy, Fundamentals of Machine Learning for Predictive Data Analytics

Content

Cost function

- The objective of a learning algorithm is to search a model to best predict the target feature on the training data given the inductive bias.
- One way to achieve this goal is to minimize the errors of the prediction over the training data.
- A cost function (loss function) is defined as the sum of the errors over the training samples.

Cost function

- A possible cost function:

$$J(\theta) = \frac{1}{n} \sum_{i=1}^{n} \frac{1}{2} (h_\theta(x^{\{i\}}) - y^{\{i\}})^2$$

- It is not a good cost function for Logistic regression because it is non-convex for a Logistic function.

Convex vs. non convex functions

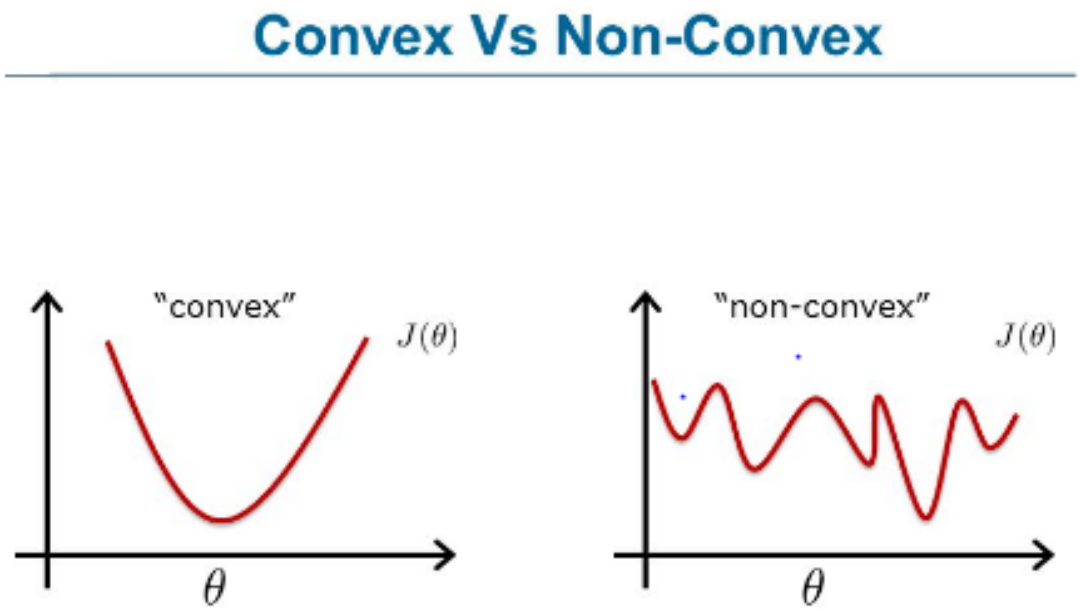


figure source: https://www.fromthegenesis.com/artificial-neural-network-part-7/

Cost function for Logistic regression

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

where:

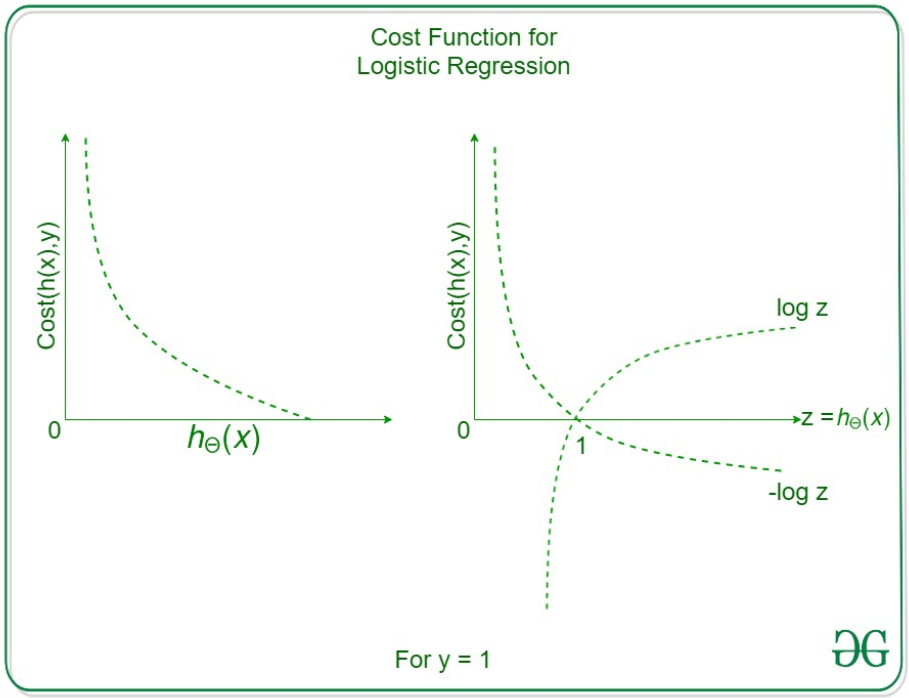
$$Cost(h_\theta(x), y) = \begin{cases} -\log(h_\theta(x)) & \text{if } y = 1 \\ -\log(1 - h_\theta(x)) & \text{if } y = 0 \end{cases}$$

$$= -\left[y \log(h_\theta(x)) + (1 - y) \log(1 - h_\theta(x))\right]$$

$$y^{\{*\}} \in \{0, 1\}$$

Cost function for Logisitc regression

$$Cost(h_\theta(x), y) = -\left[y \log(h_\theta(x)) + (1 - y) \log(1 - h_\theta(x))\right]$$



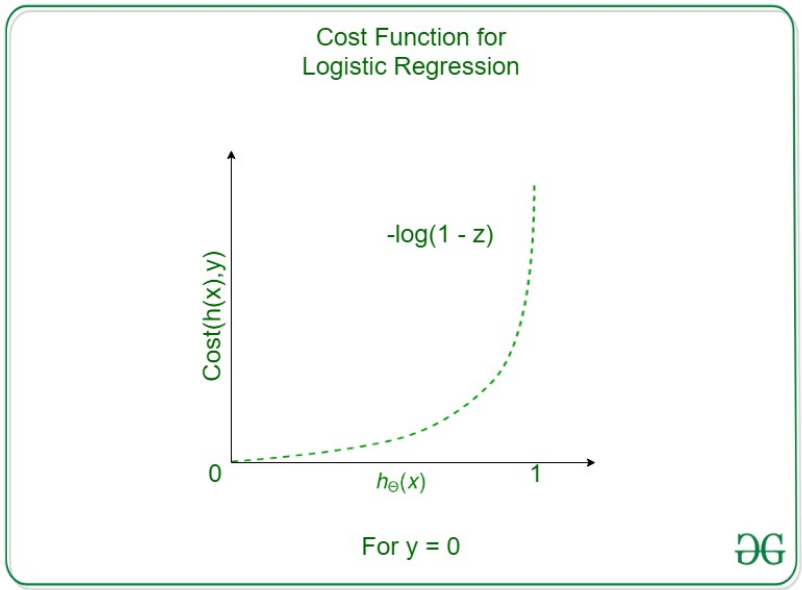


figure source: https://www.geeksforgeeks.org/ml-cost-function-in-logistic-regression/

Content

Gradient descend

- Now we have a cost function which defines the errors of a model over the training data.
- Next we need to search all the possible models to find a model with the minimal cost.
- We use a gradient descend algorithm for this purpose.

Gradient descend

Have some function $J(\theta_0, \theta_1)$

Want $\min_{\theta_0, \theta_1} J(\theta_0, \theta_1)$

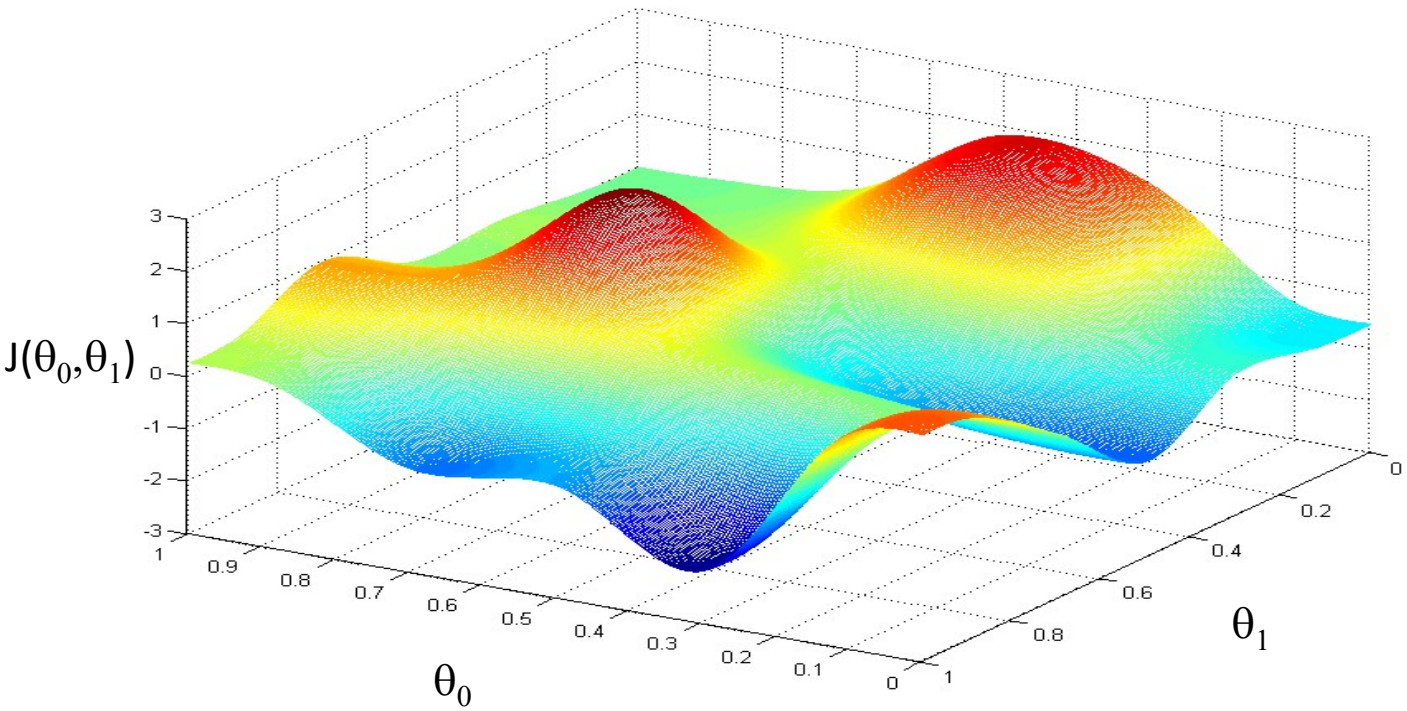
Outline:

- Start with some θ_0, θ_1
- Keep changing θ_0, θ_1 to reduce $J(\theta_0, \theta_1)$

  until we hopefully end up at a minimum

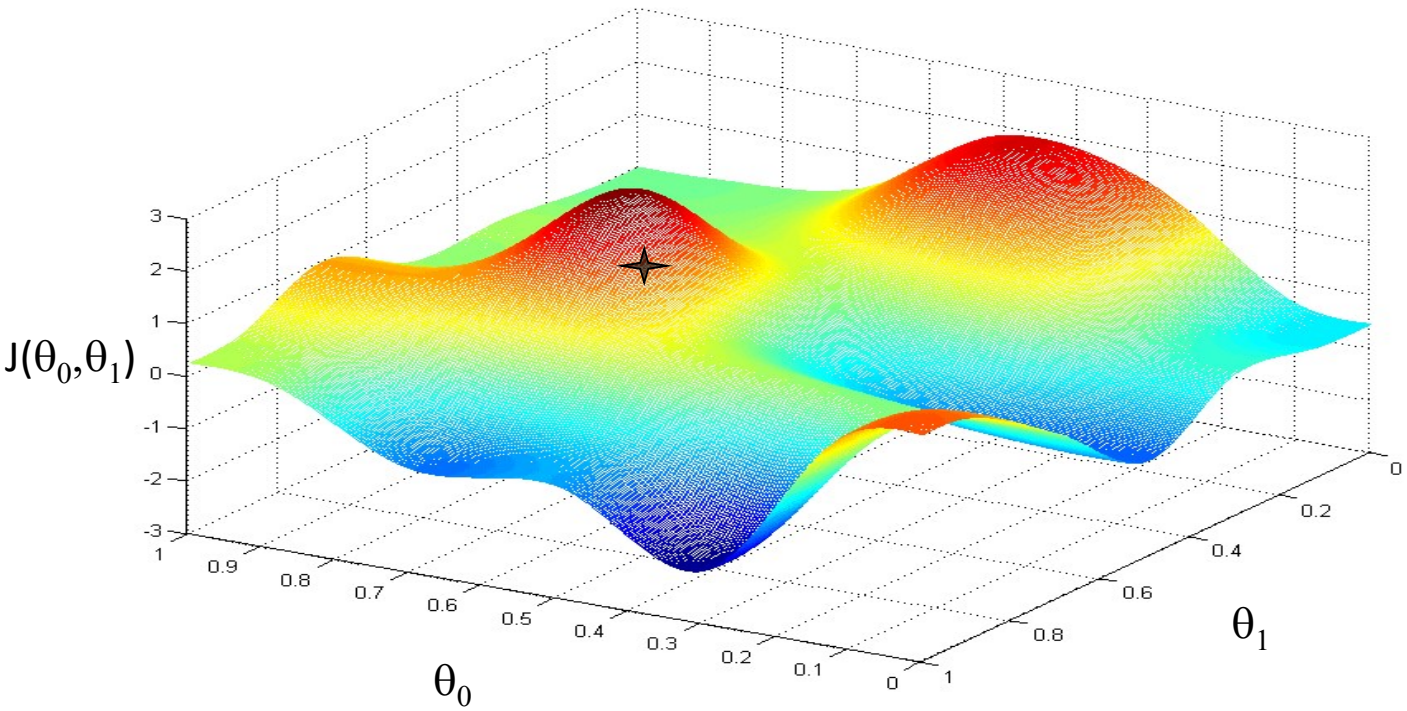
Andrew Ng, Machine Learning, Coursera course

Gradient descend



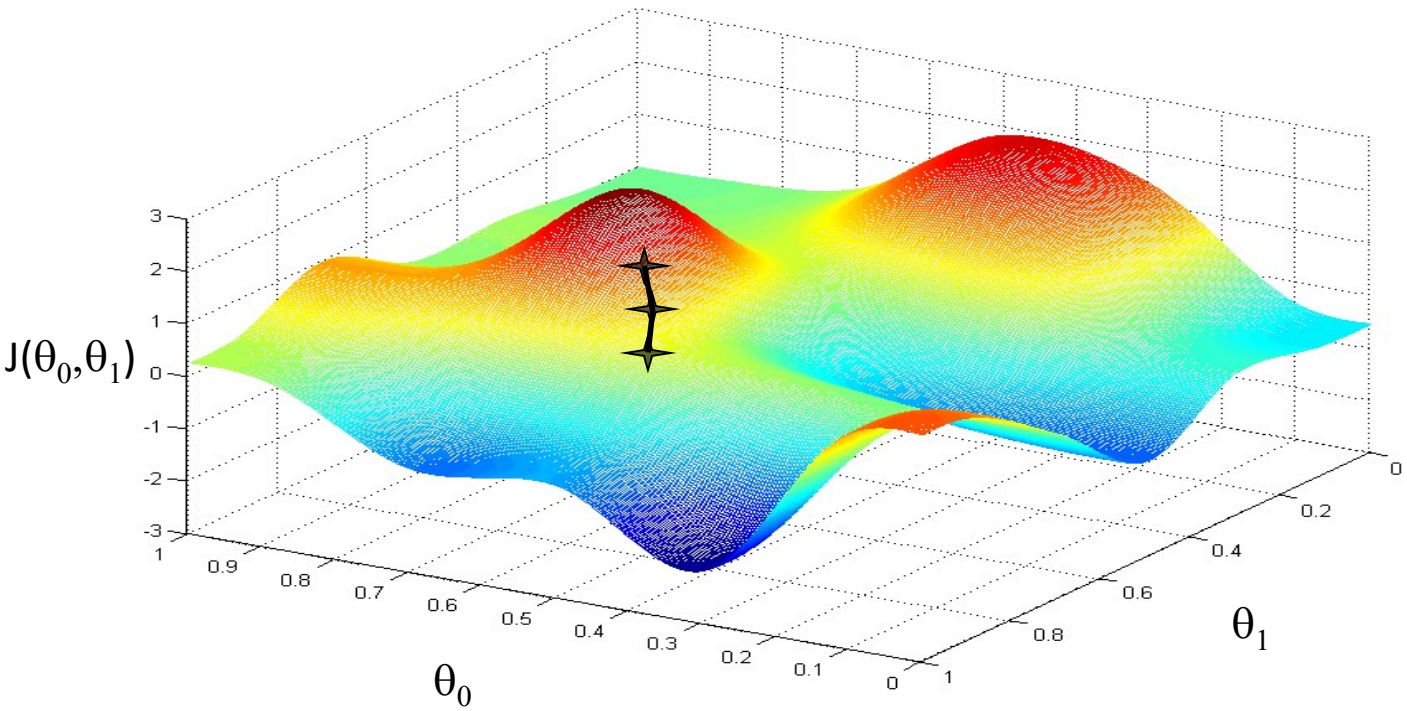
Andrew Ng, Machine Learning, Coursera course

Gradient descend



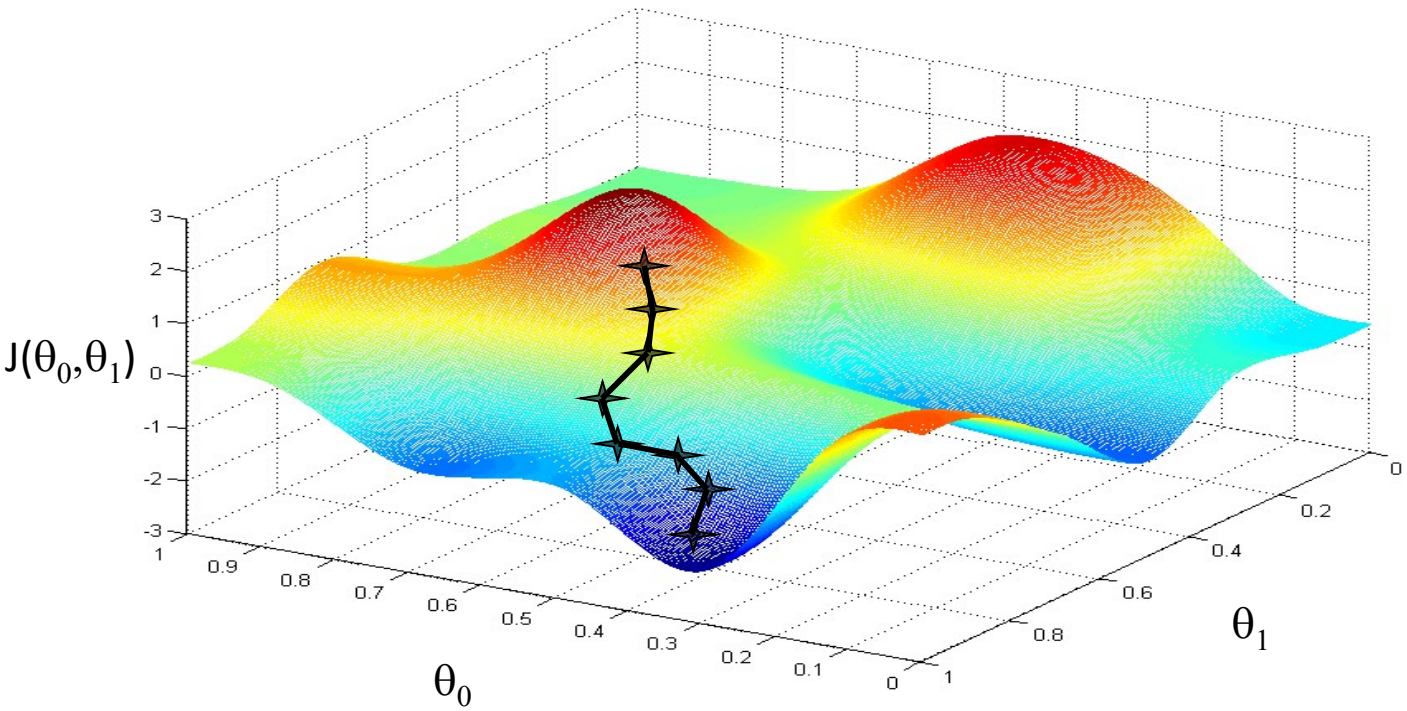
Andrew Ng, Machine Learning, Coursera course

Gradient descend



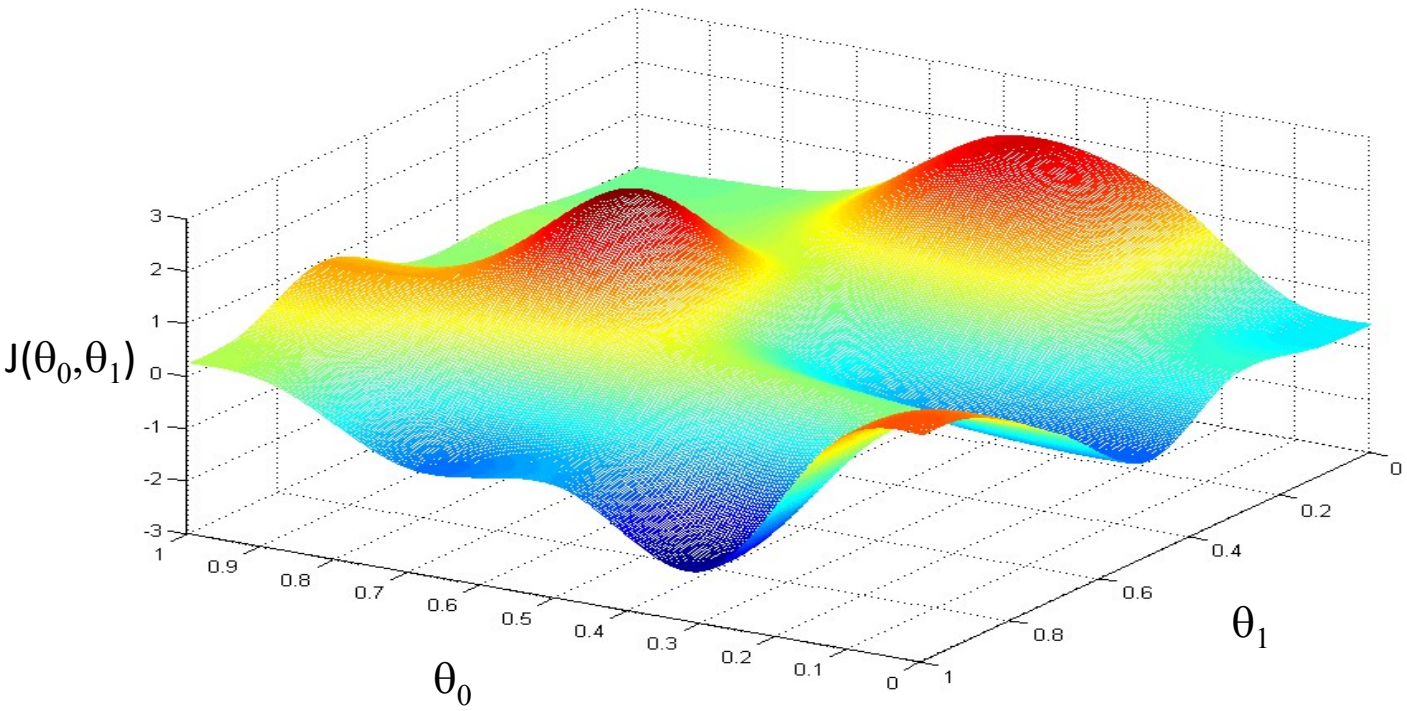
Andrew Ng, Machine Learning, Coursera course

Gradient descend



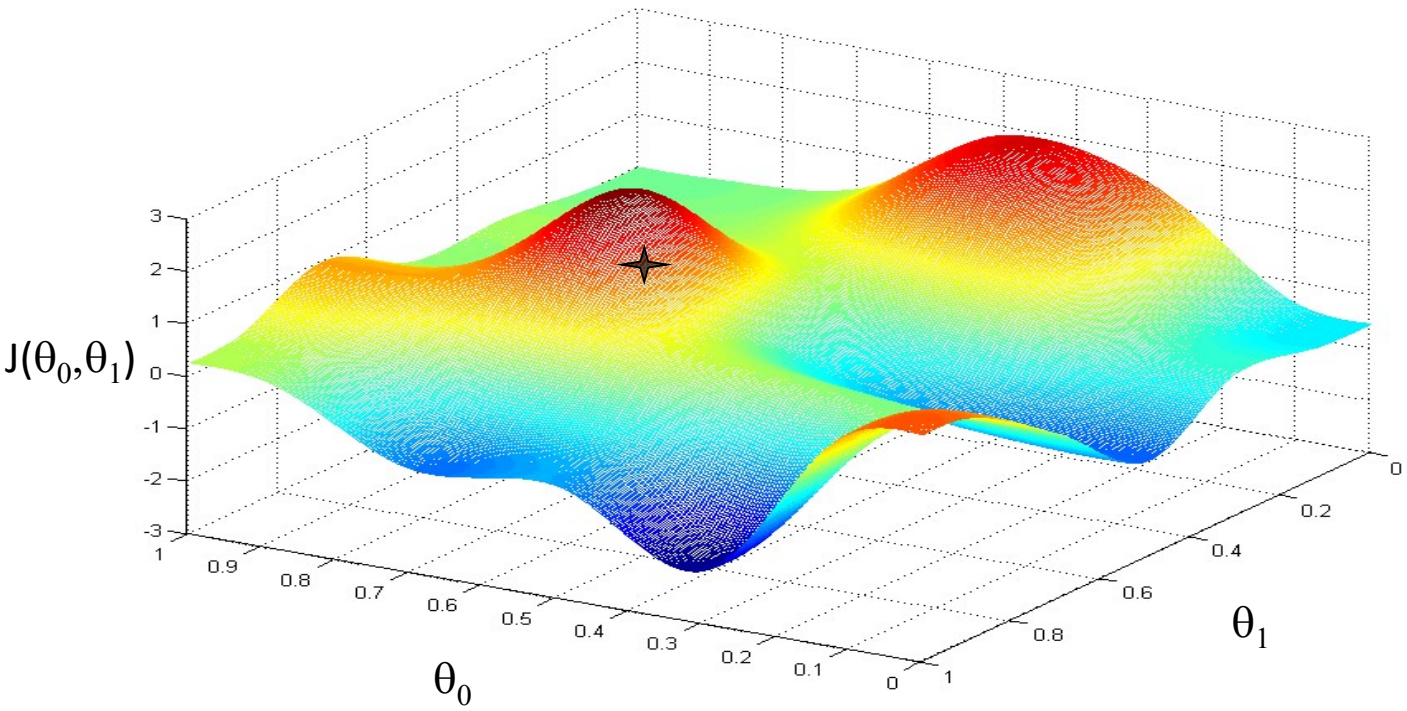
Andrew Ng, Machine Learning, Coursera course

Gradient descend



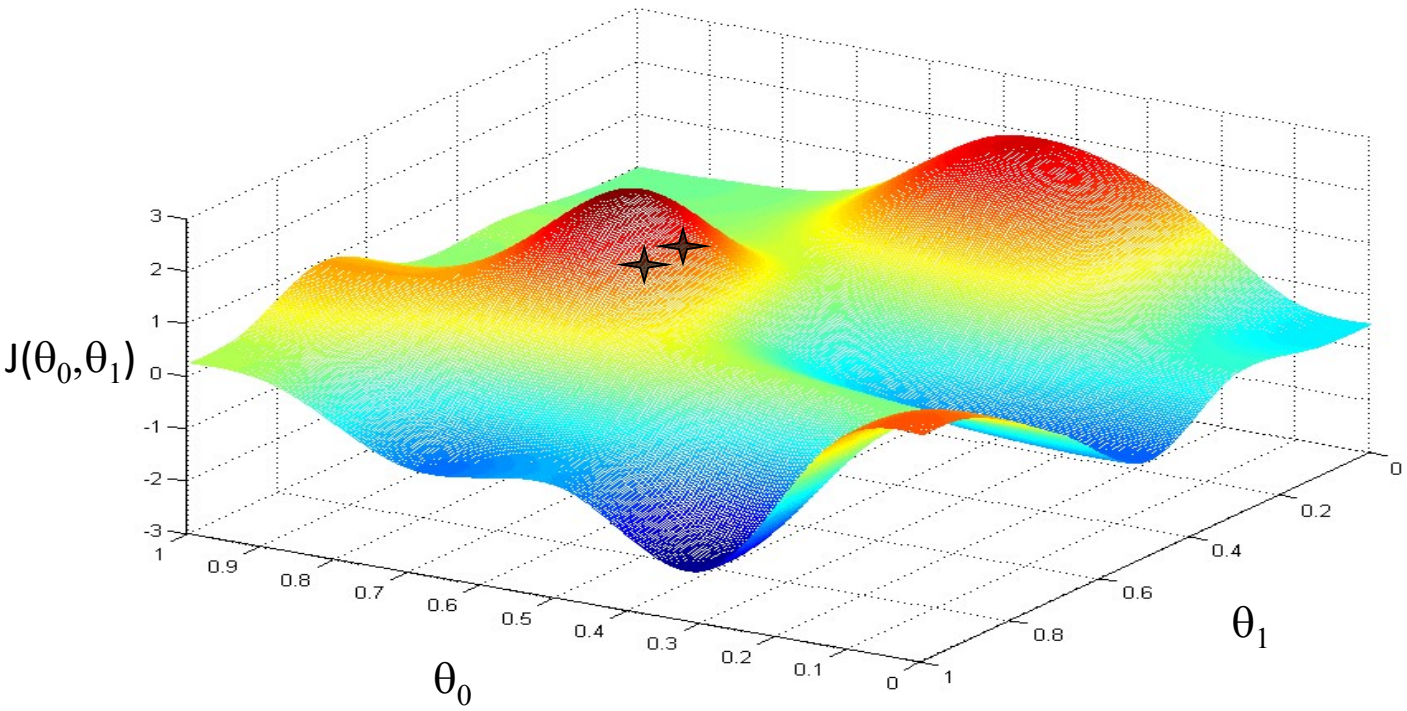
Andrew Ng, Machine Learning, Coursera course

Gradient descend



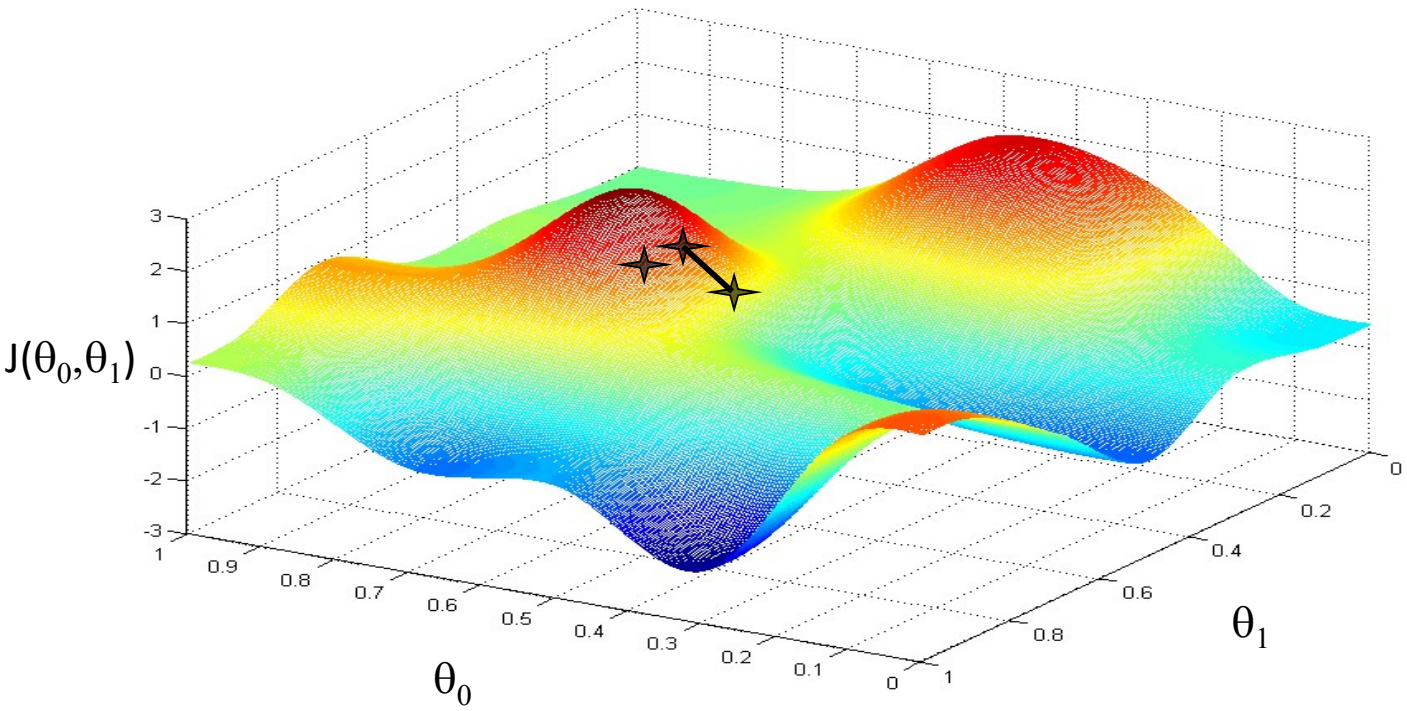
Andrew Ng, Machine Learning, Coursera course

Gradient descend



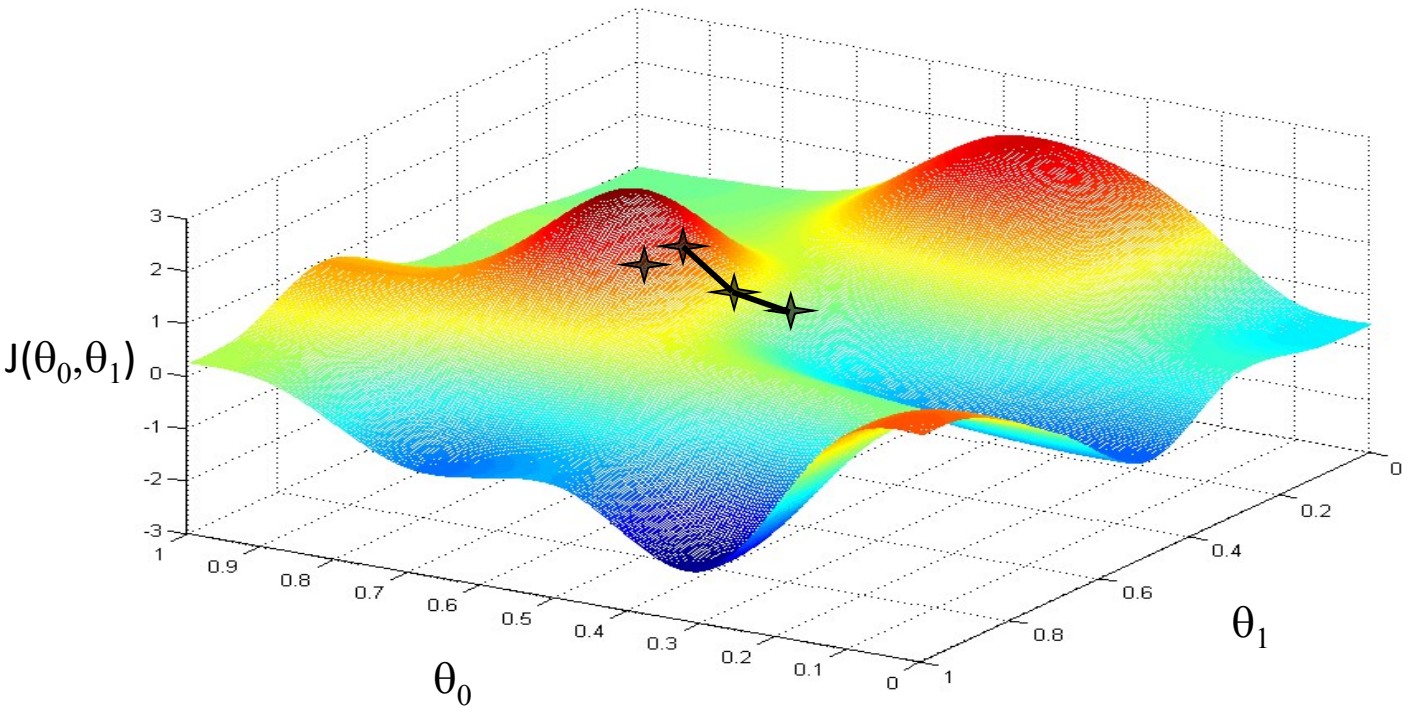
Andrew Ng, Machine Learning, Coursera course

Gradient descend



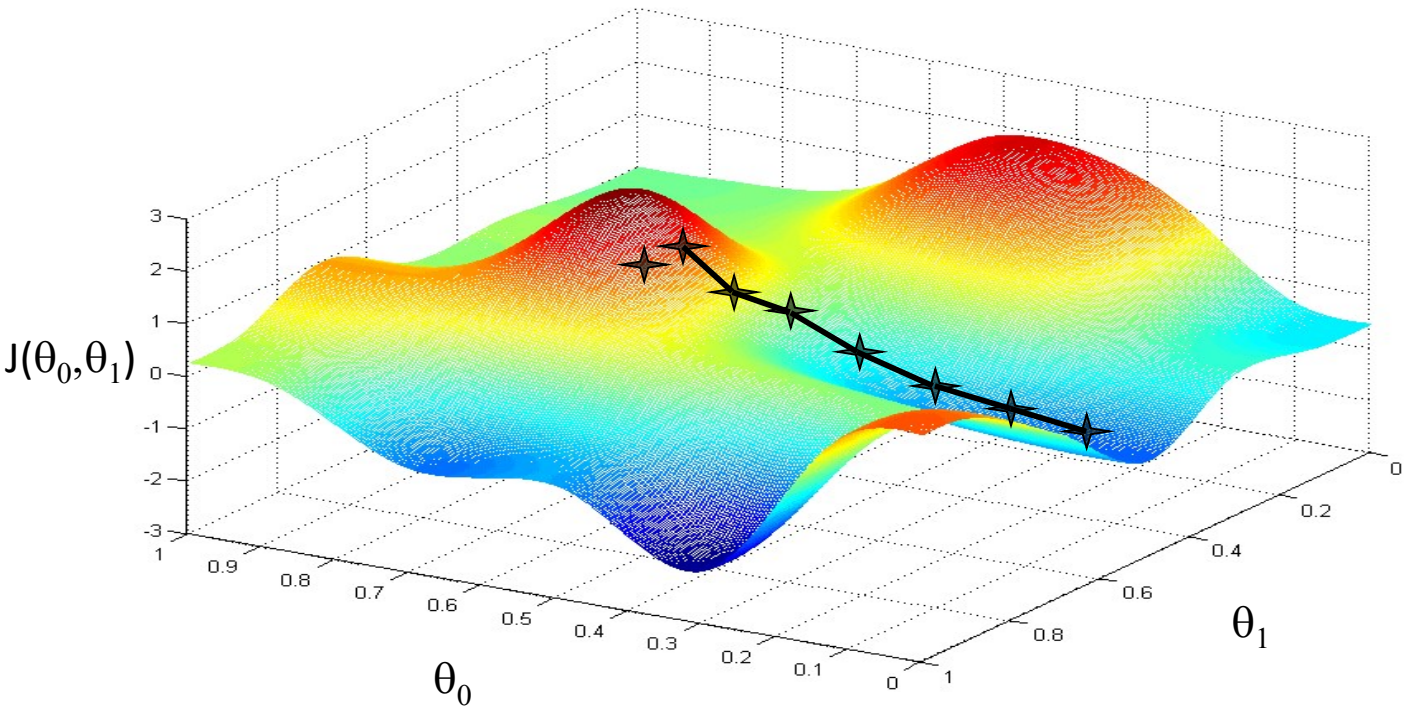
Andrew Ng, Machine Learning, Coursera course

Gradient descend



Andrew Ng, Machine Learning, Coursera course

Gradient descend



Andrew Ng, Machine Learning, Coursera course

Gradient descend algorithm

Gradient descent algorithm

repeat until convergence {

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1) \qquad \begin{array}{l}(\text{simultaneously update} \\ j = 0 \text{ and } j = 1)\end{array}$$

}

Andrew Ng, Machine Learning, Coursera course

Gradient descend algorithm

Gradient descent algorithm

repeat until convergence {

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1) \qquad (\text{for } j = 0 \text{ and } j = 1)$$

}

Correct: Simultaneous update

$$\text{temp0} := \theta_0 - \alpha \frac{\partial}{\partial \theta_0} J(\theta_0, \theta_1)$$
$$\text{temp1} := \theta_1 - \alpha \frac{\partial}{\partial \theta_1} J(\theta_0, \theta_1)$$
$$\theta_0 := \text{temp0}$$
$$\theta_1 := \text{temp1}$$

Andrew Ng, Machine Learning, Coursera course

Gradient descend algorithm

Gradient descent algorithm

repeat until convergence {

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1) \qquad (\text{for } j = 0 \text{ and } j = 1)$$

}

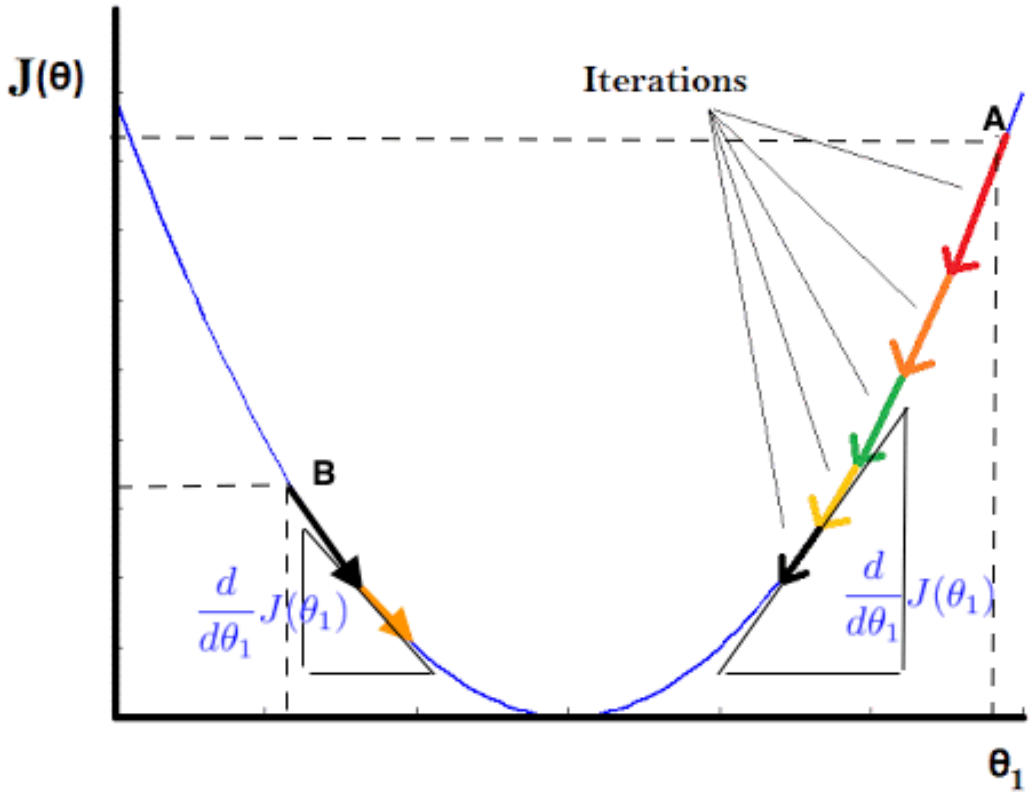
Correct: Simultaneous update

$$\text{temp0} := \theta_0 - \alpha \frac{\partial}{\partial \theta_0} J(\theta_0, \theta_1)$$
$$\text{temp1} := \theta_1 - \alpha \frac{\partial}{\partial \theta_1} J(\theta_0, \theta_1)$$
$$\theta_0 := \text{temp0}$$
$$\theta_1 := \text{temp1}$$

Incorrect:

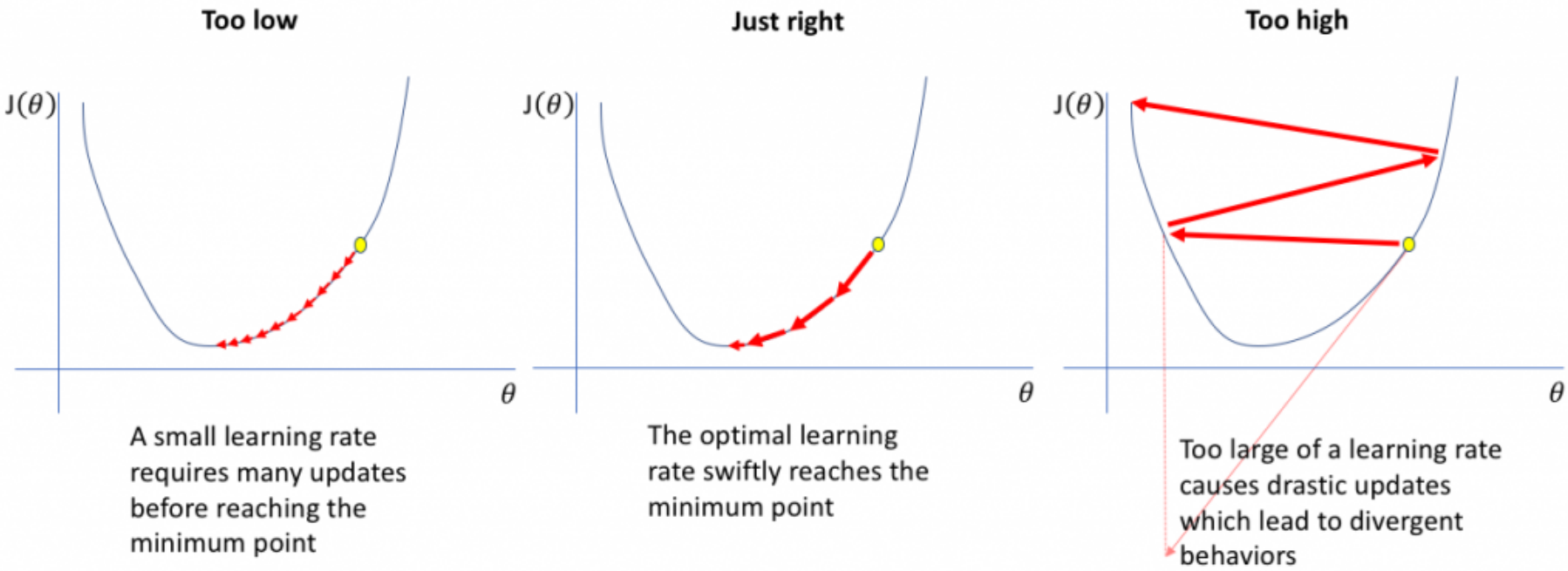
$$\text{temp0} := \theta_0 - \alpha \frac{\partial}{\partial \theta_0} J(\theta_0, \theta_1)$$
$$\theta_0 := \text{temp0}$$
$$\text{temp1} := \theta_1 - \alpha \frac{\partial}{\partial \theta_1} J(\theta_0, \theta_1)$$
$$\theta_1 := \text{temp1}$$

Andrew Ng, Machine Learning, Coursera course



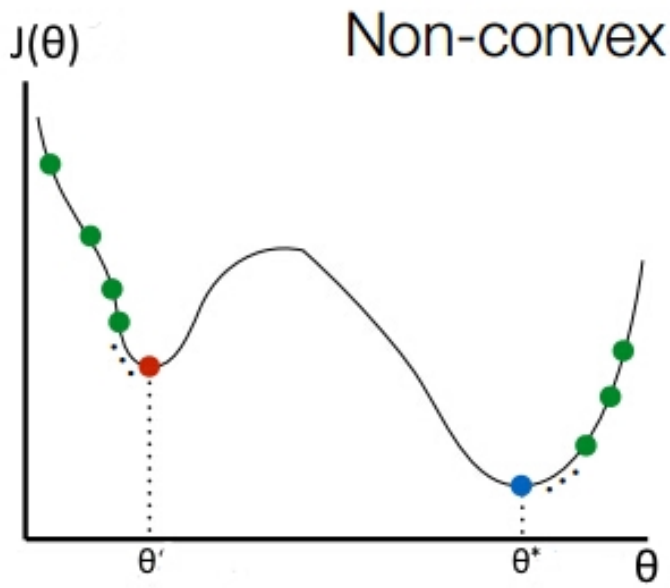
- The derivation $\frac{\partial J(\theta)}{\partial \theta}$ gives the direction of the movement.
- The learning rate α is used to adjust the size for each step.

figure source: https://machinelearningmedium.com/2017/08/15/gradient-descent/



The influence of the learning rate.

figure source: https://ithelp.ithome.com.tw/m/articles/10204032



A non-convex error surface may lead to local optima.

Gradient descend for Logistic regression

Model

$$h_\theta(x) = \frac{1}{1 + e^{-\theta^T x}}$$

Parameters

$$\theta_0, \theta_1$$

Cost Function

$$J(\theta) = -\frac{1}{n} \sum_{i=1}^{n} \left[y^{\{i\}} \log(h_\theta(x^{\{i\}})) + (1 - y^{\{i\}}) \log(1 - h_\theta(x^{\{i\}})) \right]$$

Goal

$$\underset{\theta_0, \theta_1}{\text{minimize}}\ J(\theta)$$

Gradient descend for Logistic regression

Gradient descent:

$$J(\theta) = -\frac{1}{n}\sum_{i=1}^{n}\left[y^{\{i\}}\log(h_\theta(x^{\{i\}})) + (1-y^{\{i\}})\log(1-h_\theta(x^{\{i\}}))\right]$$

Want $\min_\theta J(\theta)$:

Repeat {

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

(simultaneously update all θ_j)

}

Gradient descend for Logistic regression

$$J(\theta) = -\frac{1}{n}\sum_{i=1}^{n}\left[y^{\{i\}}\log(h_\theta(x^{\{i\}})) + (1-y^{\{i\}})\log(1-h_\theta(x^{\{i\}}))\right]$$

$$\frac{\partial}{\partial\theta_j}J(\theta) = -\frac{1}{n}\frac{\partial}{\partial\theta_j}\sum_{i=1}^{n}\left[y^{\{i\}}\log(h_\theta(x^{\{i\}})) + (1-y^{\{i\}})\log(1-h_\theta(x^{\{i\}}))\right]$$

$$= \dots\dots$$

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

Gradient descend for Logistic regression

Gradient descent:

$$J(\theta) = -\frac{1}{n}\sum_{i=1}^{n}\left[y^{\{i\}}\log(h_\theta(x^{\{i\}})) + (1-y^{\{i\}})\log(1-h_\theta(x^{\{i\}}))\right]$$

Want $\min_\theta J(\theta)$:

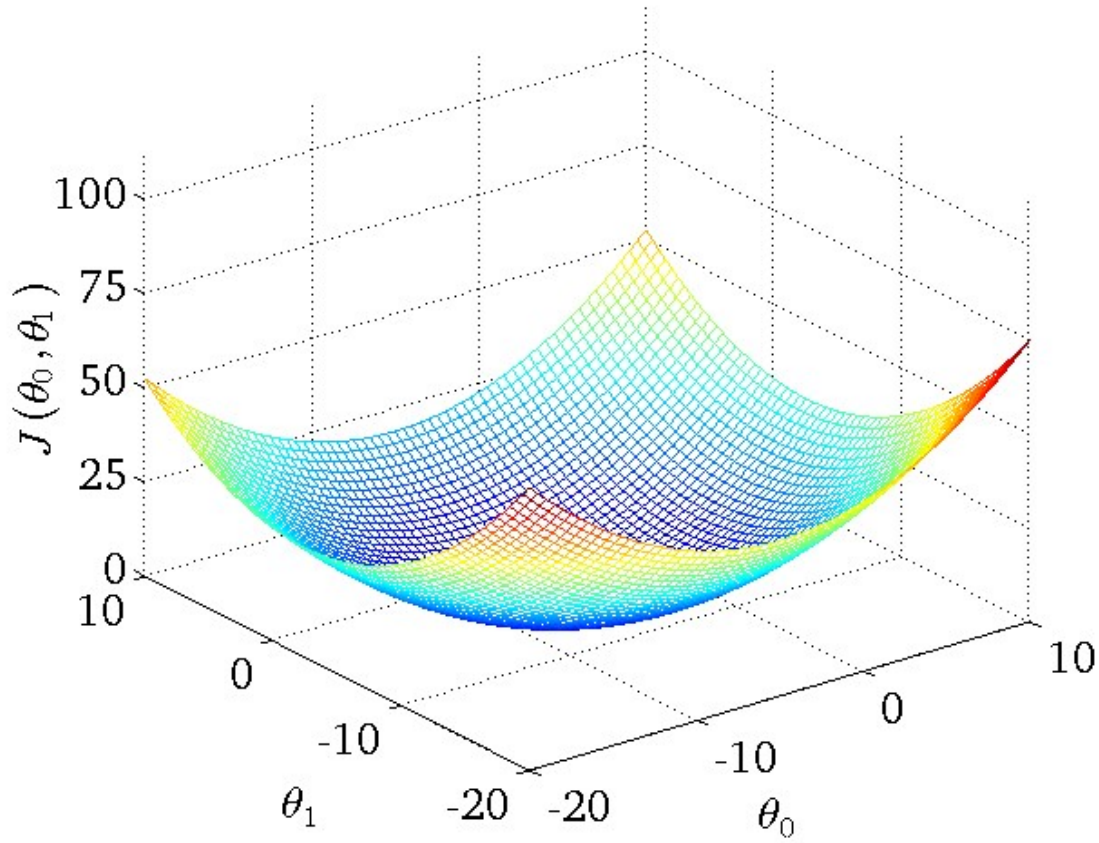
Repeat {

$\theta_j = \theta_j - \alpha\frac{1}{n}\sum_{i=1}^{n}(h_\theta(x^{\{i\}}) - y^{\{i\}})x_j^{\{i\}}$

(simultaneously update all θ_j)

}

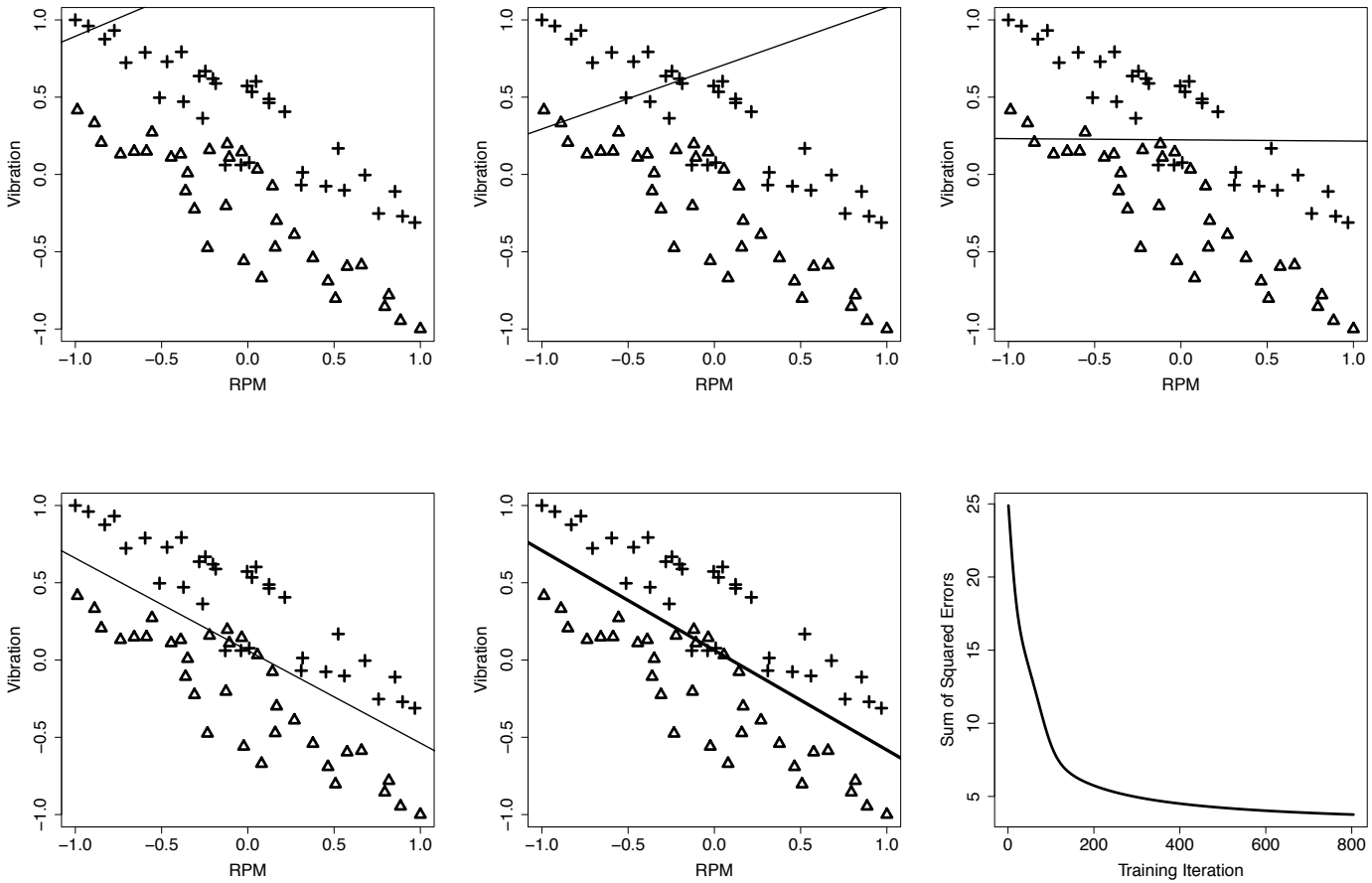
Gradient descend for logistic regression



Fortunately the error surface for logistic regression is convex.

Andrew Ng, Machine Learning, Coursera course

Gradient descend for logistic regression



John Kelleher and Brian Mac Namee and Aoife D'Arcy, Fundamentals of Machine Learning for Predictive Data Analytics

Gradient descent variants

- There are three variants of gradient descent.
  - Batch gradient descent
  - Stochastic gradient descent
  - Mini-batch gradient descent

- The difference of these algorithms is **the amount of data.**

Update equation

$$\theta = \theta - \eta * \nabla_\theta J(\theta)$$

This term is different with each method

Ruder, Sebastian. "An overview of gradient descent optimization algorithms." arXiv:1609.04747 (2016).

Batch gradient descent

This method computes the gradient of the cost function with **the entire training dataset**.

Update equation

$$\theta = \theta - \eta * \nabla_\theta J(\theta)$$

We need to calculate the gradients for the whole dataset to perform **just one update.**

Code

```
for i in range(nb_epochs):
  params_grad = evaluate_gradient(loss_function, data, params)
  params = params - learning_rate * params_grad
```

Ruder, Sebastian. "An overview of gradient descent optimization algorithms." arXiv:1609.04747 (2016).

Batch gradient descent

- Advantage
  - It is guaranteed to converge **to the global minimum for convex error surfaces and to a local minimum for non-convex surfaces.**

- Disadvantages
  - It can be **very slow**.
  - It is intractable for datasets that **do not fit in memory**.
  - It **does not allow** us to update our model **online.**

Ruder, Sebastian. "An overview of gradient descent optimization algorithms." arXiv:1609.04747 (2016).

Stochastic gradient descent

This method performs a parameter update for **each** training example $x^{(i)}$ and label $y^{(i)}$.

Update equation

$$\theta = \theta - \eta * \nabla_\theta J(\theta; x^{(i)}; y^{(i)})$$

We need to calculate the gradients for the whole dataset to perform **just one update.**

Code

```
for i in range(nb_epochs):
  np.random.shuffle(data)
  for example in data:
    params_grad = evaluate_gradient(loss_function, example, params)
    params = params - learning_rate * params_grad
```

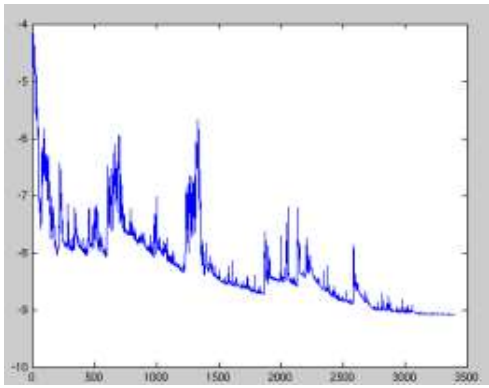
Note : we shuffle the training data at every epoch

Ruder, Sebastian. "An overview of gradient descent optimization algorithms." arXiv:1609.04747 (2016).

Stochastic gradient descent

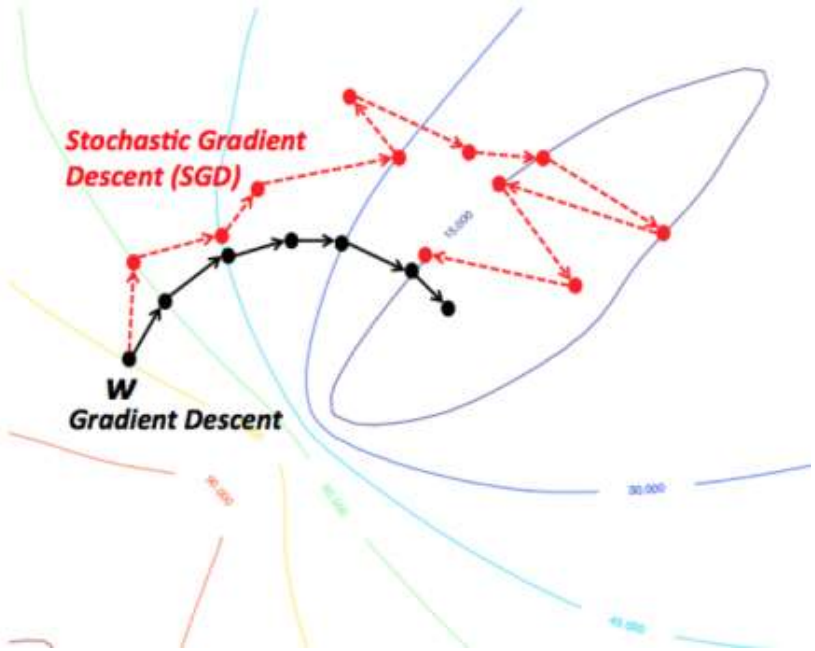
- Advantage
  - It is usually **much faster** than batch gradient descent.
  - It can be **used to learn online.**

- Disadvantages
  - It performs frequent updates with a **high variance** that cause the objective function to fluctuate heavily.



Ruder, Sebastian. "An overview of gradient descent optimization algorithms." arXiv:1609.04747 (2016).

The fluctuation : Batch vs SGD



https://wikidocs.net/3413

- Batch gradient descent converges to the minimum of the basin the parameters are placed in and the **fluctuation is small**.

- **SGD's fluctuation is large but it enables to jump to new and potentially better local minima.**

However, this ultimately complicates convergence to the exact minimum, as SGD will keep overshooting

Ruder, Sebastian. "An overview of gradient descent optimization algorithms." arXiv:1609.04747 (2016).

Learning rate of SGD

- When **we slowly decrease the learning rate, SGD shows the same convergence behaviour as batch gradient descent**
  - It almost certainly converging to a local or the global minimum for non-convex and convex optimization respectively.

Ruder, Sebastian. "An overview of gradient descent optimization algorithms." arXiv:1609.04747 (2016).

Mini-batch gradient descent

This method takes the best of both batch and SGD, and performs an update for every mini-batch of n.

Update equation

$$\theta = \theta - \eta * \nabla_\theta J(\theta; x^{(i:i+n)}; y^{(i:i+n)})$$

Code

```
for i in range(nb_epochs):
  np.random.shuffle(data)
  for batch in get_batches(data, batch_size=50):
    params_grad = evaluate_gradient(loss_function, batch, params)
    params = params - learning_rate * params_grad
```

Ruder, Sebastian. "An overview of gradient descent optimization algorithms." arXiv:1609.04747 (2016).

Mini-batch gradient descent

- Advantage :
  - It **reduces the variance** of the parameter updates.
    - This can lead to more stable convergence.
  - It can make use of highly optimized matrix optimizations common to deep learning libraries that make computing the gradient very efficiently.
- Disadvantage :
  - We have to set mini-batch size.
    - Common mini-batch sizes range between 50 and 256, but can vary for different applications.

Ruder, Sebastian. "An overview of gradient descent optimization algorithms." arXiv:1609.04747 (2016).

Trade-off

- Depending on the amount of data, they make a trade-off :
  - The **accuracy** of the parameter update
  - The **time** it takes to perform an update.

Method	Accuracy	Time	Memory Usage	Online Learning
Batch gradient descent	○	Slow	High	×
Stochastic gradient descent	△	High	Low	○
Mini-batch gradient descent	○	Midium	Midium	○

Ruder, Sebastian. "An overview of gradient descent optimization algorithms." arXiv:1609.04747 (2016).

Content

Multiclass classification

- Email foldering/tagging: Work, Friends, Family, Hobby

$$y = \begin{cases} 1, & \textit{Work} \\ 2, & \textit{Friends} \\ 3, & \textit{Family} \\ 4, & \textit{Hobby} \end{cases}$$

- Medical diagrams: Not ill, Cold, Flu
- Weather: Sunny, Cloudy, Rain, Snow

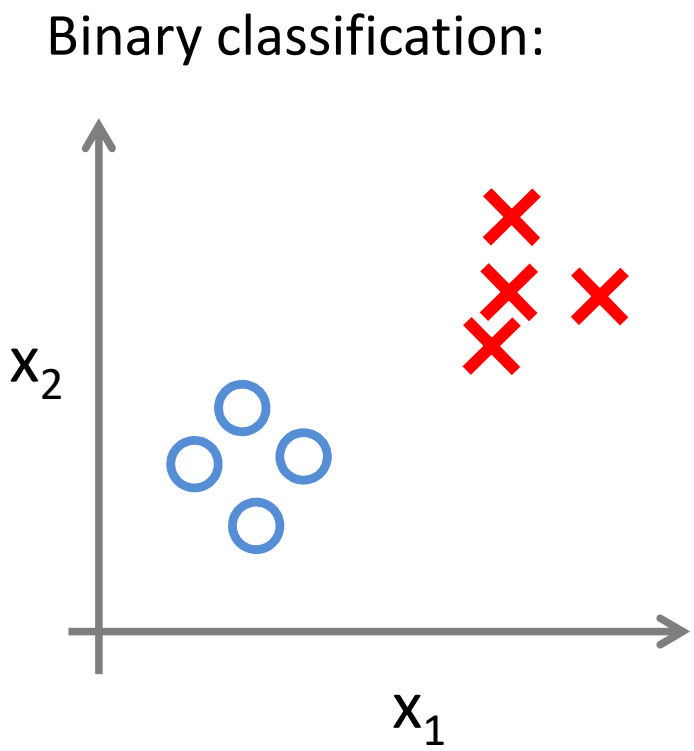
Multiclass classification

- Email foldering/tagging: Work, Friends, Family, Hobby

$$y = \begin{cases} 1, & \textit{Work} \\ 2, & \textit{Friends} \\ 3, & \textit{Family} \\ 4, & \textit{Hobby} \end{cases}$$

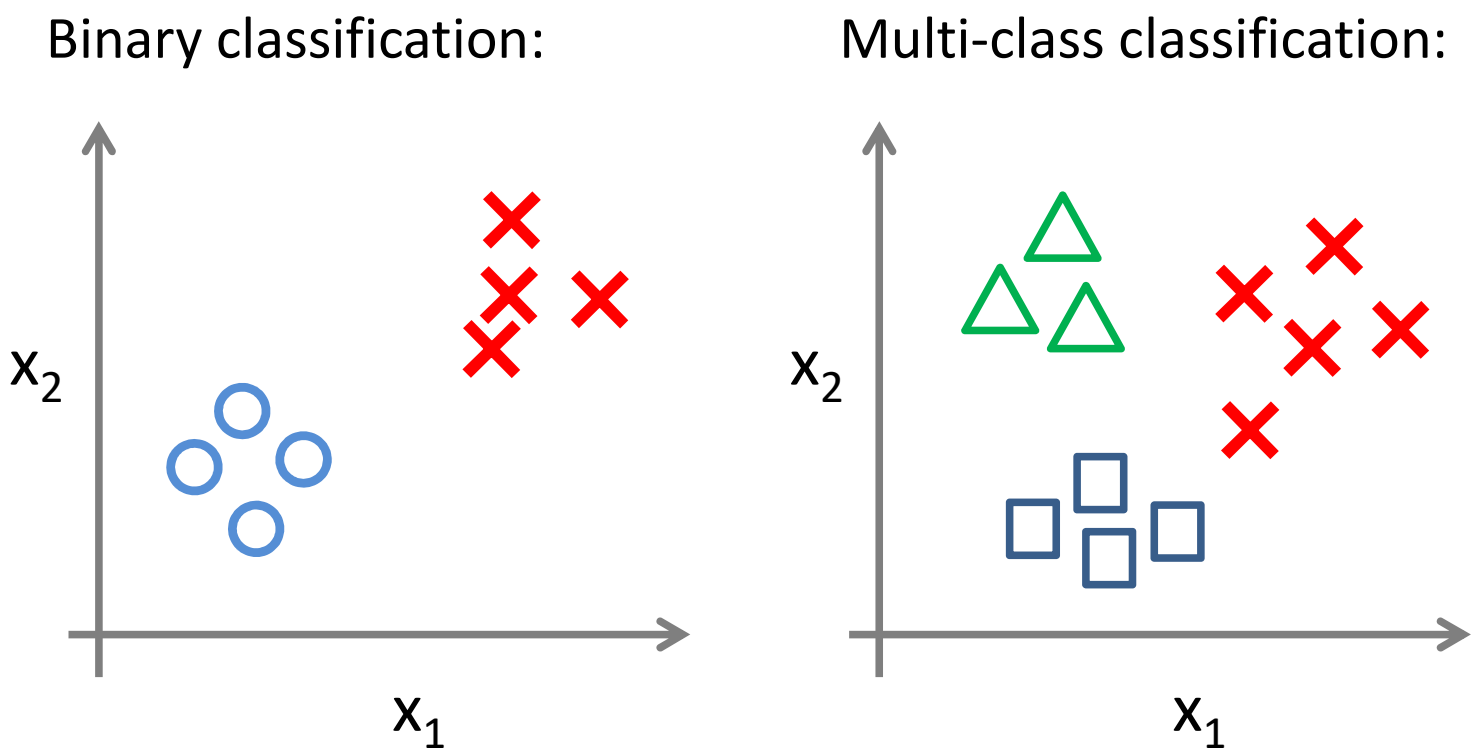
- Medical diagrams: Not ill, Cold, Flu
- Weather: Sunny, Cloudy, Rain, Snow

Multiclass classification



Andrew Ng, Machine Learning, Coursera course

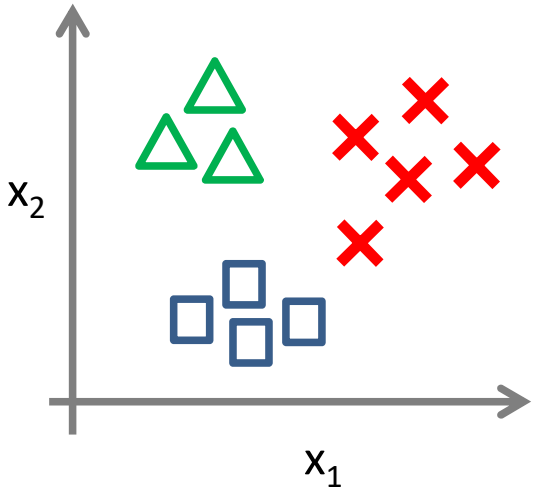
Multiclass classification



Andrew Ng, Machine Learning, Coursera course

Multiclass classification

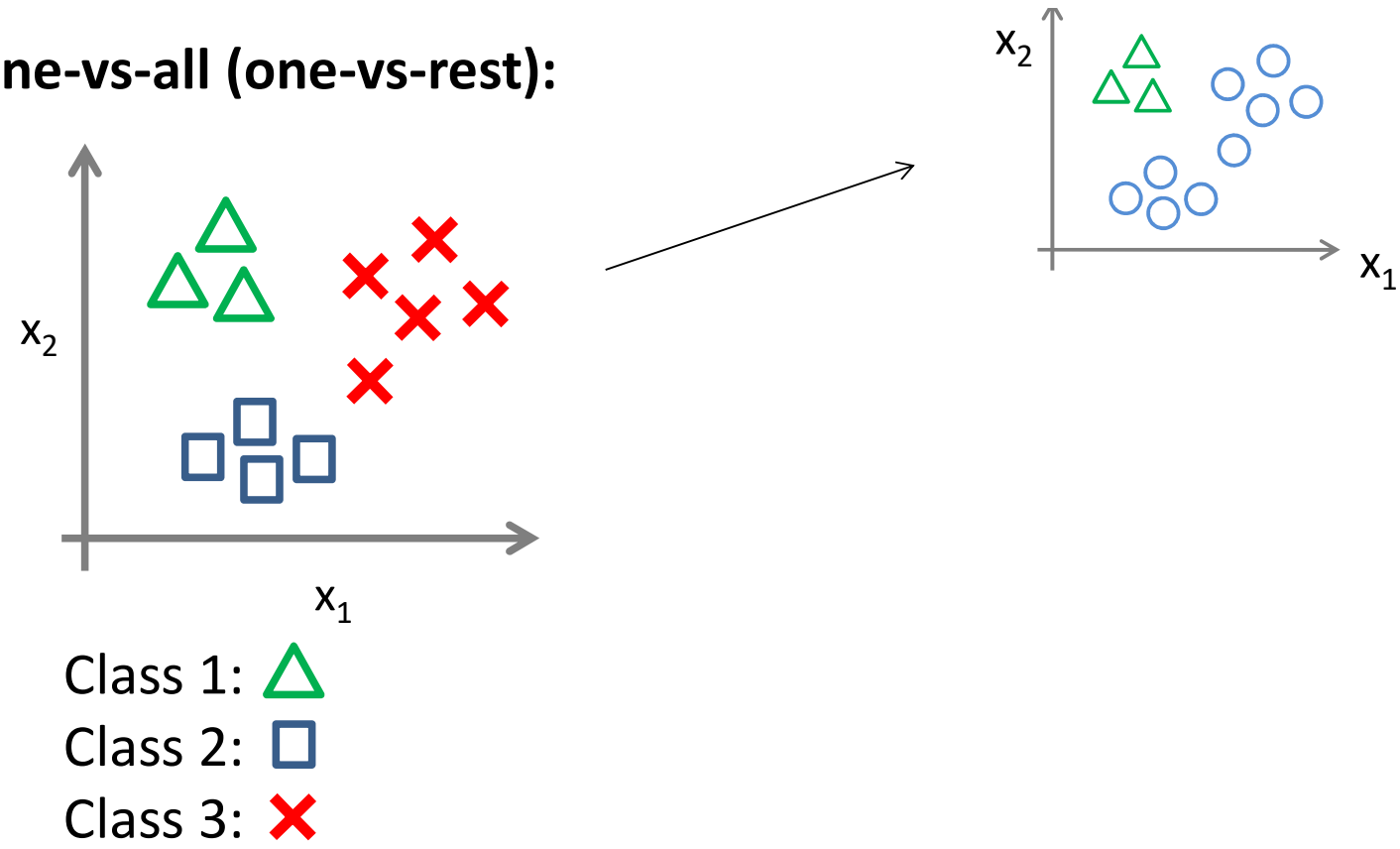
One-vs-all (one-vs-rest):



Andrew Ng, Machine Learning, Coursera course

Multiclass classification

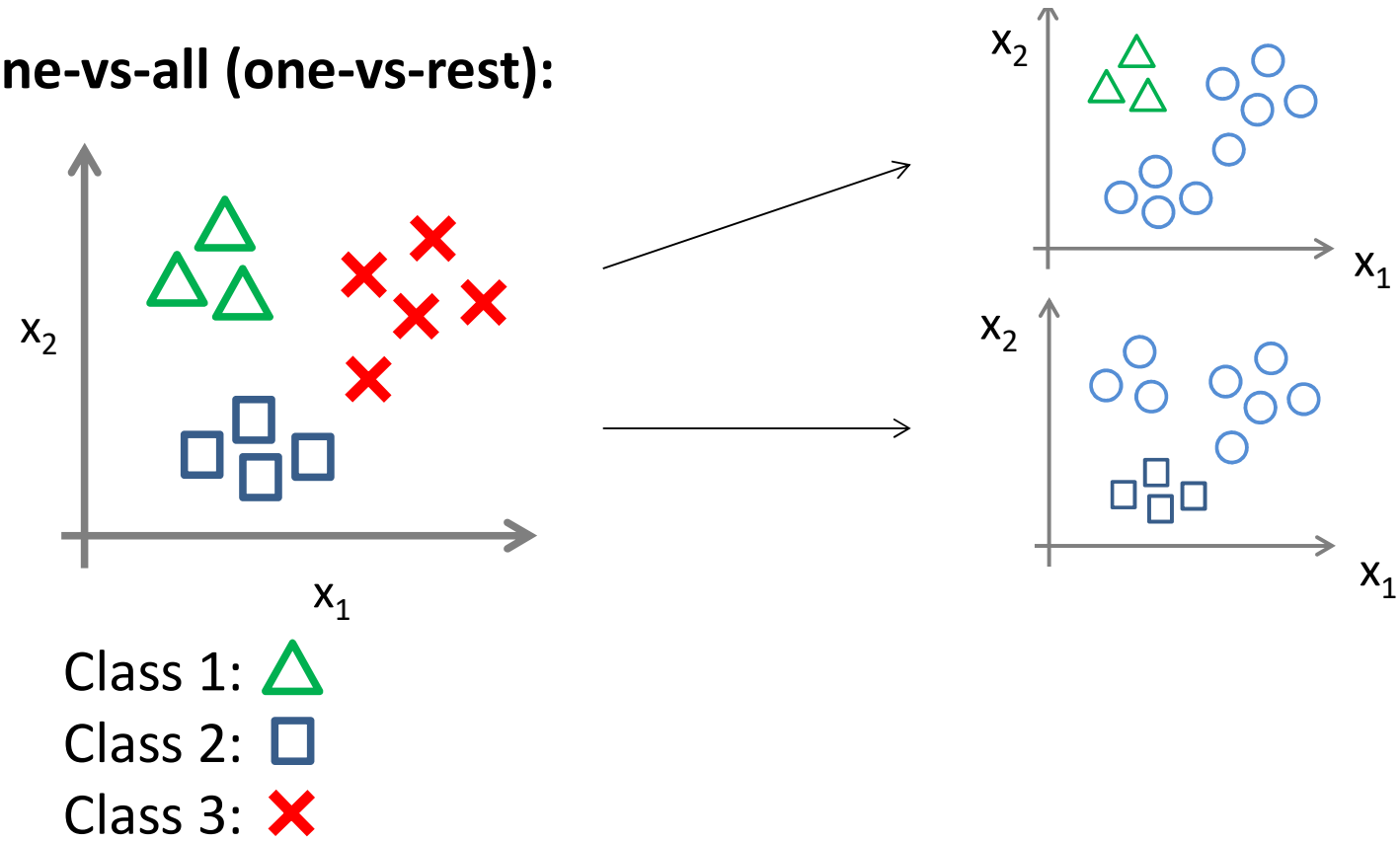
One-vs-all (one-vs-rest):



Andrew Ng, Machine Learning, Coursera course

Multiclass classification

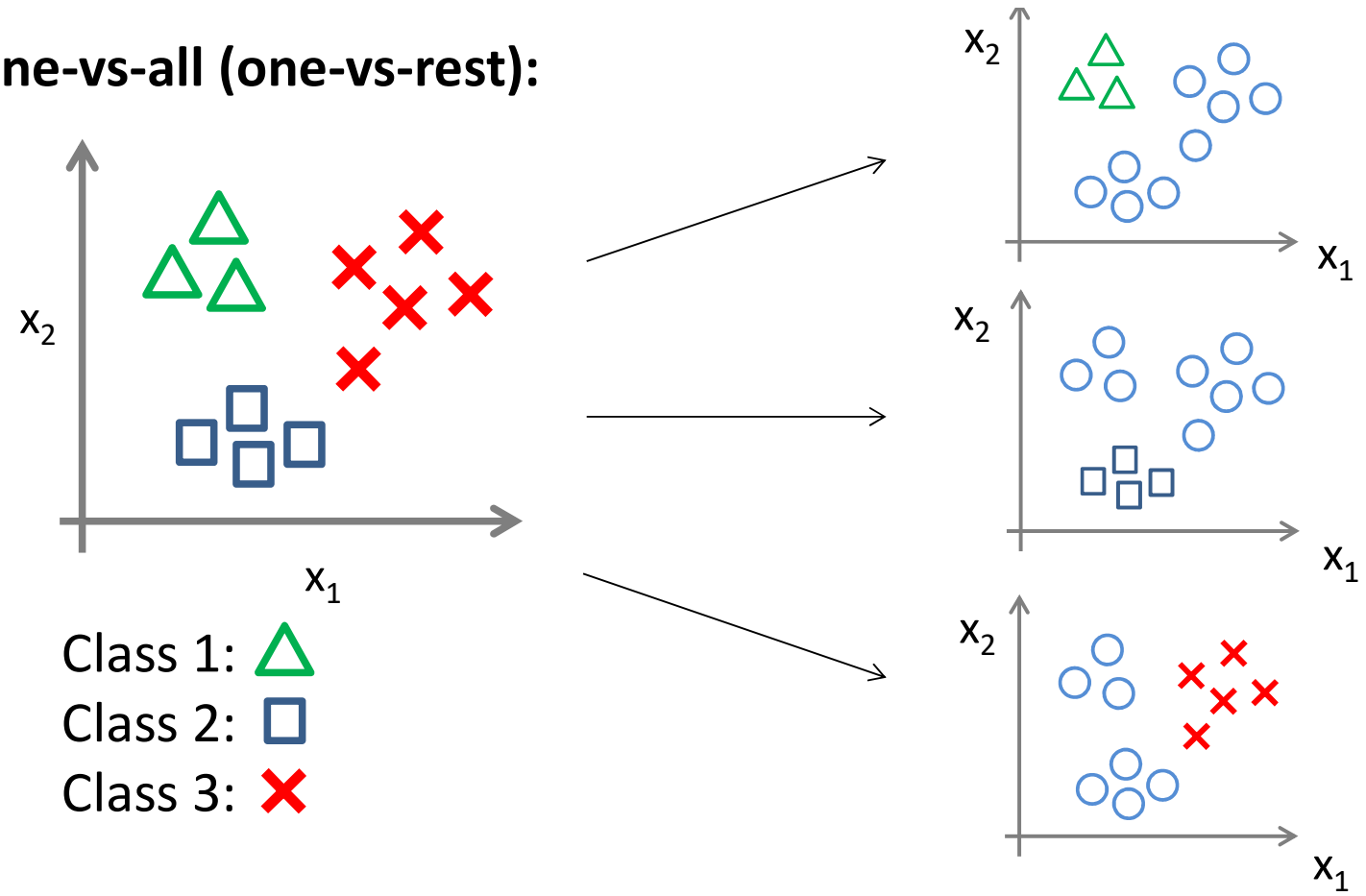
One-vs-all (one-vs-rest):



Andrew Ng, Machine Learning, Coursera course

Multiclass classification

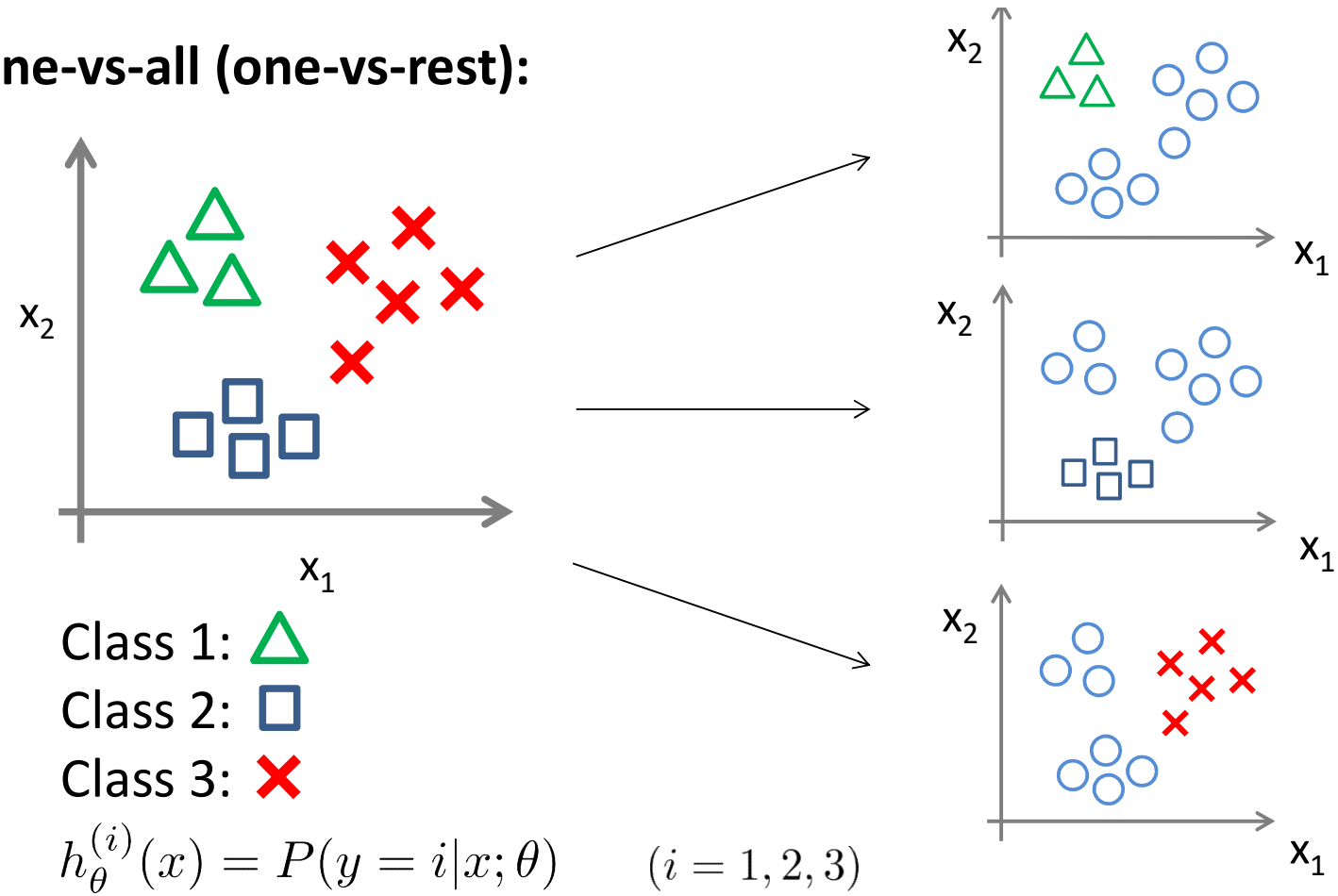
One-vs-all (one-vs-rest):



Andrew Ng, Machine Learning, Coursera course

Multiclass classification

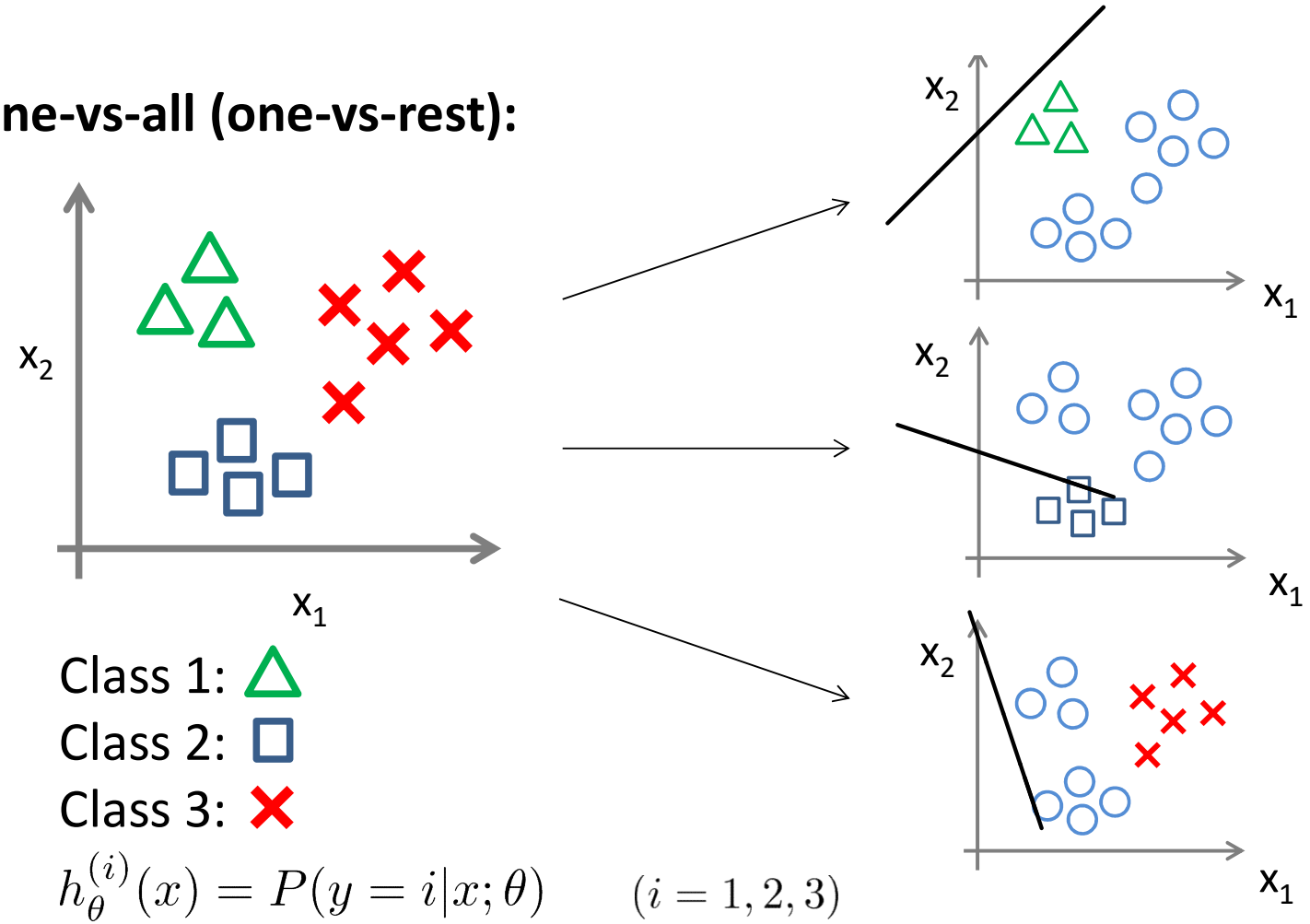
One-vs-all (one-vs-rest):



Andrew Ng, Machine Learning, Coursera course

Multiclass classification

One-vs-all (one-vs-rest):



Class 1: △

Class 2: □

Class 3: ×

$$h_\theta^{(i)}(x) = P(y = i|x; \theta) \qquad (i = 1, 2, 3)$$

Andrew Ng, Machine Learning, Coursera course

Multiclass classification

One-vs-all

Train a logistic regression classifier $h_\theta^{(i)}(x)$ for each class i to predict the probability that $y = i$.

Andrew Ng, Machine Learning, Coursera course

Multiclass classification

One-vs-all

Train a logistic regression classifier $h_\theta^{(i)}(x)$ for each class i to predict the probability that $y = i$.

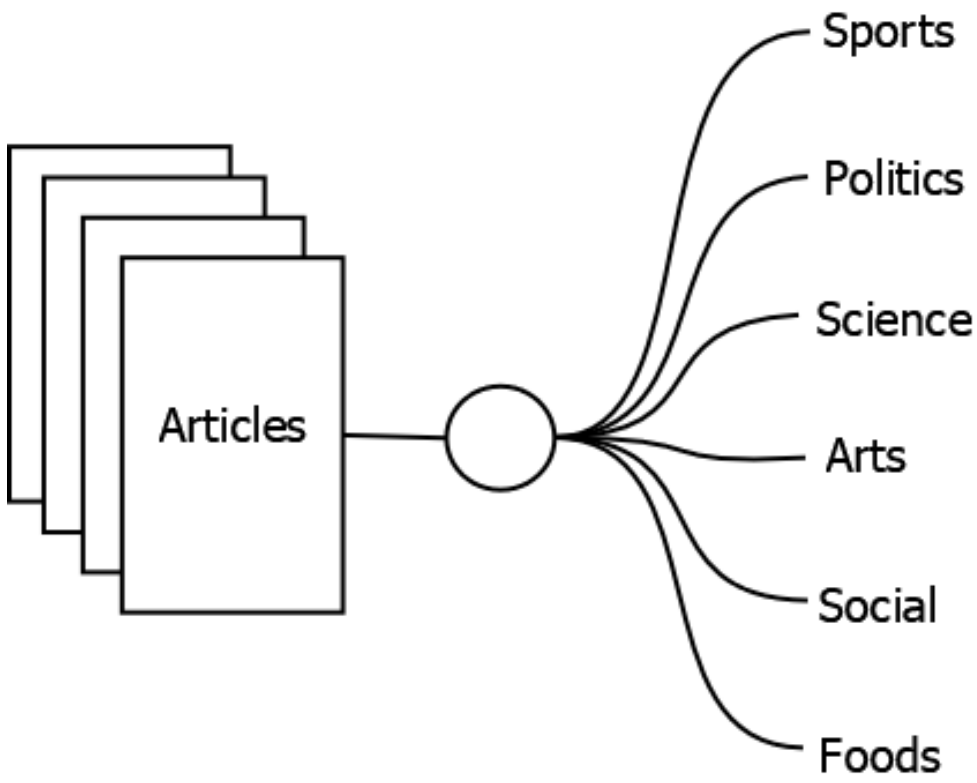
On a new input x, to make a prediction, pick the class i that maximizes

$$\max_i h_\theta^{(i)}(x)$$

Andrew Ng, Machine Learning, Coursera course

Content

Text classification

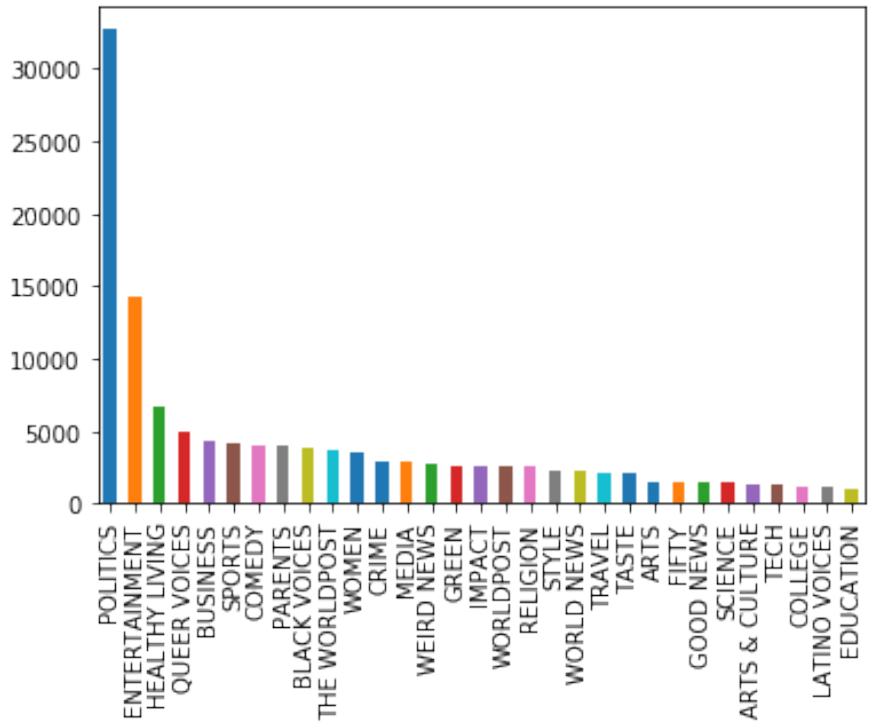


Applications

- Junk email filtering
- News topic classification
- Authorship attribution
- Sentiment analysis
- Genre classification
- Offensive language identification
- Language identification

Huffpost News Category Dataset

This dataset contains around 200k news headlines from the year 2012 to 2018 obtained from HuffPost.



https://www.kaggle.com/rmisra/news-category-dataset

Huffpost News Category Dataset

authors	category	date	headline	link	short_description
	ENTERTAINMENT	2014-10-09	Eek! Mario Lopez Admits He Never Loved Ex-Wife...	https://www.huffingtonpost.com/entry/mario-lop...	
Sarah Ruiz-Grossman	BLACK VOICES	2018-04-06	Protesters Demand Justice For Saheed Vassell, ...	https://www.huffingtonpost.com/entry/protests-...	"They murdered my son, and I want justice," hi...
Lee Moran	CRIME	2016-02-26	Police Hunt For Man Traveling The Midwest And ...	https://www.huffingtonpost.com/entry/rogaine-t...	The suspect, who is bald, reportedly has taken...
Michael Carosone, ContributorWriter, educator,...	QUEER VOICES	2014-09-07	She Inspired Me: My Tribute to Joan Rivers	https://www.huffingtonpost.com/entry/she-inspi...	I never met Joan Rivers; I always wanted to, b...
Amanda Pena	STYLE	2017-11-08	21 Affordable Holiday Gifts That Look Really E...	https://www.huffingtonpost.com/entry/affordabl...	Our idea of luxury won't break the bank.
Zahara Hill	BLACK VOICES	2017-03-06	Viola Davis Gives (Another) Moving Speech As H...	https://www.huffingtonpost.com/entry/viola-dav...	"I want people to be seen. I want them to feel...

Kavita Ganesan, Build Your First Text Classifier in Python with Logistic Regression
https://kavita-ganesan.com/news-classifier-with-logistic-regression-in-python

Procedure

- Text preprocessing
- Feature extraction
- Model training
- Model Application
- Evaluation

Text preprocessing

- Text cleaning (removing HTML/XML tags, figures, formula, etc.)
- Removing stop words
- Tokenization
- Stemming

Stop words

- A stop word is a commonly used word (such as “the”, “a”, “an”, “in”).
- Stop words are not helpful for text classification because they occur in almost all documents,
- Stop words are normally removed before applying a text classification algorithm.

Feature extraction

- Each document is represented as a vector in order to applying a classification algorithm.
- Each dimension of the input vector is called a feature.
- In text classification, the most straightforward idea is to use words as features.

doc_id	book	read	music	go
doc1	3	1	0	5
doc2	2	5	3	0
doc3	0	0	7	2

Weighting of words in document vectors

- **Term** - a word or a collocation.
- **Document** - a sequence of terms.
- **Corpus** - a set of documents.

Weighting of words in document vectors

Boolean weighting: $w_{ik} = \begin{cases} 1 & \text{if } f_{ik} > 0 \\ 0 & \text{Otherwise} \end{cases}$

Word frequency weighting: $w_{ik} = f_{ik}$

TF-IDF weighting: $w_{ik} = f_{ik} \times \log \frac{N}{n_i}$

i: word index

k: document index

f_{ik}: word frequency in a document

N: number of documents in the corpus

n_i: number of documents containing the word

Weighting of words in document vectors

- TF-IDF - short for *term frequency–inverse document frequency*, is a numerical statistic that is intended to reflect how important a word is to a document in a collection or corpus.
- The assumption behind the use of the inverse document frequency is: the more documents where a word (term) occurs, the less important the word is to that document.
- TF-IDF is proposed in information retrieval but also used in other areas including NLP.
- Which word weighting method is the best for text classification: no universal answer. It empirically depends on the data and the classification algorithm you use.

Algorithms

- Logistic regression
- Nearest neighbor
- Decision trees
- Support vector machines
- Neural networks

Further topics

- Feature selection
- Dimension reduction
- Document embeddings

Content

1. Machine Learning basics
2. Classification and logistic regression
3. Text Classification