EECS 445

Introduction to Machine Learning

Linear Classifiers and the Perceptron Algorithm

Prof. Kutty

Today’s Agenda

- Machine Learning
  - Supervised Learning
    - Linear Classifiers
      - Linear classifiers: what and why intuitively
      - Linear classifiers: algebraically and geometrically
      - Classification Algorithm: Perceptron
        - » notion of error
        - » ‘ideal’ case: linear separability
        - » without and with offset
      - Non-linear separability and linear classifiers

https://forms.gle/qzY3es1MQVTqreoY7

Announcements

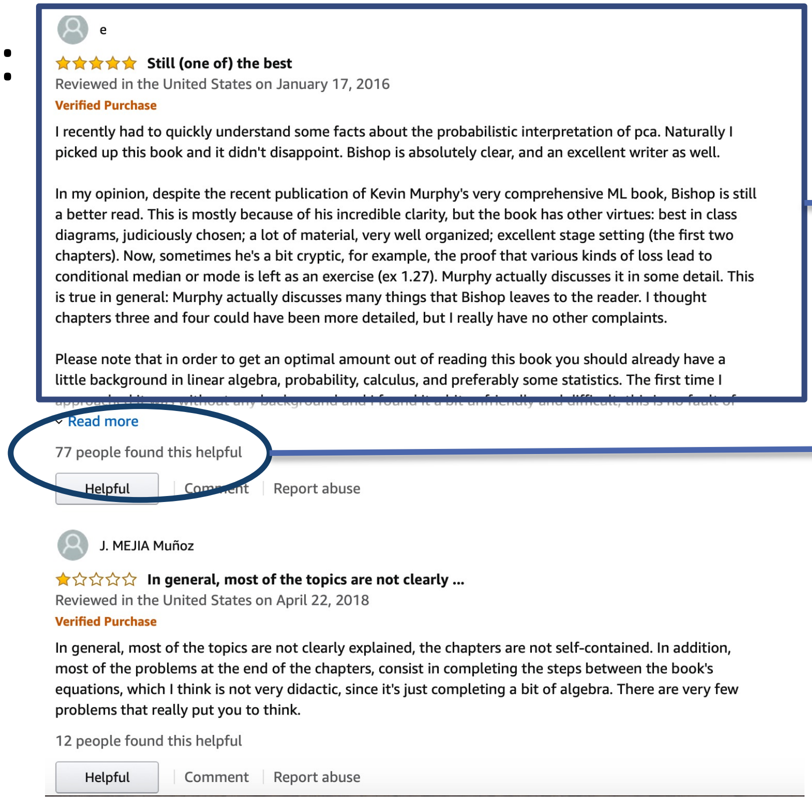
- Weekly announcements on canvas (include course highlights)
- We are aware of exam conflicts with STATS250 for the final and EECS376/EECS492 for the midterm
  - Please contact your other course instructors!
- Homework 1 is out – please be aware of due date. Check gradescope access.
  - python quickstart this Thursday (tomorrow)
- Office hours (including proffice hours) are available on calendar (multiple times: evenings, weekends; modalities: in person, hybrid, online)
  - expectations: review course material, be prepared to walk the staff through your attempt to get the most out of your time
  - Proffice hours will be focused on conceptual understanding

Supervised Learning

Linear Binary Classifier

Supervised Learning

Given:



e

★★★★★ **Still (one of) the best**

Reviewed in the United States on January 17, 2016

Verified Purchase

I recently had to quickly understand some facts about the probabilistic interpretation of pca. Naturally I picked up this book and it didn't disappoint. Bishop is absolutely clear, and an excellent writer as well.

In my opinion, despite the recent publication of Kevin Murphy's very comprehensive ML book, Bishop is still a better read. This is mostly because of his incredible clarity, but the book has other virtues: best in class diagrams, judiciously chosen; a lot of material, very well organized; excellent stage setting (the first two chapters). Now, sometimes he's a bit cryptic, for example, the proof that various kinds of loss lead to conditional median or mode is left as an exercise (ex 1.27). Murphy actually discusses it in some detail. This is true in general: Murphy actually discusses many things that Bishop leaves to the reader. I thought chapters three and four could have been more detailed, but I really have no other complaints.

Please note that in order to get an optimal amount out of reading this book you should already have a little background in linear algebra, probability, calculus, and preferably some statistics. The first time I approached it was without any background and I found it a bit unfriendly and difficult, this is no fault of

77 people found this helpful

J. MEJIA Muñoz

★☆☆☆☆ **In general, most of the topics are not clearly ...**

Reviewed in the United States on April 22, 2018

Verified Purchase

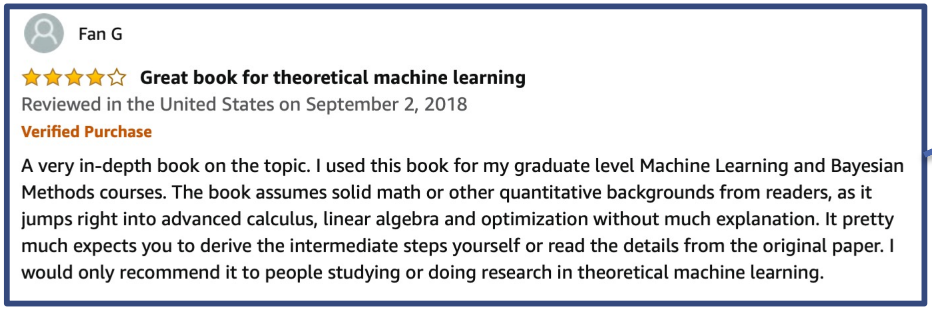
In general, most of the topics are not clearly explained, the chapters are not self-contained. In addition, most of the problems at the end of the chapters, consist in completing the steps between the book's equations, which I think is not very didactic, since it's just completing a bit of algebra. There are very few problems that really put you to think.

12 people found this helpful

features of the review

label of the review

Goal:



Fan G

★★★★☆ **Great book for theoretical machine learning**

Reviewed in the United States on September 2, 2018

Verified Purchase

A very in-depth book on the topic. I used this book for my graduate level Machine Learning and Bayesian Methods courses. The book assumes solid math or other quantitative backgrounds from readers, as it jumps right into advanced calculus, linear algebra and optimization without much explanation. It pretty much expects you to derive the intermediate steps yourself or read the details from the original paper. I would only recommend it to people studying or doing research in theoretical machine learning.

features of the review

predict the label of the review

Supervised Learning - Classification

- Labeled Dataset:
  - Features: star rating (1 – 5 stars),
    length of review (max length of 200 words).
  - Labels: helpful/unhelpful

(fractional) star rating	(fractional) review length	helpful
0.6	0.7	+
0.2	0.2	-
0.8	0.2	-
0.2	0.9	+
0.6	0.4	?

Binary Classifier

Problem: predict whether a (new unlabeled) review is helpful (+) or unhelpful (-)

Classification as an ML problem

1. Feature vector
   - Features are **statistics** or **attributes** that describe the data.
   - Represent data in terms of vectors.

$\bar{x}^{(i)} \in \mathbb{R}^2$ or

$\bar{x}^{(i)} \in \underbrace{[0,1] \times [0,1]}_{[0,1]^2}$

n training datapoints → ith datapoint

ith datapoint $\bar{x}^{(i)}$

$\bar{x}$ → represents a vector

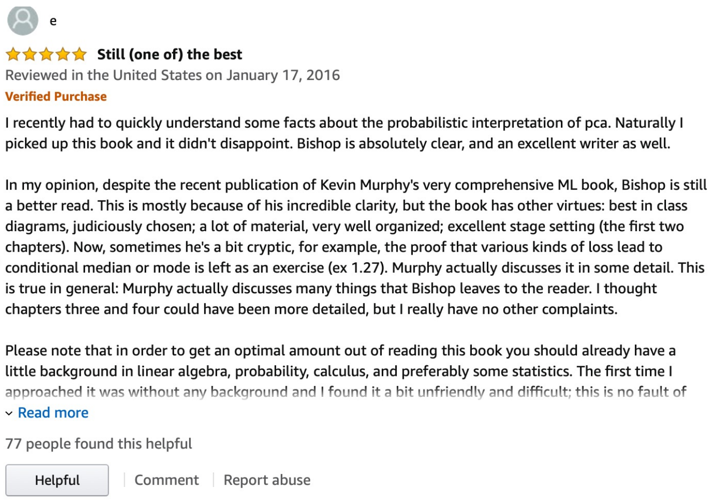
1	0.6	0.7	+
2	0.2	0.2	-
3	1	0.9	+
	0.2	0.9	-
	0.6	0.2	?

e.g.,

$\bar{x}^{(3)} = \begin{bmatrix} 1 \\ 0.9 \end{bmatrix}$

convention: column vector

star rating (as a fraction of 5 stars);
length of review (as a fraction of 200 words)



e

★★★★★ **Still (one of) the best**

Reviewed in the United States on January 17, 2016

Verified Purchase

I recently had to quickly understand some facts about the probabilistic interpretation of pca. Naturally I picked up this book and it didn't disappoint. Bishop is absolutely clear, and an excellent writer as well.

In my opinion, despite the recent publication of Kevin Murphy's very comprehensive ML book, Bishop is still a better read. This is mostly because of his incredible clarity, but the book has other virtues: best in class diagrams, judiciously chosen; a lot of material, very well organized; excellent stage setting (the first two chapters). Now, sometimes he's a bit cryptic, for example, the proof that various kinds of loss lead to conditional median or mode is left as an exercise (ex 1.27). Murphy actually discusses it in some detail. This is true in general: Murphy actually discusses many things that Bishop leaves to the reader. I thought chapters three and four could have been more detailed, but I really have no other complaints.

Please note that in order to get an optimal amount out of reading this book you should already have a little background in linear algebra, probability, calculus, and preferably some statistics. The first time I approached it was without any background and I found it a bit unfriendly and difficult; this is no fault of

In general

$\bar{x}^{(i)} \in \mathcal{X}$

↳ feature space

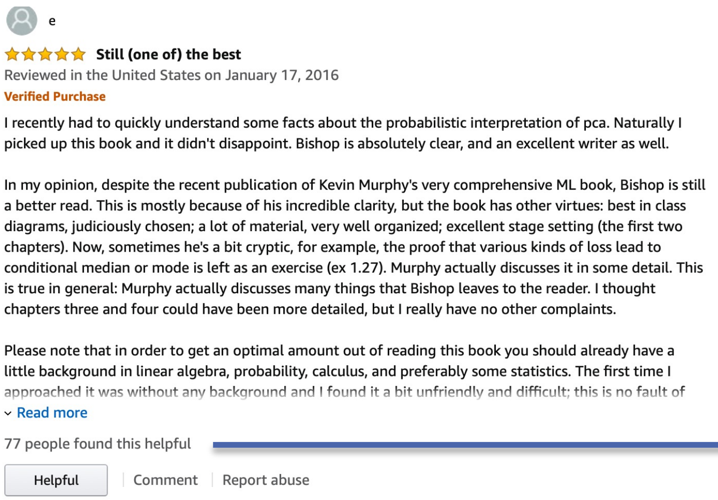
$\bar{x}^{(i)} \in \mathbb{R}^d$

d-dimensional reals

Classification as an ML problem

1. Feature vector
   - Features are **statistics** or **attributes** that describe the data.
   - Represent data in terms of vectors.

0.6	0.7	+
0.2	0.2	-
1	0.9	+
0.2	0.9	-
0.6	0.2	?



e

★★★★★ **Still (one of) the best**

Reviewed in the United States on January 17, 2016

Verified Purchase

I recently had to quickly understand some facts about the probabilistic interpretation of pca. Naturally I picked up this book and it didn't disappoint. Bishop is absolutely clear, and an excellent writer as well.

In my opinion, despite the recent publication of Kevin Murphy's very comprehensive ML book, Bishop is still a better read. This is mostly because of his incredible clarity, but the book has other virtues: best in class diagrams, judiciously chosen; a lot of material, very well organized; excellent stage setting (the first two chapters). Now, sometimes he's a bit cryptic, for example, the proof that various kinds of loss lead to conditional median or mode is left as an exercise (ex 1.27). Murphy actually discusses it in some detail. This is true in general: Murphy actually discusses many things that Bishop leaves to the reader. I thought chapters three and four could have been more detailed, but I really have no other complaints.

Please note that in order to get an optimal amount out of reading this book you should already have a little background in linear algebra, probability, calculus, and preferably some statistics. The first time I approached it was without any background and I found it a bit unfriendly and difficult; this is no fault of

77 people found this helpful

≥ 10 people then + else -

2. Labels

In general

$y^{(i)} \in \{+1, -1\}$

$y^{(i)} \in \mathcal{Y}$

label space

Classification as an ML problem

3. Training dataset

0.6	0.7	+
0.2	0.2	-
0.8	0.2	-
0.2	0.9	+
0.6	0.2	?

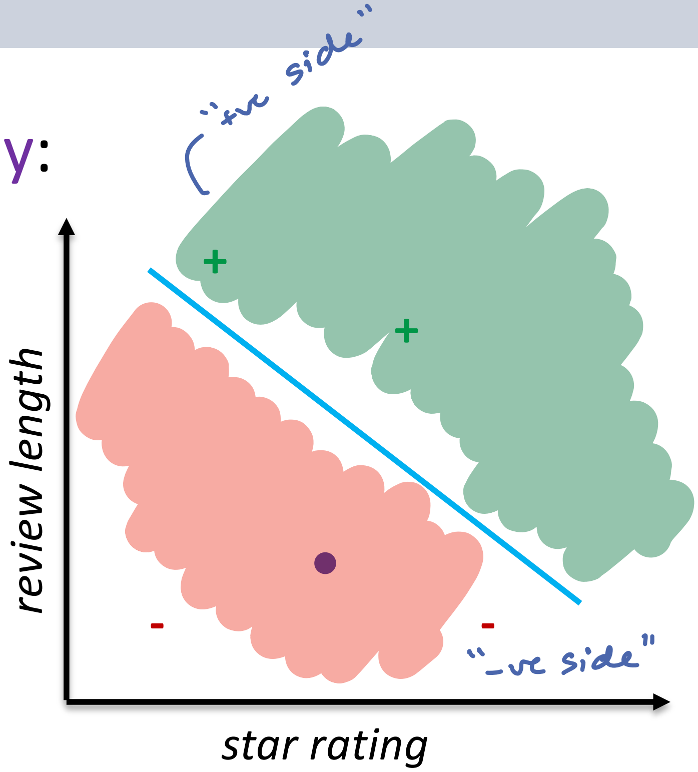
In general

$$S_n = \{(\bar{x}^{(i)}, y^{(i)})\}_{i=1}^{n}$$

This is a Supervised Learning problem

Supervised Learning - Classification

- Geometrically:



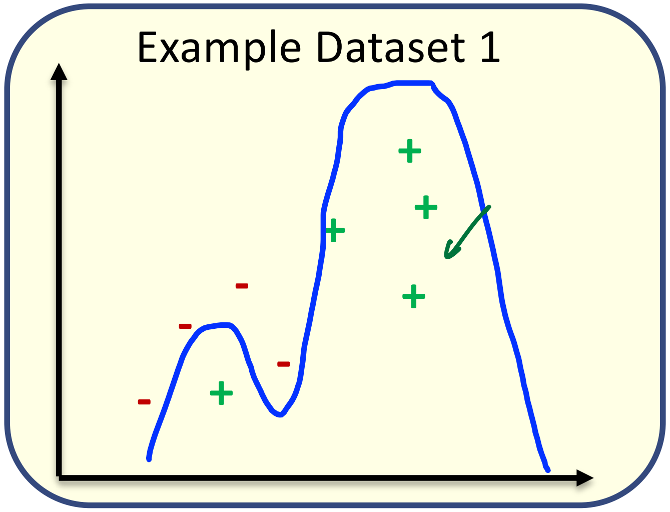
Star rating	Review length	label
0.6	0.7	+
0.2	0.2	-
0.8	0.2	-
0.2	0.9	+
0.6	0.4	?

- **Goal**: Learn a linear decision boundary

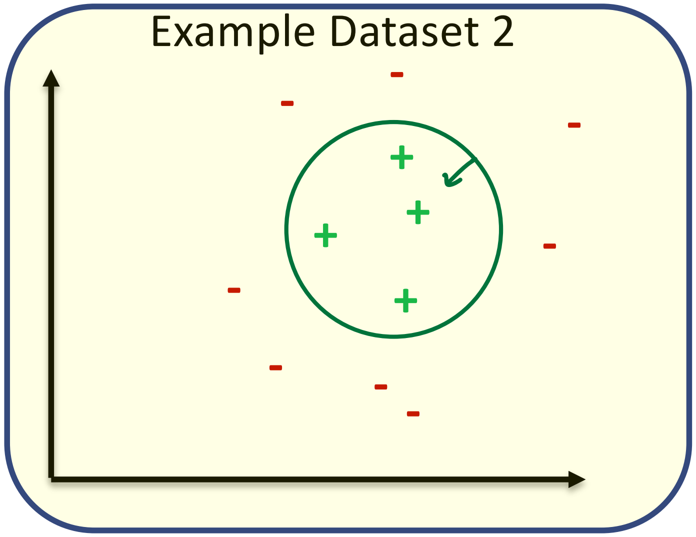
Why a linear decision boundary?

Why a linear decision boundary?

- Let $\mathcal{H}$ be the set of classifiers under consideration
  - e.g., $\mathcal{H}$ is the set of all hyperplanes (where the ambient space is $\mathbb{R}^d$)



- Too many choices not always a good thing
  - May lead to overfitting
- Solution?
  - Constrain possible choices i.e., $\mathcal{H}$



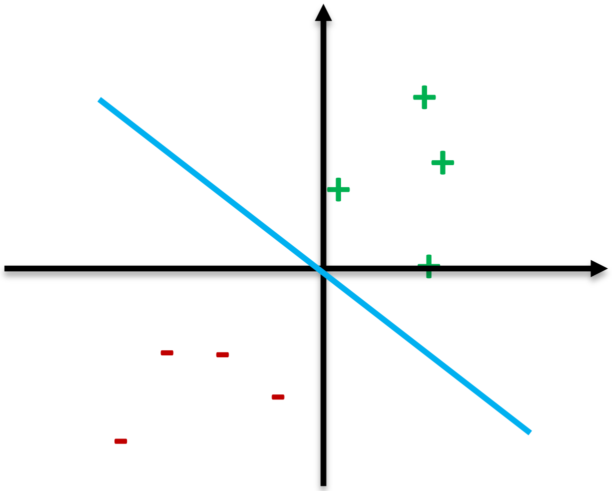
- Caution!
  - $\mathcal{H}$ cannot be too constrained either
  - May lead to underfitting

- This problem is called model selection

Linear Classifier

Goal: Learn a linear decision boundary

i.e., constrain possible choices $\mathcal{H}$ to hyperplanes



simplifying assumptions:

- constrain $\mathcal{H}$ to be the set of all hyperplanes that go through the origin
  - e.g., in $\mathbb{R}^2$ this is the set of lines that go through the origin

- constrain problem to datasets that are linearly separable

Linear Classifiers

Linear models are powerful (more than they may appear to be now)

https://forms.gle/qzY3es1MQVTqreoY7

Hyperplanes in $\mathbb{R}^d$

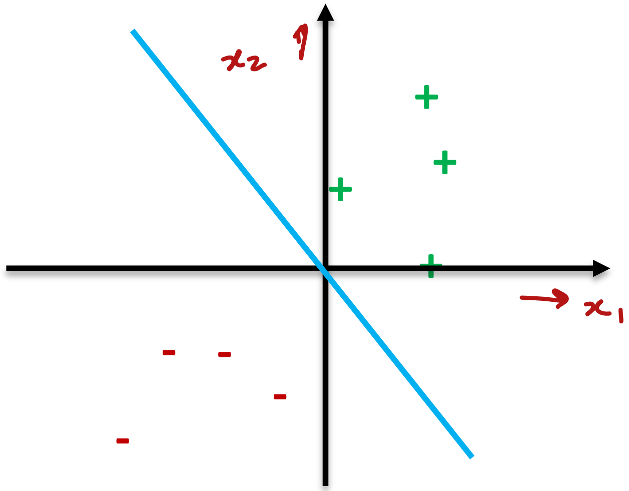
Definition: A *hyperplane* in $\mathbb{R}^d$
can be specified by parameter vector $\bar{\theta} \in \mathbb{R}^d$ and offset $b \in \mathbb{R}$

It is the set of points $\bar{\boldsymbol{x}} \in \mathbb{R}^d$ such that $\bar{\theta} \cdot \bar{\boldsymbol{x}} + \boldsymbol{b} = 0$

$$\bar{\theta} \cdot \bar{\boldsymbol{x}} = \boldsymbol{\theta}_1 \, \boldsymbol{x}_1 + \cdots + \boldsymbol{\theta}_d \, \boldsymbol{x}_d = \sum_{i=1}^{d} \boldsymbol{\theta}_i \, \boldsymbol{x}_i$$

$$\bar{x}^{(i)} = \begin{bmatrix} x_1^{(i)} \\ x_2^{(i)} \end{bmatrix}$$

$\bar{\boldsymbol{\theta}}$ is orthogonal to the hyperplane $\bar{\boldsymbol{\theta}} \cdot \bar{x} = 0$



in $\mathbb{R}^2$ this is the set of lines

$$y = \overset{\text{slope}}{m}x + \underset{\text{y intercept}}{b}$$

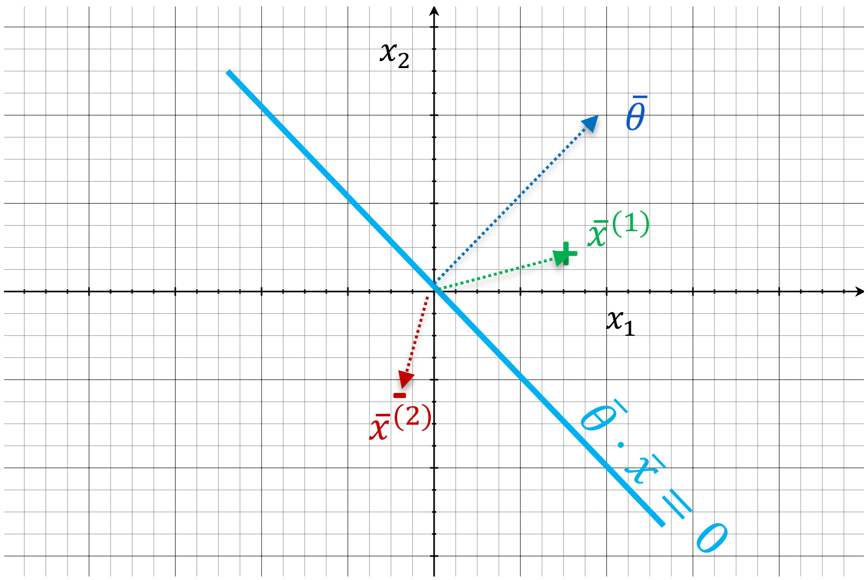
$$\theta_1 x_1 + \theta_2 x_2 + b = 0$$

$$x_2 = -\frac{\theta_1}{\theta_2} x_1 - \frac{b}{\theta_2}$$

Linear Classification in 2D

The hyperplane $\bar{\boldsymbol{\theta}} \cdot \bar{x} = 0$ can be described by the vector $\bar{\boldsymbol{\theta}}$

$\bar{\boldsymbol{\theta}}$ is orthogonal to the hyperplane $\bar{\boldsymbol{\theta}} \cdot \bar{x} = 0$



e.g., $\bar{\theta} = [10{,}10]^T$

① $\bar{\boldsymbol{\theta}} \cdot \bar{x}^{(1)} = \begin{bmatrix} 10 \\ 10 \end{bmatrix} \cdot \begin{bmatrix} x_1^{(1)} \\ x_2^{(1)} \end{bmatrix} > 0$

② $\bar{\boldsymbol{\theta}} \cdot \bar{x}^{(2)} < 0$

Claim:

$\bar{\theta} \cdot \bar{x} > 0$ → lie on the same side of the decision boundary

$\bar{\theta} \cdot \bar{x} < 0$ → lie on opposite sides of the decision boundary

$\bar{\theta} \cdot \bar{x} = 0$

↳ if datapoint $\bar{x}$ lies on the decision boundary

Recall from Linear Algebra

for $\bar{\theta}$, $\bar{\boldsymbol{x}} \in \boldsymbol{R}^{\boldsymbol{d}}$

- L2 norm of $\bar{\theta}$

$$\|\bar{\theta}\|_{\mathbf{2}}^{\mathbf{2}} = \boldsymbol{\theta}_{\mathbf{1}}^{\mathbf{2}} + \cdots + \boldsymbol{\theta}_{\boldsymbol{d}}^{\mathbf{2}}$$

- dot product

$$\bar{\theta} \cdot \bar{\boldsymbol{x}} = \boldsymbol{\theta}_{\mathbf{1}}\, \boldsymbol{x}_{\mathbf{1}} + \cdots + \boldsymbol{\theta}_{\boldsymbol{d}}\, \boldsymbol{x}_{\boldsymbol{d}} = \sum_{\boldsymbol{i}=\mathbf{1}}^{\boldsymbol{d}} \boldsymbol{\theta}_{\boldsymbol{i}}\ \boldsymbol{x}_{\boldsymbol{i}} = \|\bar{\theta}\|_{\mathbf{2}}\ \ \|\bar{x}\|_{\mathbf{2}}\ \cos \boldsymbol{\alpha}$$

Case 1: $0° \leq \alpha < 90°$ and $270° < \alpha \leq 360°$

$$\bar{\theta} \cdot \bar{\boldsymbol{x}} > \mathbf{0}$$

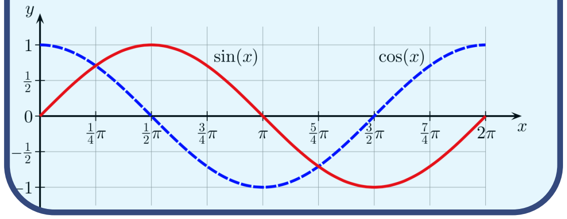
Case 2: $90° < \alpha < 270°$

$$\bar{\theta} \cdot \bar{\boldsymbol{x}} < \mathbf{0}$$

Case 3: $\alpha \in \{90°, 270°\}$

$$\bar{\theta} \cdot \bar{\boldsymbol{x}} = 0$$

recall:

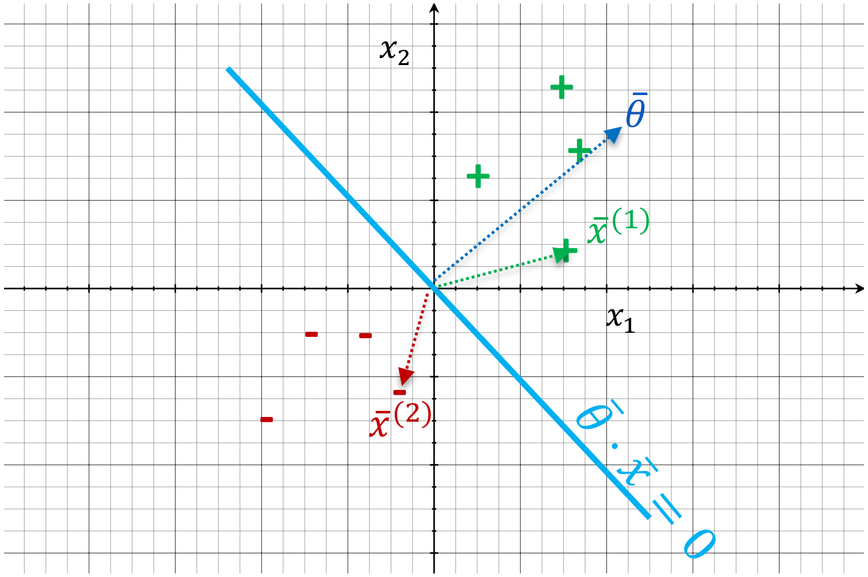


By Geek3 - Own work, CC BY 3.0,
https://commons.wikimedia.org/w/index.php?curid=10904768

Linear Classification in 2D

The hyperplane $\bar{\boldsymbol{\theta}} \cdot \bar{x} = 0$ can be described by the vector $\bar{\boldsymbol{\theta}}$

$\bar{\boldsymbol{\theta}}$ is orthogonal to the hyperplane $\bar{\boldsymbol{\theta}} \cdot \bar{x} = 0$



Goal:

Find a vector $\bar{\boldsymbol{\theta}}$ so that all training datapoints are correctly classified*

This suggests a natural way to define a linear classifier;
given a hyperplane with parameter $\bar{\boldsymbol{\theta}}$ the prediction for a datapoint $\bar{x}$ is :

$$h(\bar{x}; \bar{\theta}) = \text{sign}(\bar{\theta} \cdot \bar{x})$$

Linear Classification in 2D

Goal: Learn a linear decision boundary *through the origin*

More precisely,
learn $\bar{\theta}$ such that the hyperplane $\bar{\theta} \cdot \bar{x} = 0$ separates the positive from the negative examples

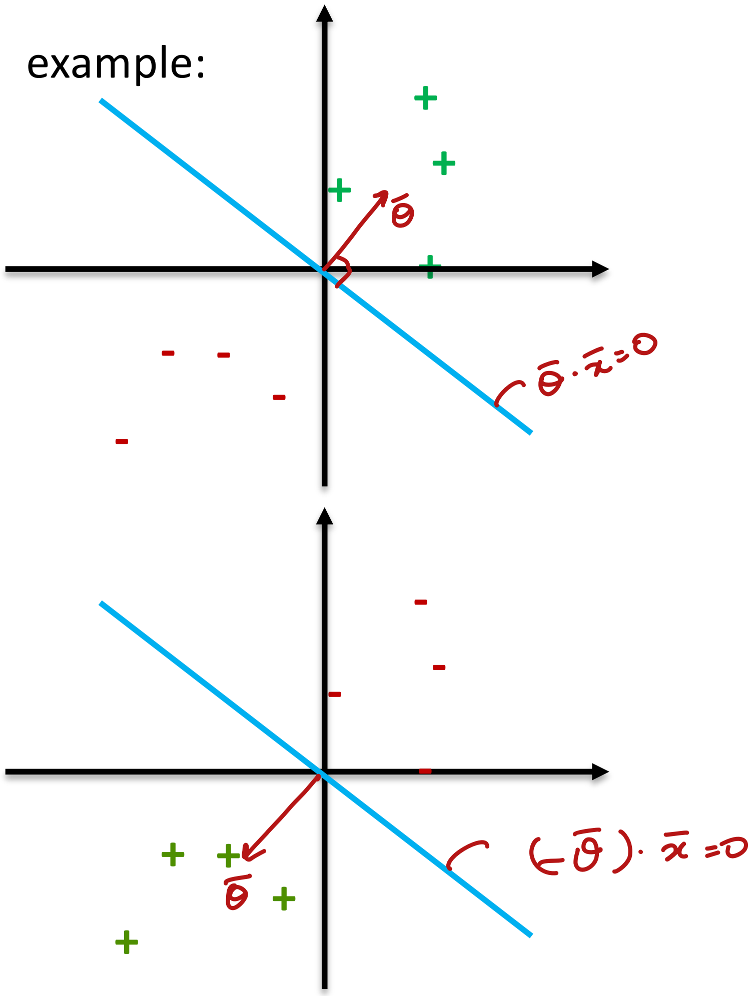
Define linear classifier as

$$h(\bar{x}; \bar{\theta}) = \text{sign}(\bar{\theta} \cdot \bar{x})$$

The choice of $\bar{\theta}$ determines:

1. orientation of the hyperplane
   e.g., in $\mathbb{R}^2$

2. the *predicted* class label


example:
+
+
+
+
$\bar{\theta}$
$\bar{\theta} \cdot \bar{x} = 0$
-
-
-
-
-
-
-
-
+
+
$\bar{\theta}$
+
+
$(-\bar{\theta}) \cdot \bar{x} = 0$

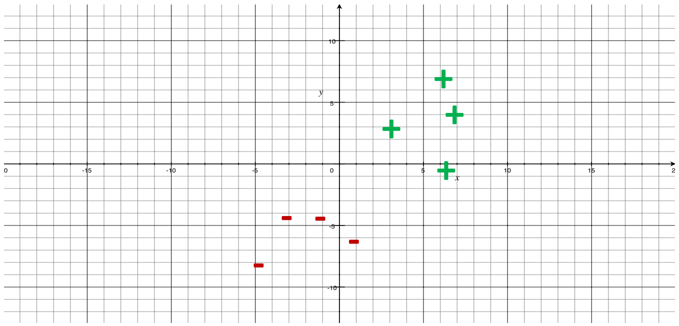
Note: In d dimensions this defines a hyperplane that goes through the origin

Linear separability: definition

without offset

Given training examples

$$S_n = \{(\bar{x}^{(i)}, y^{(i)})\}_{i=1}^{n}$$



linearly separable

we say the data are linearly separable

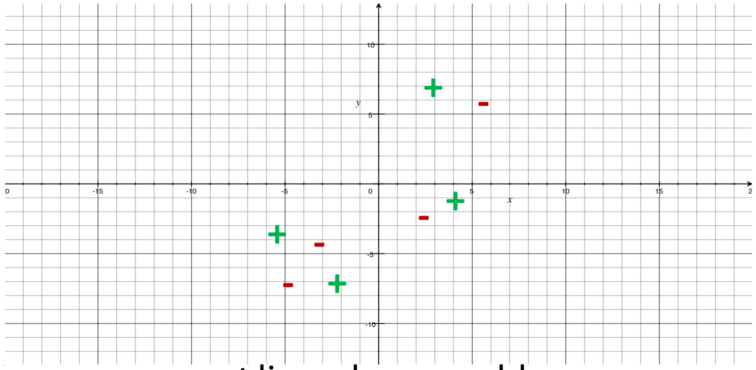
if there exists $\bar{\theta} \in \mathbb{R}^d$

such that $\text{sign}(\bar{\theta} \cdot \bar{x}^{(i)}) = y^{(i)}$

for $i = 1, \ldots, n$. We refer to

$\bar{\theta} \cdot \bar{x} = 0$

as a separating hyperplane



not linearly separable

Selecting $\bar{\theta}$

- Find $\bar{\theta}$ that works well on training data $S_n = \{(\bar{x}^{(i)}, y^{(i)})\}_{i=1}^{n}$
- Training error

$$E_n(\bar{\theta}) \overset{\in [0,1]}{=} \frac{1}{n}\sum_{i=1}^{n} [\![y^{(i)} \neq h(\bar{x}^{(i)};\bar{\theta})]\!]$$

$$= \frac{1}{n}\sum_{i=1}^{n} [\![y^{(i)}(\bar{\theta}\cdot\bar{x}^{(i)}) \leq 0]\!]$$

Indicator function:

$$[\![p]\!] = \begin{cases} 1, & p \text{ is true} \\ 0, & \text{otherwise} \end{cases}$$

Notice that

True label $y^{(i)}$	Predicted label $h(\bar{x}^{(i)};\bar{\theta})$	$y^{(i)}h(\bar{x}^{(i)};\bar{\theta})$
+1	+1	+1
+1	-1	-1
-1	+1	-1
-1	-1	+1

$h(\bar{x}^{(i)};\bar{\theta}) = \text{sign}(\bar{\theta}\cdot\bar{x}^{(i)})$

$y^{(i)}\,\bar{\theta}\cdot\bar{x}^{(i)}$ < 0 misclassified

> 0 correctly classified

$= 0$ misclassified ($\bar{x}^{(i)}$ lies on the decision boundary)

Training Error gives us the fraction of misclassified points in a training dataset

Linear Classifier

Given: training data $S_n = \{(\bar{x}^{(i)}, y^{(i)})\}_{i=1}^{n}$

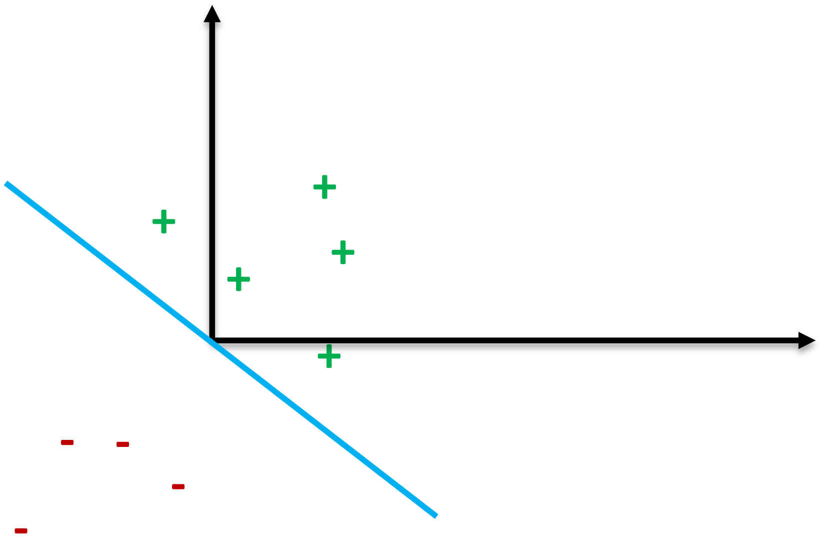
$\bar{x}^{(i)} \in \mathbb{R}^d$

$y^{(i)} \in \{+1, -1\}$

Goal: Learn a linear decision boundary (through the origin) that minimizes training error

$$h(\bar{x}; \bar{\theta}) = \text{sign}(\bar{\theta} \cdot \bar{x})$$

$\bar{\theta} \in \mathbb{R}^d$



$$E_n(\bar{\theta}) = \frac{1}{n}\sum_{i=1}^{n} [\![y^{(i)} \neq h(\bar{x}^{(i)}; \bar{\theta})]\!]$$

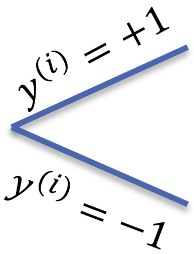
$$= \frac{1}{n}\sum_{i=1}^{n} [\![y^{(i)} \left(\bar{\theta} \cdot \bar{x}^{(i)}\right) \leq 0]\!]$$

simplifying assumptions: linear separability

The Perceptron Algorithm

Perceptron Algorithm

```
On input $S_n = \{(\bar{x}^{(i)}, y^{(i)})\}_{i=1}^{n}$
**Initialize** $k$ = 0, $\bar{\boldsymbol{\theta}}^{(\mathbf{0})}$ = $\bar{\mathbf{0}}$
**while** there exists a misclassified point
    **for** $i = 1, \dots, n$
        **if** $y^{(i)}(\bar{\theta}^{(k)} \cdot \bar{x}^{(i)}) \leq 0$
            $\bar{\theta}^{(k+1)} = \bar{\theta}^{(k)} + y^{(i)}\bar{x}^{(i)}$
            $k$++
```



$\bar{\theta}^{(k+1)} = \bar{\theta}^{(k)} + \bar{x}^{(i)}$

$\bar{\theta}^{(k+1)} = \bar{\theta}^{(k)} - \bar{x}^{(i)}$

If the data are linearly separable

then this algorithm finds a separating hyperplane

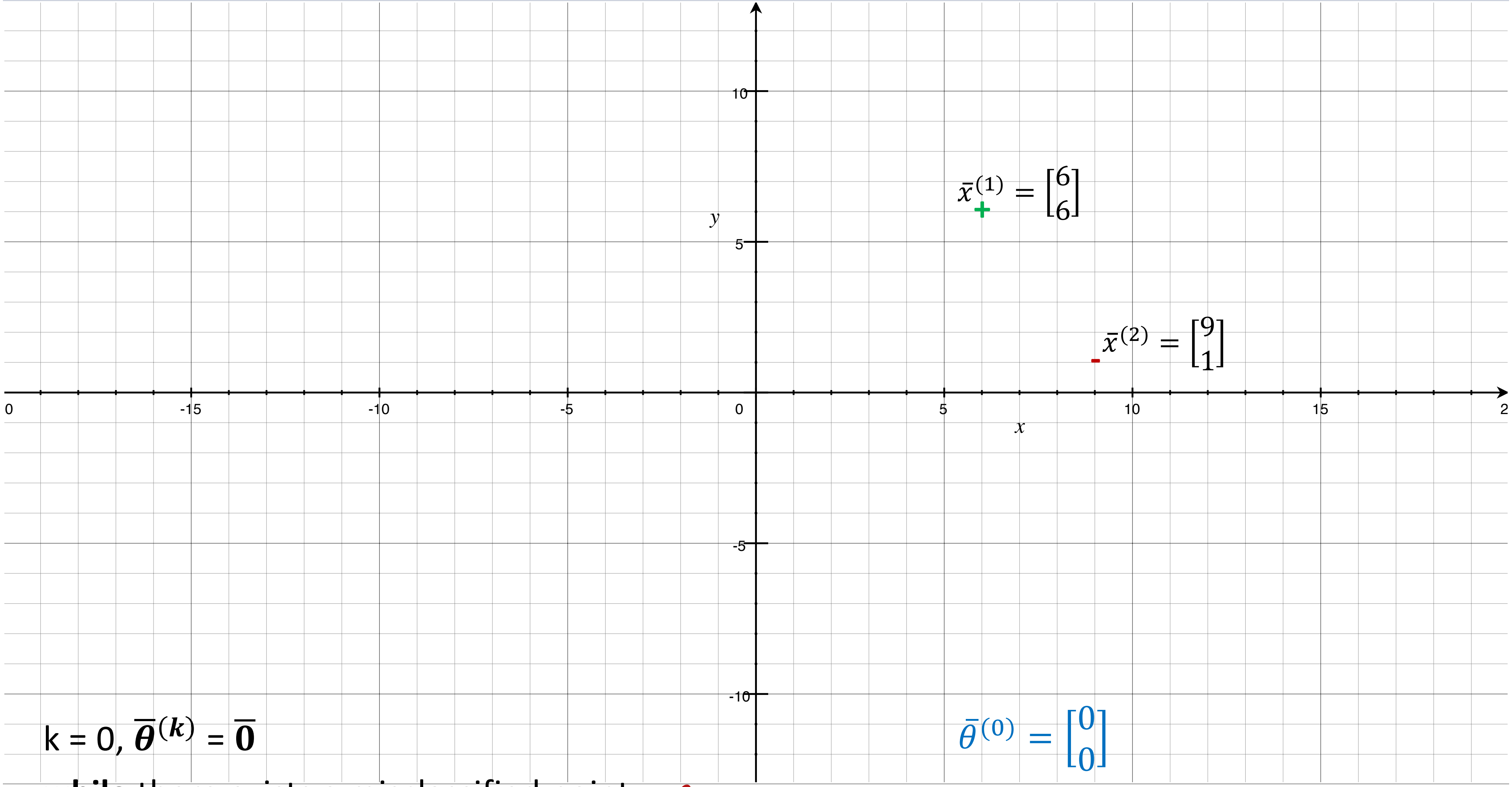
Notes and Observations

Note: the perceptron is actually a (type of) single layer Neural Network

  - building block of Deep Learning
  - more on this later in the course

- The perceptron algorithm
  - updates based on a single (misclassified) point at a time
  - moves the hyperplane in the 'right' direction based on that point

Theorem: The perceptron algorithm *converges* after a finite number of steps when the training examples are linearly separable

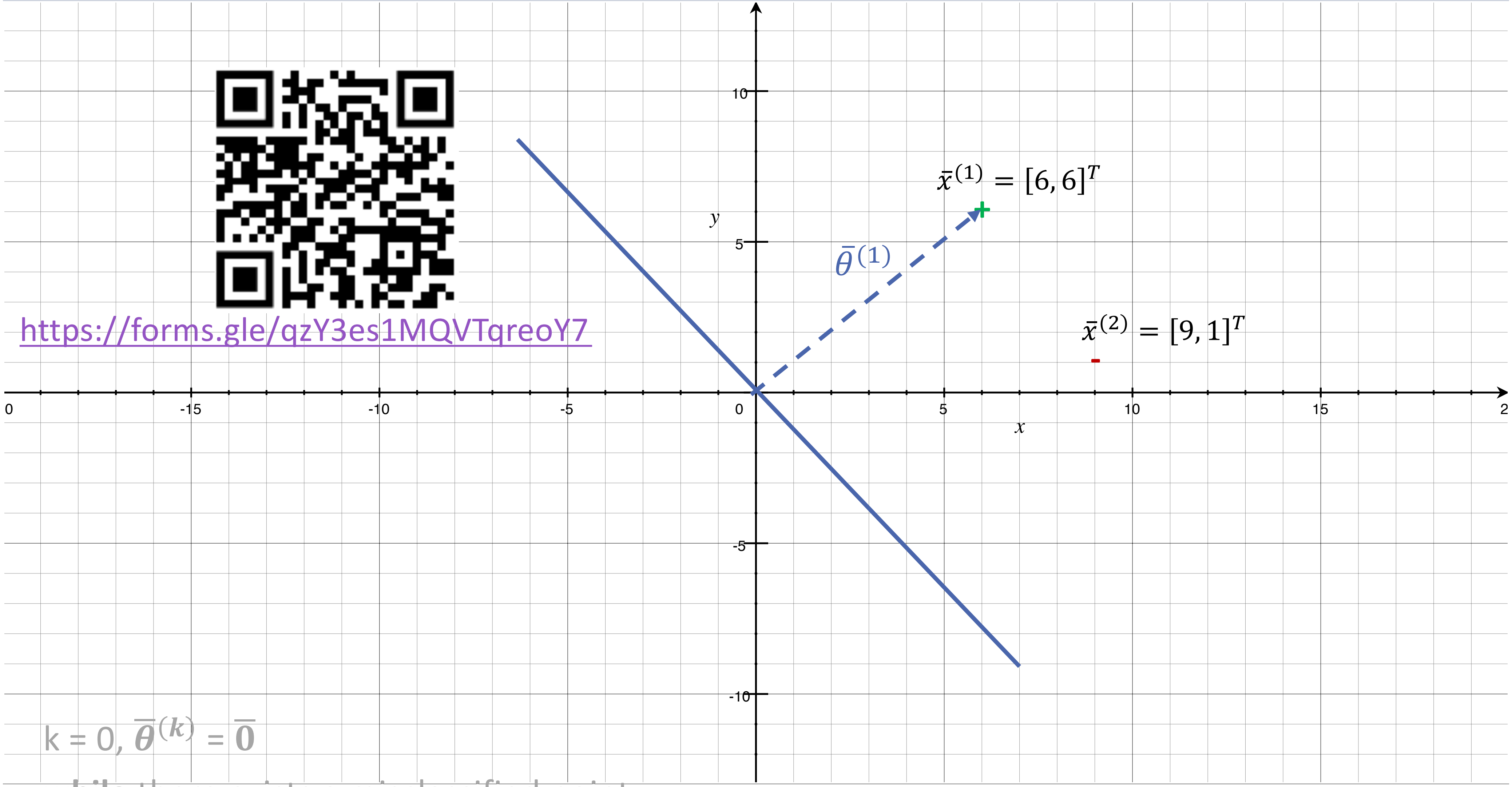


```
k = 0, $\overline{\boldsymbol{\theta}}^{(k)} = \overline{\mathbf{0}}$
while there exists a misclassified point
    for i = 1, …,n
        if $\boldsymbol{y}^{(i)}\left(\overline{\boldsymbol{\theta}}^{(k)} \cdot \overline{\boldsymbol{x}}^{(i)}\right) \leq \mathbf{0}$
            $\overline{\boldsymbol{\theta}}^{(k+1)} = \overline{\boldsymbol{\theta}}^{(k)} + \boldsymbol{y}^{(i)}\overline{\boldsymbol{x}}^{(i)}$
            k++
```

$$\overset{+1}{y^{(1)}}\left(\bar{\theta}^{(0)} \cdot \overset{\begin{bmatrix}6\\6\end{bmatrix}}{\bar{x}^{(1)}}\right) \le 0$$

$$\bar{\theta}^{(1)} = \bar{\theta}^{(0)} + \begin{bmatrix}6\\6\end{bmatrix} = \begin{bmatrix}6\\6\end{bmatrix}$$

	dimension 1	dimension 2	label
Datapoint 1	6	6	+
Datapoint 2	9	1	-



https://forms.gle/qzY3es1MQVTqreoY7

```
k = 0, $\overline{\boldsymbol{\theta}}^{(k)} = \overline{\mathbf{0}}$
while there exists a misclassified point
    for i = 1, …,n
        if $y^{(i)}\left(\overline{\boldsymbol{\theta}}^{(k)} \cdot \overline{x}^{(i)}\right) \leq 0$
            $\overline{\boldsymbol{\theta}}^{(k+1)} = \overline{\boldsymbol{\theta}}^{(k)} + y^{(i)}\overline{x}^{(i)}$
            k++
```

$\bar{\theta}^{(0)} = [0,0]^T$

$\bar{\theta}^{(2)}$ =

A. no more updates

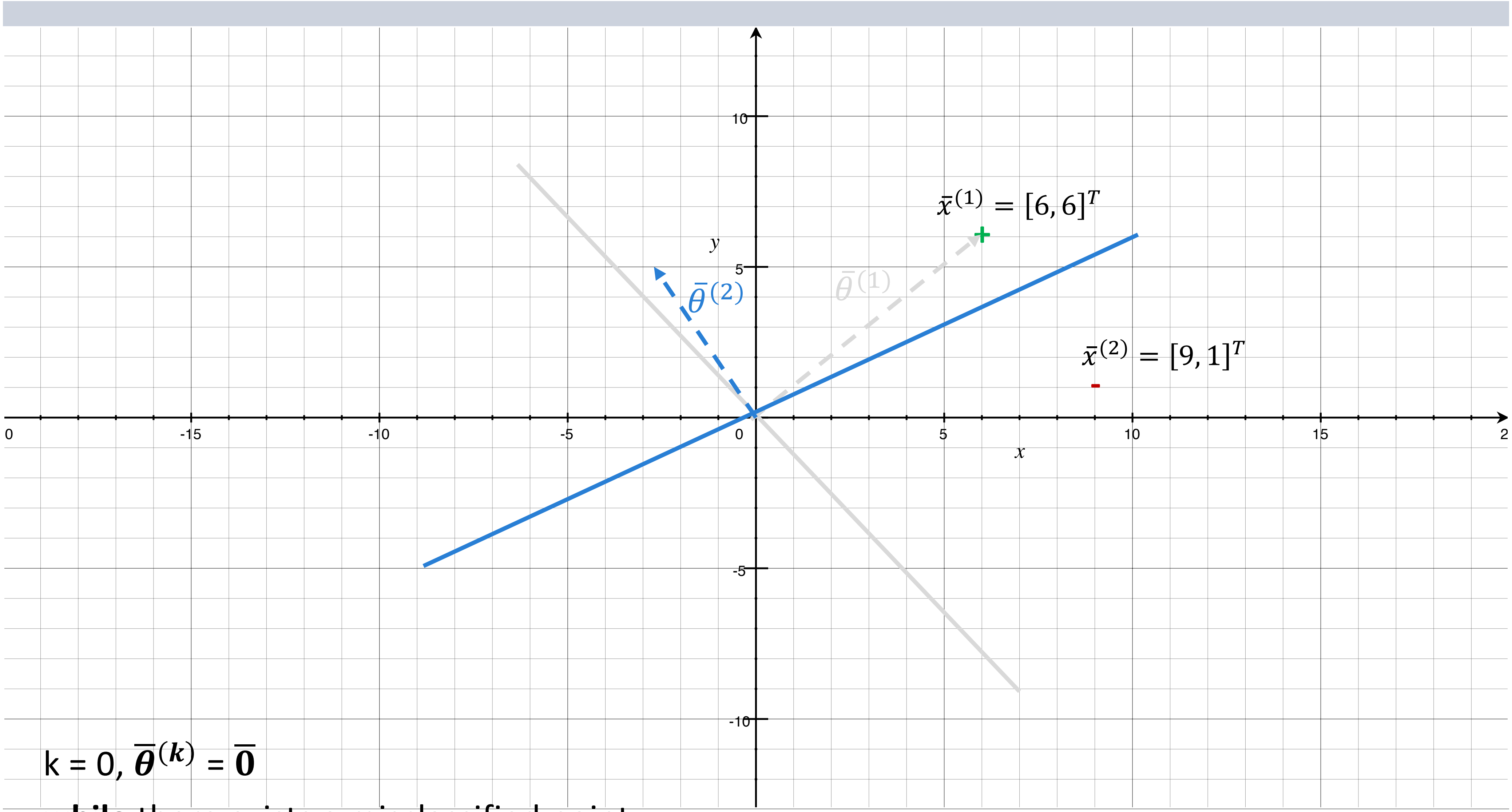
B. $[15, 7]^T$

C. $[-3, 5]^T$

D. unsure

$k = 0, i = 1$

$\bar{\theta}^{(1)} = \bar{\theta}^{(0)} + \bar{x}^{(1)} = [6,6]^T$



```
k = 0, $\overline{\boldsymbol{\theta}}^{(k)} = \overline{\mathbf{0}}$
while there exists a misclassified point
    for i = 1, …,n
        if $y^{(i)}\left(\overline{\boldsymbol{\theta}}^{(k)} \cdot \overline{\boldsymbol{x}}^{(i)}\right) \leq \mathbf{0}$
            $\overline{\boldsymbol{\theta}}^{(k+1)} = \overline{\boldsymbol{\theta}}^{(k)} + y^{(i)}\overline{\boldsymbol{x}}^{(i)}$
            k++
```

$\bar{\theta}^{(0)} = [0,0]^T$

$k = 0, i = 1$

$\bar{\theta}^{(1)} = \bar{\theta}^{(0)} + \bar{x}^{(1)} = [6,6]^T$

$k = 1, i = 2$

$\bar{\theta}^{(2)} = \bar{\theta}^{(1)} - \bar{x}^{(2)} = [-3,5]^T$