MIT · 6.036 | Introduction to Machine Learning (2020)

MIT 6.036(2020)· 课程资料包 @ShowMeAI

视频
中英双语字幕

课件
一键打包下载

笔记
官方笔记翻译

代码
作业项目解析

视频 · B 站 [扫码或点击链接]
https://www.bilibili.com/video/BV1y44y187wN

课件 & 代码 · 博客 [扫码或点击链接]
http://blog.showmeai.tech/mit-6.036

机器学习 循环神经 神经网络 感知器
特征构建 聚类 马尔可夫决策过程 网络
随机森林 决策树 逻辑回归 卷积神经网络 状态机

Awesome AI Courses Notes Cheatsheets 是 ShowMeAI 资料库的分支系列，覆盖最具知名度的 **TOP50+** 门 AI 课程，旨在为读者和学习者提供一整套高品质中文学习笔记和速查表。

点击课程名称，跳转至课程**资料包**页面，**一键下载**课程全部资料！

机器学习	深度学习	自然语言处理	计算机视觉
Stanford · CS229	Stanford · CS230	Stanford · CS224n	Stanford · CS231n
# Awesome AI Courses Notes Cheatsheets· 持续更新中			
知识图谱	**图机器学习**	**深度强化学习**	**自动驾驶**
Stanford · CS520	Stanford · CS224W	UCBerkeley · CS285	MIT · 6.S094

微信公众号

资料下载方式 2：扫码点击底部菜单栏

称为 **AI 内容创作者？** 回复 [添砖加瓦]

6.036/6.862: Introduction to Machine Learning

Lecture: starts Tuesdays 9:35am (Boston time zone)

Course website: introml.odl.mit.edu

Who's talking? Prof. Tamara Broderick

Questions? Ask on Discourse: discourse.odl.mit.edu

Materials: Will all be available at course website

Today's Plan

I. (More) logistics

II. Machine learning setup

III. Linear classifiers

6.036/6.862: Introduction to Machine Learning

Lecture: starts Tuesdays 9:35am (Boston time zone)

Course website: introml.odl.mit.edu

Who's talking? Prof. Tamara Broderick

Questions? Ask on Discourse: discourse.odl.mit.edu

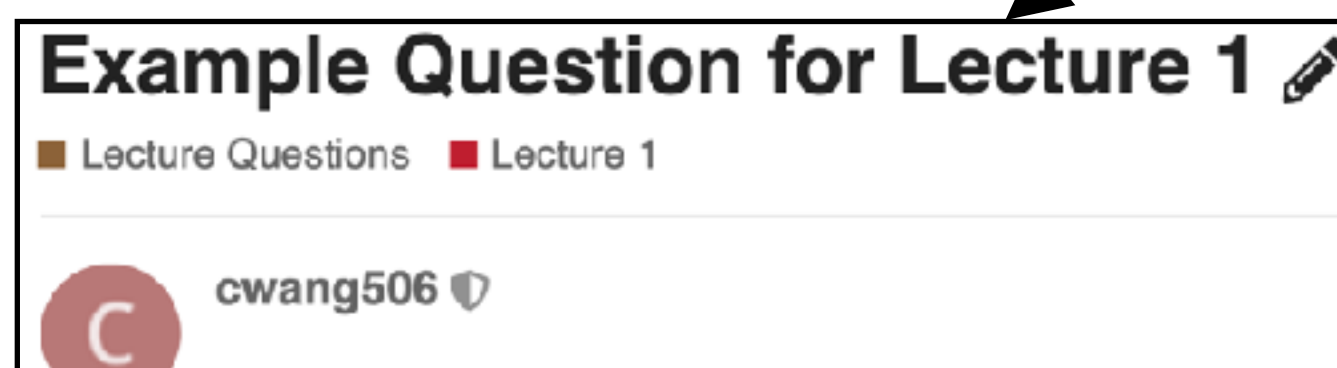
Materials: Will all be available at course website

Today's Plan

I. (More) logistics

II. Machine learning setup

III. Linear classifiers



(set "Lecture 1" category)

Is Introduction to Machine Learning (6.036/6.862) right for you?

Is Introduction to Machine Learning (6.036/6.862) right for you?

Computer Science Prerequisites

Is Introduction to Machine Learning (6.036/6.862) right for you?

Computer Science Prerequisites

- Python programming

Is Introduction to Machine Learning (6.036/6.862) right for you?

Computer Science Prerequisites

- Python programming
- Algorithms (read & understand pseudocode)

Is Introduction to Machine Learning (6.036/6.862) right for you?

Computer Science Prerequisites

- Python programming
- Algorithms (read & understand pseudocode)

Math Prerequisites

Is Introduction to Machine Learning (6.036/6.862) right for you?

Computer Science Prerequisites

- Python programming
- Algorithms (read & understand pseudocode)

Math Prerequisites

- Matrix manipulations (inverse, transpose, multiplication, etc.)

Is Introduction to Machine Learning (6.036/6.862) right for you?

Computer Science Prerequisites

- Python programming
- Algorithms (read & understand pseudocode)

Math Prerequisites

- Matrix manipulations (inverse, transpose, multiplication, etc.)
- Points and planes in dimension > 2

Is Introduction to Machine Learning (6.036/6.862) right for you?

Computer Science Prerequisites

- Python programming
- Algorithms (read & understand pseudocode)

Math Prerequisites

- Matrix manipulations (inverse, transpose, multiplication, etc.)
- Points and planes in dimension > 2
- Gradients

Is Introduction to Machine Learning (6.036/6.862) right for you?

Computer Science Prerequisites

- Python programming
- Algorithms (read & understand pseudocode)

Math Prerequisites

- Matrix manipulations (inverse, transpose, multiplication, etc.)
- Points and planes in dimension > 2
- Gradients
- Basic discrete probability (random variables, independence, conditioning, etc.)

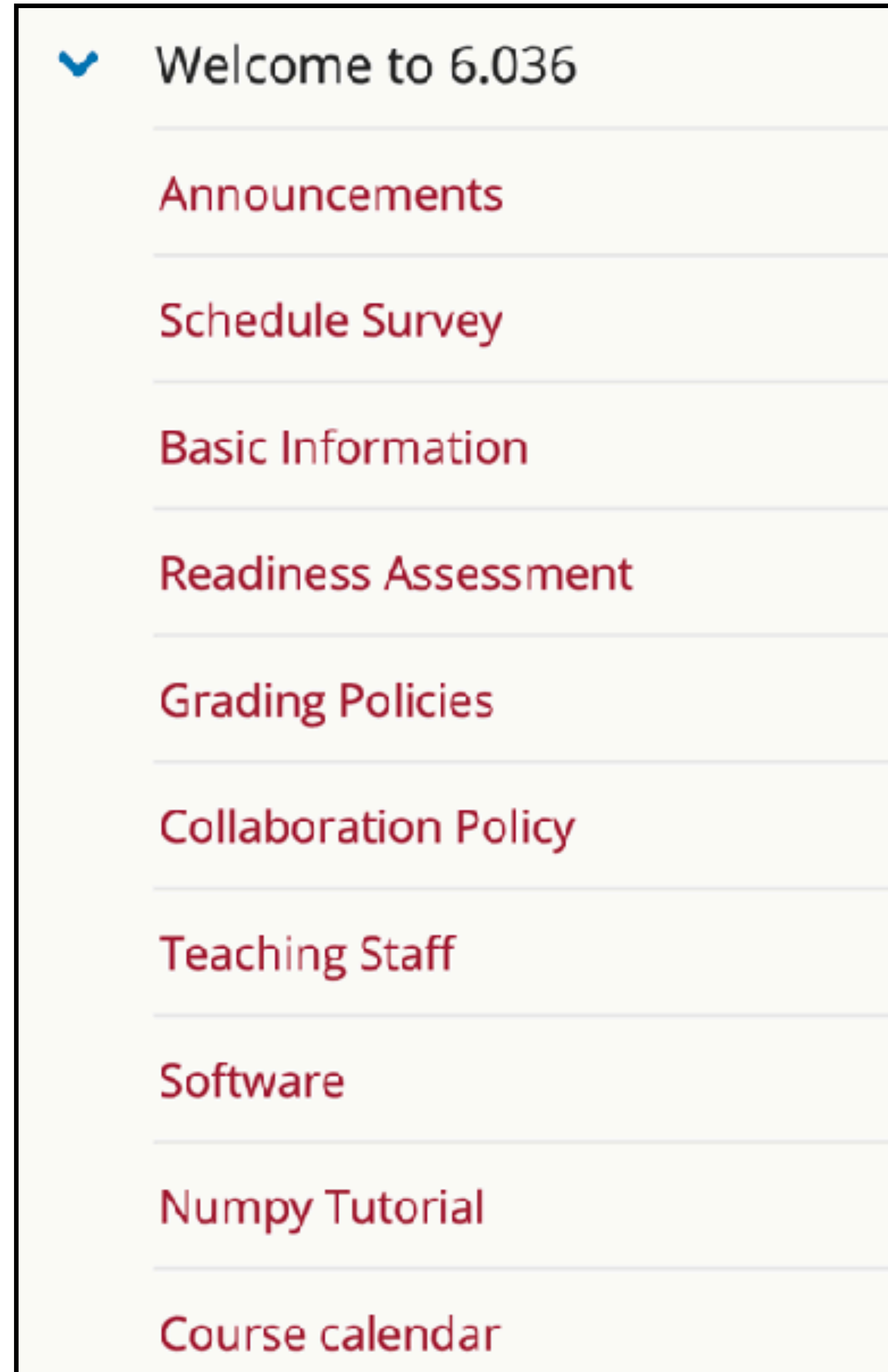
Is Introduction to Machine Learning (6.036/6.862) right for you?

Computer Science Prerequisites

- Python programming
- Algorithms (read & understand pseudocode)

Math Prerequisites

- Matrix manipulations (inverse, transpose, multiplication, etc.)
- Points and planes in dimension > 2
- Gradients
- Basic discrete probability (random variables, independence, conditioning, etc.)



Is Introduction to Machine Learning (6.036/6.862) right for you?

MIT EECS
ELECTRICAL ENGINEERING AND COMPUTER SCIENCE

Computer Science Prerequisites

- Python programming
- Algorithms (read & understand pseudocode)

Math Prerequisites

- Matrix manipulations (inverse, transpose, multiplication, etc.)
- Points and planes in dimension > 2
- Gradients
- Basic discrete probability (random variables, independence, conditioning, etc.)

Welcome to 6.036
- Announcements
- Schedule Survey
- Basic Information
- Readiness Assessment
- Grading Policies
- Collaboration Policy
- Teaching Staff
- Software
- Numpy Tutorial
- Course calendar

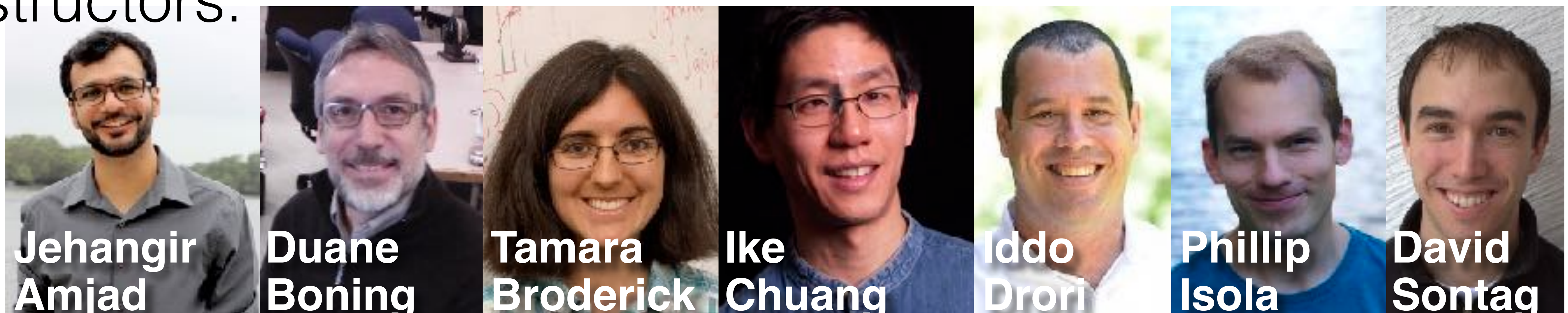
6.036/6.862: Introduction to Machine Learning

6.036/6.862: Introduction to Machine Learning, Staff

MIT EECS
ELECTRICAL ENGINEERING AND COMPUTER SCIENCE

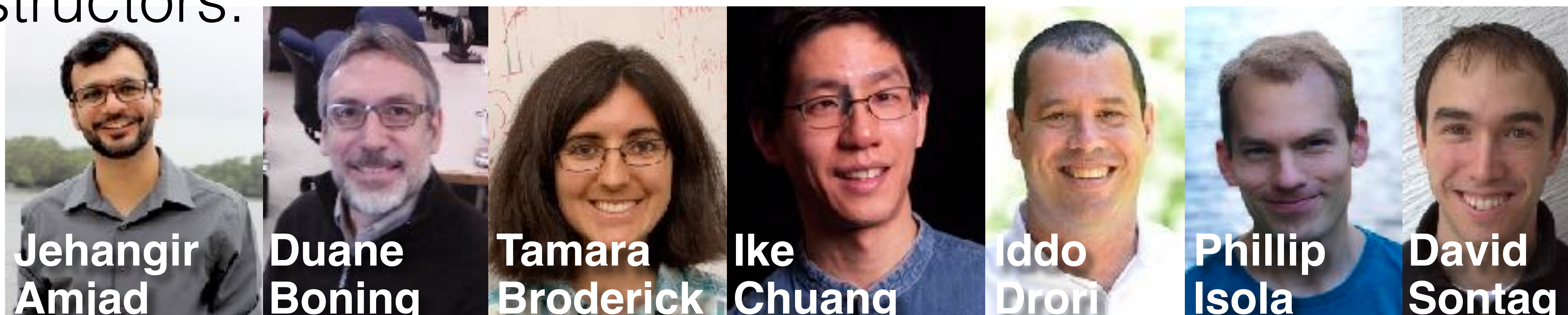
6.036/6.862: Introduction to Machine Learning, Staff

Instructors:

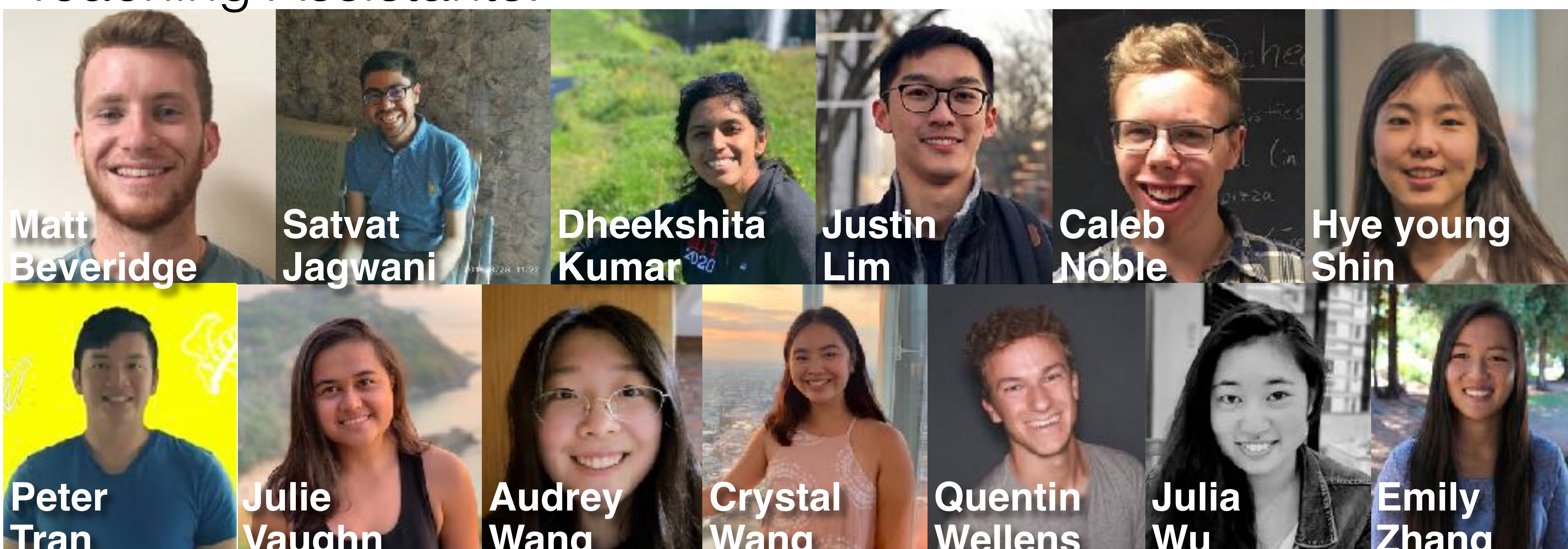


6.036/6.862: Introduction to Machine Learning, Staff

Instructors:



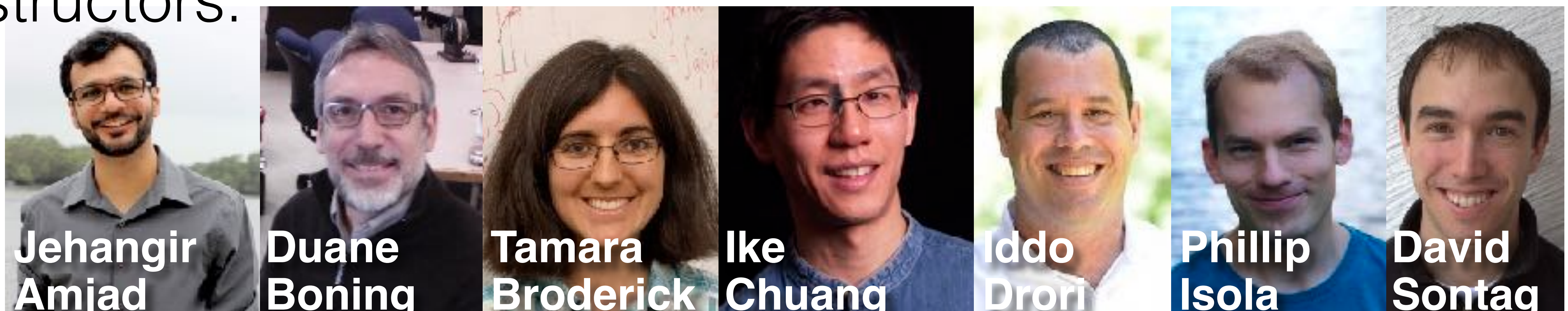
Teaching Assistants:



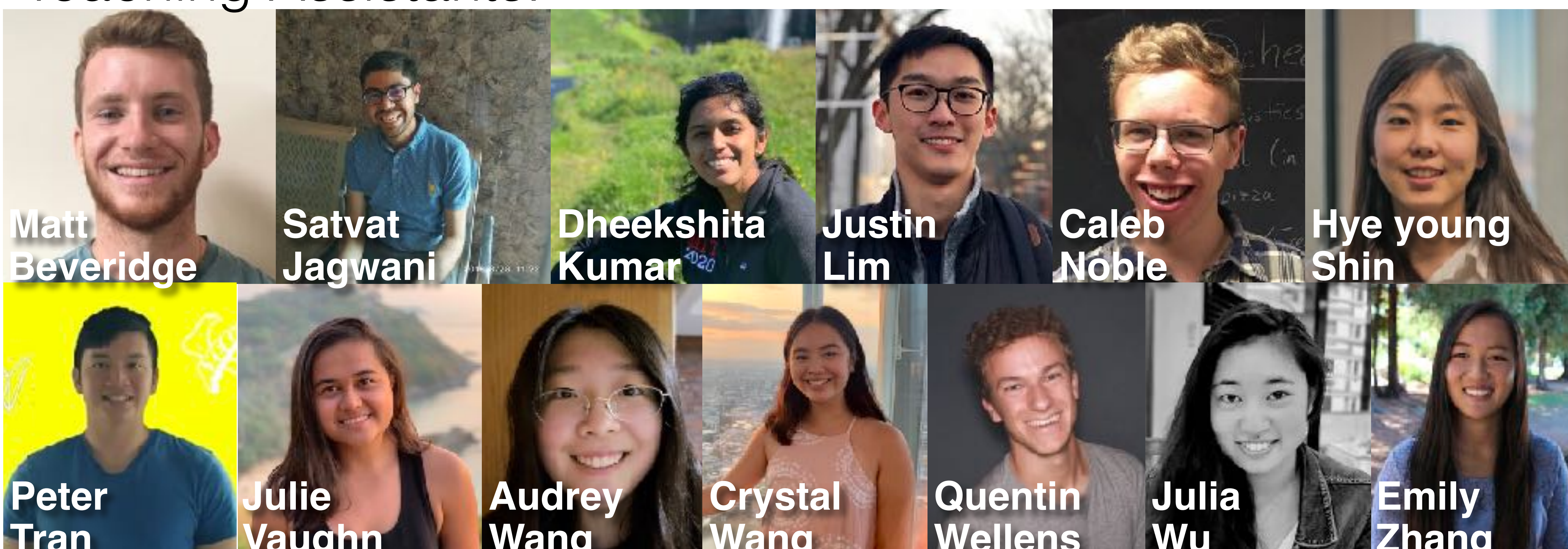
MIT EECS
ELECTRICAL ENGINEERING AND COMPUTER SCIENCE

6.036/6.862: Introduction to Machine Learning, Staff

Instructors:



Teaching Assistants:



And Lab Assistants!

6.036/6.862: Introduction to Machine Learning, Weekly Plan

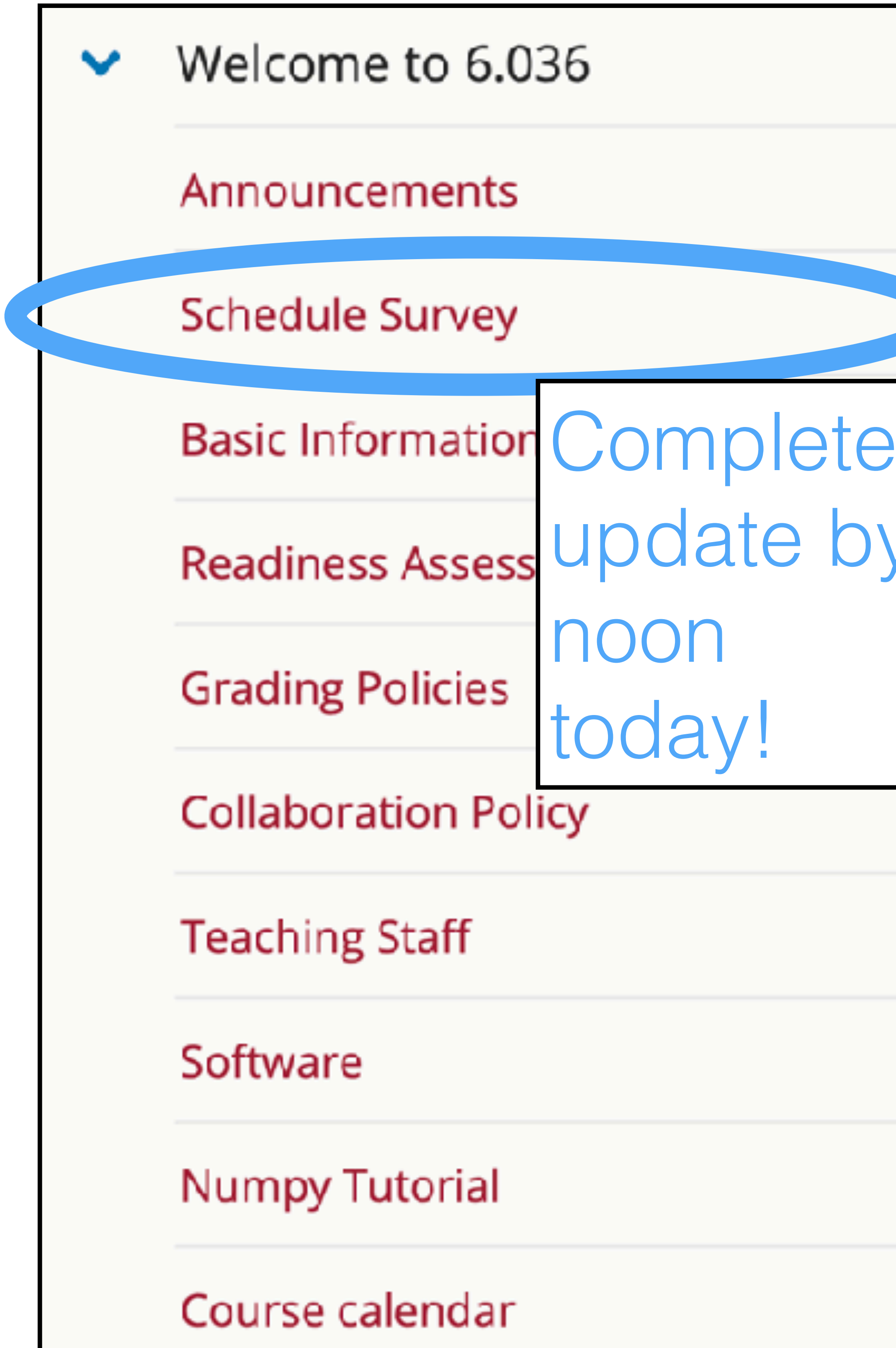
- Welcome to 6.036
  - Announcements
  - Schedule Survey
  - Basic Information
  - Readiness Assessment
  - Grading Policies
  - Collaboration Policy
  - Teaching Staff
  - Software
  - Numpy Tutorial
  - Course calendar

6.036/6.862: Introduction to Machine Learning, Weekly Plan

Welcome to 6.036

- Announcements
- Schedule Survey
- Basic Information
- Readiness Assessment
- Grading Policies
- Collaboration Policy
- Teaching Staff
- Software
- Numpy Tutorial
- Course calendar

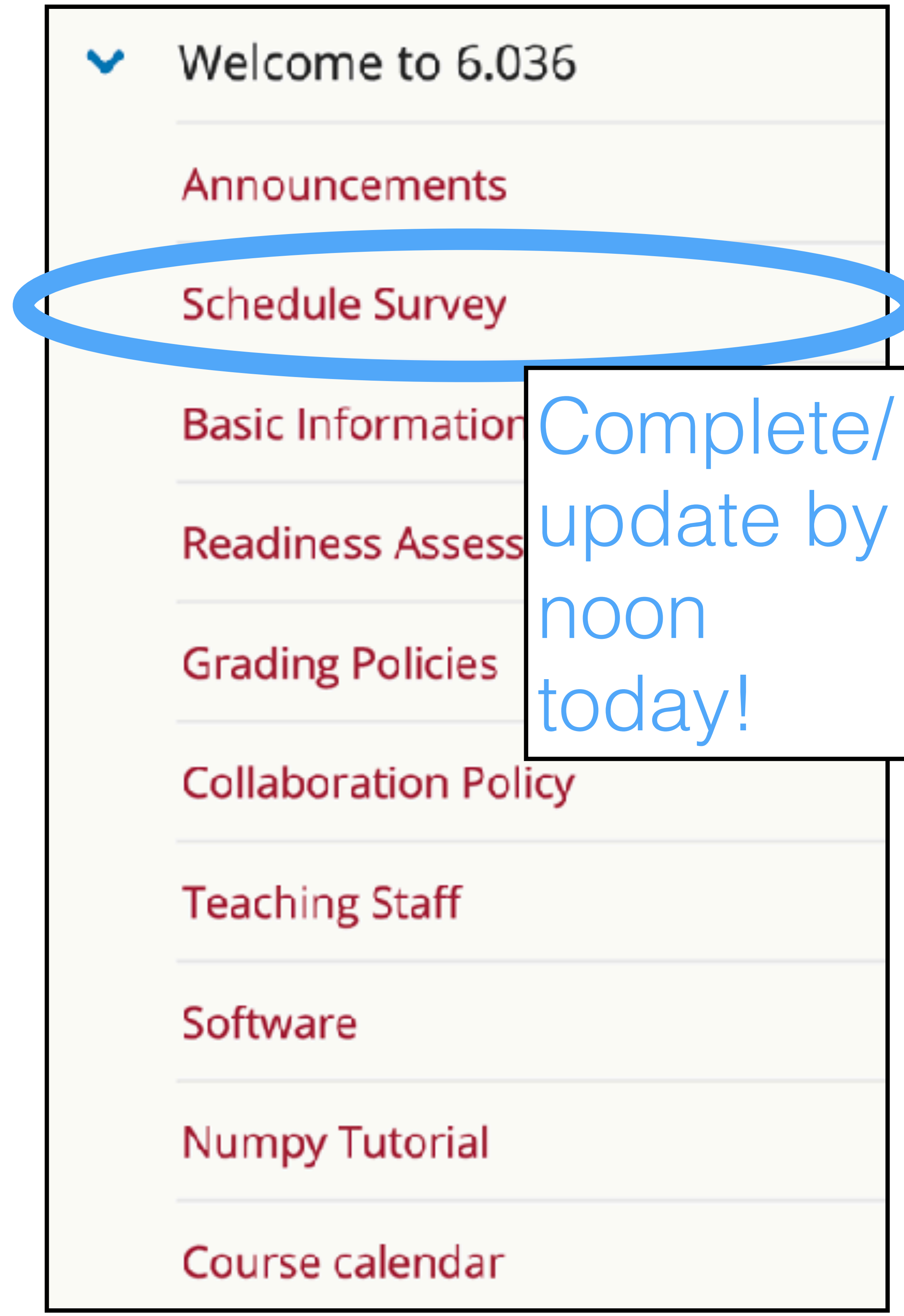
6.036/6.862: Introduction to Machine Learning, Weekly Plan



Complete/ update by noon today!

6.036/6.862: Introduction to Machine Learning, Weekly Plan

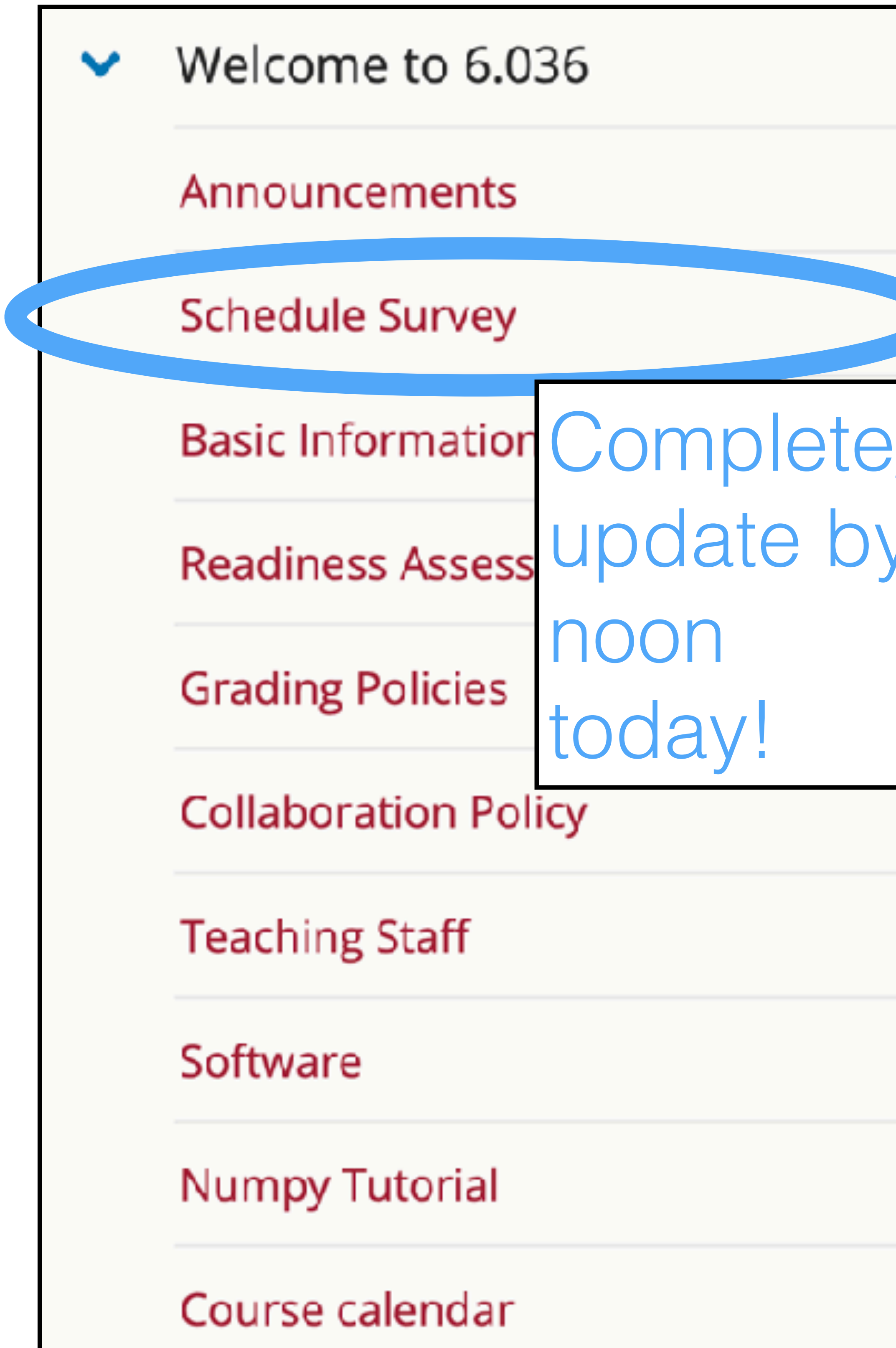
- **Lecture** + course notes



Complete/ update by noon today!

6.036/6.862: Introduction to Machine Learning, Weekly Plan

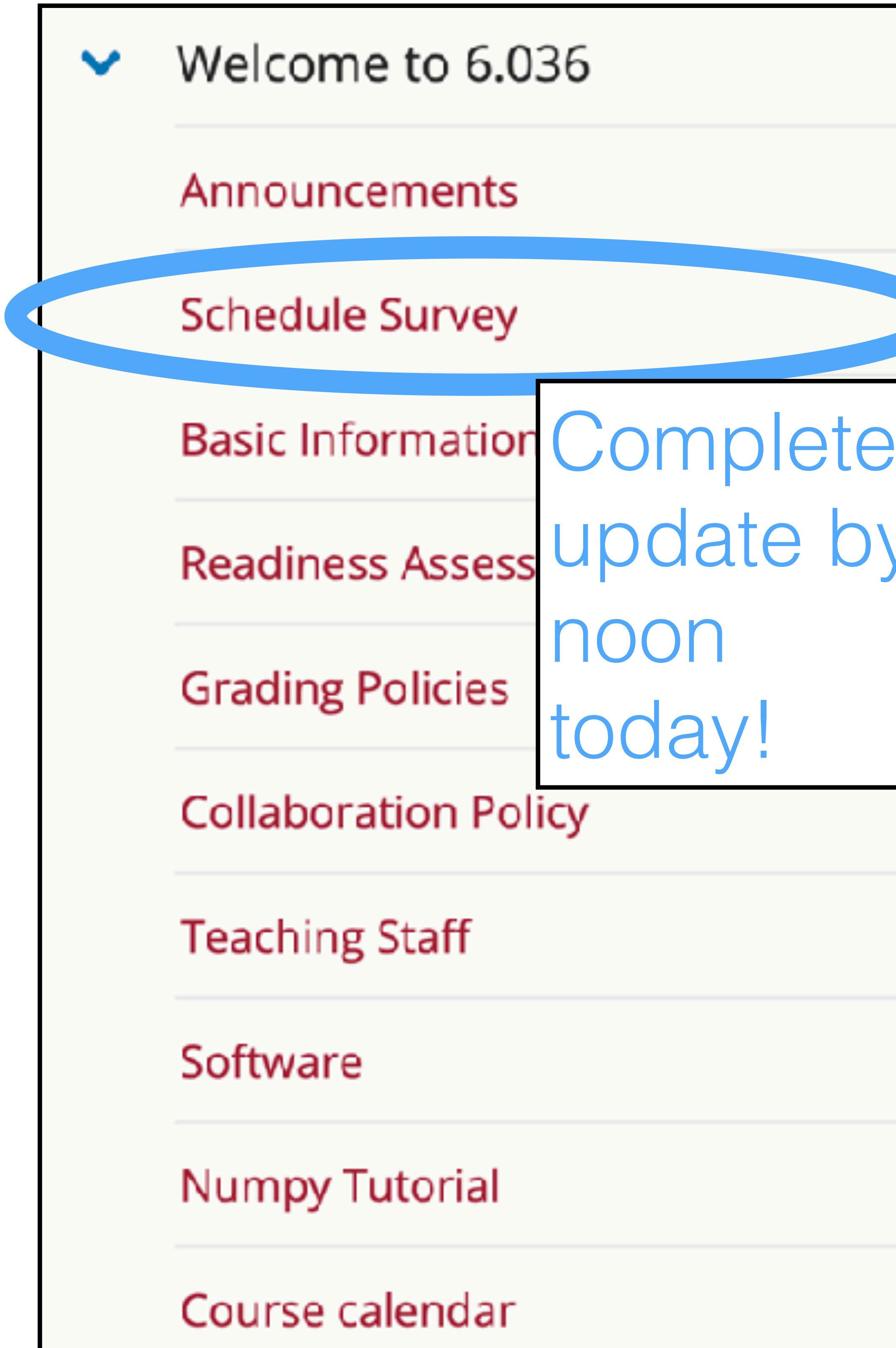
- **Lecture** + course notes
- **Exercises**
  - Due 9am before lecture



Complete/update by noon today!

6.036/6.862: Introduction to Machine Learning, Weekly Plan

- **Lecture** + course notes
- **Exercises**
  - Due 9am before lecture
- **Lab** (*synchronous, required!*)

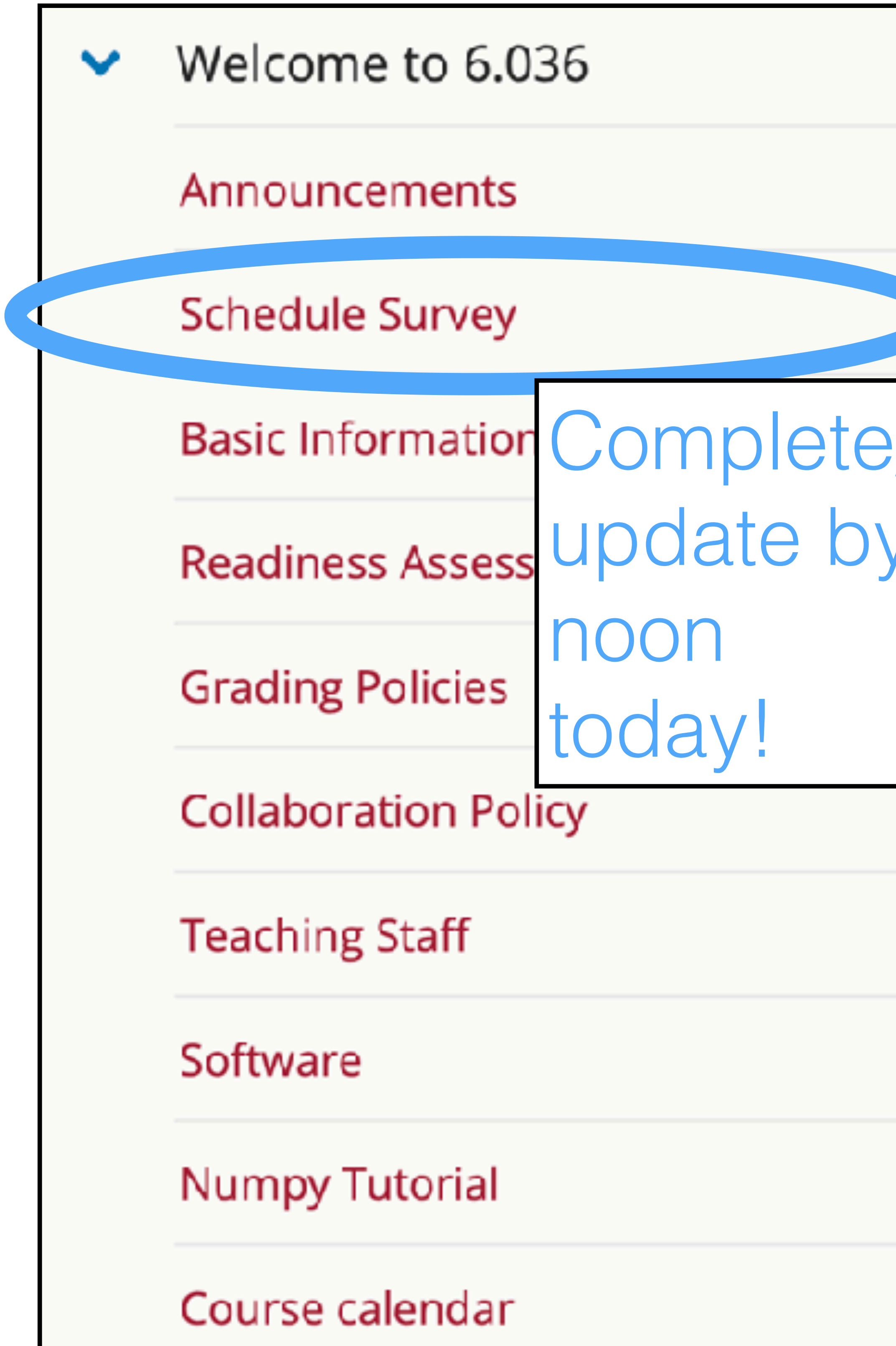


Complete/ update by noon today!

6.036/6.862: Introduction to Machine Learning, Weekly Plan

MIT EECS ELECTRICAL ENGINEERING AND COMPUTER SCIENCE

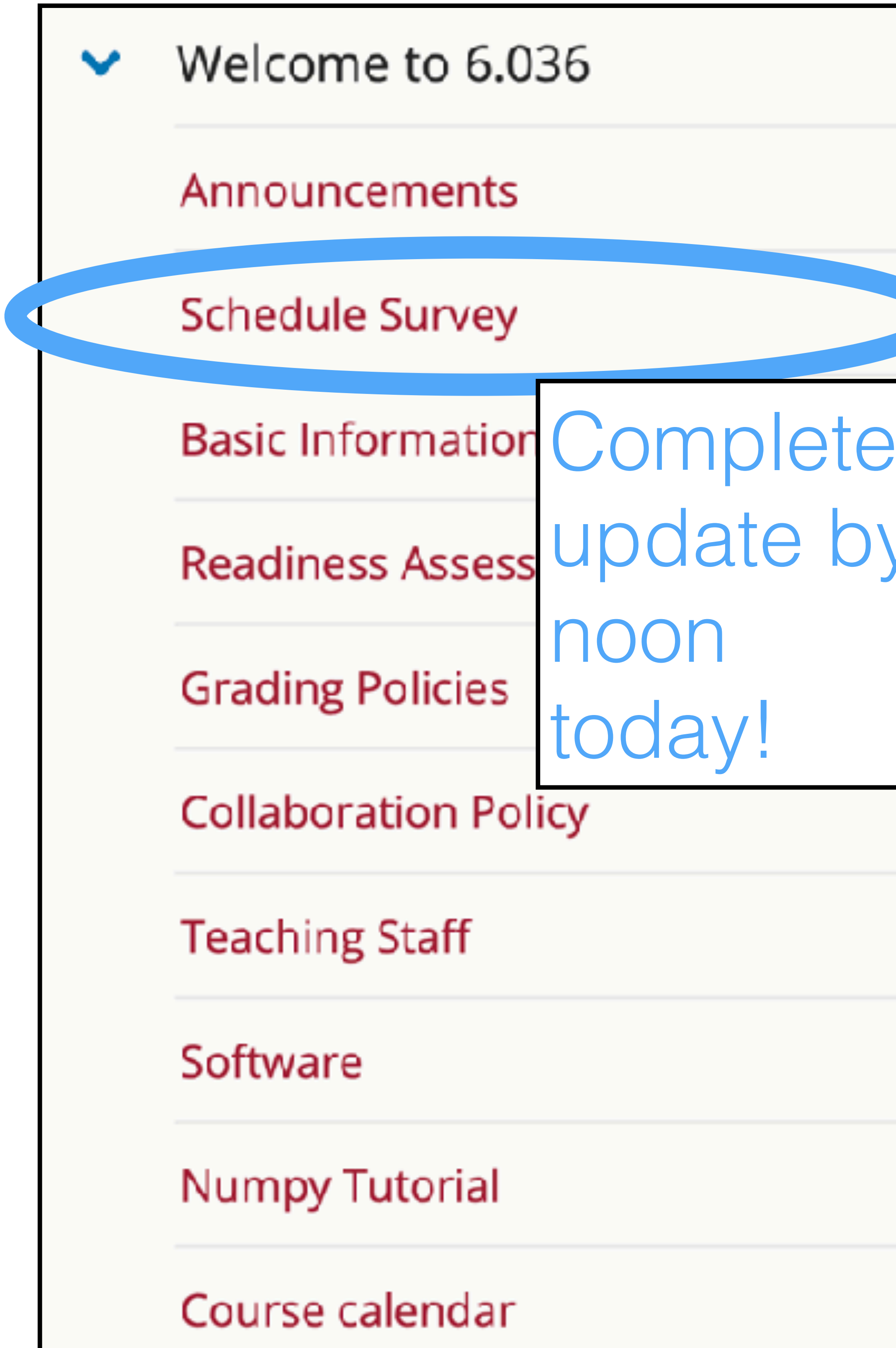
- **Lecture** + course notes
- **Exercises**
    - Due 9am before lecture
- **Lab** (*synchronous, required!*)
    - (New!) MLyPod: 10 students



Complete/update by noon today!

6.036/6.862: Introduction to Machine Learning, Weekly Plan

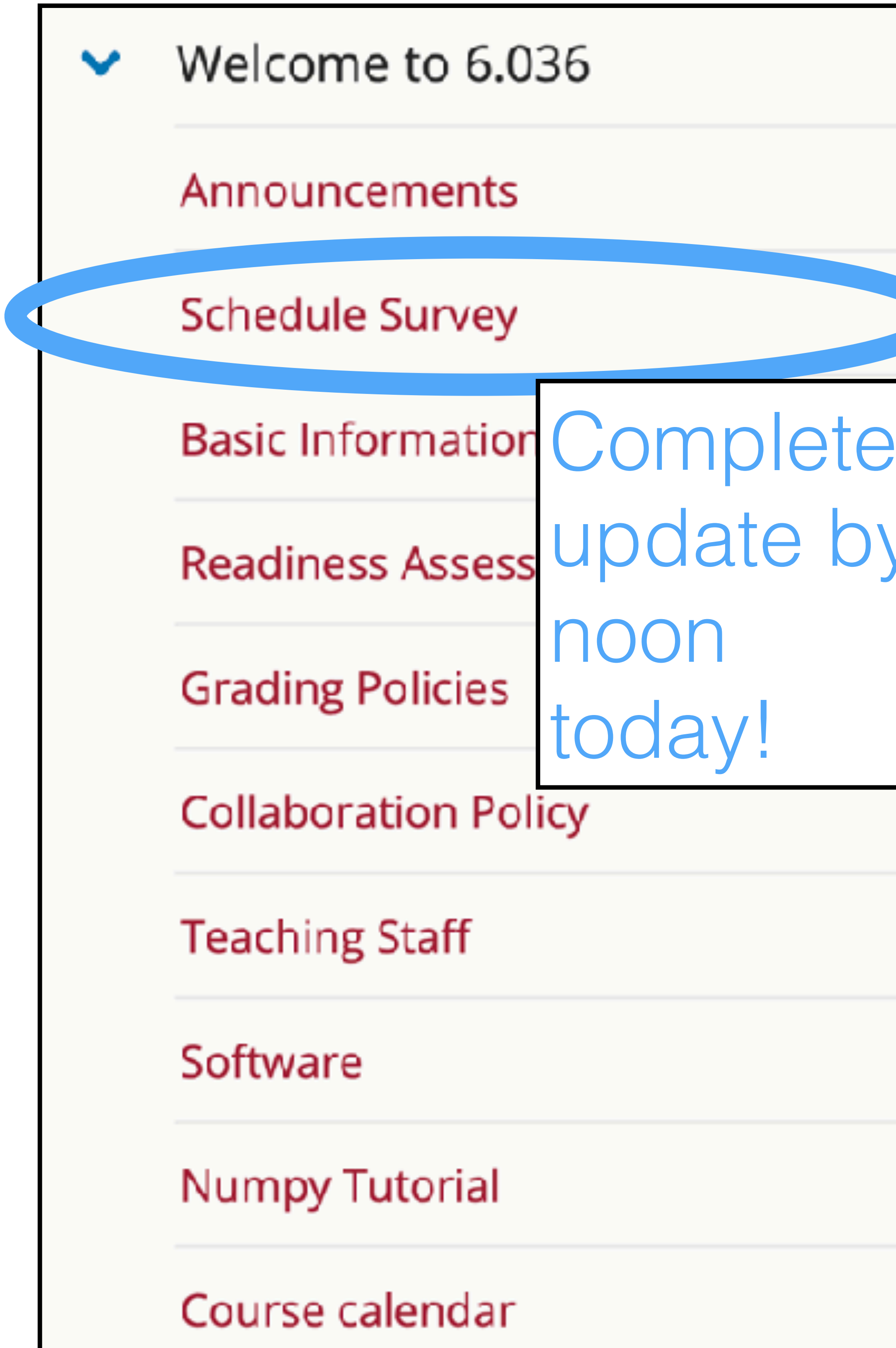
- **Lecture** + course notes
- **Exercises**
  - Due 9am before lecture
- **Lab** (*synchronous, required!*)
  - (New!) MLyPod: 10 students
  - Work in groups of 2 to 3; check off with staff



Complete/update by noon today!

6.036/6.862: Introduction to Machine Learning, Weekly Plan

- **Lecture** + course notes
- **Exercises**
  - Due 9am before lecture
- **Lab** (*synchronous, required!*)
  - (New!) MLyPod: 10 students
  - Work in groups of 2 to 3; check off with staff
- **Homework**



Complete/update by noon today!

6.036/6.862: Introduction to Machine Learning, Weekly Plan

- **Lecture** + course notes
- **Exercises**
  - Due 9am before lecture
- **Lab** (*synchronous, required!*)
  - (New!) MLyPod: 10 students
  - Work in groups of 2 to 3; check off with staff
- **Homework**

6.036/6.862: Introduction to Machine Learning, Weekly Plan

- **Lecture** + course notes
- **Exercises**
  - Due 9am before lecture
- **Lab** (*synchronous, required!*)
  - (New!) MLyPod: 10 students
  - Work in groups of 2 to 3; check off with staff
- **Homework**

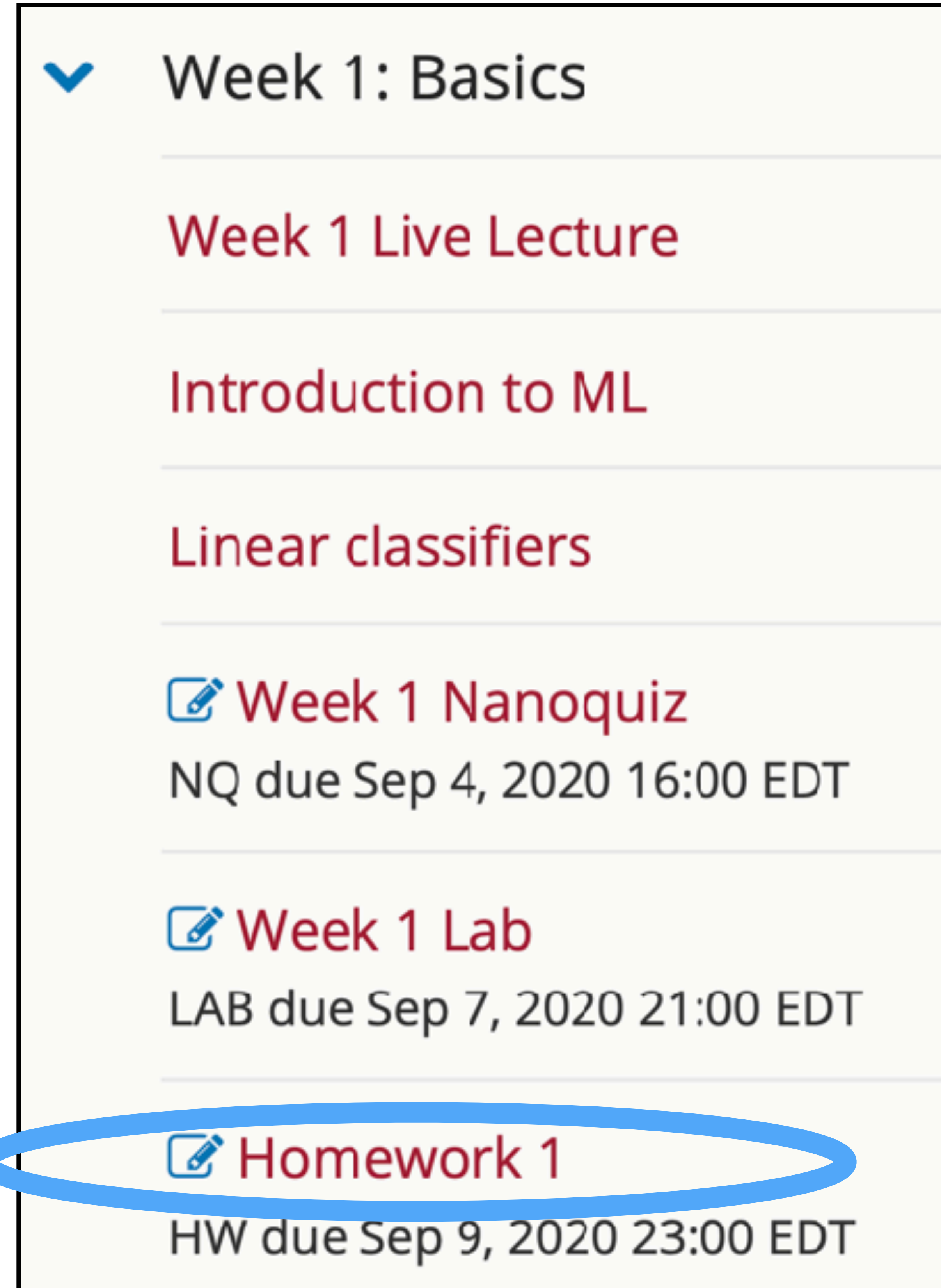
Week 1: Basics

- Week 1 Live Lecture
- Introduction to ML
- Linear classifiers
- Week 1 Nanoquiz
  NQ due Sep 4, 2020 16:00 EDT
- Week 1 Lab
  LAB due Sep 7, 2020 21:00 EDT
- Homework 1
  HW due Sep 9, 2020 23:00 EDT

MIT EECS
ELECTRICAL ENGINEERING AND COMPUTER SCIENCE

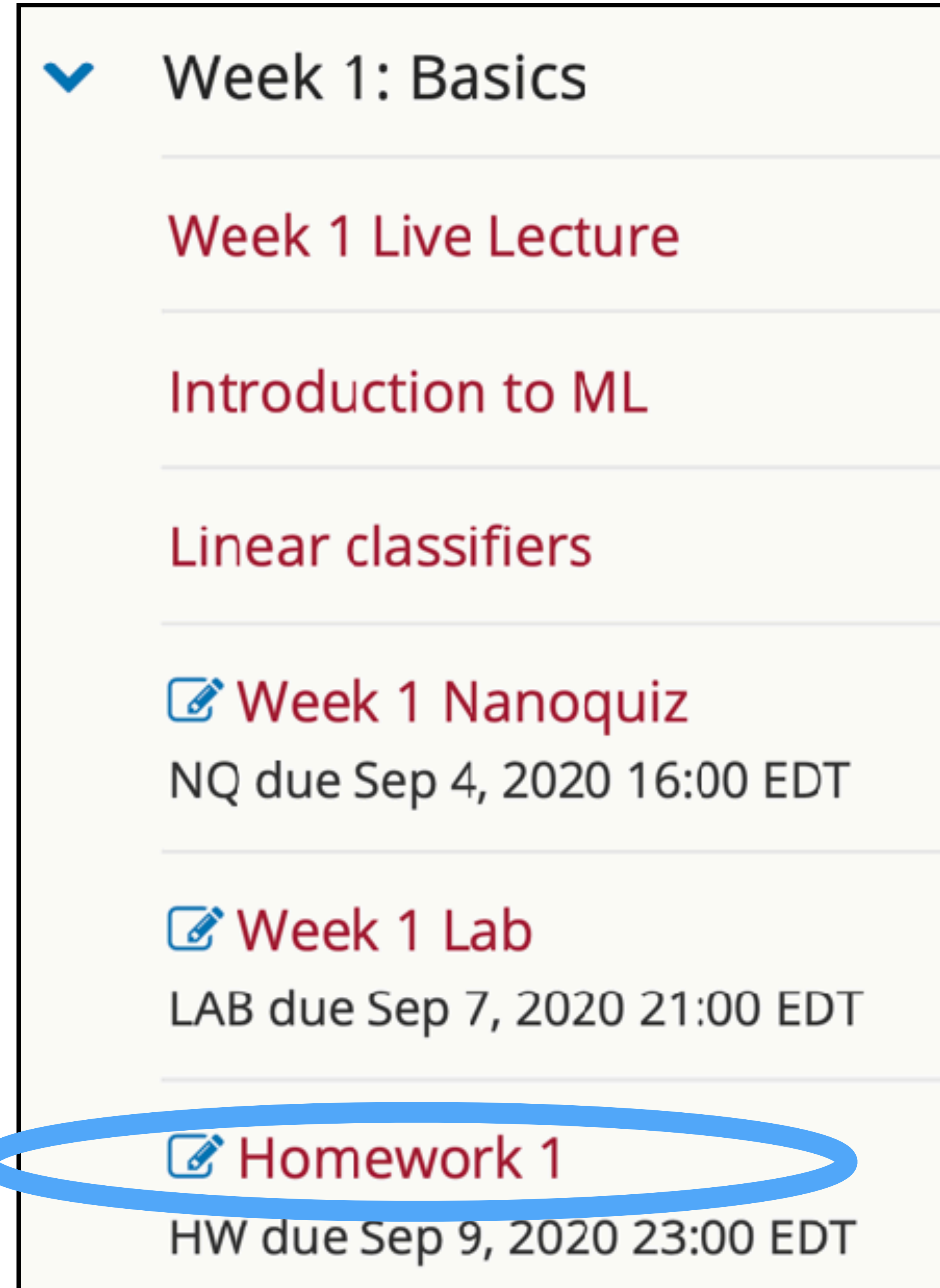
6.036/6.862: Introduction to Machine Learning, Weekly Plan

- **Lecture** + course notes
- **Exercises**
  - Due 9am before lecture
- **Lab** (*synchronous, required!*)
  - (New!) MLyPod: 10 students
  - Work in groups of 2 to 3; check off with staff
- **Homework**



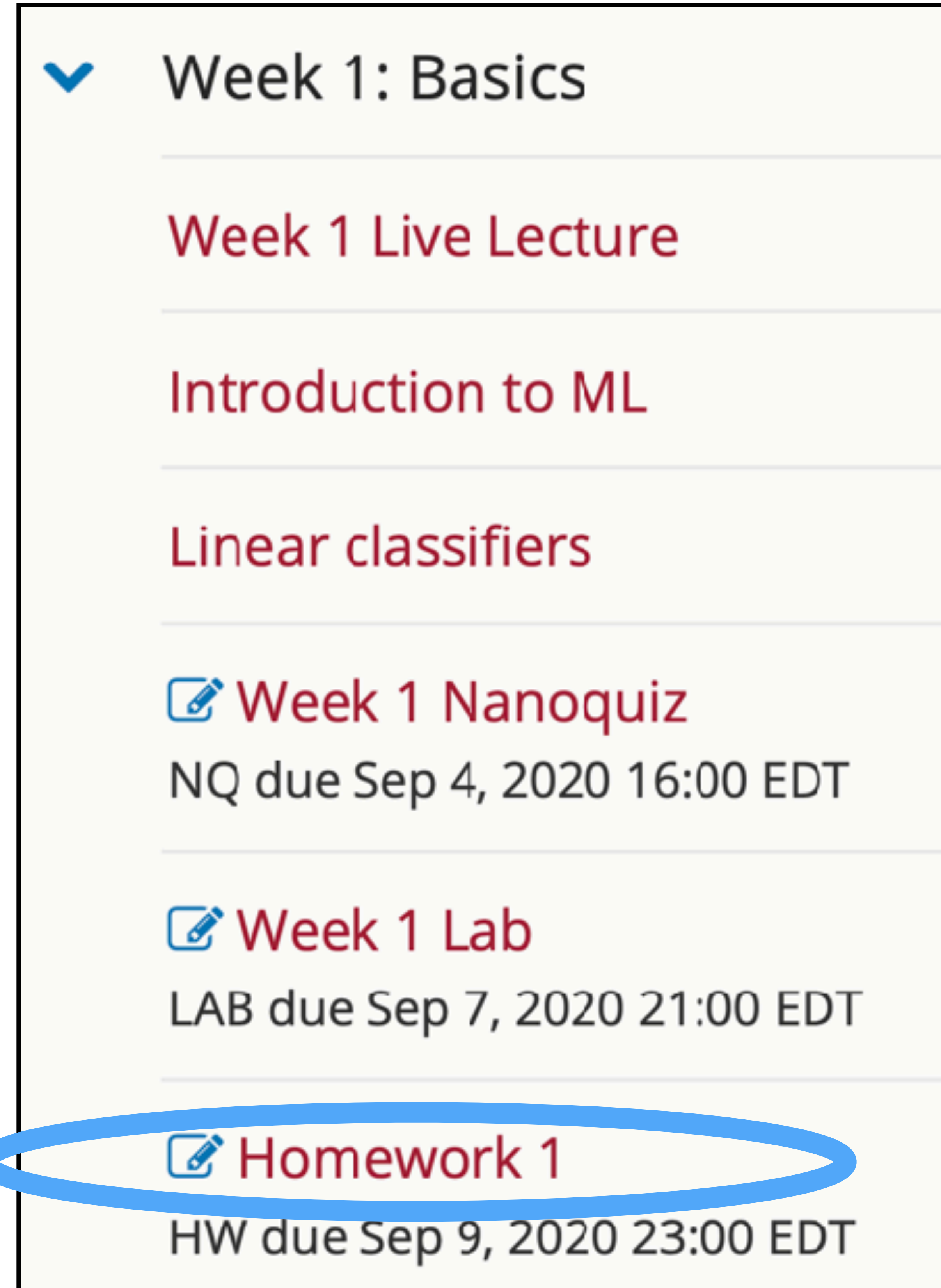
6.036/6.862: Introduction to Machine Learning, Weekly Plan

- **Lecture** + course notes
- **Exercises**
  - Due 9am before lecture
- **Lab** (*synchronous, required!*)
  - (New!) MLyPod: 10 students
  - Work in groups of 2 to 3; check off with staff
- **Homework**
- **Nanoquiz** (no midterm/final)
  - Timed



6.036/6.862: Introduction to Machine Learning, Weekly Plan

- **Lecture** + course notes
- **Exercises**
  - Due 9am before lecture
- **Lab** (*synchronous, required!*)
  - (New!) MLyPod: 10 students
  - Work in groups of 2 to 3; check off with staff
- **Homework**
- **Nanoquiz** (no midterm/final)
  - Timed
- **Office hours**



6.036/6.862: Introduction to Machine Learning, Weekly Plan

- **Lecture** + course notes
- **Exercises**
  - Due 9am before lecture
- **Lab** (*synchronous, required!*)
  - (New!) MLyPod: 10 students
  - Work in groups of 2 to 3; check off with staff
- **Homework**
- **Nanoquiz** (no midterm/final)
  - Timed
- **Office hours**
- **6.862:** project (canvas.mit.edu)

Week 1: Basics

- Week 1 Live Lecture
- Introduction to ML
- Linear classifiers
- Week 1 Nanoquiz
  NQ due Sep 4, 2020 16:00 EDT
- Week 1 Lab
  LAB due Sep 7, 2020 21:00 EDT
- Homework 1
  HW due Sep 9, 2020 23:00 EDT

Machine learning (ML): why & what

Machine learning (ML): why & what

Machine learning algorithm confirms 50 new exoplanets in historic first

by R. Dallon Adams in Innovation
on August 26, 2020, 9:07 AM PST

A new machine learning technique can be used to sift through massive datasets to discern exoplanets from false positives.

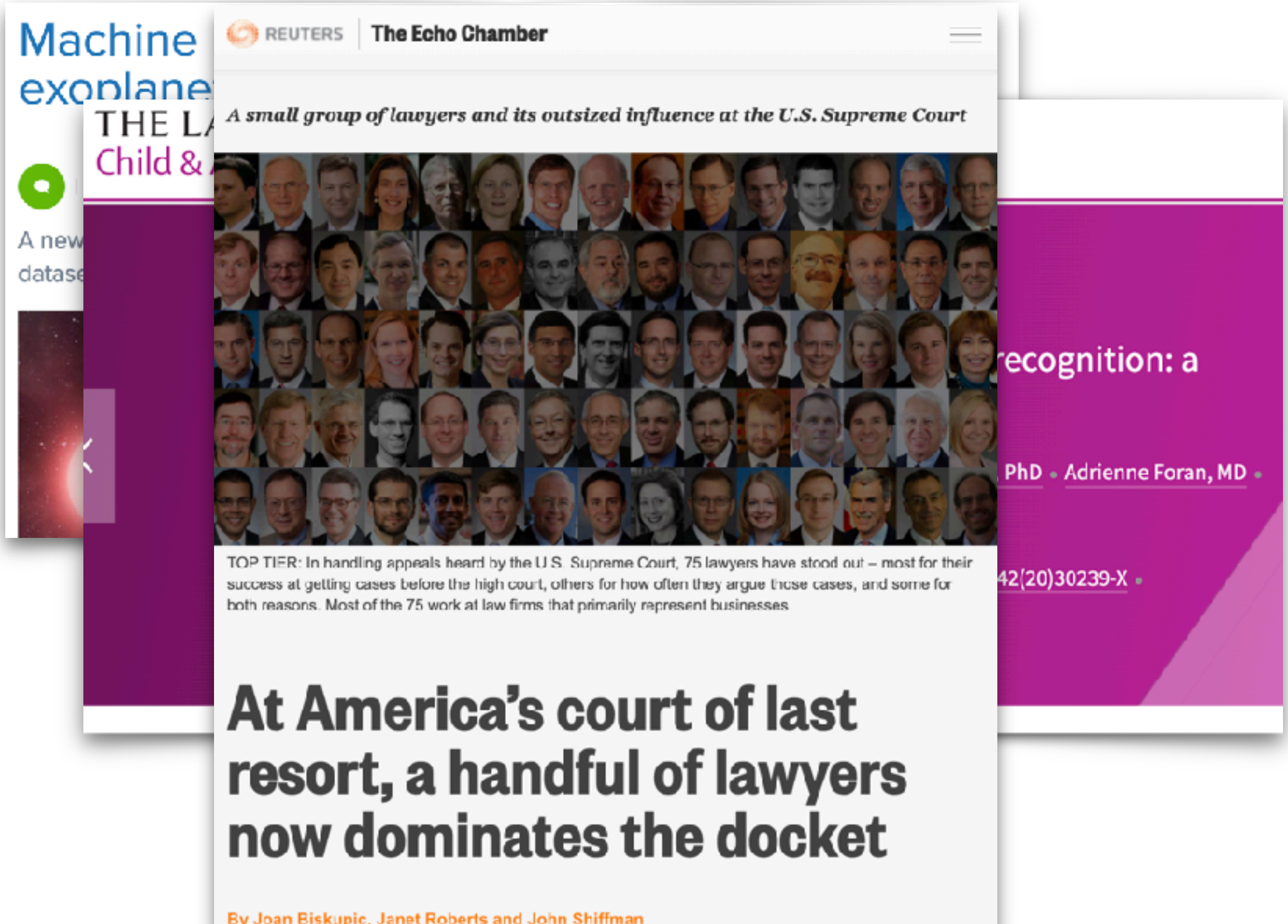
TechRepublic.

[https://www.techrepublic.com/article/machine-learning-algorithm-confirms-50-new-exoplanets-in-historic-first/]

Machine learning (ML): why & what

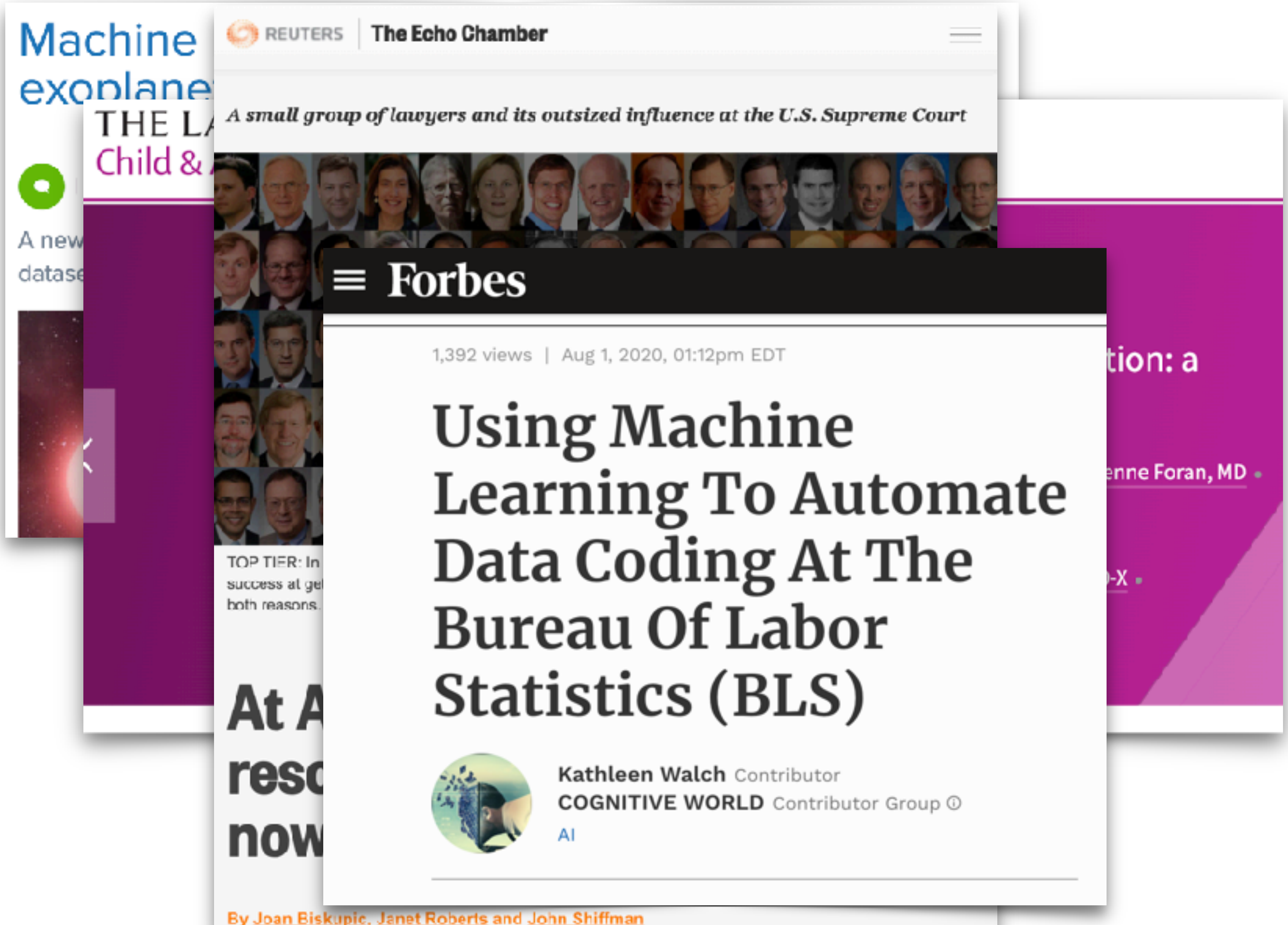
[https://www.thelancet.com/journals/lanchi/article/PIIS2352-4642(20)30239-X/fulltext]

Machine learning (ML): why & what



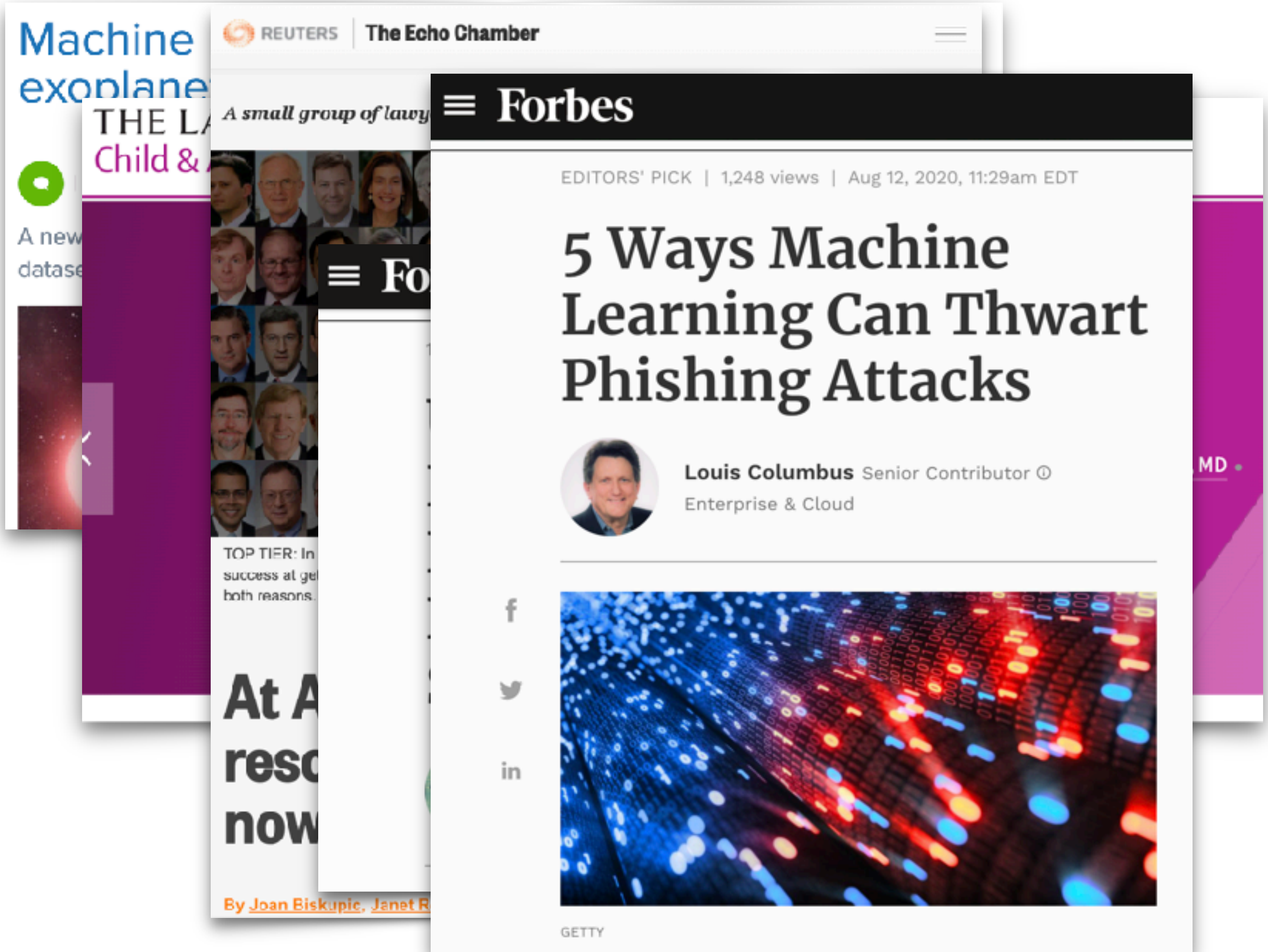
[https://www.reuters.com/investigates/special-report/scotus/]

Machine learning (ML): why & what



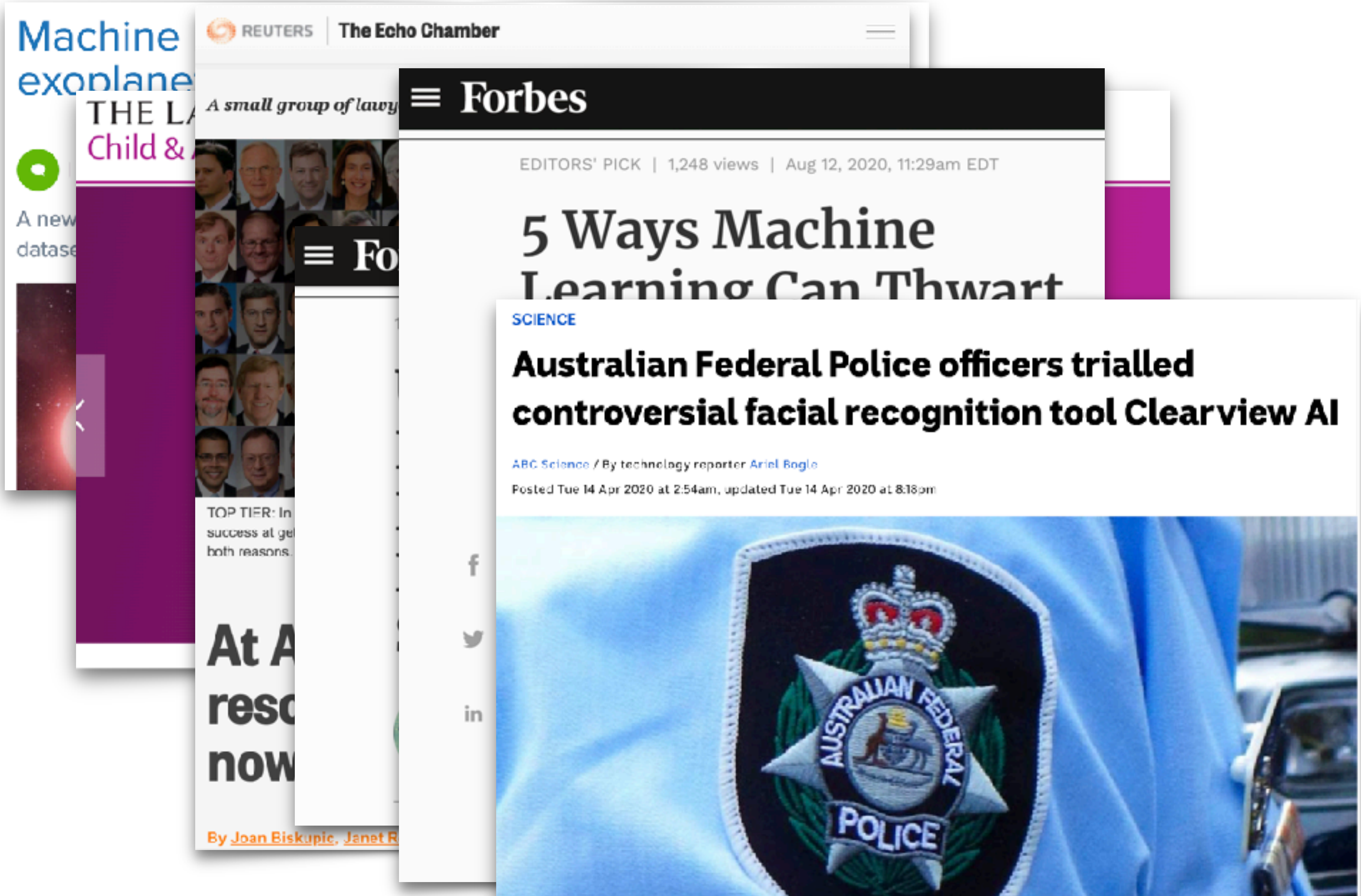
[https://www.forbes.com/sites/cognitiveworld/2020/08/01/using-machine-learning-to-automate-data-coding-at-the-bureau-of-labor-statistics-bls]

Machine learning (ML): why & what



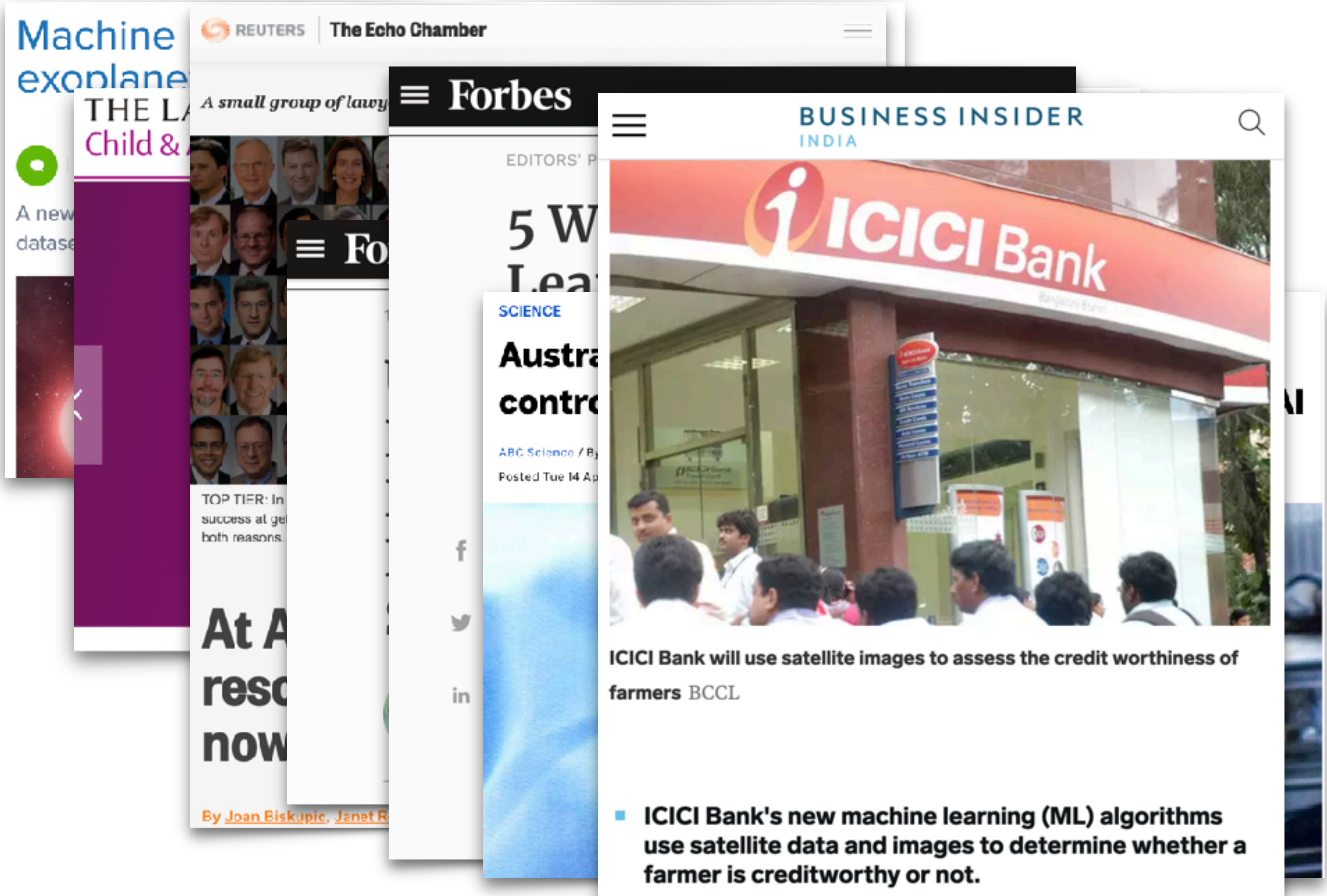
[https://www.forbes.com/sites/louiscolumbus/2020/08/12/5-ways-machine-learning-can-thwart-phishing-attacks]

Machine learning (ML): why & what



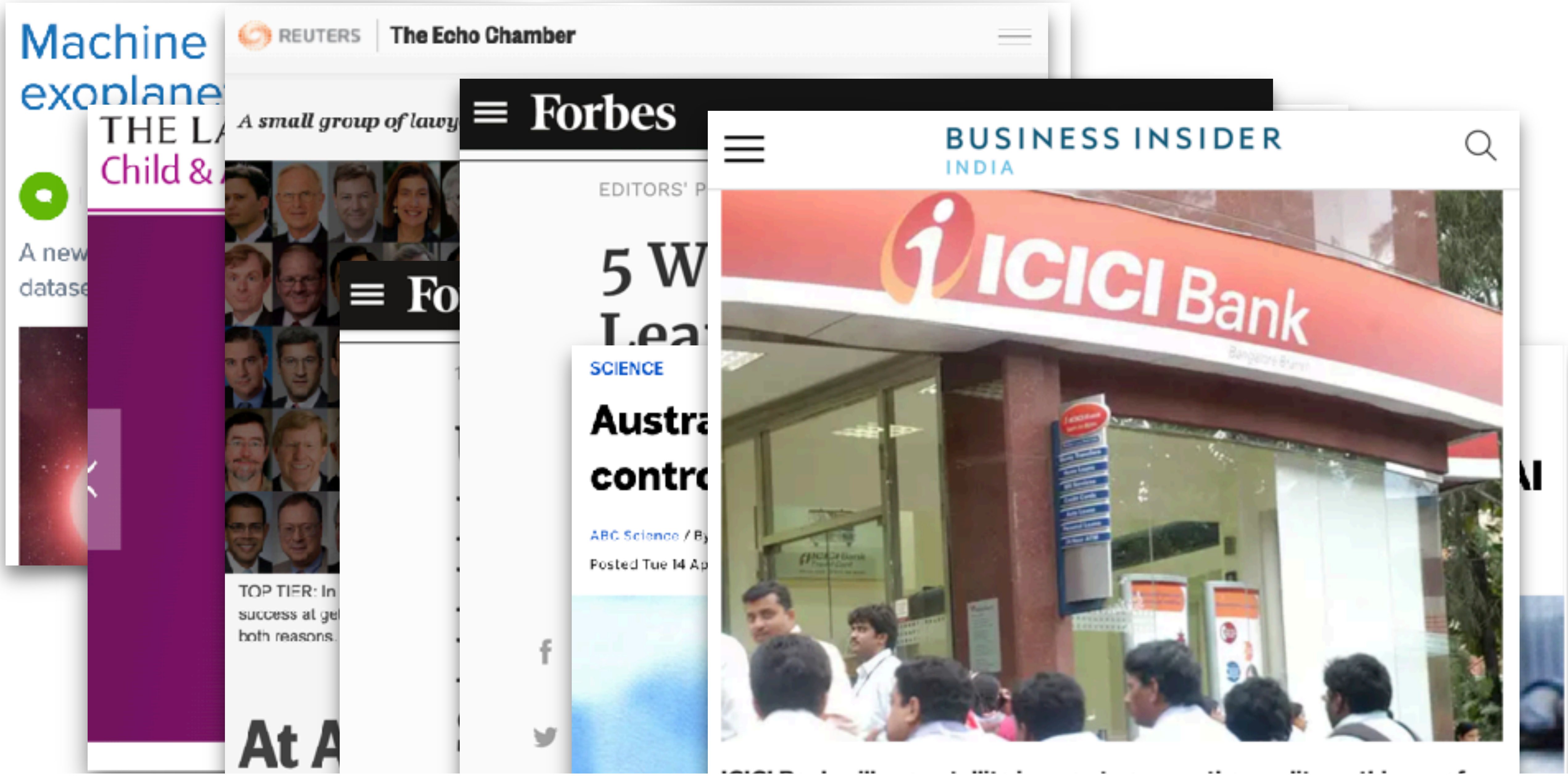
[https://www.abc.net.au/news/science/2020-04-14/clearview-ai-facial-recognition-tech-australian-federal-police/12146894]

Machine learning (ML): why & what

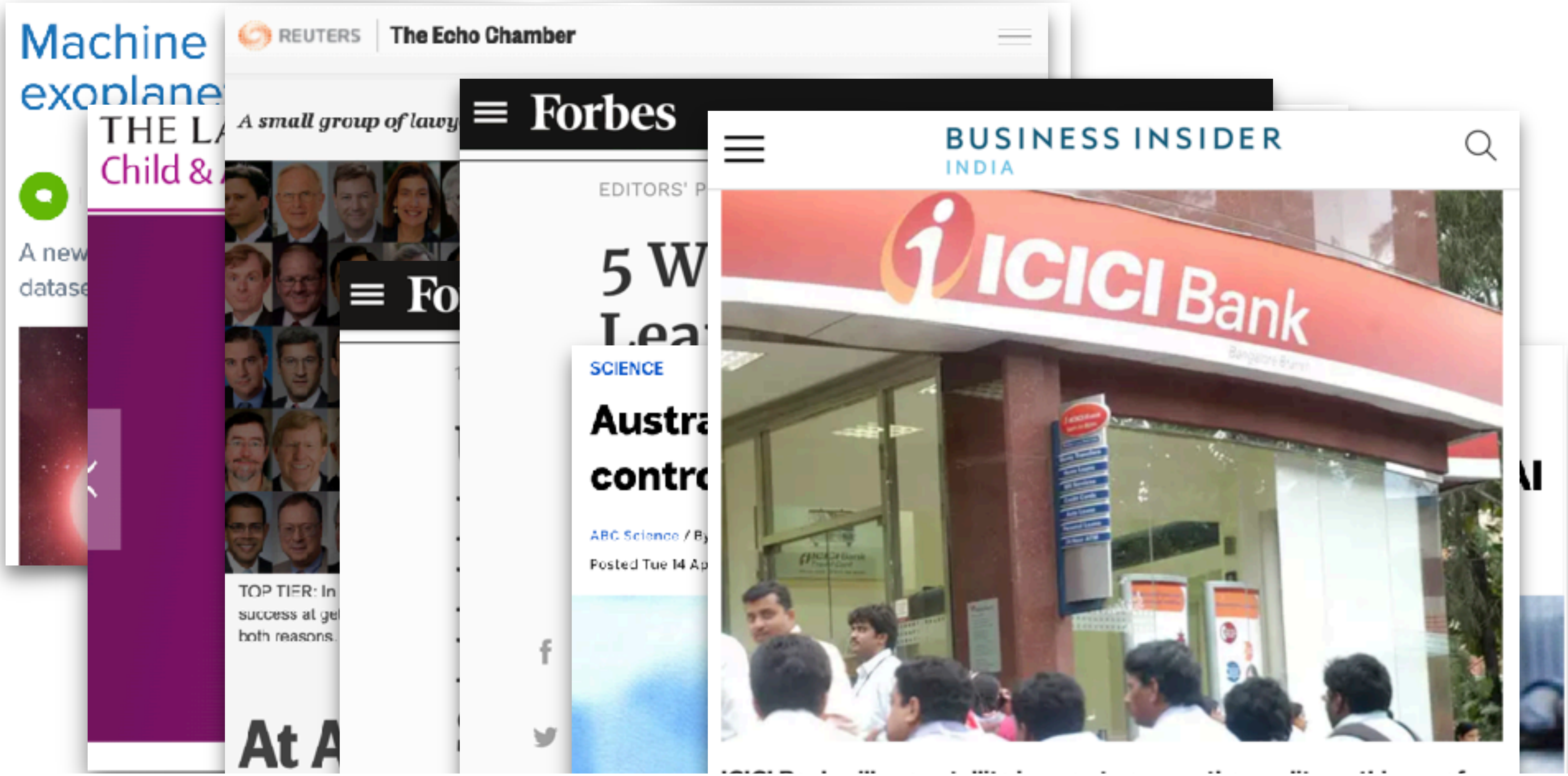


[https://www.businessinsider.in/finance/banks/news/icici-bank-is-looking-to-pump-out-more-kisan-credit-cards-with-satellite-data-and-machine-learning-at-its-side/articleshow/77741696.cms]

Machine learning (ML): why & what

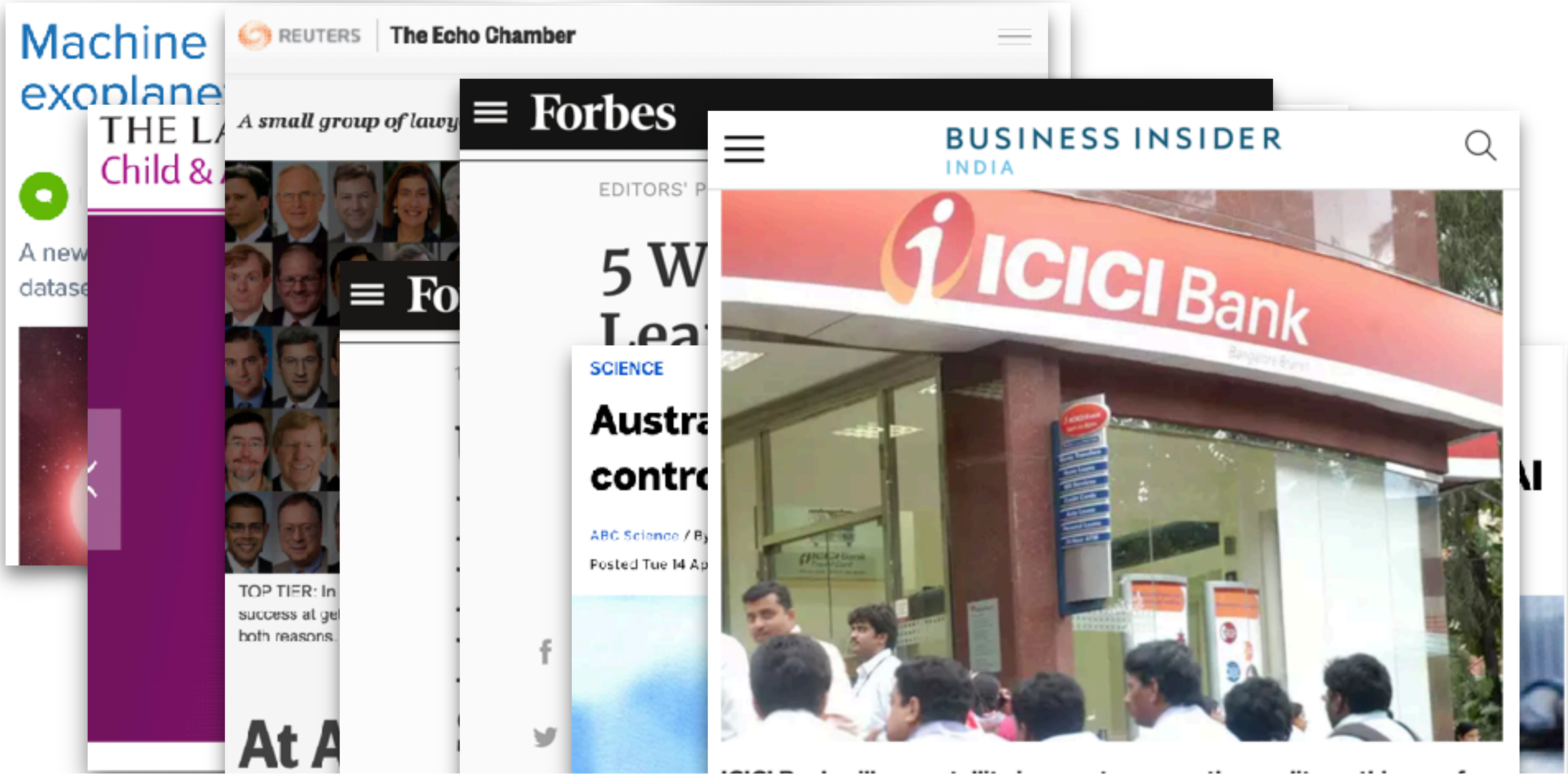


Machine learning (ML): why & what



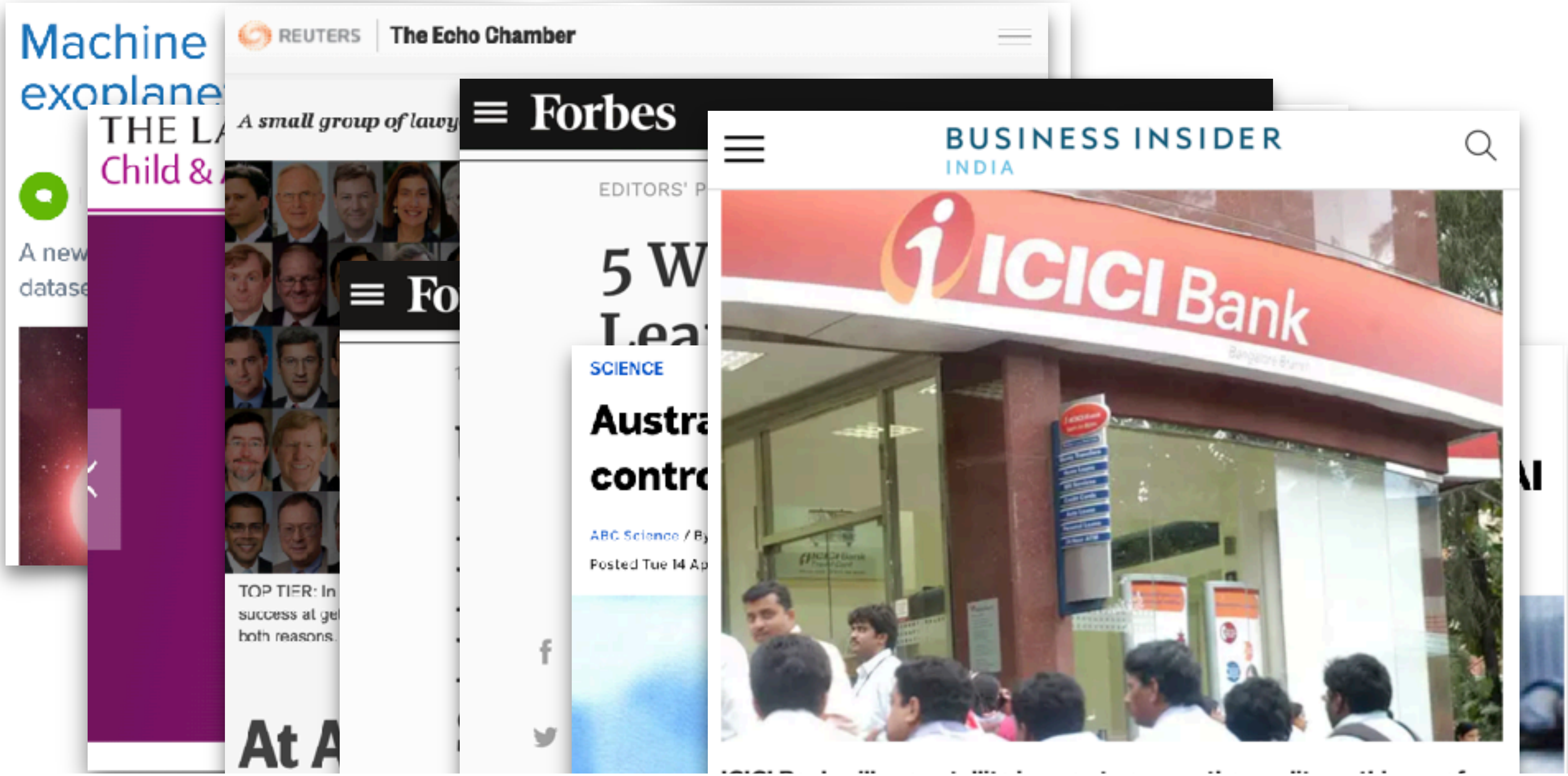
- **What is ML?**

Machine learning (ML): why & what



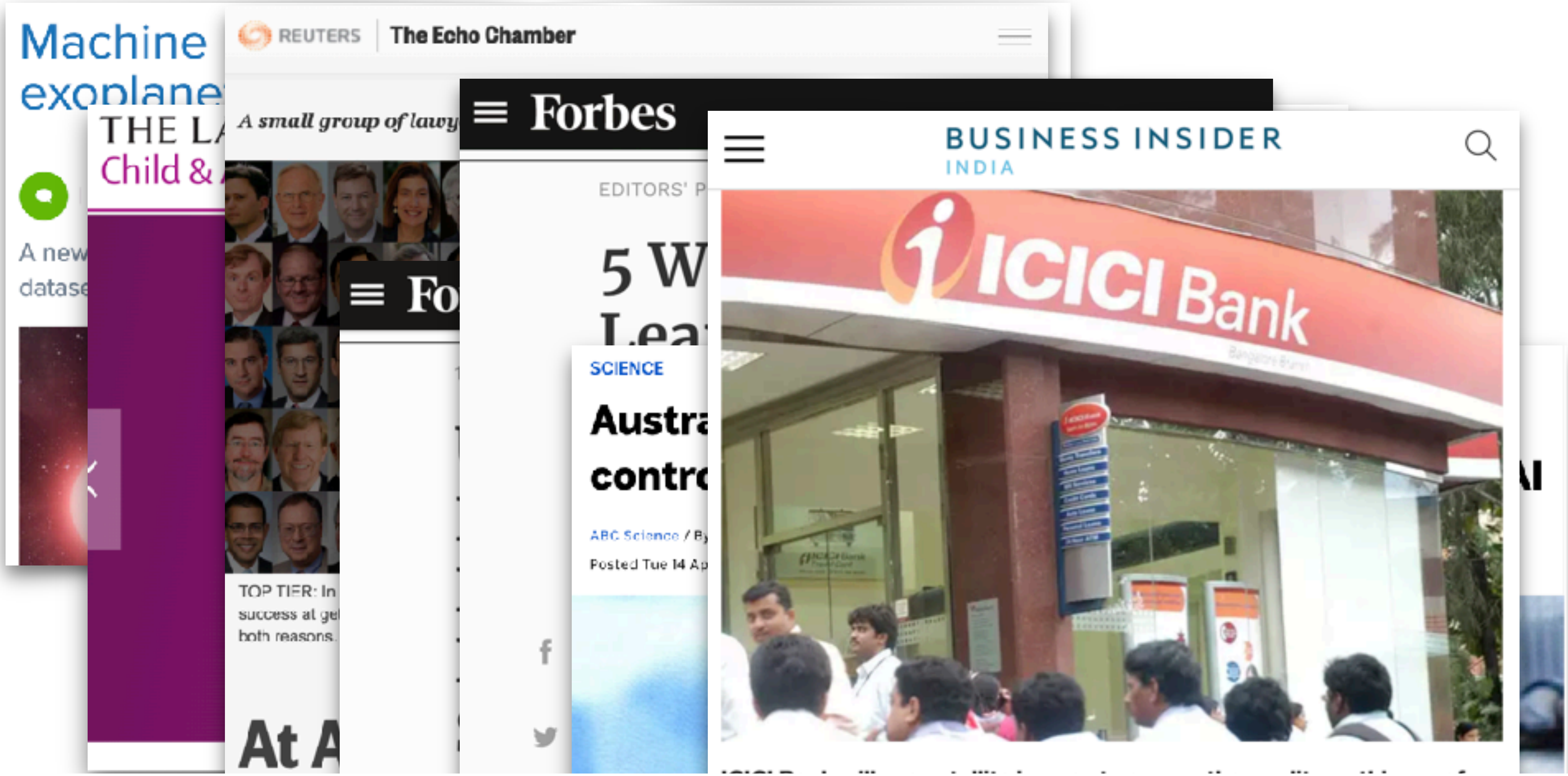
- **What is ML?** A set of methods for making decisions from data. (See the rest of the course!)

Machine learning (ML): why & what



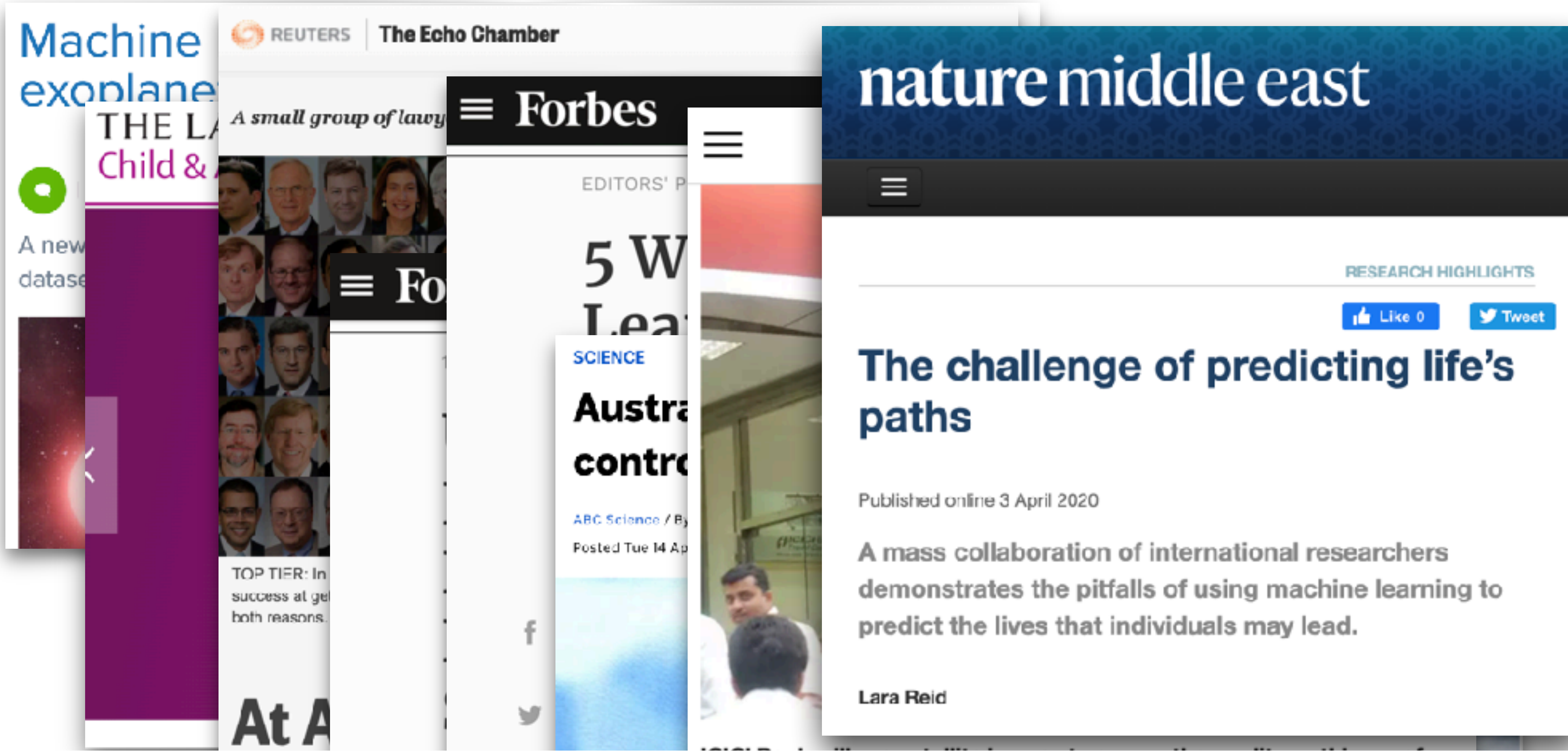
- **What is ML?** A set of methods for making decisions from data. (See the rest of the course!)
- **Why study ML?** To apply; to understand; to evaluate

Machine learning (ML): why & what



- **What is ML?** A set of methods for making decisions from data. (See the rest of the course!)
- **Why study ML?** To apply; to understand; to evaluate
- **Notes:** ML is not magic. ML is built on math.

Machine learning (ML): why & what



- **What is ML?** A set of methods for making decisions from data. (See the rest of the course!)
- **Why study ML?** To apply; to understand; to evaluate
- **Notes:** ML is not magic. ML is built on math.

[https://www.natureasia.com/en/nmiddleeast/article/10.1038/nmiddleeast.2020.46]

Getting started

Getting started

What do we have?

Getting started

What do we have? (Training) data

Getting started

What do we have? (Training) data

- *n* training data points

Getting started

What do we have? (Training) data

- *n* training data points
- For data point $i \in \{1, \ldots, n\}$

Getting started

What do we have? (Training) data

- *n* training data points
- For data point $i \in \{1, \ldots, n\}$
  - Feature vector

Getting started

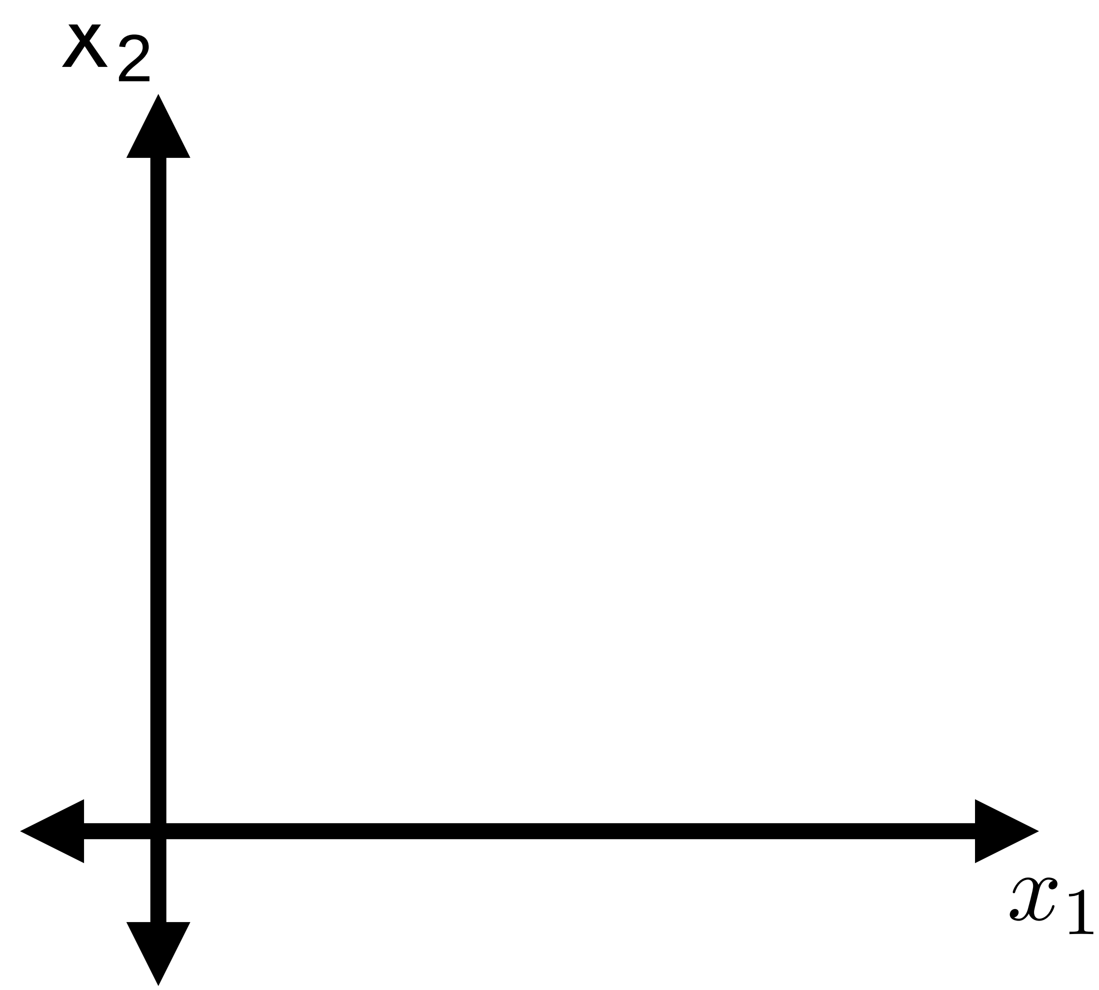
What do we have? (Training) data

- *n* training data points
- For data point $i \in \{1, \ldots, n\}$
  - Feature vector
    $x^{(i)} = (x_1^{(i)}, \ldots, x_d^{(i)})^{!} \;\; ! \;\; R^d$

Getting started

What do we have? (Training) data

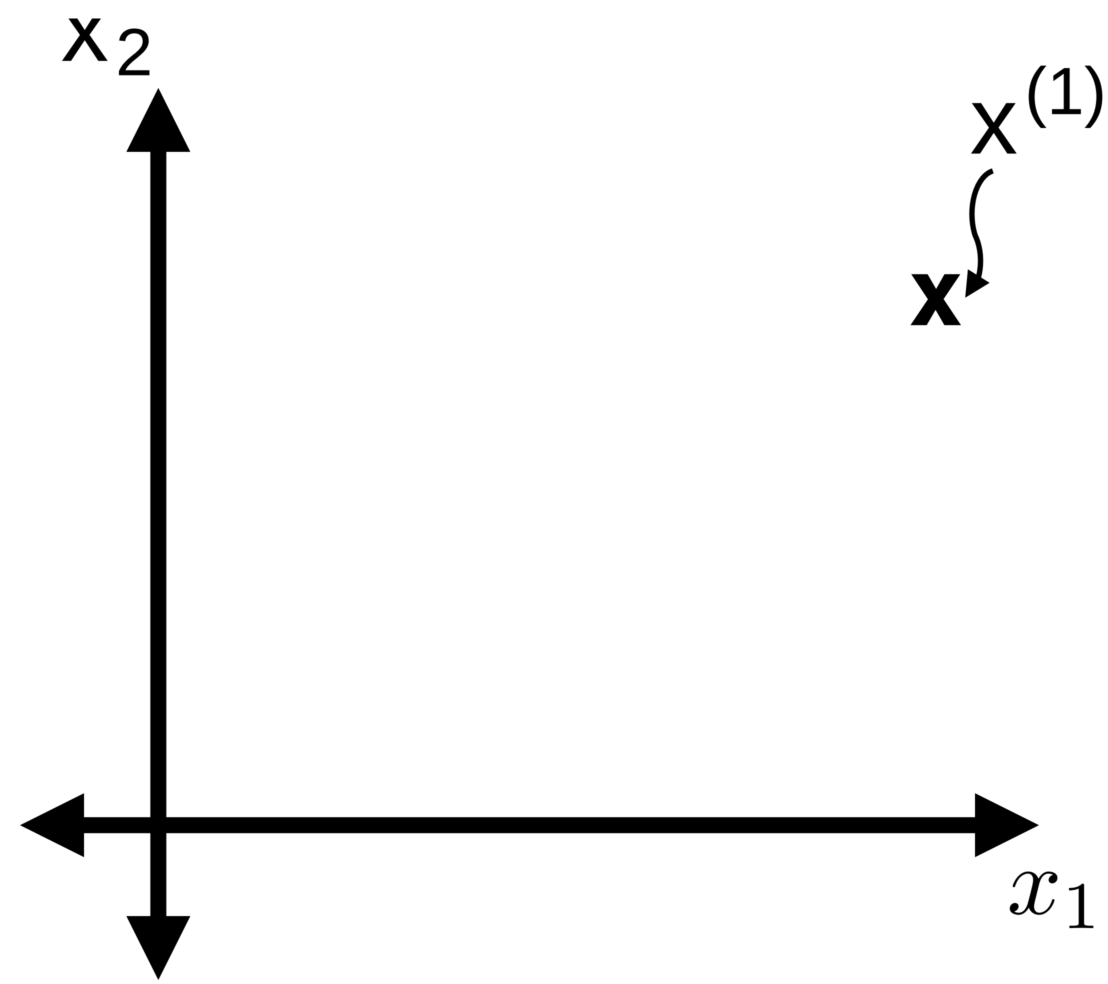
- *n* training data points
- For data point $i \in \{1, \ldots, n\}$
  - Feature vector
    $x^{(i)} = (x_1^{(i)}, \ldots, x_d^{(i)})^! \;\; ! \;\; R^d$



Getting started

What do we have? (Training) data

- *n* training data points
- For data point $i \in \{1, \ldots, n\}$
  - Feature vector
    $\mathbf{x}^{(i)} = (x_1^{(i)}, \ldots, x_d^{(i)})^\top \in \mathbb{R}^d$

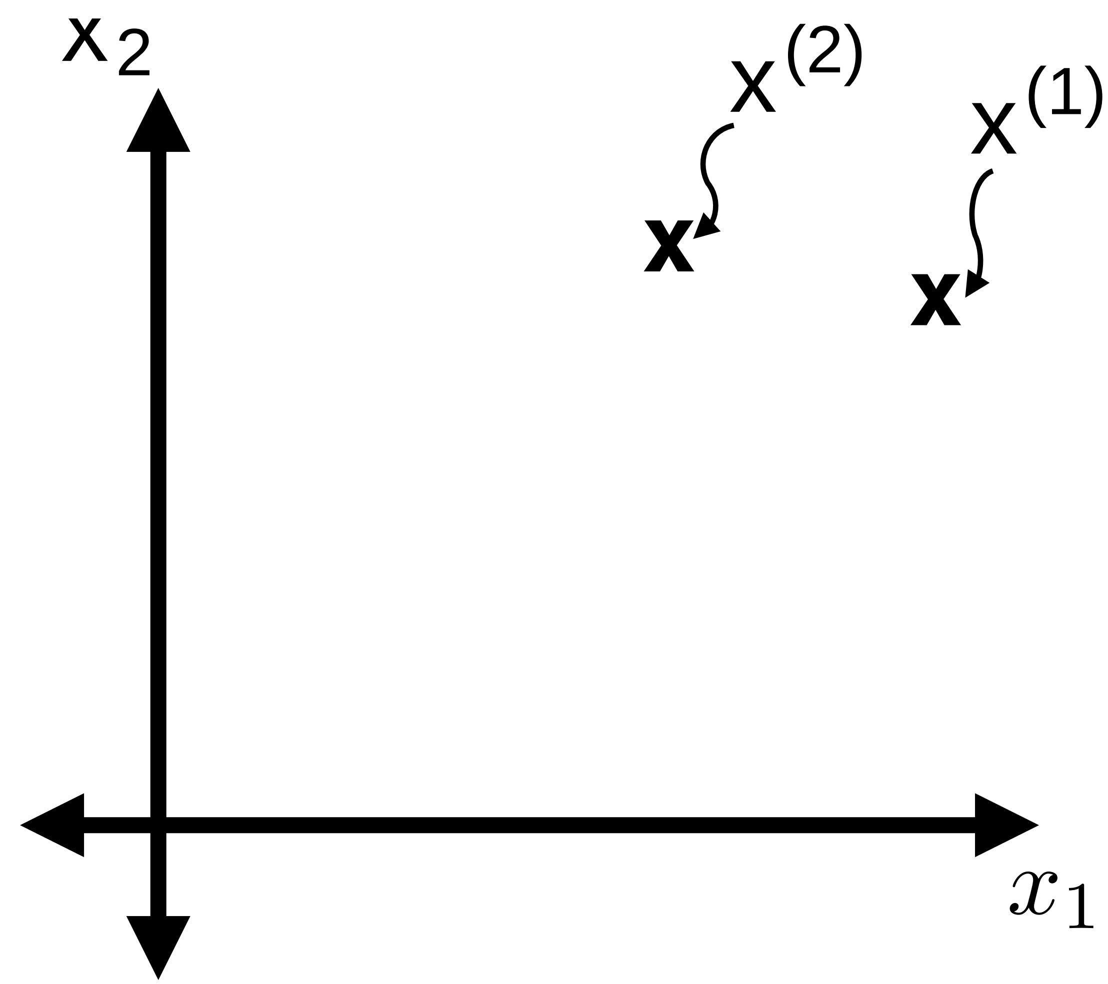


Getting started

What do we have? (Training) data

- *n* training data points
- For data point $i \in \{1, \ldots, n\}$
  - Feature vector

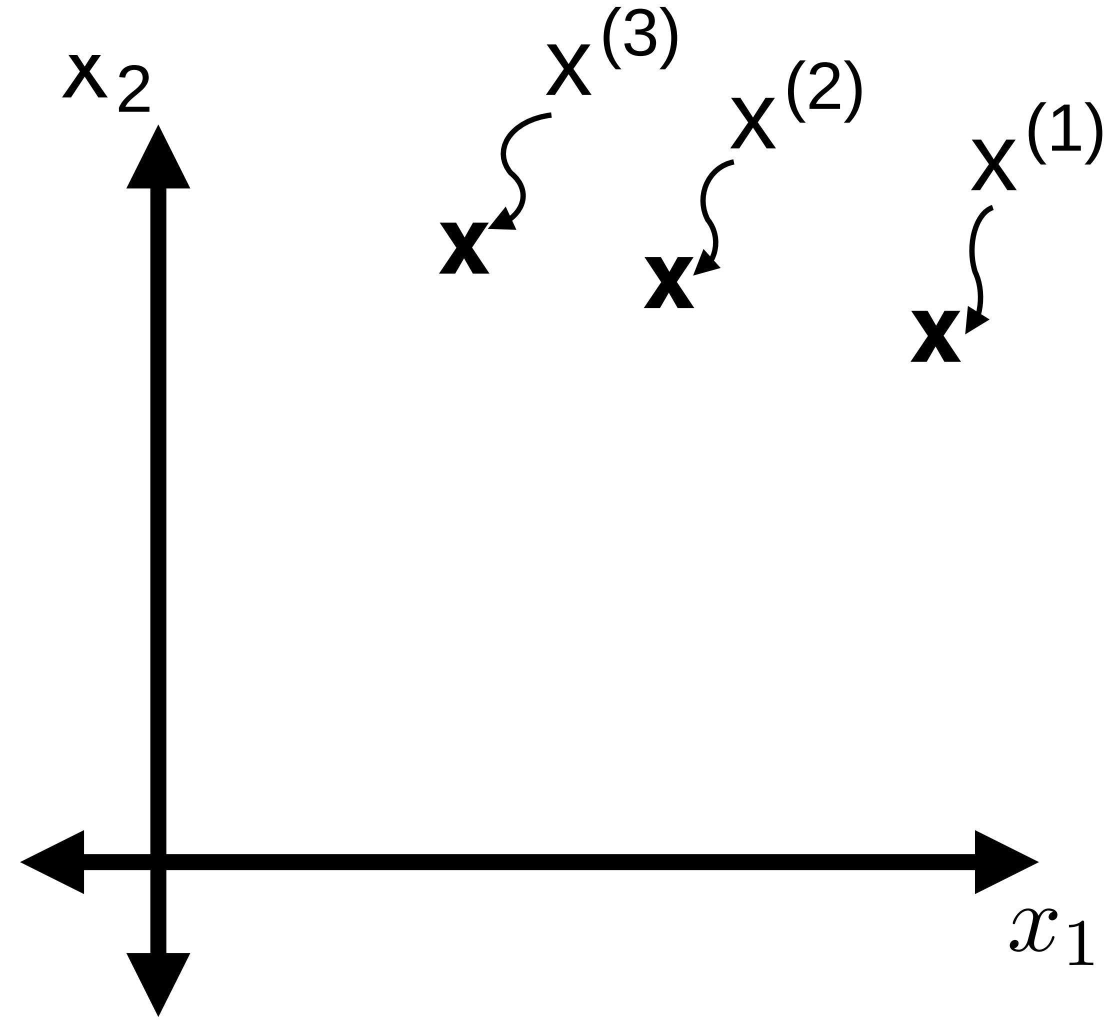
    $x^{(i)} = (x_1^{(i)}, \ldots, x_d^{(i)})^{!} \;\; ! \;\; R^d$



Getting started

What do we have? (Training) data

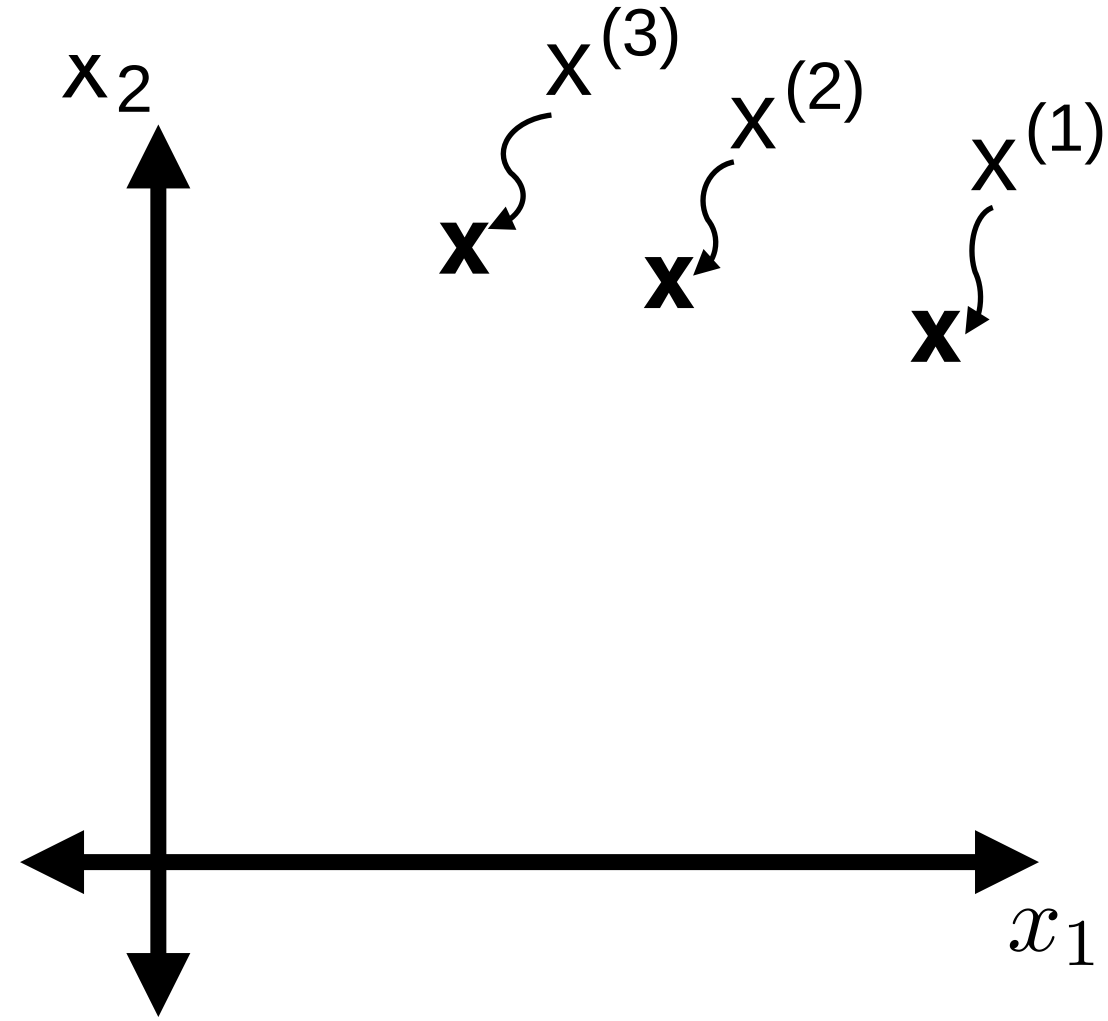
- *n* training data points
- For data point $i \in \{1, \ldots, n\}$
  - Feature vector
    $x^{(i)} = (x_1^{(i)}, \ldots, x_d^{(i)})^{!} \;\; ! \;\; R^d$



Getting started

What do we have? (Training) data

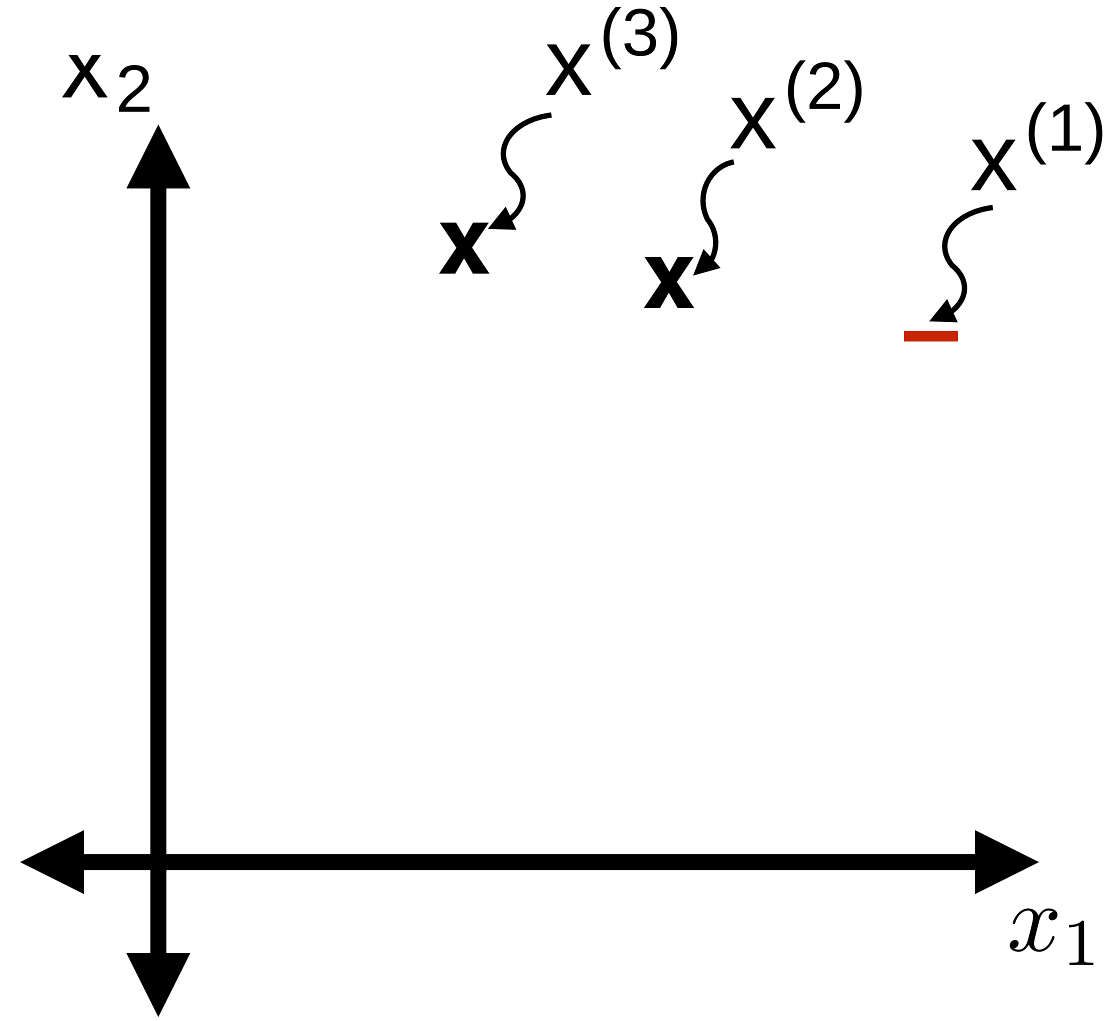
- *n* training data points
- For data point $i \in \{1, \ldots, n\}$
  - Feature vector $x^{(i)} = (x_1^{(i)}, \ldots, x_d^{(i)})^{!} \;\; ! \;\; R^d$
  - Label $y^{(i)} \;\; ! \;\; \{" \; 1, +1\}$



Getting started

What do we have? (Training) data

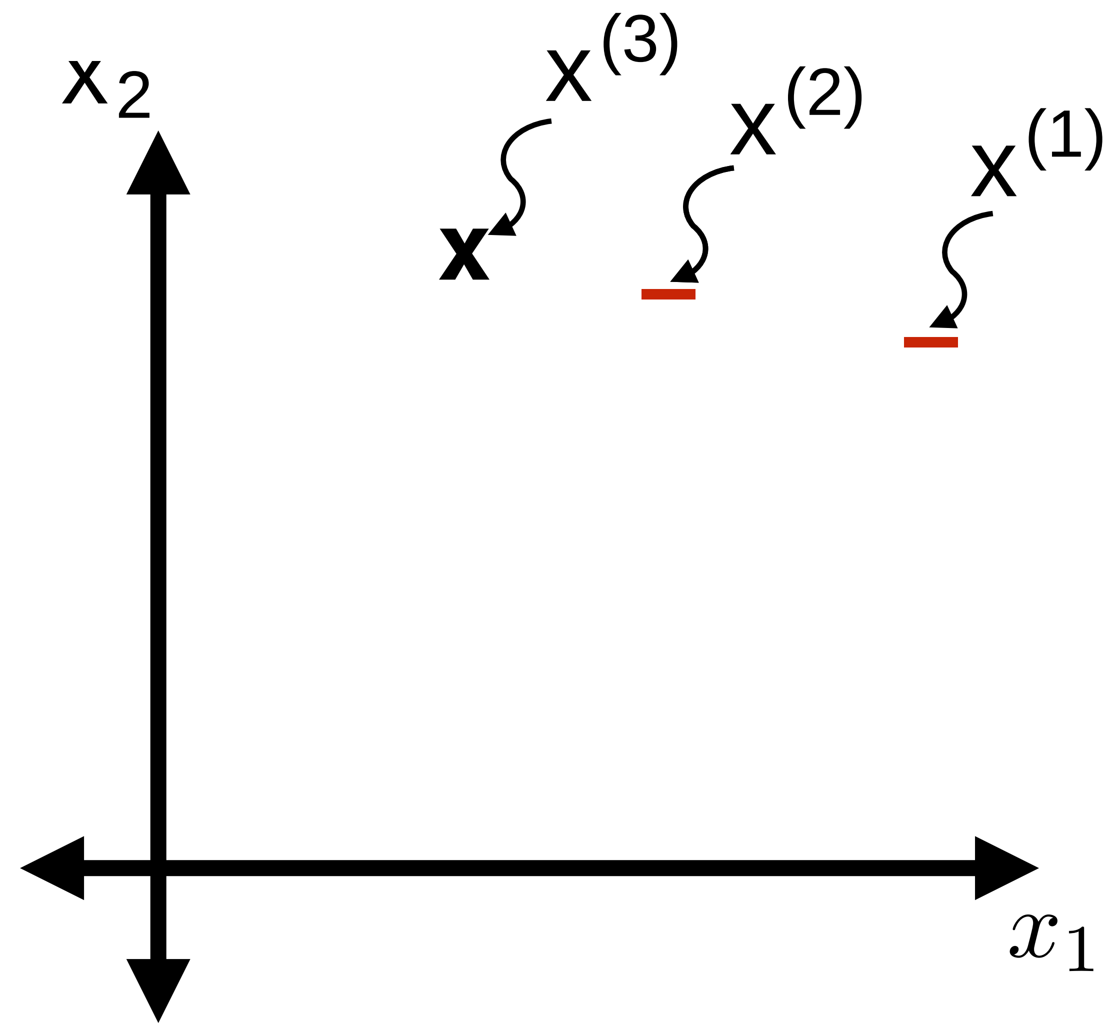
- n training data points
- For data point $i \in \{1, \ldots, n\}$
    - Feature vector $x^{(i)} = (x_1^{(i)}, \ldots, x_d^{(i)})^! \; ! \; R^d$
    - Label $y^{(i)} \; ! \; \{" \, 1, +1\}$



Getting started

What do we have? (Training) data

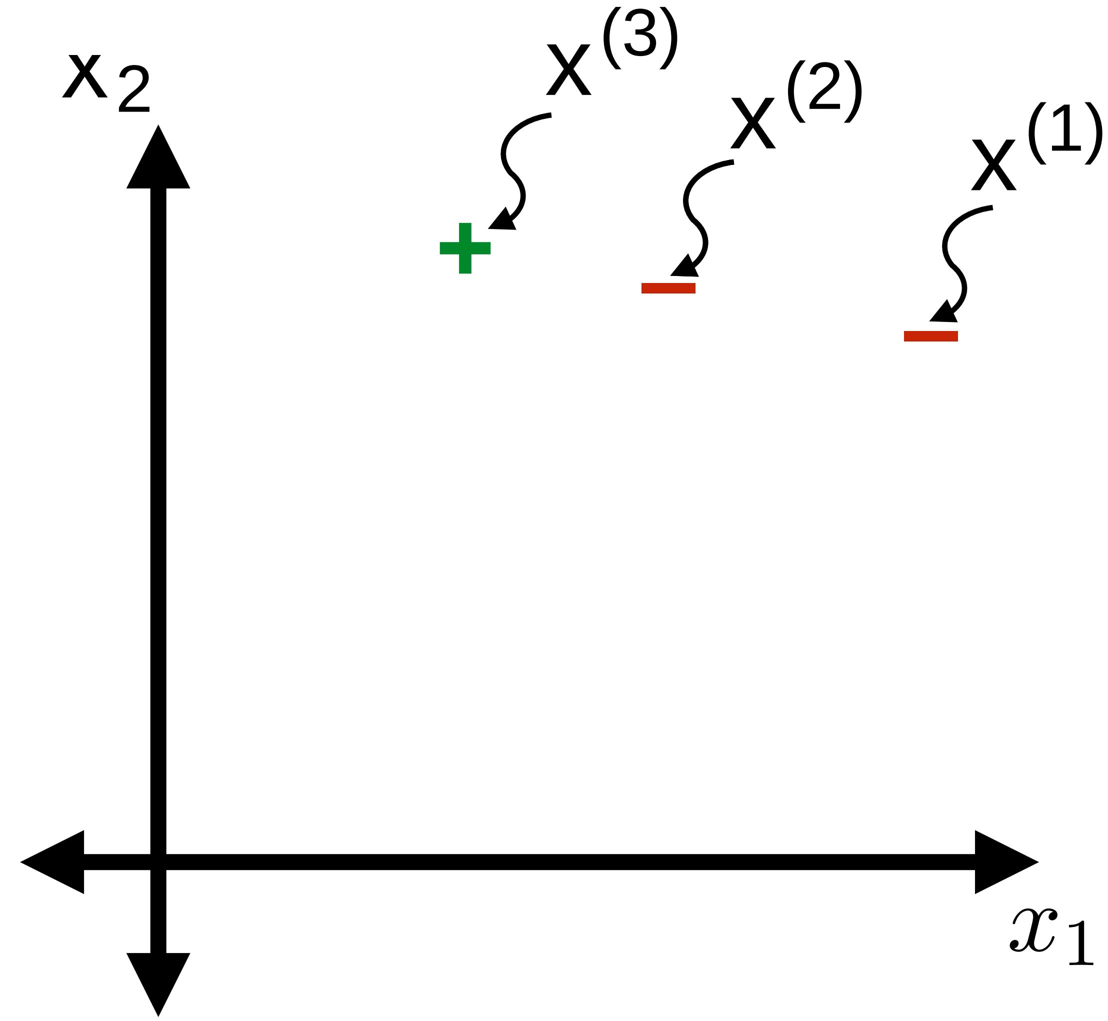
- *n* training data points
- For data point $i \in \{1, \ldots, n\}$
  - Feature vector $x^{(i)} = (x_1^{(i)}, \ldots, x_d^{(i)})^\top \in \mathbb{R}^d$
  - Label $y^{(i)} \in \{-1, +1\}$



Getting started

What do we have? (Training) data

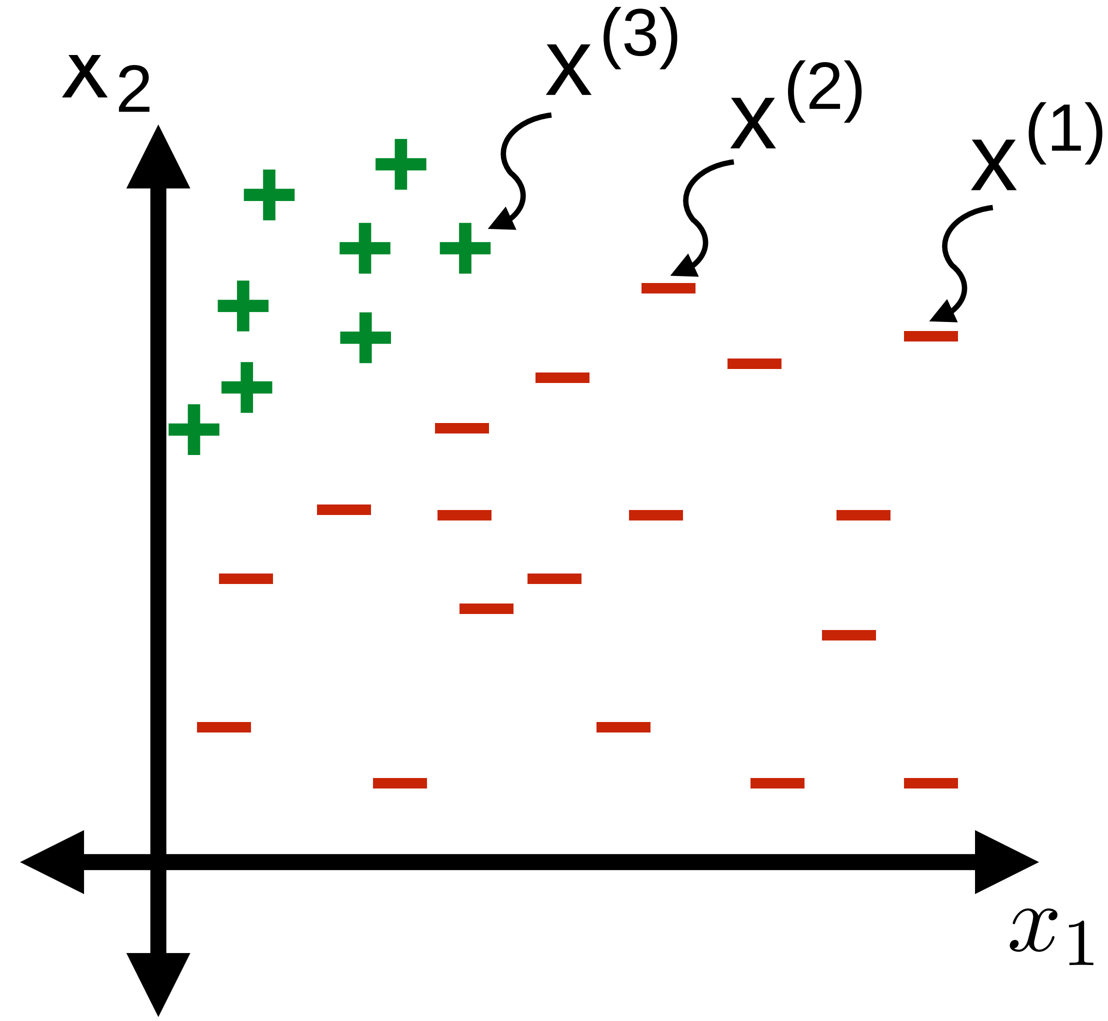
- *n* training data points
- For data point $i \in \{1, \ldots, n\}$
  - Feature vector
    $x^{(i)} = (x_1^{(i)}, \ldots, x_d^{(i)})^{!} \; ! \; R^d$
  - Label $y^{(i)} \; ! \; \{" \; 1, +1\}$



Getting started

What do we have? (Training) data

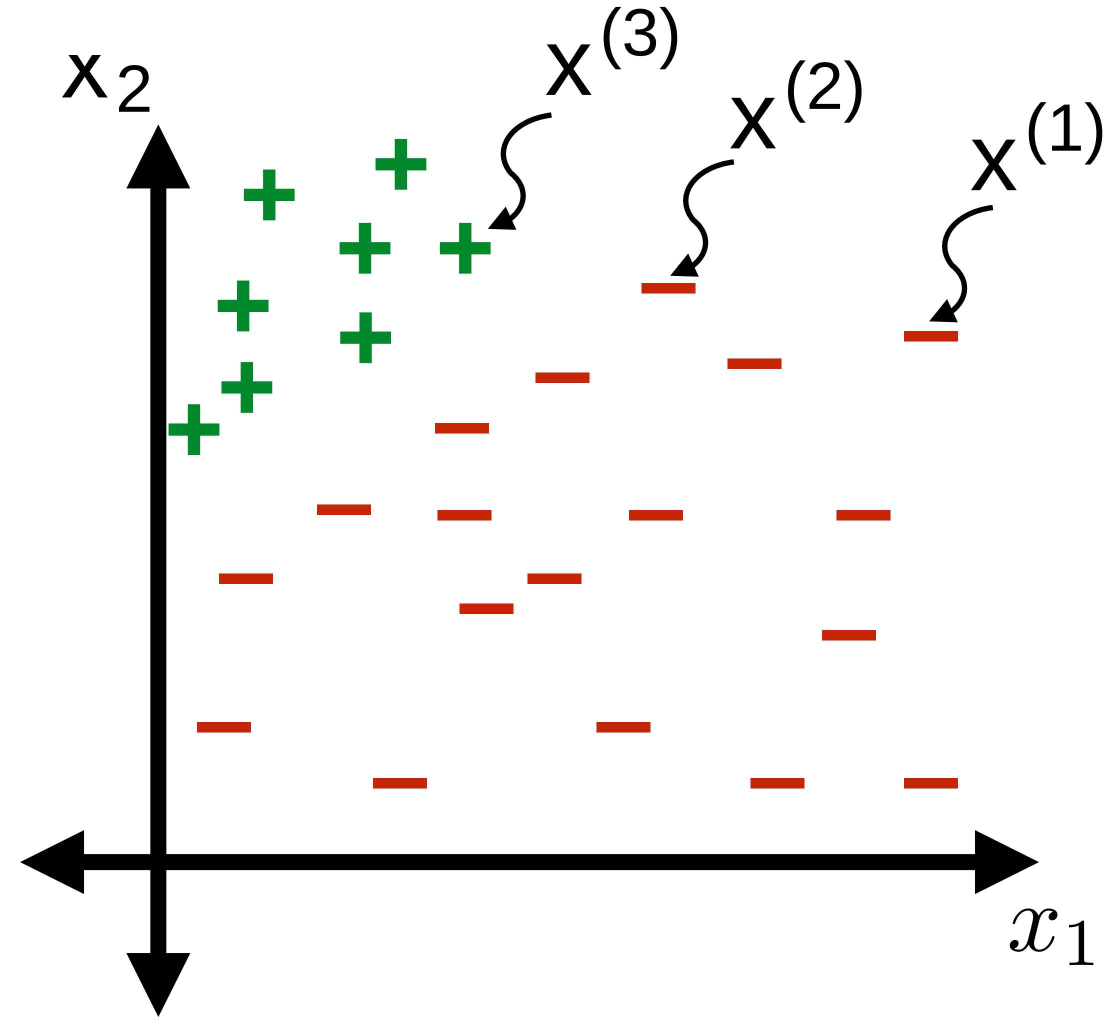
- *n* training data points
- For data point $i \in \{1, \ldots, n\}$
  - Feature vector
    $x^{(i)} = (x_1^{(i)}, \ldots, x_d^{(i)})^{!} \; ! \; R^d$
  - Label $y^{(i)} \; ! \; \{" \; 1, +1\}$



Getting started

What do we have? (Training) data

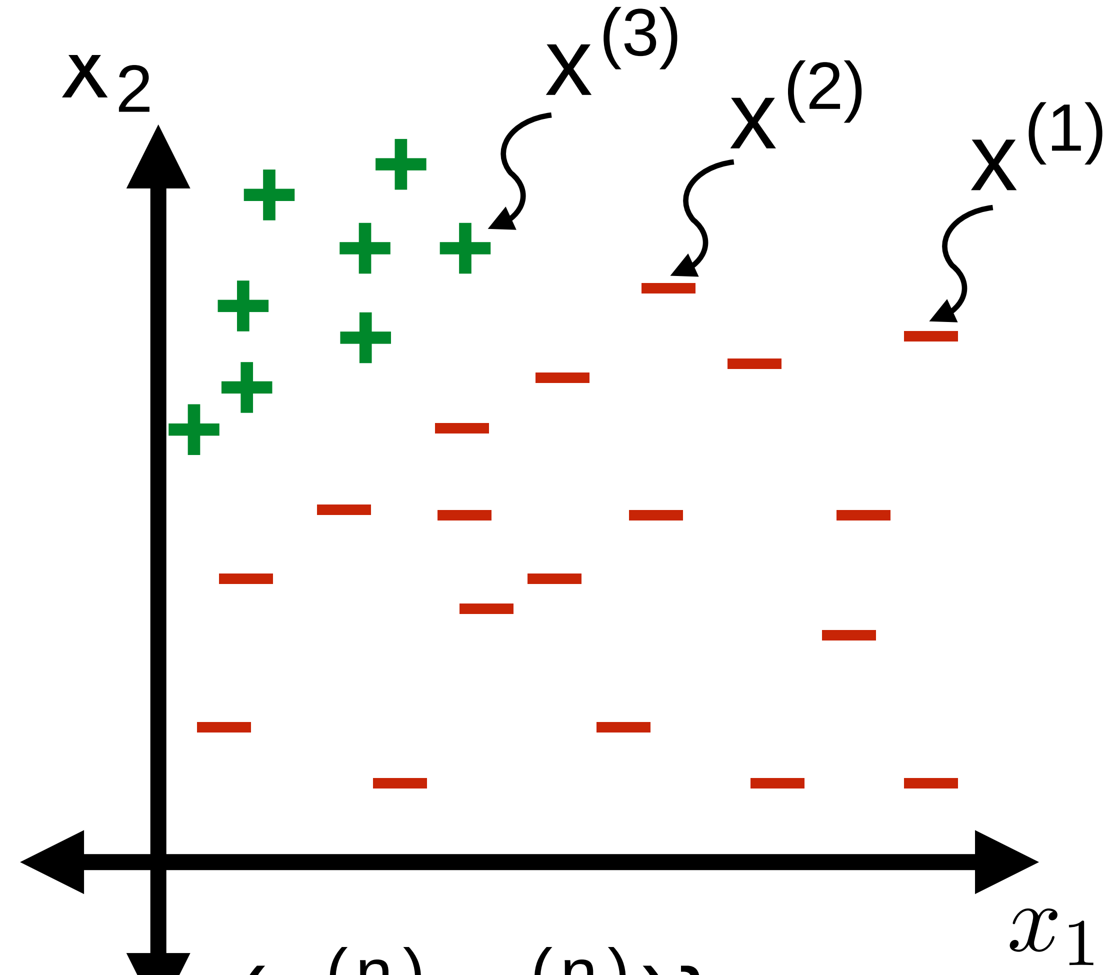
- *n* training data points
- For data point $i \in \{1, \ldots, n\}$
  - Feature vector
    $x^{(i)} = (x_1^{(i)}, \ldots, x_d^{(i)})^{!} \;\; ! \;\; R^d$
  - Label $y^{(i)} \;\; ! \;\; \{" \; 1, +1\}$
- Training data



Getting started

What do we have? (Training) data

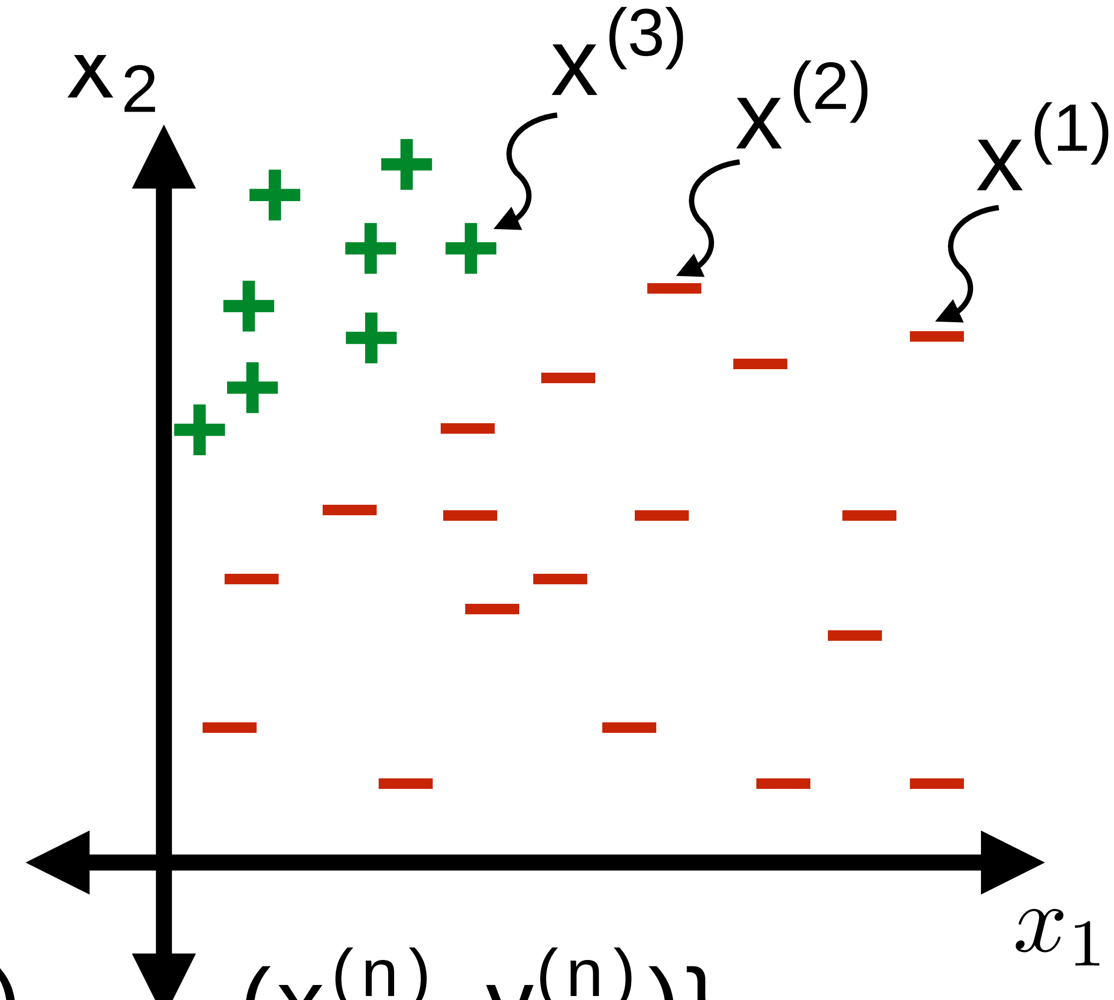
- *n* training data points
- For data point $i \in \{1, \ldots, n\}$
  - Feature vector $x^{(i)} = (x_1^{(i)}, \ldots, x_d^{(i)})^\top \in \mathbb{R}^d$
  - Label $y^{(i)} \in \{-1, +1\}$
- Training data $D_n = \{(x^{(1)}, y^{(1)}), \ldots, (x^{(n)}, y^{(n)})\}$



Getting started

What do we have? (Training) data

- *n* training data points
- For data point $i \in \{1, \ldots, n\}$
  - Feature vector $x^{(i)} = (x_1^{(i)}, \ldots, x_d^{(i)})^\top \in \mathbb{R}^d$
  - Label $y^{(i)} \in \{-1, +1\}$
- Training data $D_n = \{(x^{(1)}, y^{(1)}), \ldots, (x^{(n)}, y^{(n)})\}$

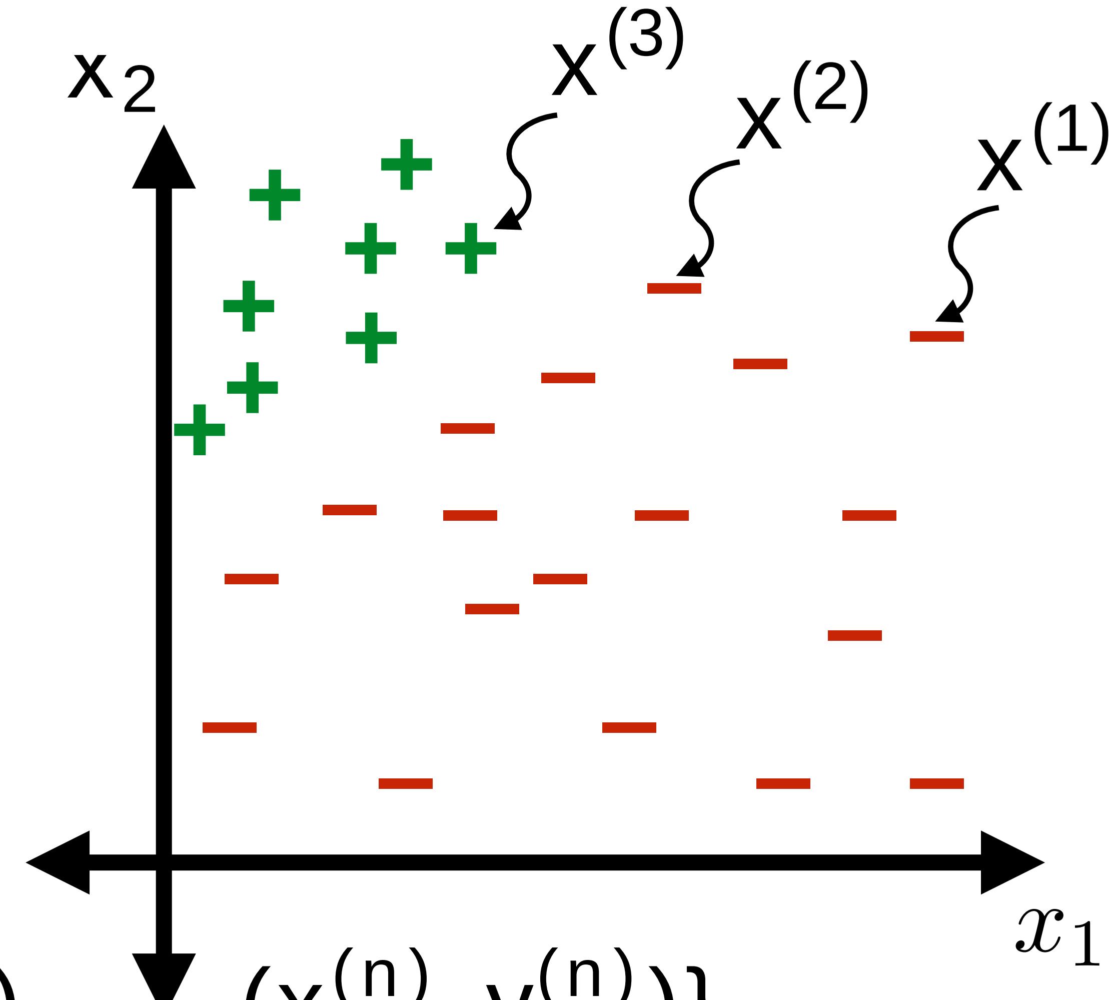


What do we want?

Getting started

What do we have? (Training) data

- *n* training data points
- For data point $i \in \{1, \ldots, n\}$
  - Feature vector $x^{(i)} = (x_1^{(i)}, \ldots, x_d^{(i)})^! \; ! \; R^d$
  - Label $y^{(i)} \; ! \; \{" \; 1, +1\}$
- Training data $D_n = \{(x^{(1)}, y^{(1)}), \ldots, (x^{(n)}, y^{(n)})\}$

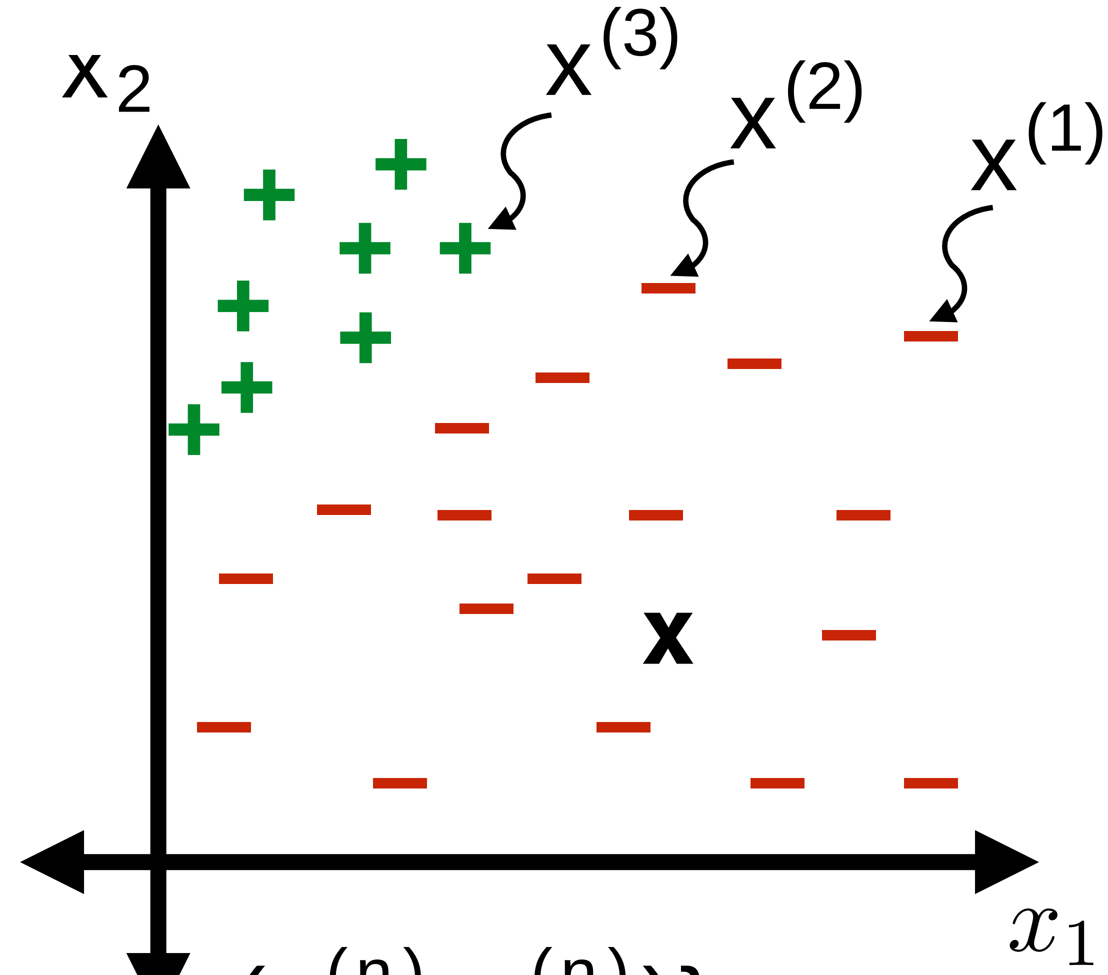


What do we want? A good way to label new points

Getting started

What do we have? (Training) data

- *n* training data points
- For data point $i \in \{1, \ldots, n\}$
  - Feature vector $x^{(i)} = (x_1^{(i)}, \ldots, x_d^{(i)})^\top \in \mathbb{R}^d$
  - Label $y^{(i)} \in \{-1, +1\}$
- Training data $D_n = \{(x^{(1)}, y^{(1)}), \ldots, (x^{(n)}, y^{(n)})\}$

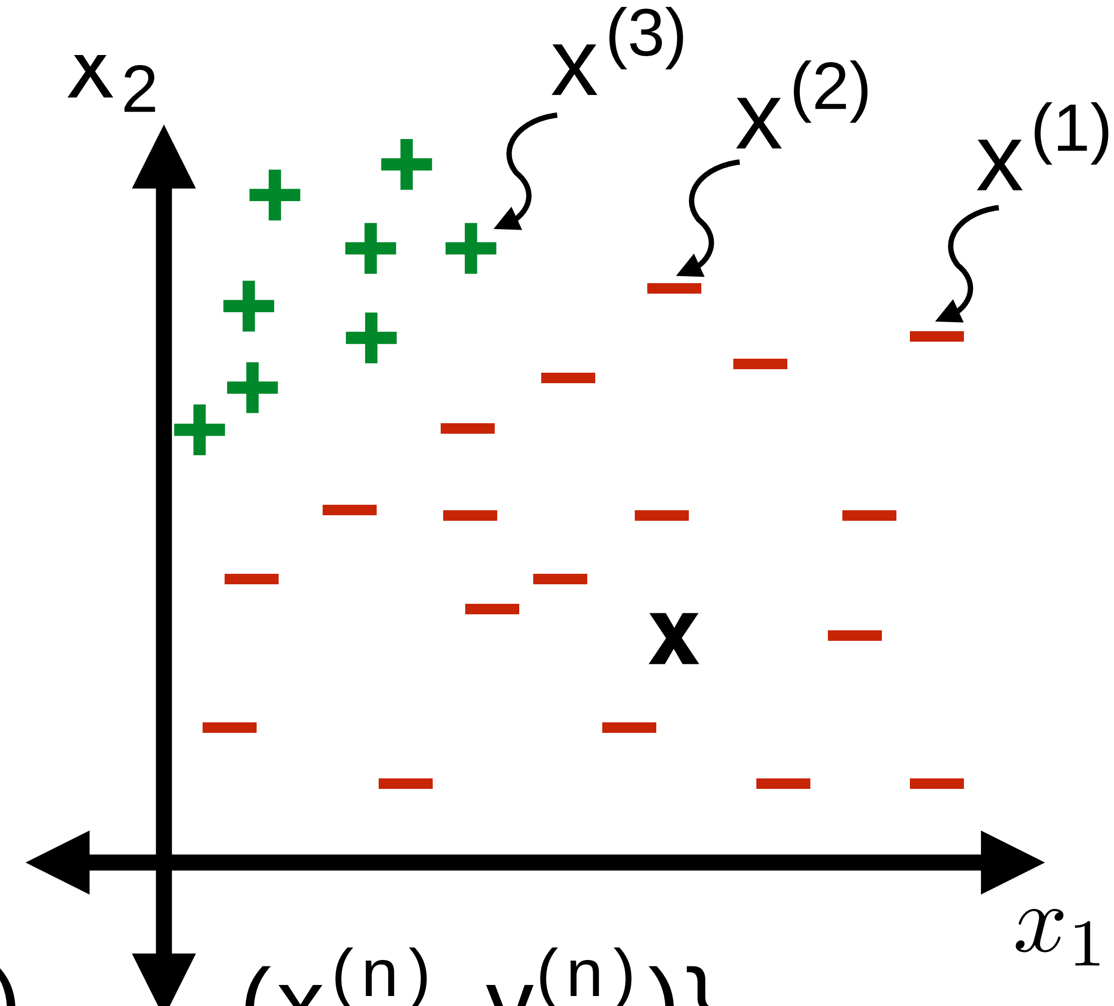


What do we want? A good way to label new points

Getting started

What do we have? (Training) data

- *n* training data points
- For data point $i \in \{1, \ldots, n\}$
  - Feature vector $x^{(i)} = (x_1^{(i)}, \ldots, x_d^{(i)})^\top \in R^d$
  - Label $y^{(i)} \in \{-1, +1\}$
- Training data $D_n = \{(x^{(1)}, y^{(1)}), \ldots, (x^{(n)}, y^{(n)})\}$

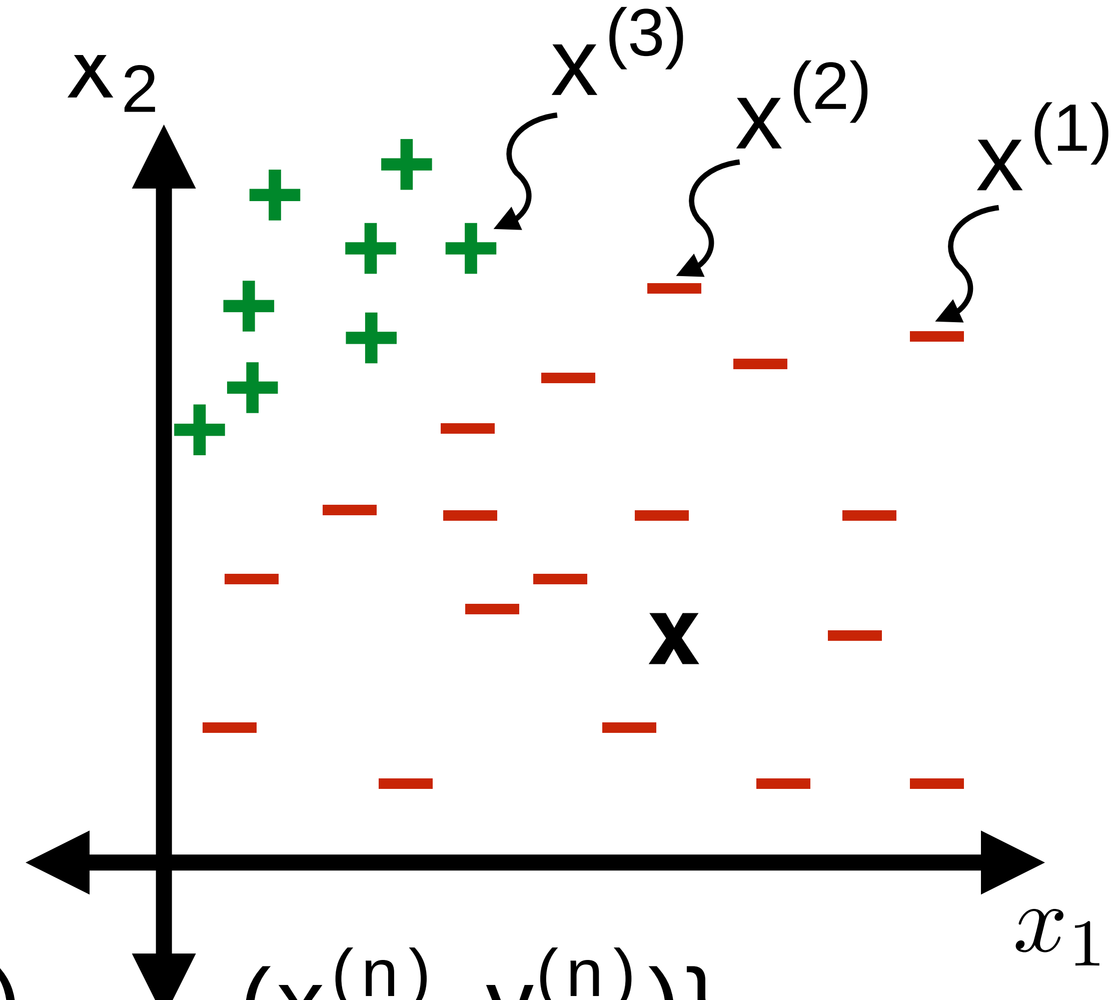


What do we want? A good way to label new points

Getting started

What do we have? (Training) data

- *n* training data points
- For data point $i \in \{1, \ldots, n\}$
    - Feature vector $x^{(i)} = (x_1^{(i)}, \ldots, x_d^{(i)})^\top \in \mathbb{R}^d$
    - Label $y^{(i)} \in \{-1, +1\}$
- Training data $D_n = \{(x^{(1)}, y^{(1)}), \ldots, (x^{(n)}, y^{(n)})\}$

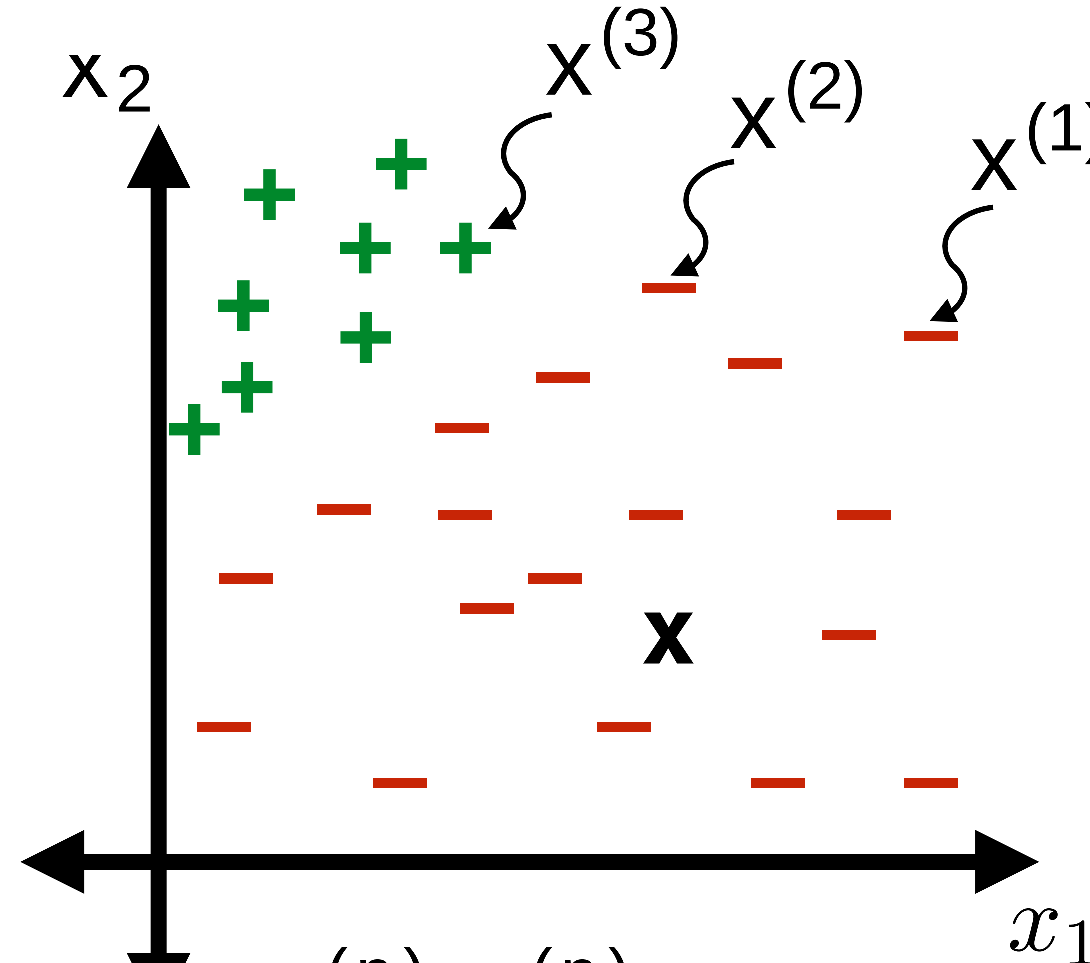


What do we want? A good way to label new points

Getting started

What do we have? (Training) data

- *n* training data points
- For data point $i \in \{1, \ldots, n\}$
  - Feature vector $x^{(i)} = (x_1^{(i)}, \ldots, x_d^{(i)})^! \; ! \; R^d$
  - Label $y^{(i)} \; ! \; \{" \, 1, +1\}$
- Training data $D_n = \{(x^{(1)}, y^{(1)}), \ldots, (x^{(n)}, y^{(n)})\}$

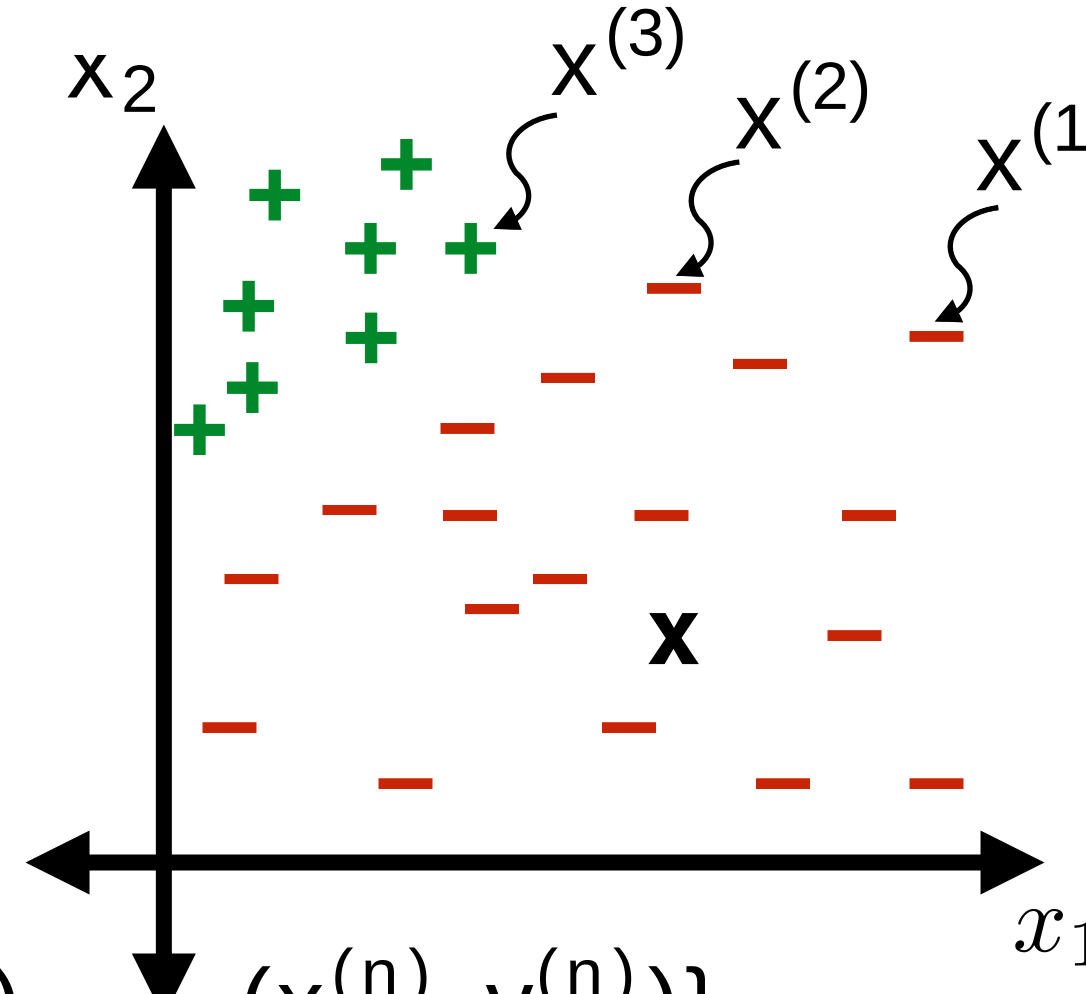


What do we want? A good way to label new points

Getting started

What do we have? (Training) data

- *n* training data points
- For data point $i \in \{1, \ldots, n\}$
    - Feature vector $x^{(i)} = (x_1^{(i)}, \ldots, x_d^{(i)})^\top \in \mathbb{R}^d$
    - Label $y^{(i)} \in \{-1, +1\}$
- Training data $D_n = \{(x^{(1)}, y^{(1)}), \ldots, (x^{(n)}, y^{(n)})\}$

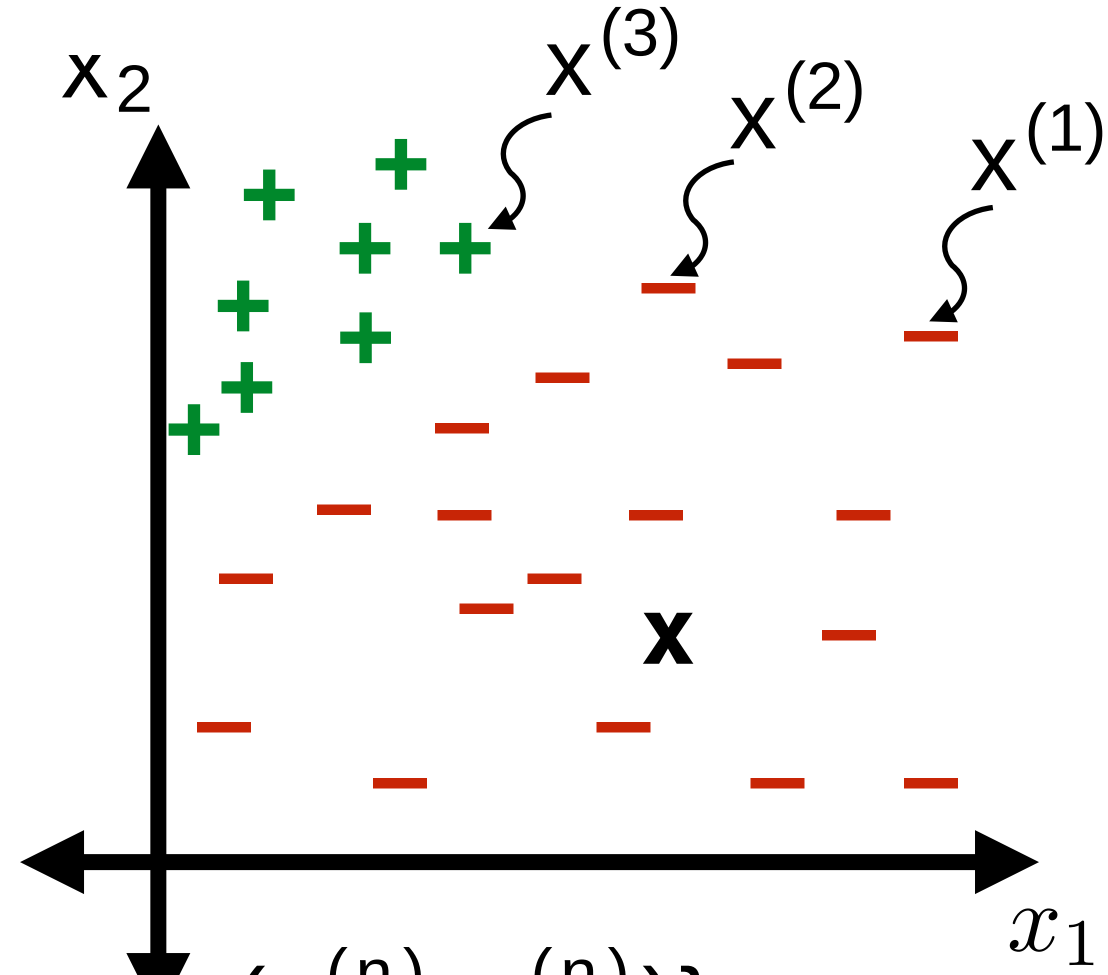


What do we want? A good way to label new points

- How to label?

Getting started

What do we have? (Training) data



- n training data points
- For data point $i \in \{1, \ldots, n\}$
  - Feature vector $x^{(i)} = (x_1^{(i)}, \ldots, x_d^{(i)})^\top \in \mathbb{R}^d$
  - Label $y^{(i)} \in \{-1, +1\}$
- Training data $\mathcal{D}_n = \{(x^{(1)}, y^{(1)}), \ldots, (x^{(n)}, y^{(n)})\}$

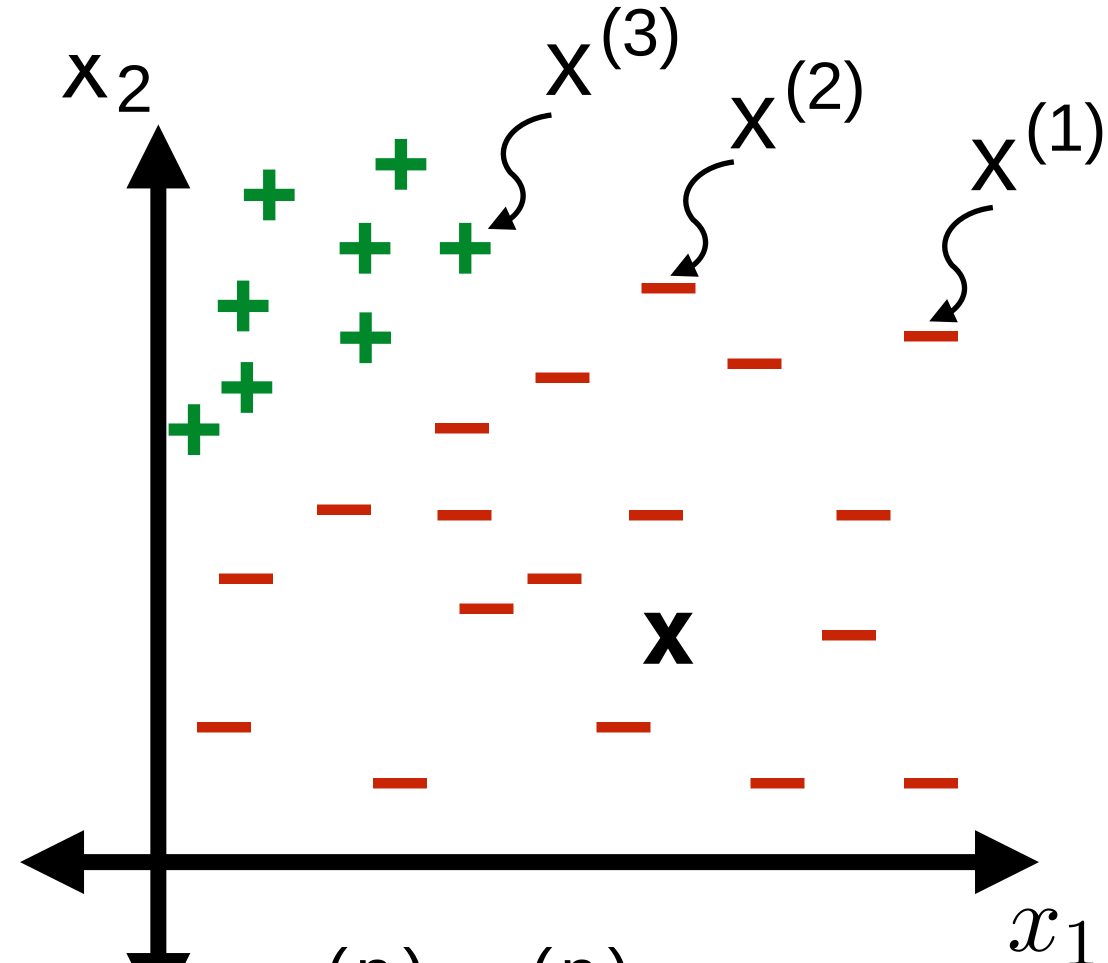
What do we want? A good way to label new points

- How to label? Hypothesis $h : \mathbb{R}^d \to \{-1, +1\}$

Getting started

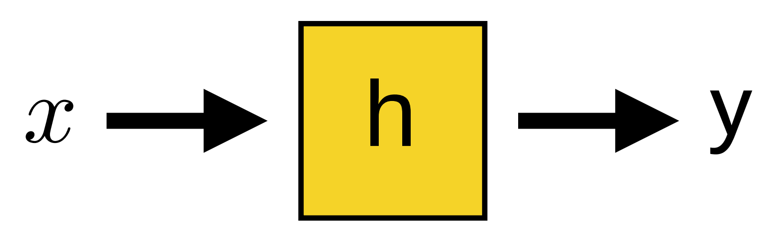
What do we have? (Training) data

- *n* training data points
- For data point $i \in \{1, \ldots, n\}$
  - Feature vector $x^{(i)} = (x_1^{(i)}, \ldots, x_d^{(i)})^\top \in \mathbb{R}^d$
  - Label $y^{(i)} \in \{-1, +1\}$
- Training data $D_n = \{(x^{(1)}, y^{(1)}), \ldots, (x^{(n)}, y^{(n)})\}$



What do we want? A good way to label new points

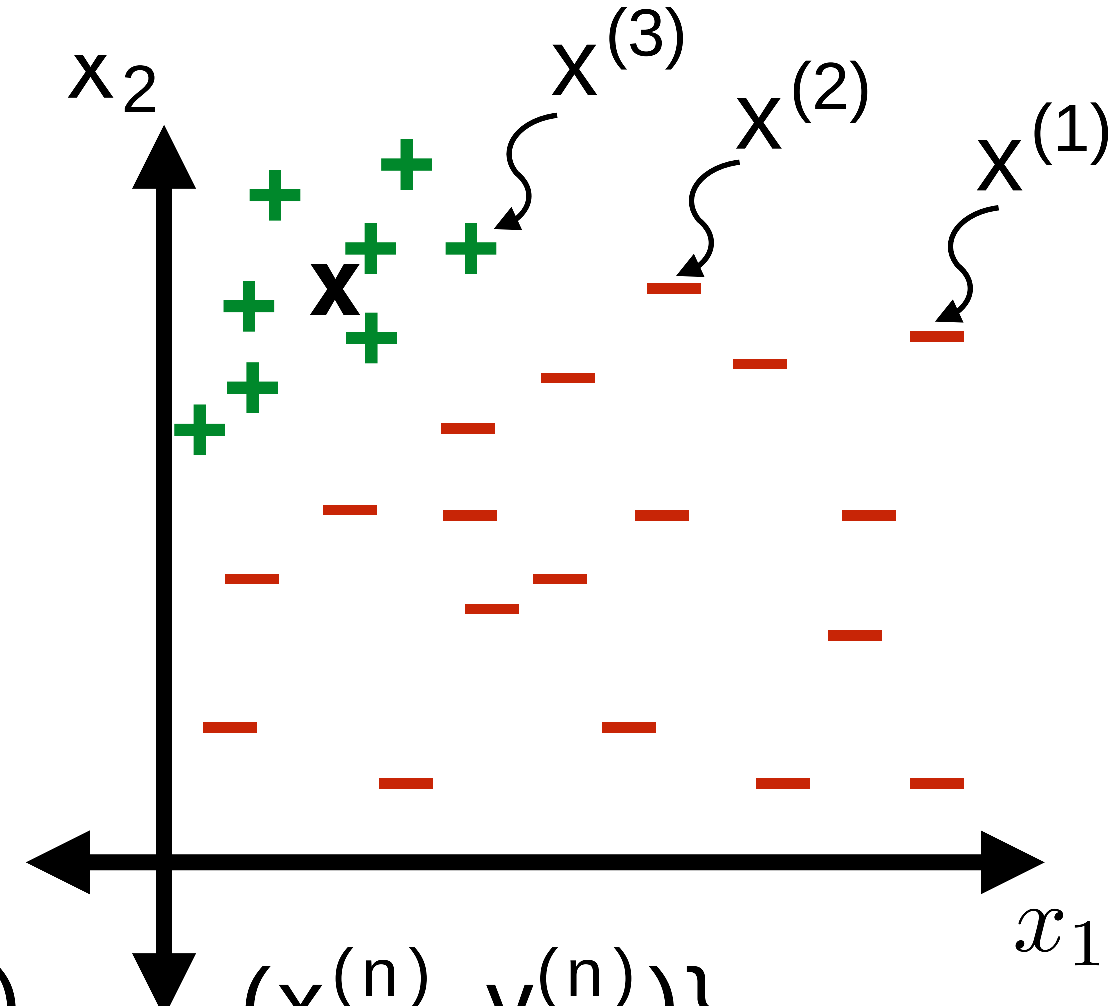
- How to label? Hypothesis $h : \mathbb{R}^d \to \{-1, +1\}$



Getting started

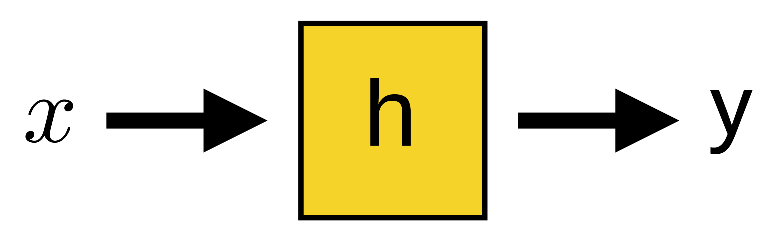
What do we have? (Training) data

- n training data points
- For data point $i \in \{1, \ldots, n\}$
  - Feature vector $x^{(i)} = (x_1^{(i)}, \ldots, x_d^{(i)})^\top \in \mathbb{R}^d$
  - Label $y^{(i)} \in \{-1, +1\}$
- Training data $D_n = \{(x^{(1)}, y^{(1)}), \ldots, (x^{(n)}, y^{(n)})\}$

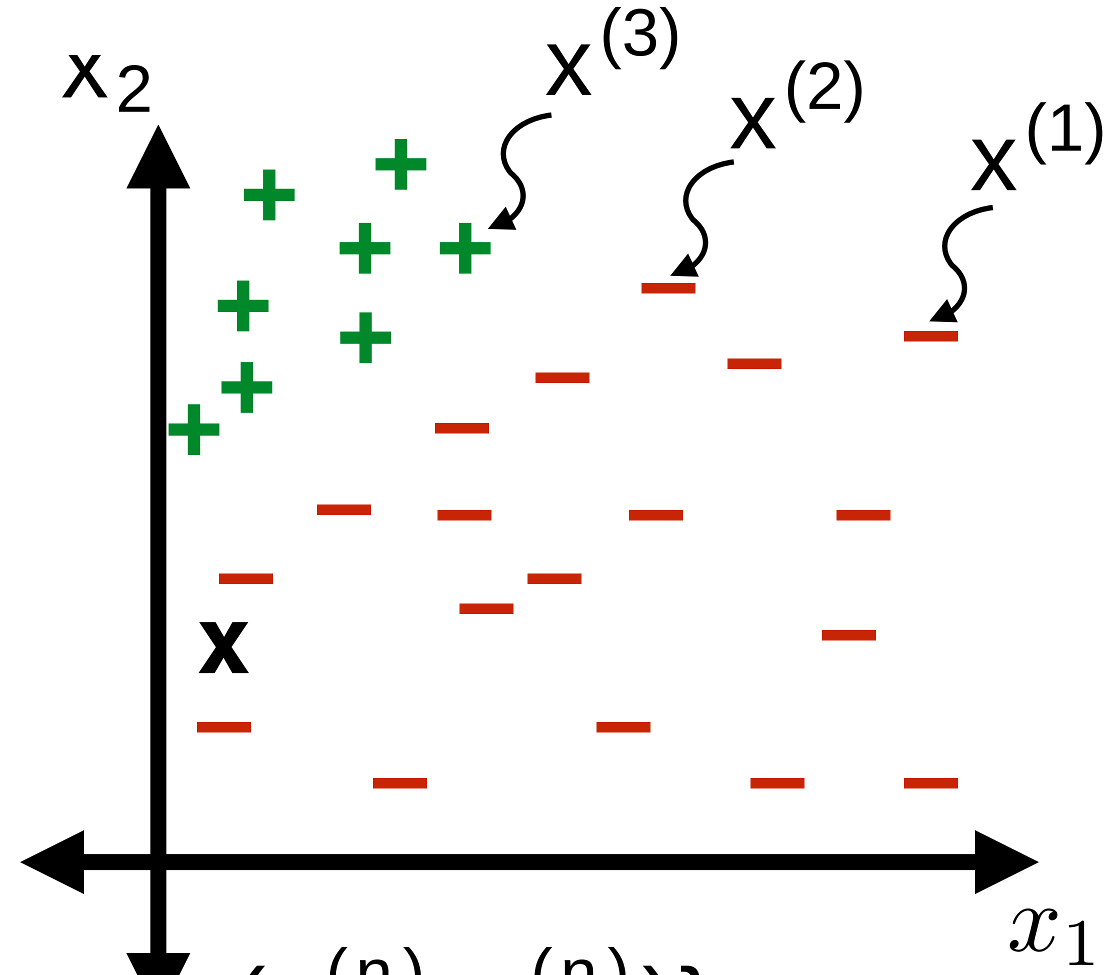


What do we want? A good way to label new points

- How to label? Hypothesis $h : \mathbb{R}^d \to \{-1, +1\}$



Getting started



What do we have? (Training) data

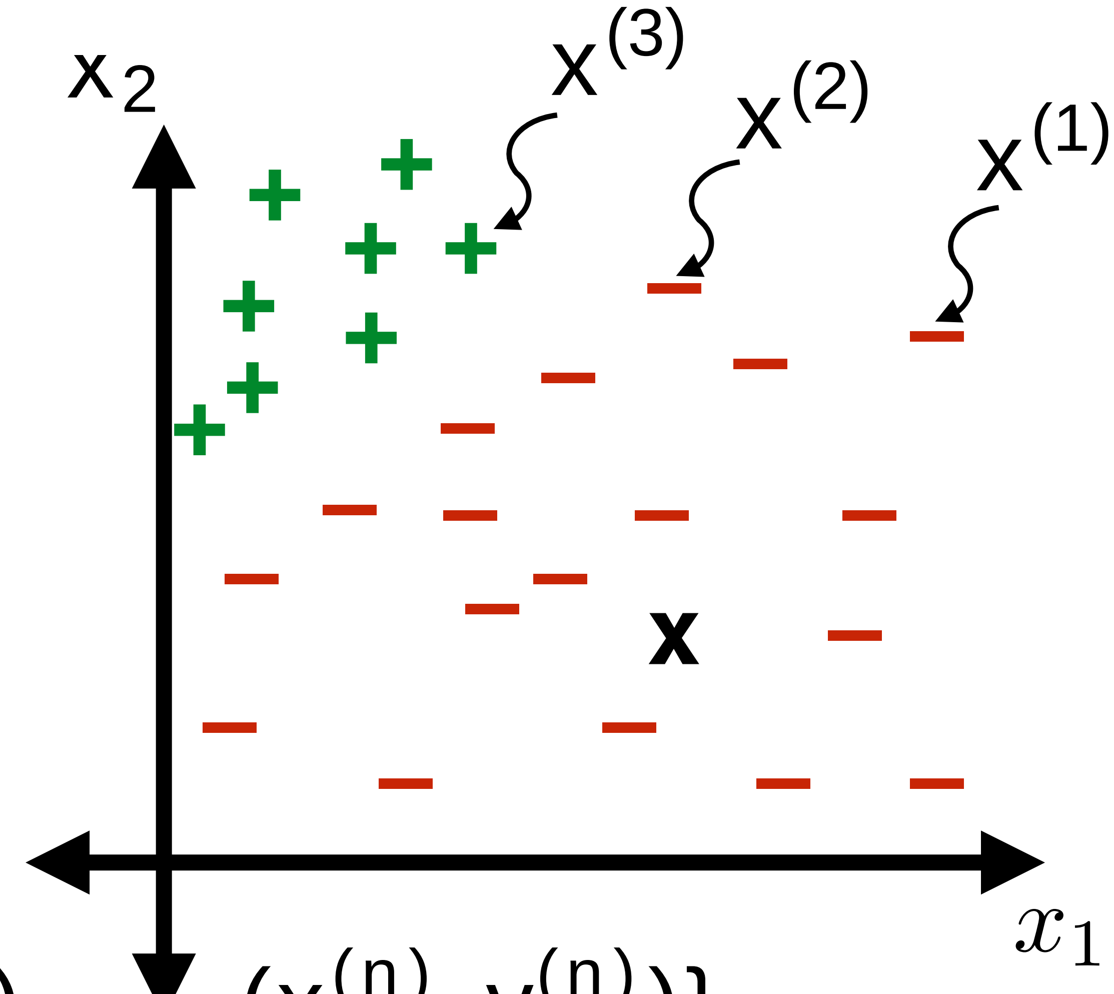
- n training data points
- For data point $i \in \{1, \dots, n\}$
  - Feature vector $x^{(i)} = (x_1^{(i)}, \dots, x_d^{(i)})^\top \in \mathbb{R}^d$
  - Label $y^{(i)} \in \{-1, +1\}$
- Training data $D_n = \{(x^{(1)}, y^{(1)}), \dots, (x^{(n)}, y^{(n)})\}$

What do we want? A good way to label new points

- How to label? Hypothesis $h : \mathbb{R}^d \to \{-1, +1\}$

$$x \to \boxed{h} \to y$$

Getting started

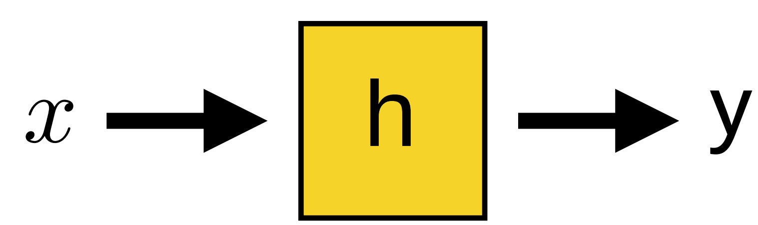


What do we have? (Training) data

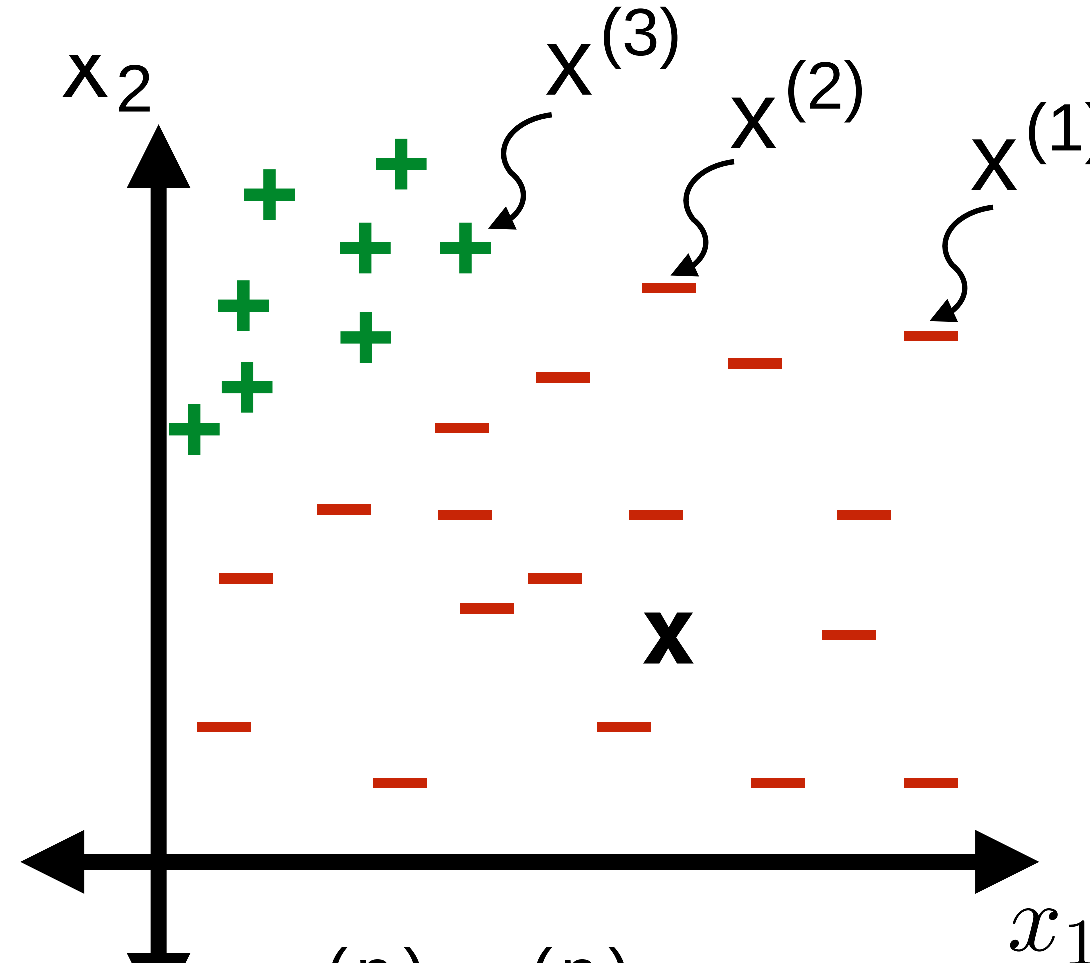
- *n* training data points
- For data point $i \in \{1, \ldots, n\}$
  - Feature vector $x^{(i)} = (x_1^{(i)}, \ldots, x_d^{(i)})^\top \in \mathbb{R}^d$
  - Label $y^{(i)} \in \{-1, +1\}$
- Training data $D_n = \{(x^{(1)}, y^{(1)}), \ldots, (x^{(n)}, y^{(n)})\}$

What do we want? A good way to label new points

- How to label? Hypothesis $h : \mathbb{R}^d \to \{-1, +1\}$



Getting started

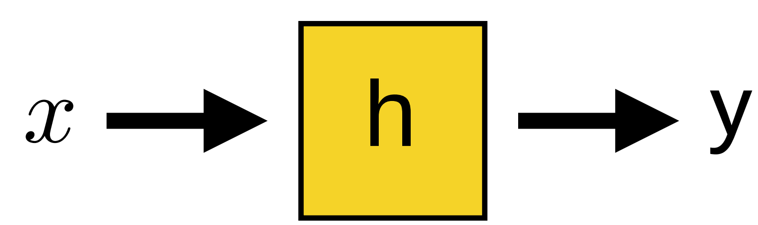


What do we have? (Training) data

- *n* training data points
- For data point $i \in \{1, \ldots, n\}$
    - Feature vector $x^{(i)} = (x_1^{(i)}, \ldots, x_d^{(i)})^\top \in \mathbb{R}^d$
    - Label $y^{(i)} \in \{-1, +1\}$
- Training data $D_n = \{(x^{(1)}, y^{(1)}), \ldots, (x^{(n)}, y^{(n)})\}$

What do we want? A good way to label new points

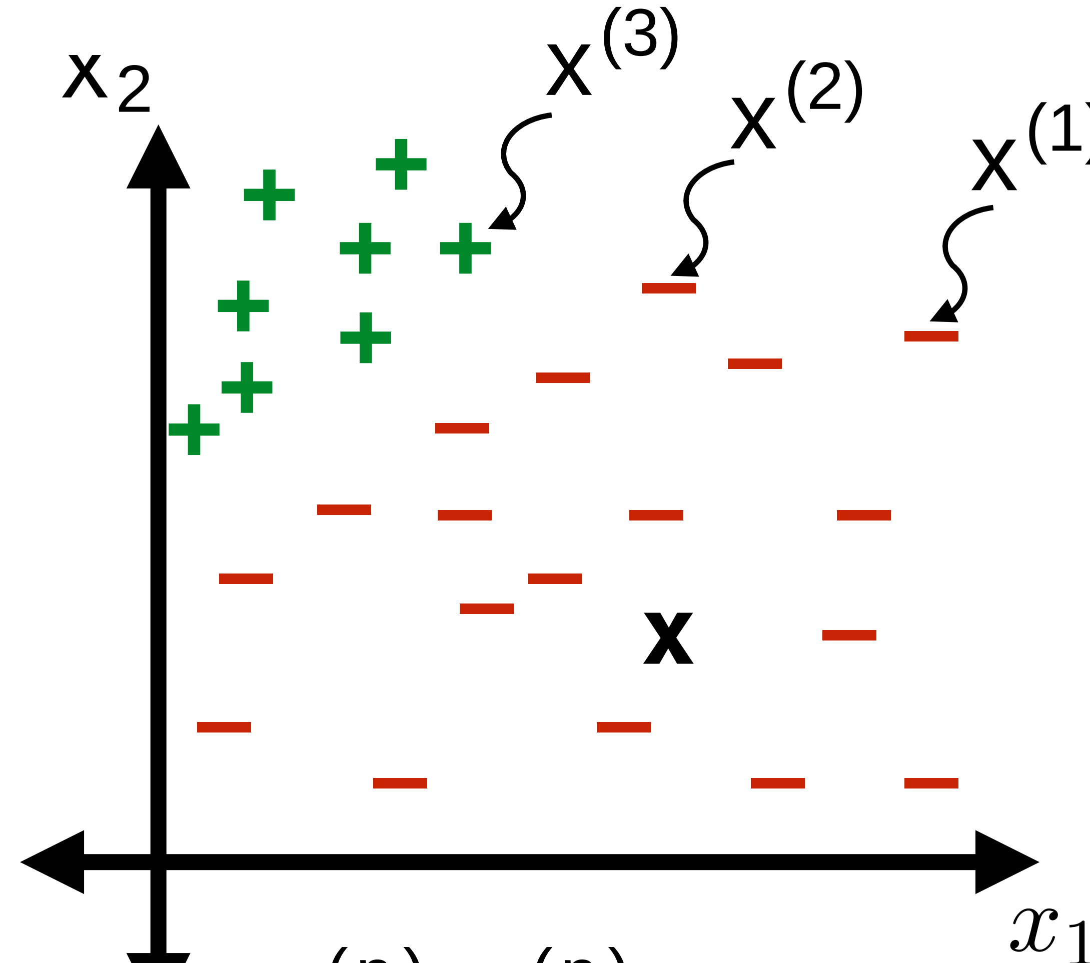
- How to label? Hypothesis $h : \mathbb{R}^d \rightarrow \{-1, +1\}$



- Example *h*: For any *x*, *h*(*x*) = +1

Getting started

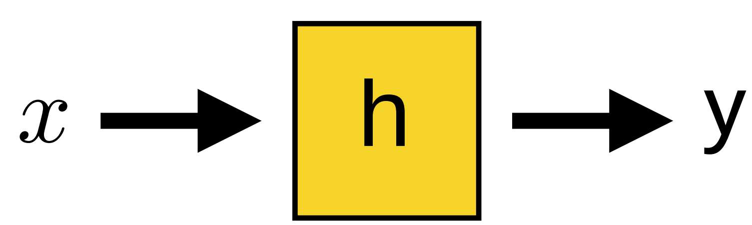
What do we have? (Training) data



- *n* training data points
- For data point $i \in \{1, \ldots, n\}$
    - Feature vector $x^{(i)} = (x_1^{(i)}, \ldots, x_d^{(i)})^\top \in \mathbb{R}^d$
    - Label $y^{(i)} \in \{-1, +1\}$
- Training data $D_n = \{(x^{(1)}, y^{(1)}), \ldots, (x^{(n)}, y^{(n)})\}$

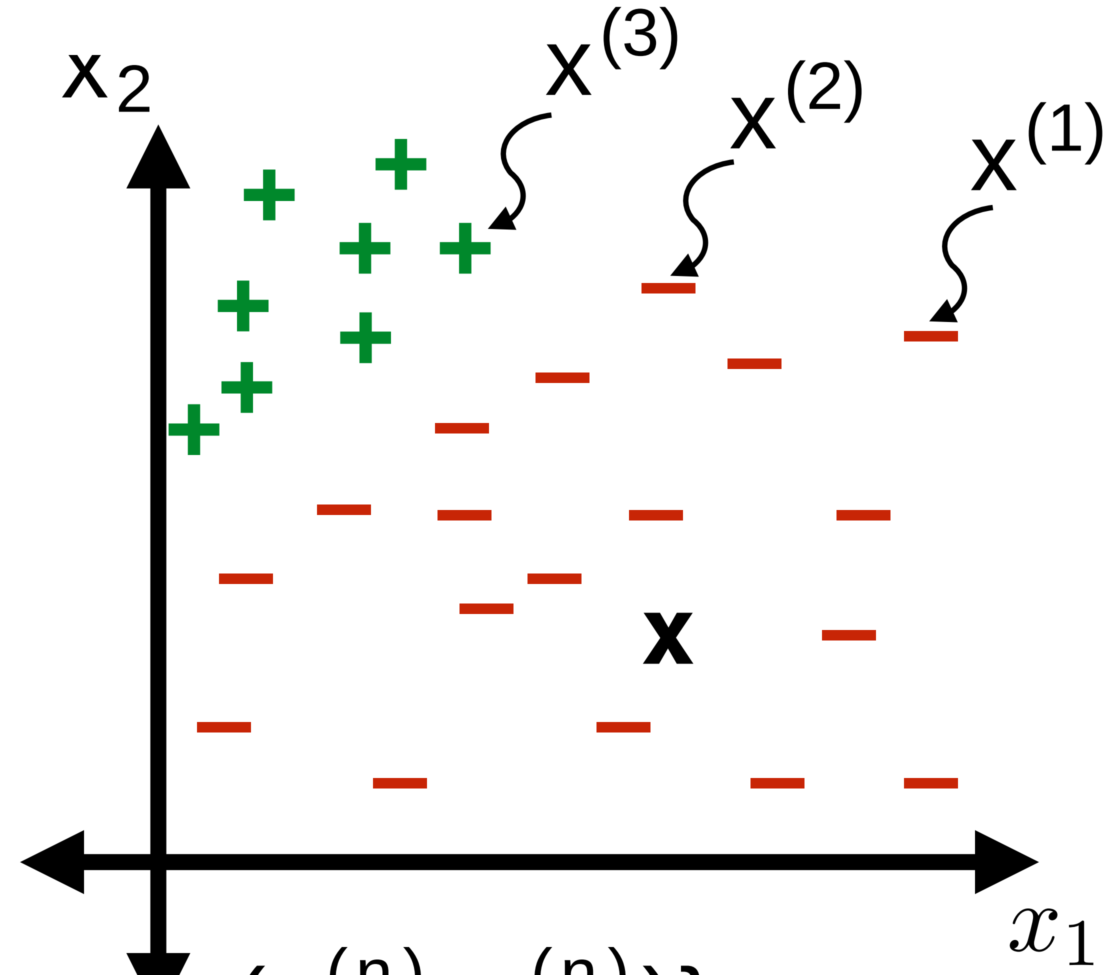
What do we want? A good way to label new points

- How to label? Hypothesis $h : \mathbb{R}^d \to \{-1, +1\}$



- Example *h*: For any *x*, *h*(*x*) = +1
    - Is this a hypothesis?

Getting started

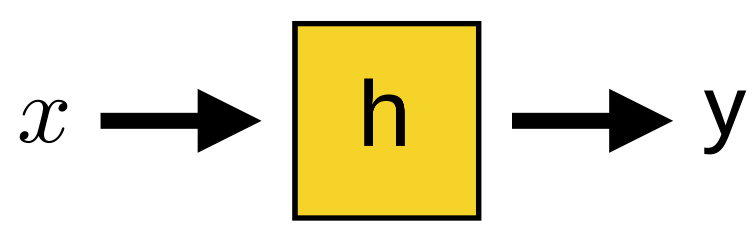


What do we have? (Training) data

- *n* training data points
- For data point $i \in \{1, \ldots, n\}$
  - Feature vector $x^{(i)} = (x_1^{(i)}, \ldots, x_d^{(i)})^{!} \;\; ! \;\; R^d$
  - Label $y^{(i)} \;\; ! \;\; \{" \; 1, +1\}$
- Training data $D_n = \{(x^{(1)}, y^{(1)}), \ldots, (x^{(n)}, y^{(n)})\}$

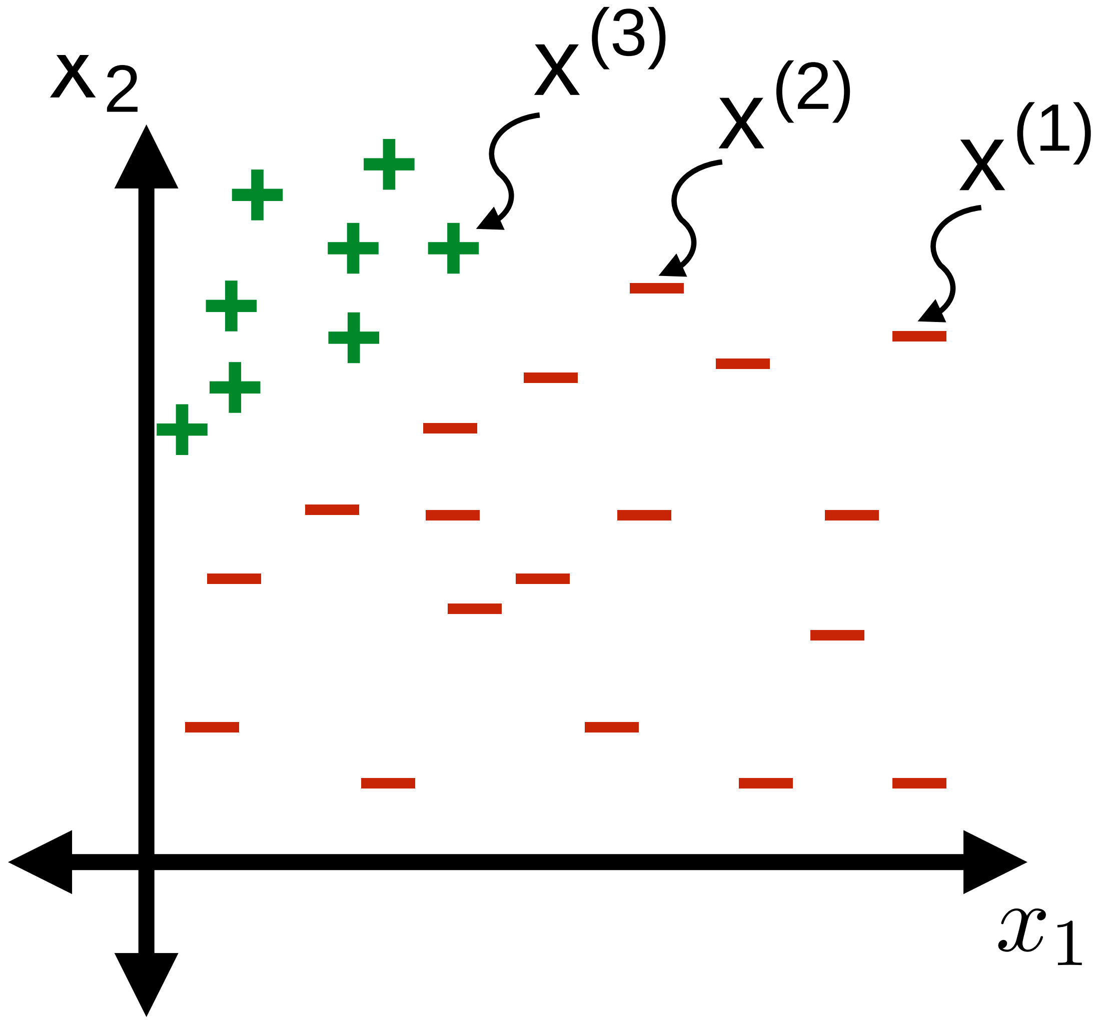
What do we want? A good way to label new points

- How to label? Hypothesis $h : R^d \;\; ! \;\; \{" \; 1, +1\}$



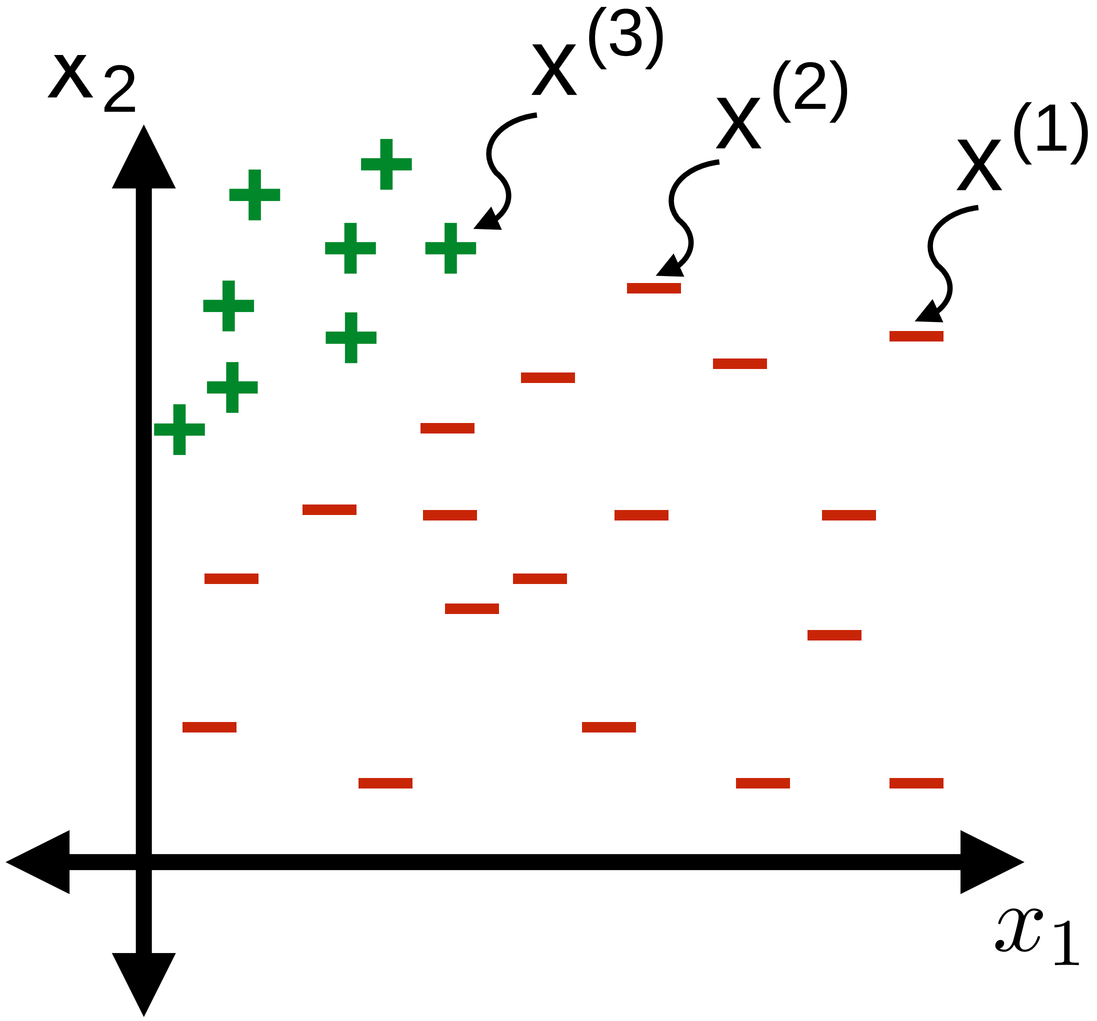
- Example *h*: For any *x*, *h*(*x*) = +1
  - Is this a good hypothesis?

Linear classifiers



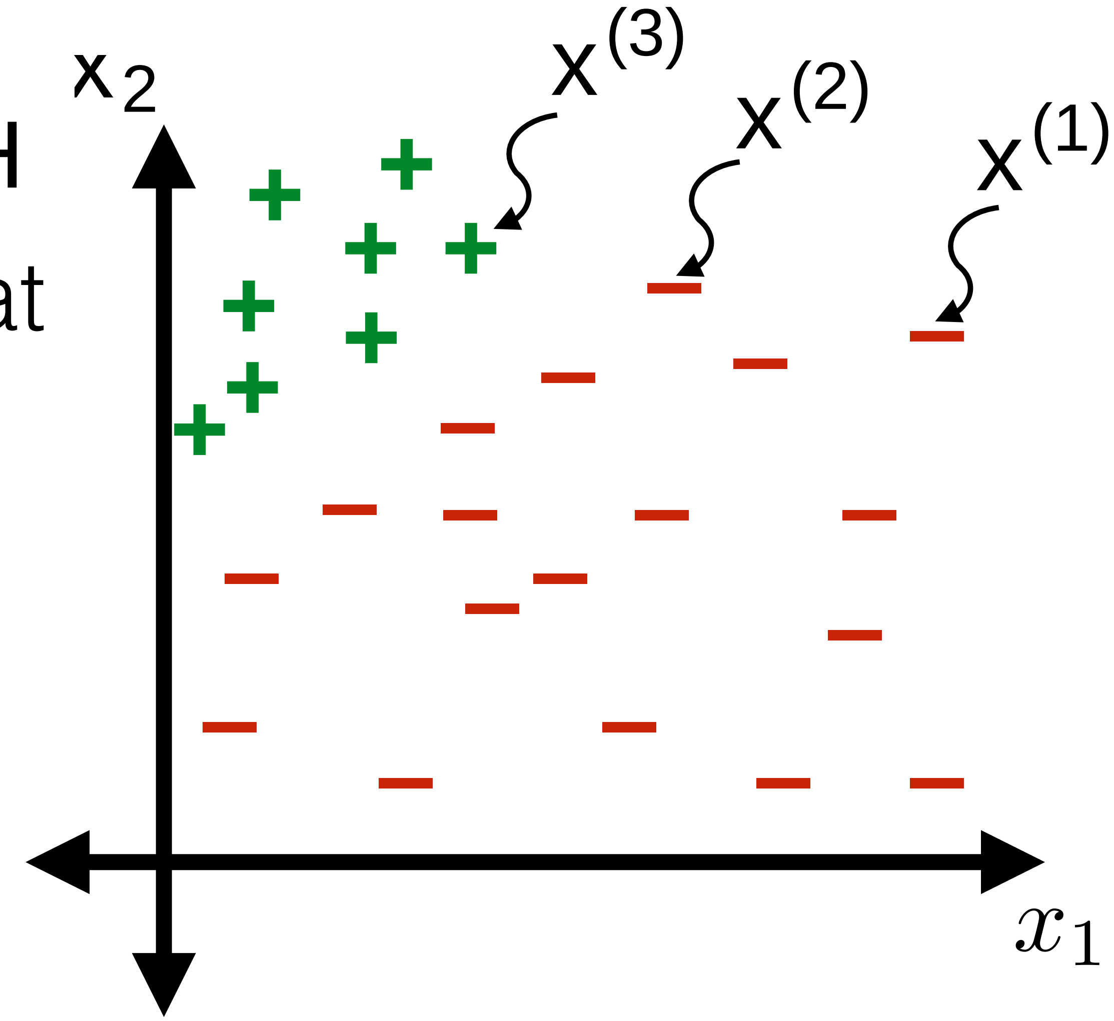
Linear classifiers

- Hypothesis class $\mathcal{H}$: set of **h** ! H



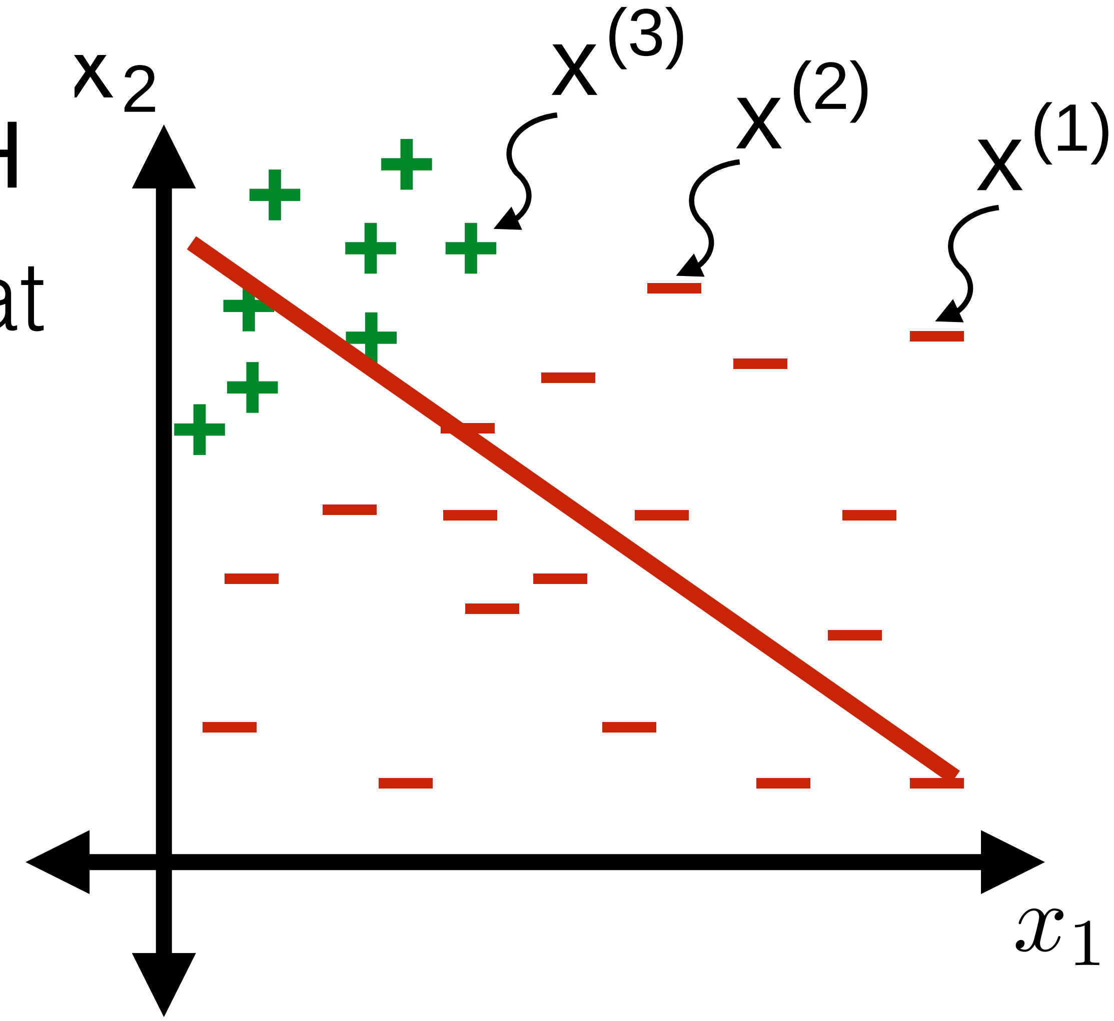
Linear classifiers

- Hypothesis class $\mathcal{H}$: set of **h** ! **H**
  - Example $\mathcal{H}$: All hypotheses that label +1 on one side of a line and -1 on the other side



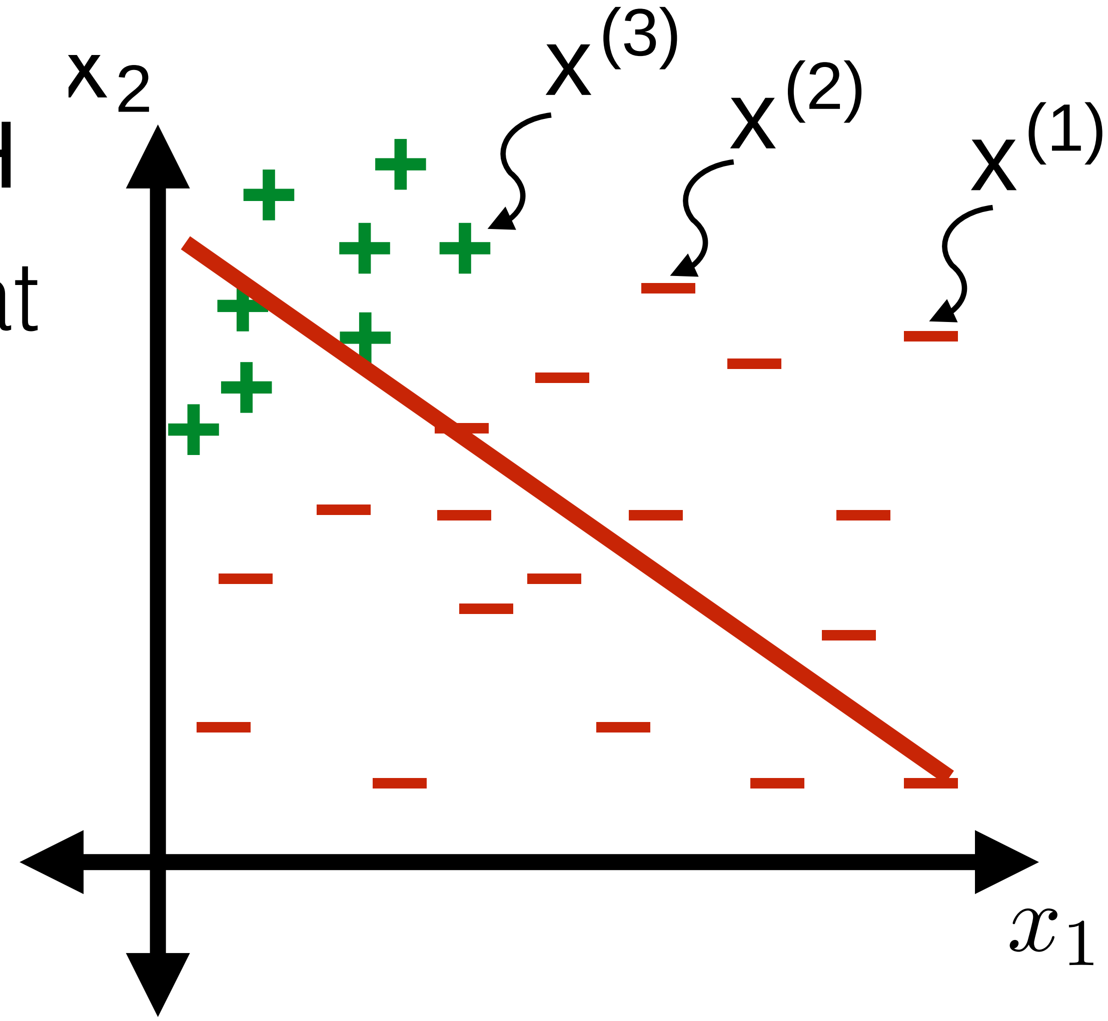
Linear classifiers

- Hypothesis class $\mathcal{H}$: set of **h** ! **H**
  - Example $\mathcal{H}$: All hypotheses that label +1 on one side of a line and -1 on the other side



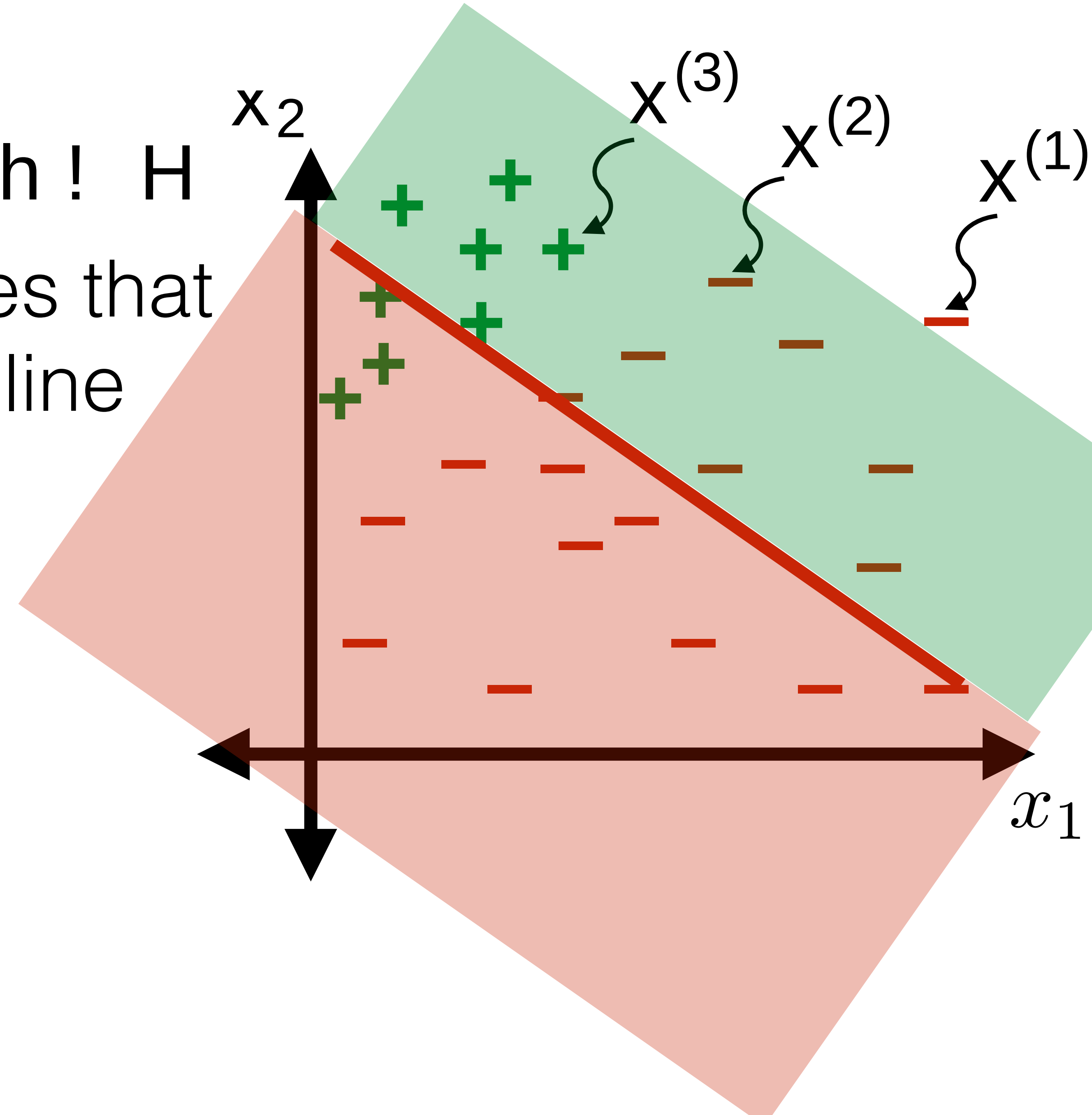
Linear classifiers

- Hypothesis class $\mathcal{H}$: set of **h** ! **H**
  - Example $\mathcal{H}$: All hypotheses that label +1 on one side of a line and -1 on the other side



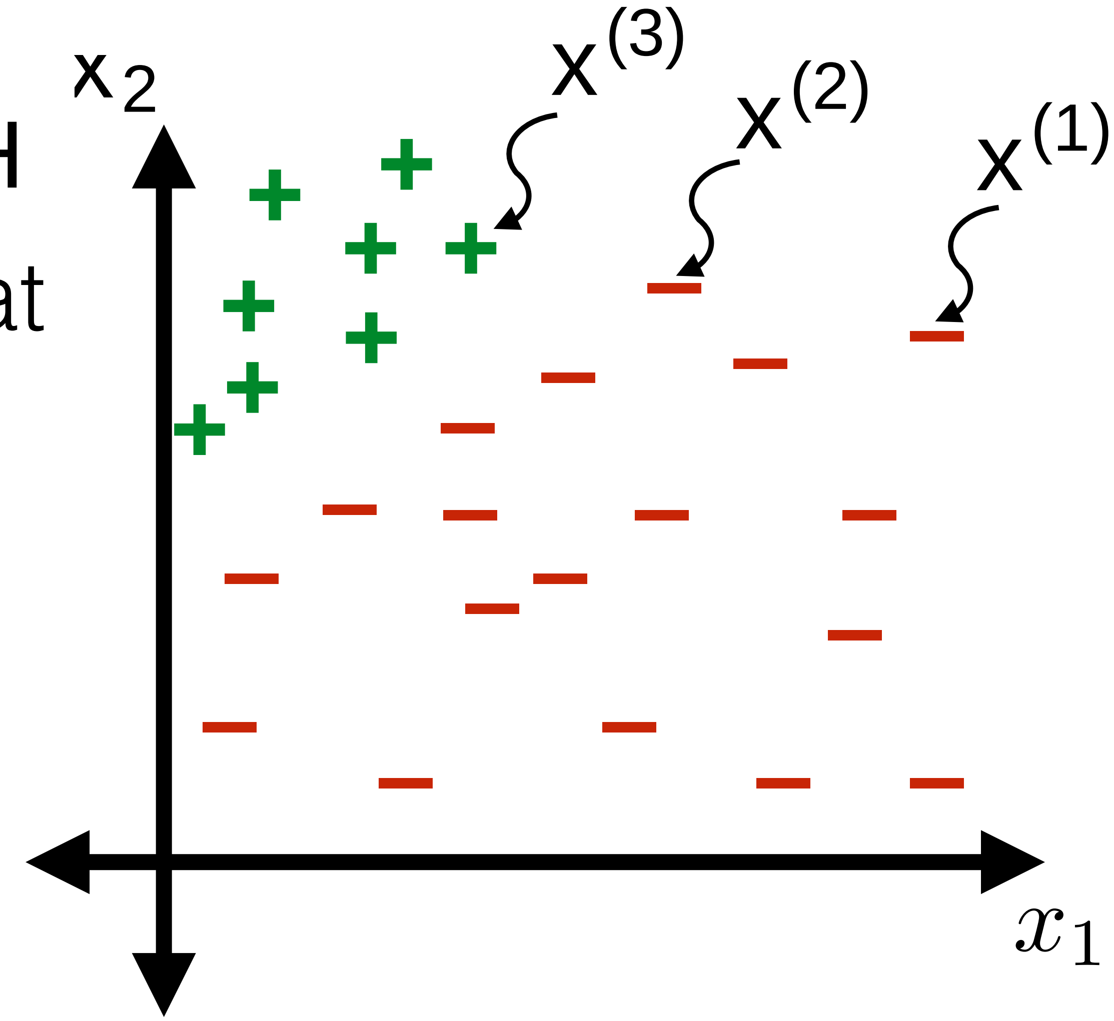
Linear classifiers

- Hypothesis class $\mathcal{H}$: set of **h** ! **H**
  - Example $\mathcal{H}$: All hypotheses that label +1 on one side of a line and -1 on the other side



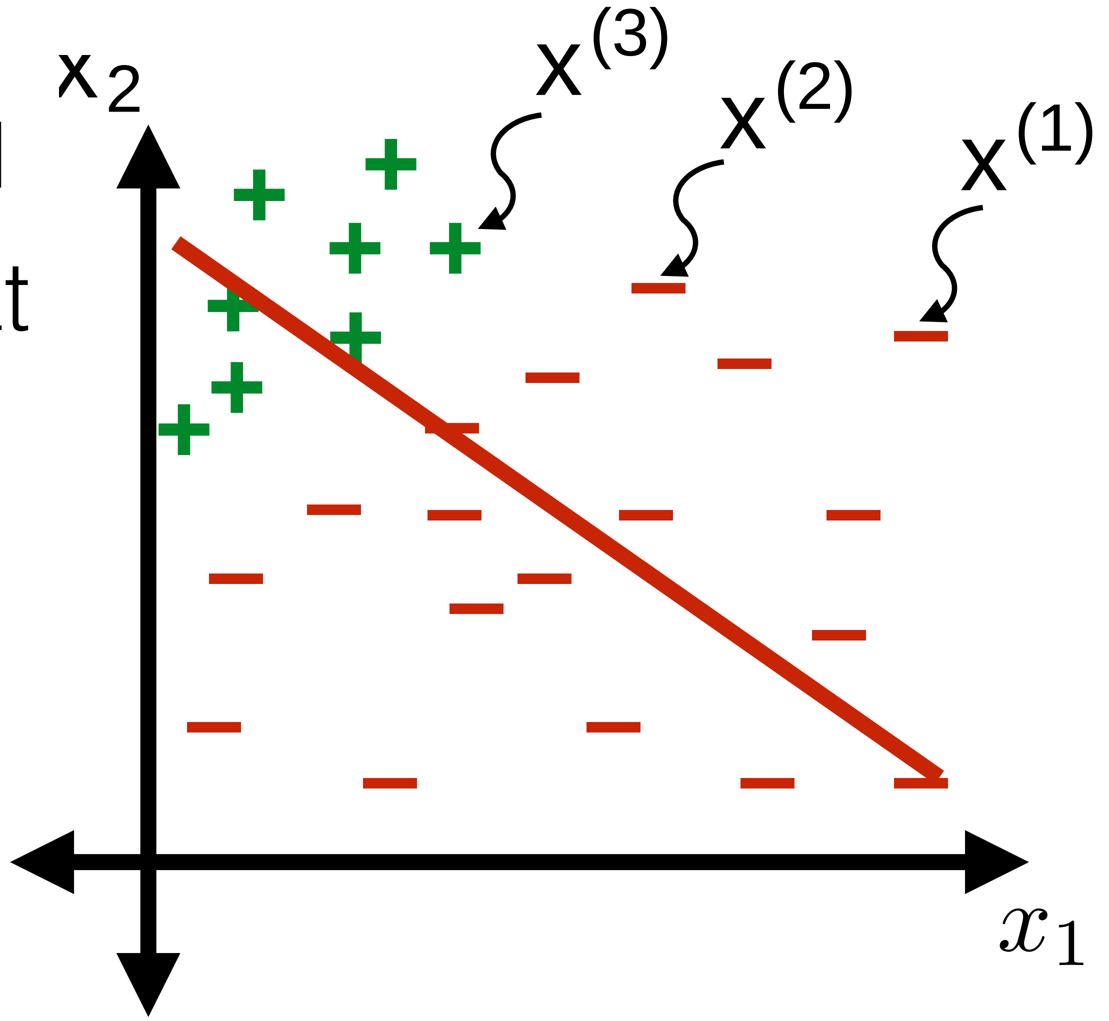
Linear classifiers

- Hypothesis class $\mathcal{H}$: set of h ! H
  - Example $\mathcal{H}$: All hypotheses that label +1 on one side of a line and -1 on the other side



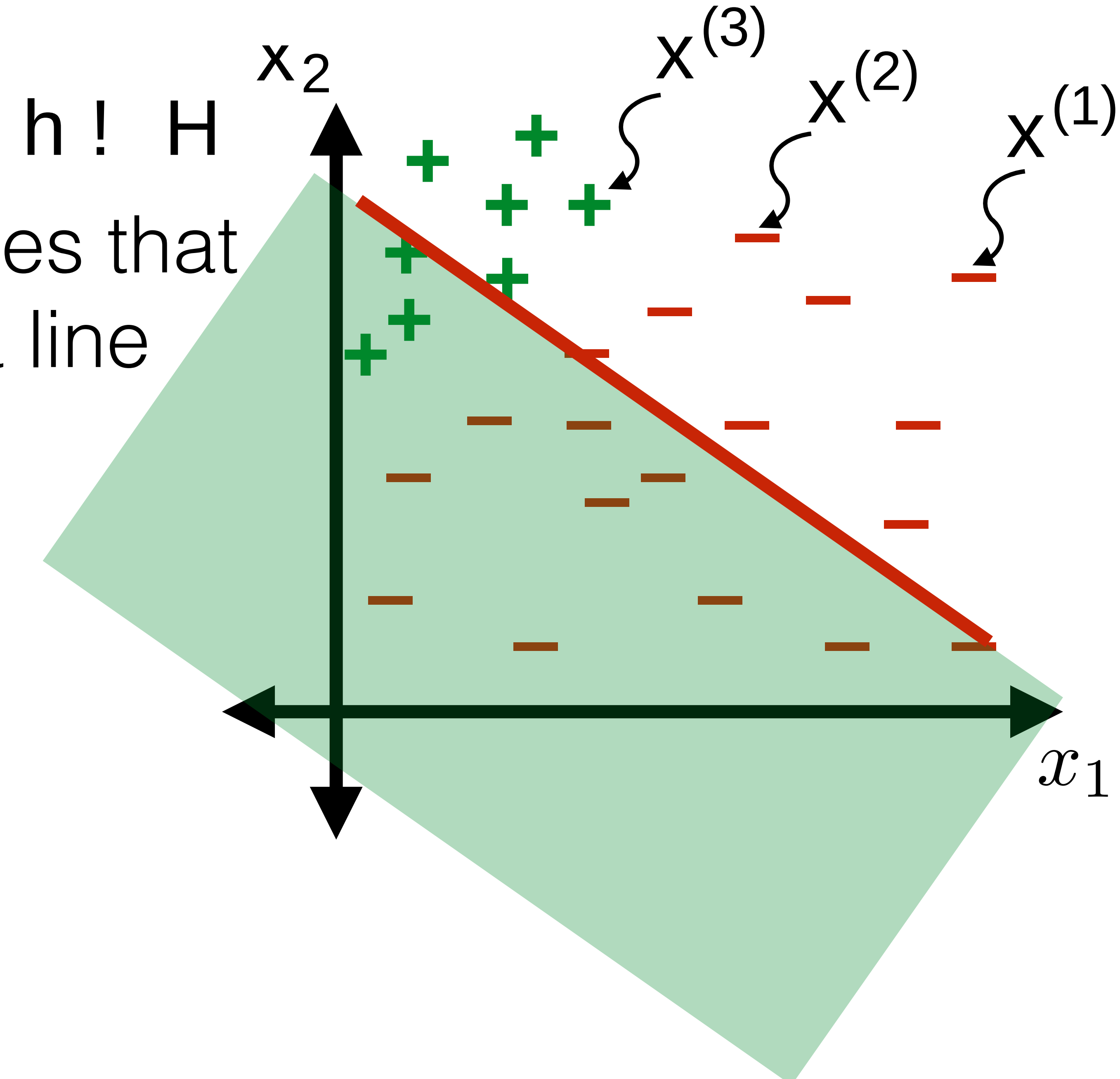
Linear classifiers

- Hypothesis class $\mathcal{H}$: set of **h** ! **H**
  - Example $\mathcal{H}$: All hypotheses that label +1 on one side of a line and -1 on the other side



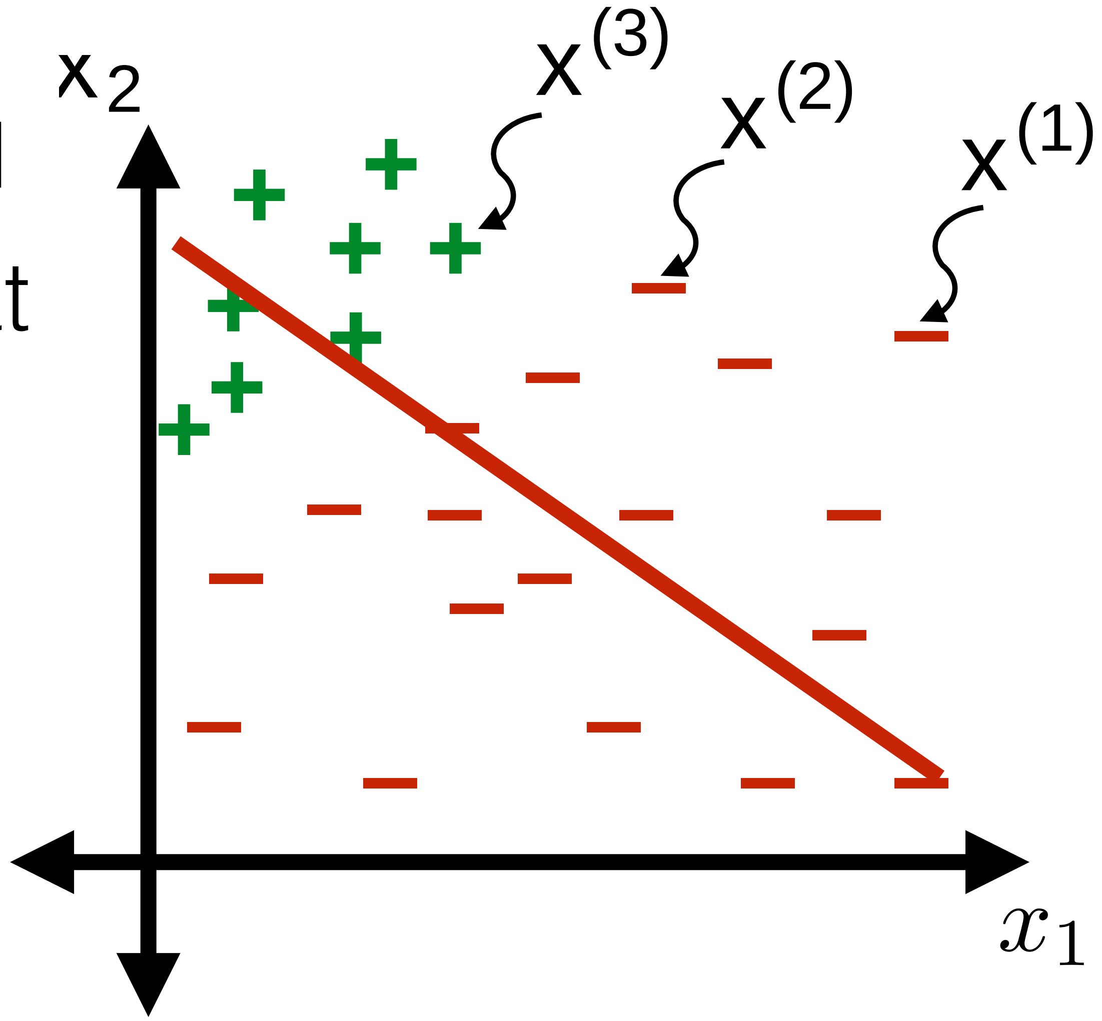
Linear classifiers

- Hypothesis class $\mathcal{H}$: set of h ! H
  - Example $\mathcal{H}$: All hypotheses that label +1 on one side of a line and -1 on the other side



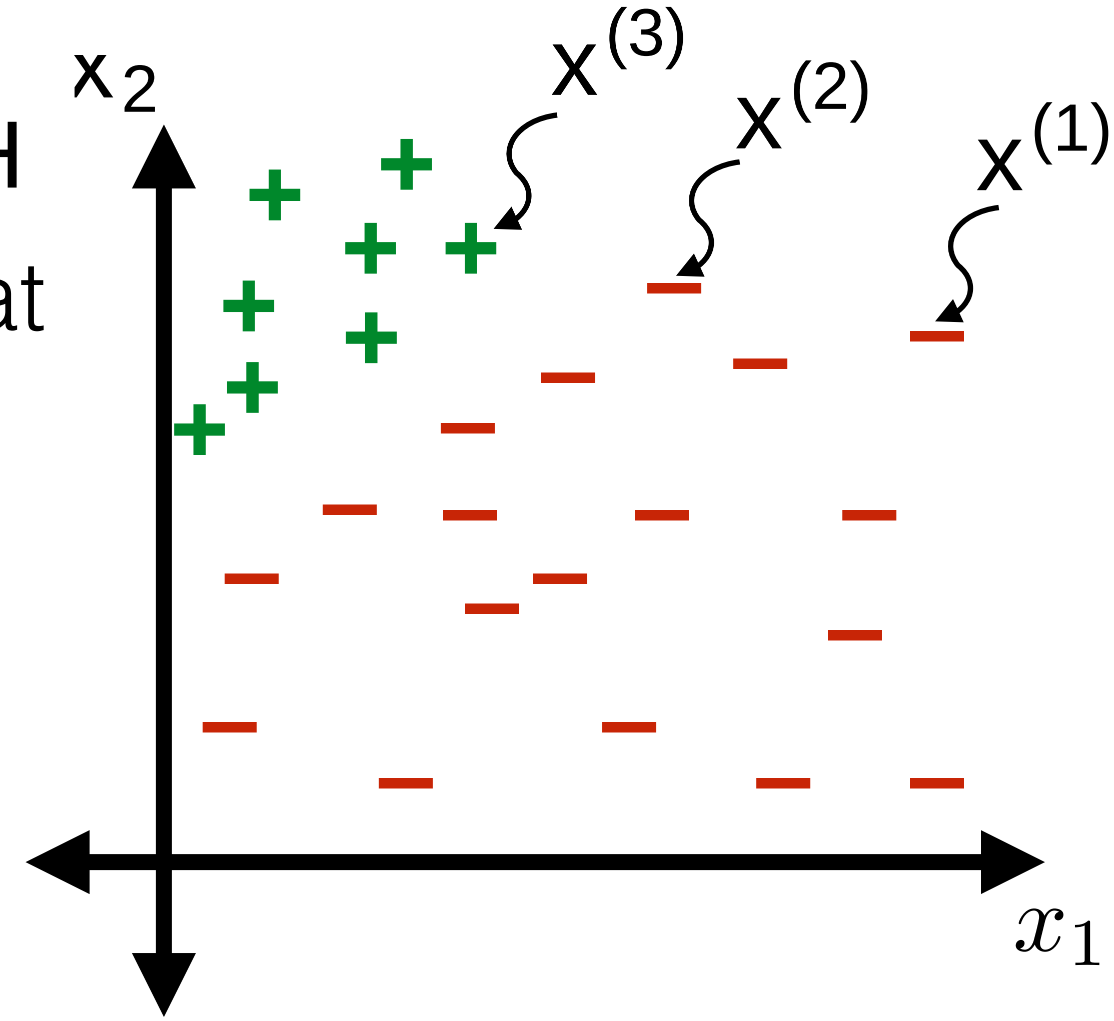
Linear classifiers

- Hypothesis class $\mathcal{H}$: set of **h** ! **H**
  - Example $\mathcal{H}$: All hypotheses that label +1 on one side of a line and -1 on the other side



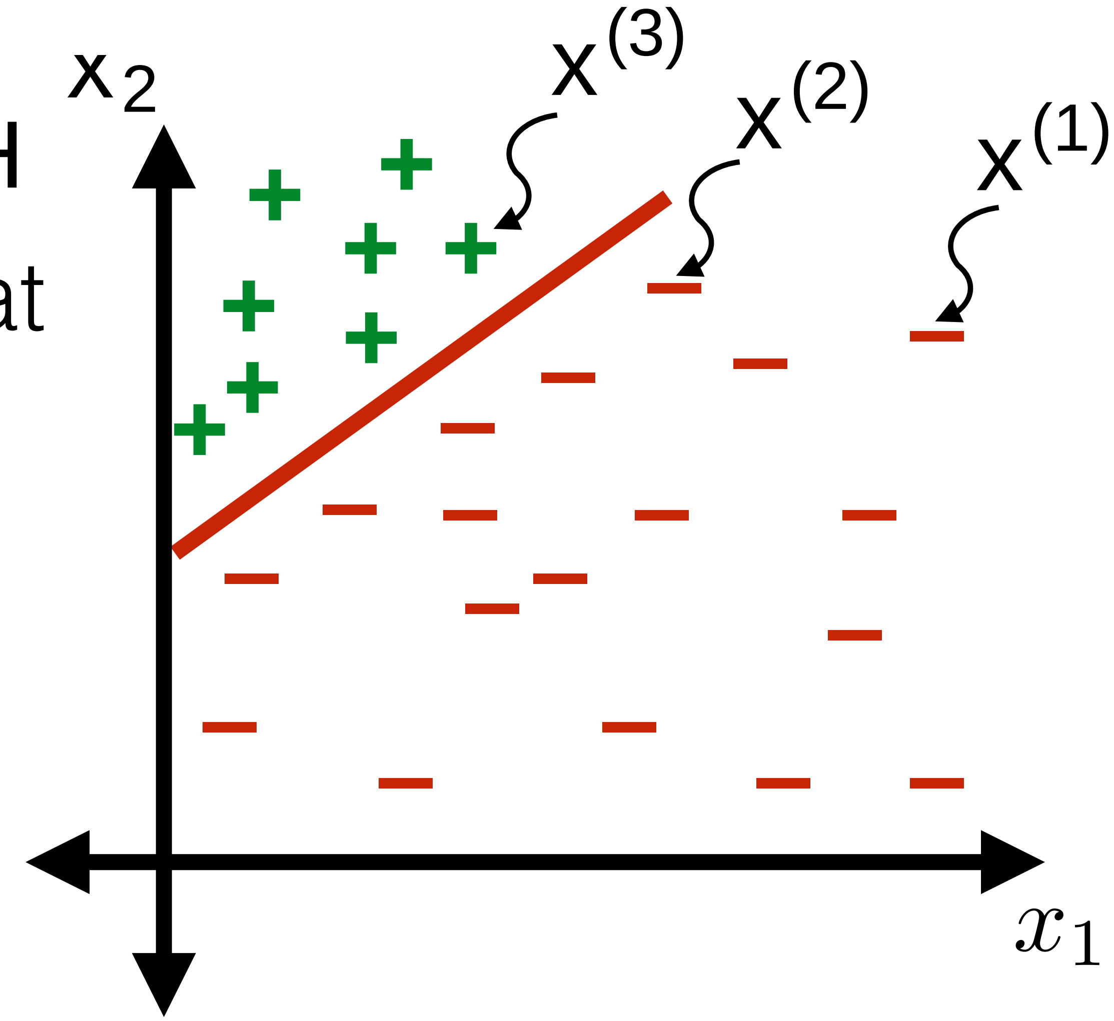
Linear classifiers

- Hypothesis class $\mathcal{H}$: set of **h** ! **H**
  - Example $\mathcal{H}$: All hypotheses that label +1 on one side of a line and -1 on the other side



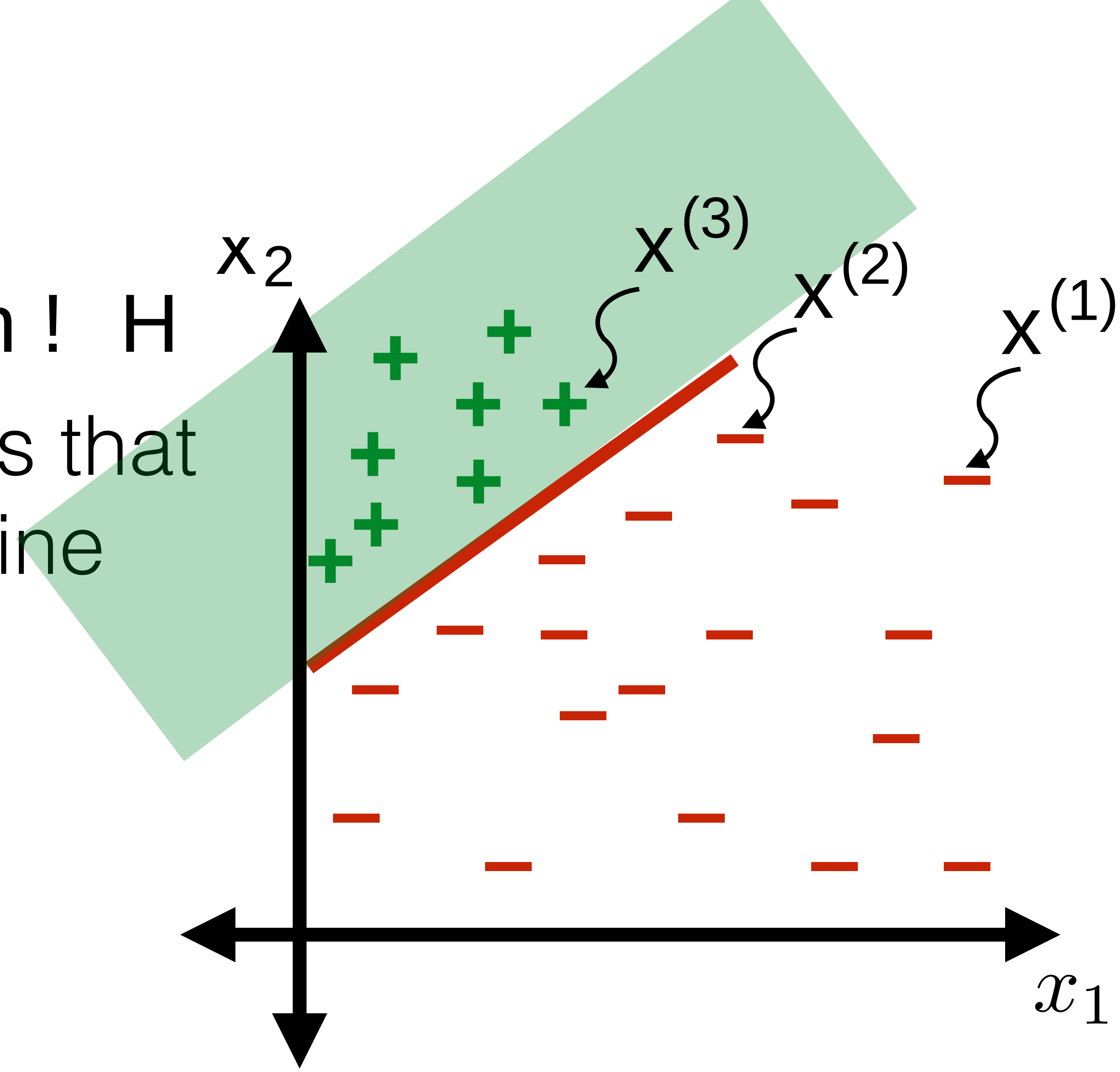
Linear classifiers

- Hypothesis class $\mathcal{H}$: set of **h** ! **H**
  - Example $\mathcal{H}$: All hypotheses that label +1 on one side of a line and -1 on the other side



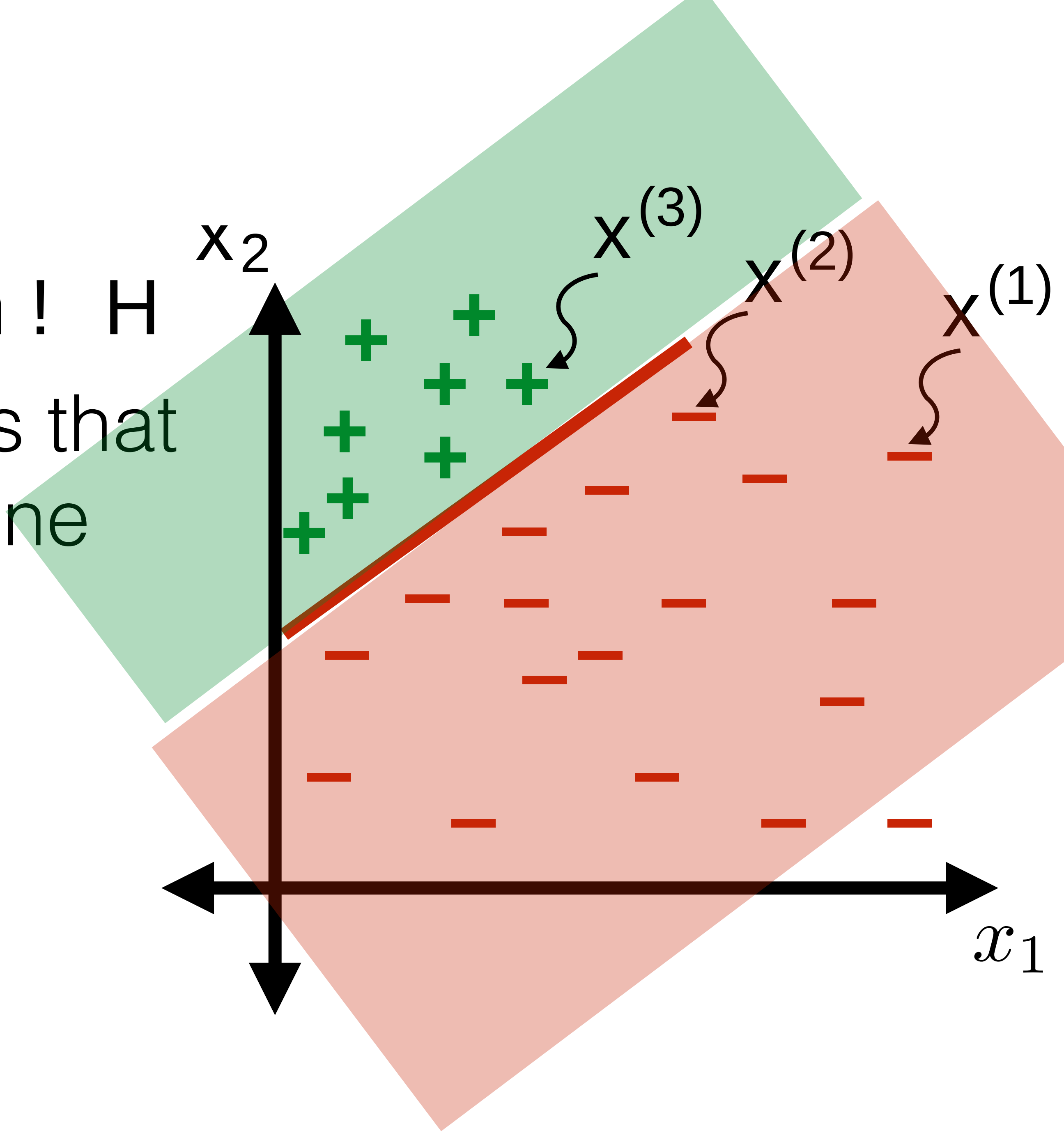
Linear classifiers

- Hypothesis class $\mathcal{H}$: set of **h** ! **H**
    - Example $\mathcal{H}$: All hypotheses that label +1 on one side of a line and -1 on the other side



Linear classifiers

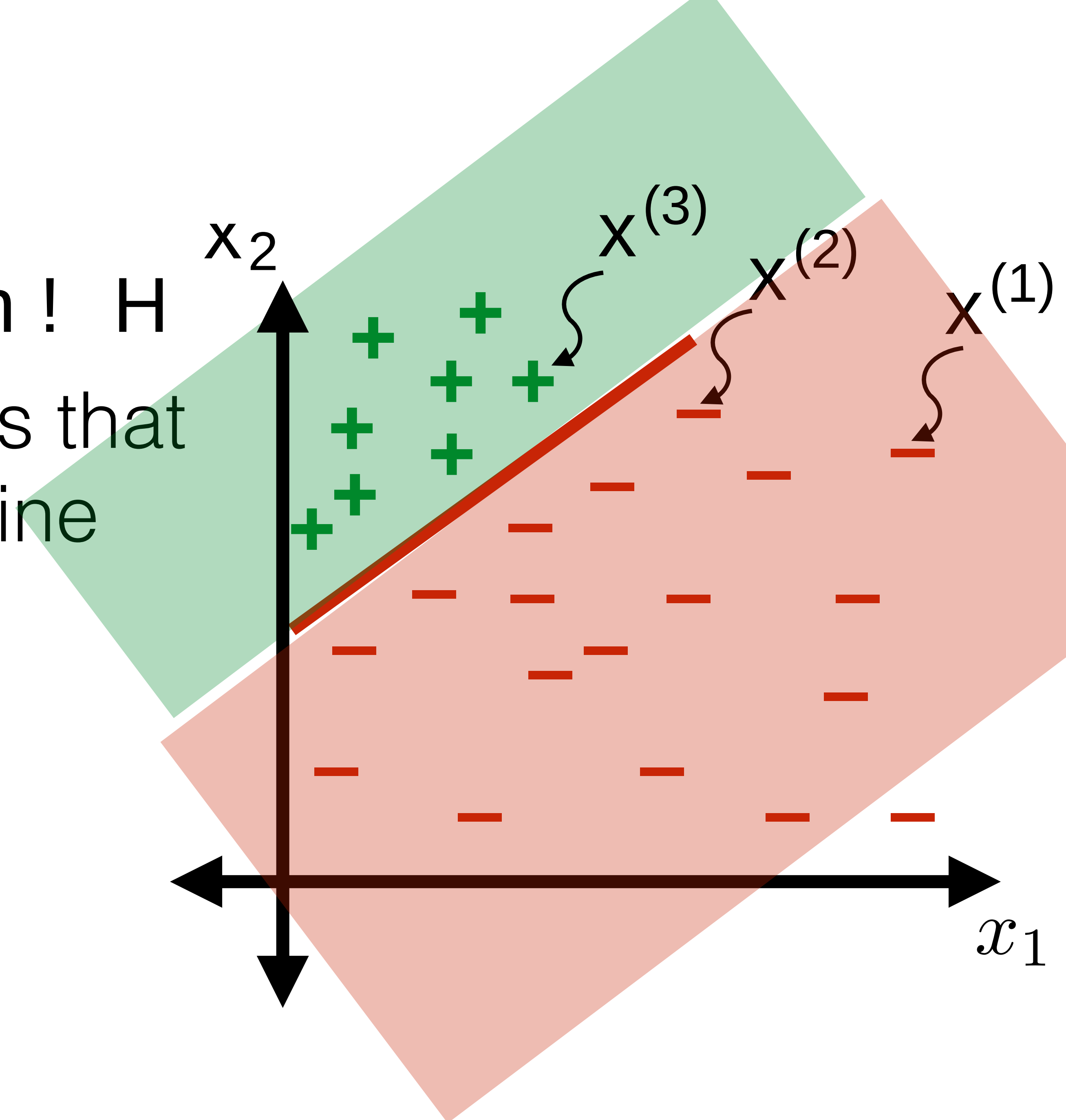
- Hypothesis class $\mathcal{H}$: set of **h ! H**
  - Example $\mathcal{H}$: All hypotheses that label +1 on one side of a line and -1 on the other side



Linear classifiers

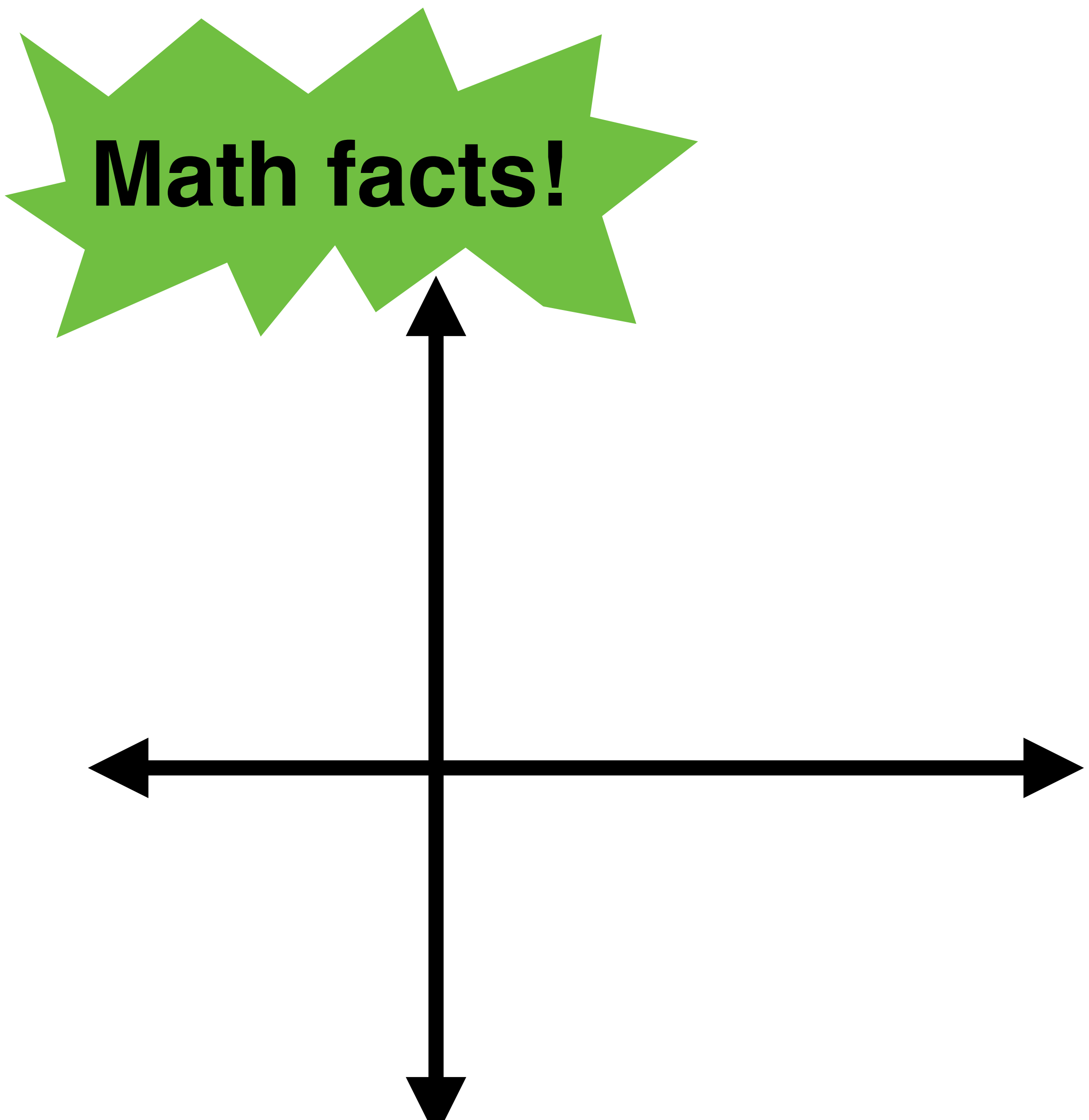
- Hypothesis class $\mathcal{H}$: set of **h** ! **H**
  - Example $\mathcal{H}$: All hypotheses that label +1 on one side of a line and -1 on the other side

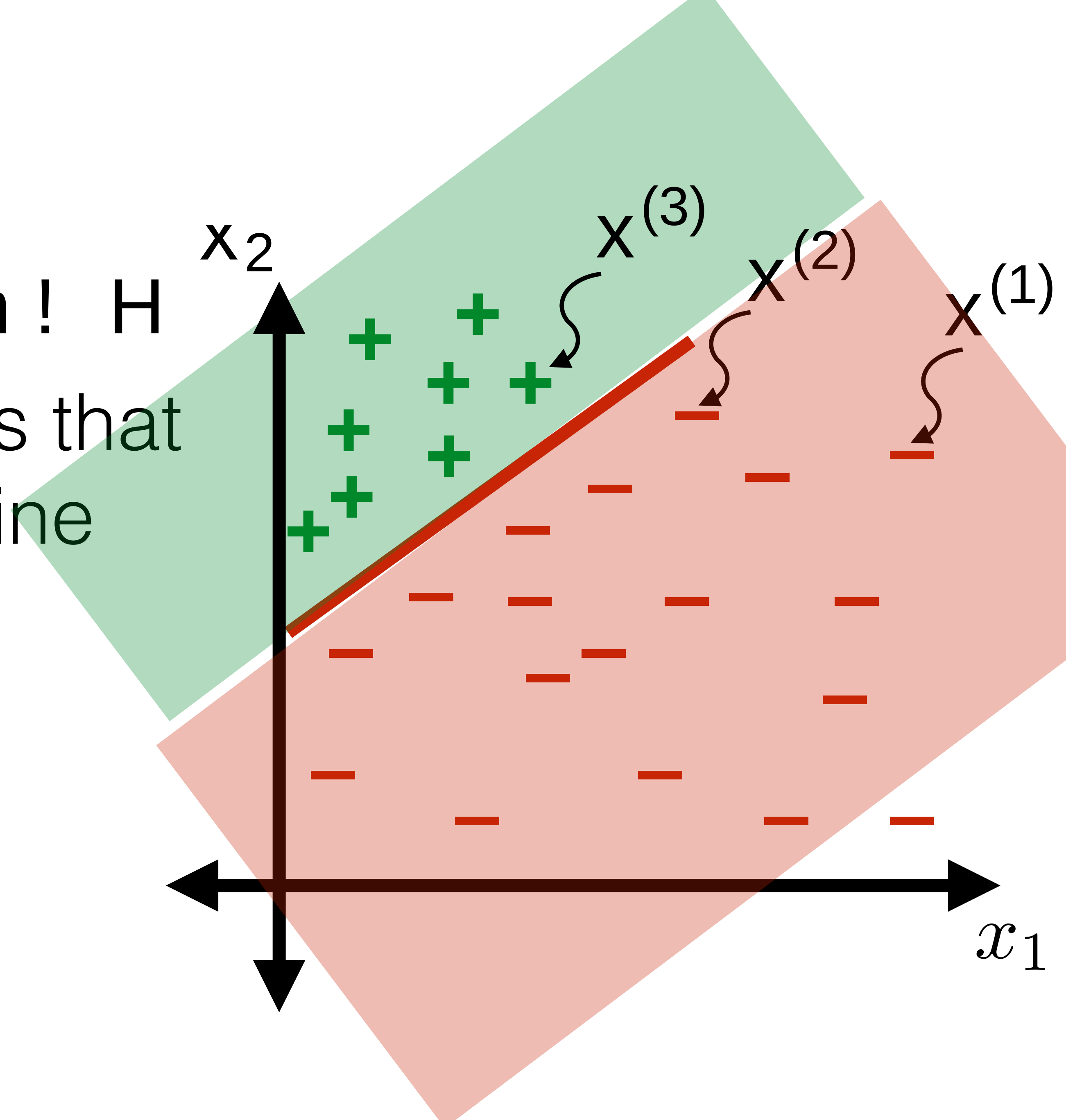
Math facts!



Linear classifiers

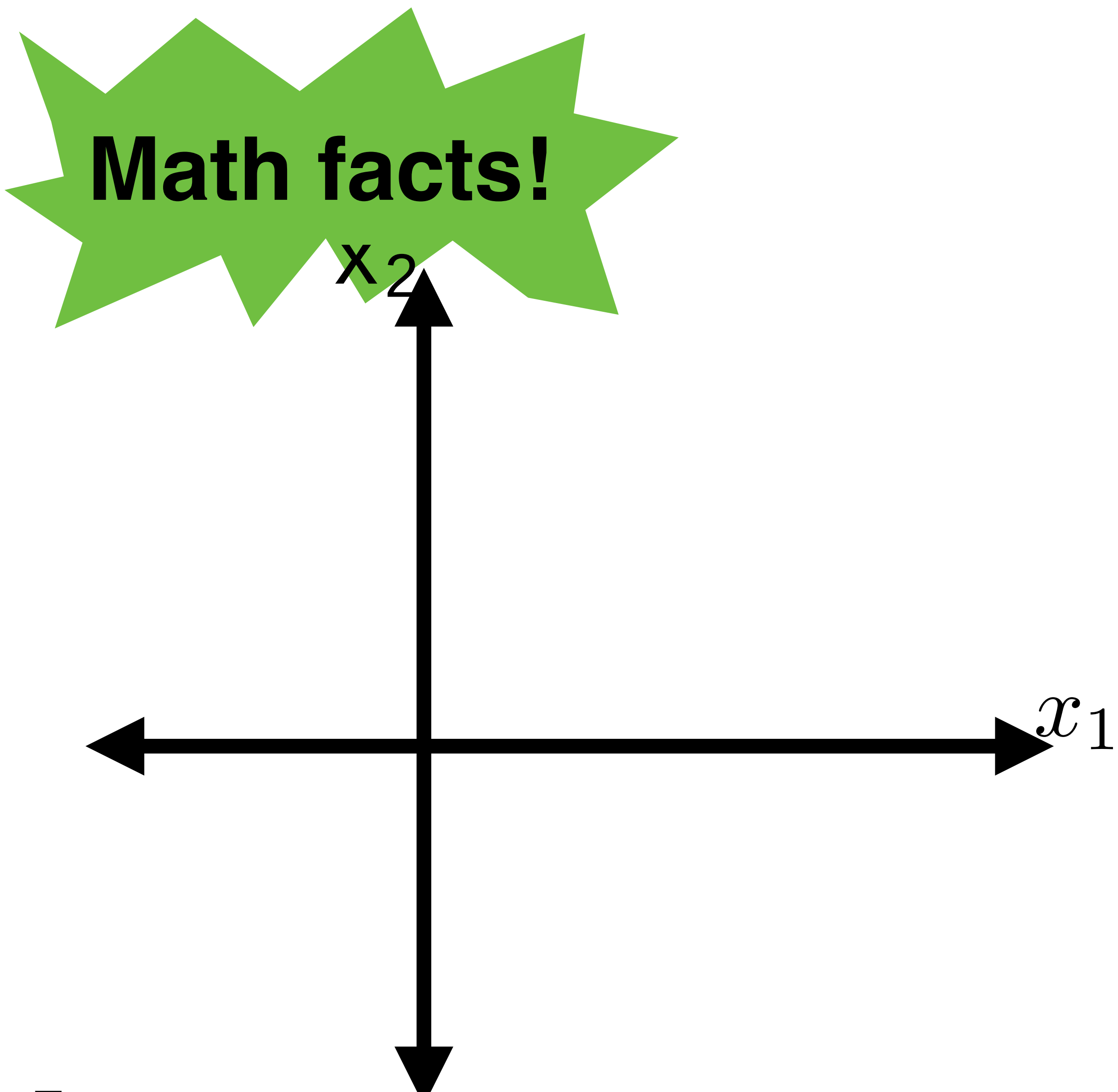
- Hypothesis class $\mathcal{H}$: set of **h** ! **H**
  - Example $\mathcal{H}$: All hypotheses that label +1 on one side of a line and -1 on the other side

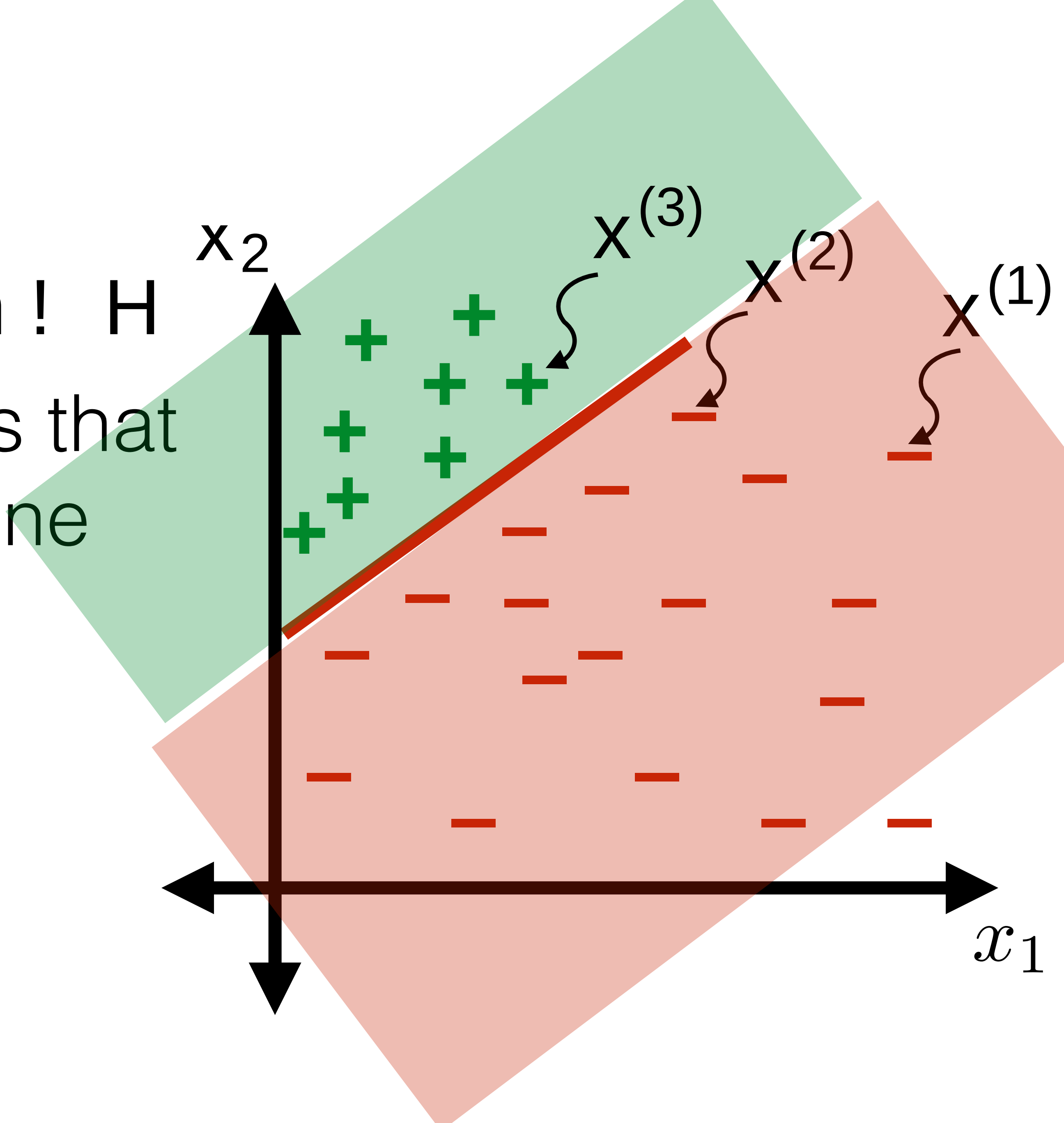




Linear classifiers

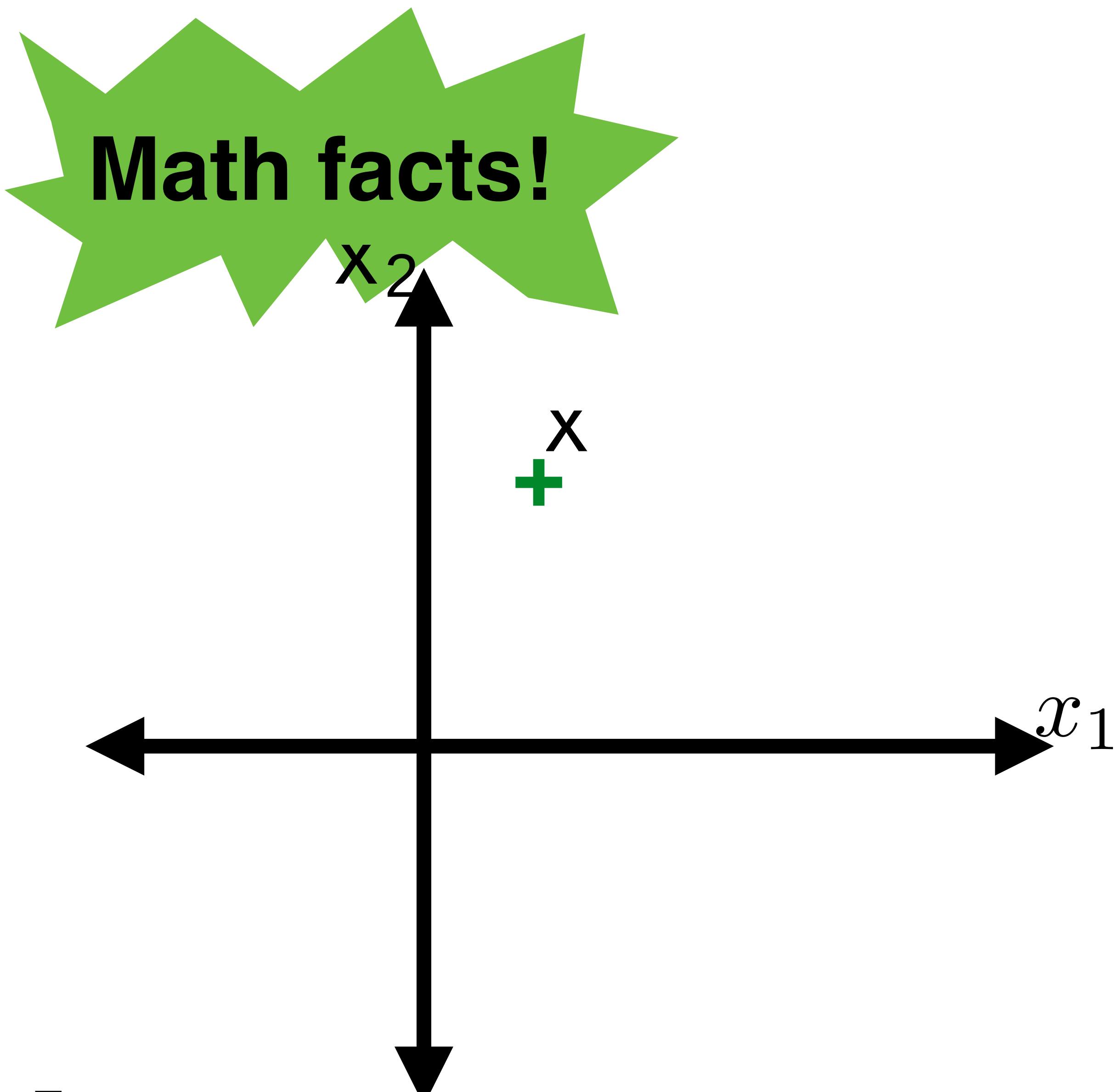
- Hypothesis class $\mathcal{H}$: set of **h** ! **H**
  - Example $\mathcal{H}$: All hypotheses that label +1 on one side of a line and -1 on the other side

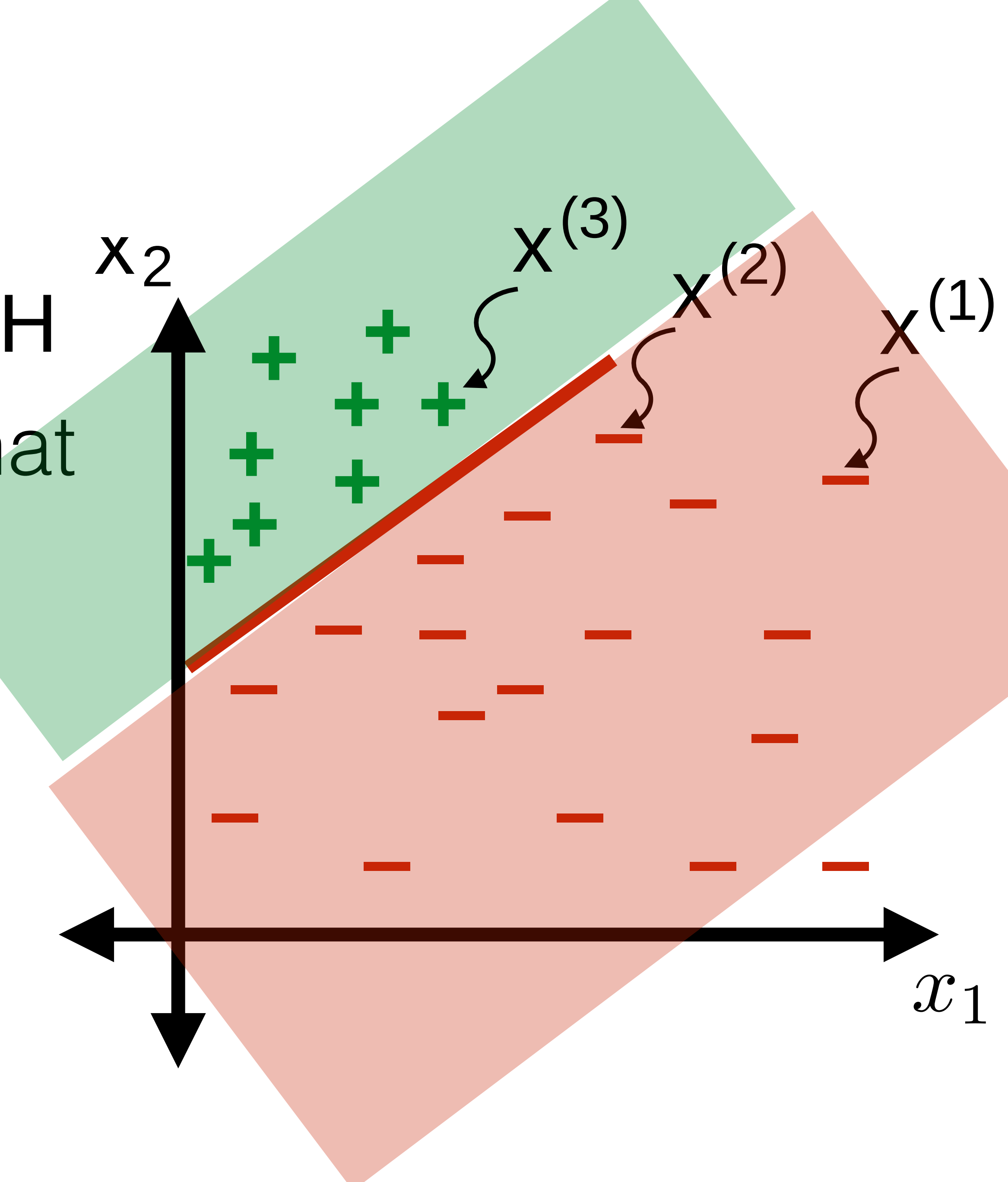




Linear classifiers

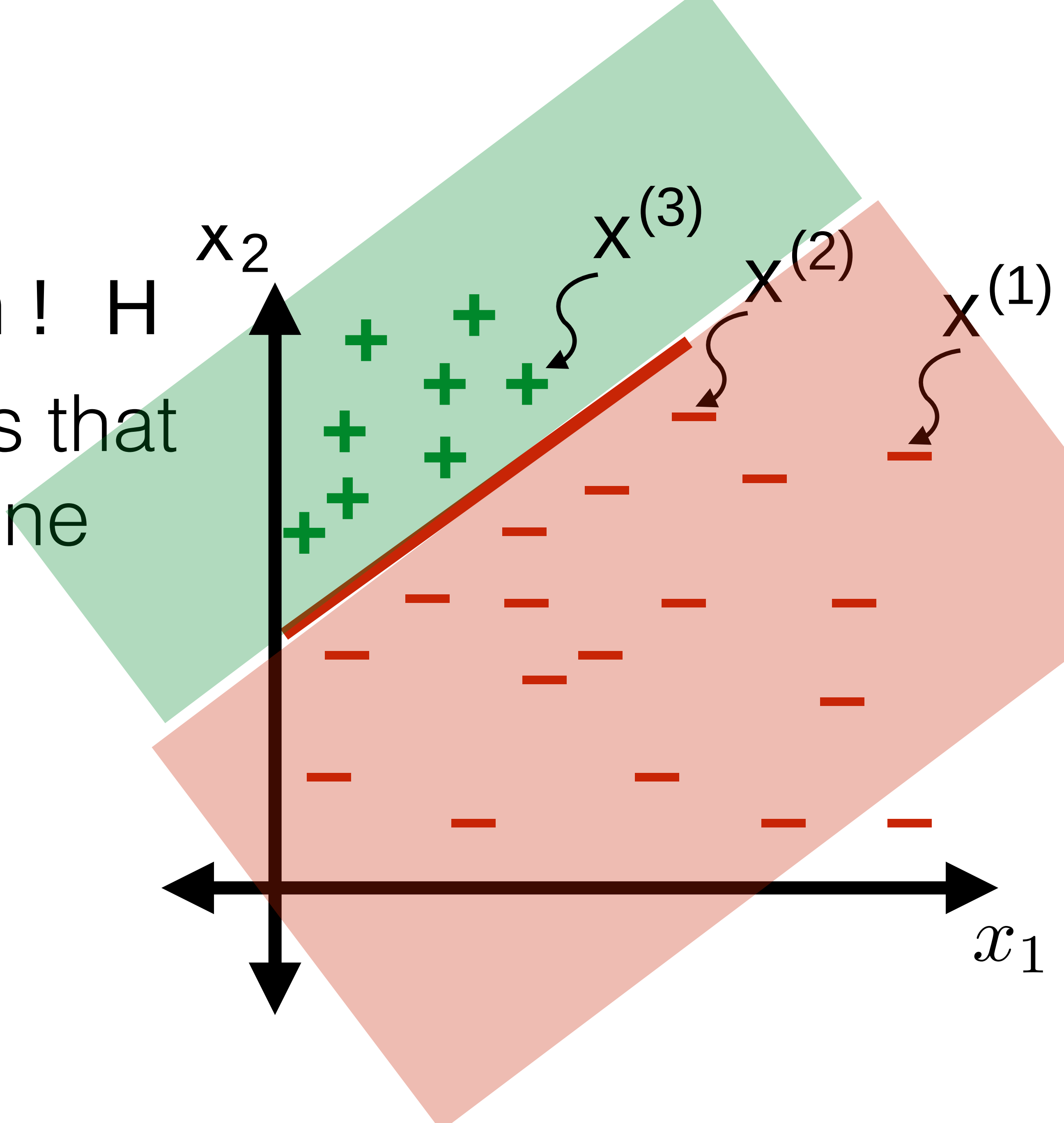
- Hypothesis class $\mathcal{H}$: set of **h** ! **H**
    - Example $\mathcal{H}$: All hypotheses that label +1 on one side of a line and -1 on the other side

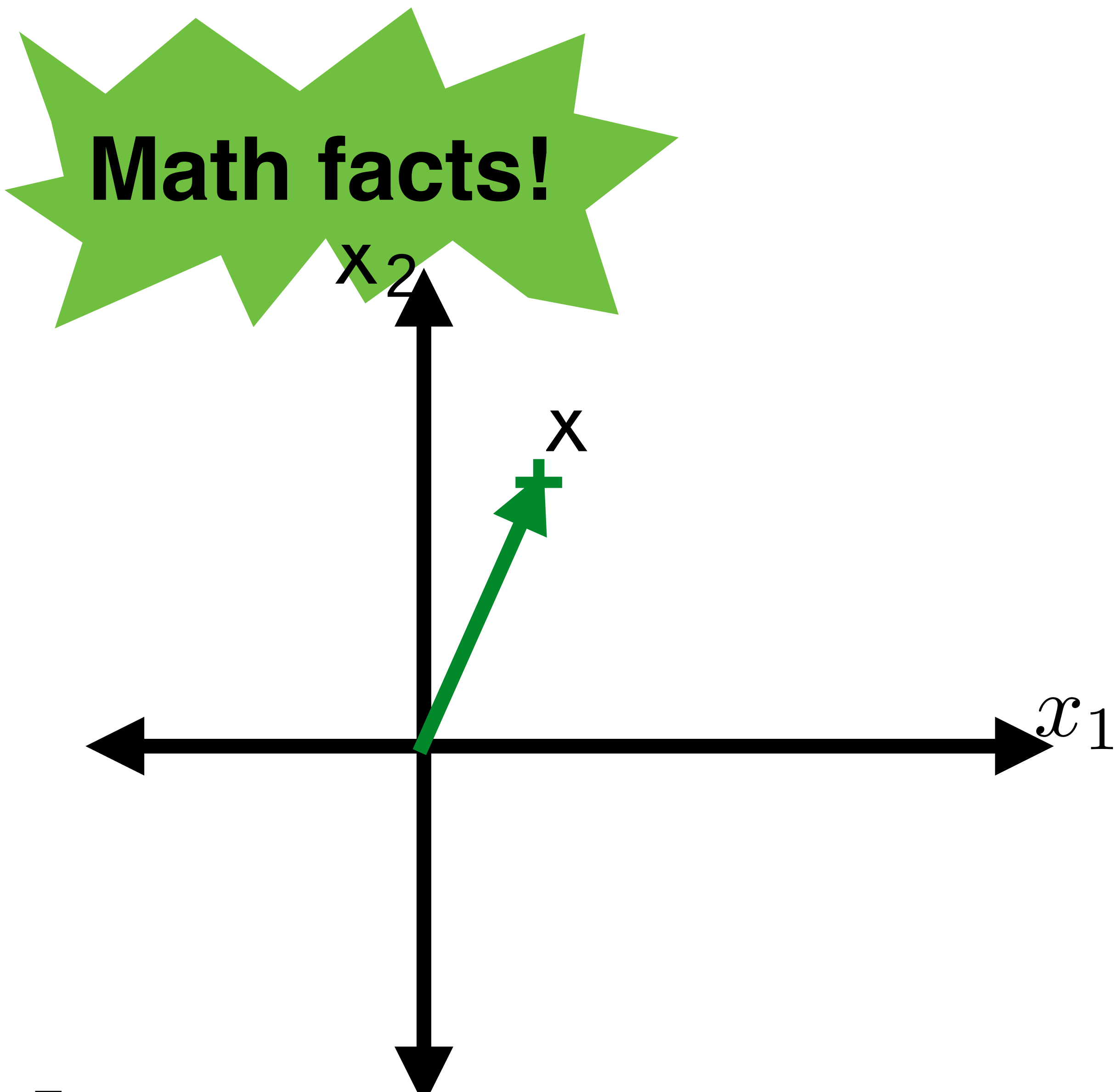




Linear classifiers

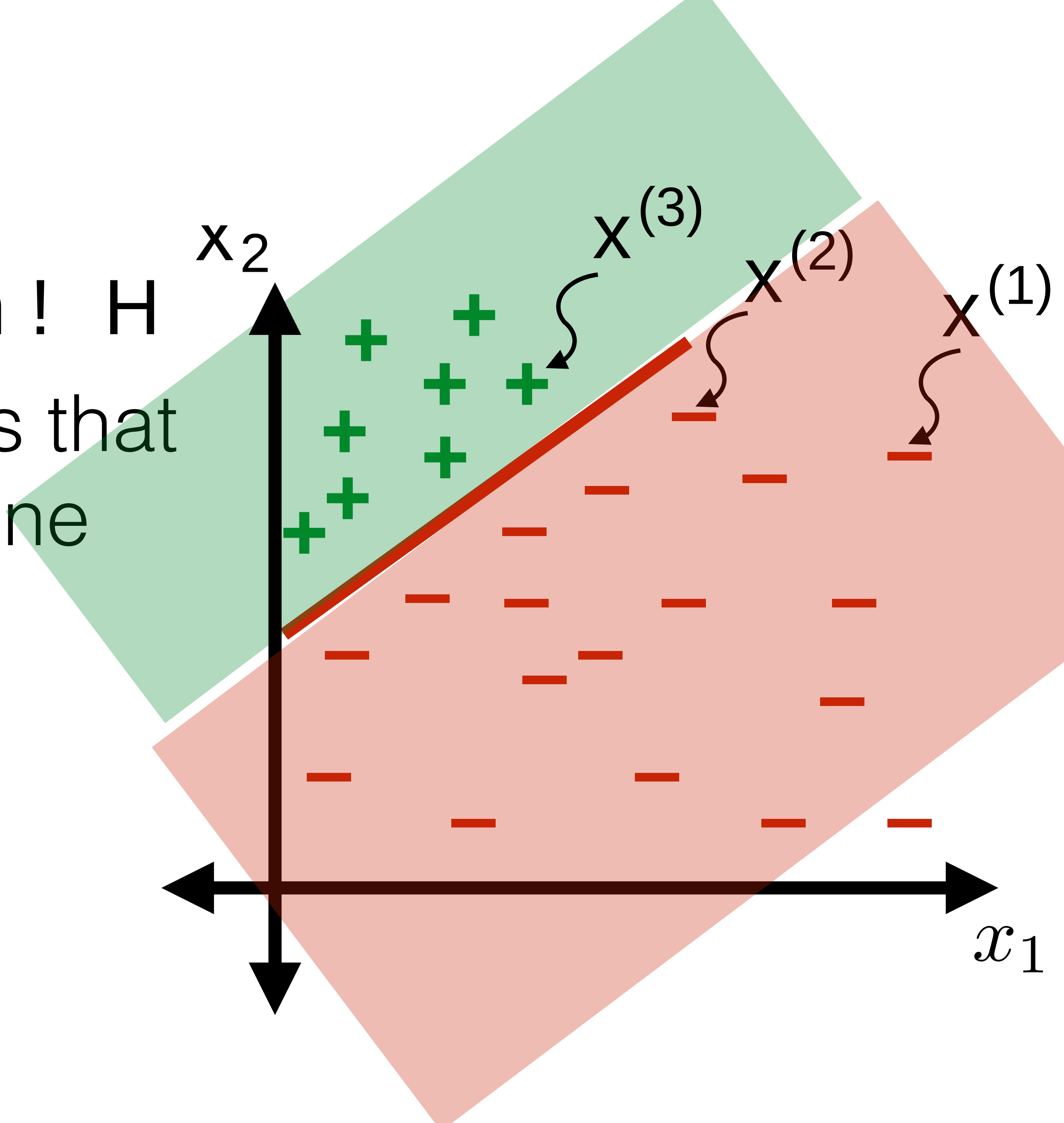
- Hypothesis class $\mathcal{H}$: set of **h** ! **H**
  - Example $\mathcal{H}$: All hypotheses that label +1 on one side of a line and -1 on the other side

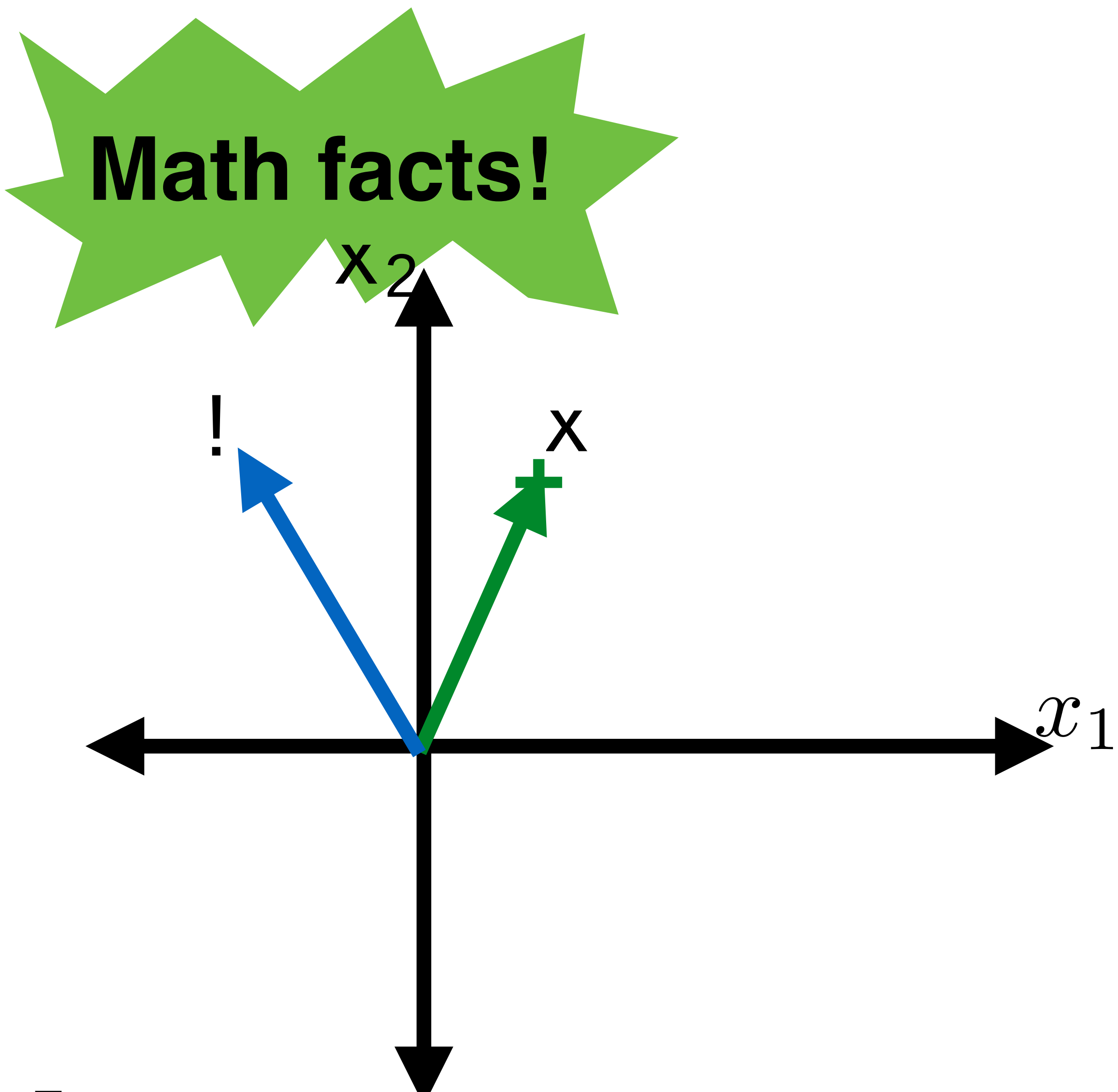




Linear classifiers

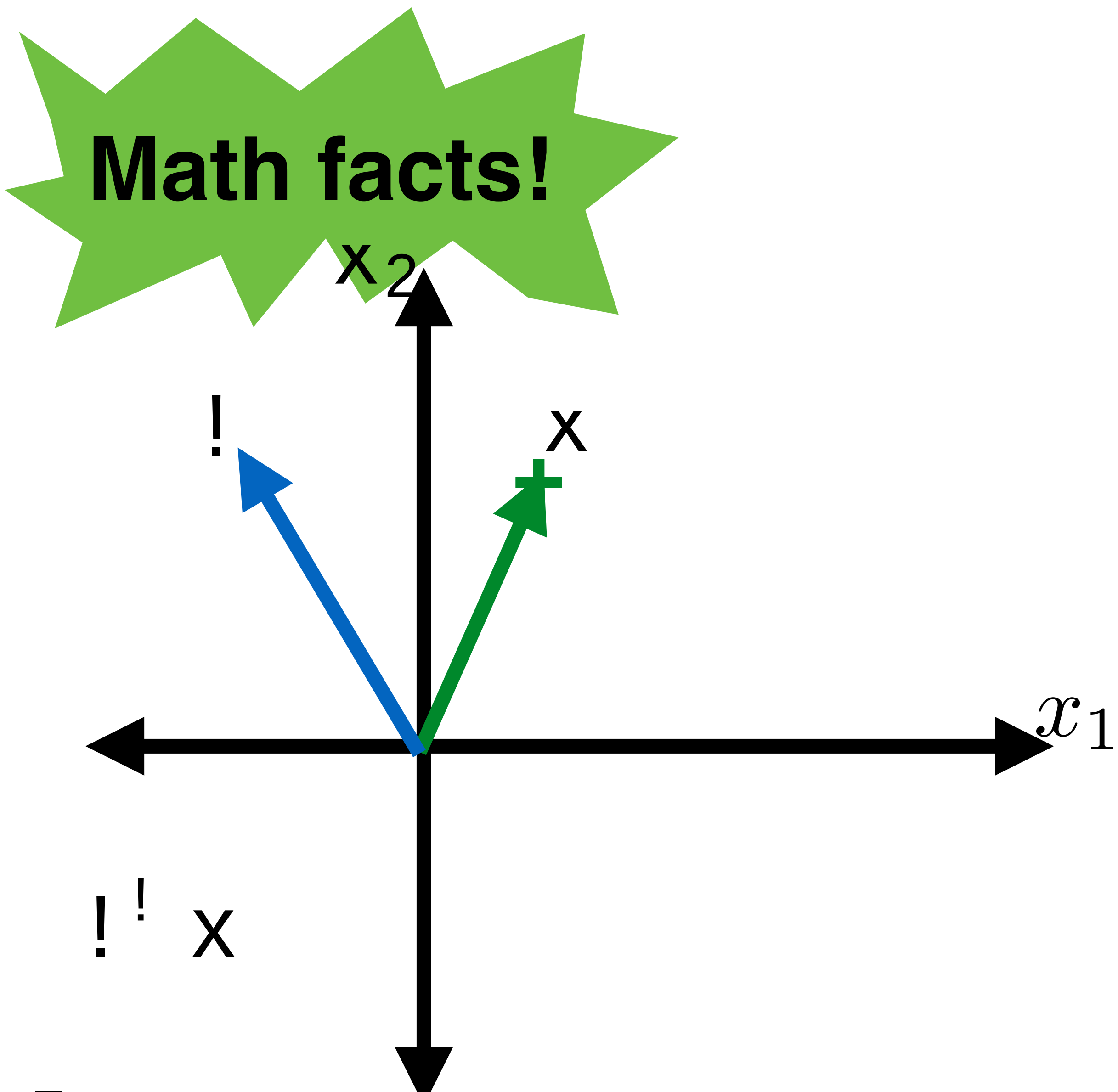
- Hypothesis class $\mathcal{H}$: set of **h** ! **H**
  - Example $\mathcal{H}$: All hypotheses that label +1 on one side of a line and -1 on the other side

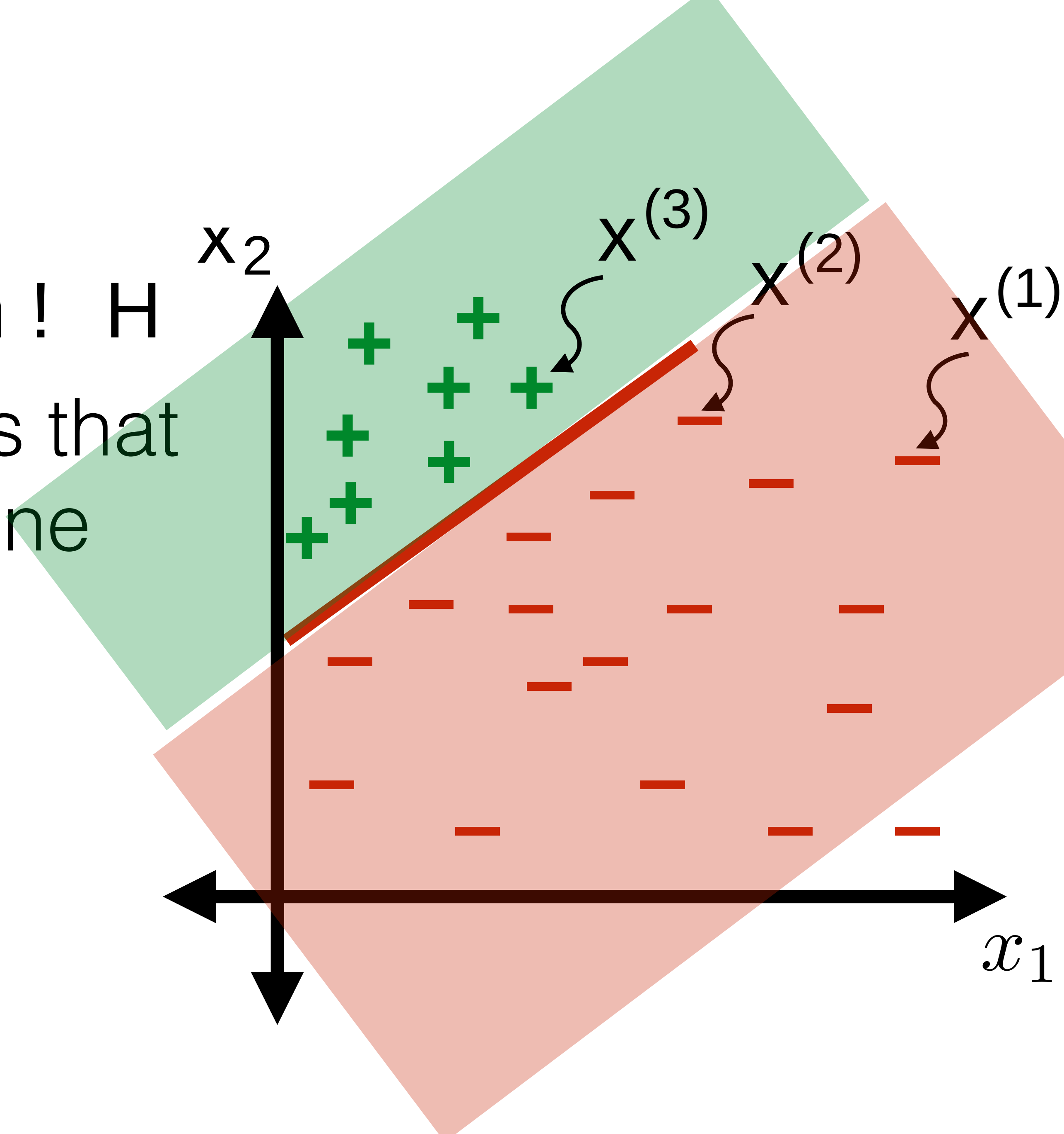




Linear classifiers

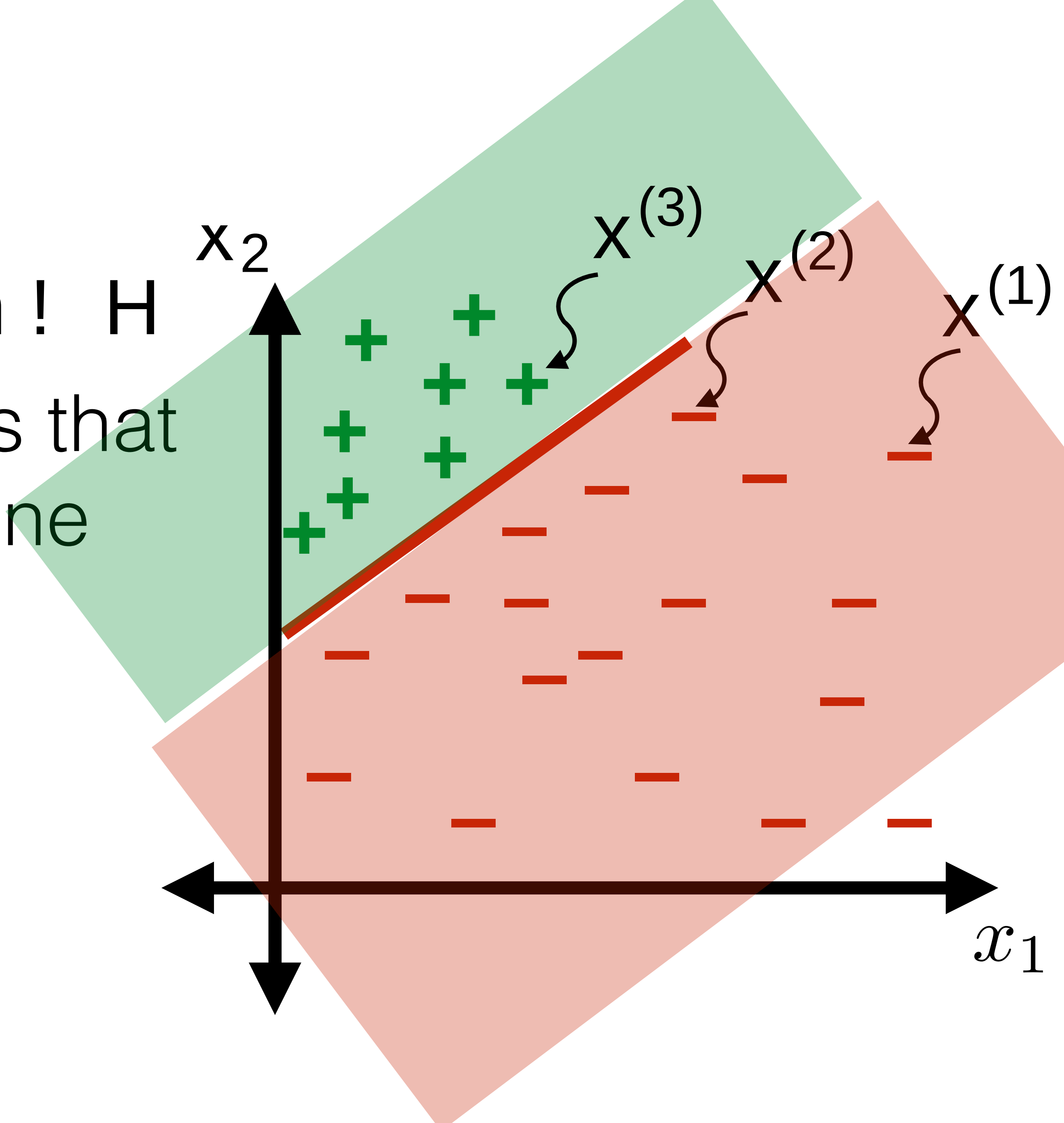
- Hypothesis class $\mathcal{H}$: set of **h** ! **H**
    - Example $\mathcal{H}$: All hypotheses that label +1 on one side of a line and -1 on the other side

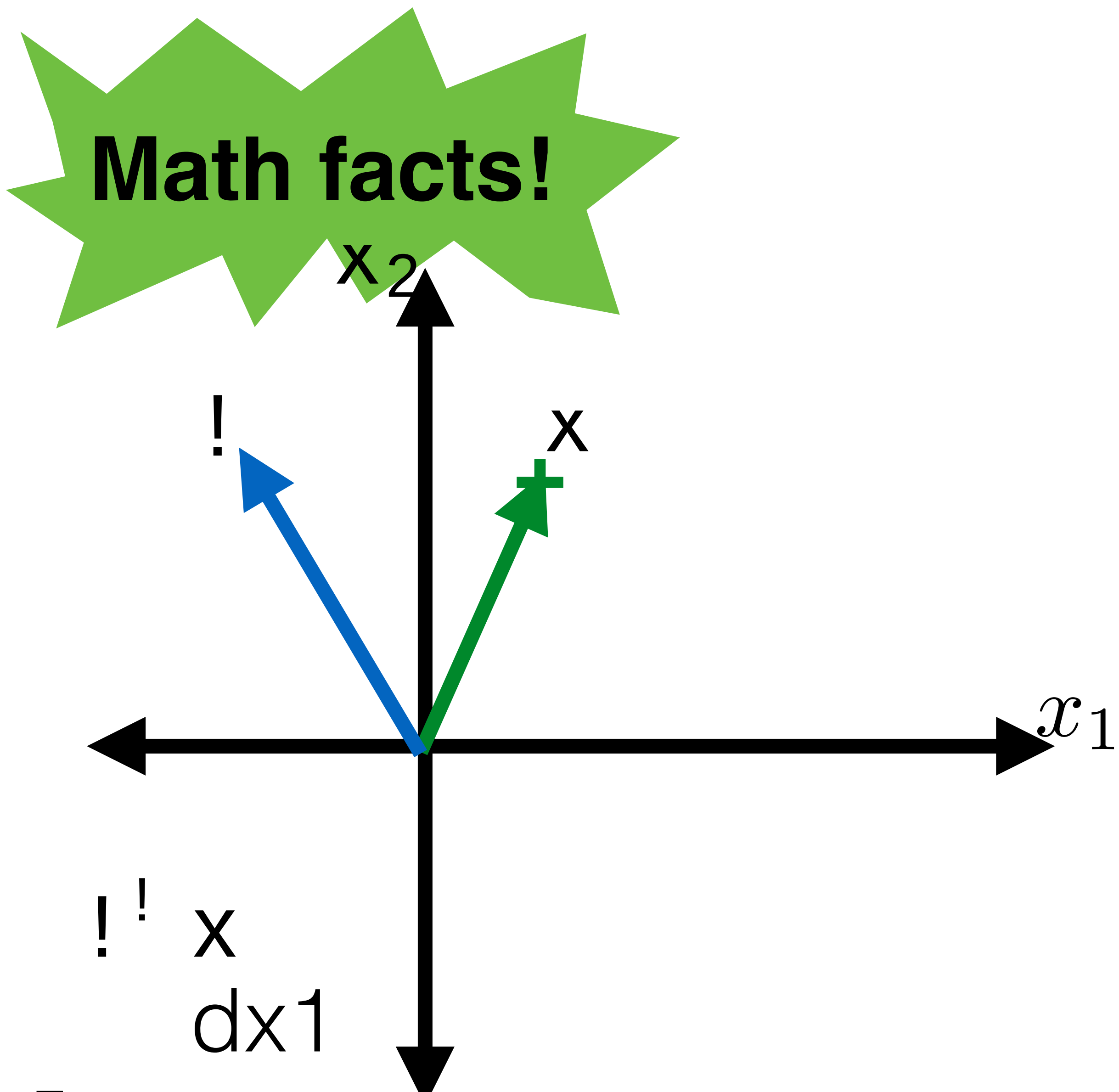




Linear classifiers

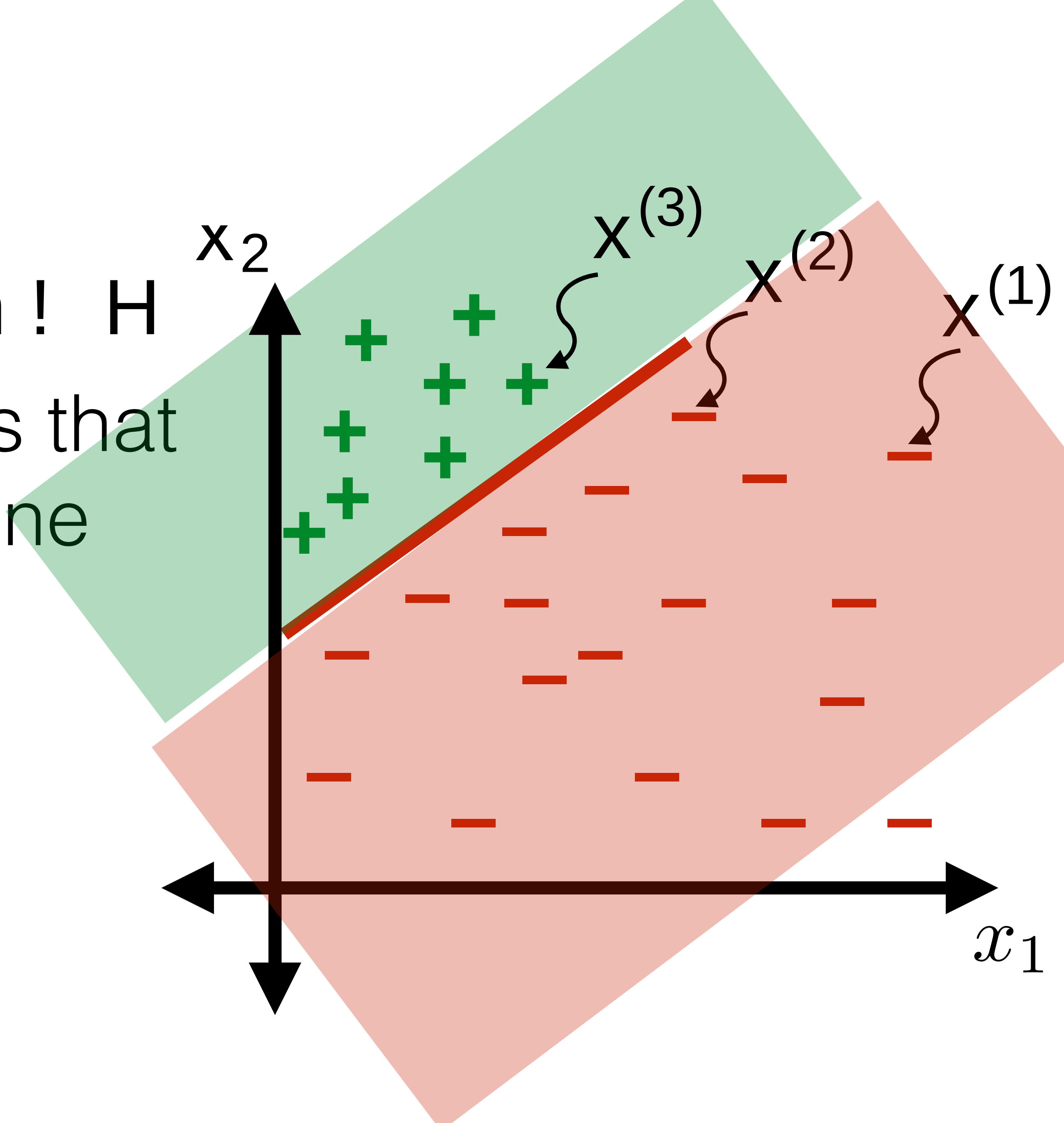
- Hypothesis class $\mathcal{H}$: set of **h** ! **H**
  - Example $\mathcal{H}$: All hypotheses that label +1 on one side of a line and -1 on the other side

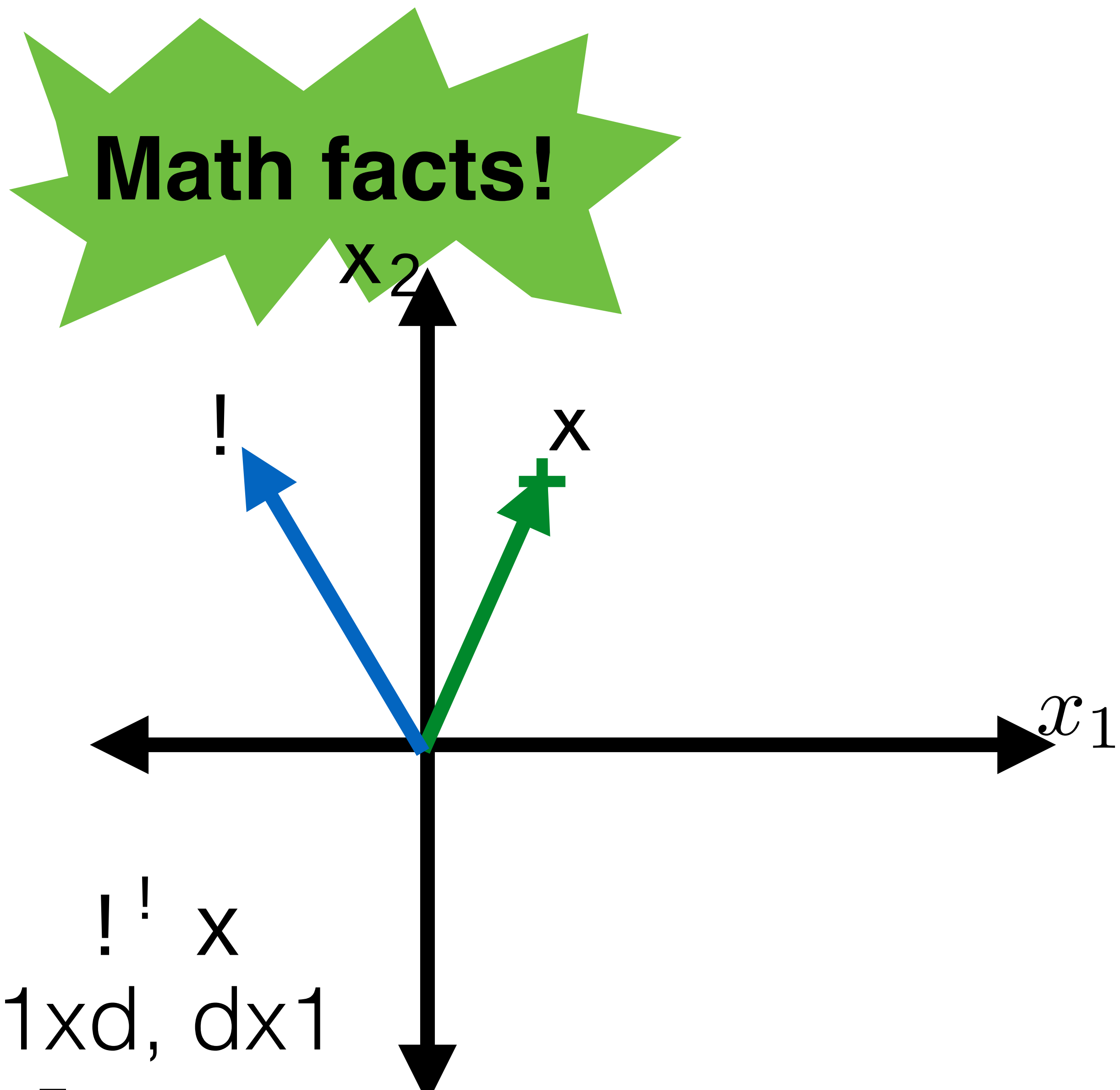




Linear classifiers

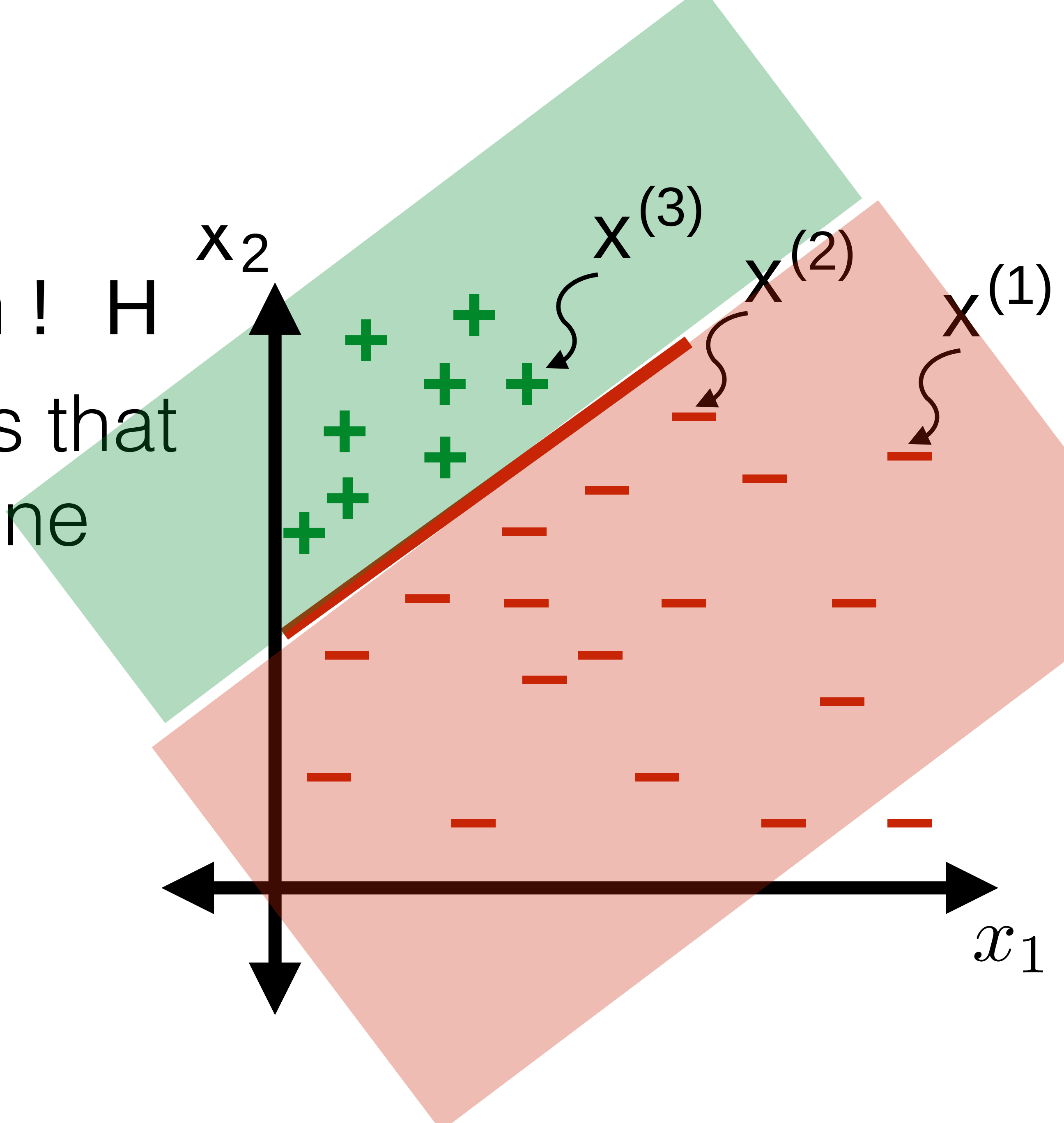
- Hypothesis class $\mathcal{H}$: set of **h ! H**
  - Example $\mathcal{H}$: All hypotheses that label +1 on one side of a line and -1 on the other side

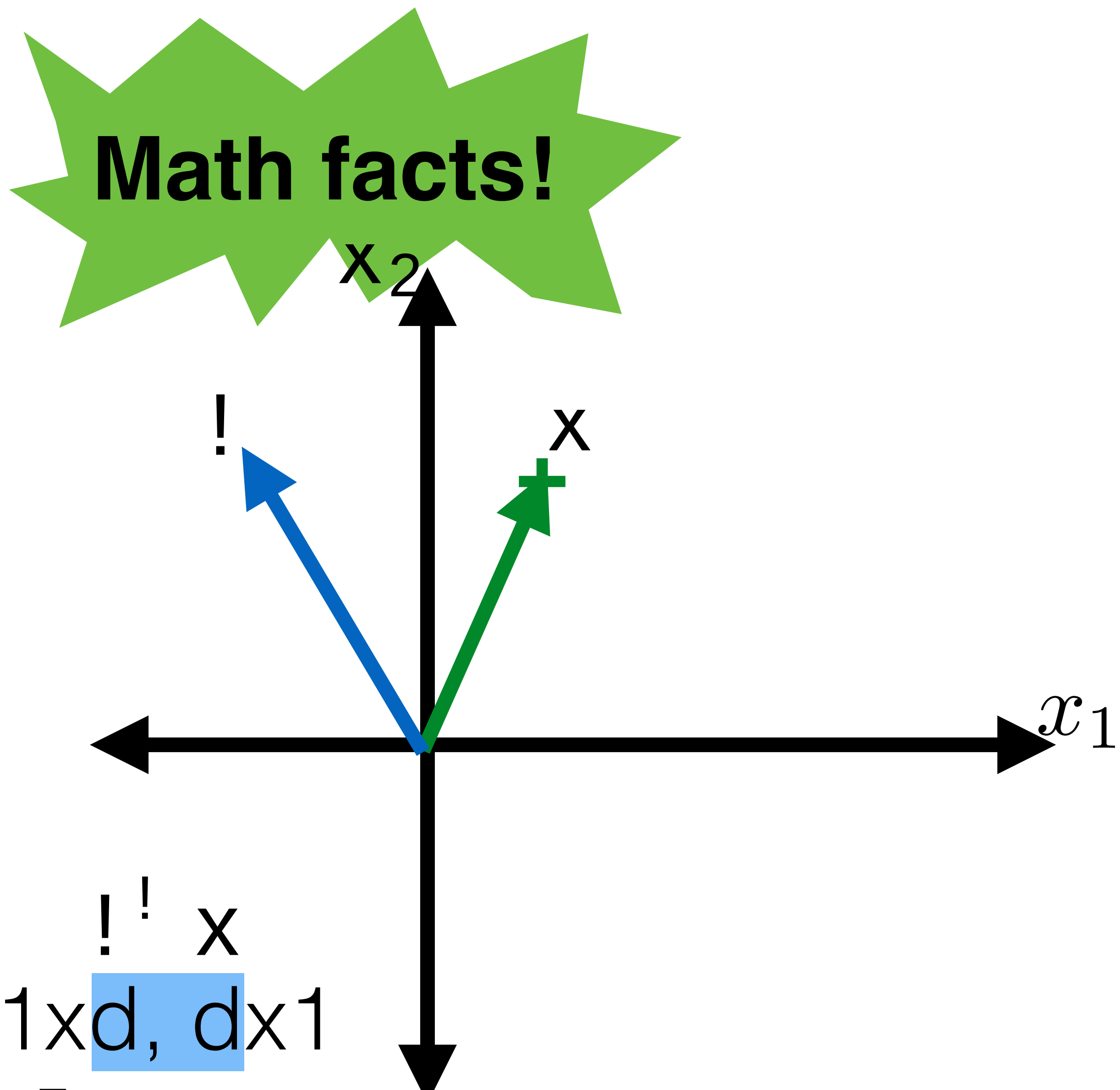




Linear classifiers

- Hypothesis class $\mathcal{H}$: set of **h** ! **H**
  - Example $\mathcal{H}$: All hypotheses that label +1 on one side of a line and -1 on the other side



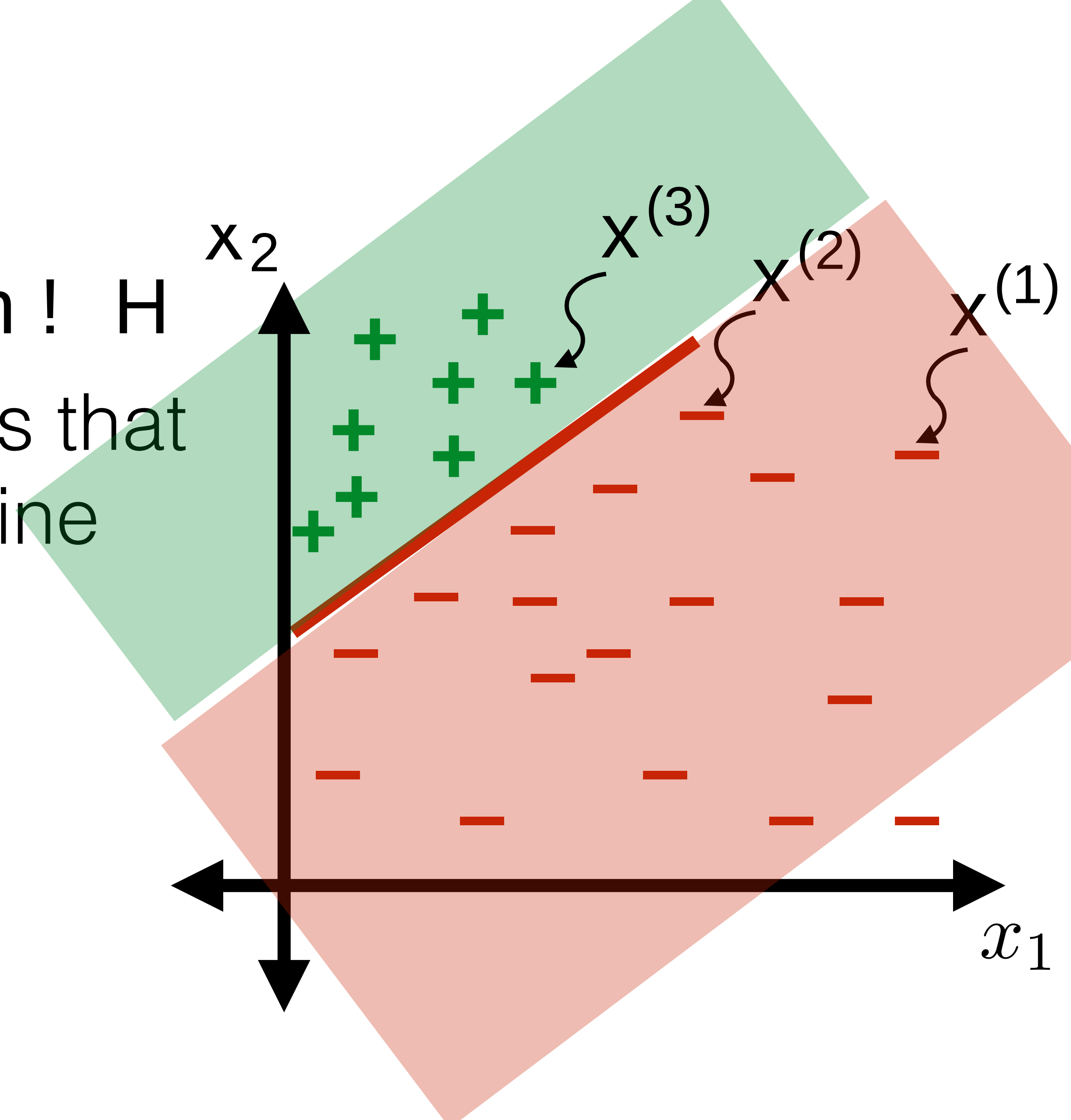


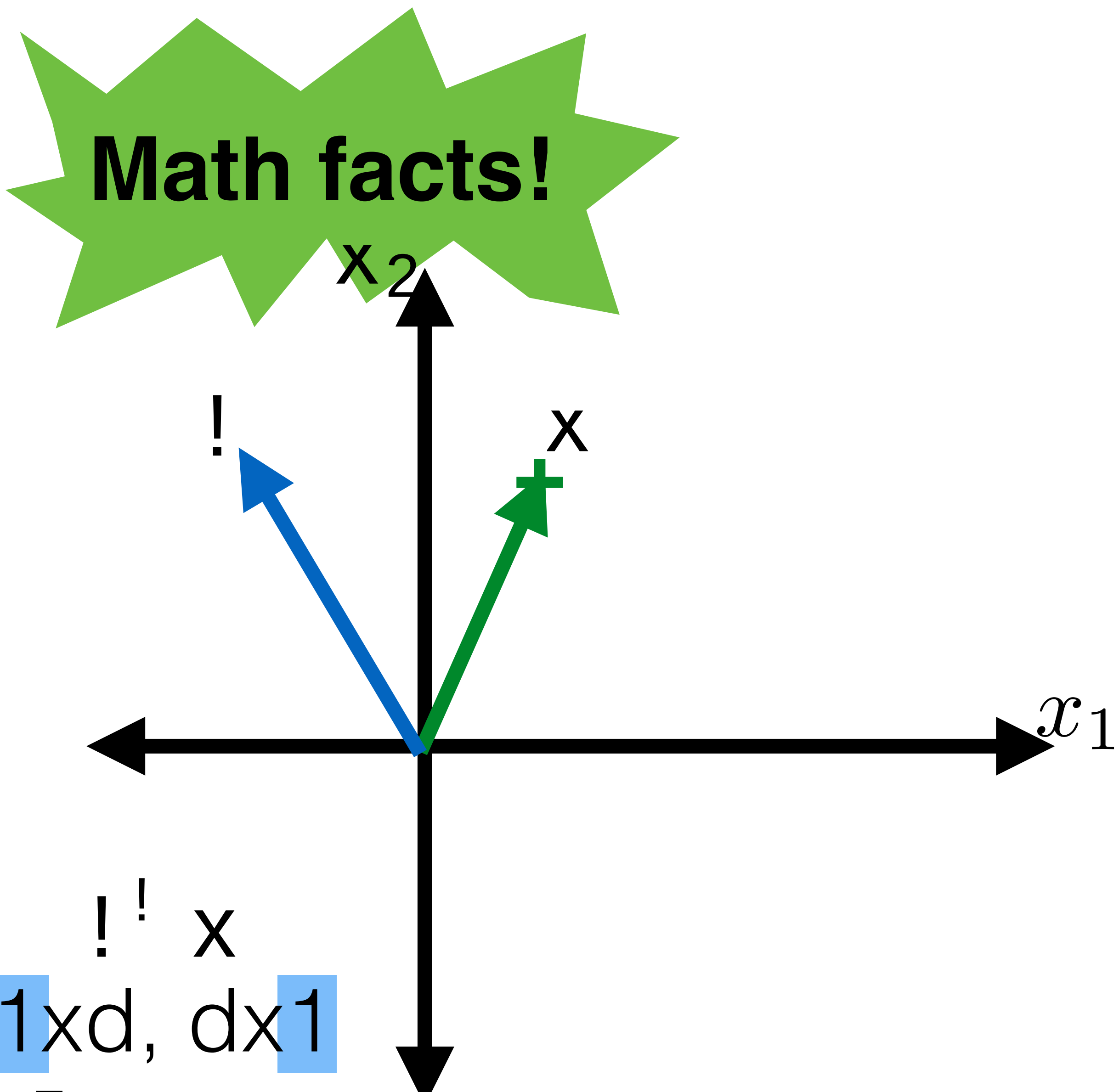
! ! x

1xd, dx1

Linear classifiers

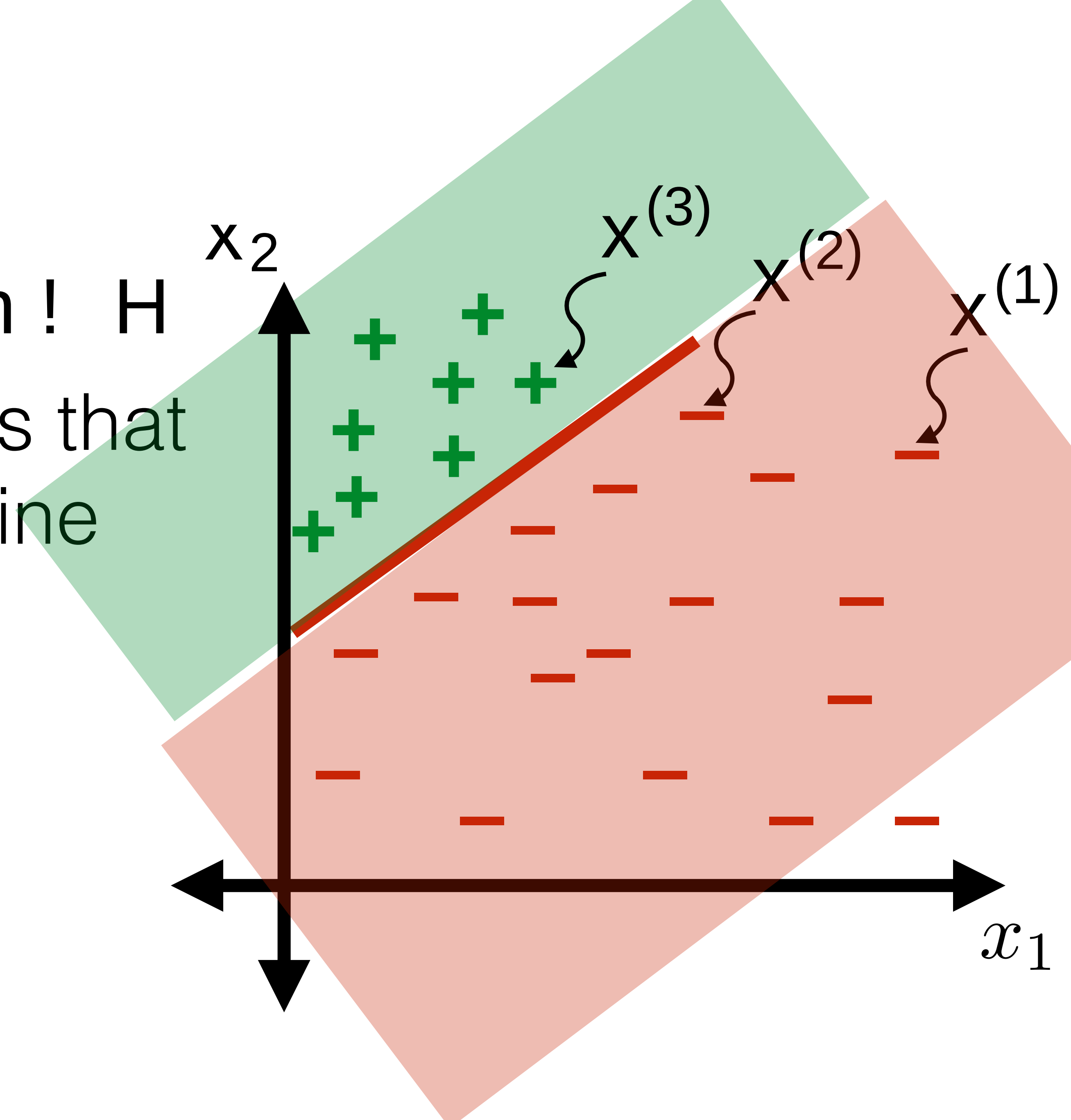
- Hypothesis class $\mathcal{H}$: set of **h ! H**
  - Example $\mathcal{H}$: All hypotheses that label +1 on one side of a line and -1 on the other side



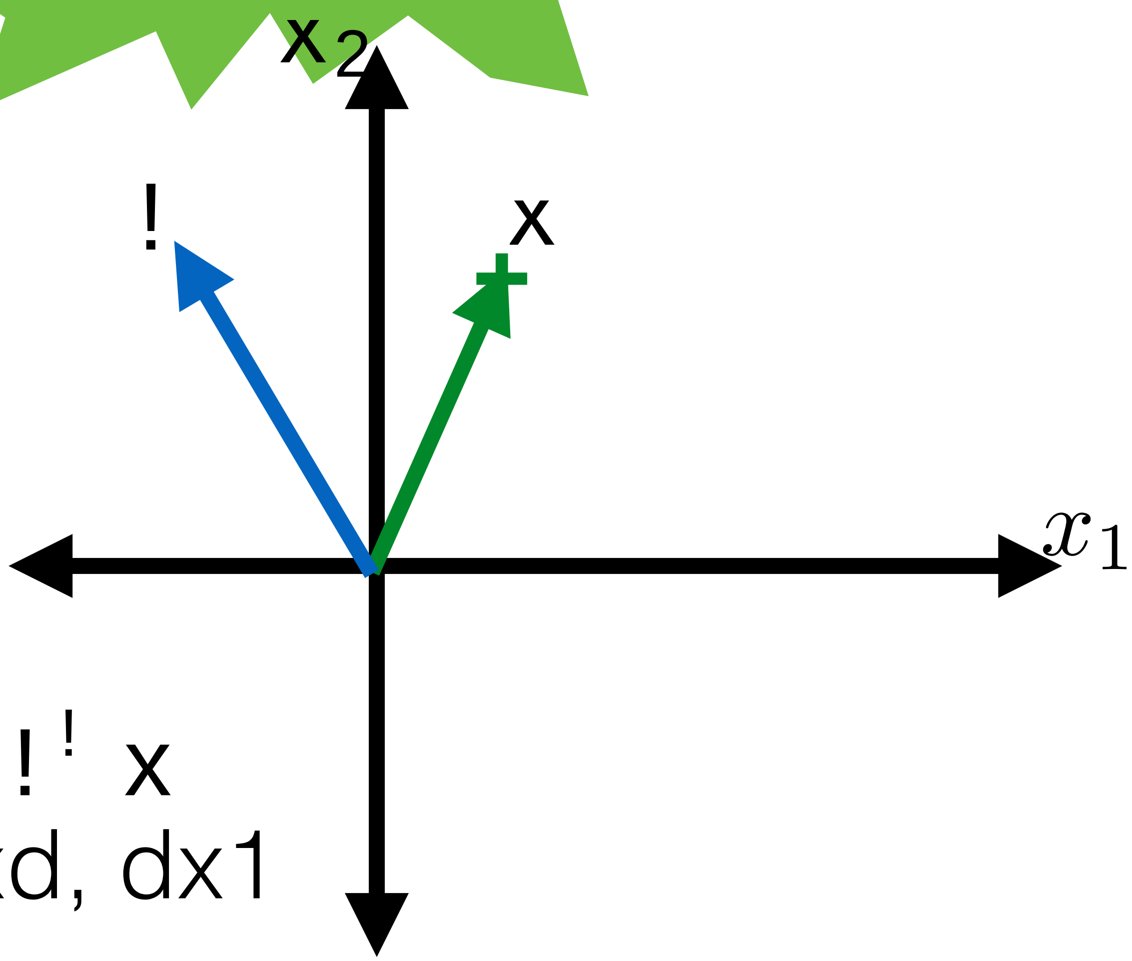


Linear classifiers

- Hypothesis class $\mathcal{H}$: set of **h ! H**
  - Example $\mathcal{H}$: All hypotheses that label +1 on one side of a line and -1 on the other side



Math facts!

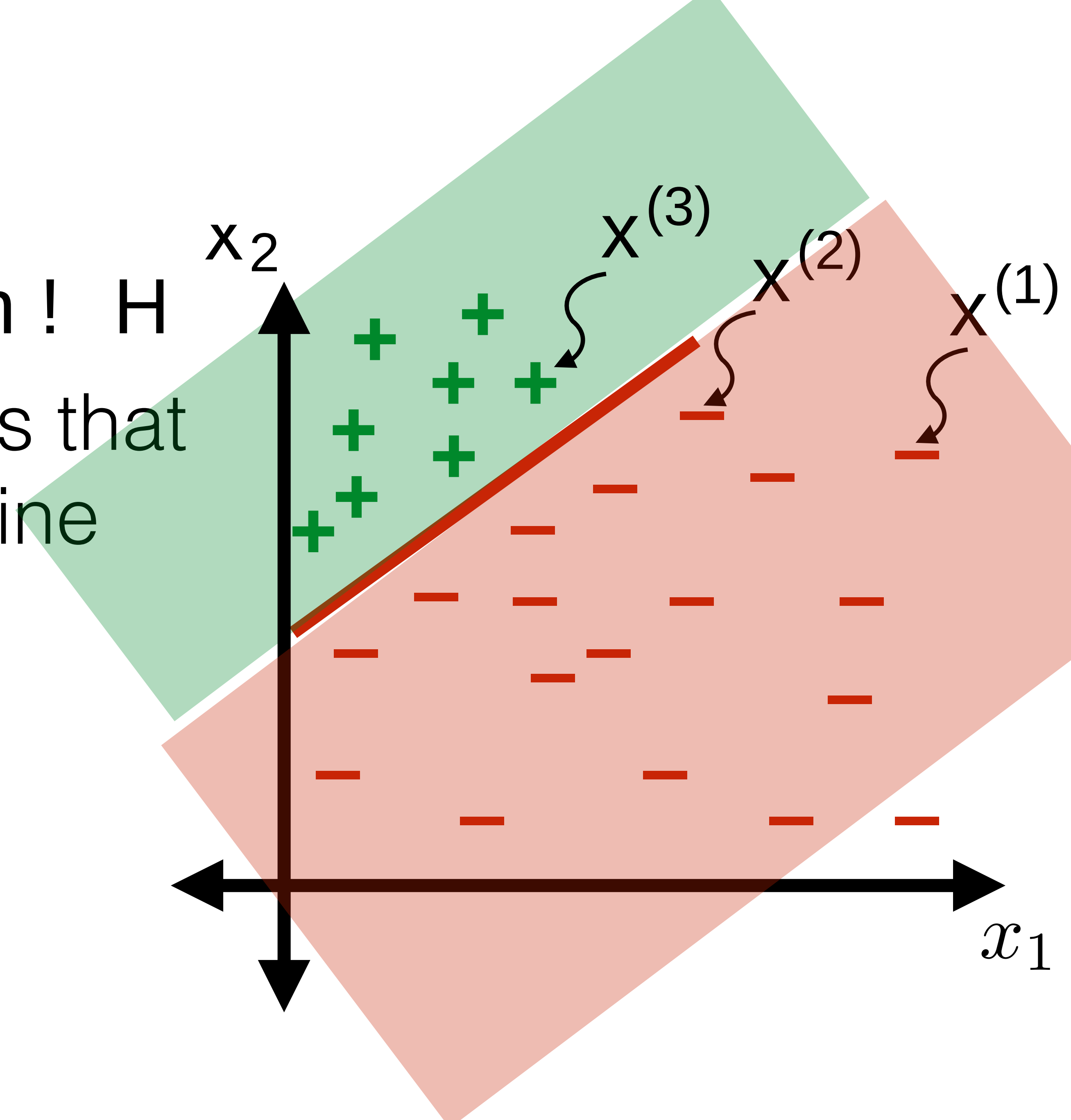


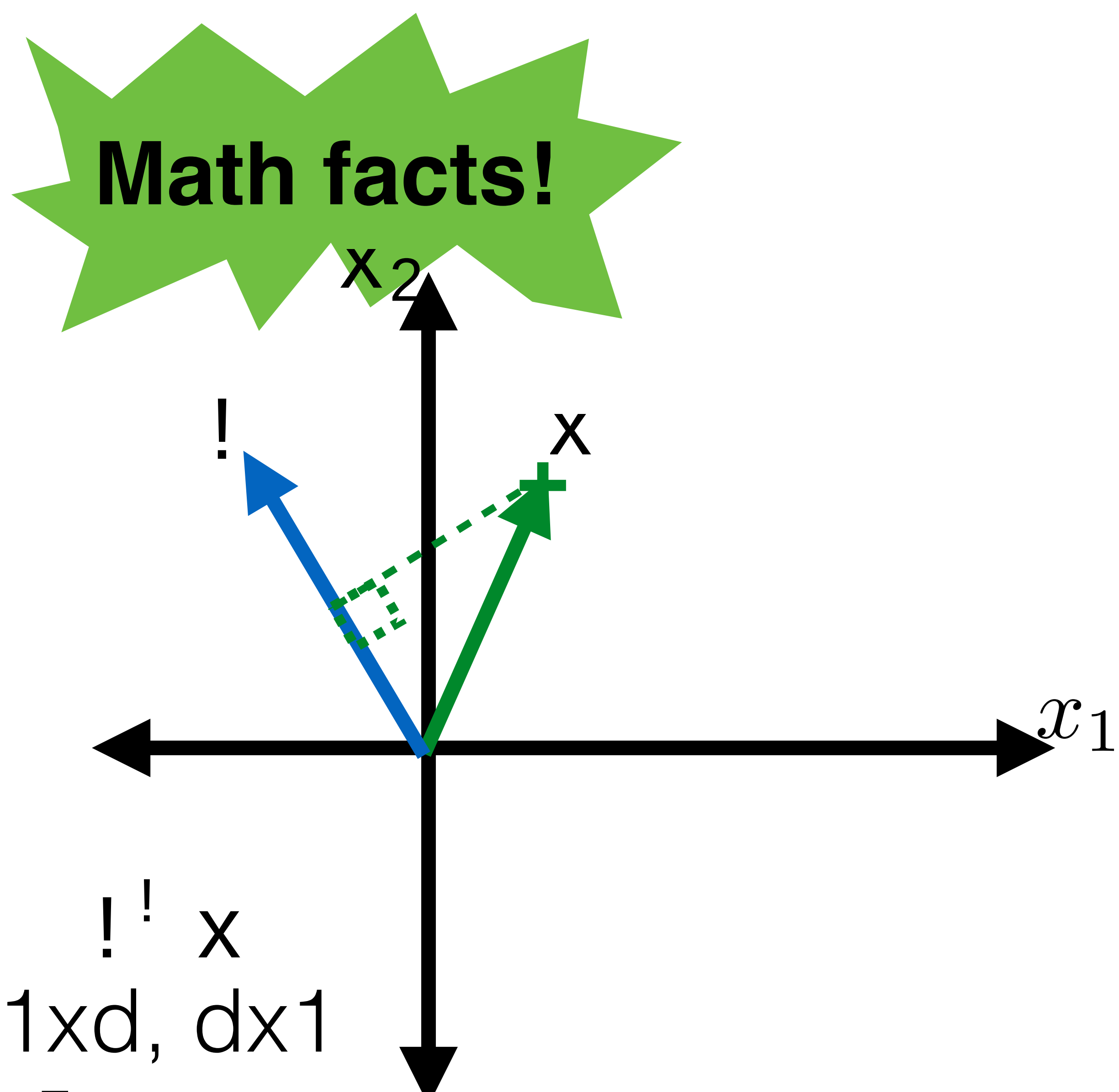
! ! x

1xd, dx1

Linear classifiers

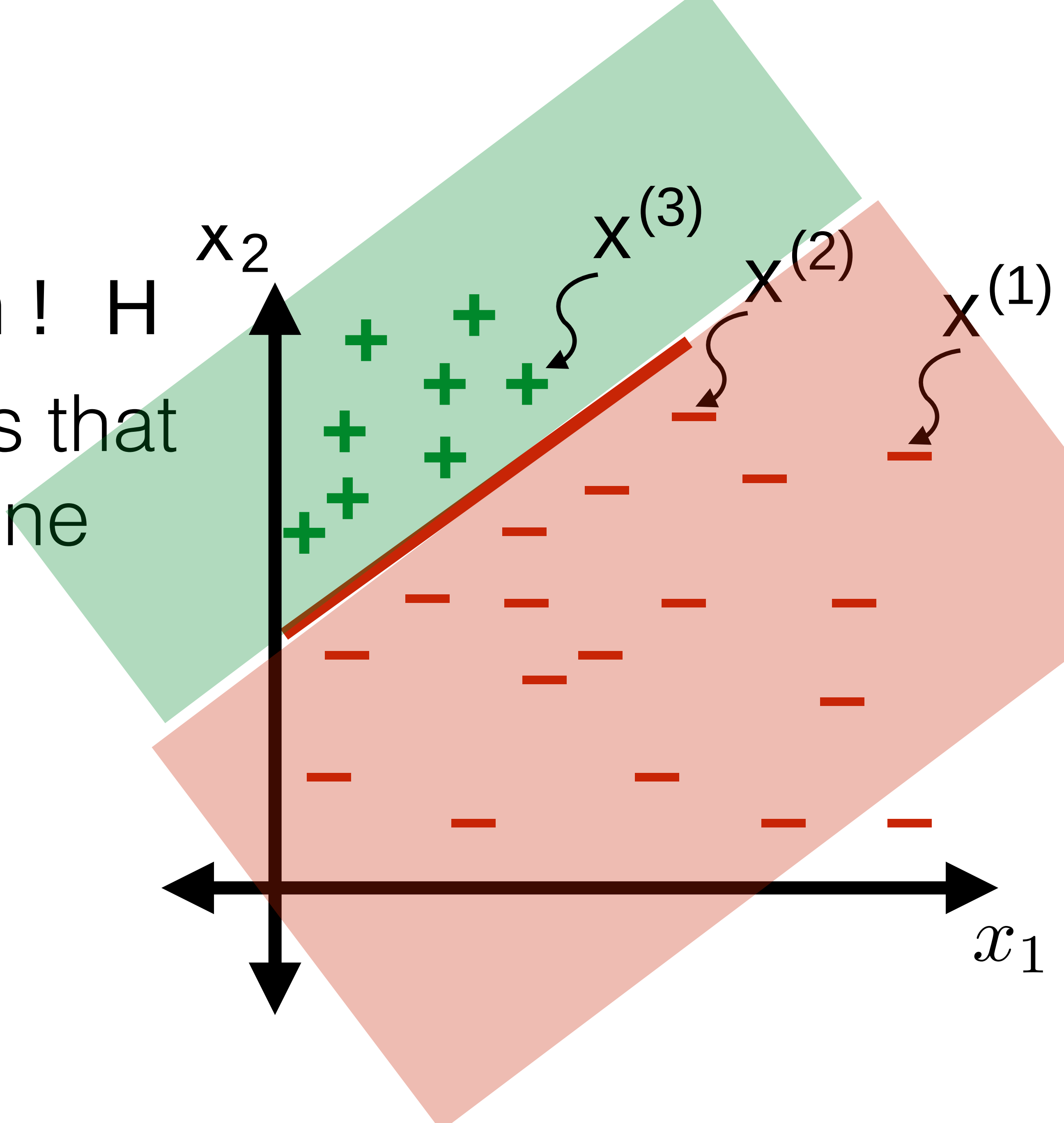
- Hypothesis class $\mathcal{H}$: set of **h ! H**
  - Example $\mathcal{H}$: All hypotheses that label +1 on one side of a line and -1 on the other side

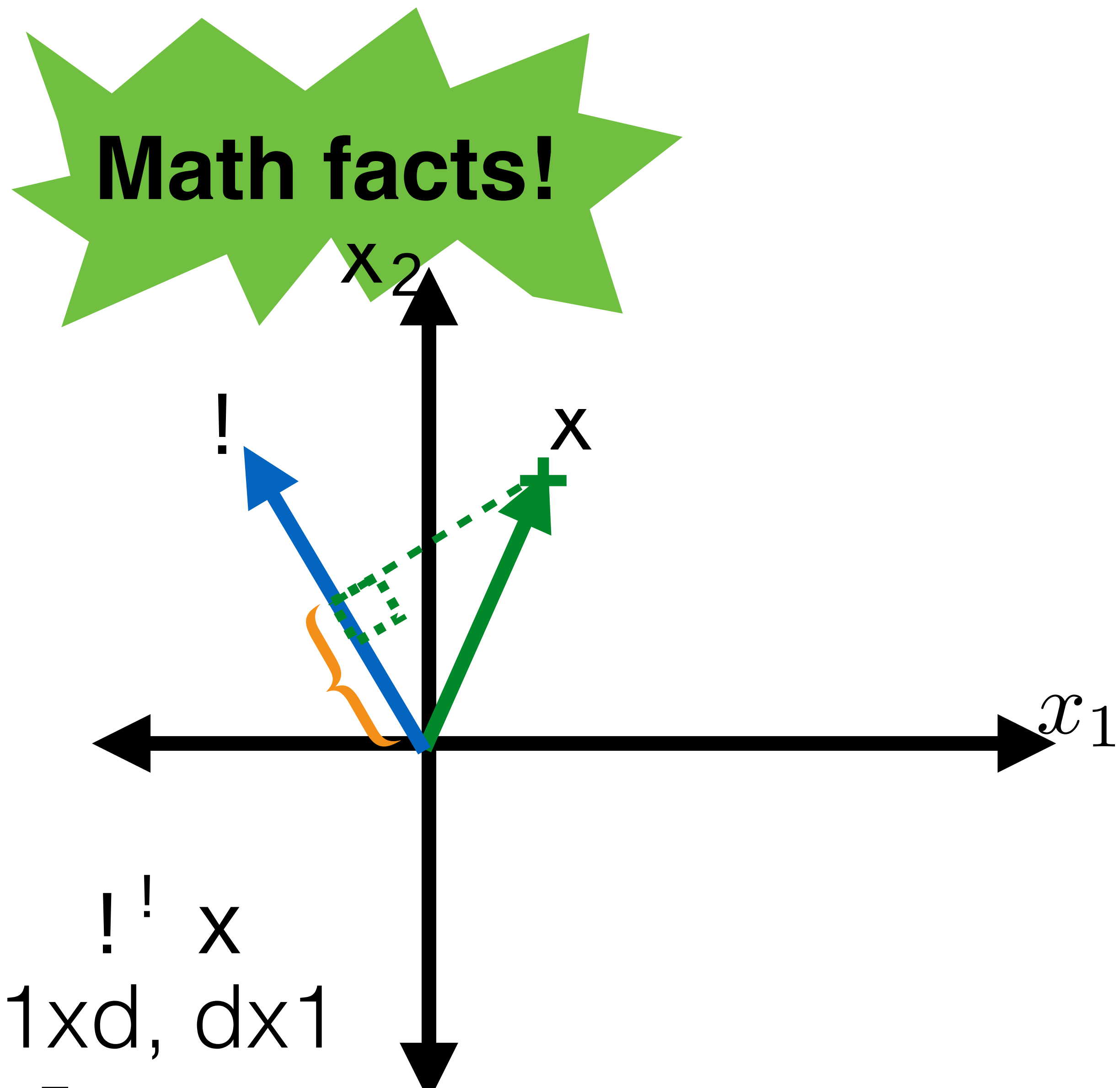




Linear classifiers

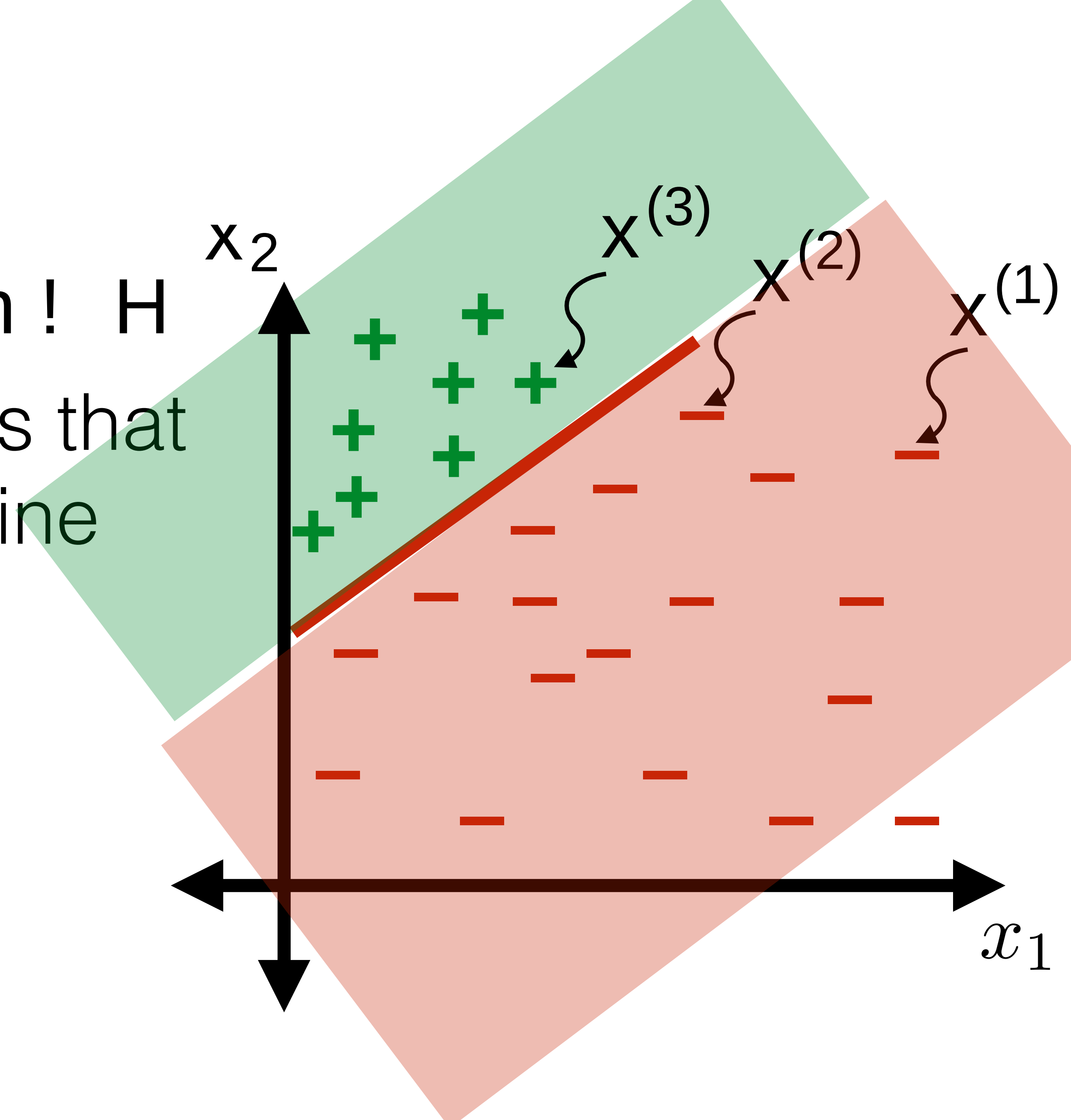
- Hypothesis class $\mathcal{H}$: set of **h** ! **H**
  - Example $\mathcal{H}$: All hypotheses that label +1 on one side of a line and -1 on the other side

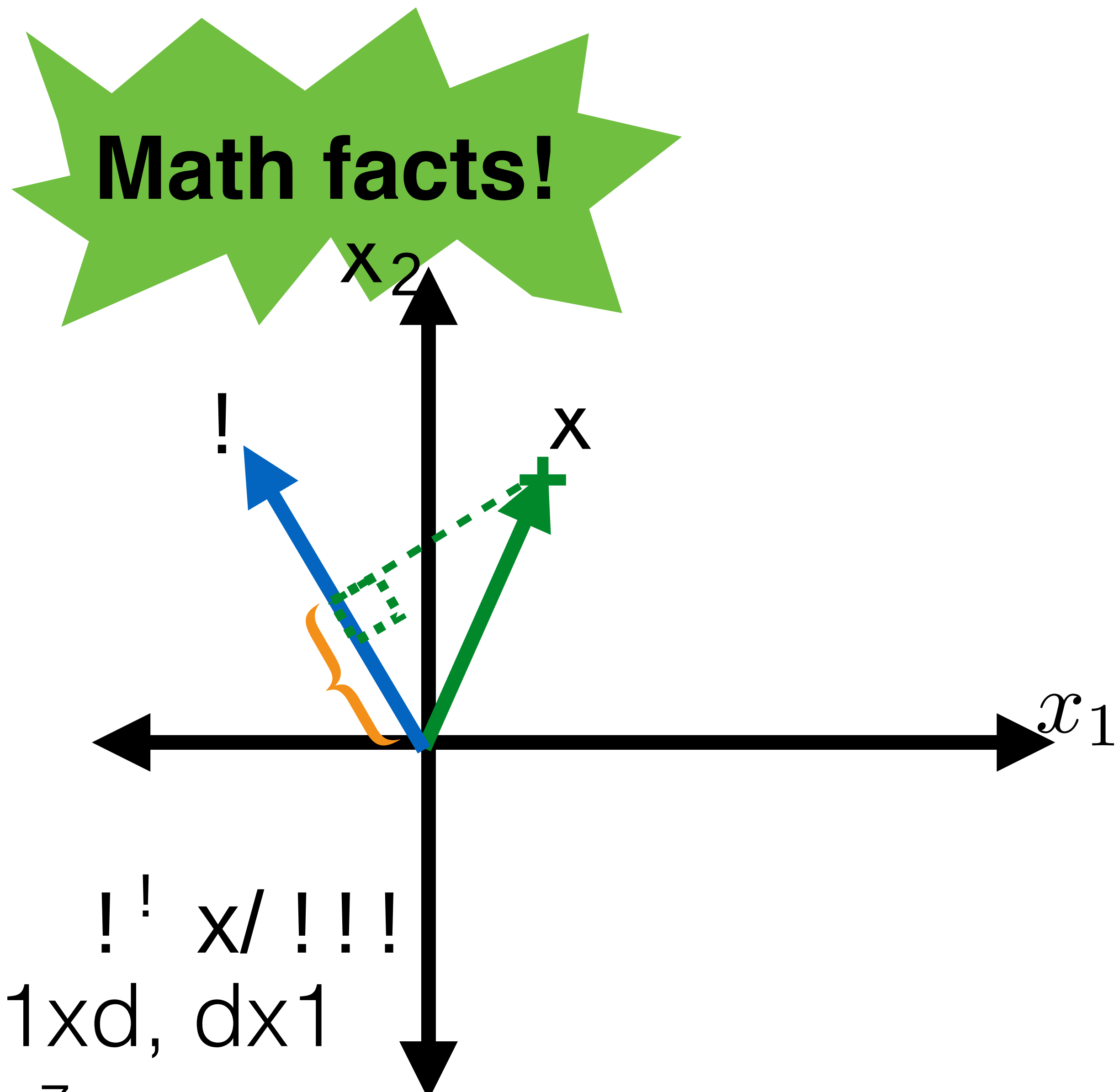




Linear classifiers

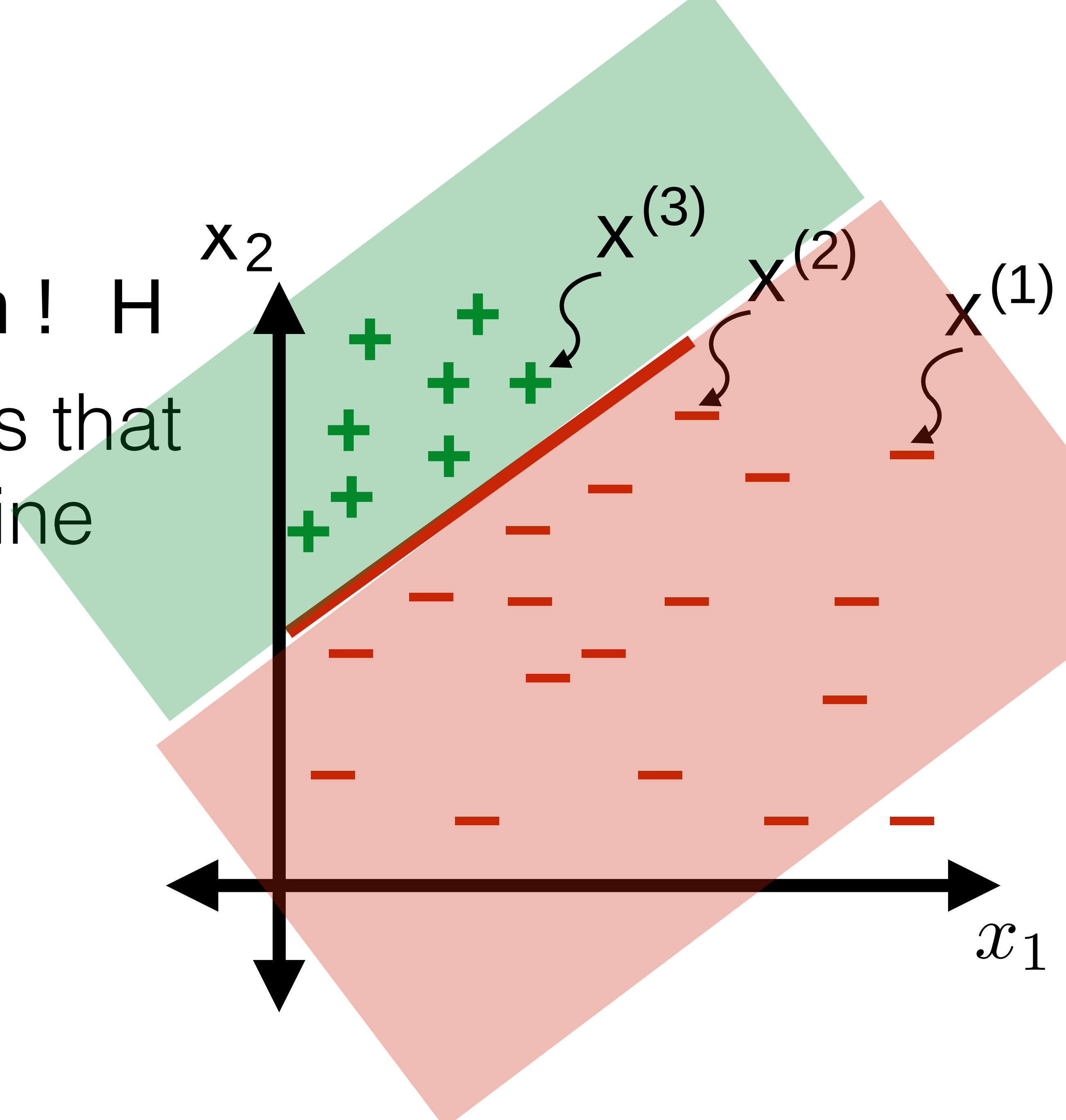
- Hypothesis class $\mathcal{H}$: set of **h** ! **H**
  - Example $\mathcal{H}$: All hypotheses that label +1 on one side of a line and -1 on the other side

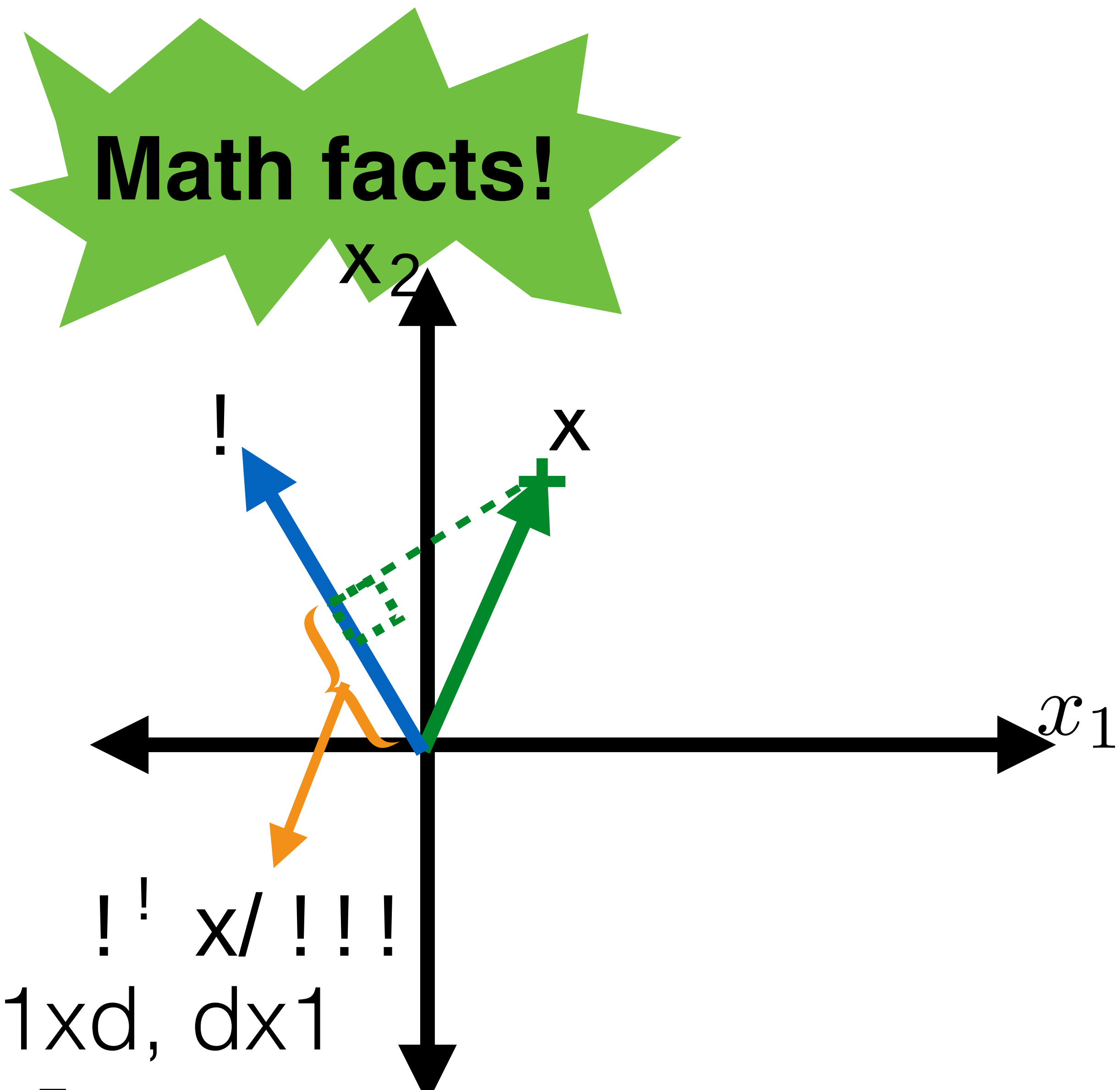




Linear classifiers

- Hypothesis class $\mathcal{H}$: set of **h** ! **H**
  - Example $\mathcal{H}$: All hypotheses that label +1 on one side of a line and -1 on the other side

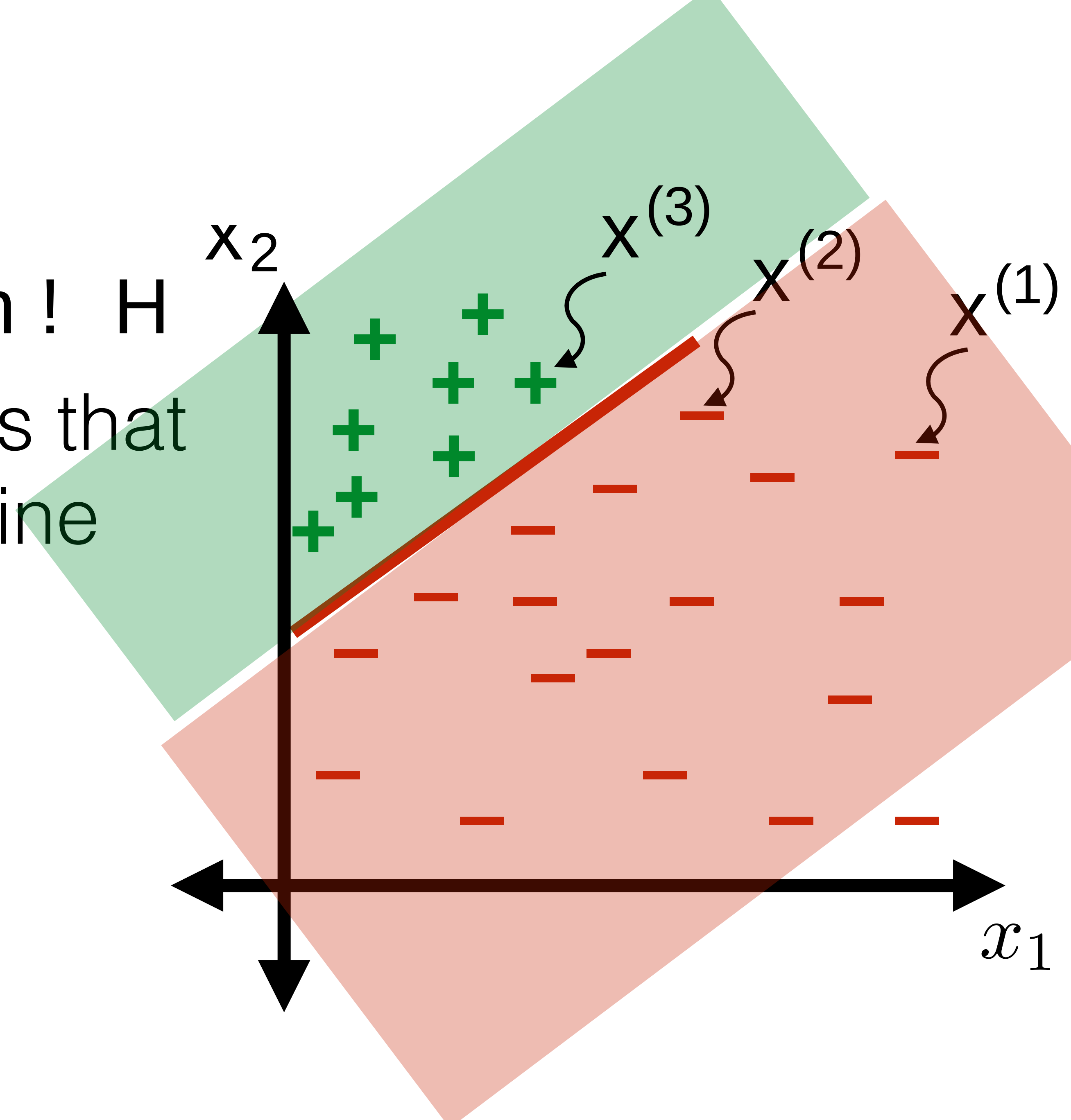


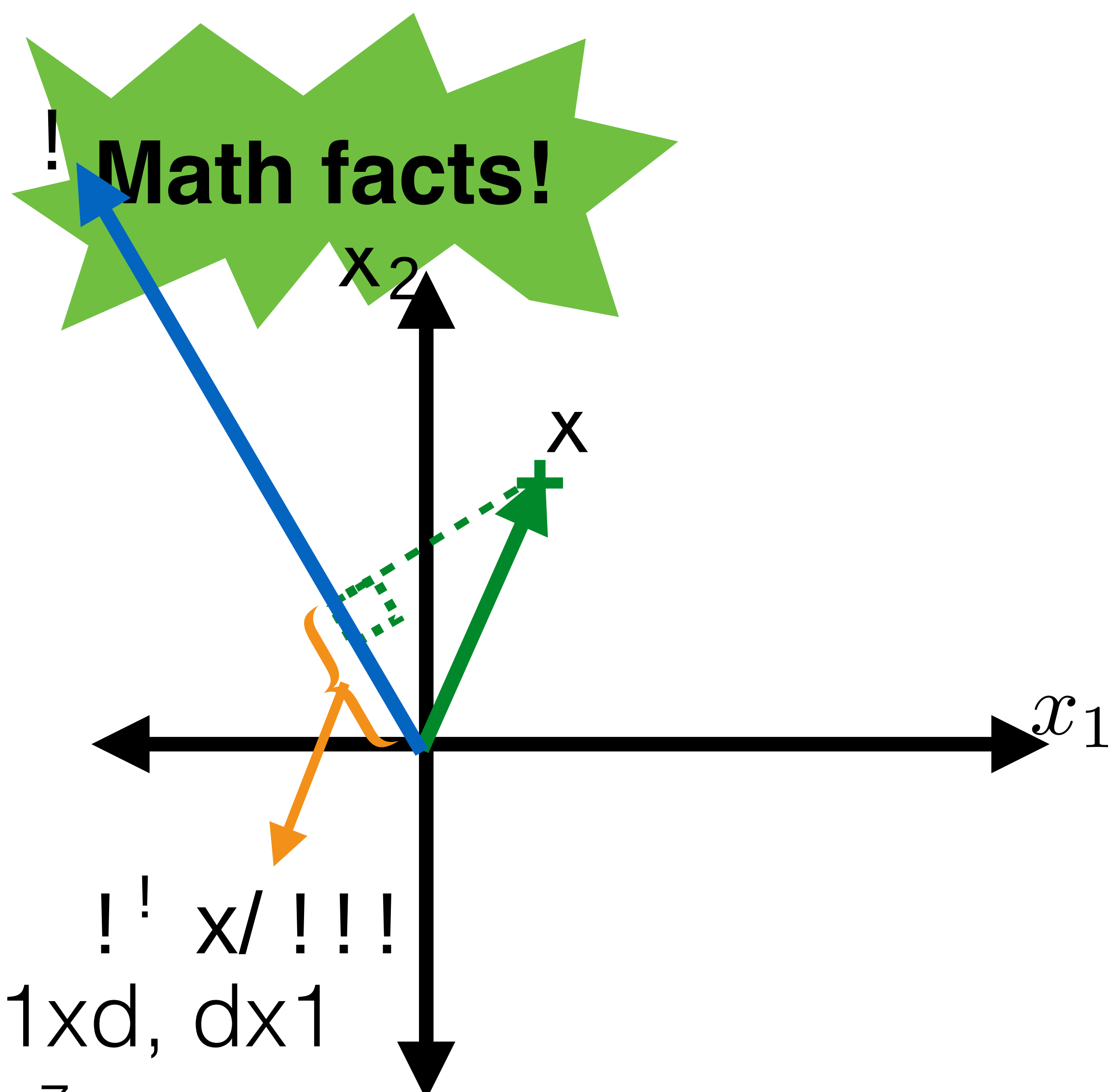


! ! x/ ! ! !
1xd, dx1

Linear classifiers

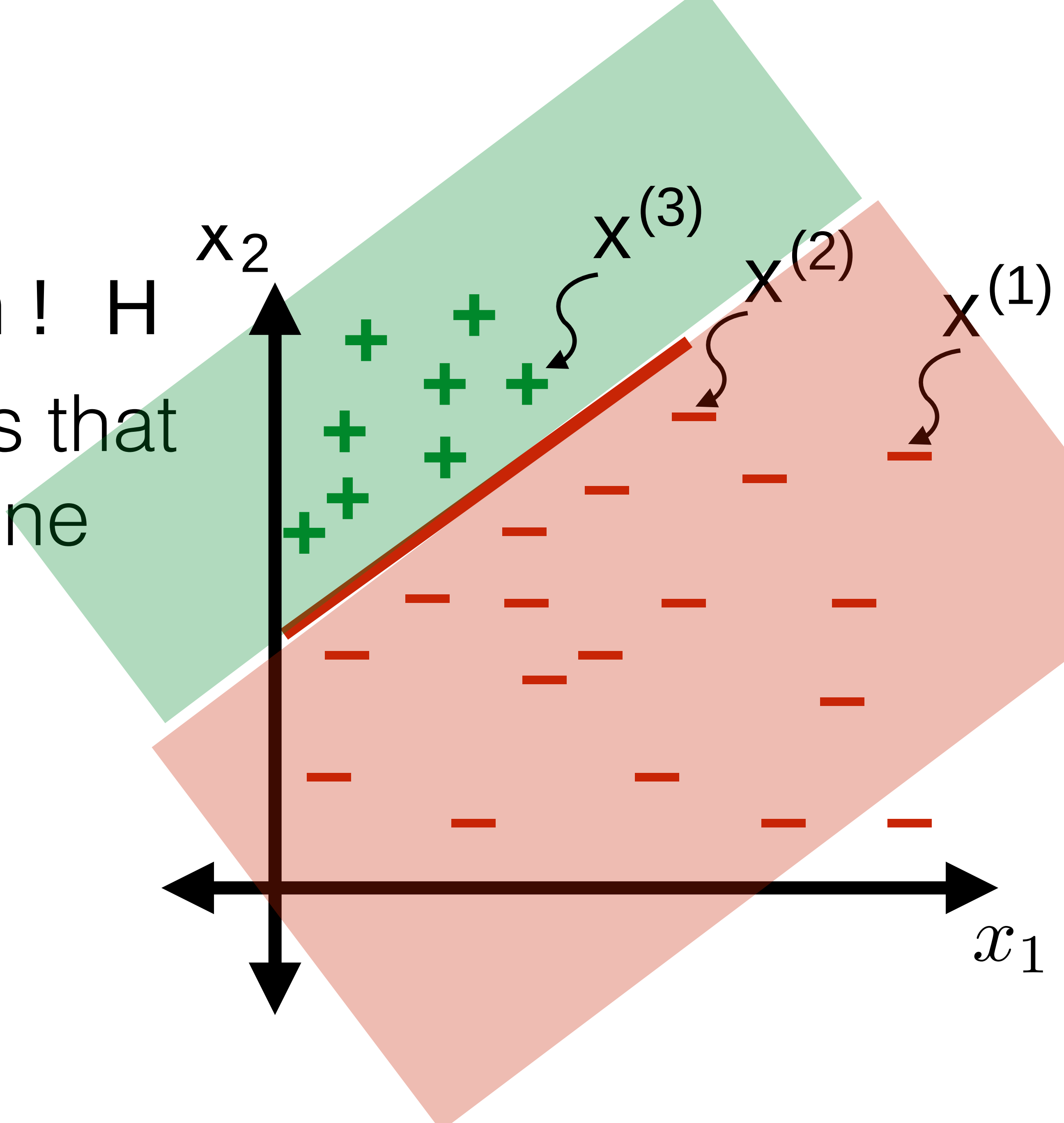
- Hypothesis class $\mathcal{H}$: set of **h** ! **H**
    - Example $\mathcal{H}$: All hypotheses that label +1 on one side of a line and -1 on the other side

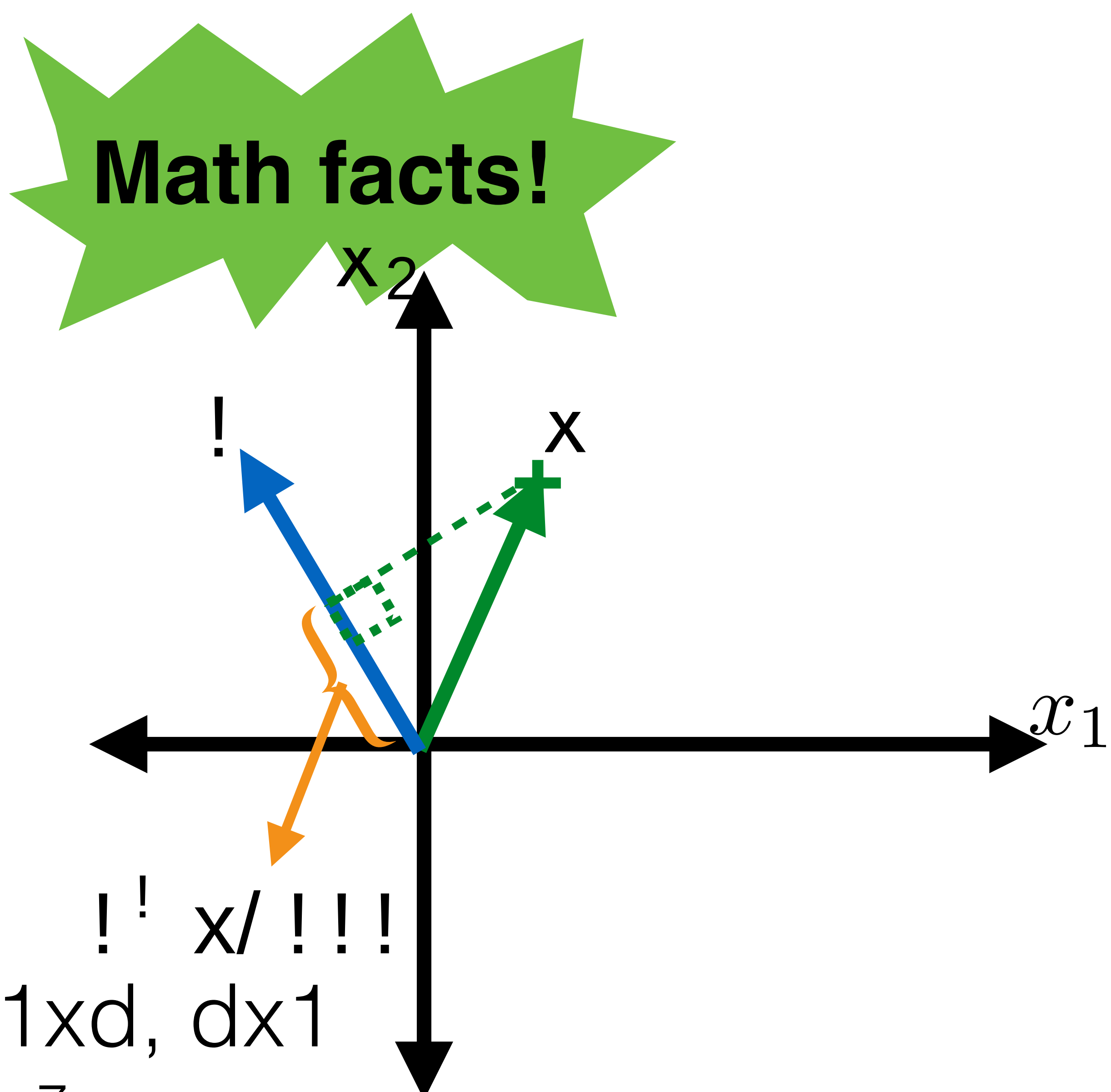




Linear classifiers

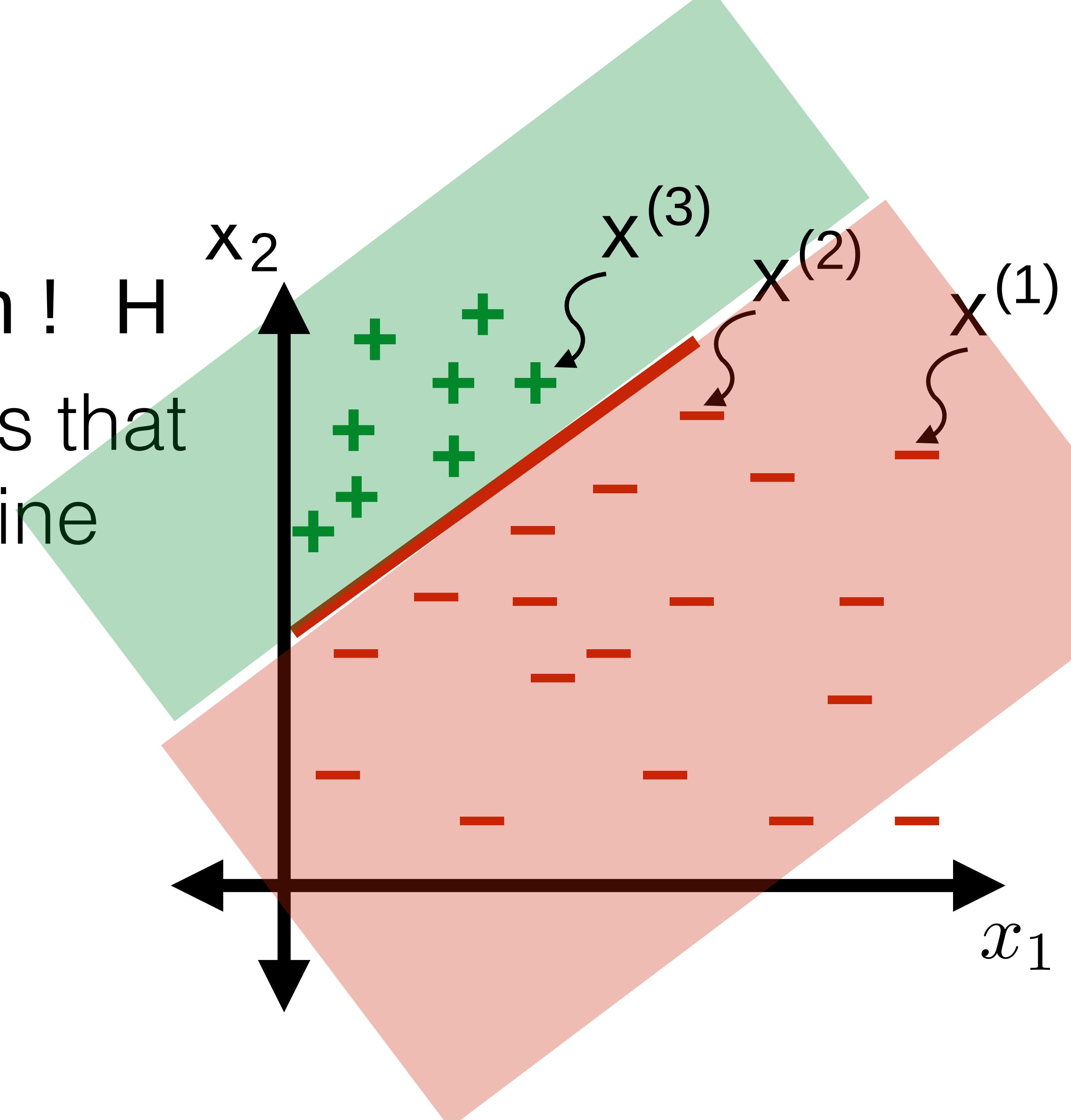
- Hypothesis class $\mathcal{H}$: set of **h** ! **H**
  - Example $\mathcal{H}$: All hypotheses that label +1 on one side of a line and -1 on the other side



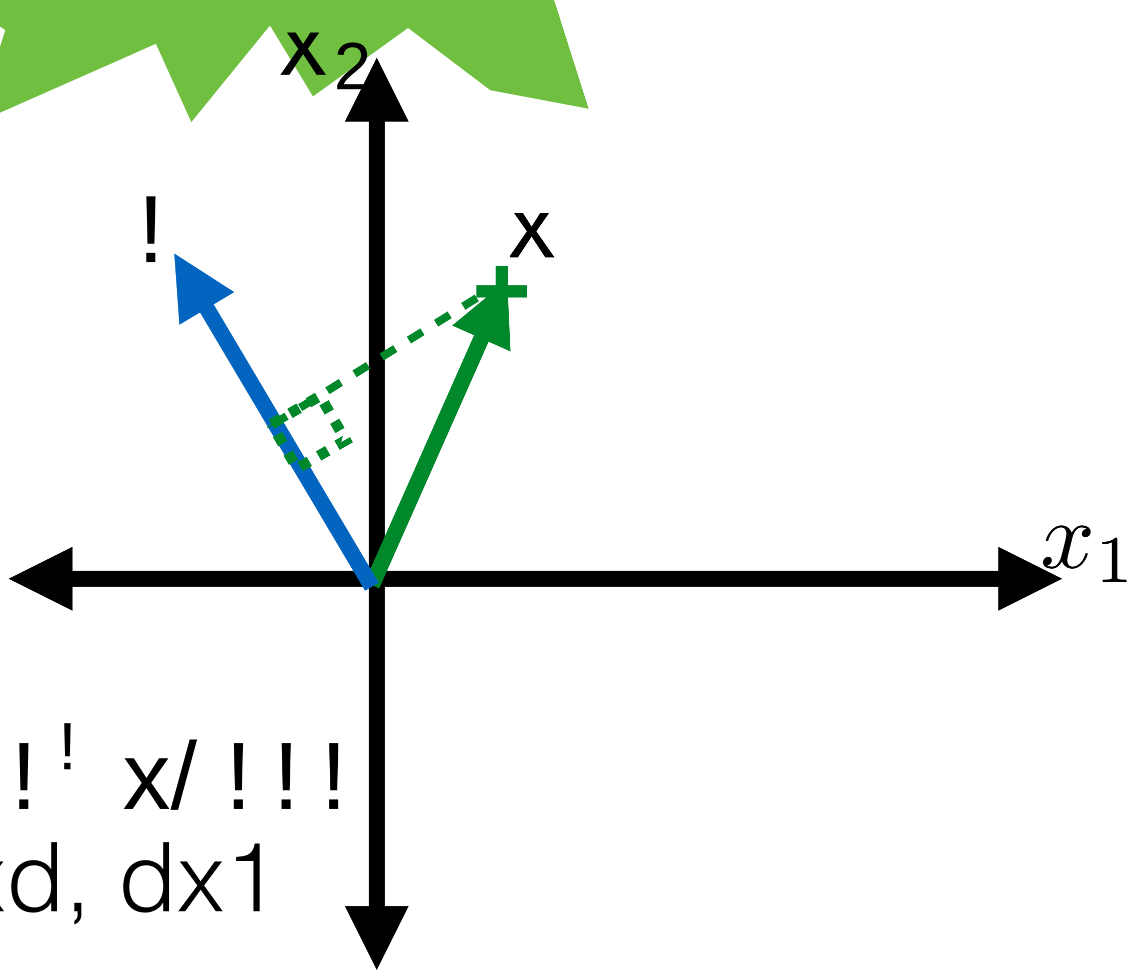


Linear classifiers

- Hypothesis class $\mathcal{H}$: set of **h ! H**
  - Example $\mathcal{H}$: All hypotheses that label +1 on one side of a line and -1 on the other side



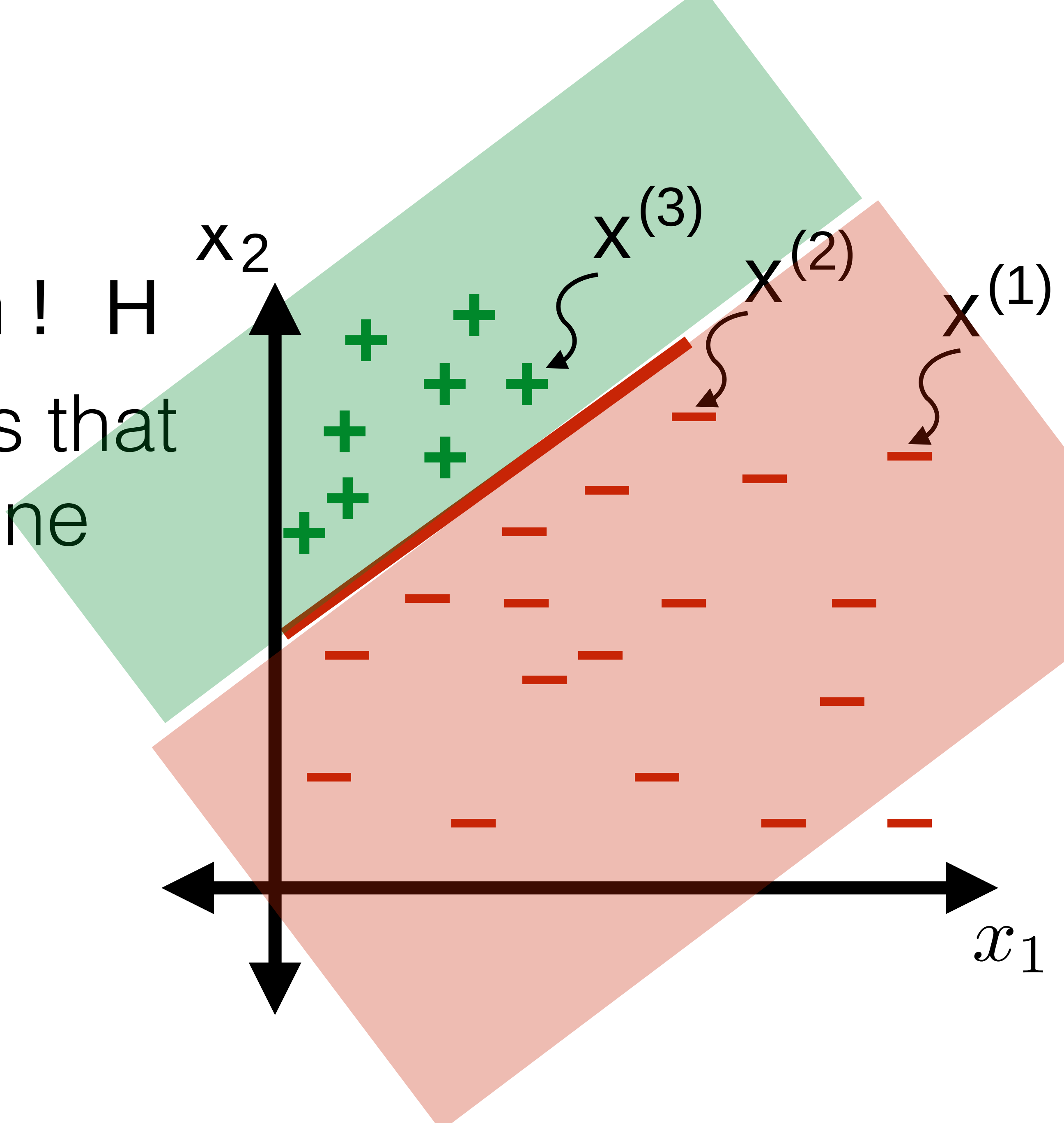
Math facts!

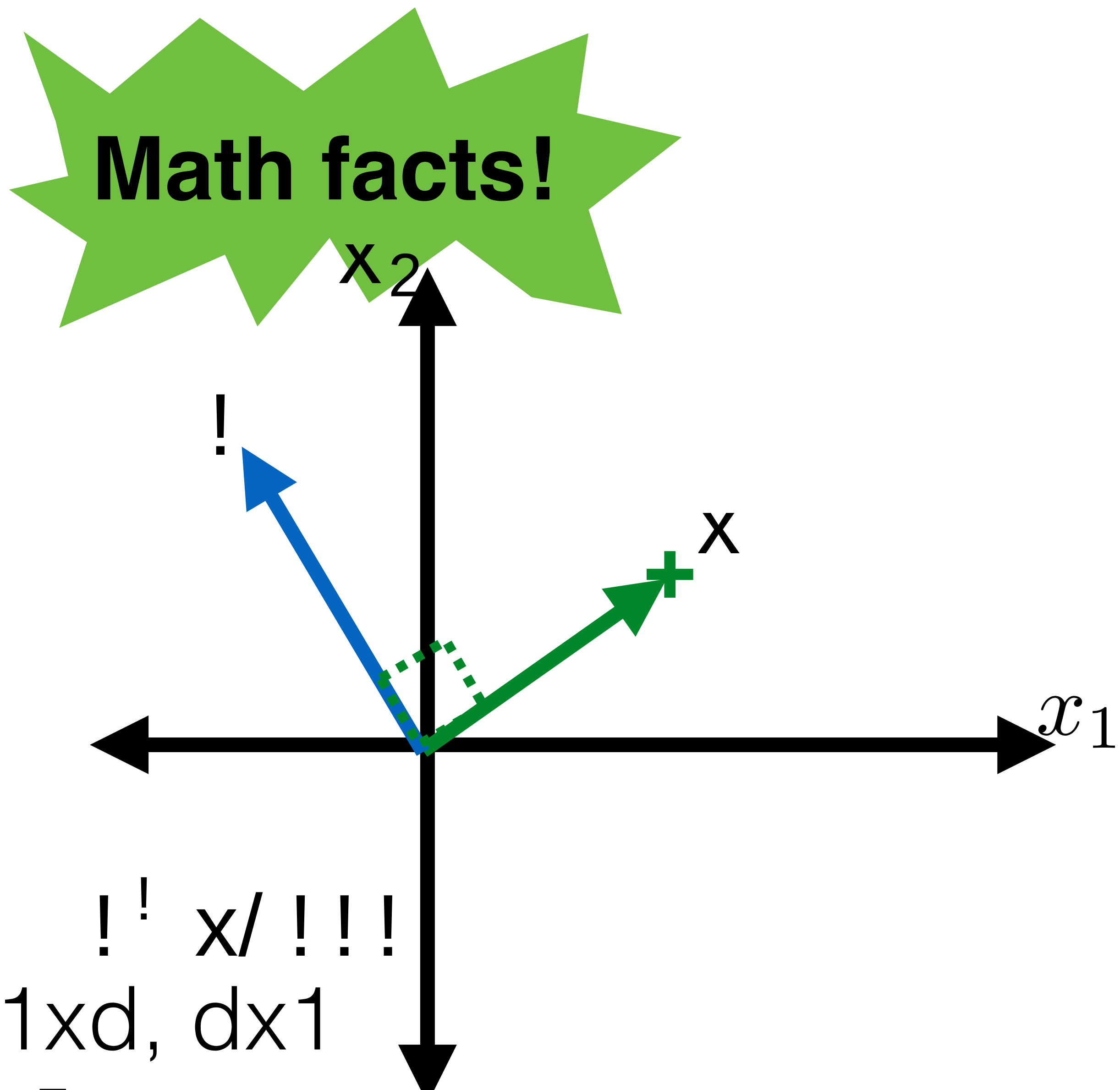


! ! x/ ! ! !
1xd, dx1

Linear classifiers

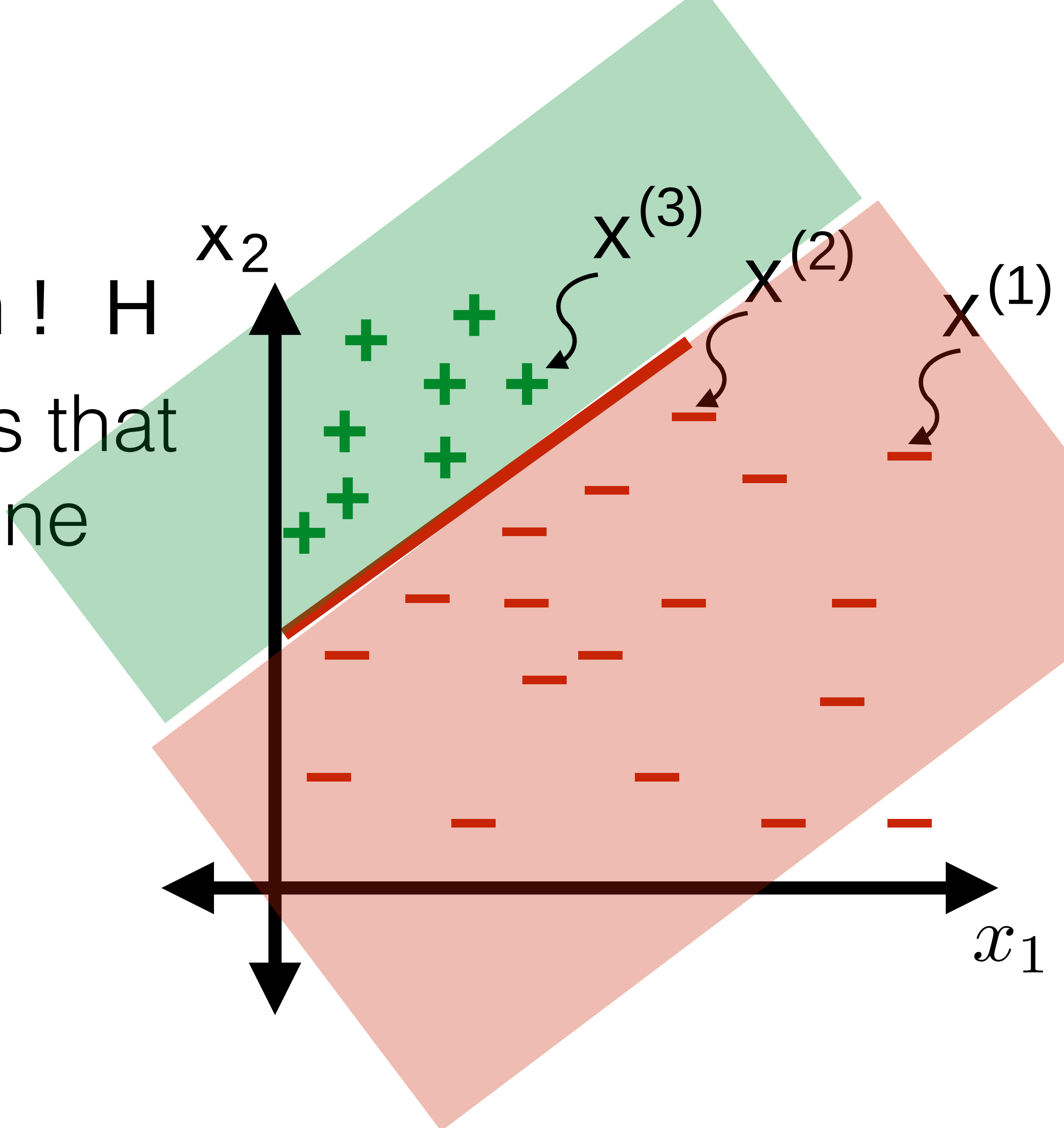
- Hypothesis class $\mathcal{H}$: set of **h** ! **H**
  - Example $\mathcal{H}$: All hypotheses that label +1 on one side of a line and -1 on the other side

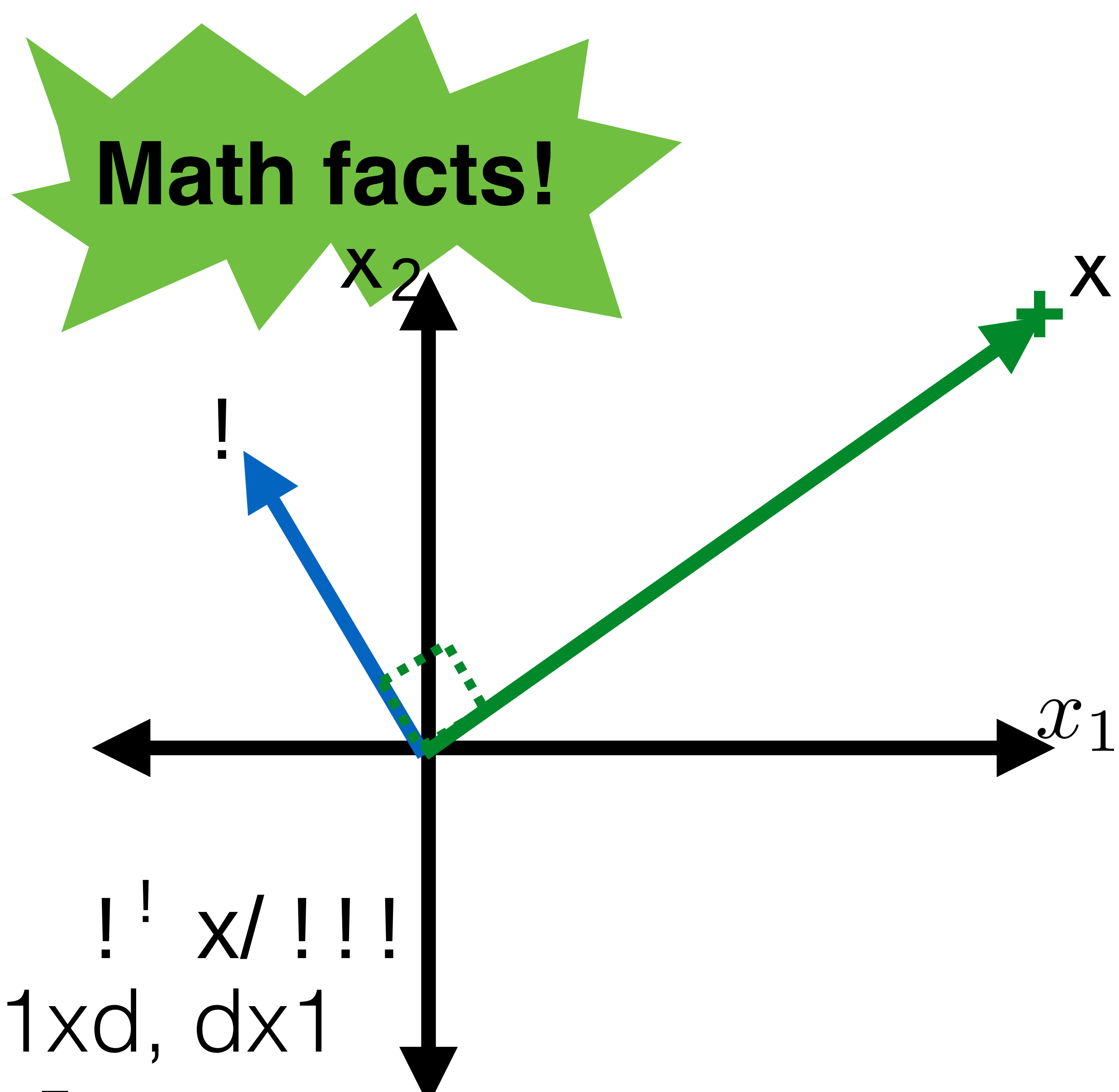




Linear classifiers

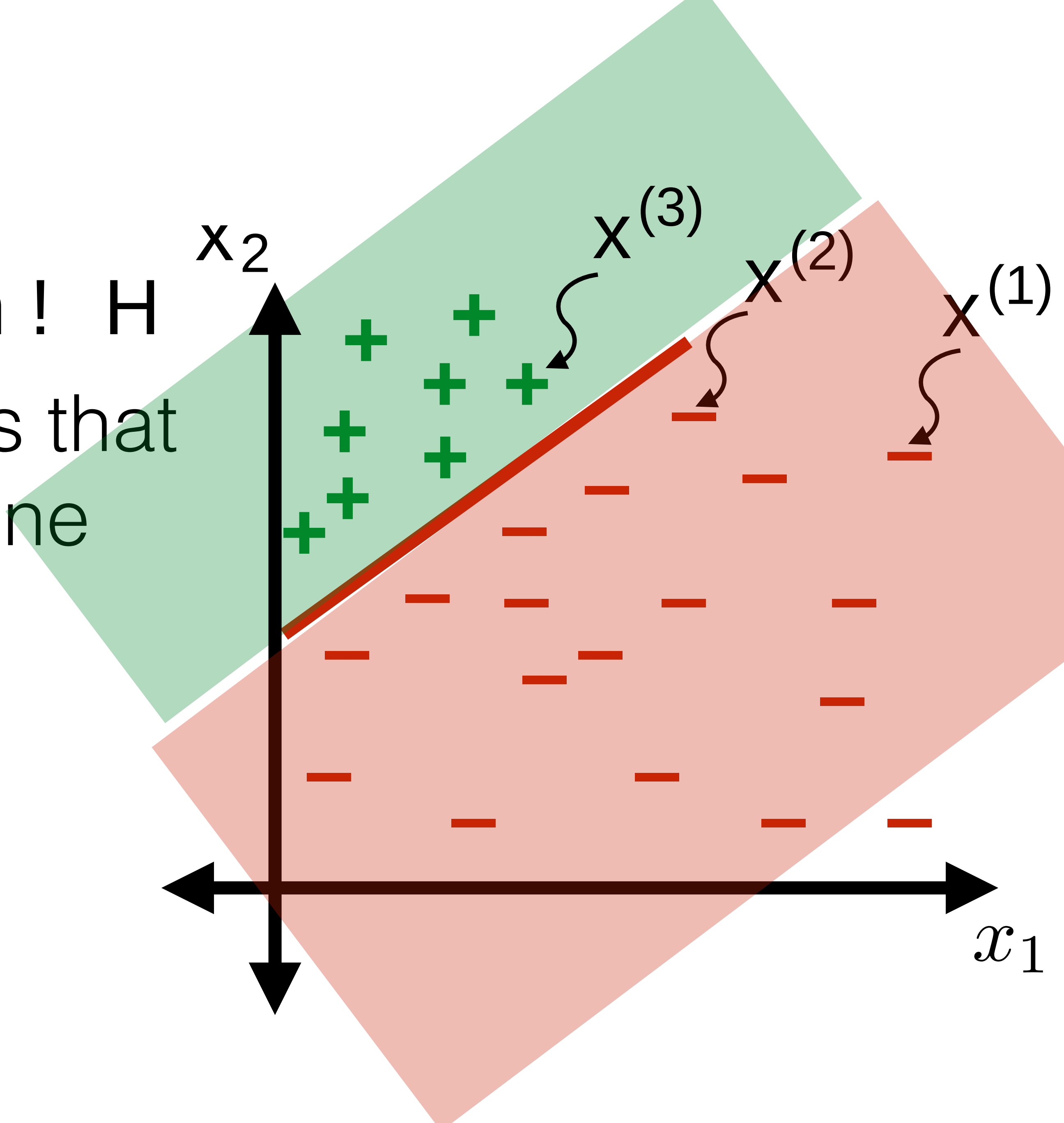
- Hypothesis class $\mathcal{H}$: set of **h** ! **H**
  - Example $\mathcal{H}$: All hypotheses that label +1 on one side of a line and -1 on the other side

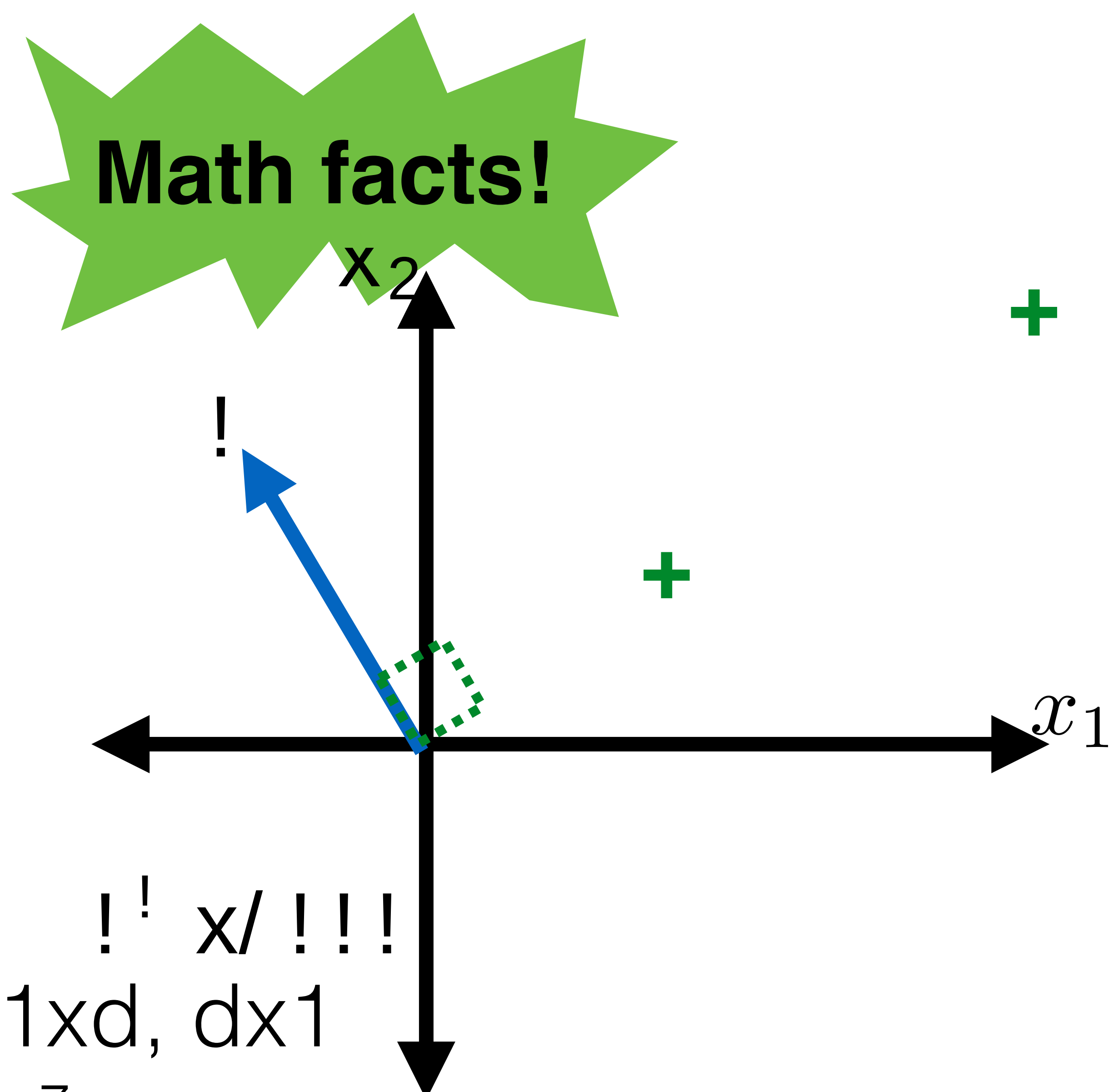




Linear classifiers

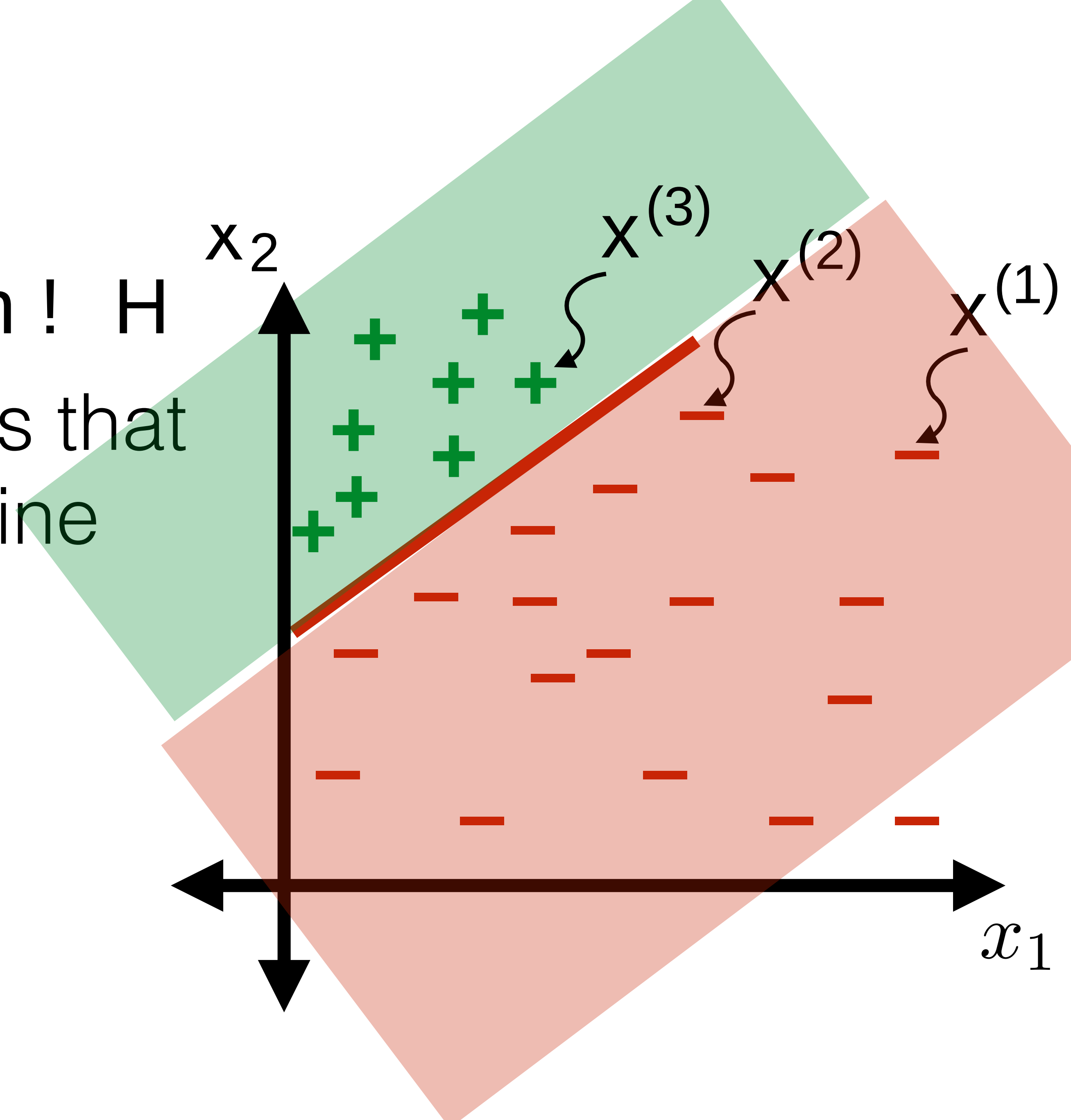
- Hypothesis class $\mathcal{H}$: set of **h** ! **H**
  - Example $\mathcal{H}$: All hypotheses that label +1 on one side of a line and -1 on the other side



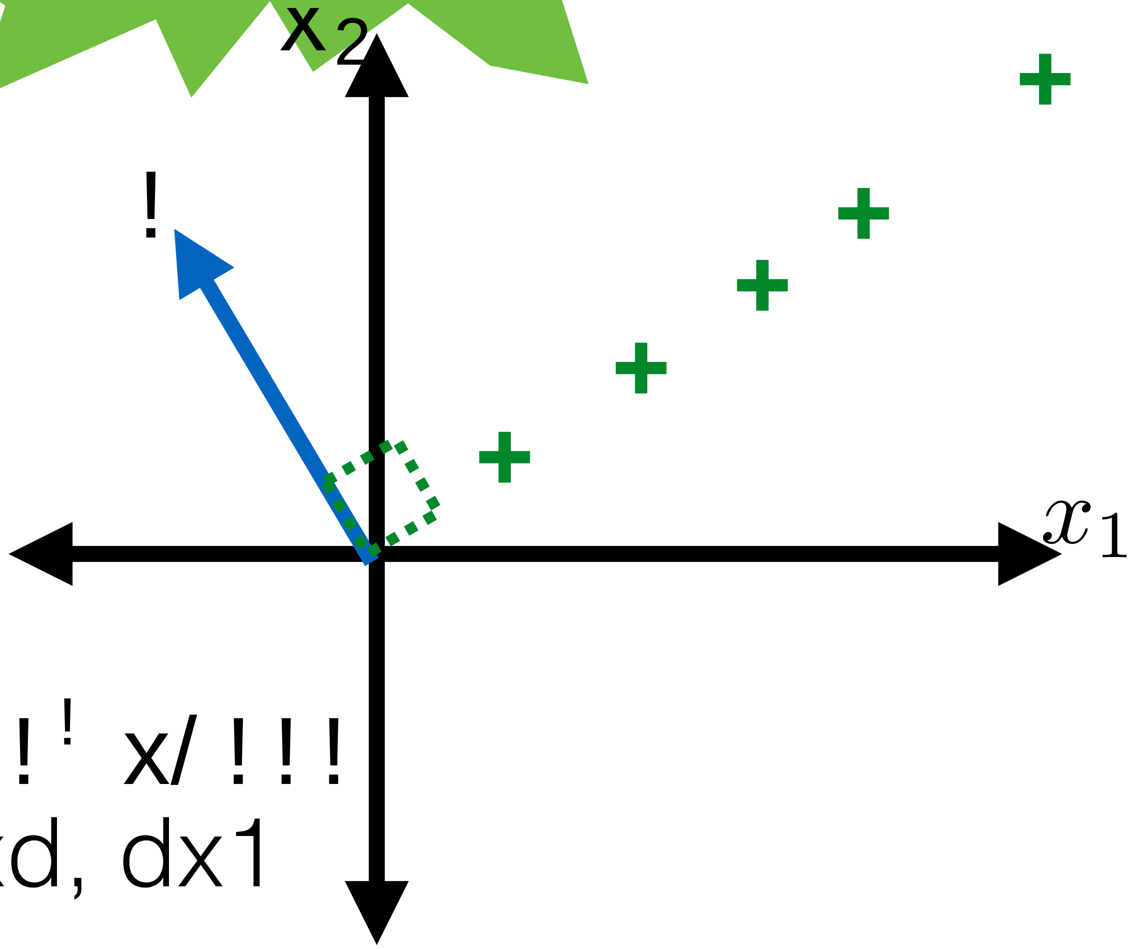


Linear classifiers

- Hypothesis class $\mathcal{H}$: set of **h ! H**
  - Example $\mathcal{H}$: All hypotheses that label +1 on one side of a line and -1 on the other side



Math facts!

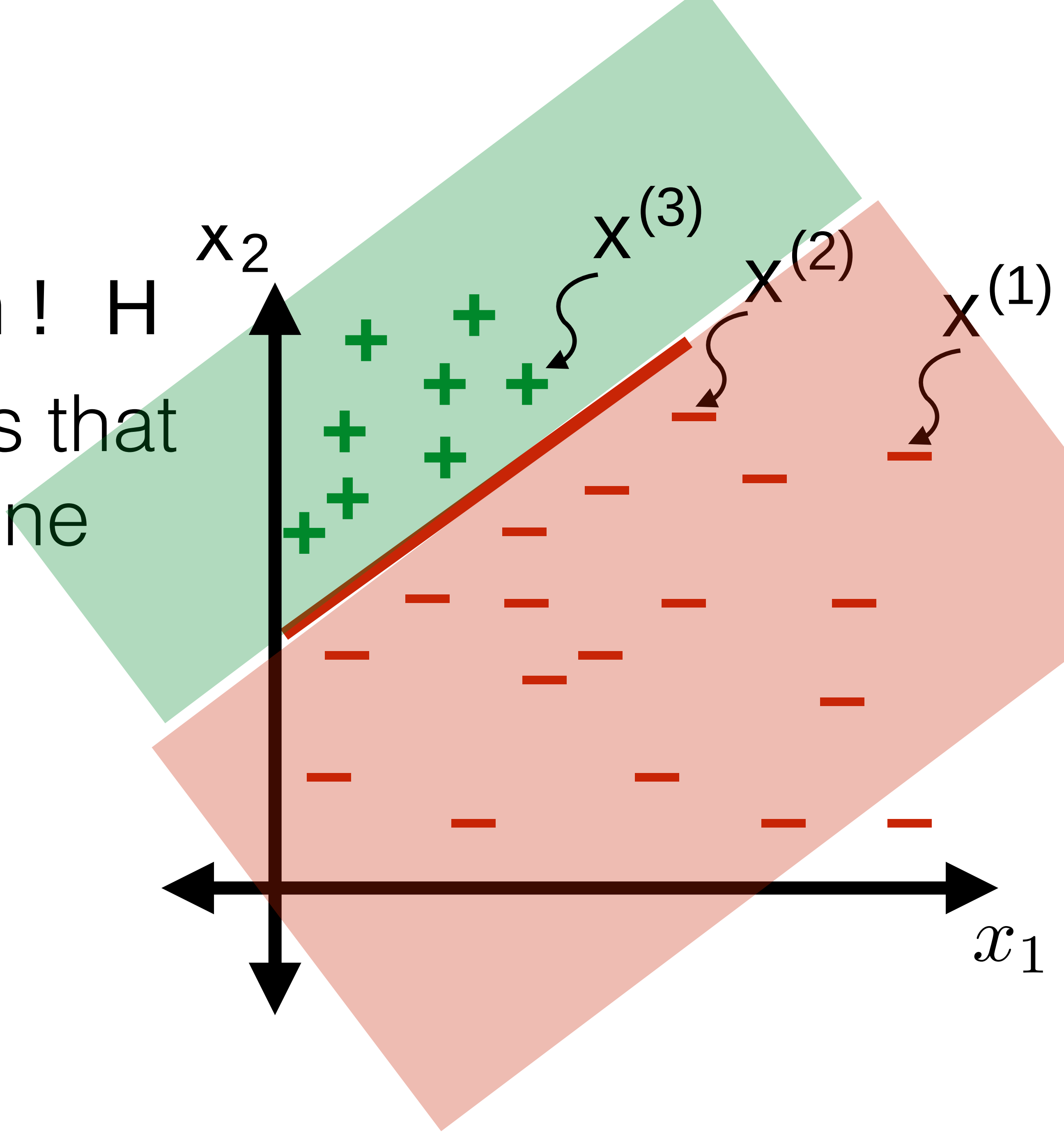


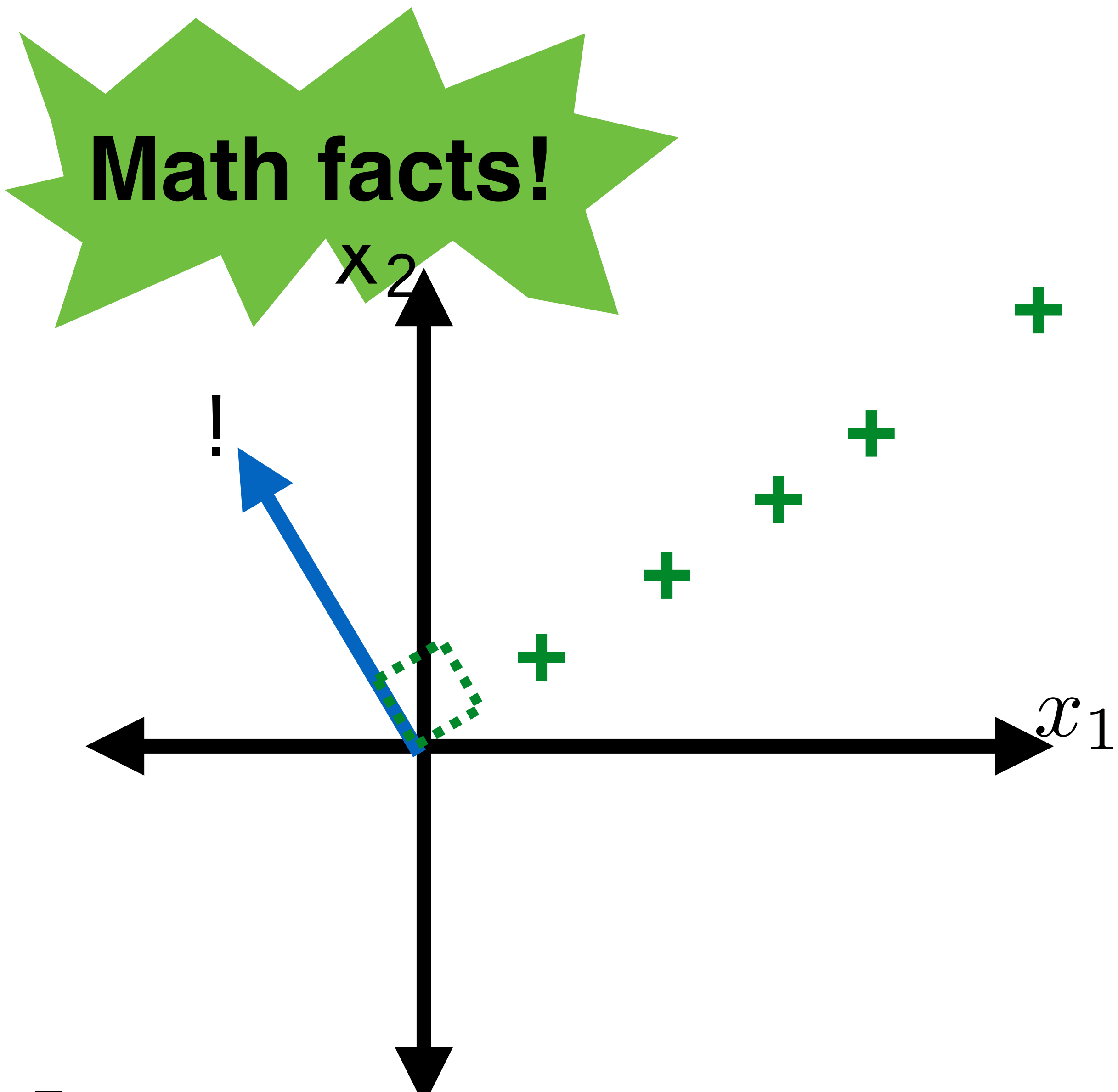
! ! x/ ! ! !

1xd, dx1

Linear classifiers

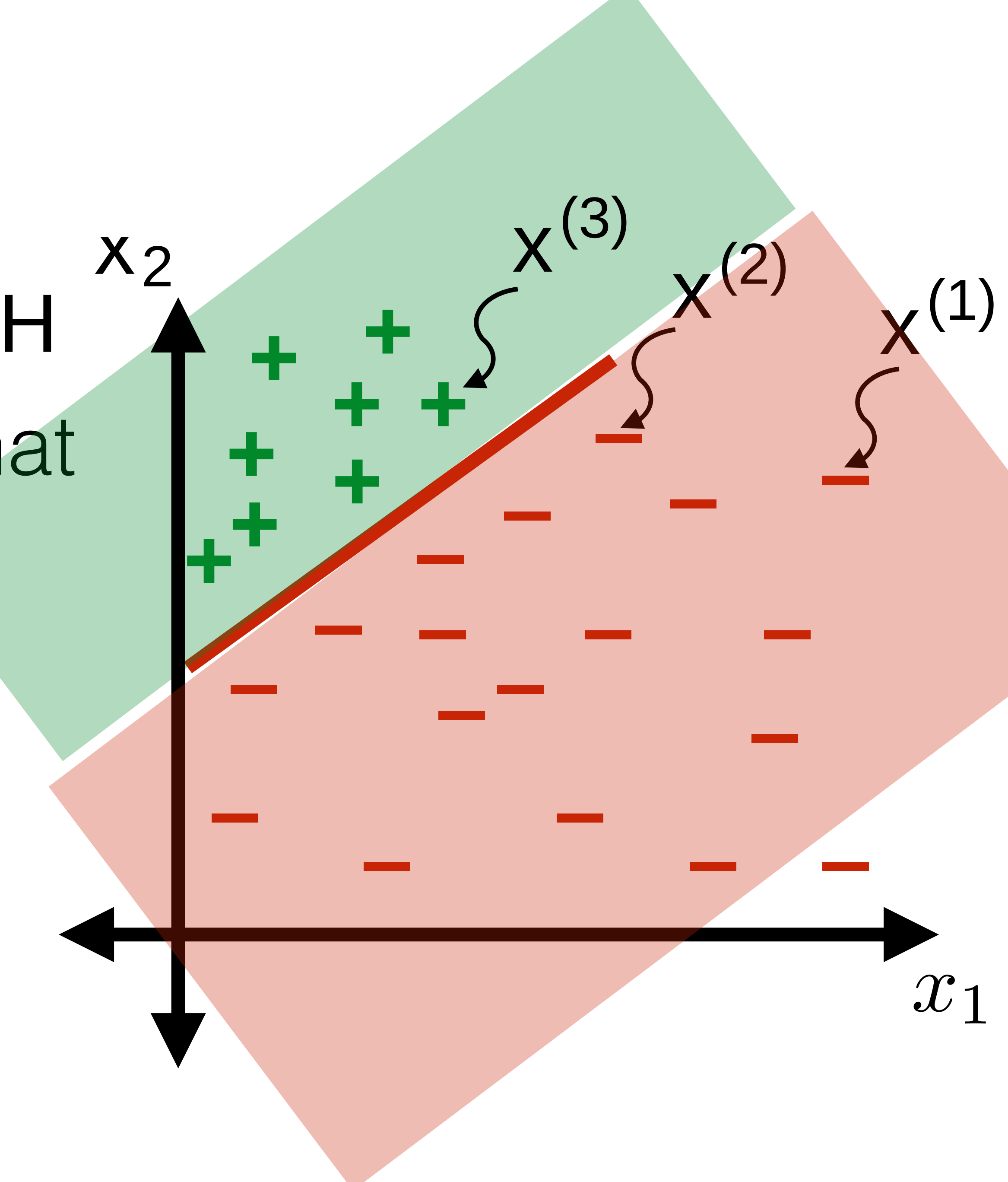
- Hypothesis class $\mathcal{H}$: set of **h ! H**
  - Example $\mathcal{H}$: All hypotheses that label +1 on one side of a line and -1 on the other side

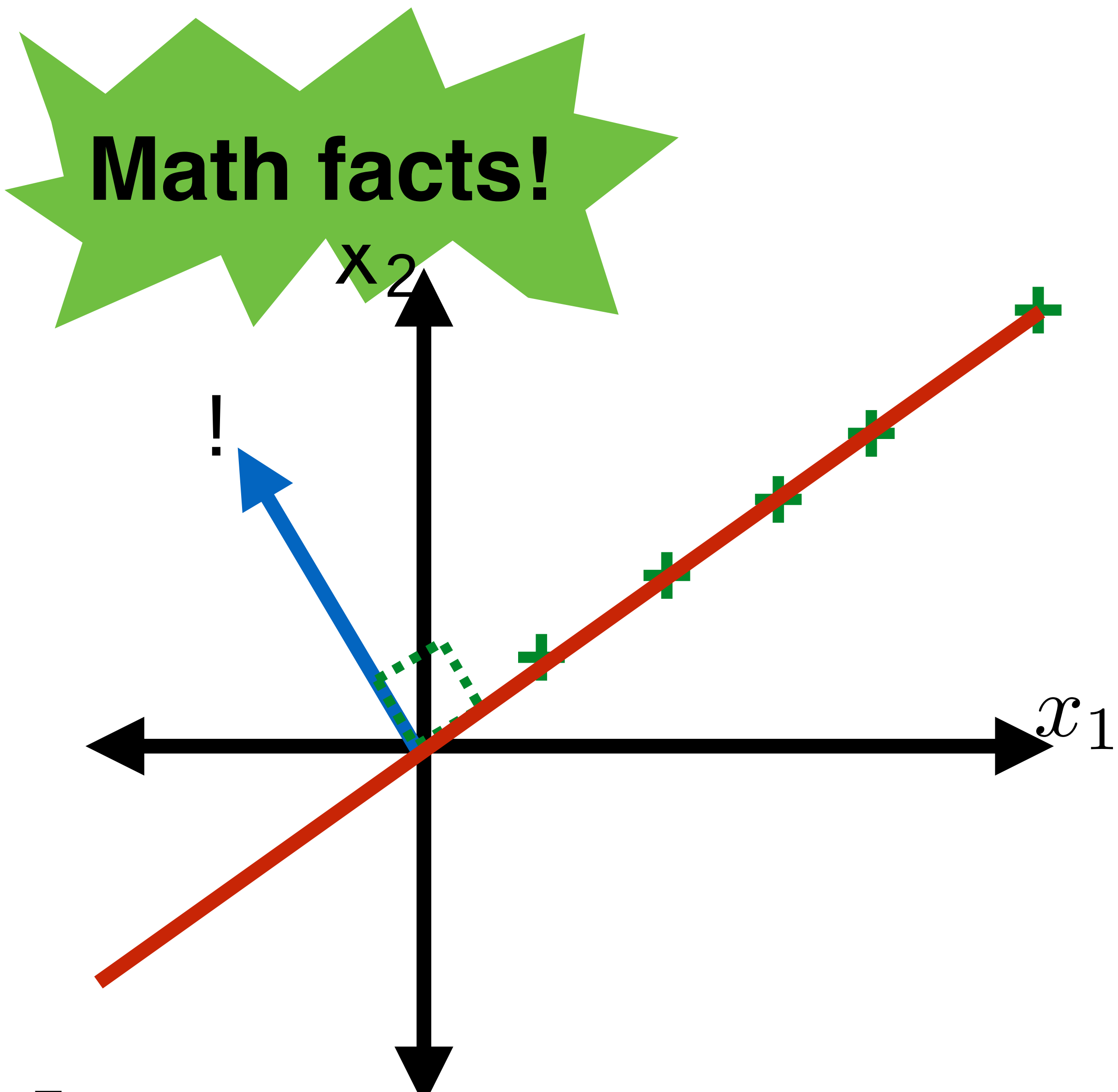




Linear classifiers

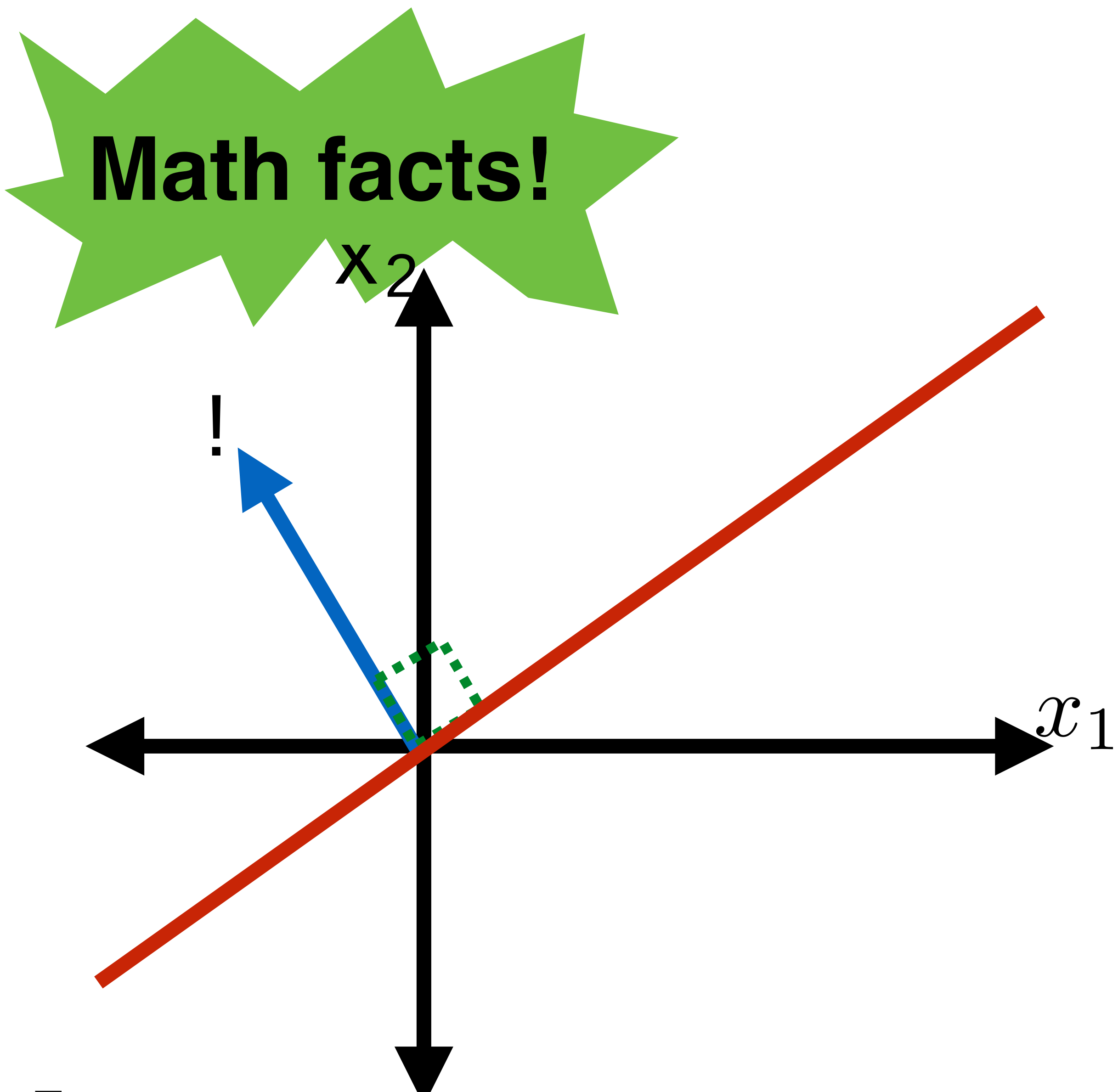
- Hypothesis class $\mathcal{H}$: set of **h** ! **H**
  - Example $\mathcal{H}$: All hypotheses that label +1 on one side of a line and -1 on the other side

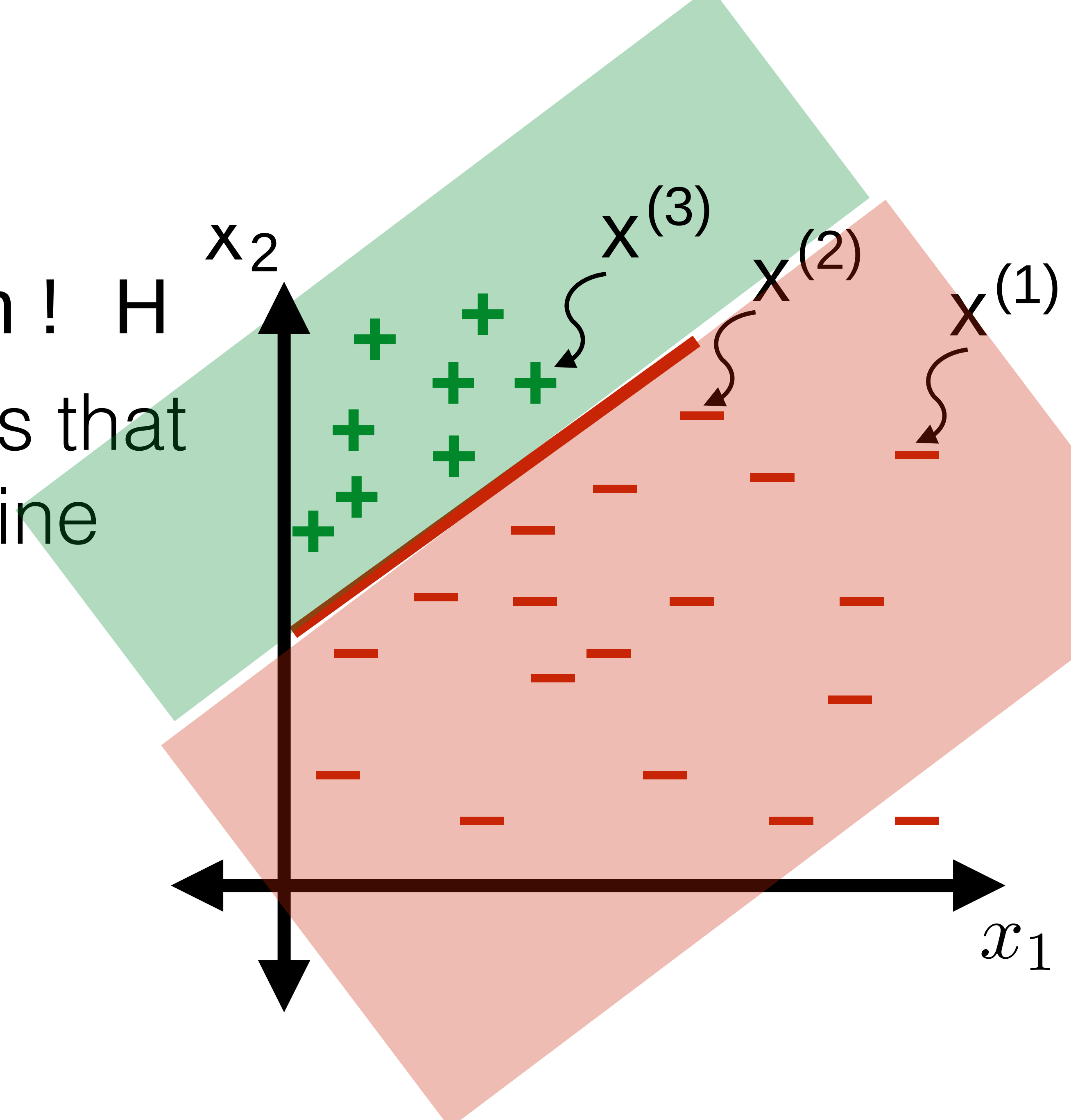




Linear classifiers

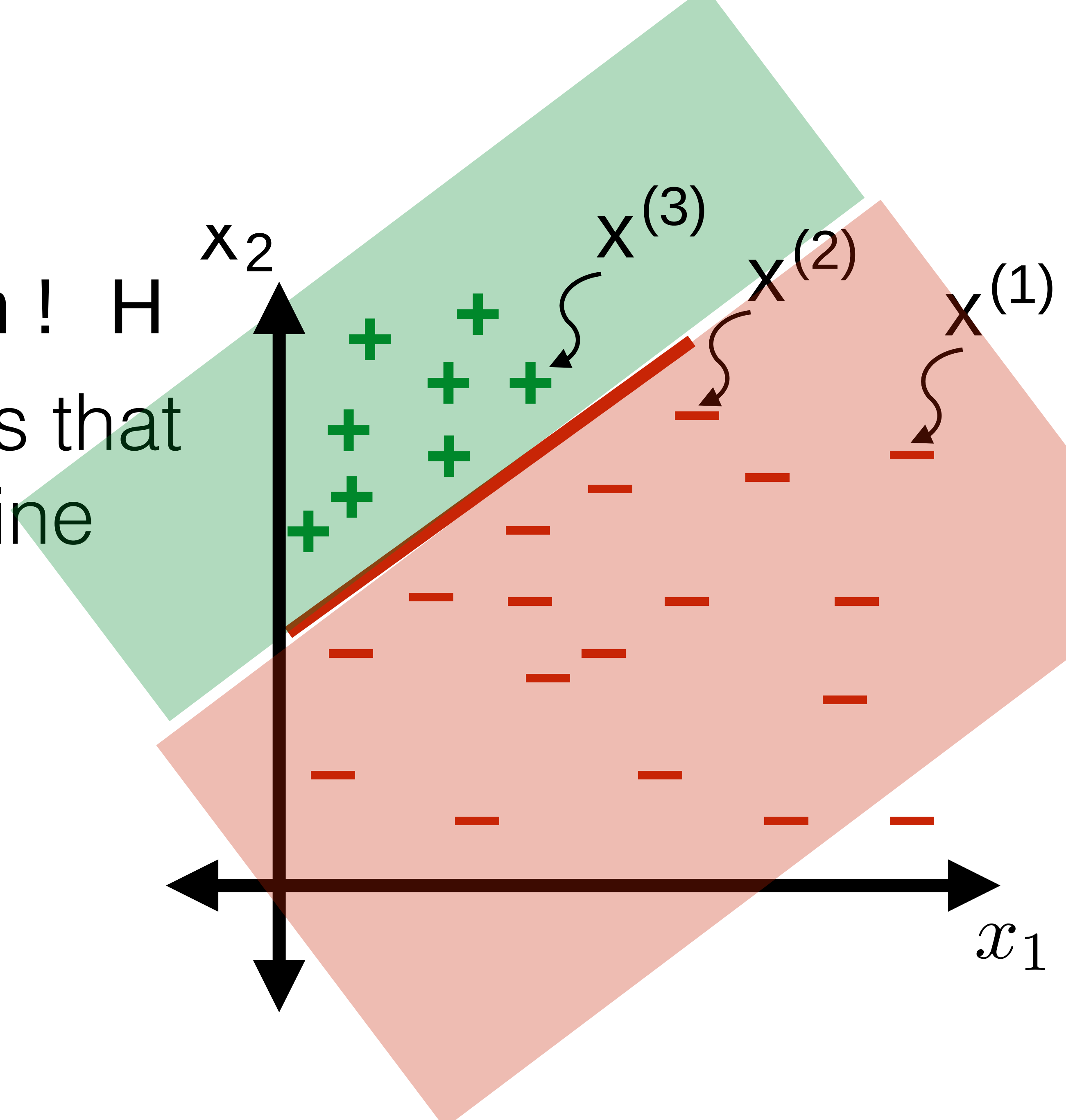
- Hypothesis class $\mathcal{H}$: set of **h ! H**
    - Example $\mathcal{H}$: All hypotheses that label +1 on one side of a line and -1 on the other side

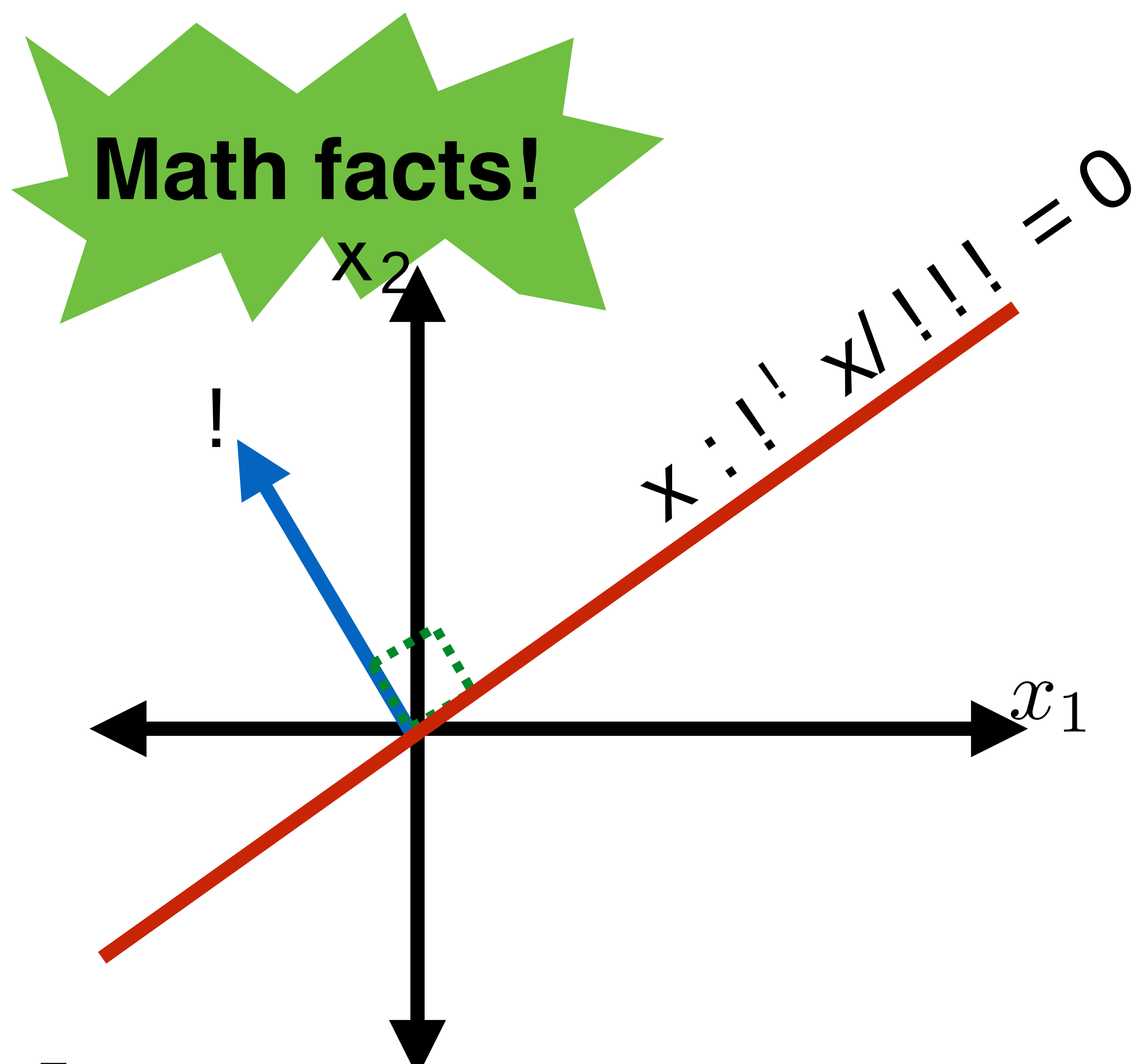




Linear classifiers

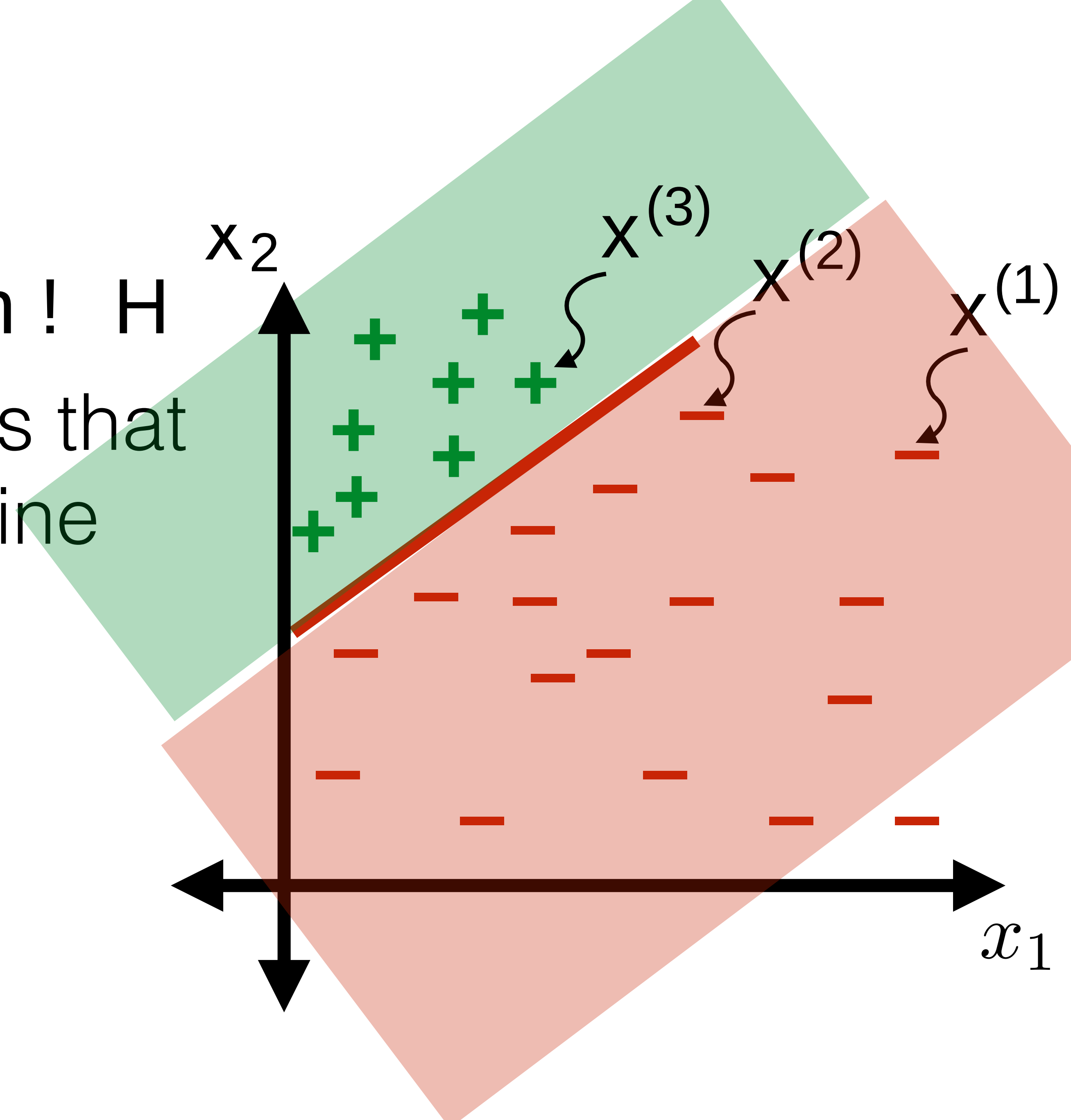
- Hypothesis class $\mathcal{H}$: set of **h ! H**
  - Example $\mathcal{H}$: All hypotheses that label +1 on one side of a line and -1 on the other side

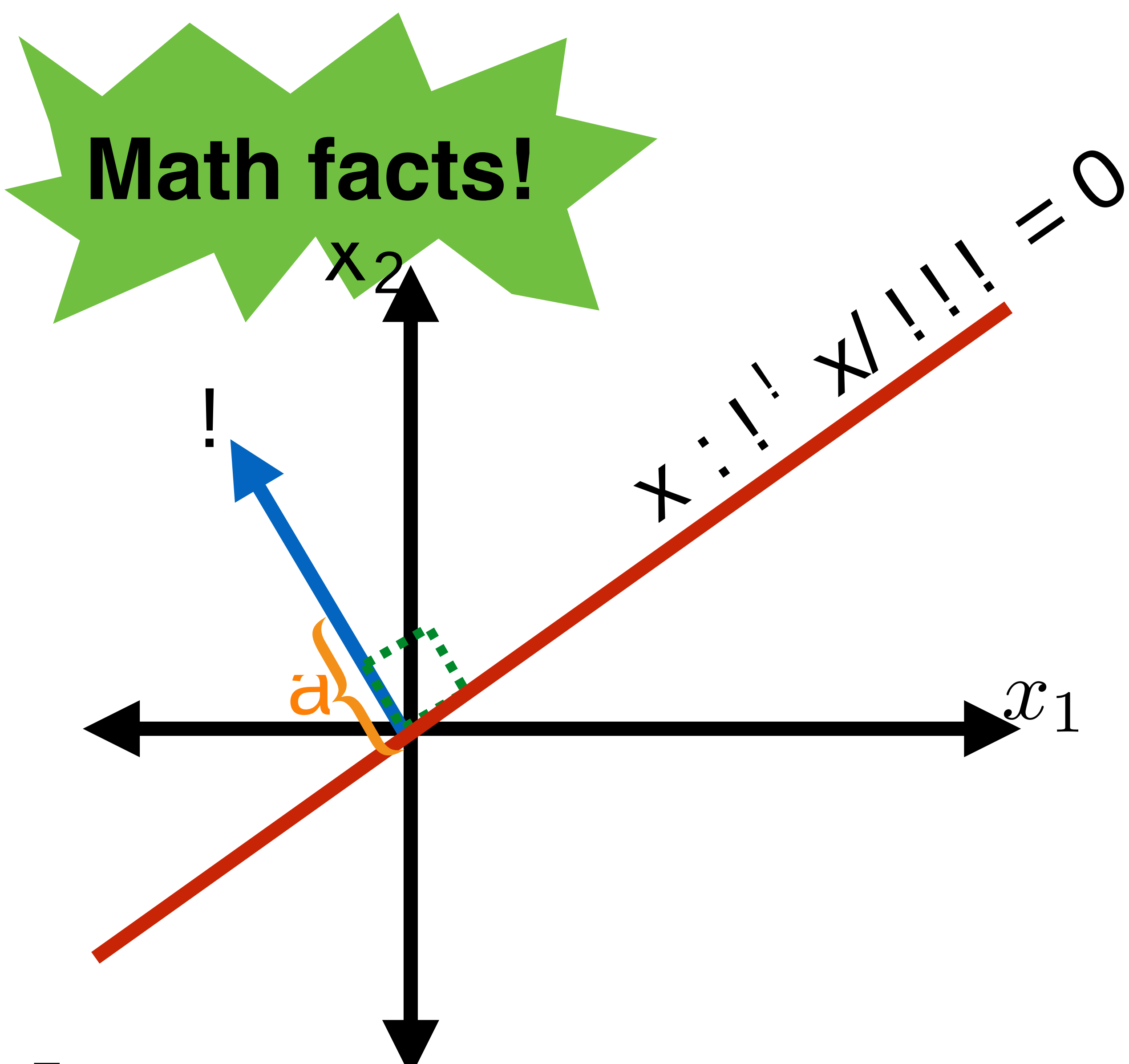




Linear classifiers

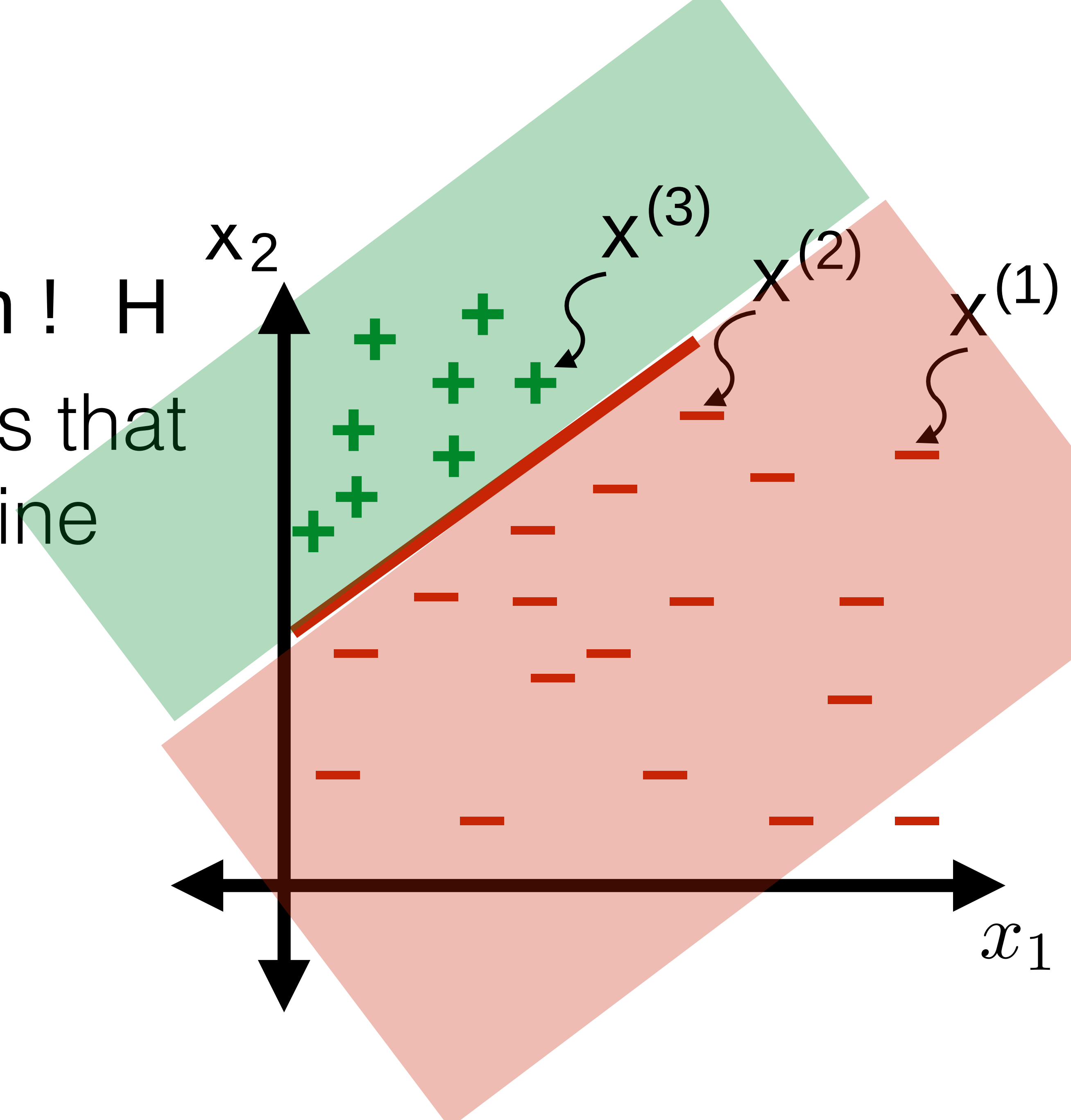
- Hypothesis class $\mathcal{H}$: set of **h ! H**
  - Example $\mathcal{H}$: All hypotheses that label +1 on one side of a line and -1 on the other side



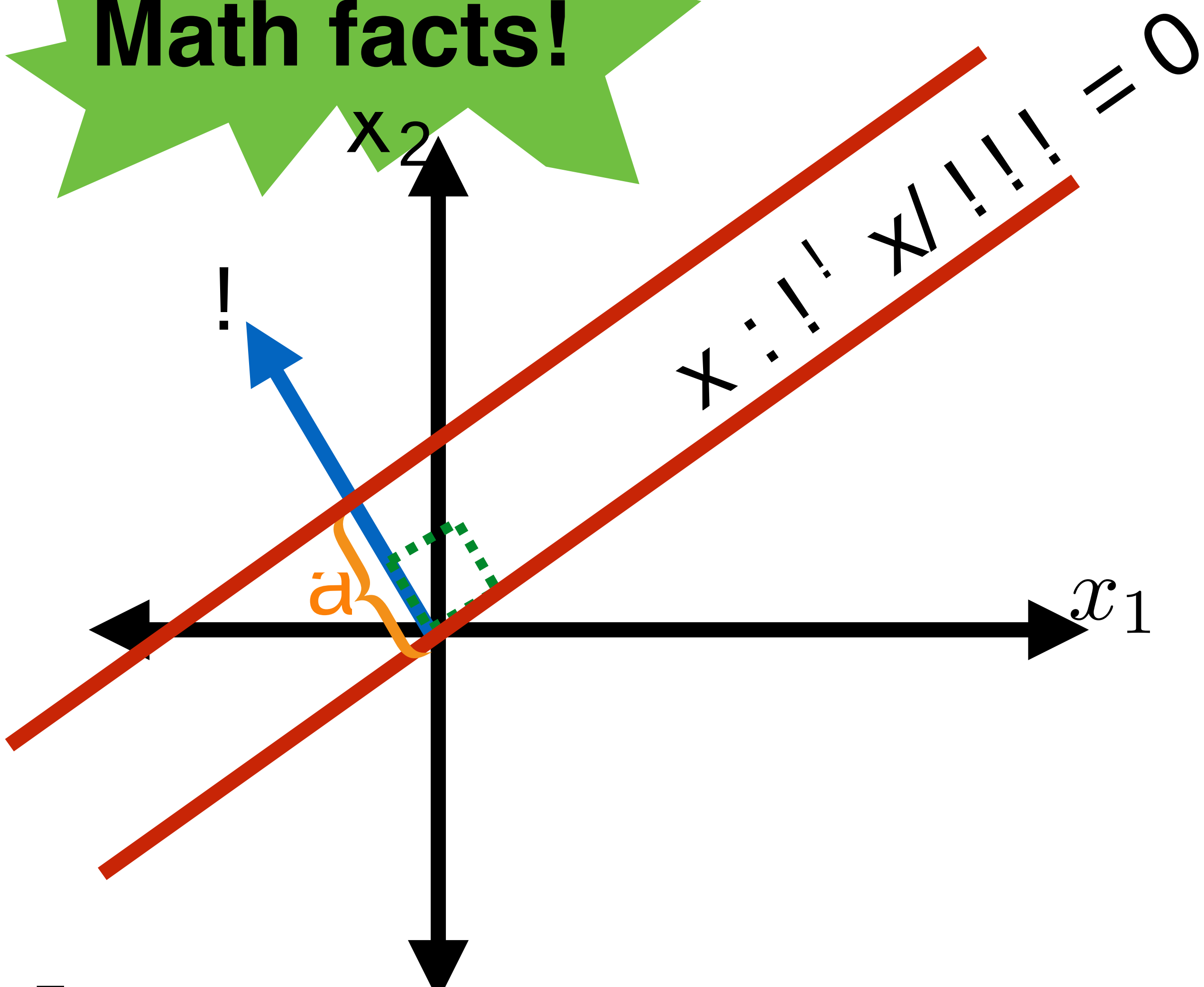


Linear classifiers

- Hypothesis class $\mathcal{H}$: set of **h ! H**
    - Example $\mathcal{H}$: All hypotheses that label +1 on one side of a line and -1 on the other side

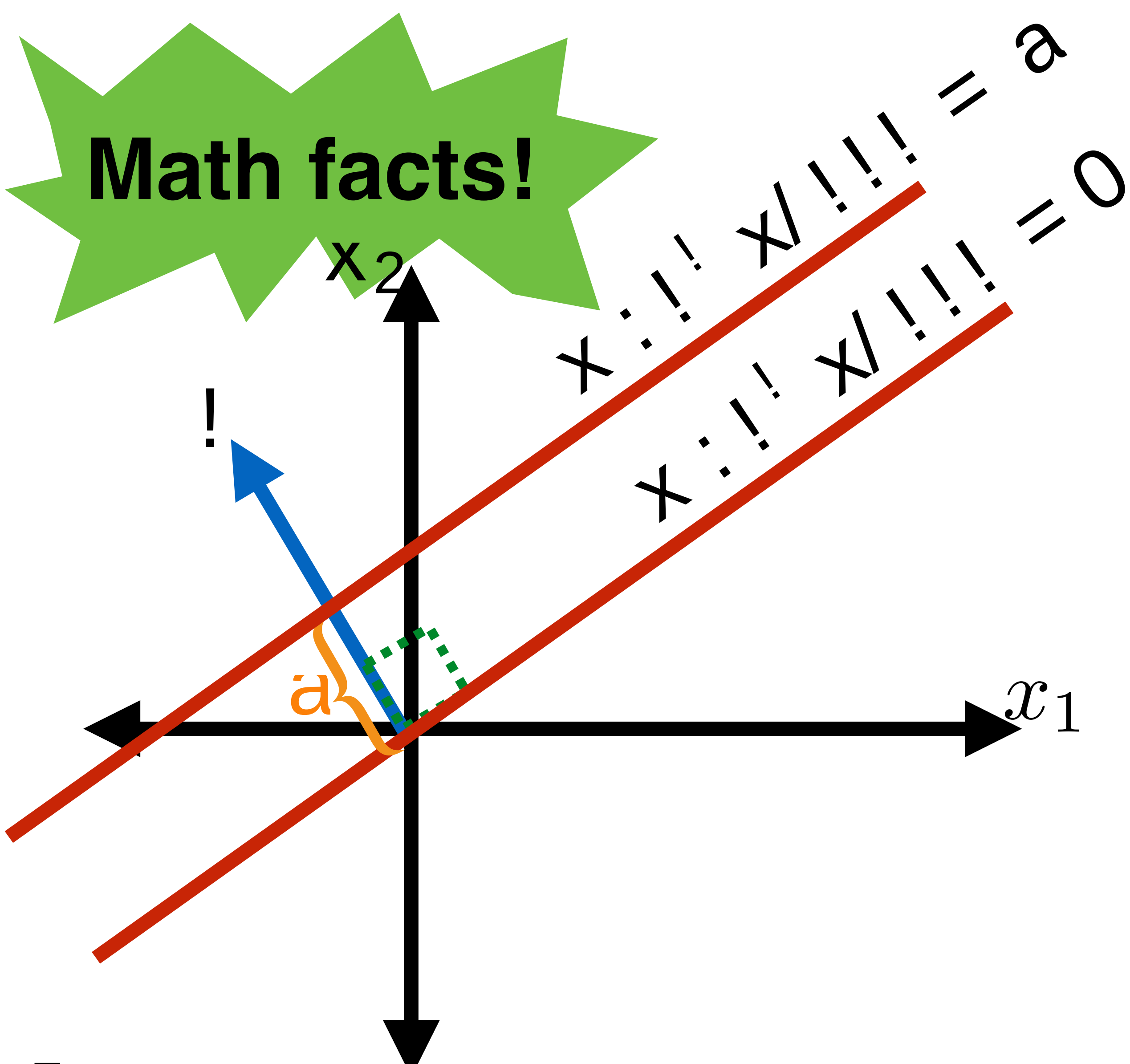


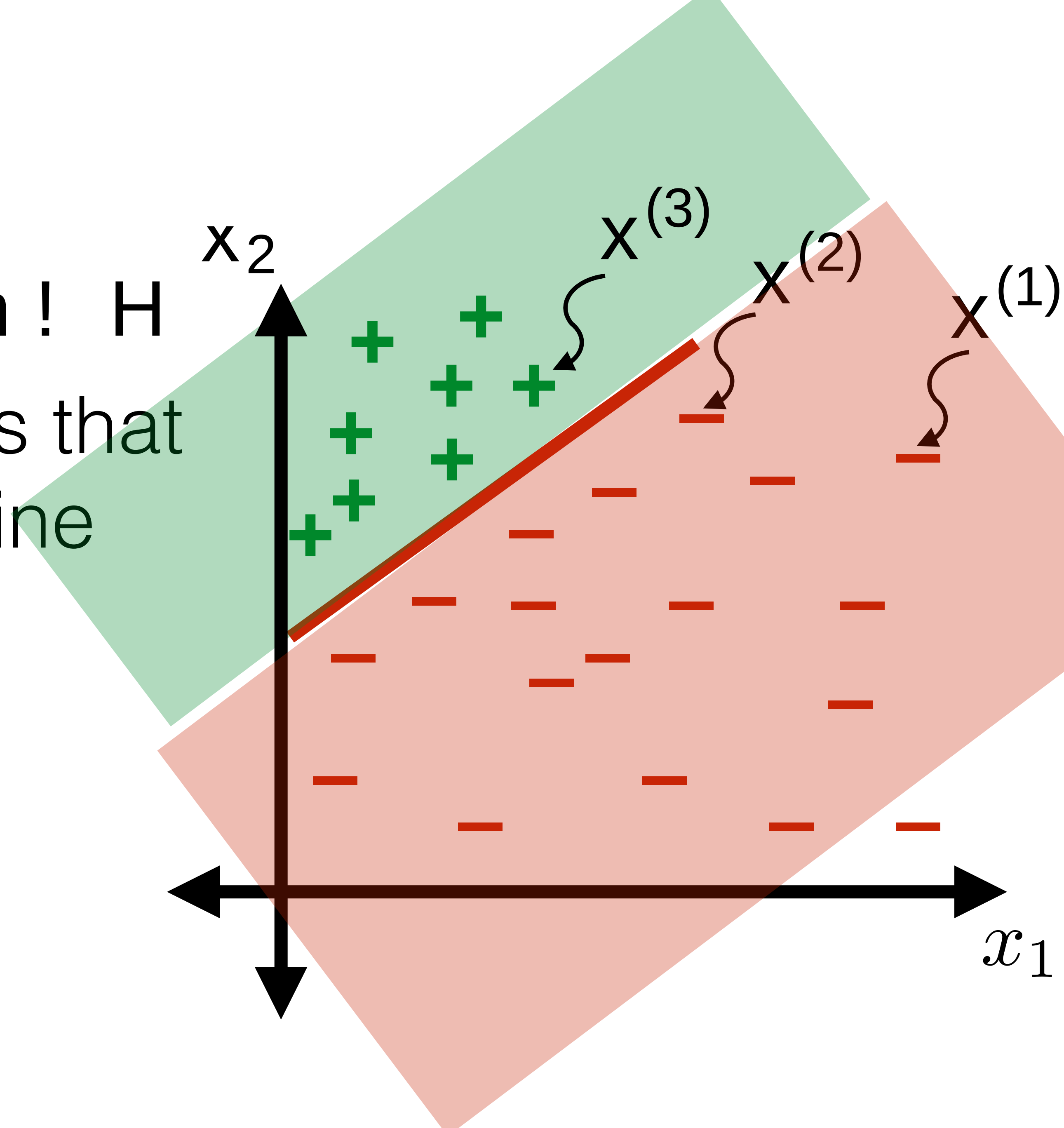
Math facts!



Linear classifiers

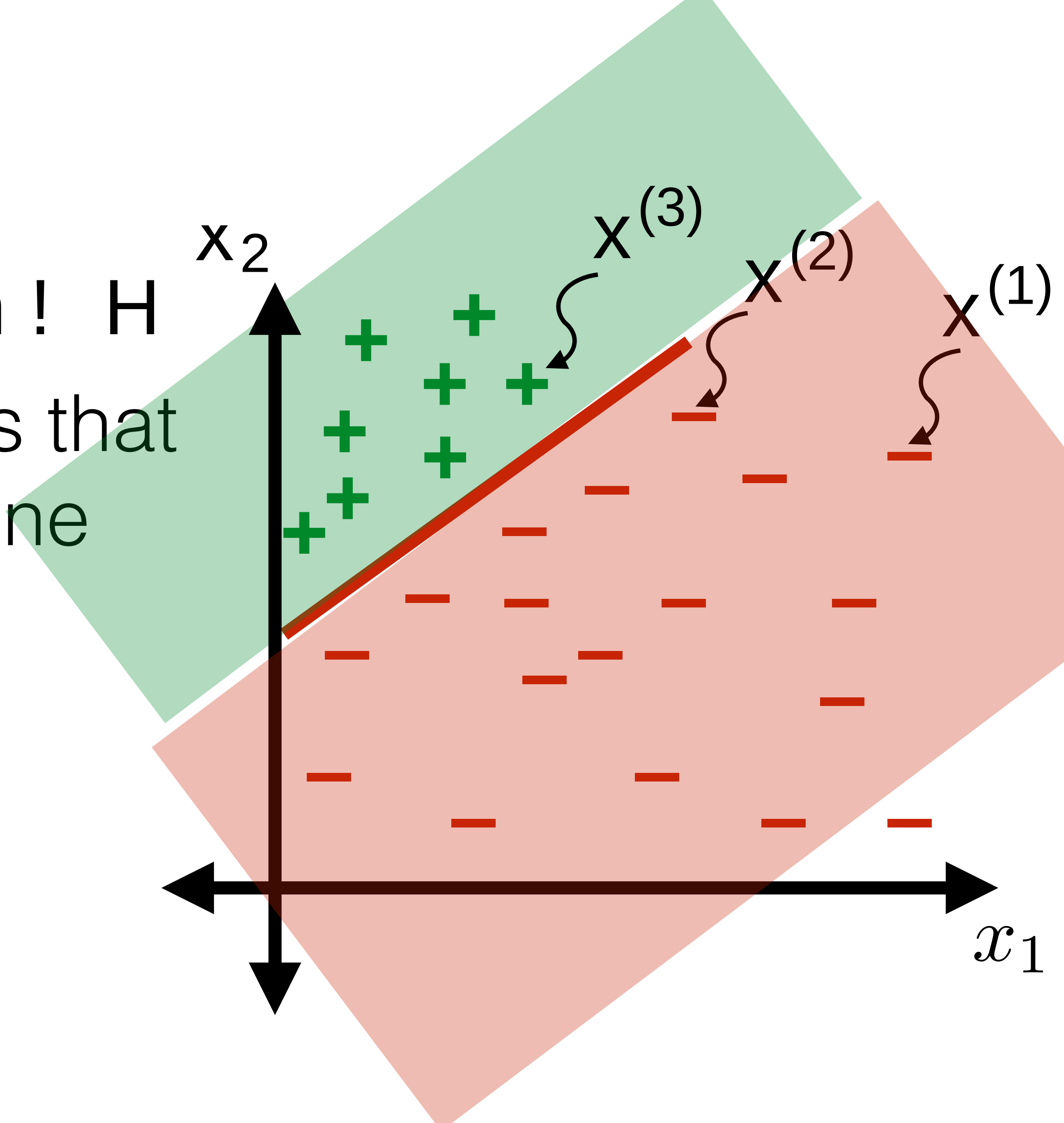
- Hypothesis class $\mathcal{H}$: set of **h ! H**
    - Example $\mathcal{H}$: All hypotheses that label +1 on one side of a line and -1 on the other side

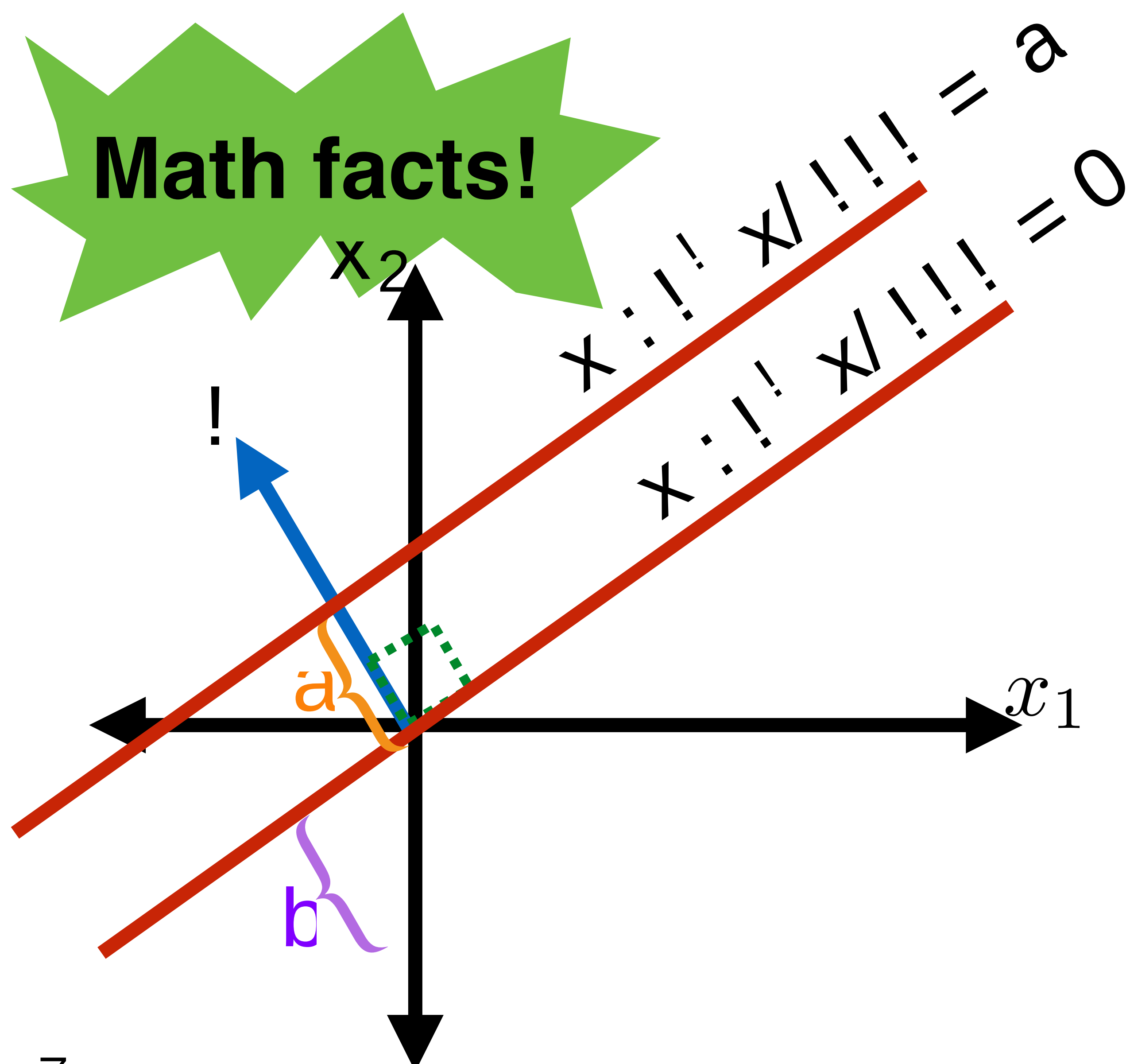




Linear classifiers

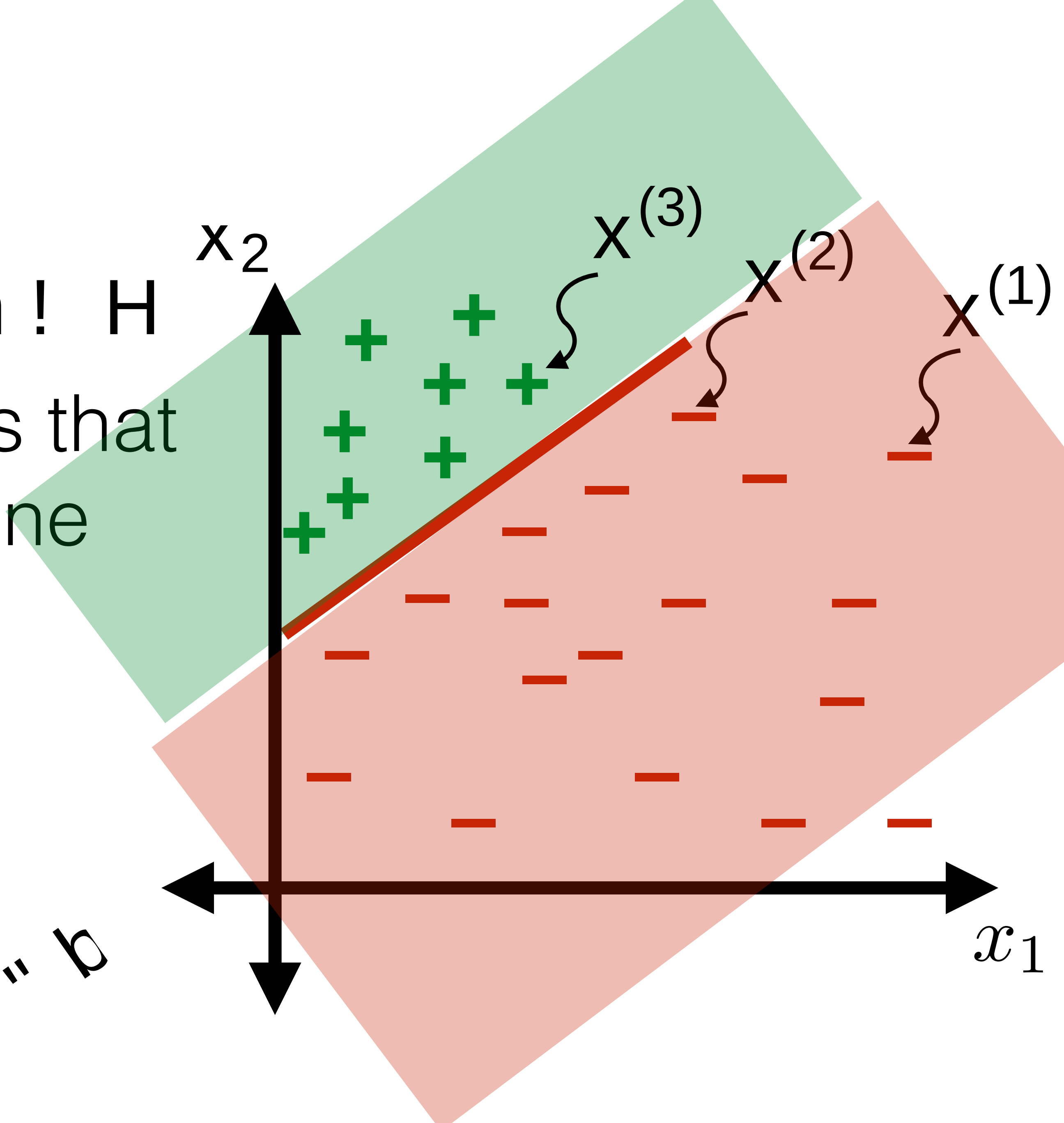
- Hypothesis class $\mathcal{H}$: set of **h ! H**
  - Example $\mathcal{H}$: All hypotheses that label +1 on one side of a line and -1 on the other side

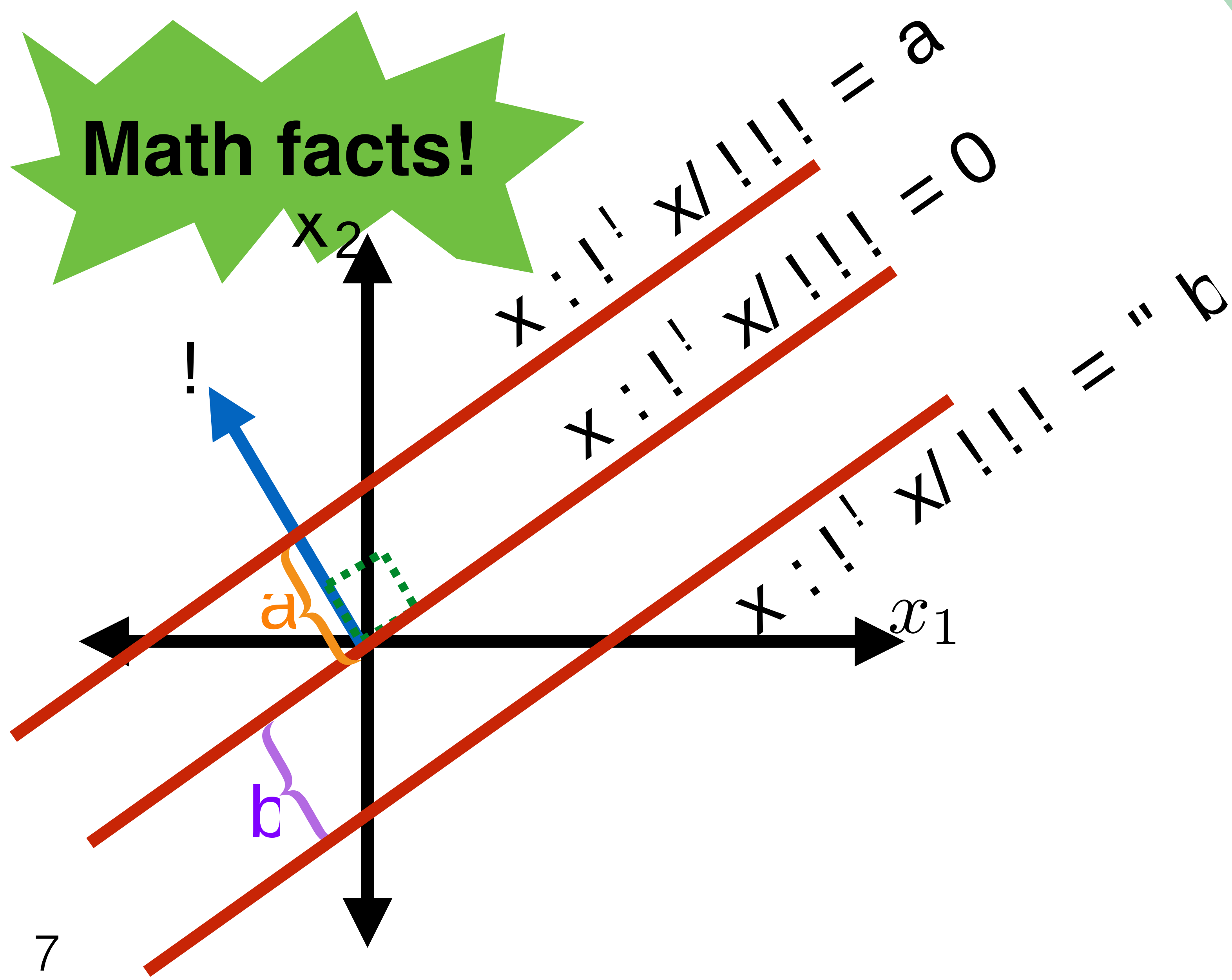




Linear classifiers

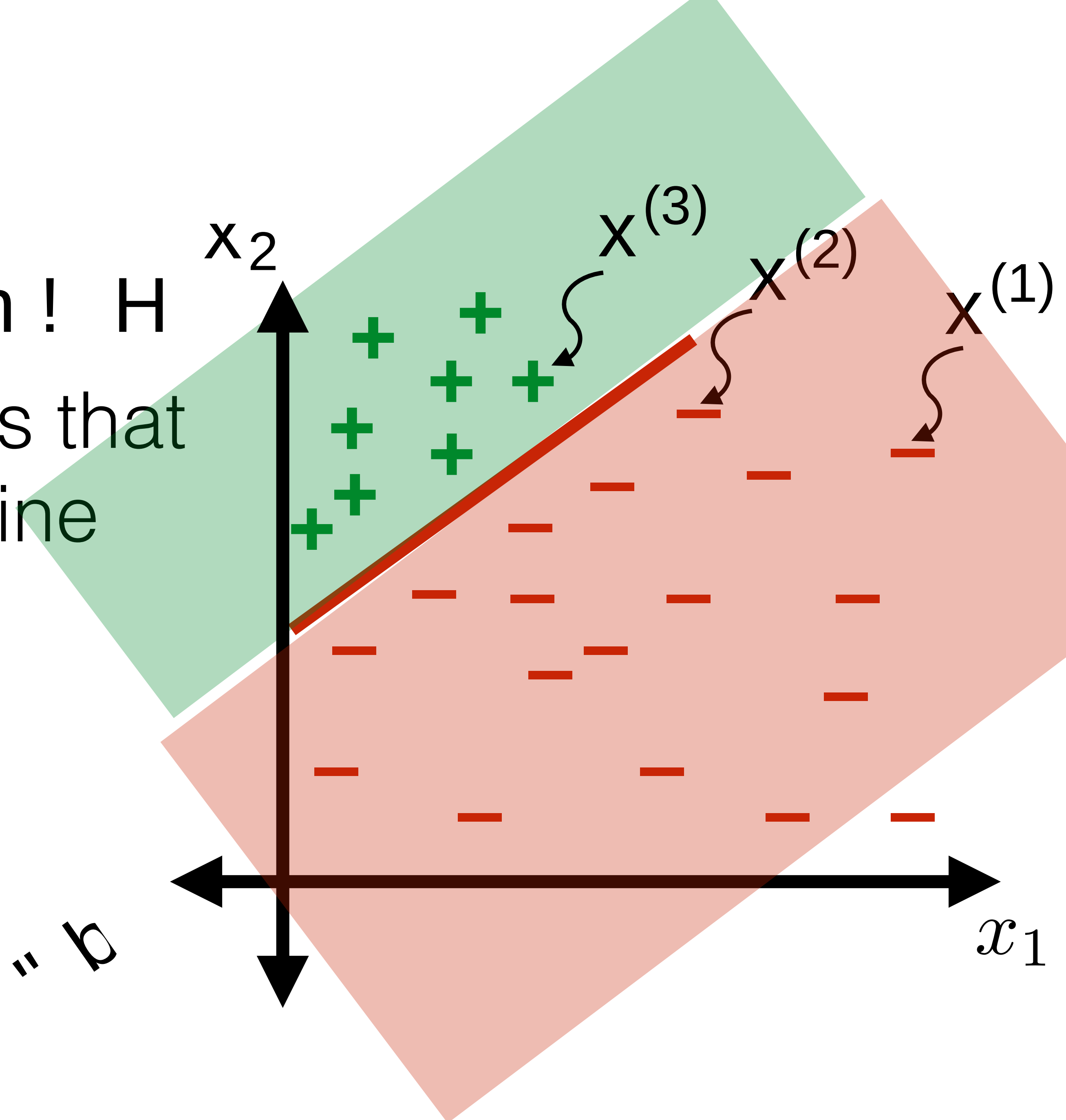
- Hypothesis class $\mathcal{H}$: set of **h ! H**
  - Example $\mathcal{H}$: All hypotheses that label +1 on one side of a line and -1 on the other side

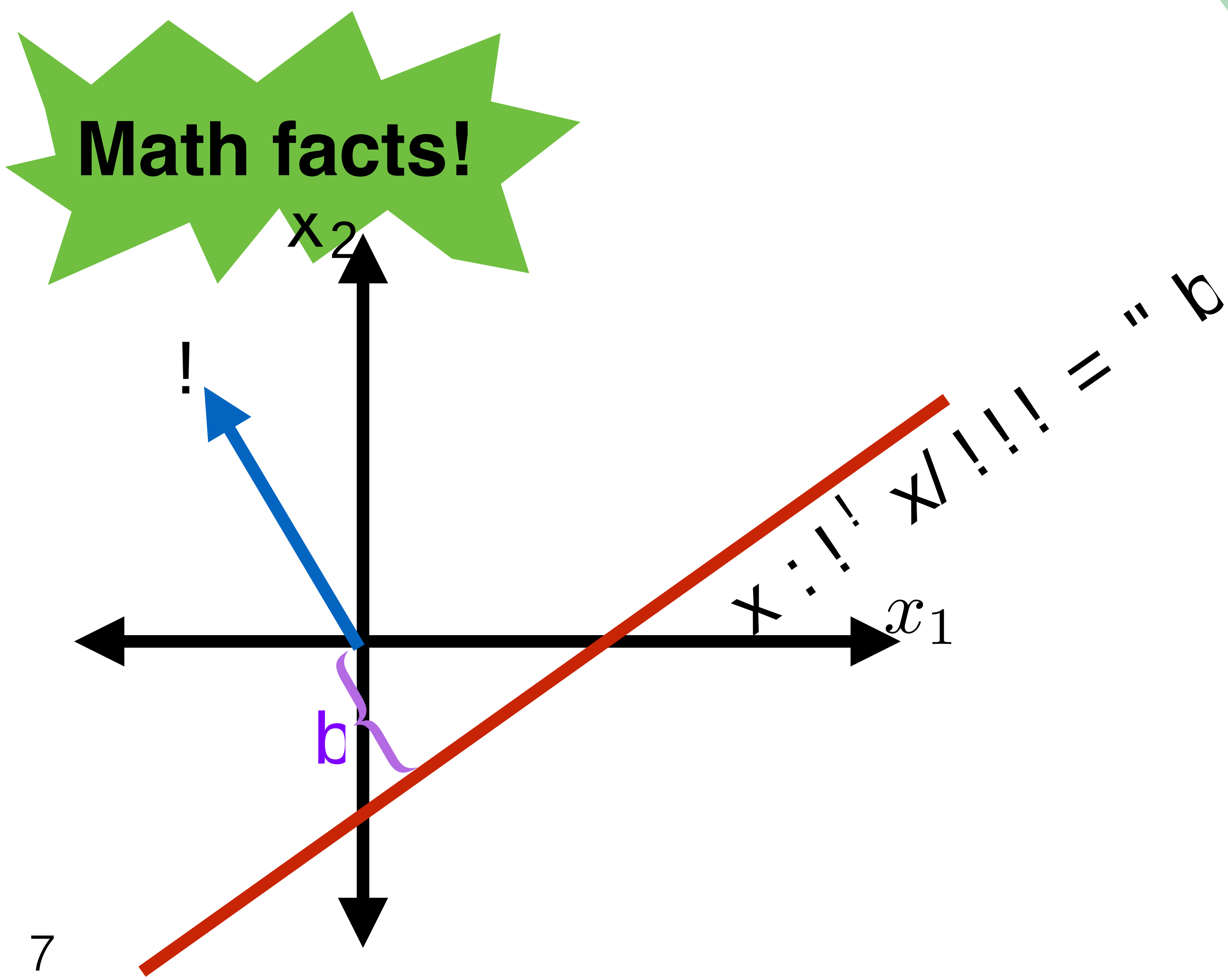




Linear classifiers

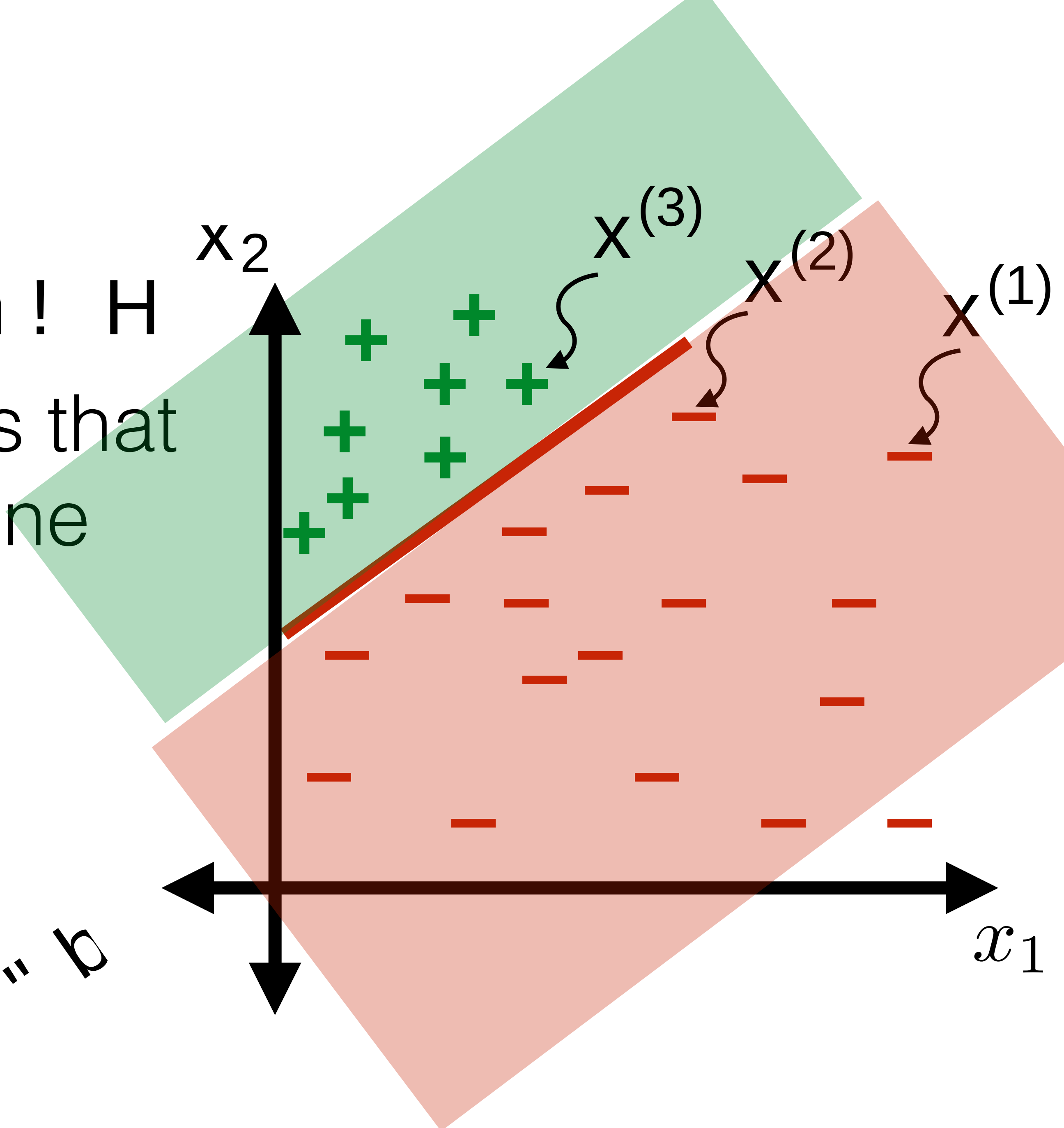
- Hypothesis class $\mathcal{H}$: set of **h ! H**
  - Example $\mathcal{H}$: All hypotheses that label +1 on one side of a line and -1 on the other side

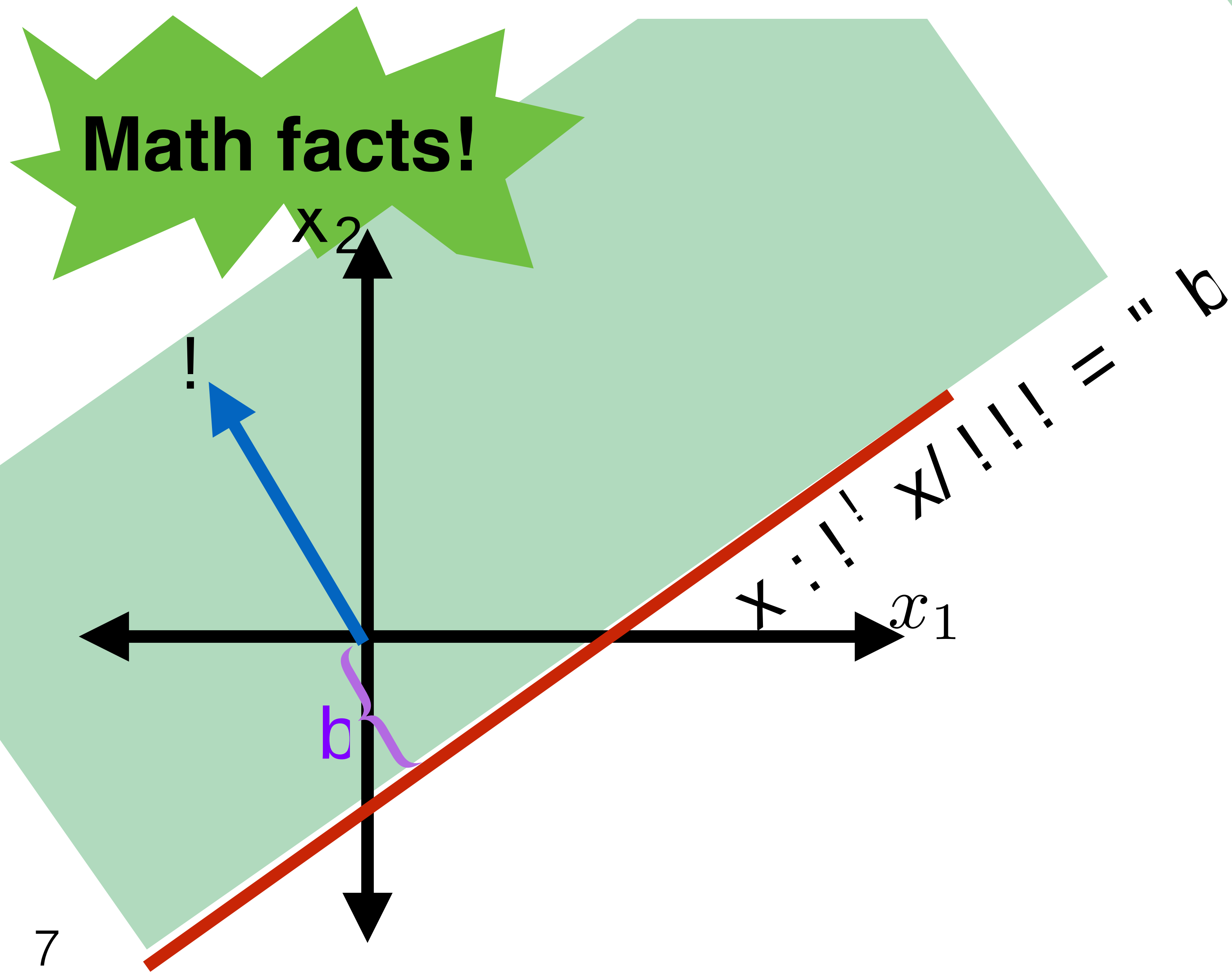




Linear classifiers

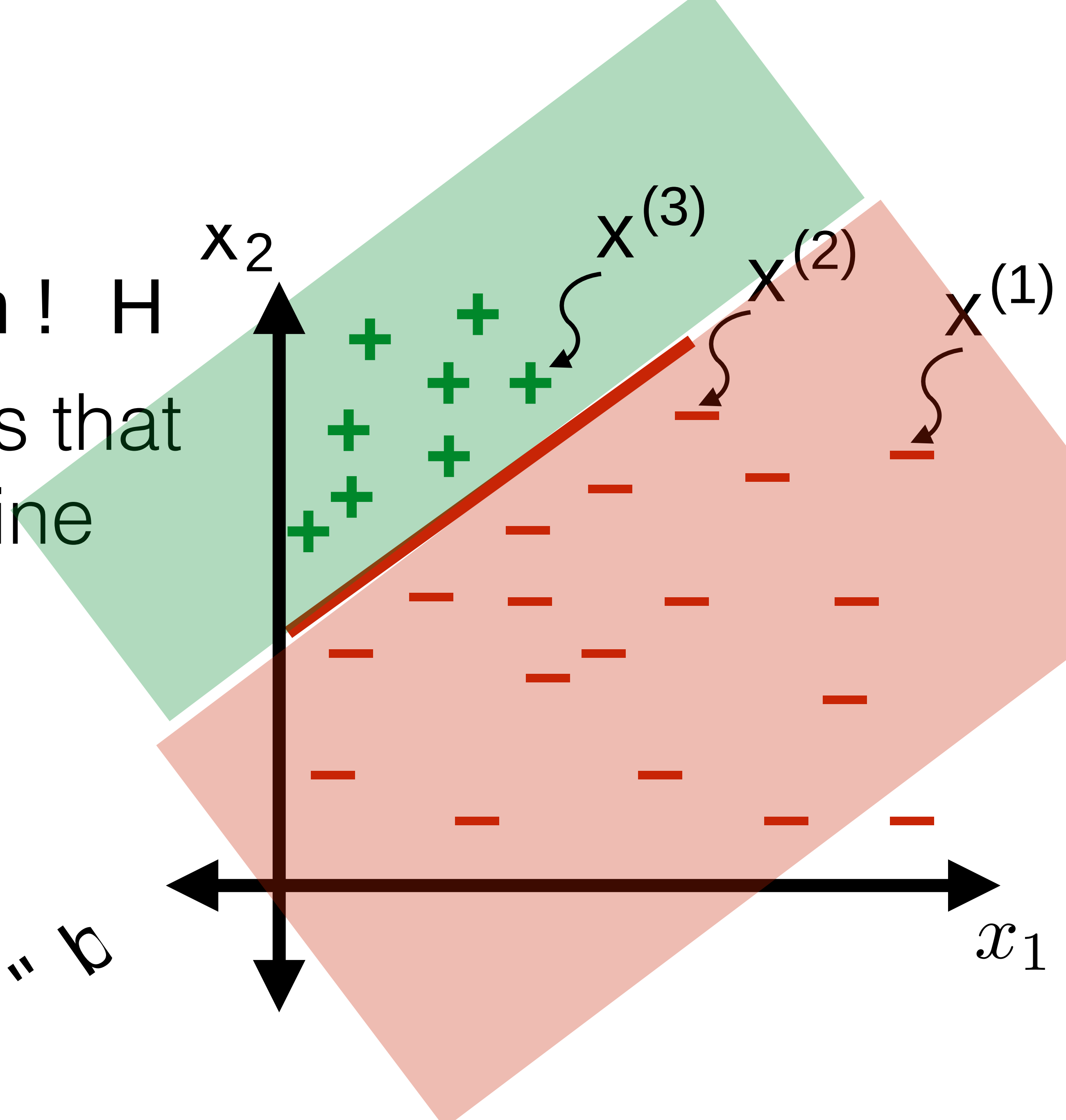
- Hypothesis class $\mathcal{H}$: set of **h ! H**
  - Example $\mathcal{H}$: All hypotheses that label +1 on one side of a line and -1 on the other side

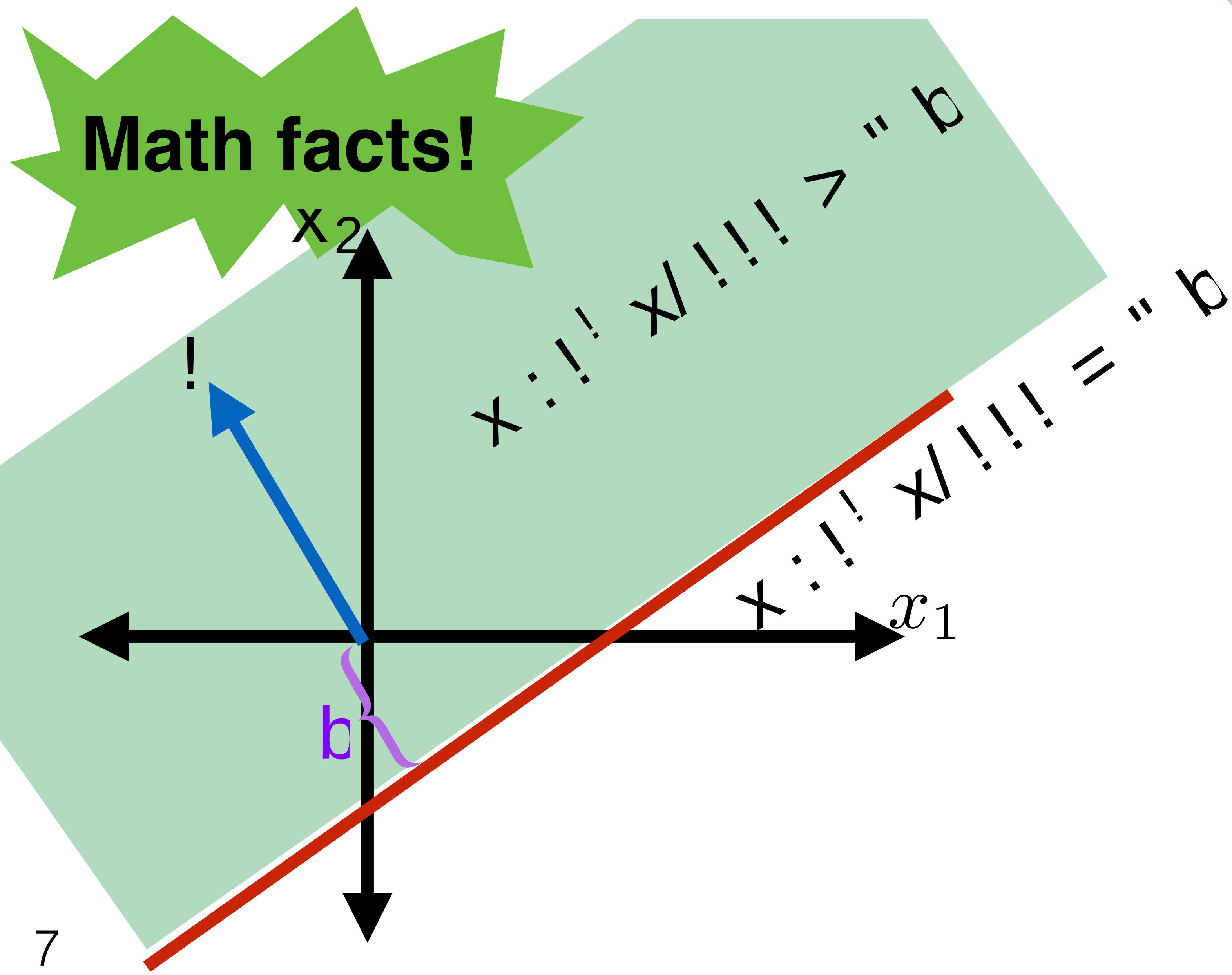




Linear classifiers

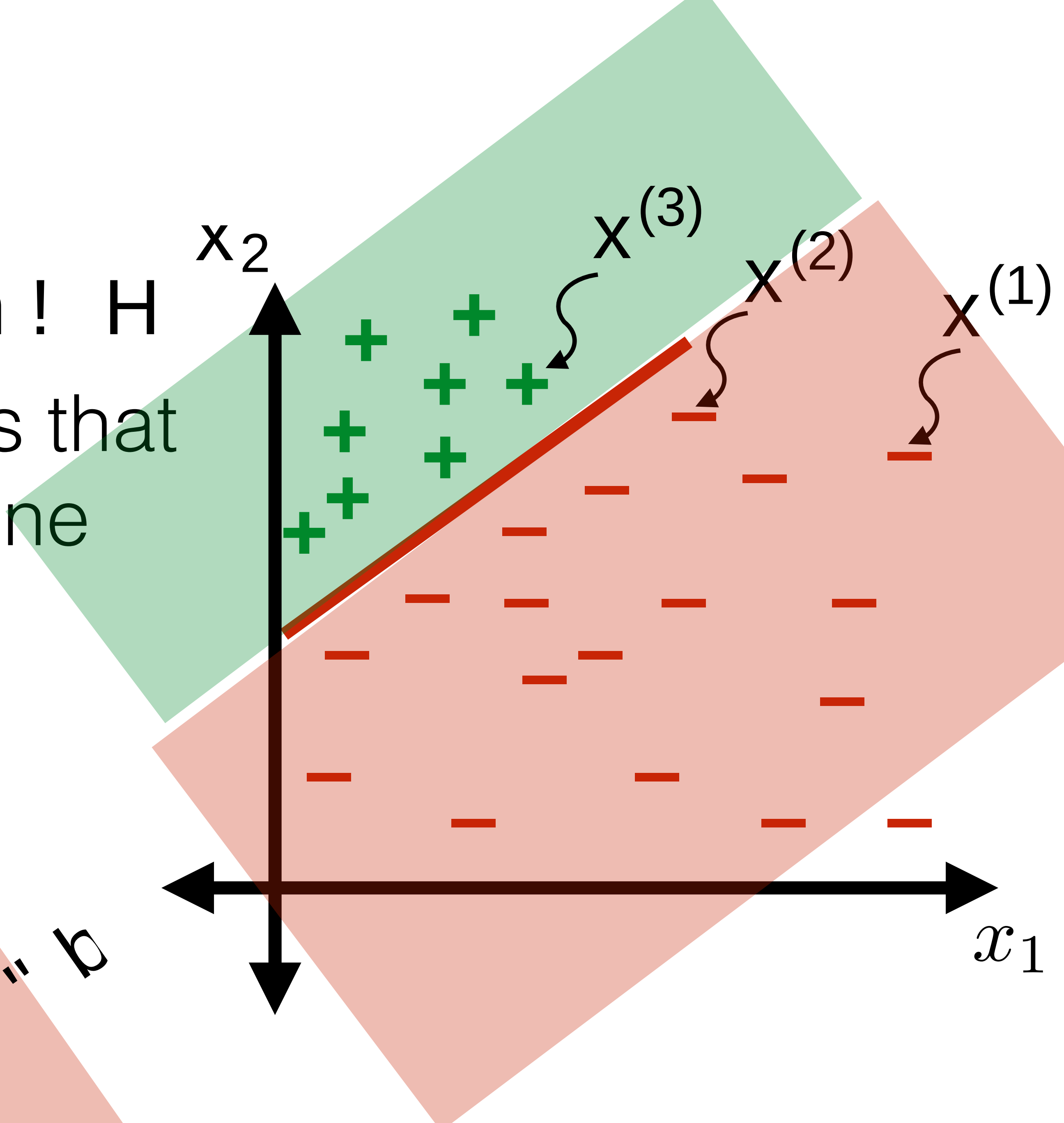
- Hypothesis class $\mathcal{H}$: set of **h** ! **H**
  - Example $\mathcal{H}$: All hypotheses that label +1 on one side of a line and -1 on the other side

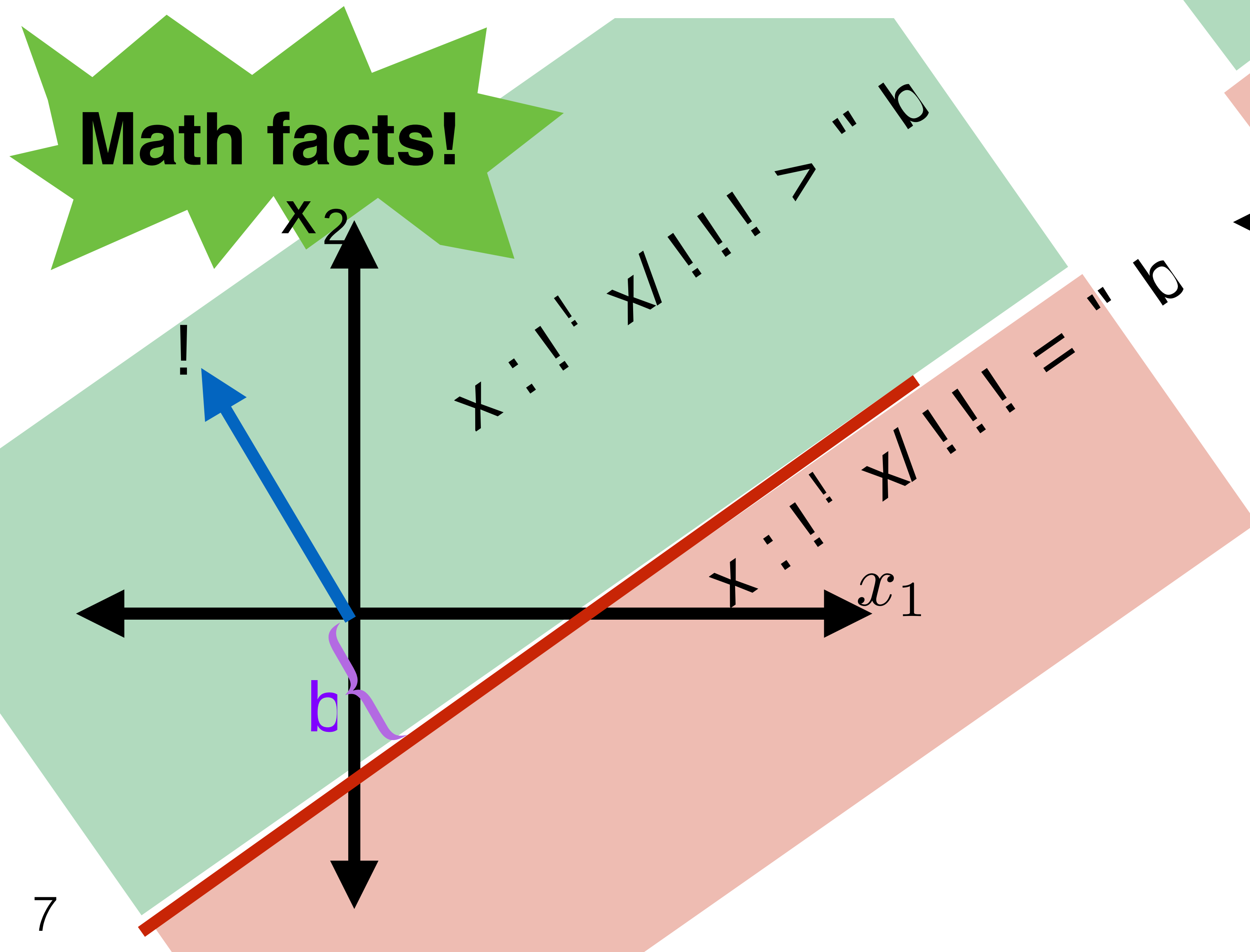




Linear classifiers

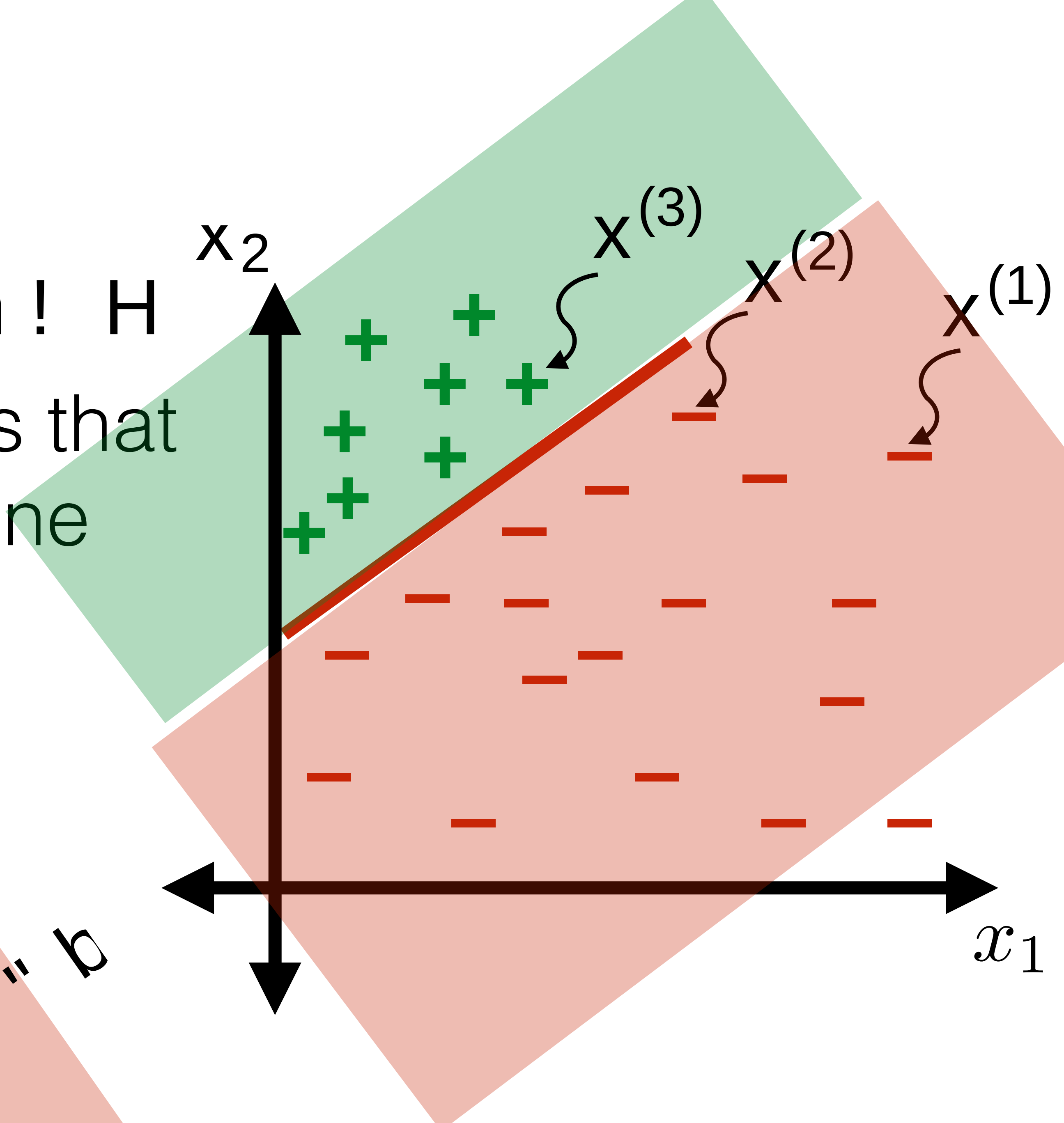
- Hypothesis class $\mathcal{H}$: set of **h ! H**
  - Example $\mathcal{H}$: All hypotheses that label +1 on one side of a line and -1 on the other side

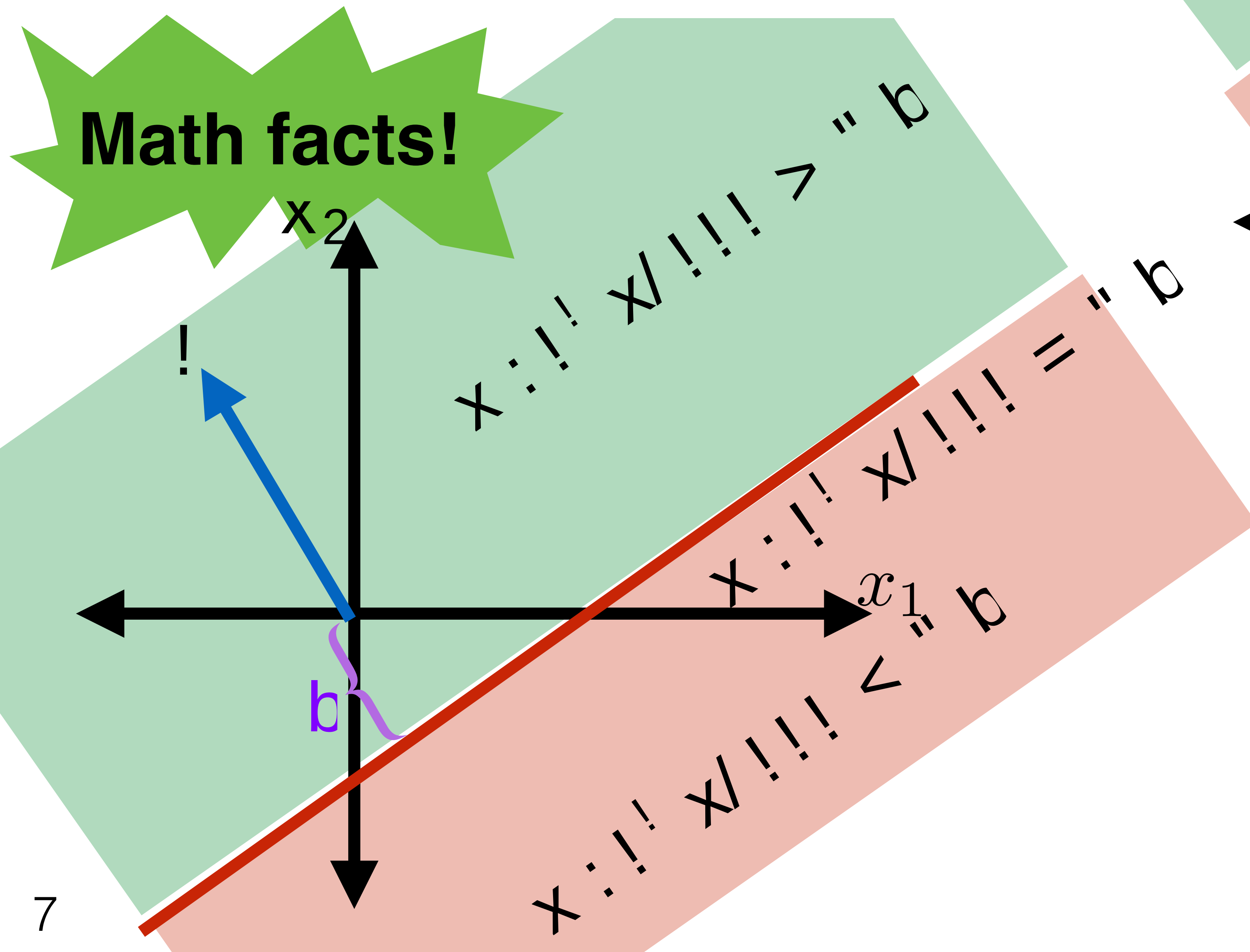




Linear classifiers

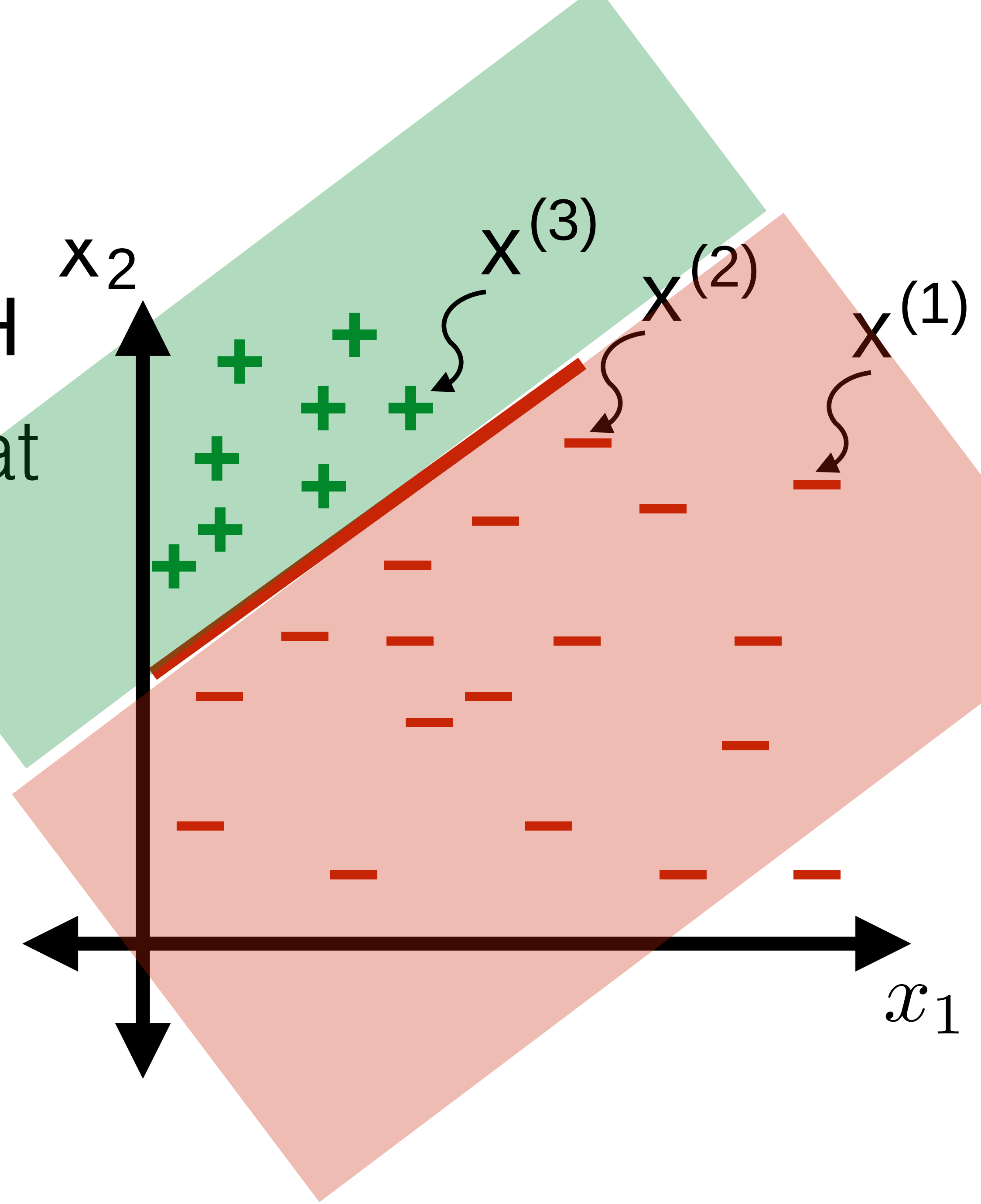
- Hypothesis class $\mathcal{H}$: set of **h ! H**
  - Example $\mathcal{H}$: All hypotheses that label +1 on one side of a line and -1 on the other side

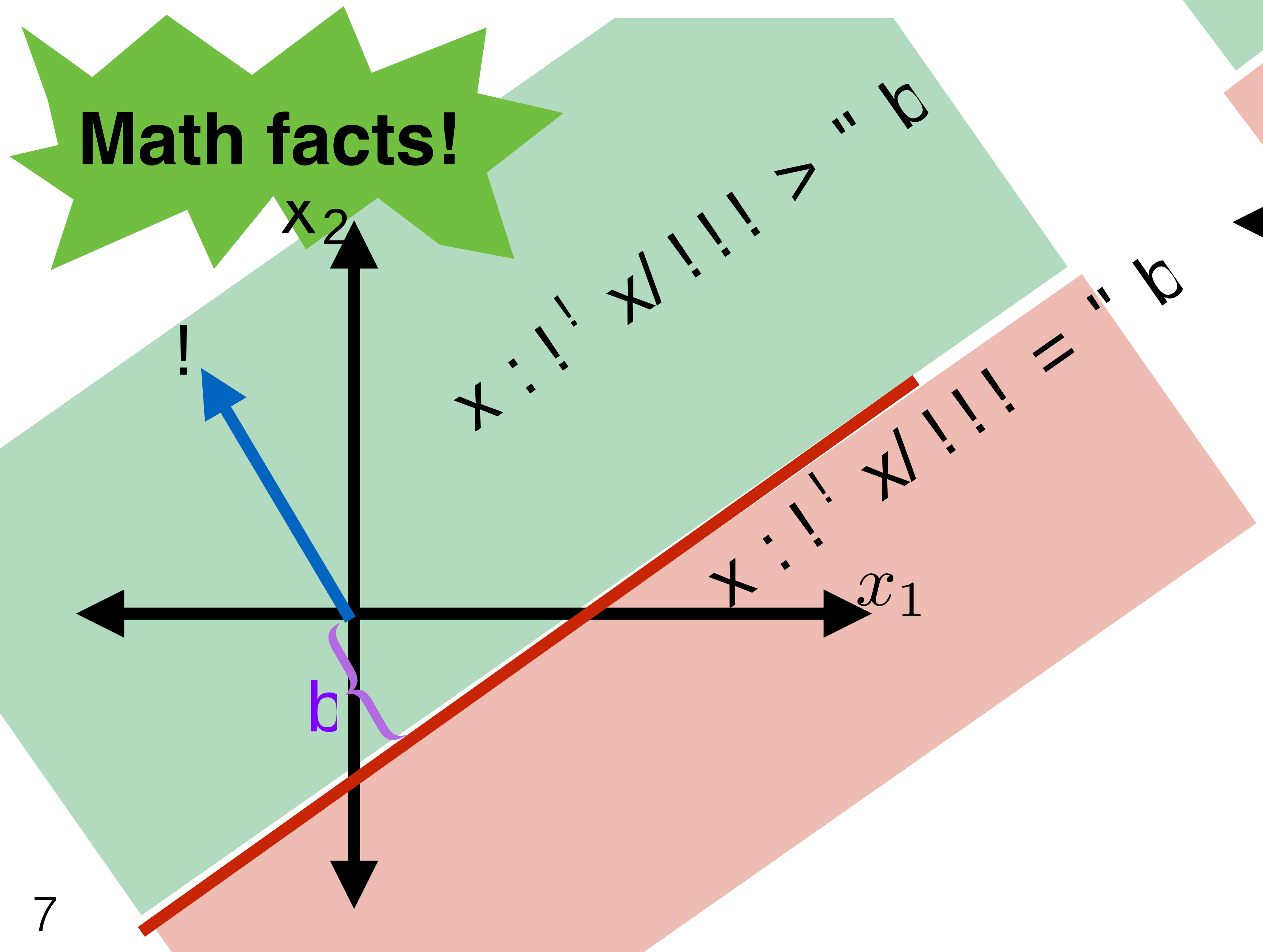




Linear classifiers

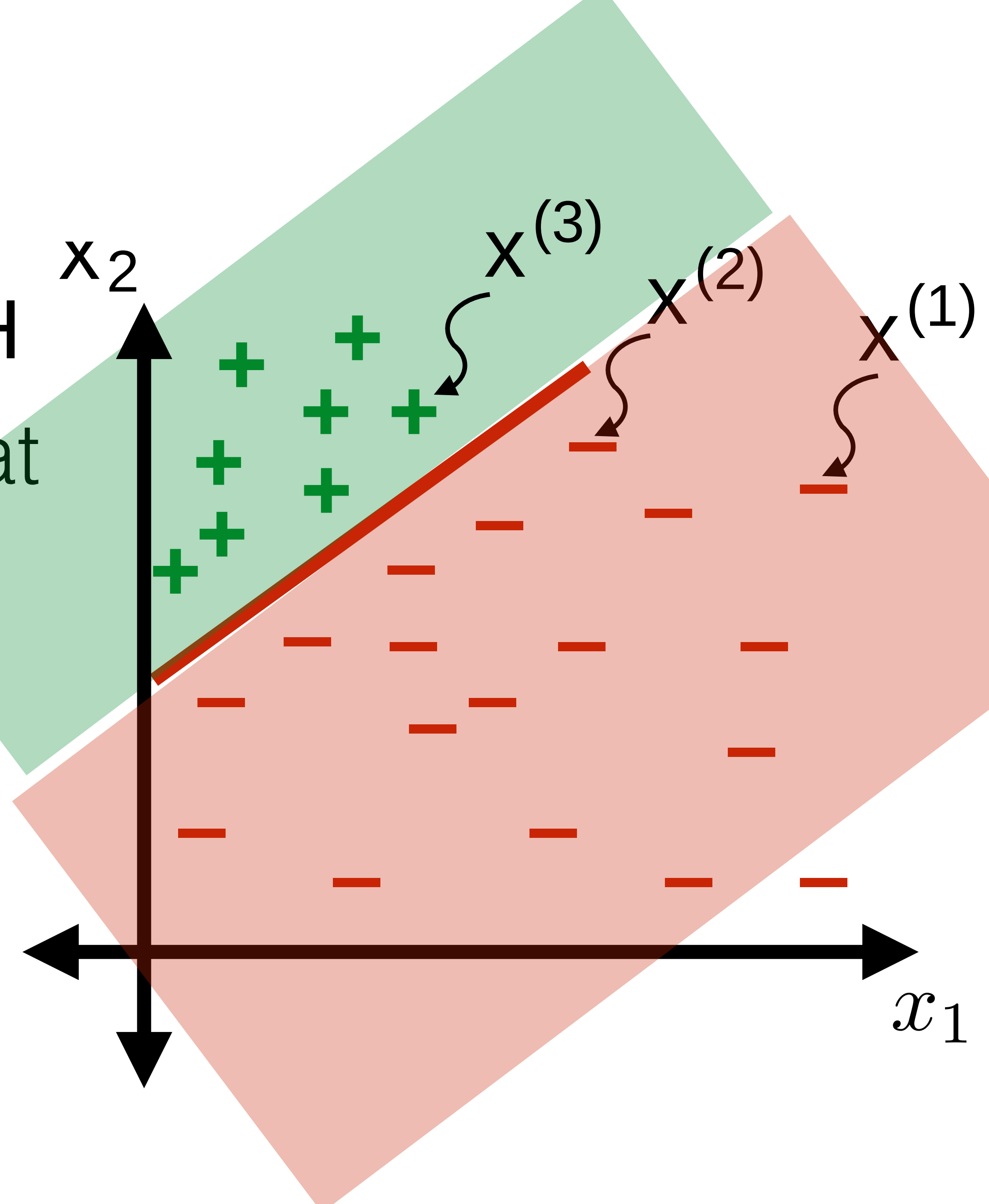
- Hypothesis class $\mathcal{H}$: set of **h ! H**
  - Example $\mathcal{H}$: All hypotheses that label +1 on one side of a line and -1 on the other side

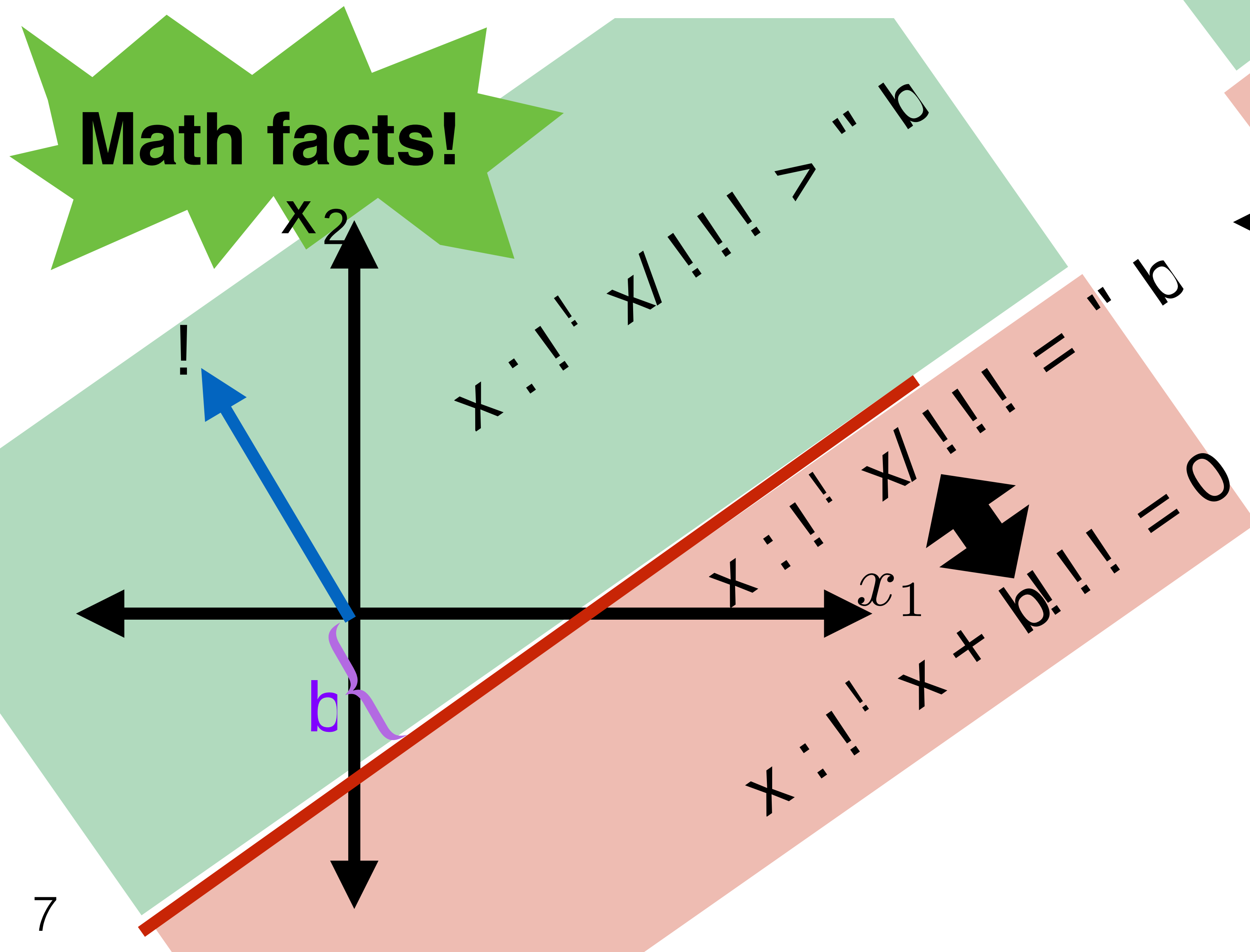




Linear classifiers

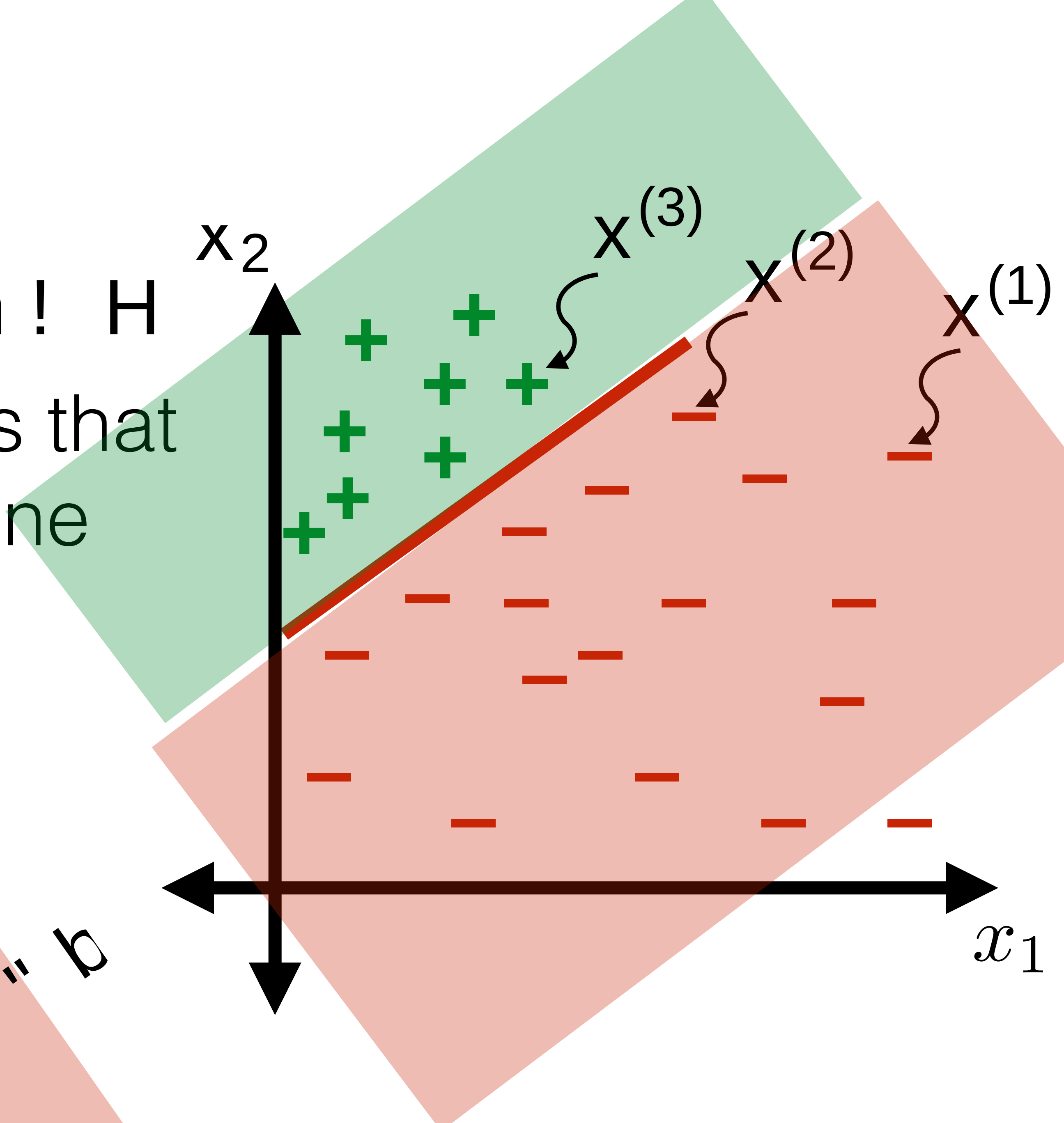
- Hypothesis class $\mathcal{H}$: set of **h ! H**
  - Example $\mathcal{H}$: All hypotheses that label +1 on one side of a line and -1 on the other side

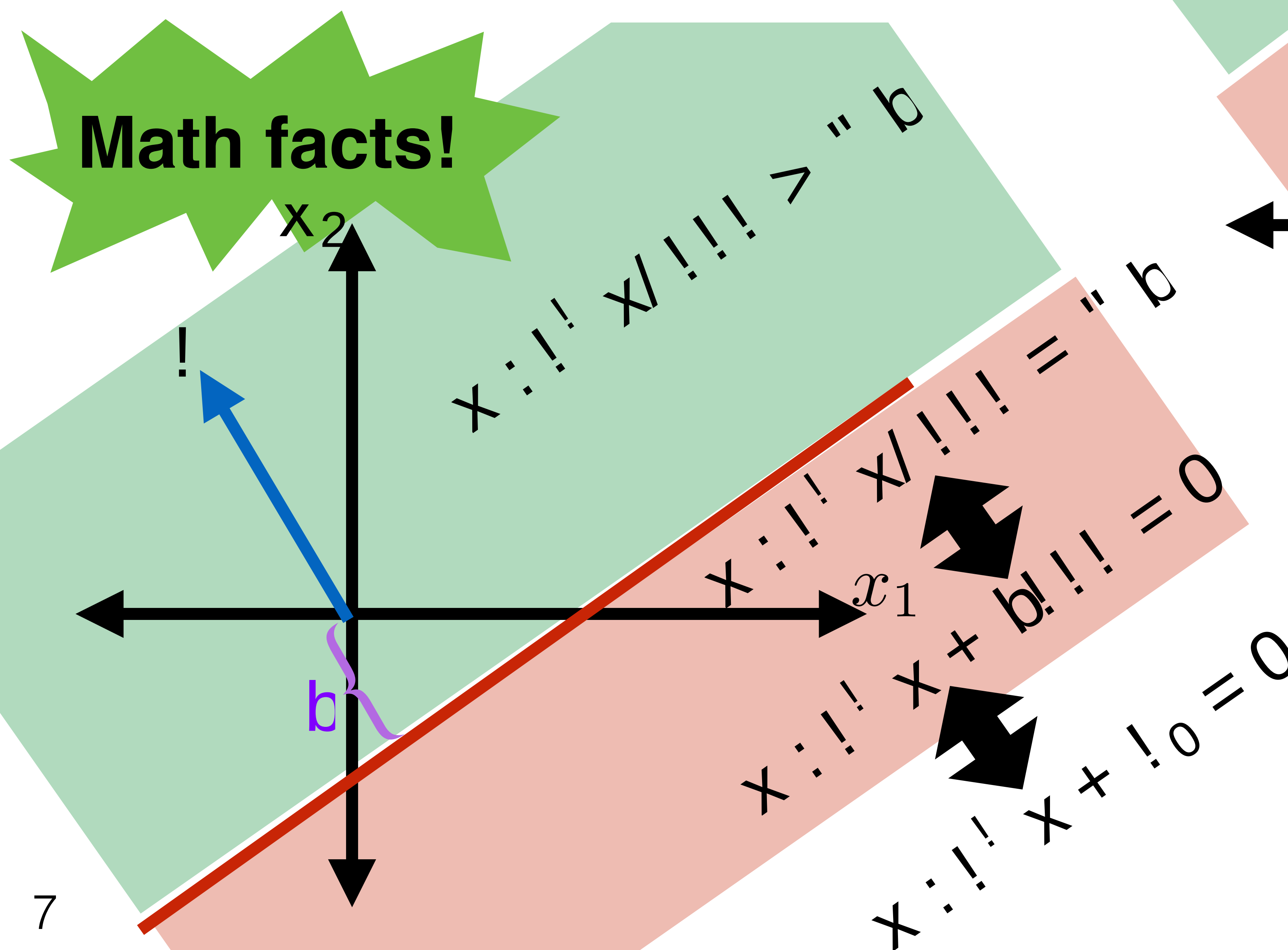




Linear classifiers

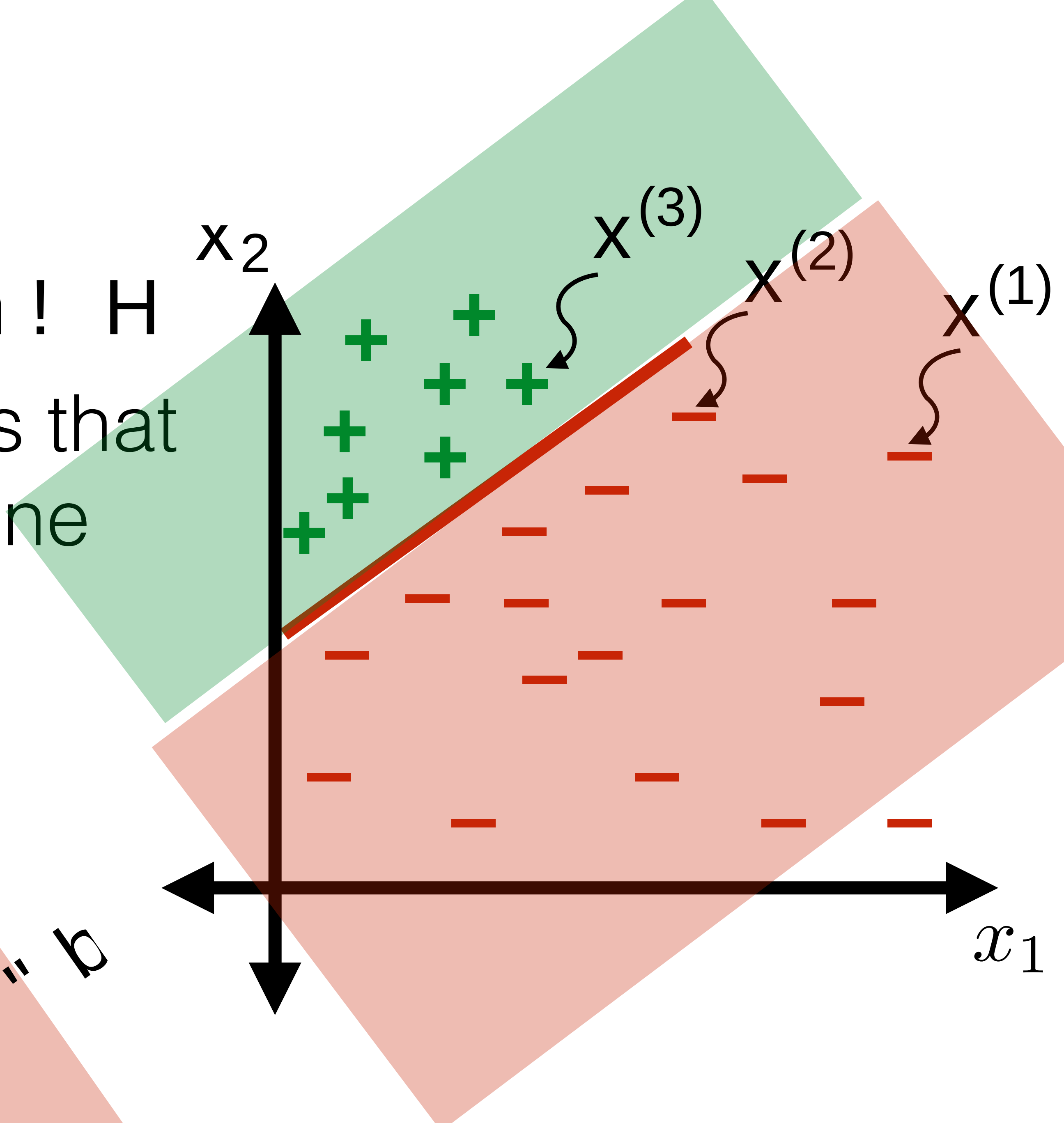
- Hypothesis class $\mathcal{H}$: set of **h** ! **H**
  - Example $\mathcal{H}$: All hypotheses that label +1 on one side of a line and -1 on the other side

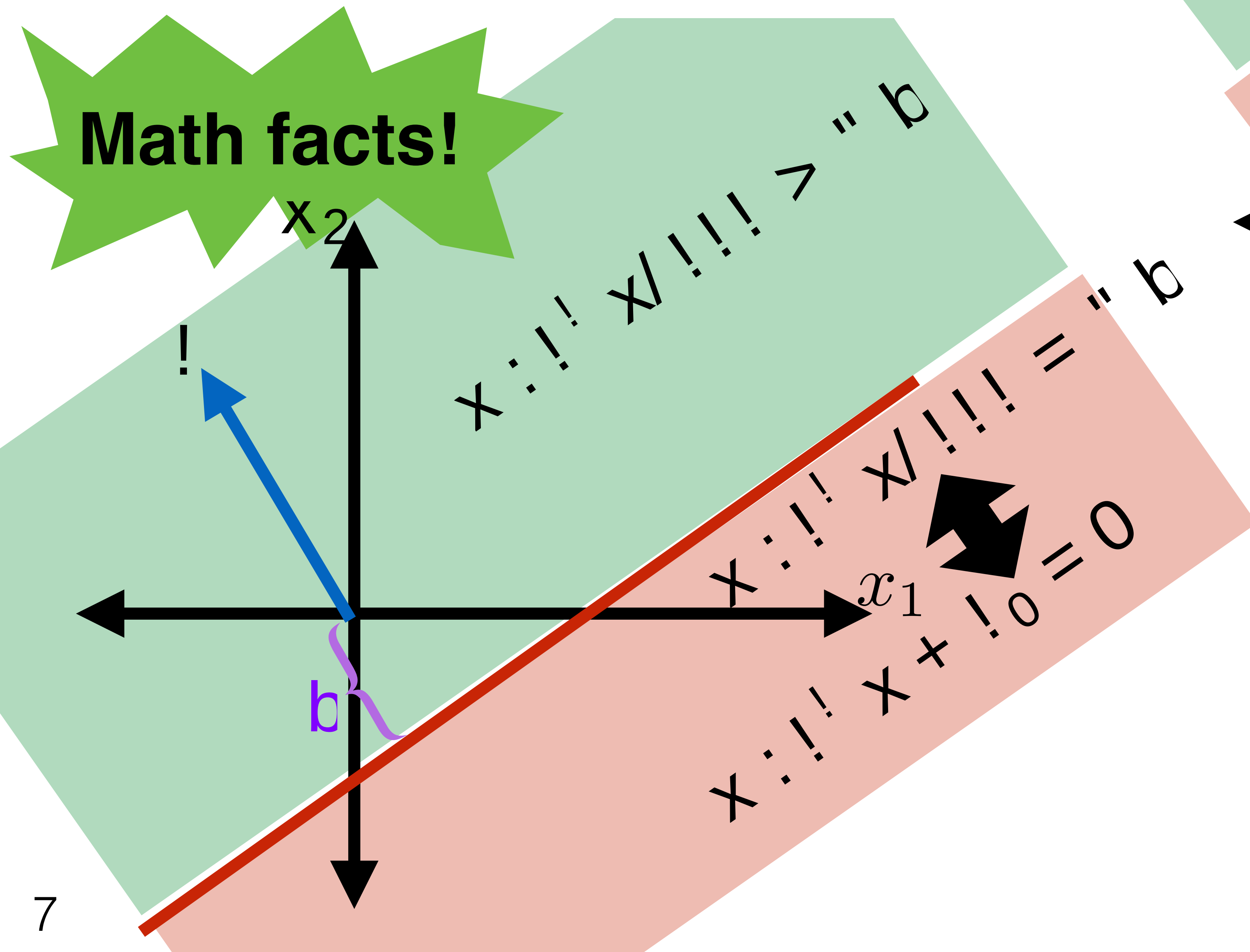




Linear classifiers

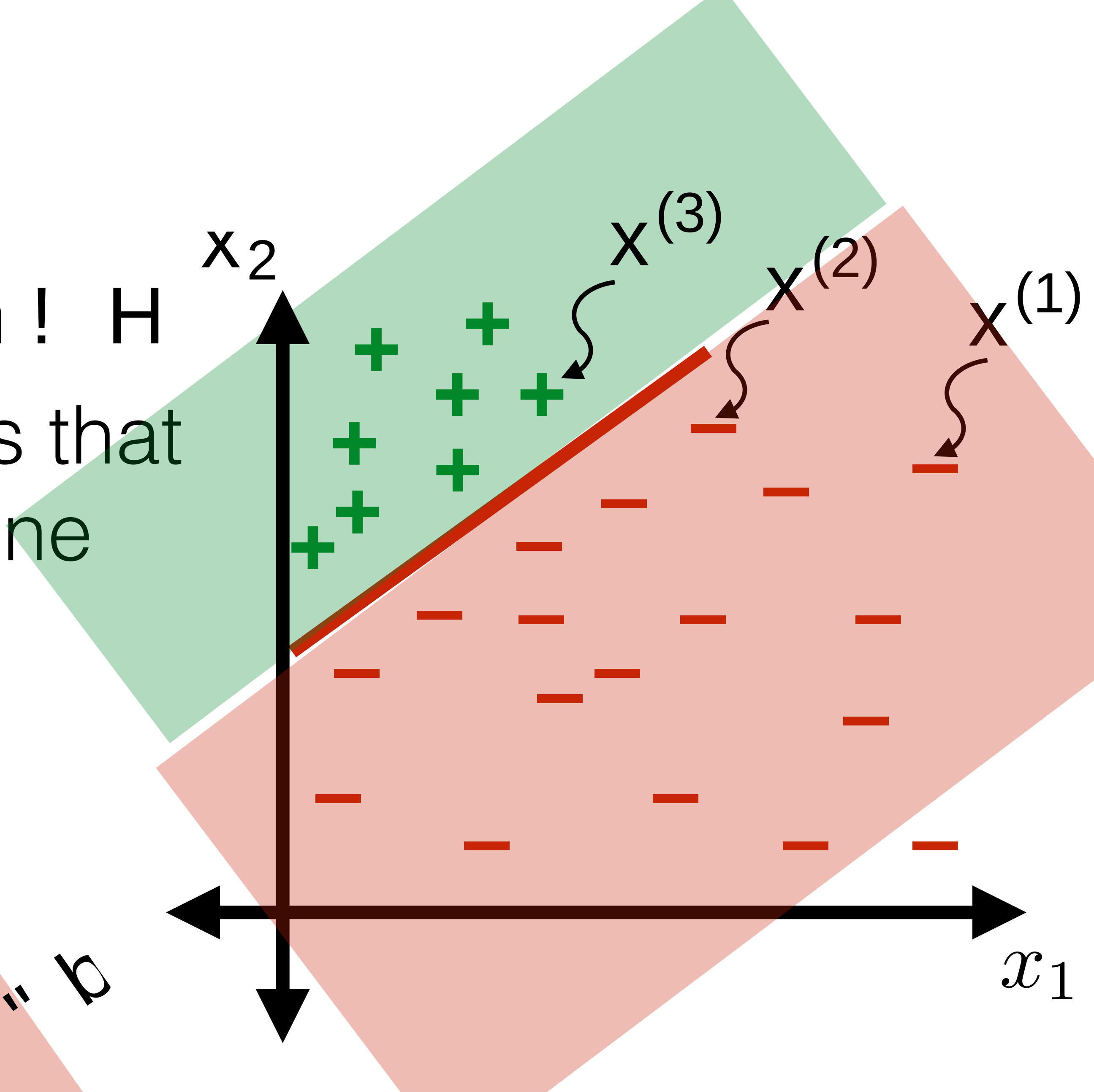
- Hypothesis class $\mathcal{H}$: set of **h** ! **H**
  - Example $\mathcal{H}$: All hypotheses that label +1 on one side of a line and -1 on the other side

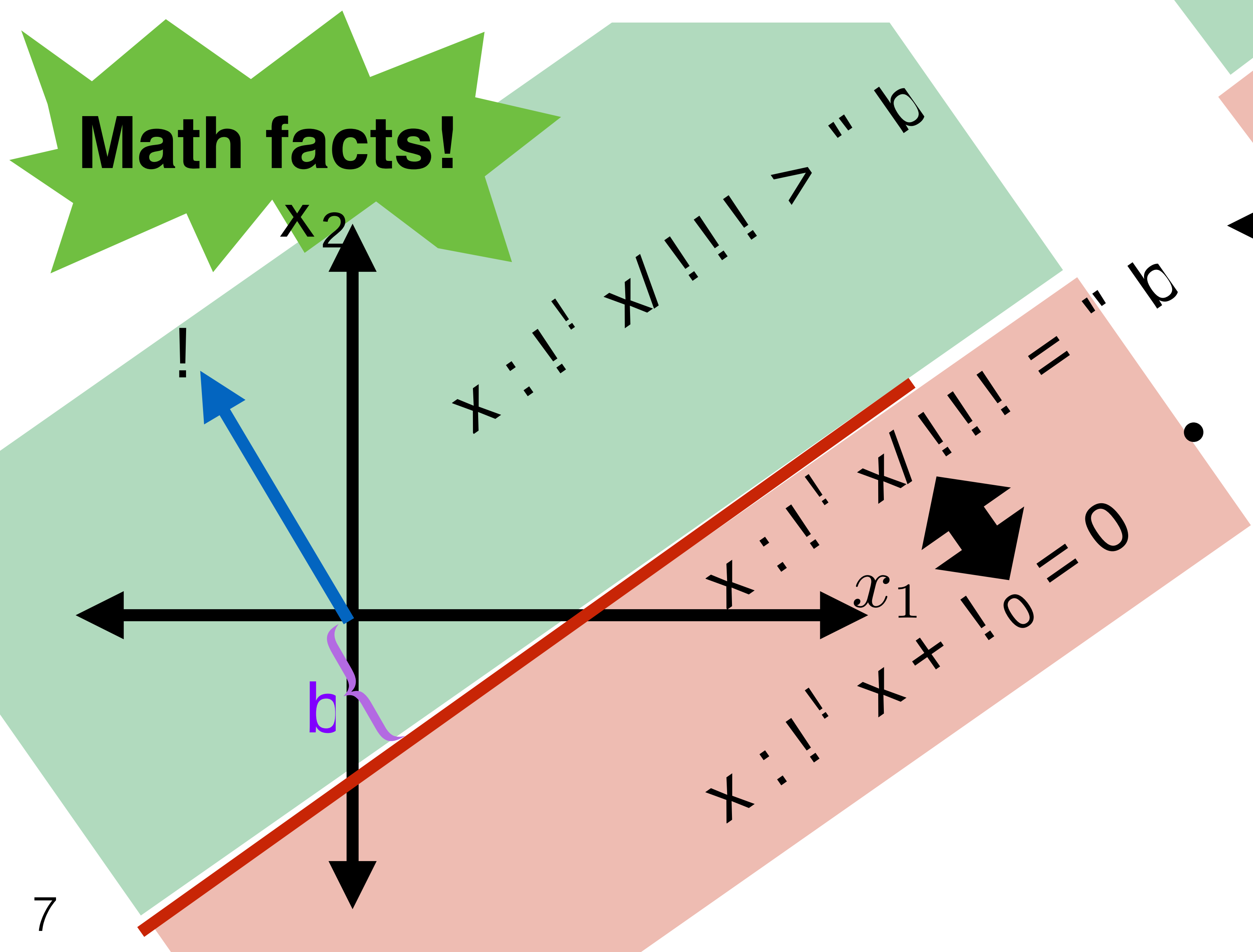




Linear classifiers

- Hypothesis class $\mathcal{H}$: set of **h ! H**
  - Example $\mathcal{H}$: All hypotheses that label +1 on one side of a line and -1 on the other side

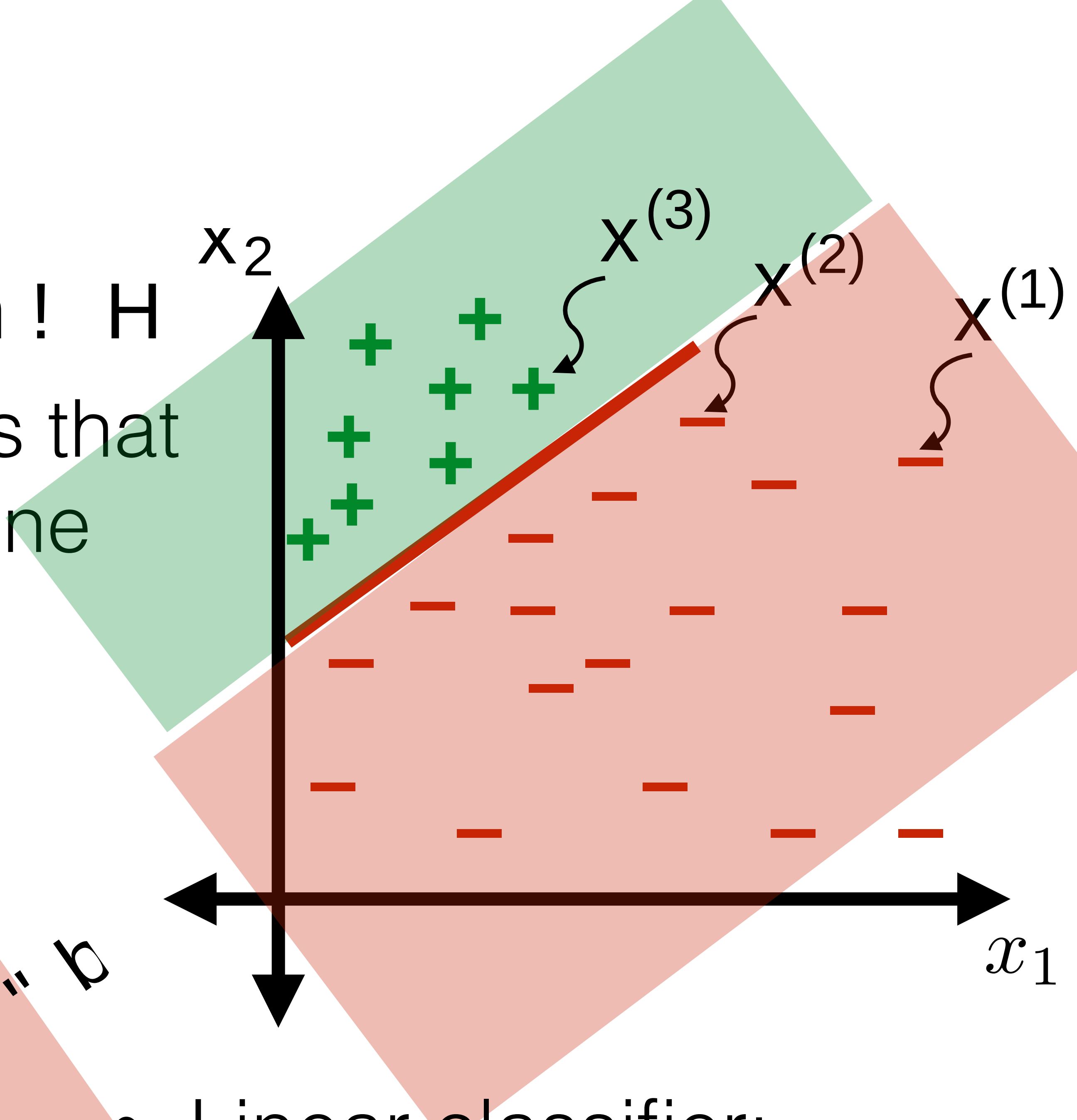


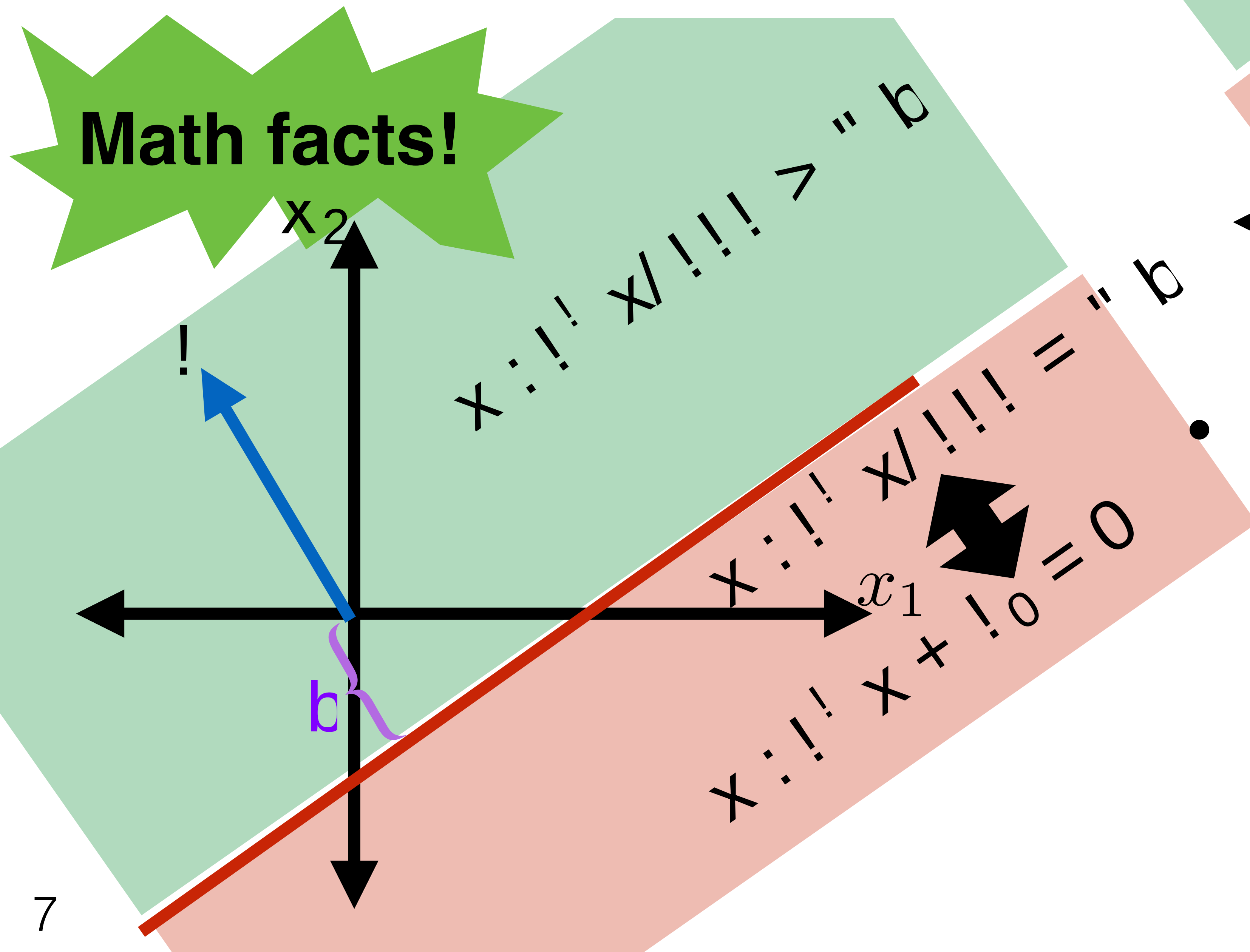


- Linear classifier:

Linear classifiers

- Hypothesis class $\mathcal{H}$: set of **h ! H**
  - Example $\mathcal{H}$: All hypotheses that label +1 on one side of a line and -1 on the other side

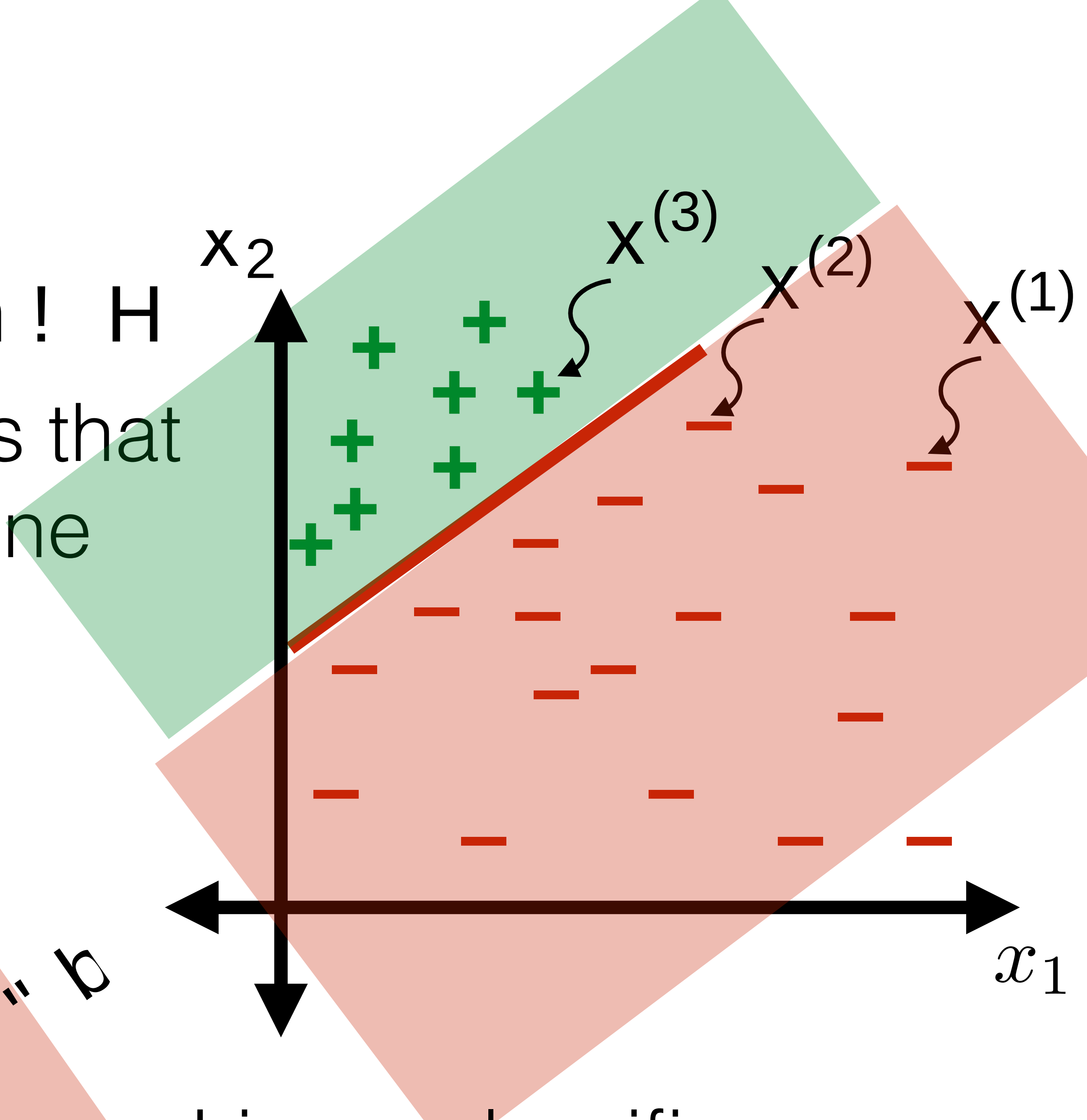




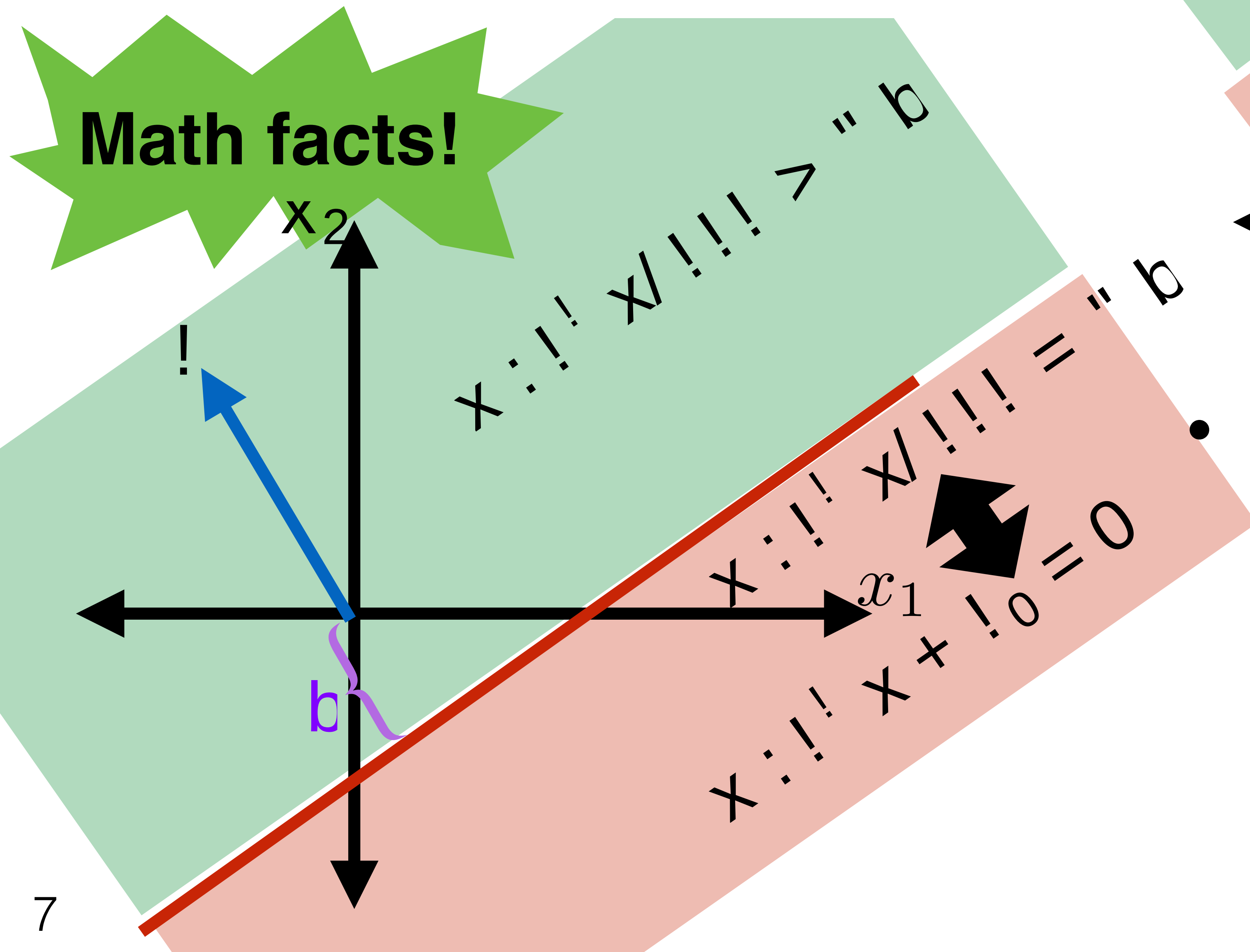
- Linear classifier:

  h(x) = sign(! ! x + ! $_0$)

Linear classifiers



- Hypothesis class $\mathcal{H}$: set of **h** ! **H**
  - Example $\mathcal{H}$: All hypotheses that label +1 on one side of a line and -1 on the other side



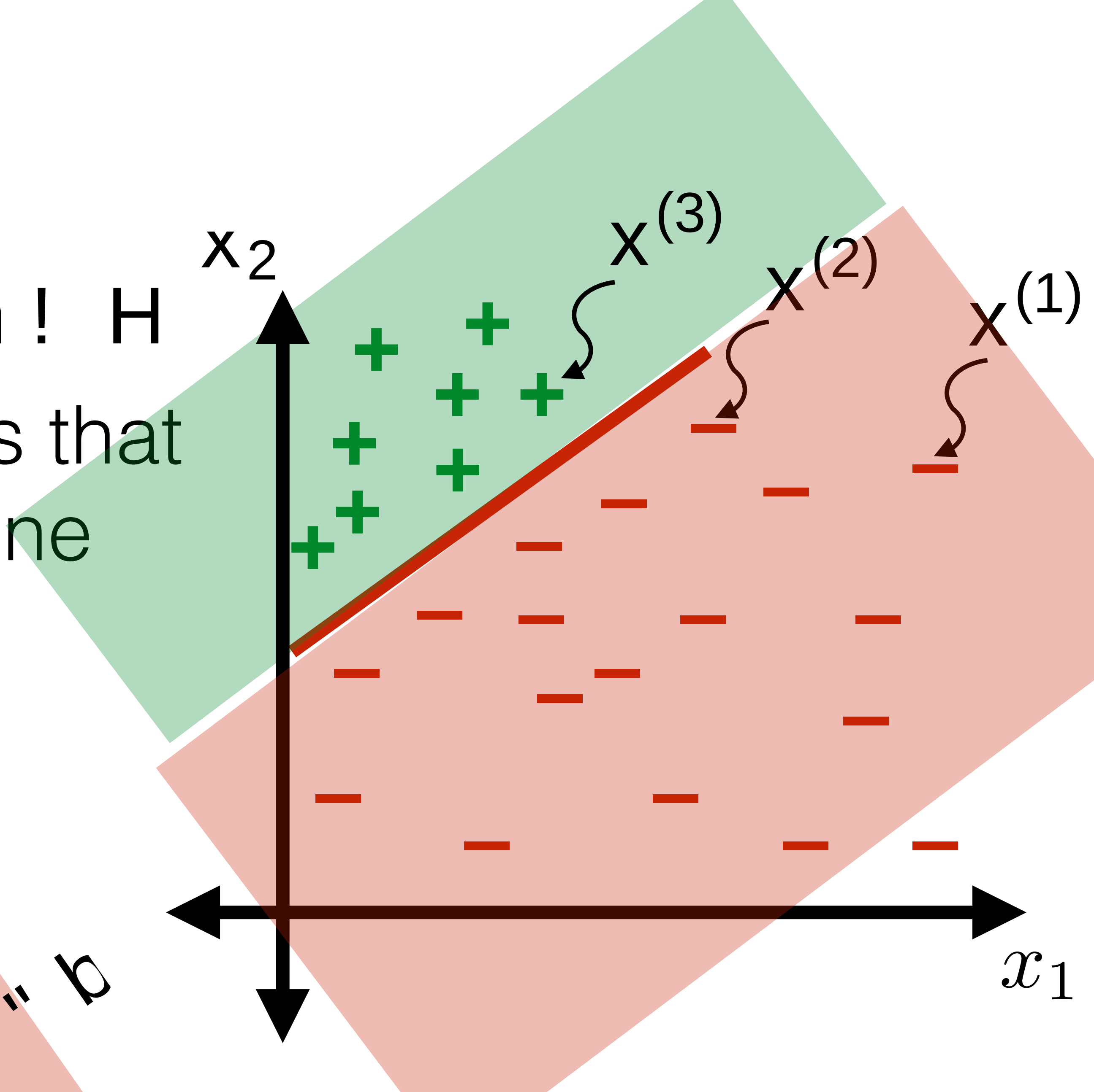
- Linear classifier:

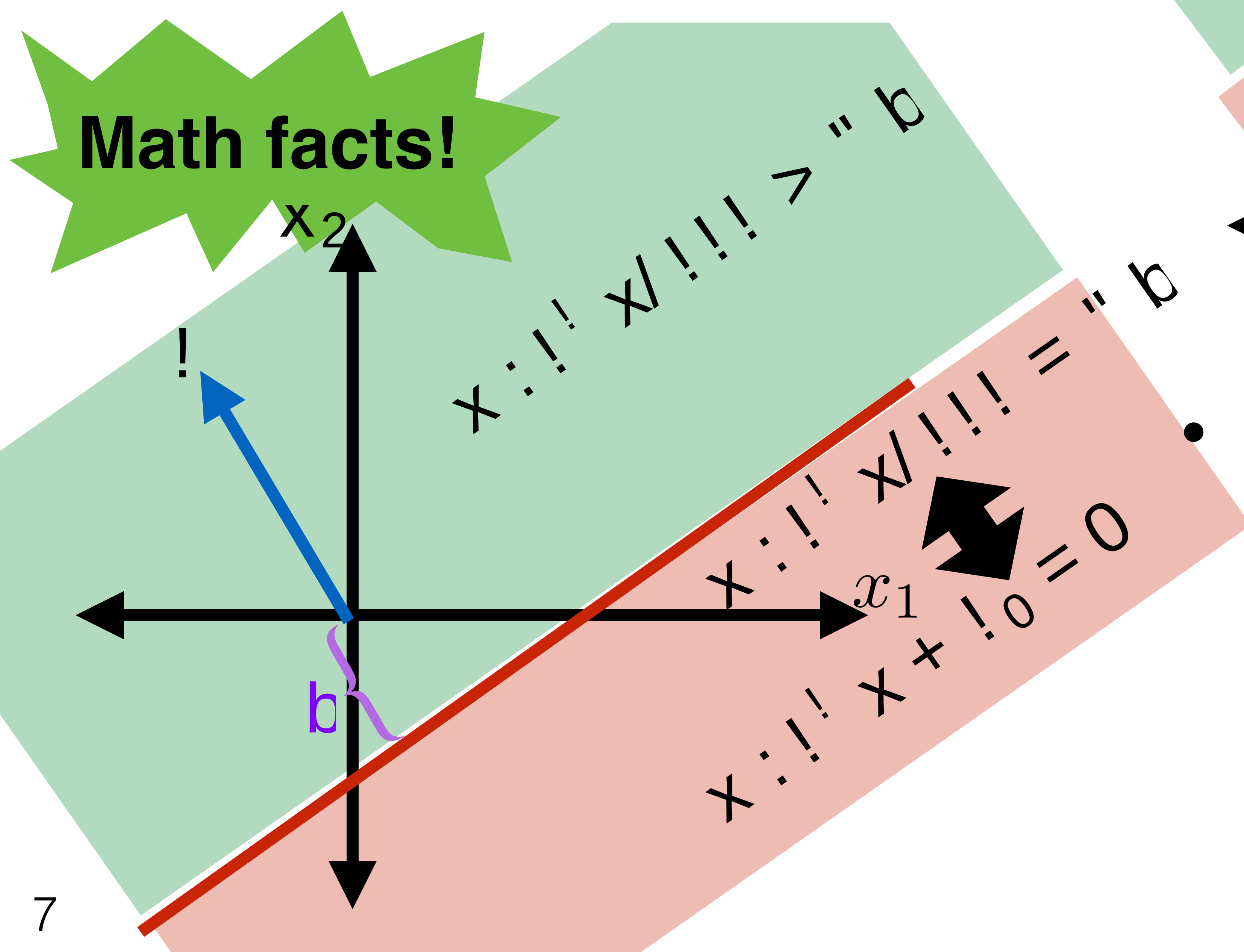
$$h(x) = \text{sign}(!^{!}\ x + !_0)$$

$$= \begin{cases} +1 \text{ if } !^{!}\ x + !_0 > 0 \\ !\ 1 \text{ if } !^{!}\ x + !_0 < 0 \end{cases}$$

Linear classifiers

- Hypothesis class $\mathcal{H}$: set of **h** ! **H**
  - Example $\mathcal{H}$: All hypotheses that label +1 on one side of a line and -1 on the other side





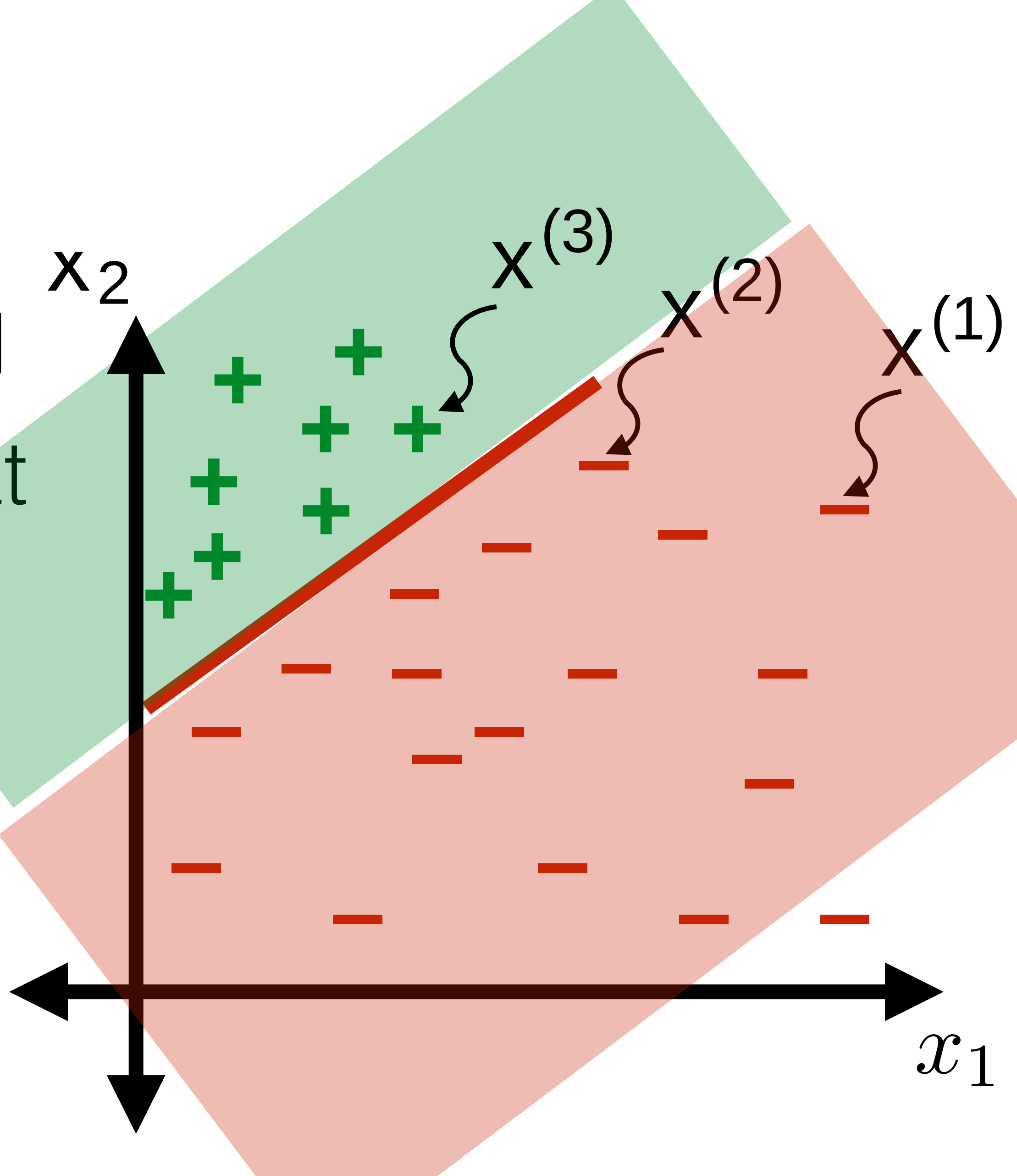
- Linear classifier:

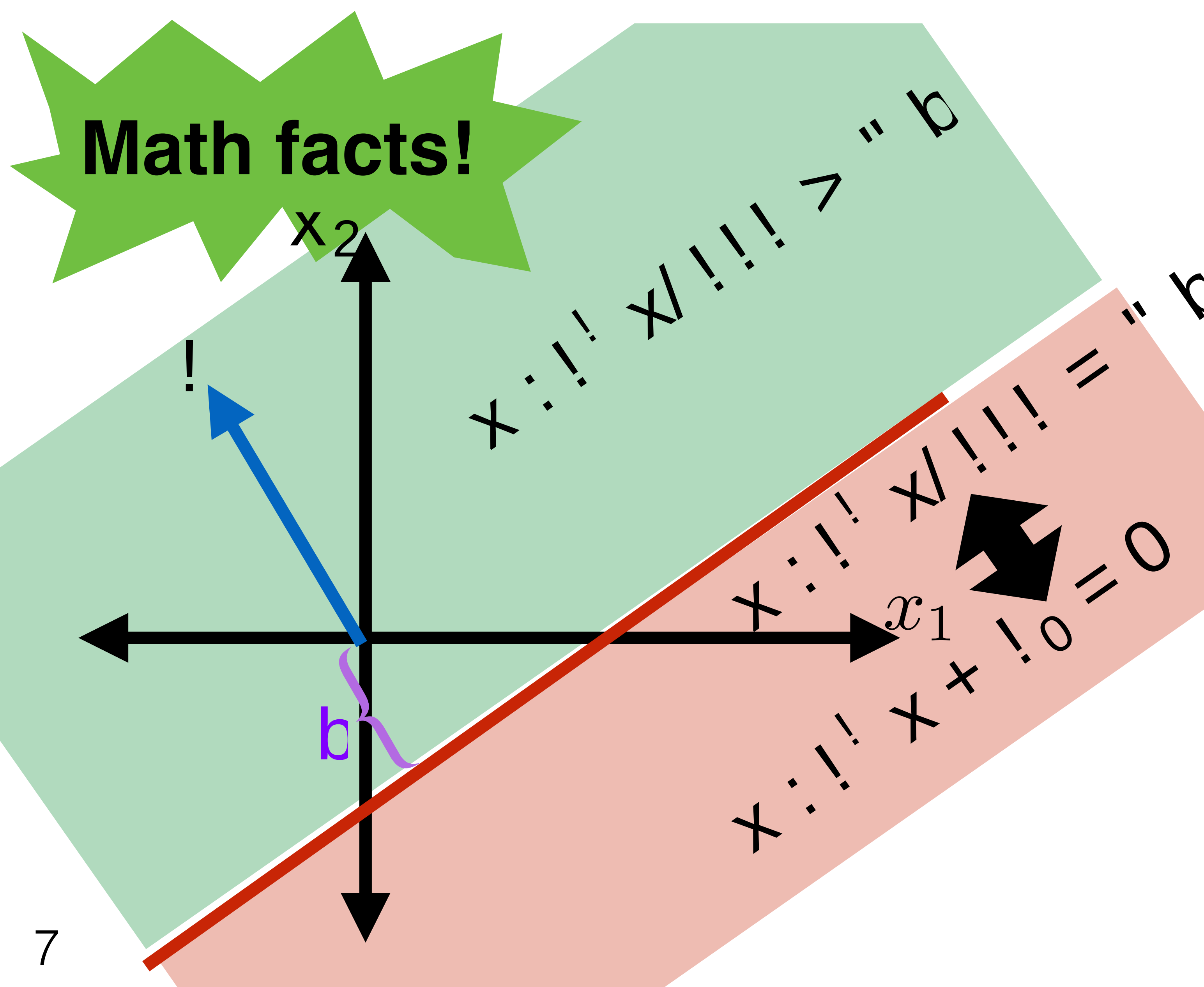
$$h(x) = \text{sign}(\,!^{!}\, x + !_0)$$

$$= \begin{cases} +1 \text{ if } !^{!}\, x + !_0 > 0 \\ ! \ 1 \text{ if } !^{!}\, x + !_0 < 0 \end{cases}$$

Linear classifiers

- Hypothesis class $\mathcal{H}$: set of **h ! H**
  - Example $\mathcal{H}$: All hypotheses that label +1 on one side of a line and -1 on the other side





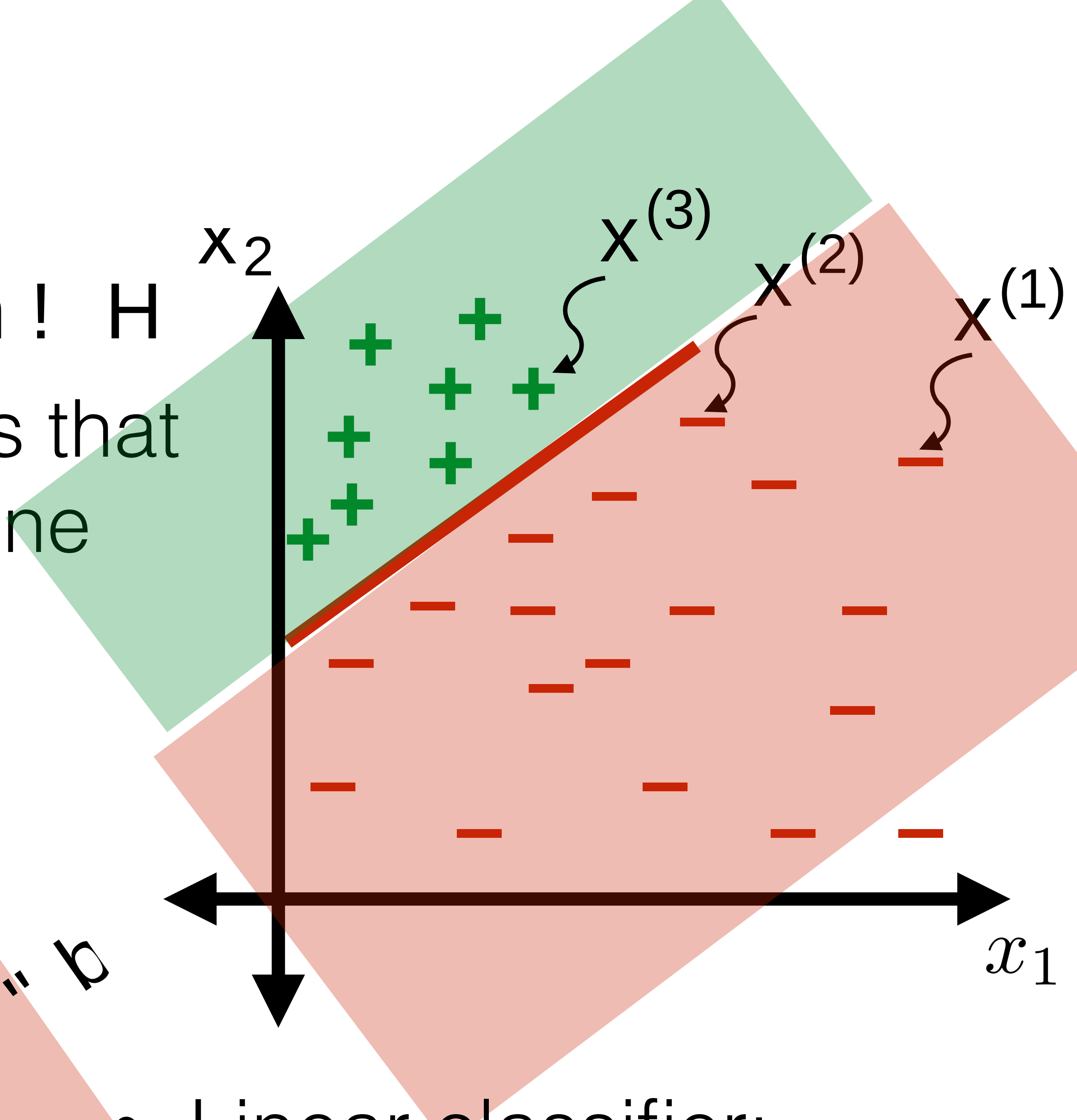
- Linear classifier:

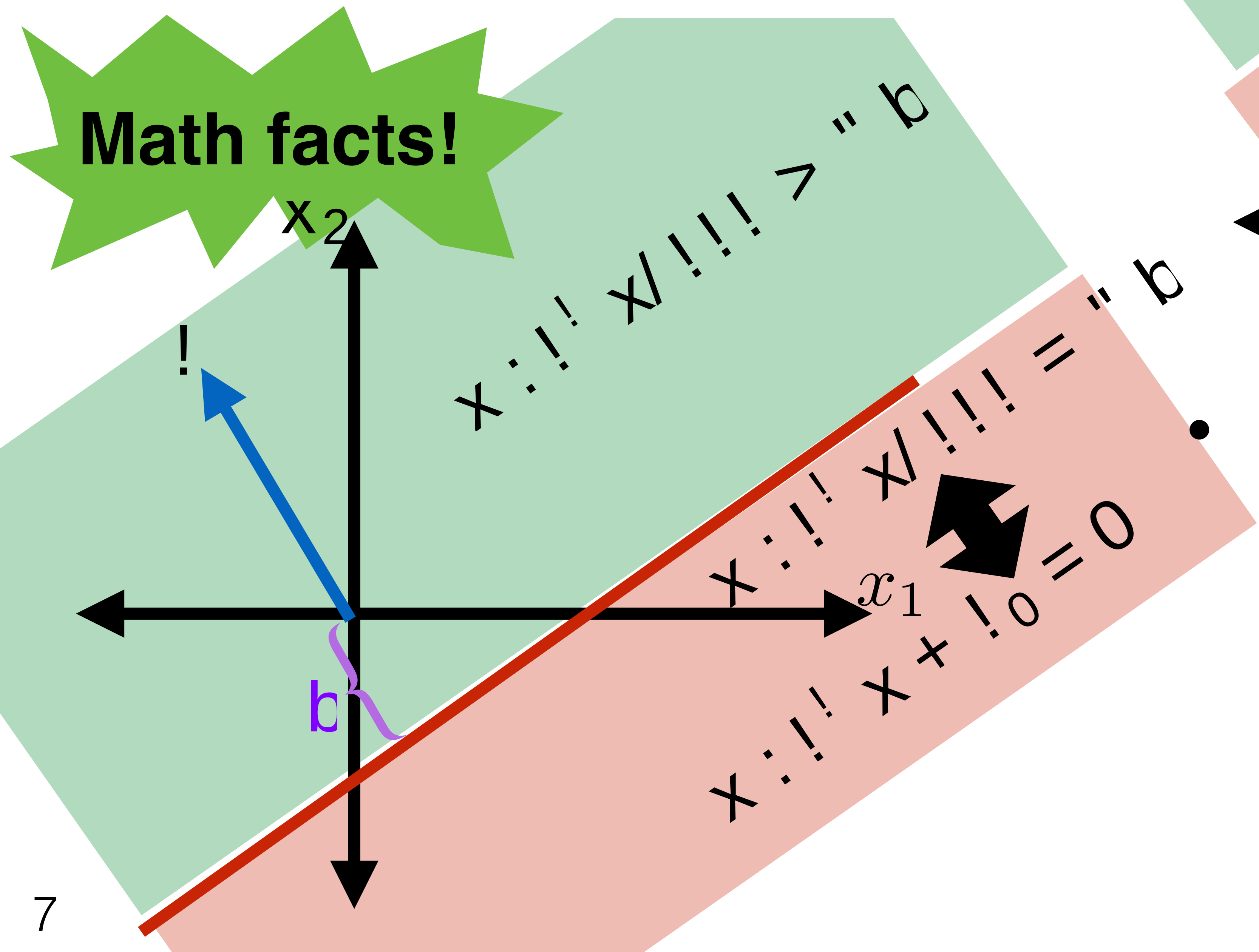
$$h(x) = \text{sign}(!^{!}\, x + !_0)$$

$$= \begin{cases} +1 \text{ if } !^{!}\, x + !_0 > 0 \\ ! \ 1 \text{ if } !^{!}\, x + !_0 \ \text{"}\ 0 \end{cases}$$

Linear classifiers

- Hypothesis class $\mathcal{H}$: set of **h ! H**
  - Example $\mathcal{H}$: All hypotheses that label +1 on one side of a line and -1 on the other side





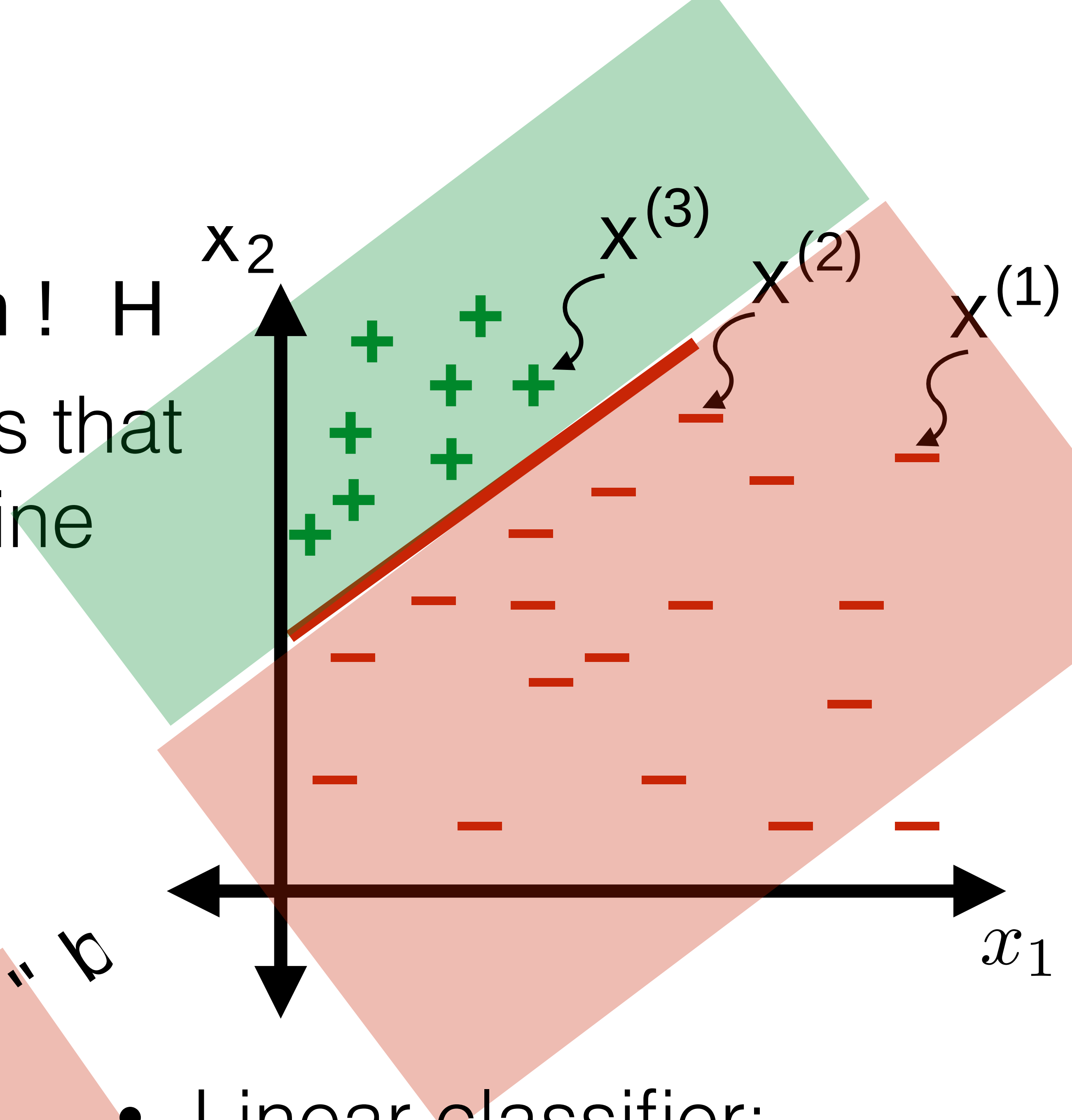
- Linear classifier:

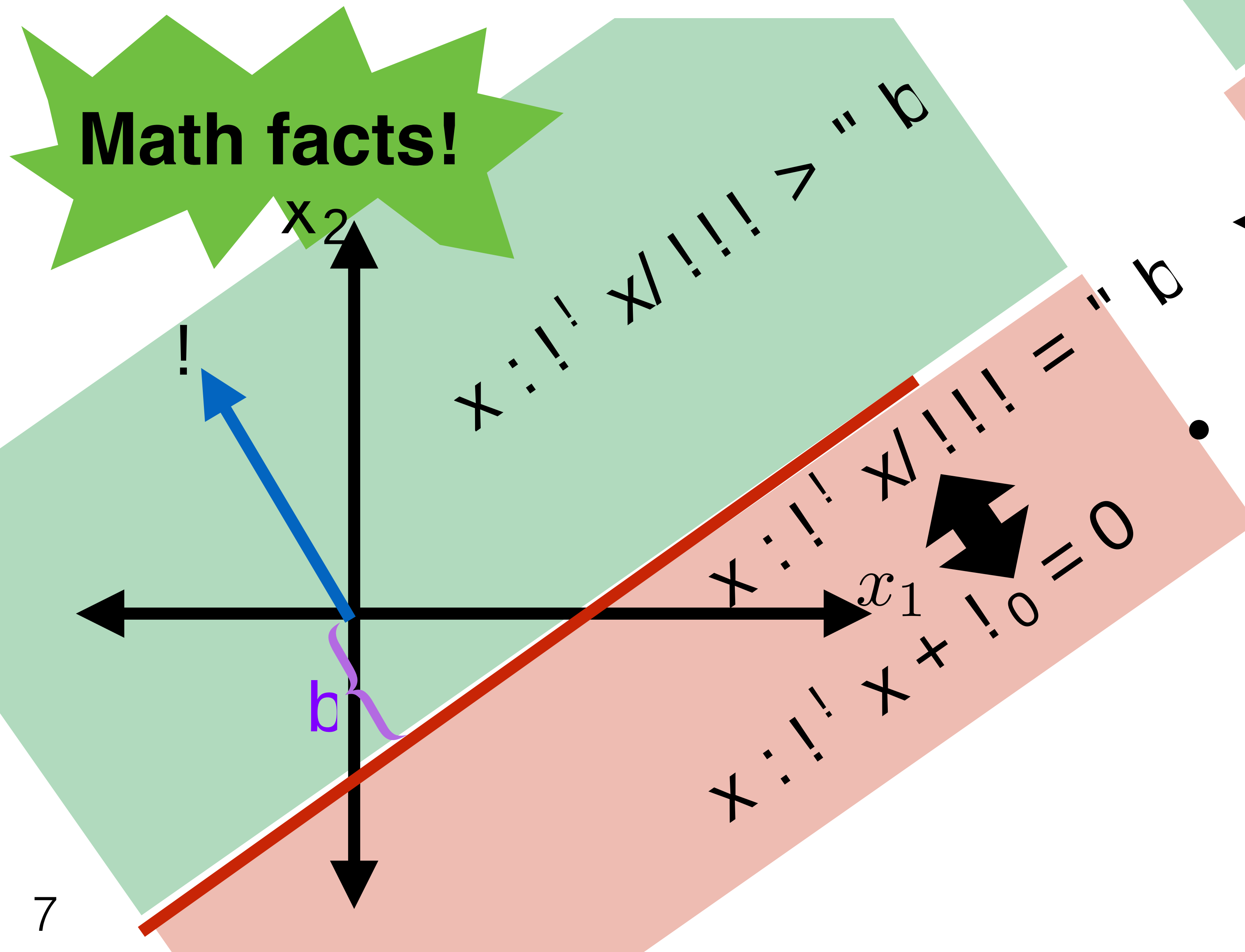
$$h(x) = \mathrm{sign}(!^{!}\, x + !_0)$$

$$= \begin{cases} +1 \text{ if } !^{!}\, x + !_0 > 0 \\ !\ 1 \text{ if } !^{!}\, x + !_0 \text{ " } 0 \end{cases}$$

Linear classifiers

- Hypothesis class $\mathcal{H}$: set of **h ! H**
  - Example $\mathcal{H}$: All hypotheses that label +1 on one side of a line and -1 on the other side





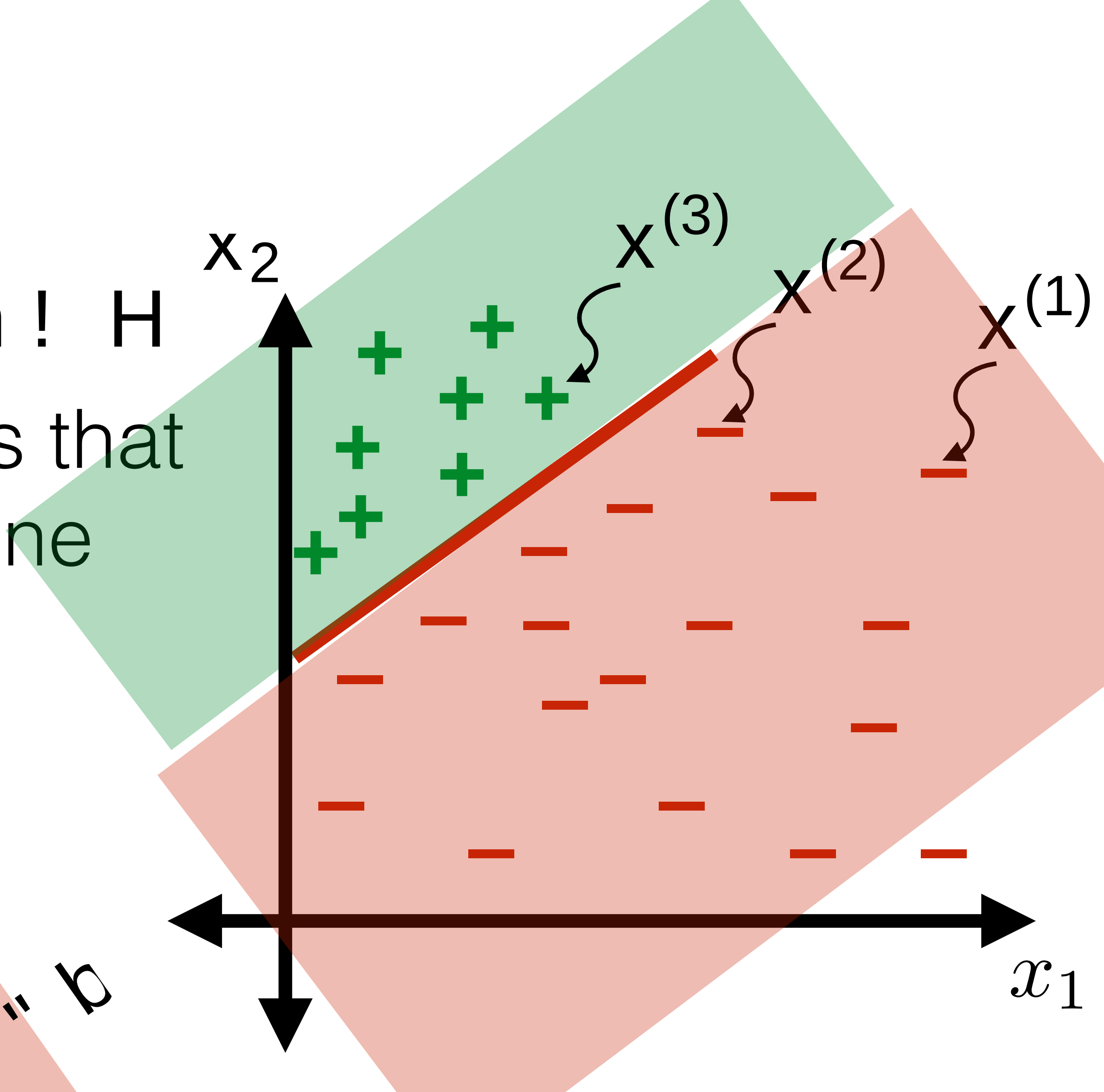
- Linear classifier:

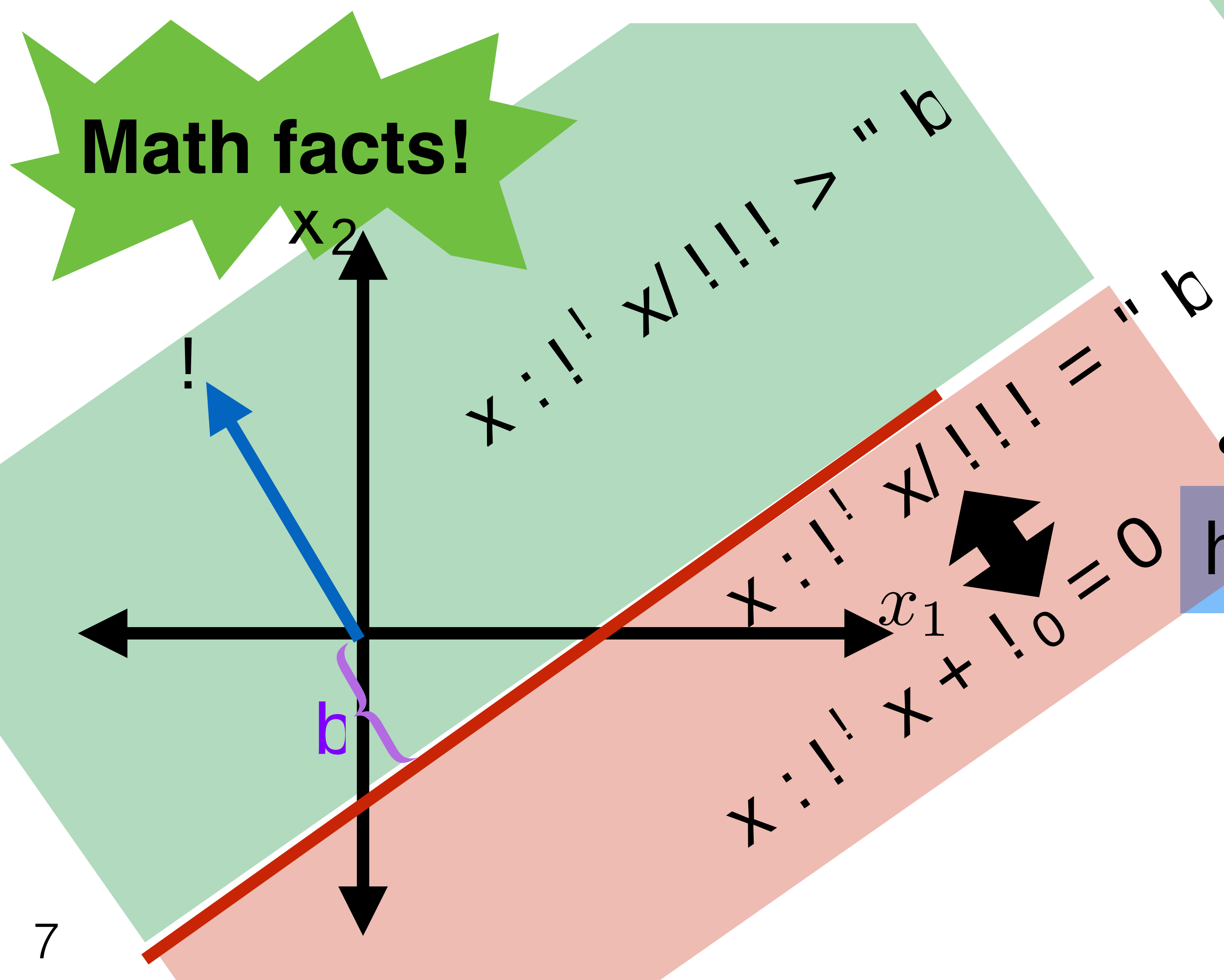
$$h(x) = \text{sign}(!^! x + !_0)$$

$$= \begin{cases} +1 \text{ if } !^! x + !_0 > 0 \\ ! \; 1 \text{ if } !^! x + !_0 \;" \; 0 \end{cases}$$

Linear classifiers

- Hypothesis class $\mathcal{H}$: set of **h ! H**
  - Example $\mathcal{H}$: All hypotheses that label +1 on one side of a line and -1 on the other side





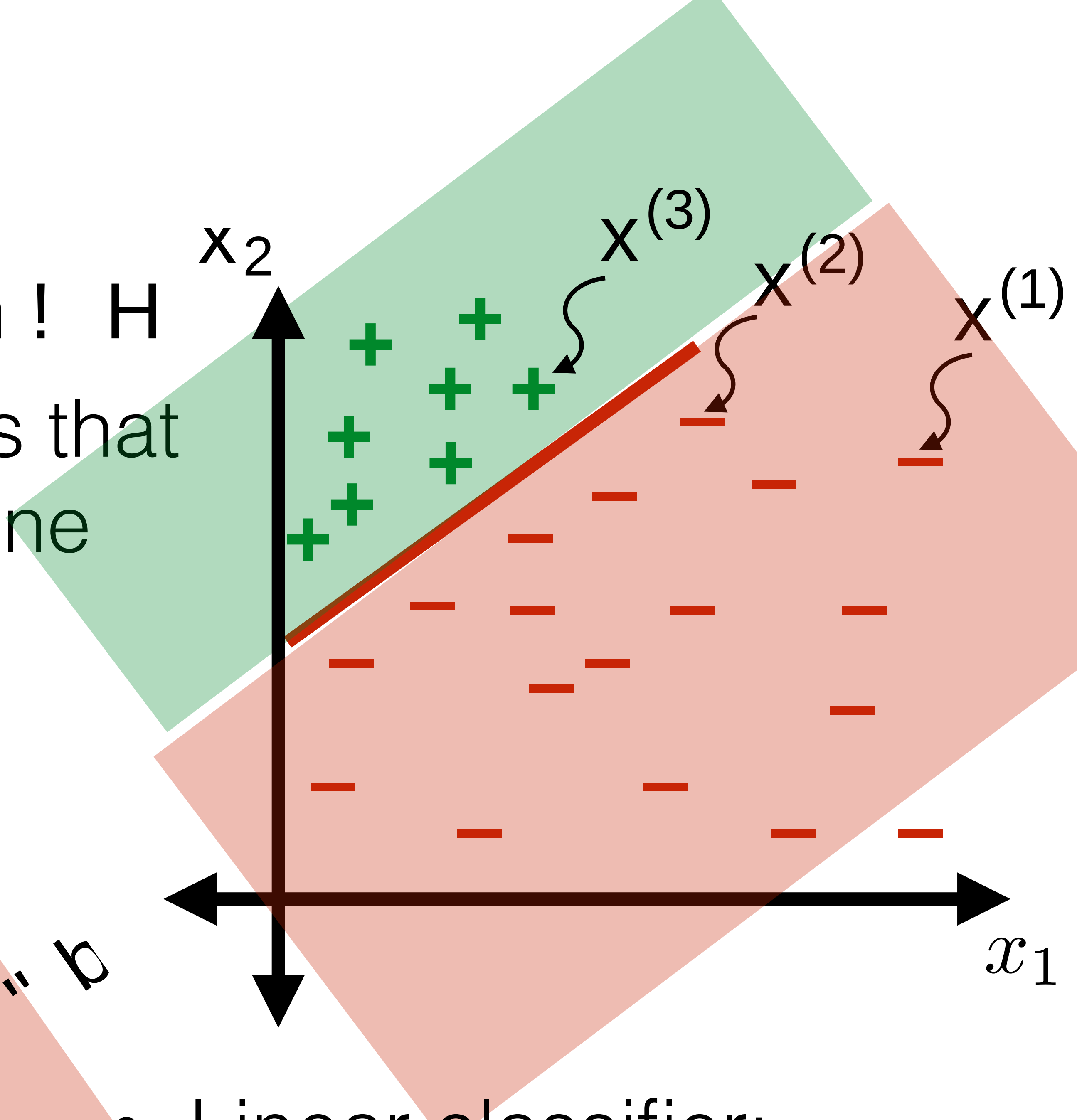
- Linear classifier:

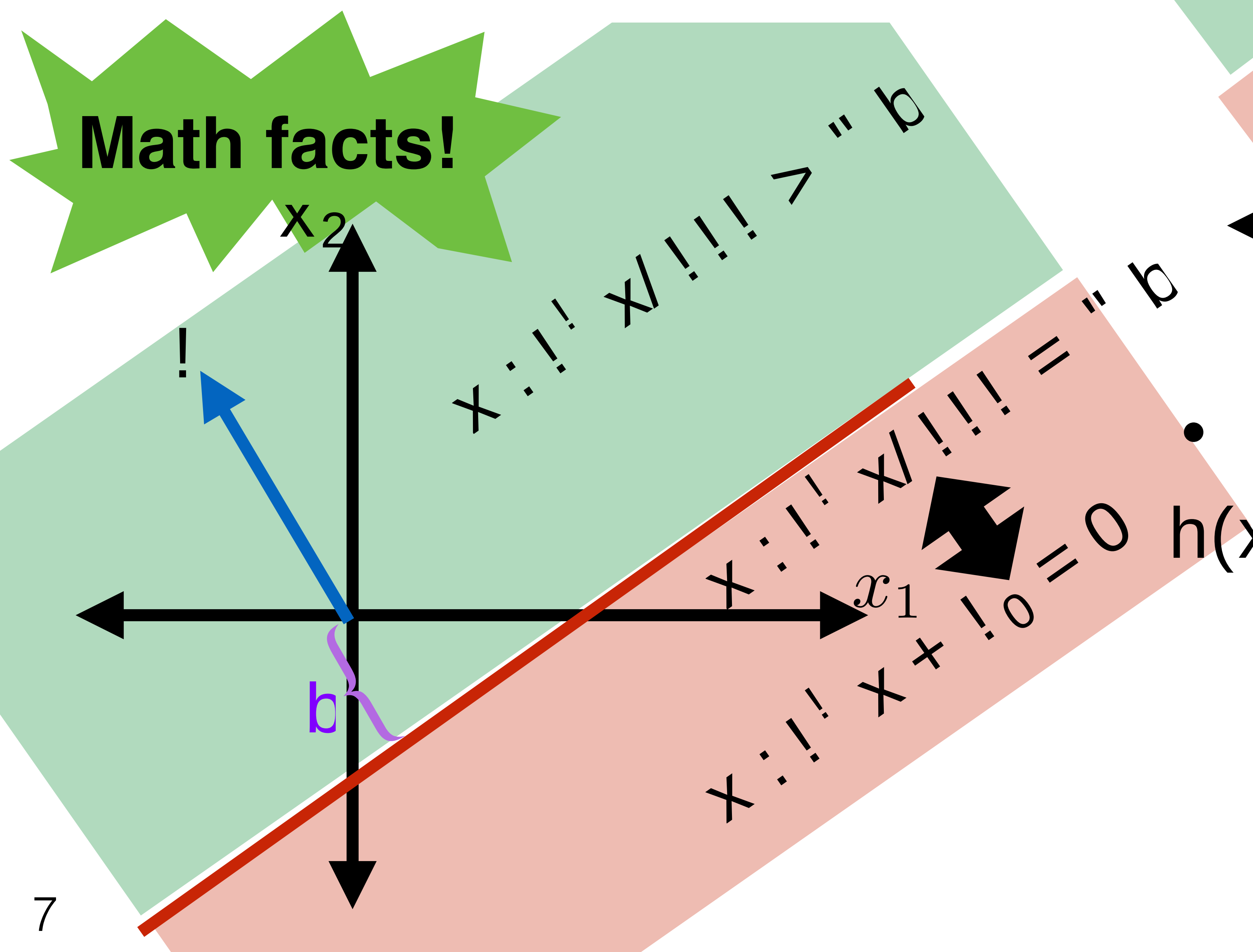
h(x; ! , !$_0$) = sign(! $^!$ x + !$_0$)

= +1 if ! $^!$ x + !$_0$ > 0
! 1 if ! $^!$ x + !$_0$ " 0

Linear classifiers

- Hypothesis class $\mathcal{H}$: set of **h** ! **H**
  - Example $\mathcal{H}$: All hypotheses that label +1 on one side of a line and -1 on the other side





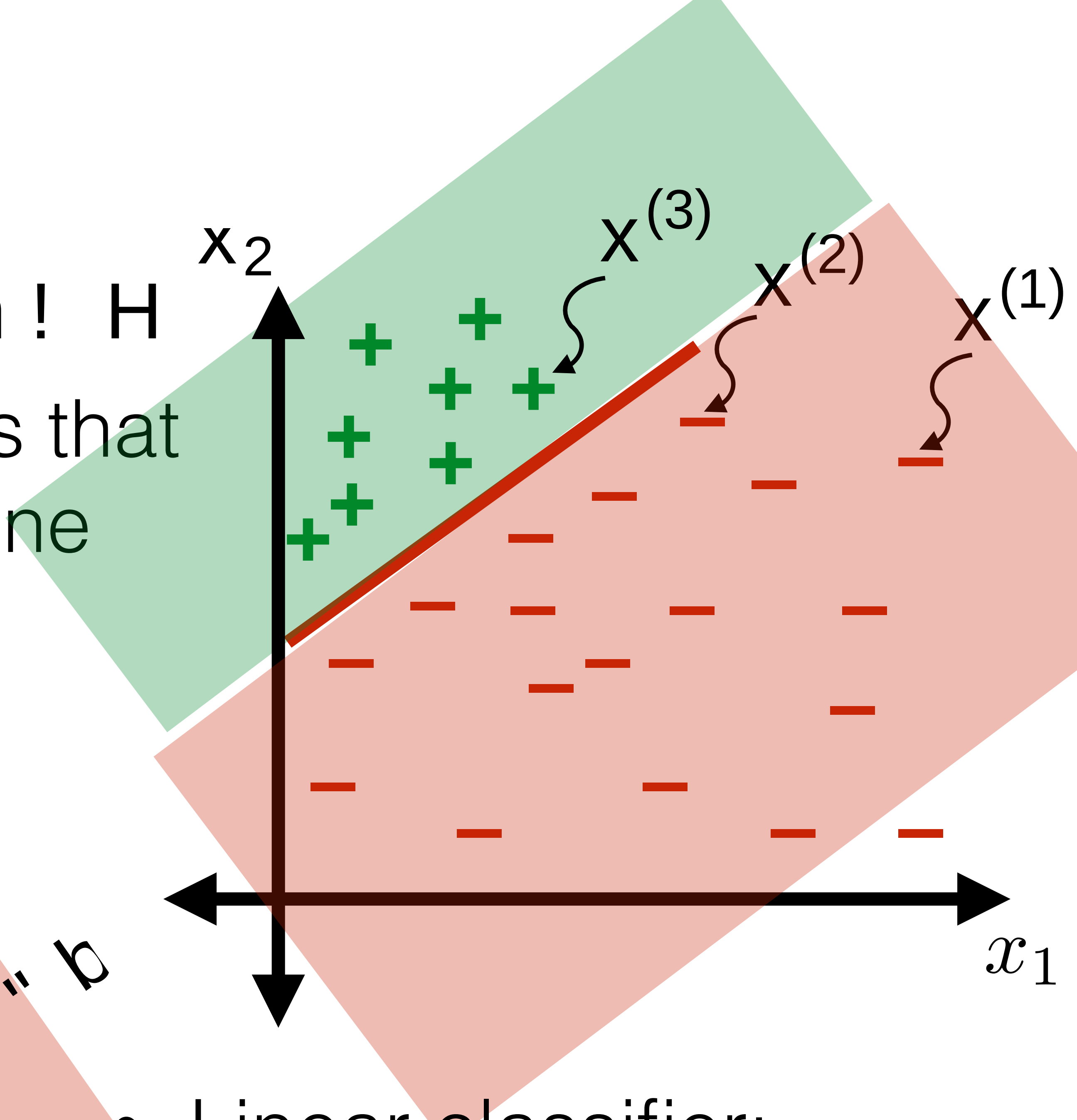
- Linear classifier:

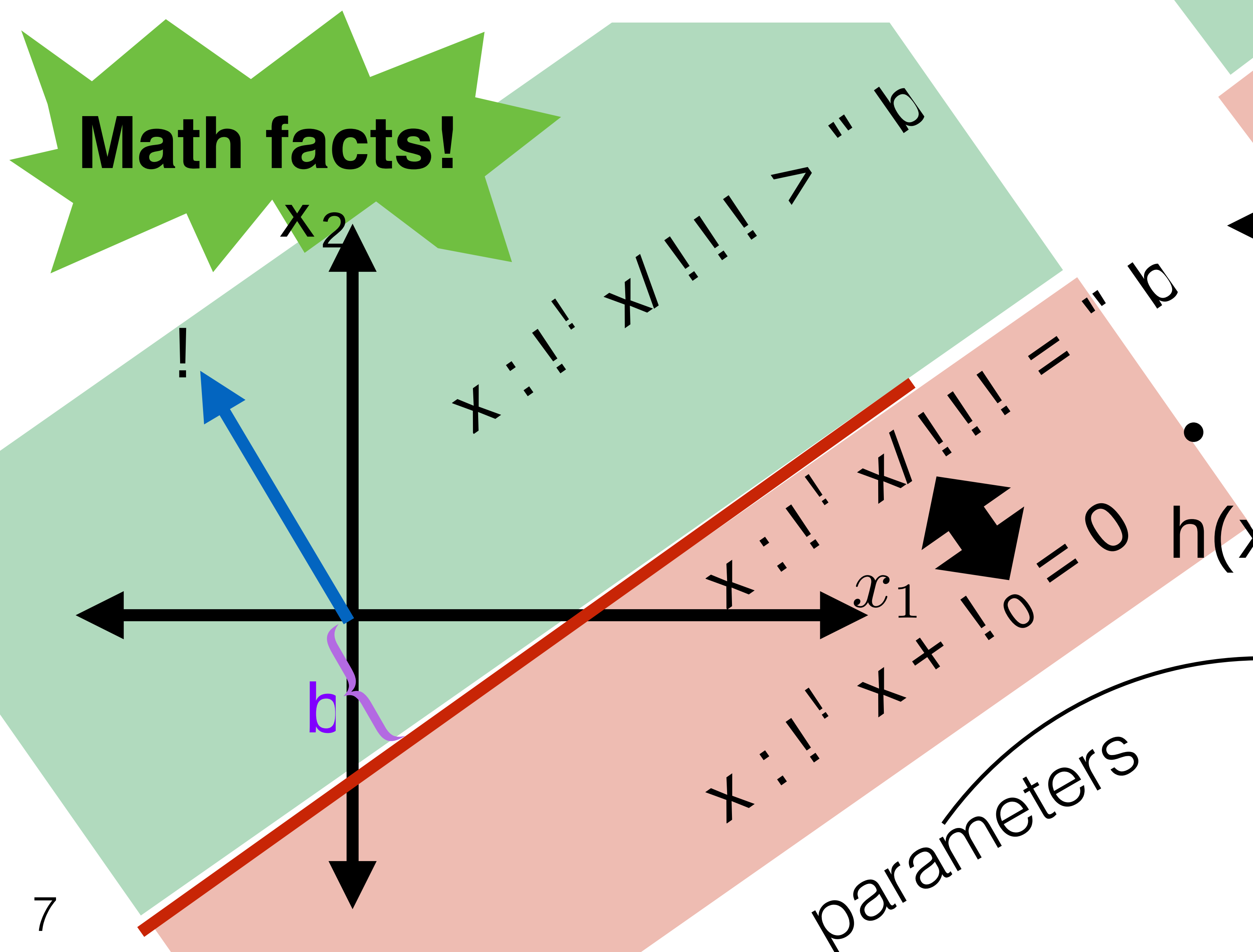
$$h(x; !, !_0) = \text{sign}(!^{!} x + !_0)$$

$$= \begin{cases} +1 \text{ if } !^{!} x + !_0 > 0 \\ ! \ 1 \text{ if } !^{!} x + !_0 \text{ " } 0 \end{cases}$$

Linear classifiers

- Hypothesis class $\mathcal{H}$: set of **h ! H**
  - Example $\mathcal{H}$: All hypotheses that label +1 on one side of a line and -1 on the other side





- Linear classifier:

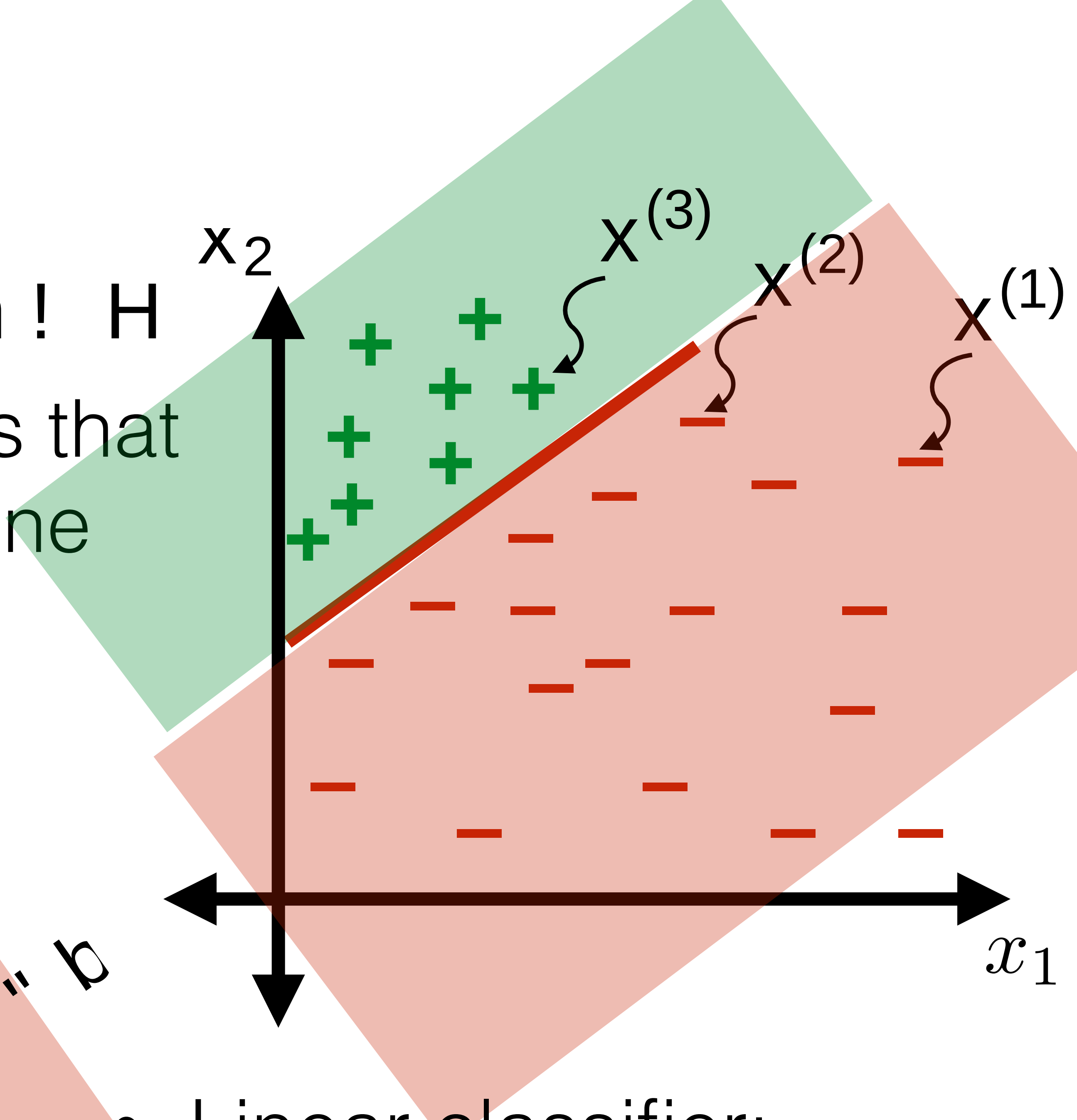
h(x; ! , ! 0) = sign(! ! x + ! 0)

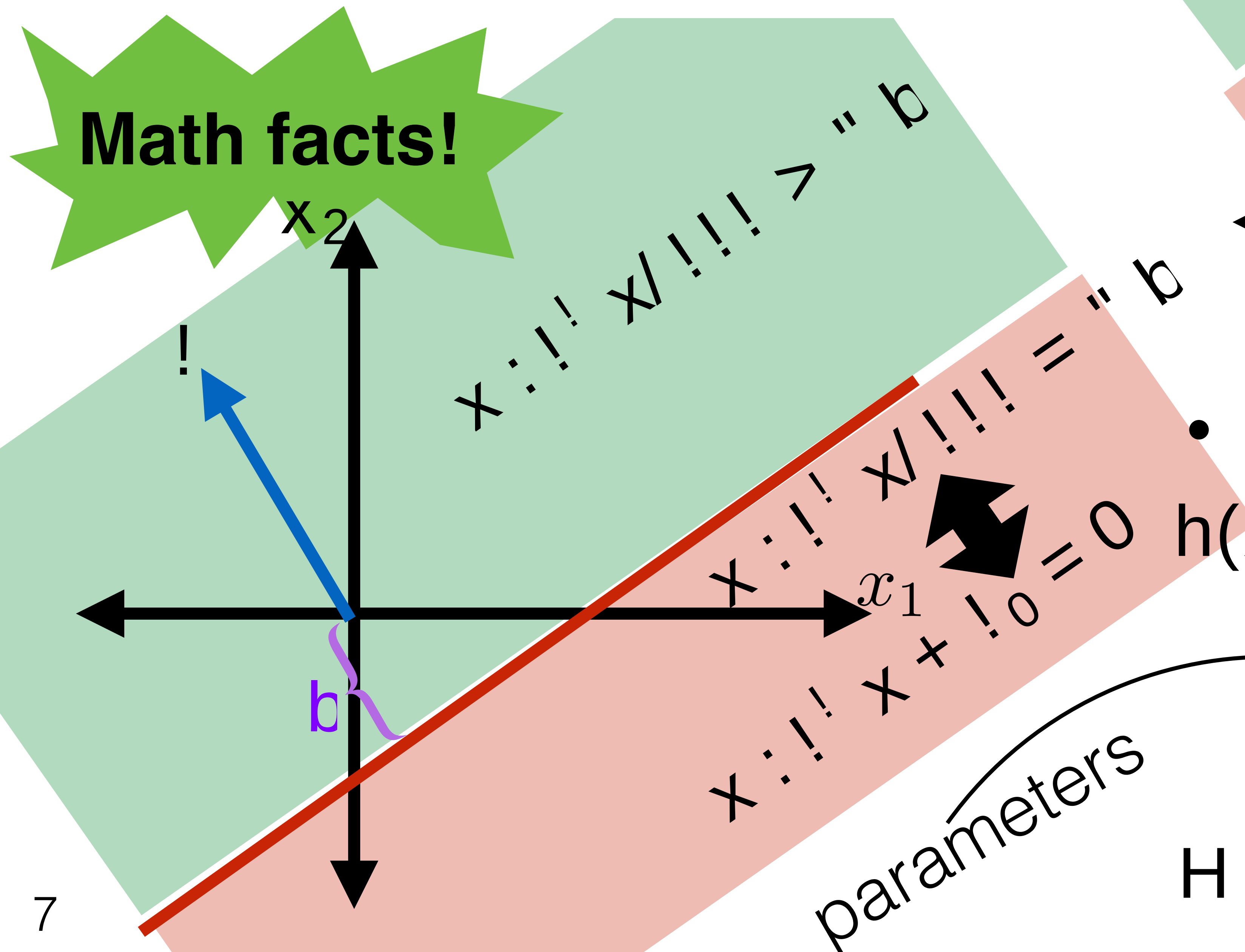
= +1 if ! ! x + ! 0 > 0
! 1 if ! ! x + ! 0 " 0

parameters

Linear classifiers

- Hypothesis class $\mathcal{H}$: set of **h ! H**
  - Example $\mathcal{H}$: All hypotheses that label +1 on one side of a line and -1 on the other side





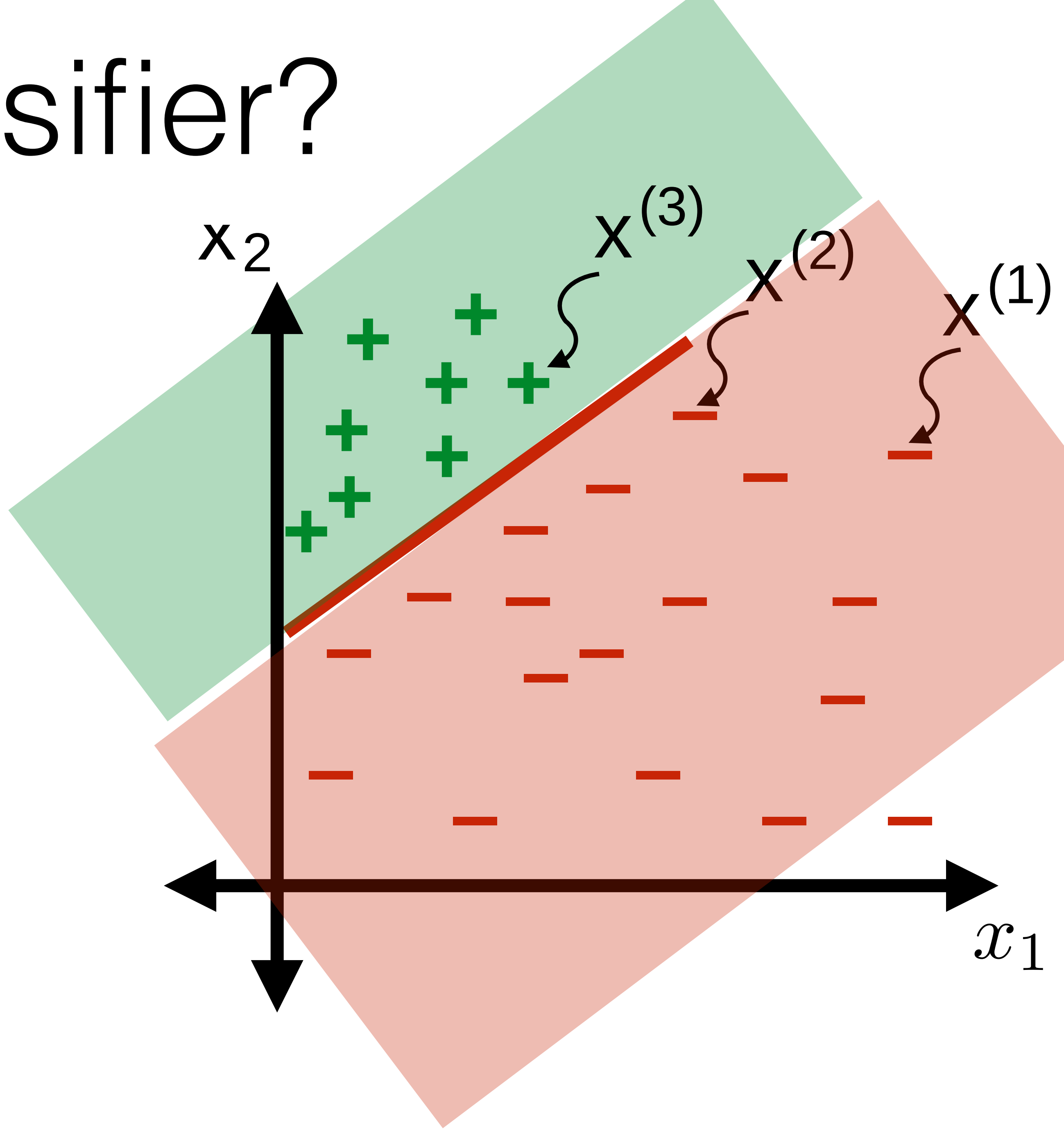
- Linear classifier:

$h(x;\,!\,,!_0) = \text{sign}(\,!^{\,!}\ x + !_0)$

$$= \begin{cases} +1 \text{ if } \ !^{\,!}\ x + !_0 > 0 \\ !\ 1 \text{ if } \ !^{\,!}\ x + !_0\ "\ 0 \end{cases}$$

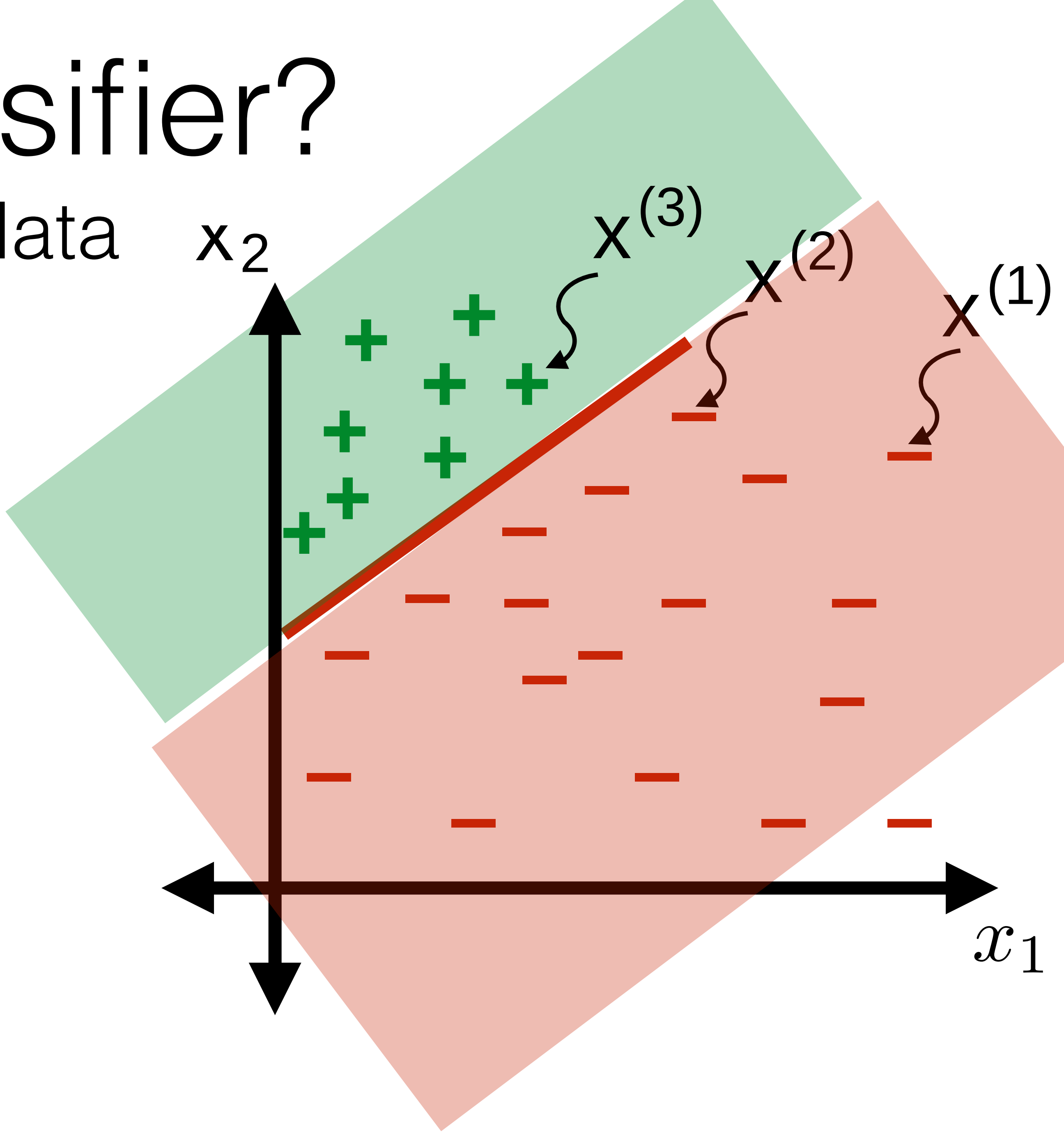
H = set of all such h

How good is a classifier?



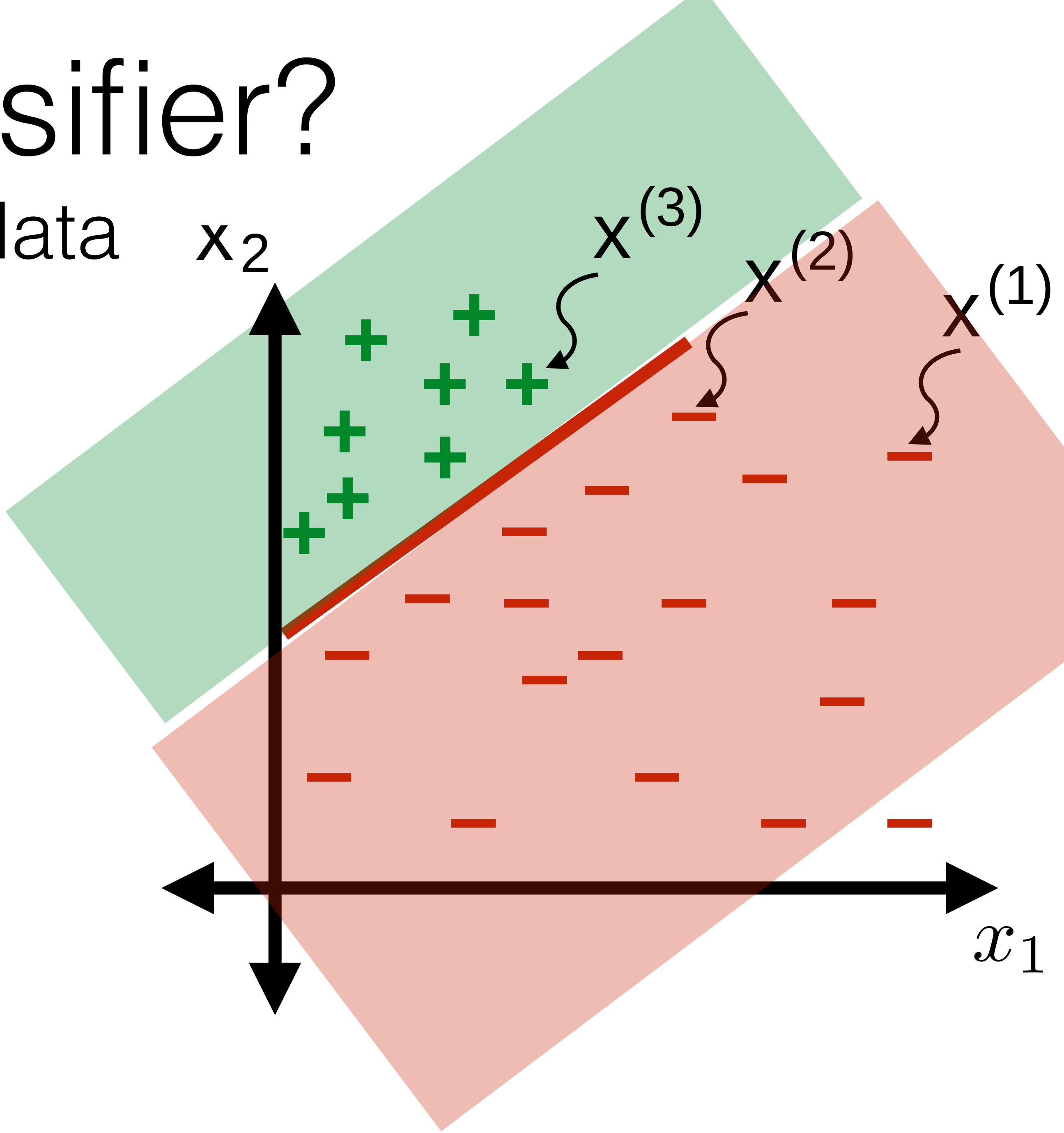
How good is a classifier?

- Should predict well on future data



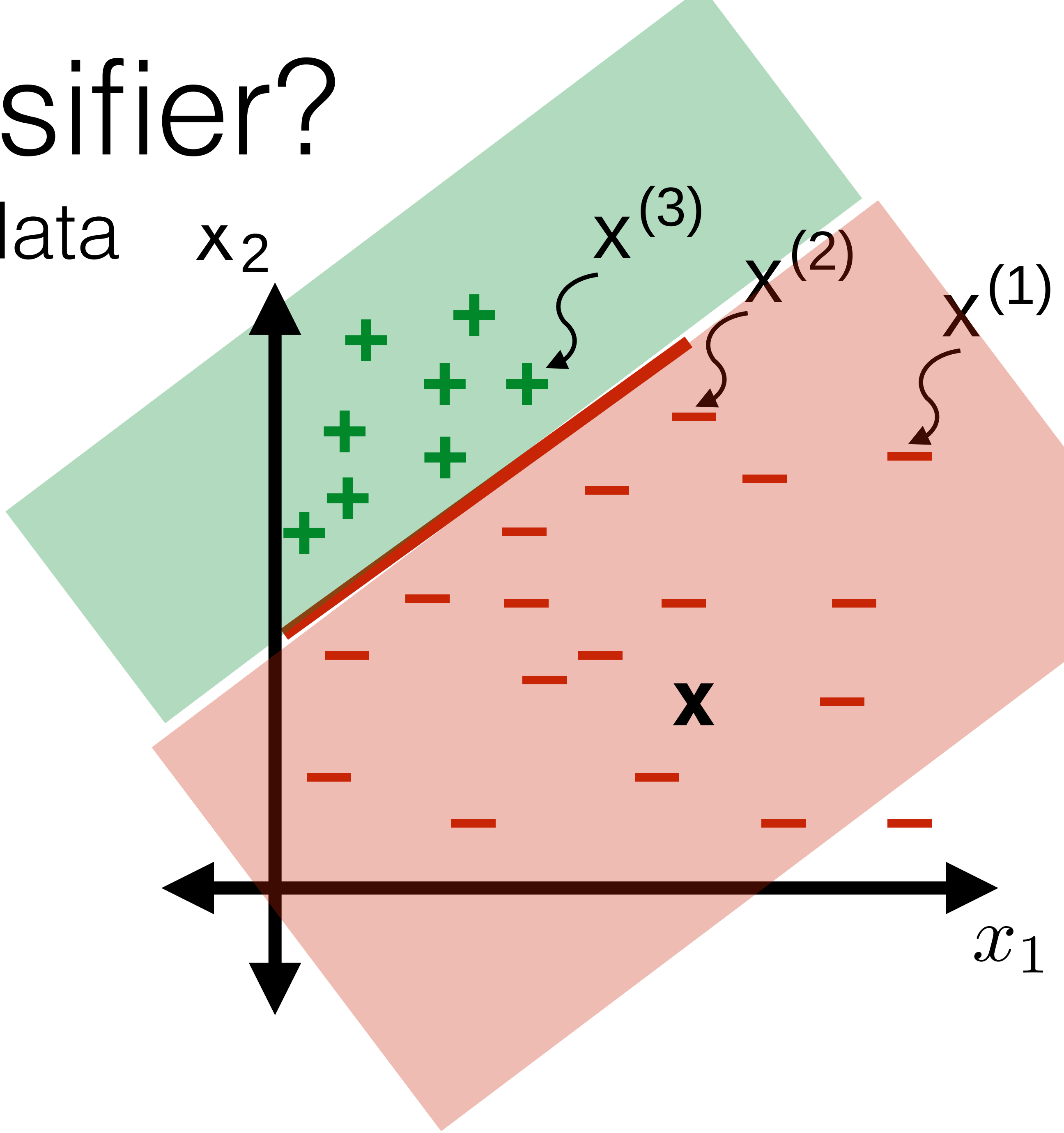
How good is a classifier?

- Should predict well on future data
- How good is a classifier at a single point?



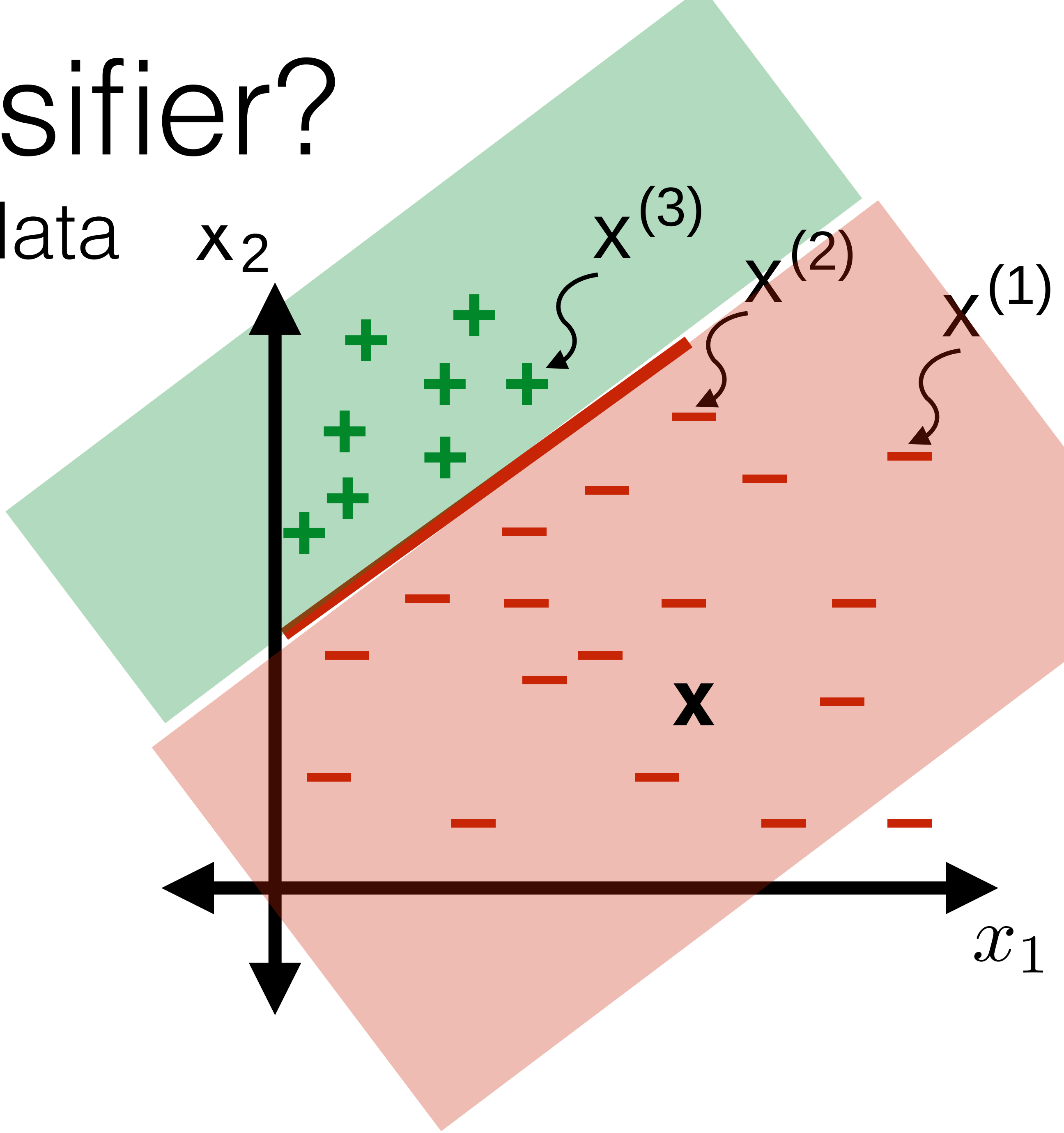
How good is a classifier?

- Should predict well on future data
- How good is a classifier at a single point?



How good is a classifier?

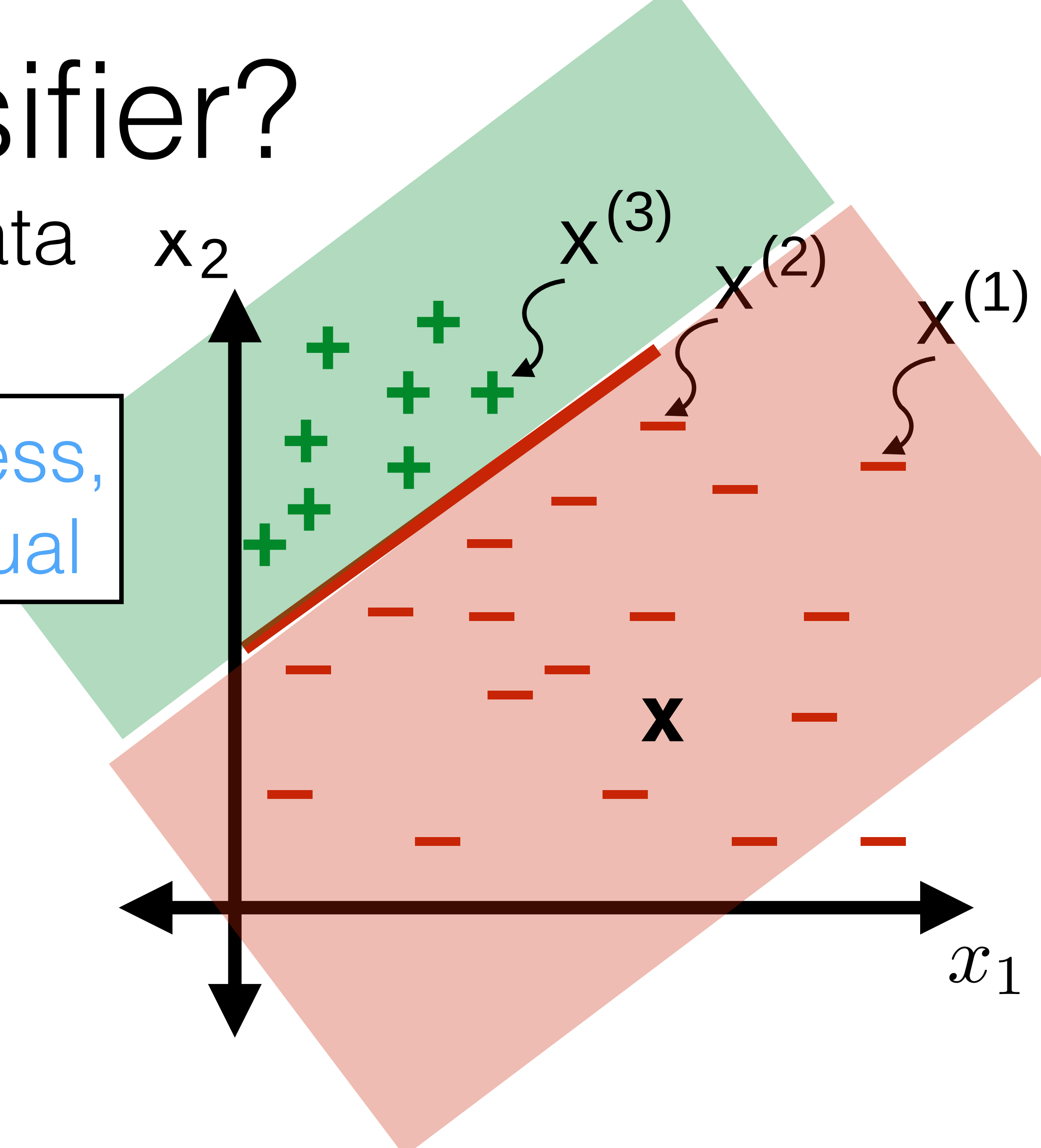
- Should predict well on future data
- How good is a classifier at a single point? Loss $L(g, a)$



How good is a classifier?

- Should predict well on future data
- How good is a classifier at a single point? Loss $L(g, a)$

g: guess,
a: actual

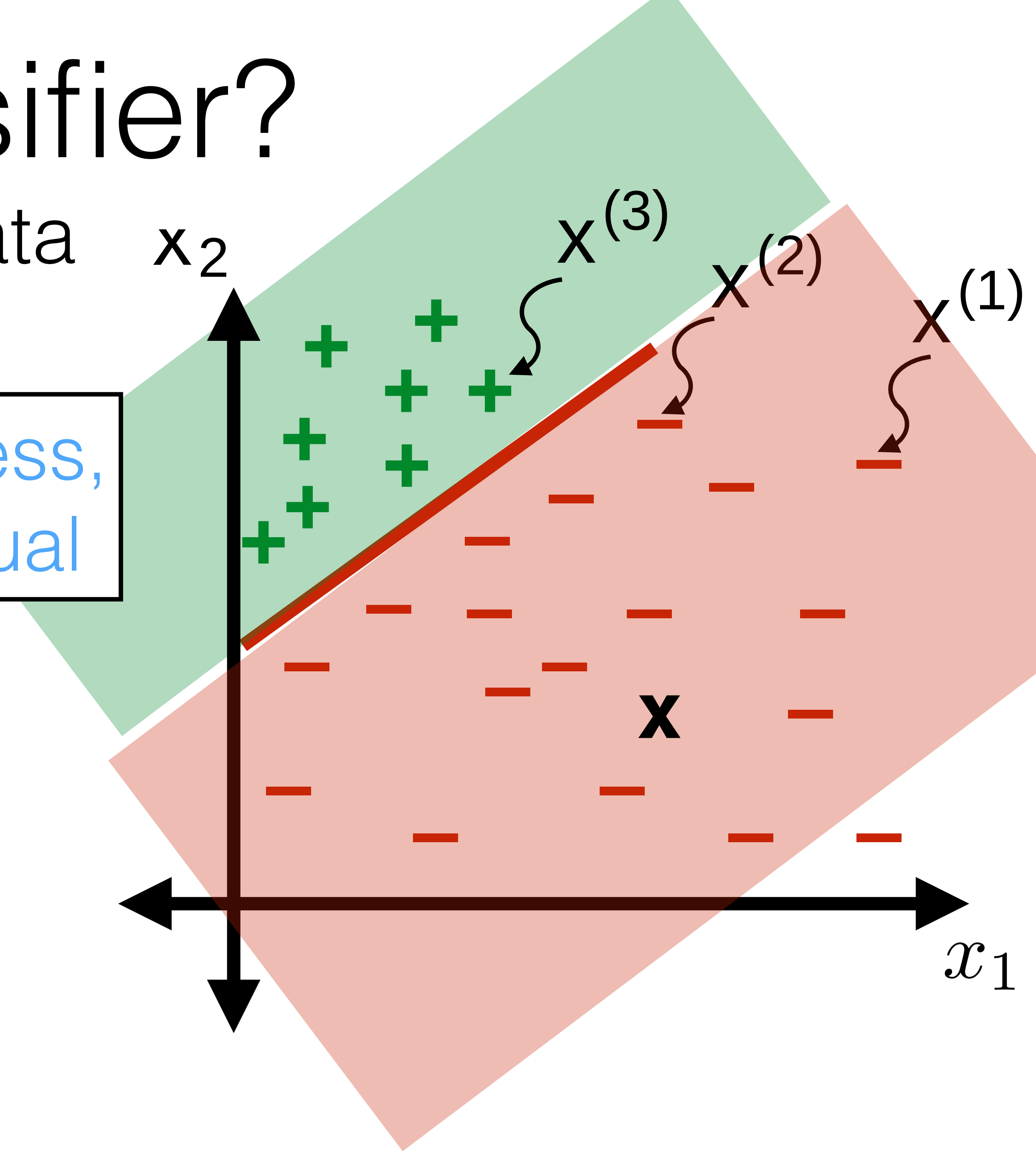


How good is a classifier?

- Should predict well on future data
- How good is a classifier at a single point? Loss $L(g, a)$

g: guess,
a: actual

  - Example: 0-1 loss



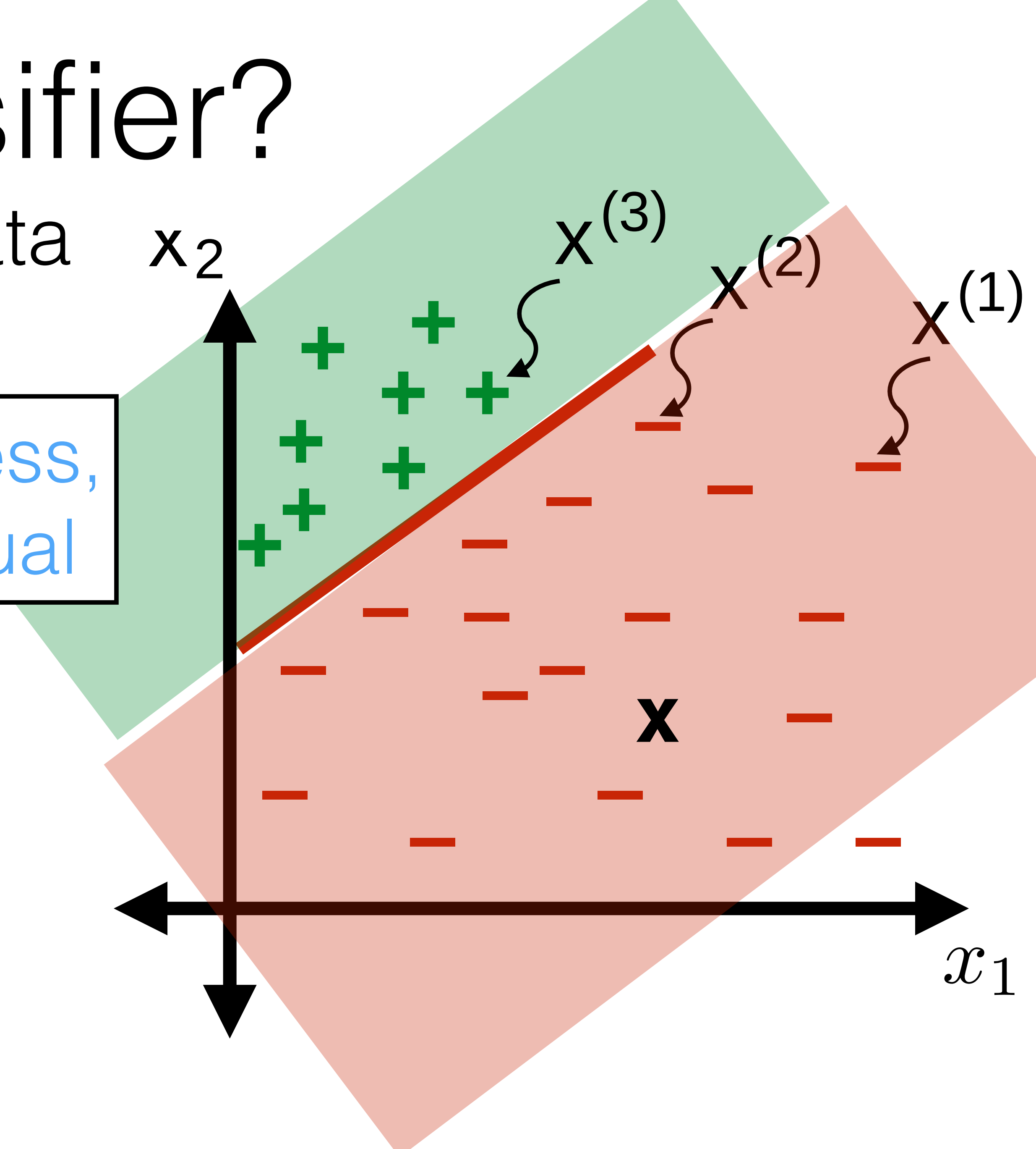
How good is a classifier?

- Should predict well on future data
- How good is a classifier at a single point? Loss $L(g, a)$

g: guess,
a: actual

  - Example: 0-1 loss

$$L(g, a) = \begin{cases} 0 & \text{if } g = a \\ 1 & \text{else} \end{cases}$$



How good is a classifier?

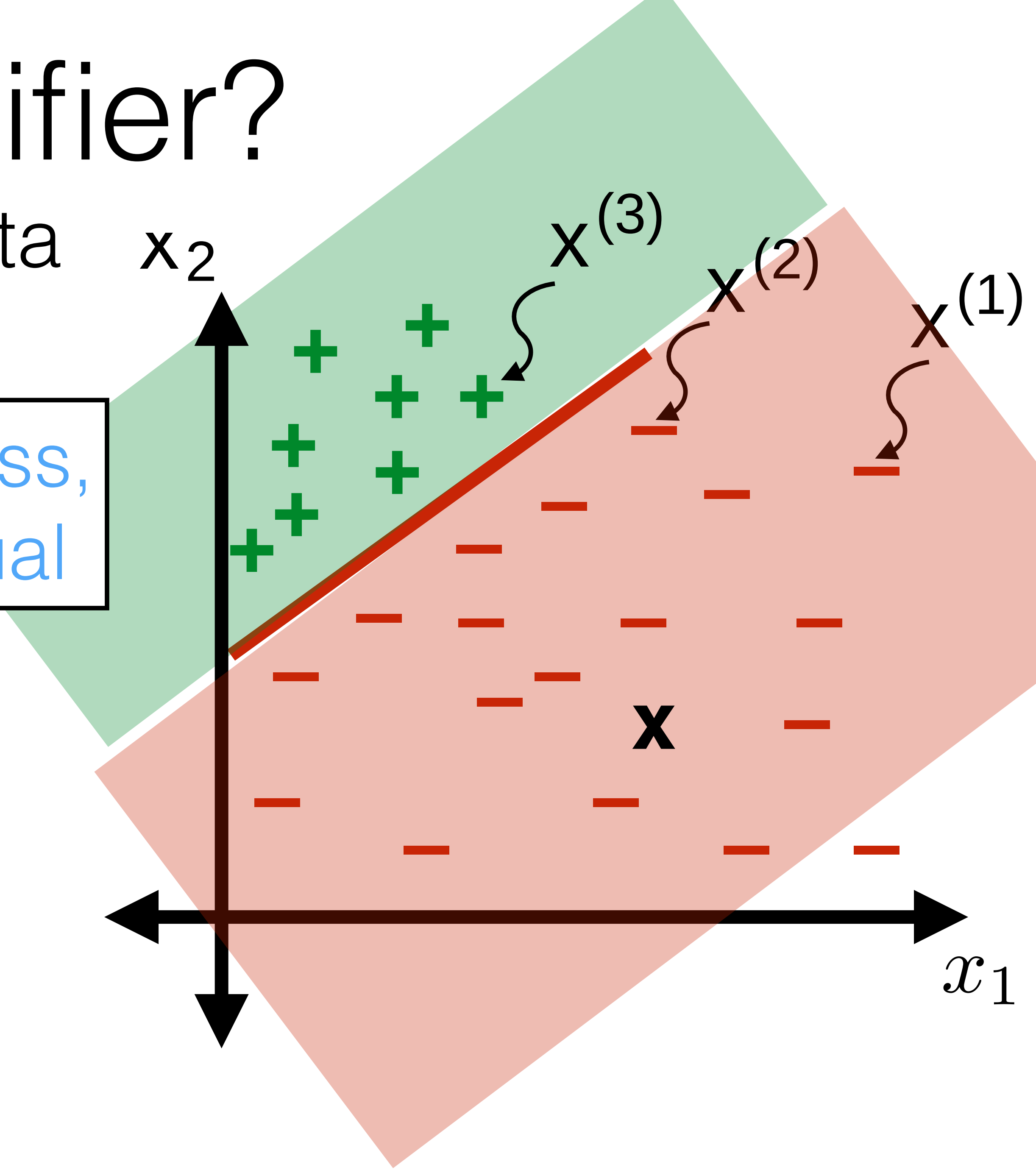
- Should predict well on future data
- How good is a classifier at a single point? Loss $L(g, a)$

g: guess,
a: actual

  - Example: 0-1 loss

$$L(g, a) = \begin{cases} 0 & \text{if } g = a \\ 1 & \text{else} \end{cases}$$

  - Example: asymmetric loss



How good is a classifier?

- Should predict well on future data
- How good is a classifier at a single point? Loss $L(g, a)$

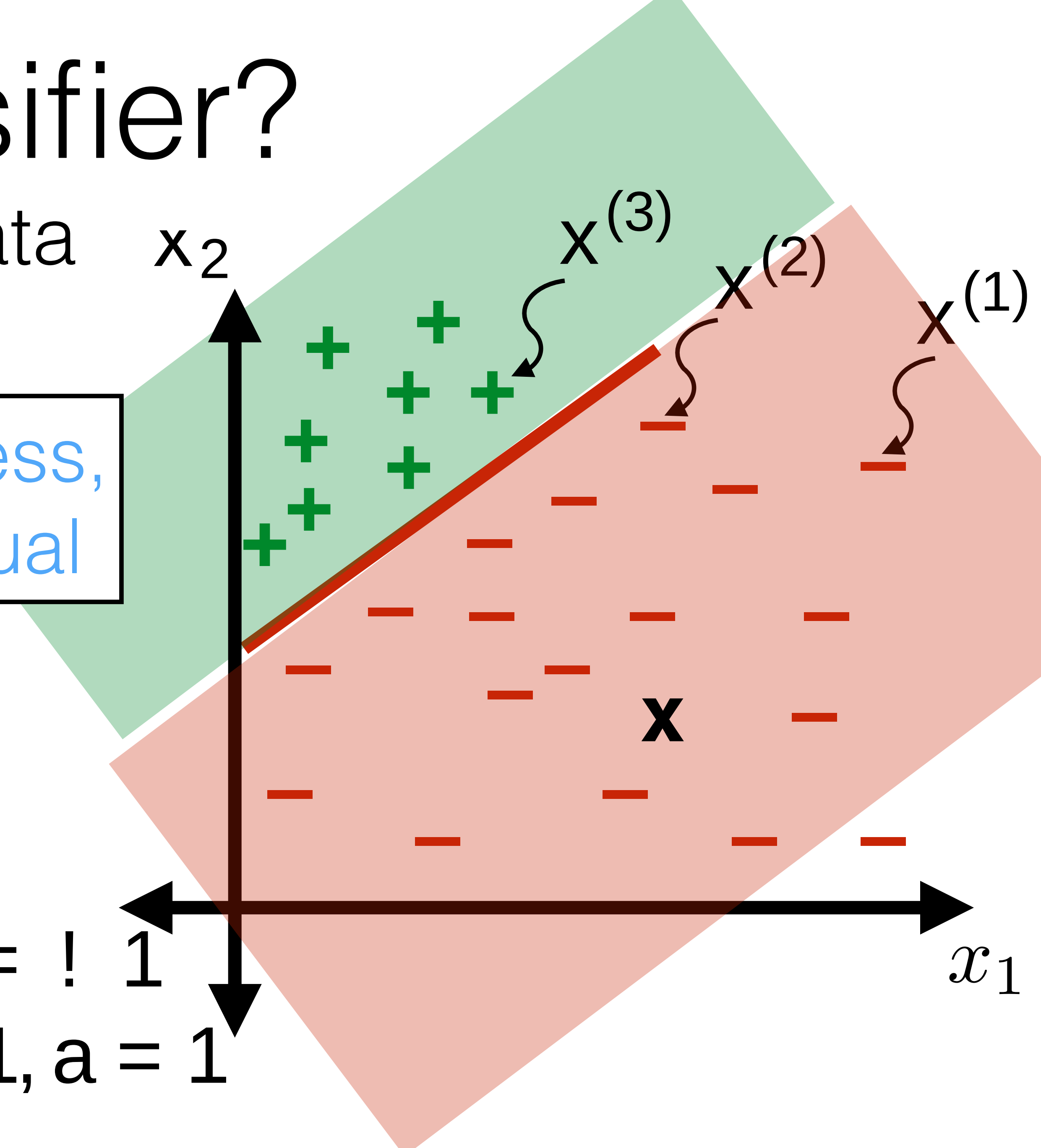
g: guess,
a: actual

  - Example: 0-1 loss

$$L(g, a) = \begin{cases} 0 \text{ if } g = a \\ 1 \text{ else} \end{cases}$$

  - Example: asymmetric loss

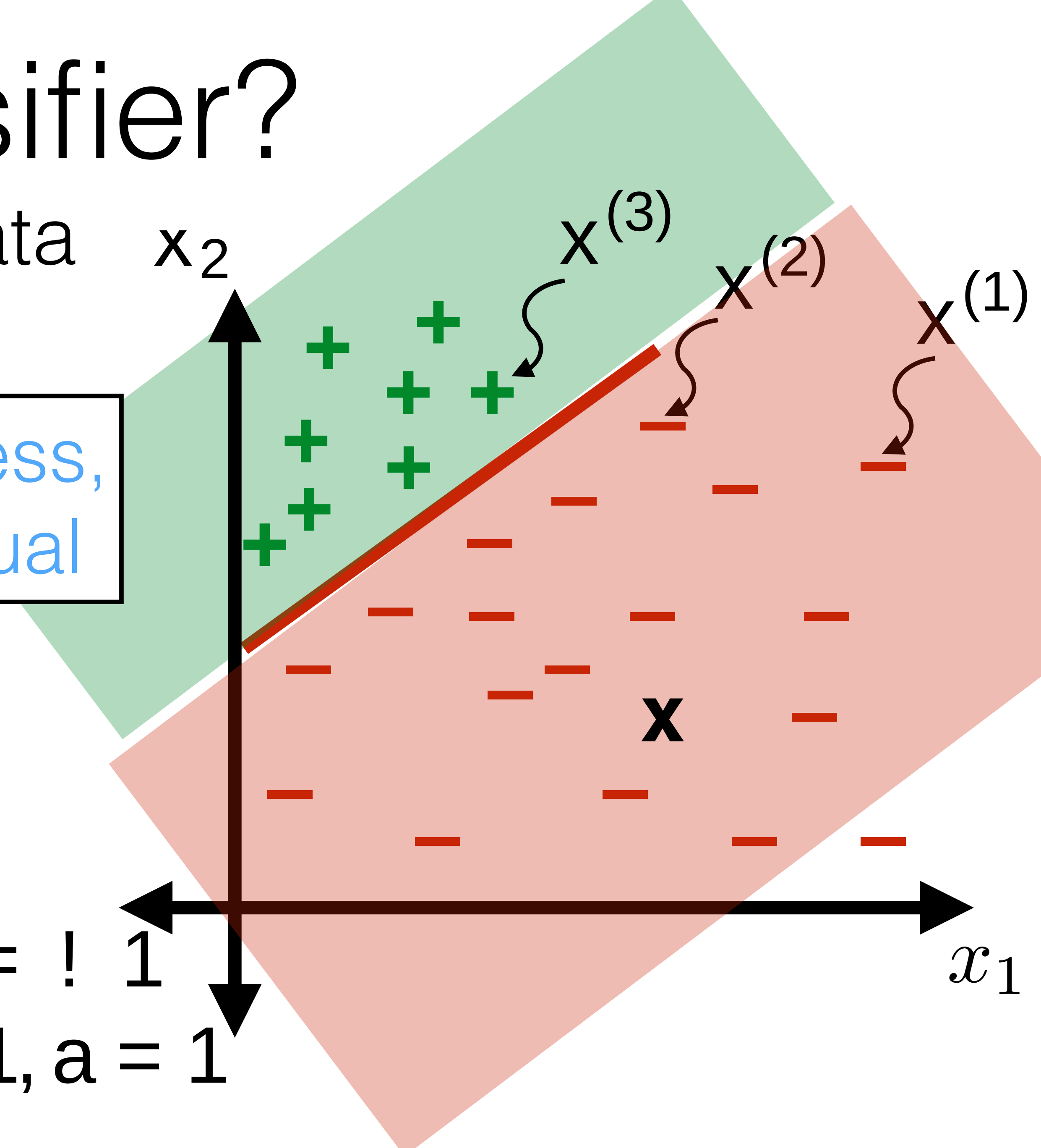
$$L(g, a) = \begin{cases} 1 \text{ if } g = 1, a \neq 1 \\ 100 \text{ if } g \neq 1, a = 1 \\ 0 \text{ else} \end{cases}$$



How good is a classifier?

- Should predict well on future data
- How good is a classifier at a single point? Loss $L(g, a)$

g: guess,
a: actual



  - Example: 0-1 loss

$$L(g, a) = \begin{cases} 0 \text{ if } g = a \\ 1 \text{ else} \end{cases}$$

  - Example: asymmetric loss

$$L(g, a) = \begin{cases} 1 \text{ if } g = 1, a = -1 \\ 100 \text{ if } g = -1, a = 1 \\ 0 \text{ else} \end{cases}$$

- Test error (n' new points):

How good is a classifier?

- Should predict well on future data
- How good is a classifier at a single point? Loss $L(g, a)$

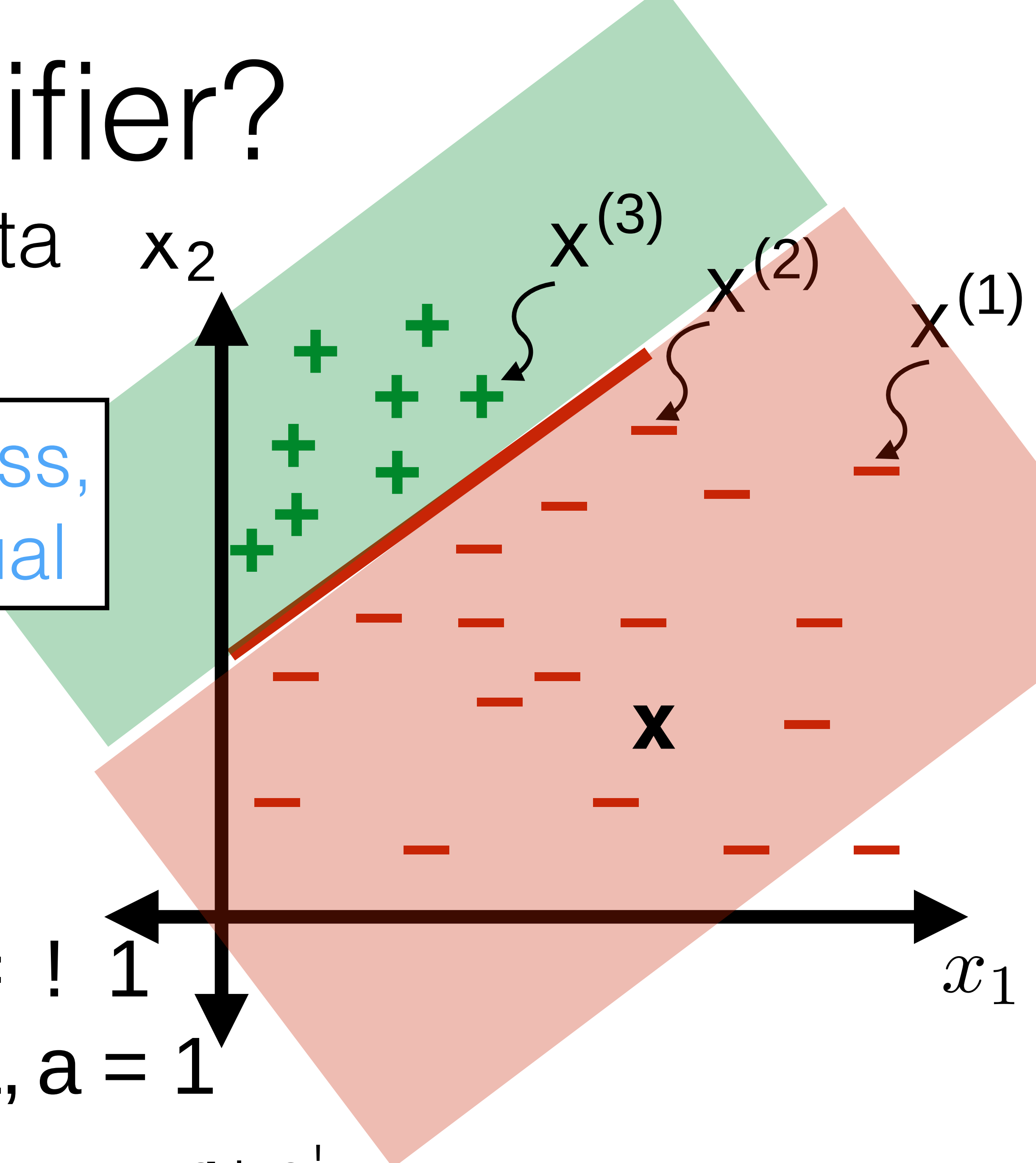
g: guess,
a: actual

  - Example: 0-1 loss

$$L(g,a) = \begin{cases} 0 \text{ if } g = a \\ 1 \text{ else} \end{cases}$$

  - Example: asymmetric loss

$$L(g,a) = \begin{cases} 1 \text{ if } g = 1, a = -1 \\ 100 \text{ if } g = -1, a = 1 \\ 0 \text{ else} \end{cases}$$



- Test error (n' new points): $E(h) = \frac{1}{n'} \sum_{i=n+1}^{n+n'} L(h(x^{(i)}), y^{(i)})$

How good is a classifier?

- Should predict well on future data
- How good is a classifier at a single point? Loss $L(g,a)$

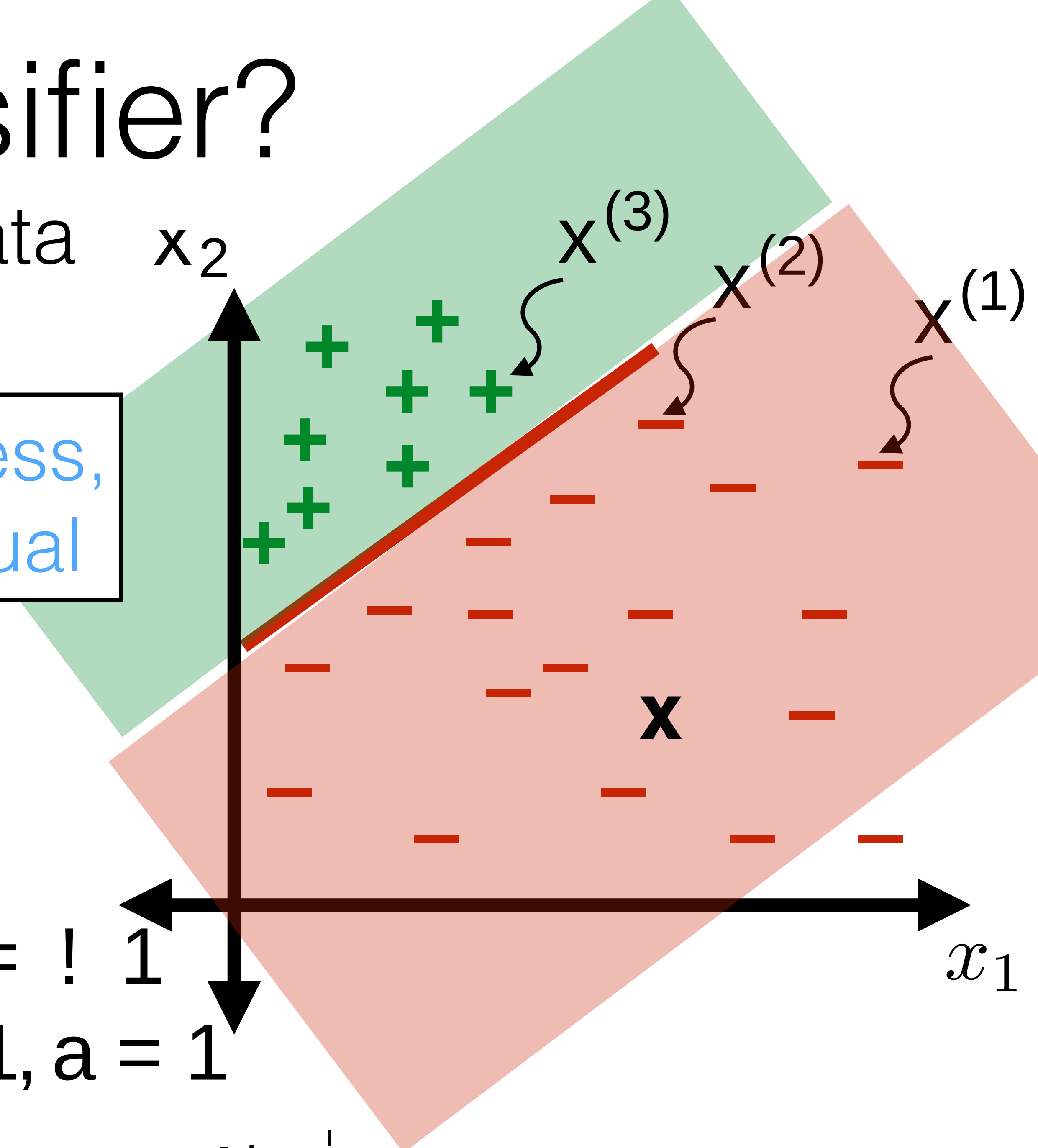
g: guess,
a: actual

  - Example: 0-1 loss

$$L(g,a) = \begin{cases} 0 \text{ if } g = a \\ 1 \text{ else} \end{cases}$$

  - Example: asymmetric loss

$$L(g,a) = \begin{cases} 1 \text{ if } g = 1, a = -1 \\ 100 \text{ if } g = -1, a = 1 \\ 0 \text{ else} \end{cases}$$



- Test error (n' new points): $E(h) = \frac{1}{n'} \sum_{i=n+1}^{n+n'} L(h(x^{(i)}), y^{(i)})$
- Training error:

How good is a classifier?

- Should predict well on future data
- How good is a classifier at a single point? Loss $L(g, a)$

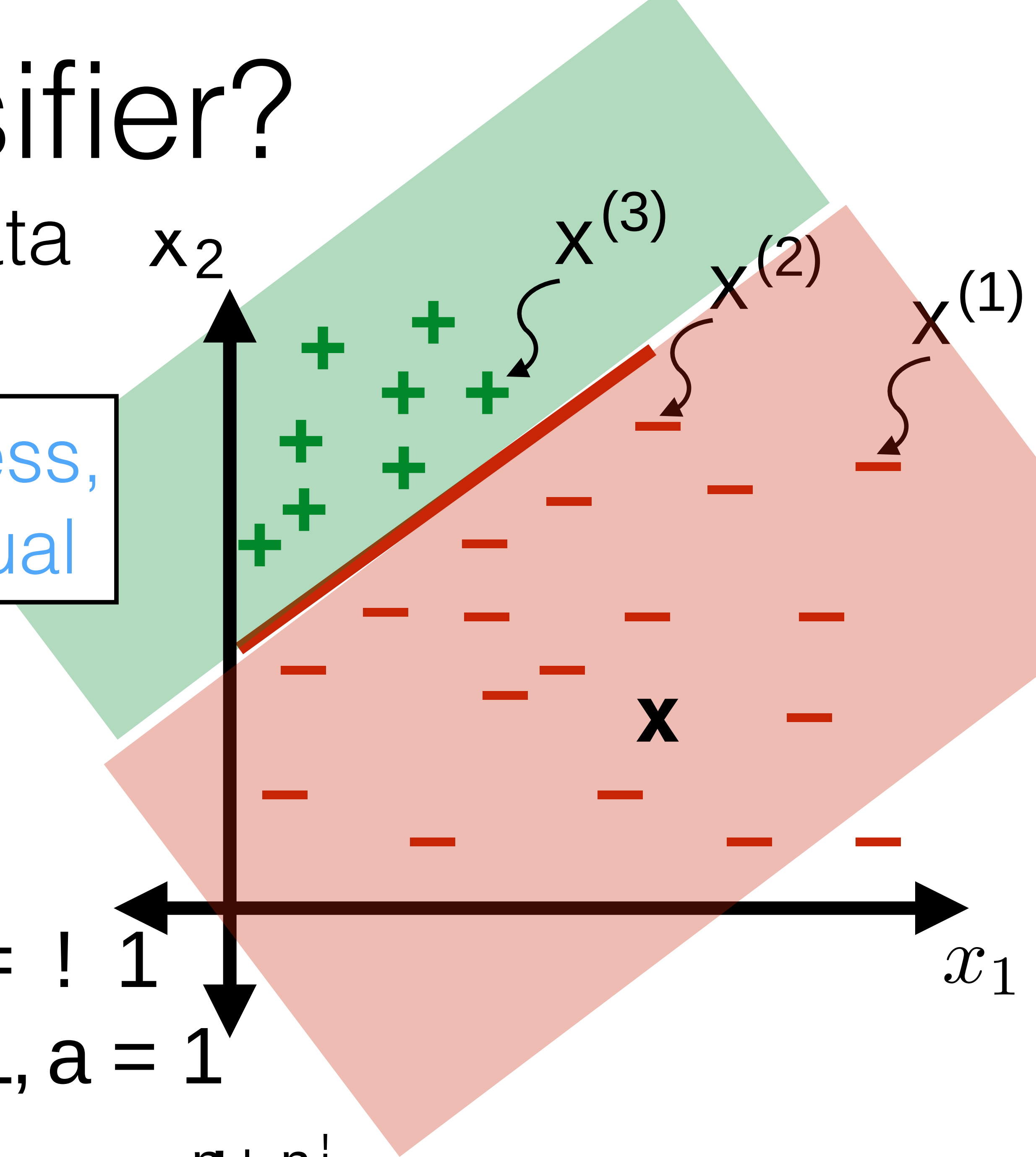
g: guess,
a: actual

  - Example: 0-1 loss

$$L(g,a) = \begin{cases} 0 \text{ if } g = a \\ 1 \text{ else} \end{cases}$$

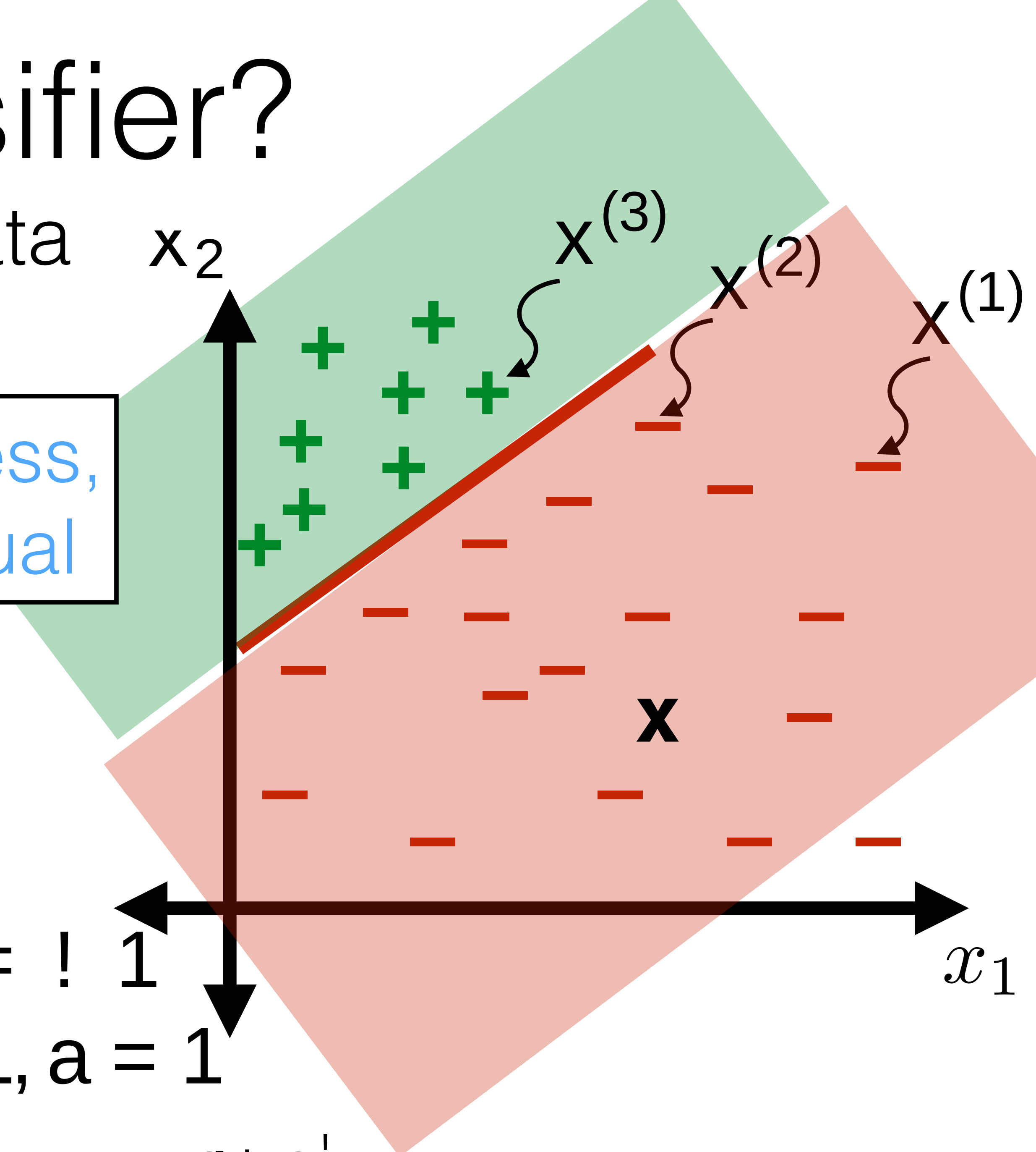
  - Example: asymmetric loss

$$L(g,a) = \begin{cases} 1 \text{ if } g = 1, a \neq 1 \\ 100 \text{ if } g \neq 1, a = 1 \\ 0 \text{ else} \end{cases}$$



- Test error (n' new points): $E(h) = \frac{1}{n'} \sum_{i=n+1}^{n+n'} L(h(x^{(i)}), y^{(i)})$
- Training error: $E_n(h) = \frac{1}{n} \sum_{i=1}^{n} L(h(x^{(i)}), y^{(i)})$

How good is a classifier?



- Should predict well on future data
- How good is a classifier at a single point? Loss $L(g, a)$

g: guess,
a: actual

  - Example: 0-1 loss

$$L(g,a) = \begin{cases} 0 & \text{if } g = a \\ 1 & \text{else} \end{cases}$$

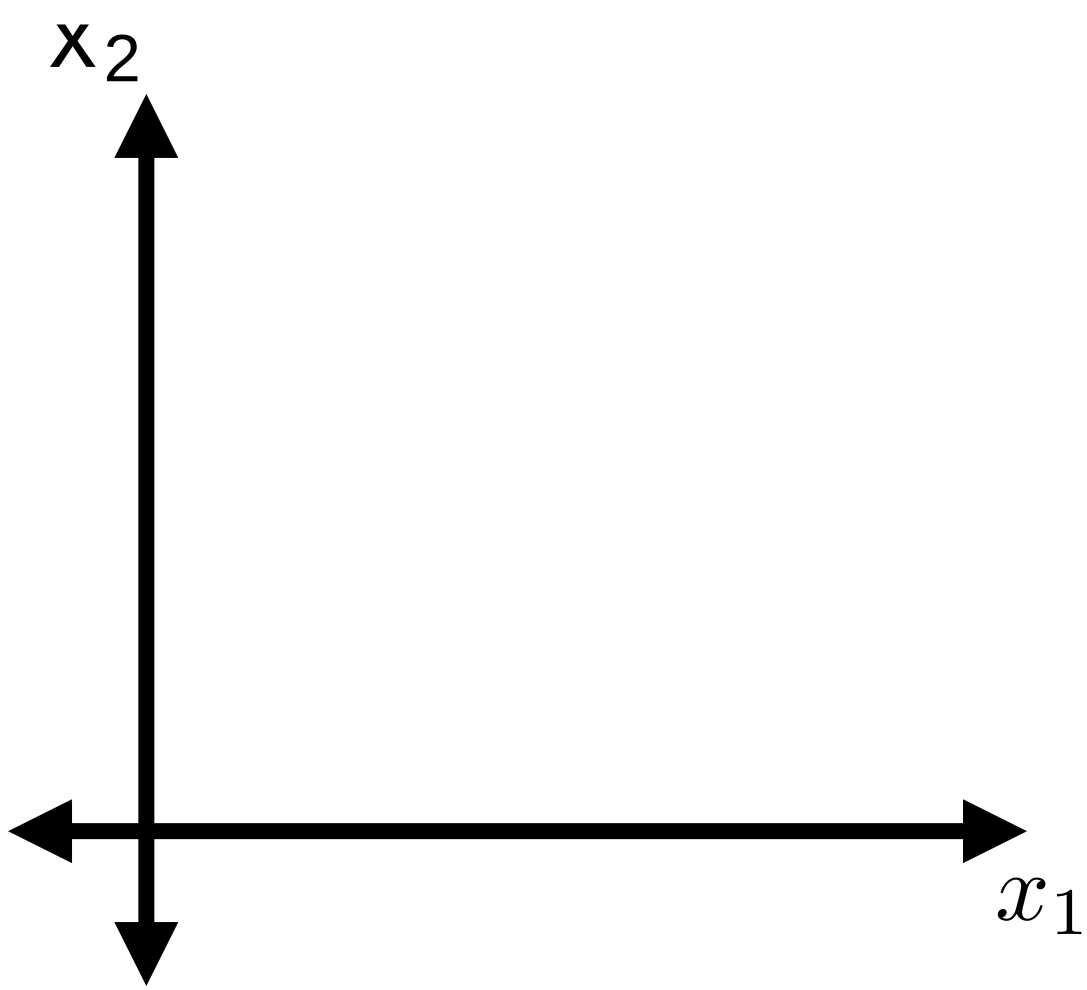
  - Example: asymmetric loss

$$L(g,a) = \begin{cases} 1 & \text{if } g = 1, a = -1 \\ 100 & \text{if } g = -1, a = 1 \\ 0 & \text{else} \end{cases}$$

- Test error (n' new points): $E(h) = \frac{1}{n'} \sum_{i=n+1}^{n+n'} L(h(x^{(i)}), y^{(i)})$
- Training error: $E_n(h) = \frac{1}{n} \sum_{i=1}^{n} L(h(x^{(i)}), y^{(i)})$
- Prefer h to $\tilde{h}$ if $E_n(h) < E_n(\tilde{h})$

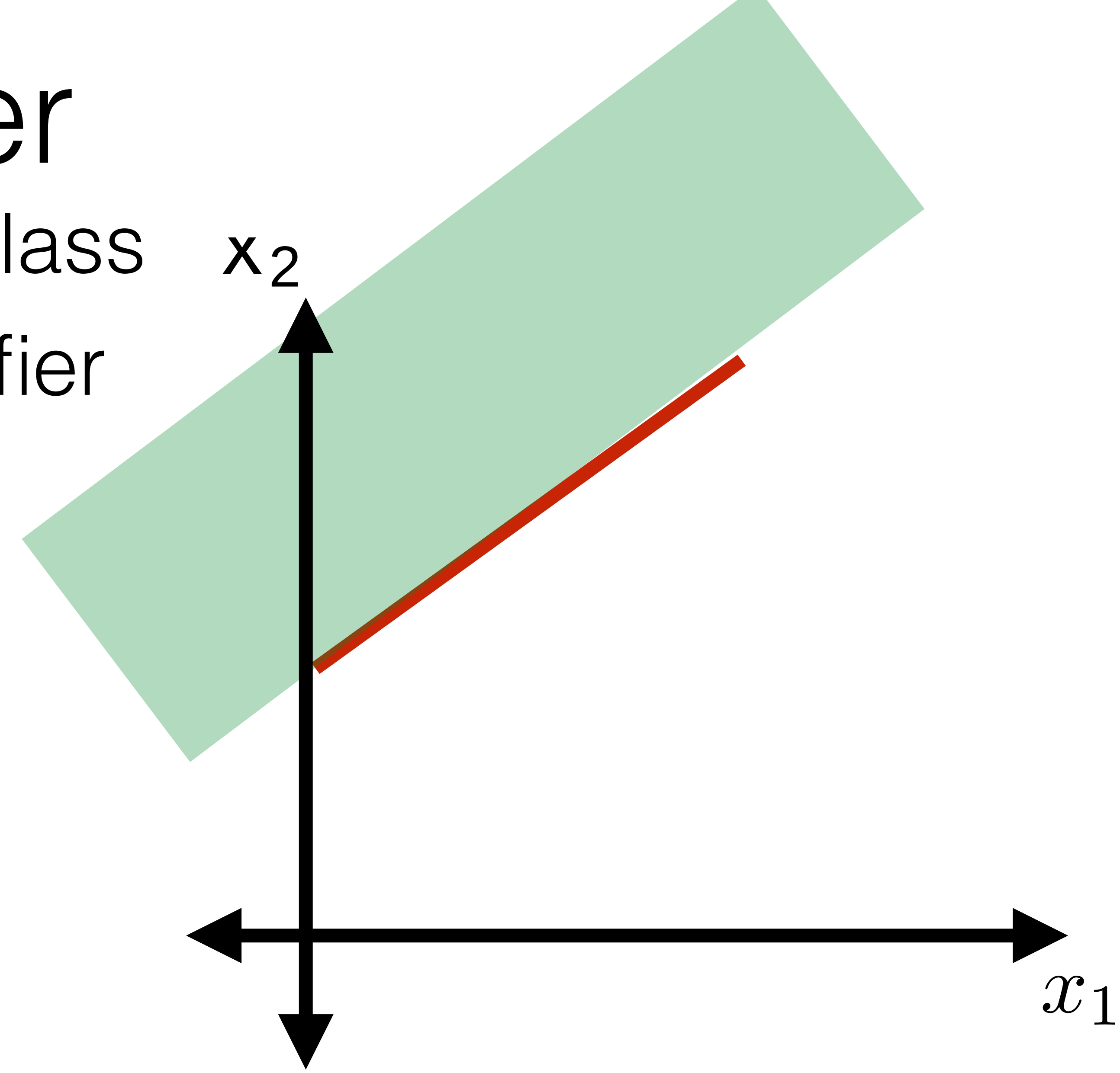
Learning a classifier

- Have data; have hypothesis class
- Want to choose a good classifier
    - Recall: $x \rightarrow \boxed{h} \rightarrow y$



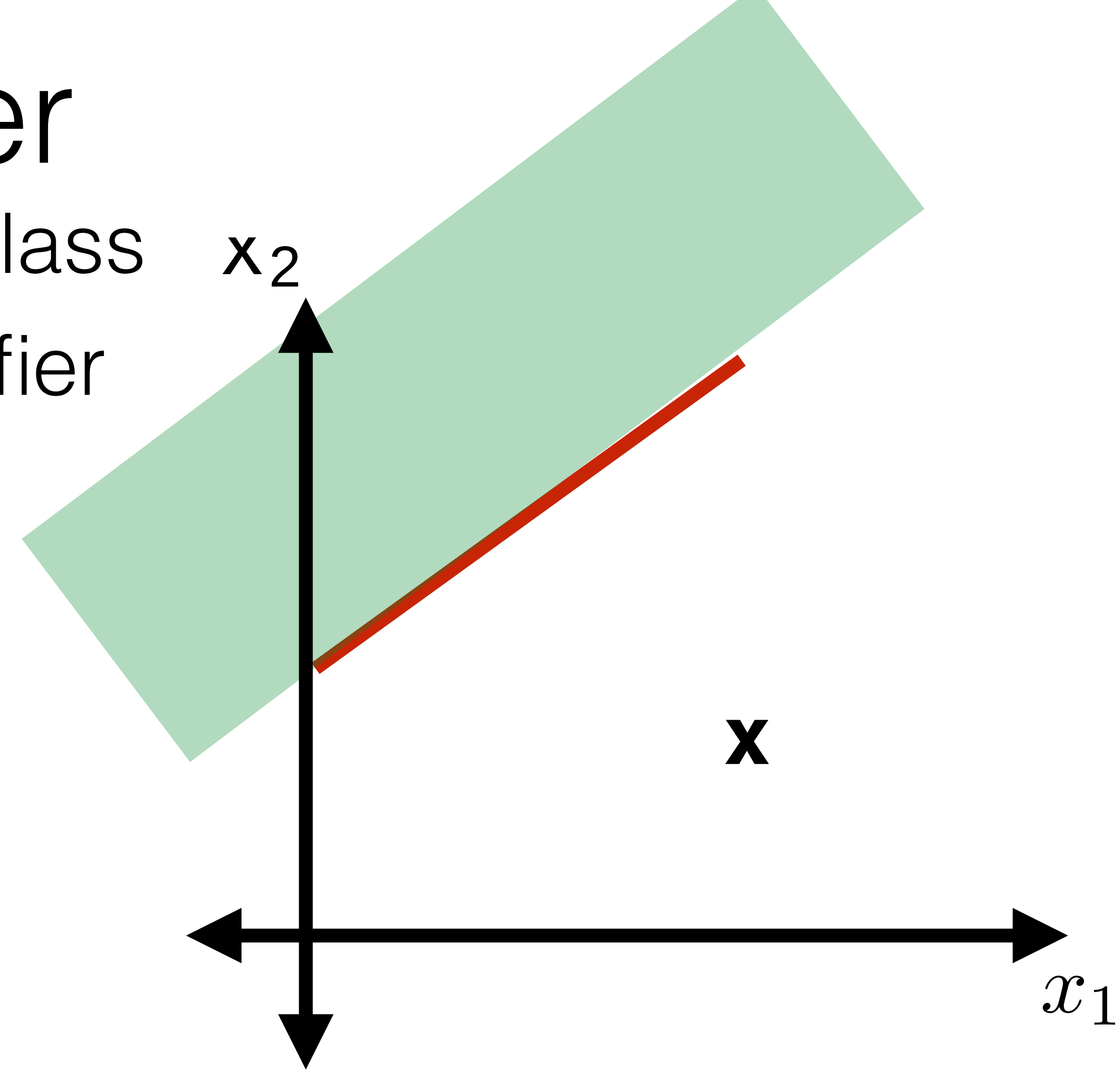
Learning a classifier

- Have data; have hypothesis class
- Want to choose a good classifier
  - Recall: $x \rightarrow \boxed{h} \rightarrow y$



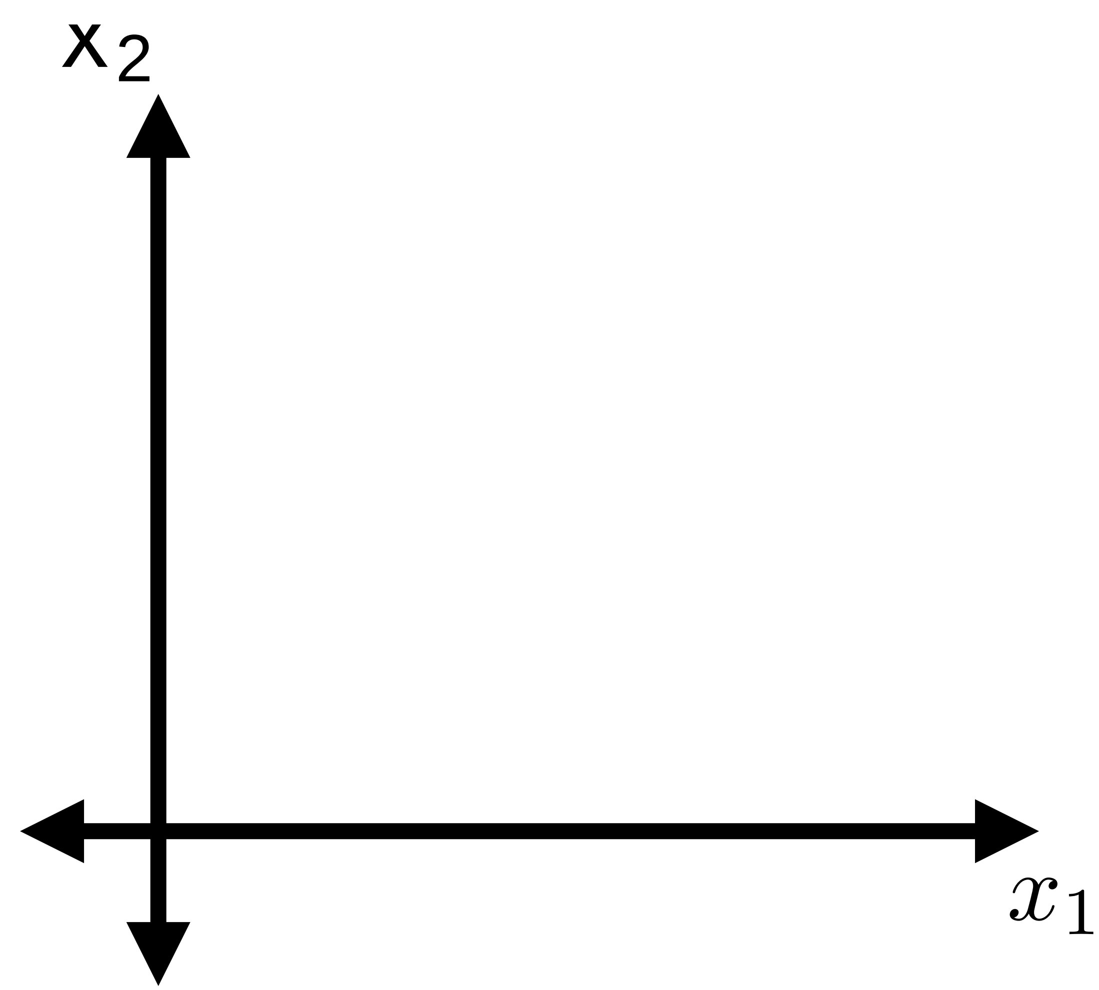
Learning a classifier

- Have data; have hypothesis class
- Want to choose a good classifier
  - Recall: $x \rightarrow \boxed{h} \rightarrow y$



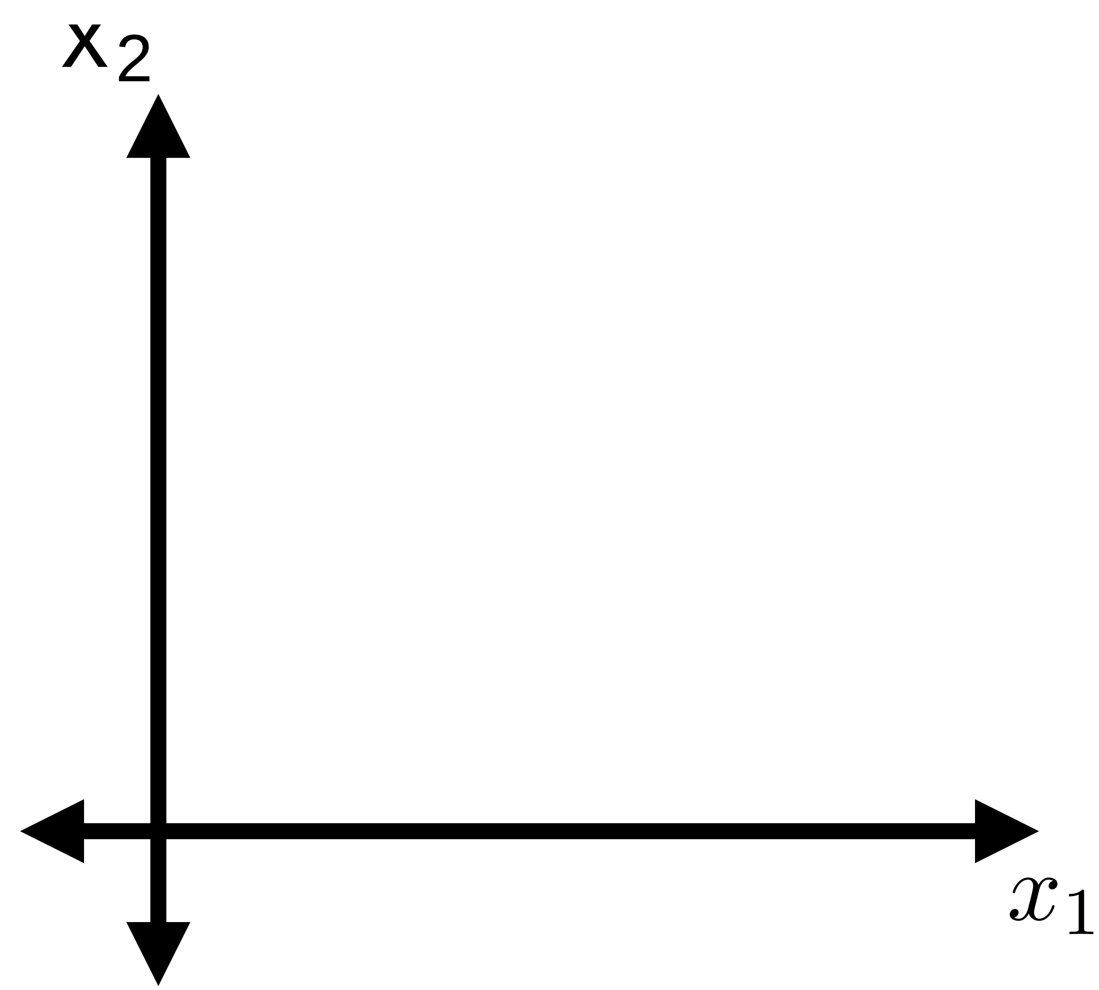
Learning a classifier

- Have data; have hypothesis class
- Want to choose a good classifier
  - Recall: $x \rightarrow \boxed{h} \rightarrow y$



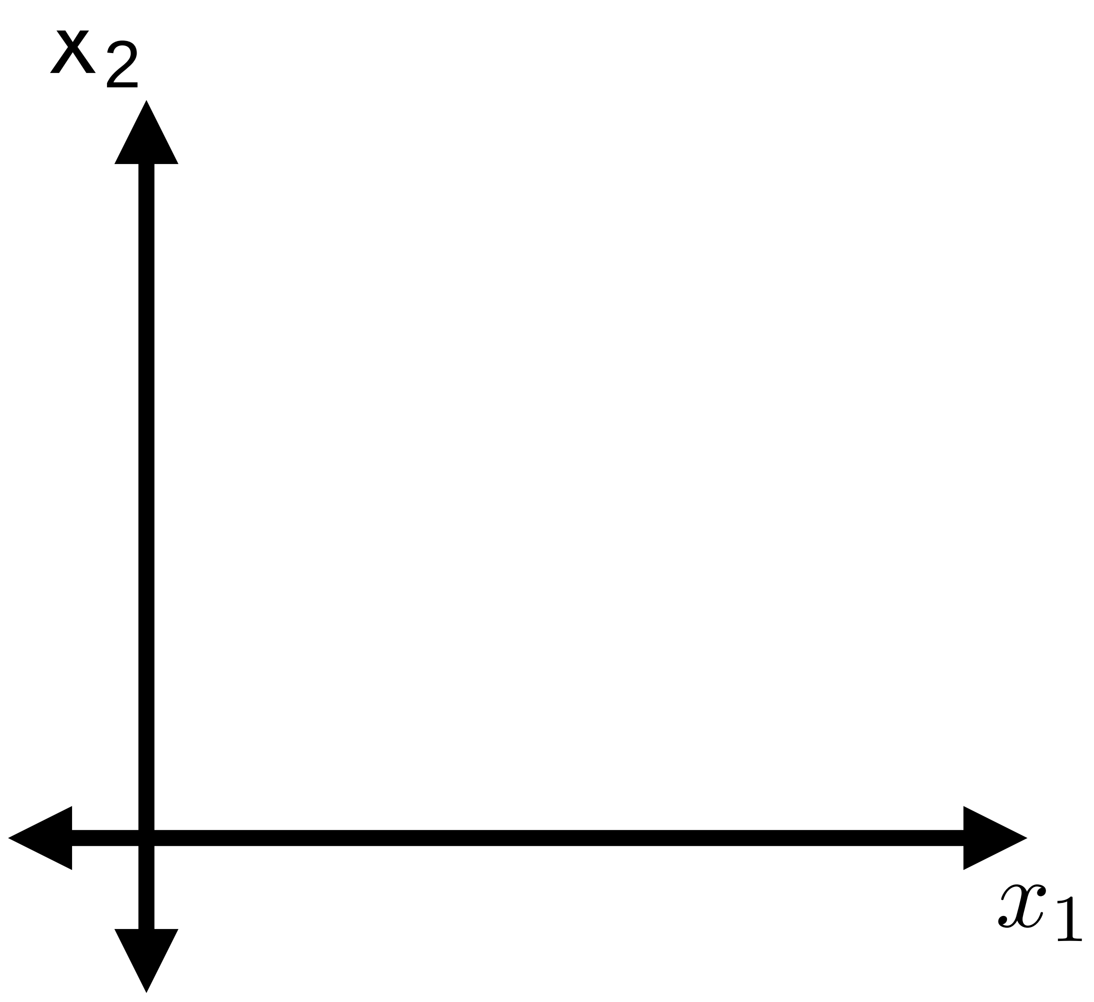
Learning a classifier

- Have data; have hypothesis class
- Want to choose a good classifier
    - Recall: $x \rightarrow \boxed{h} \rightarrow y$
    - New:



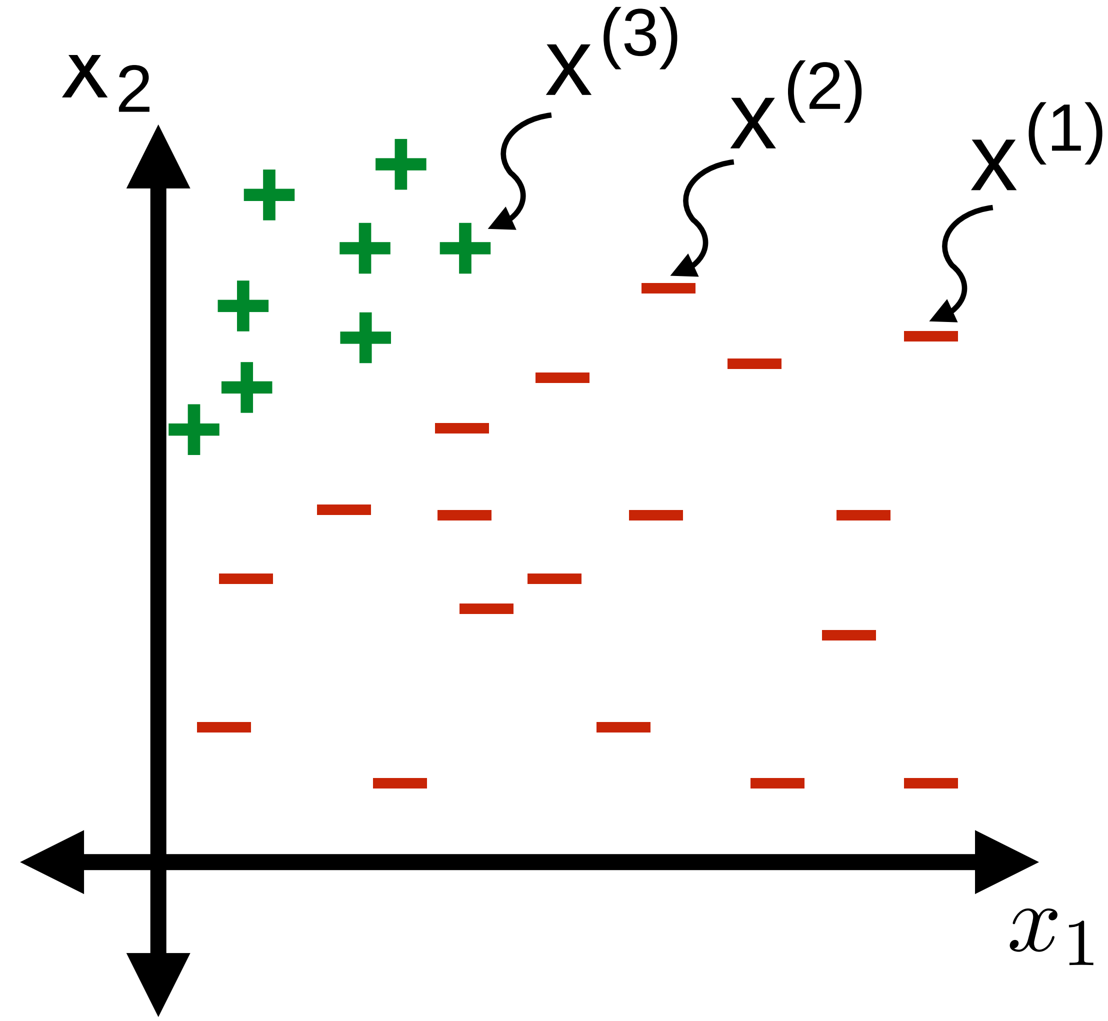
Learning a classifier

- Have data; have hypothesis class
- Want to choose a good classifier
  - Recall: x → h → y
  - New: D_n → learning algorithm → h



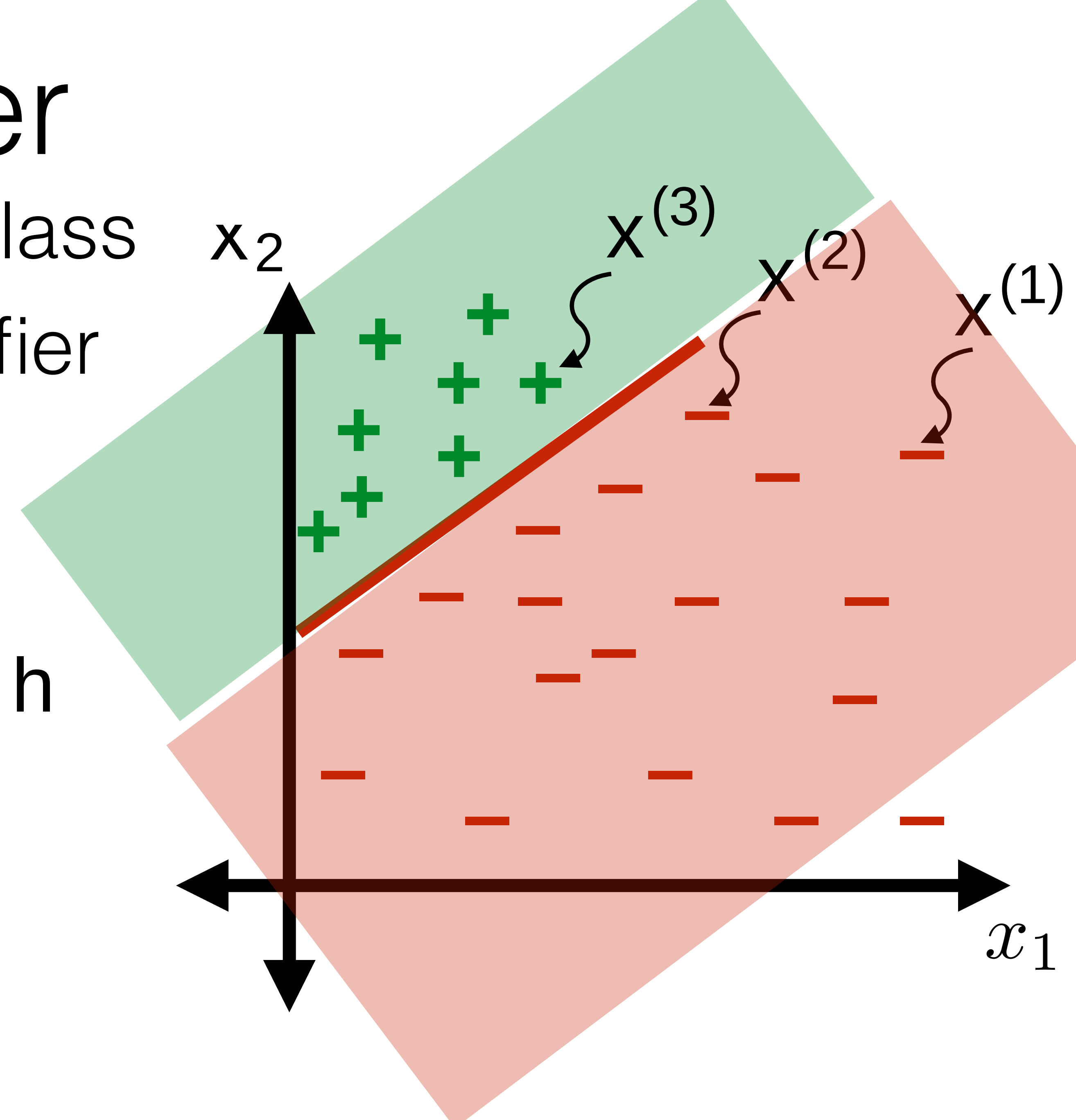
Learning a classifier

- Have data; have hypothesis class
- Want to choose a good classifier
  - Recall: $x \rightarrow \boxed{h} \rightarrow y$
  - New: $D_n \rightarrow \boxed{\text{learning algorithm}} \rightarrow h$



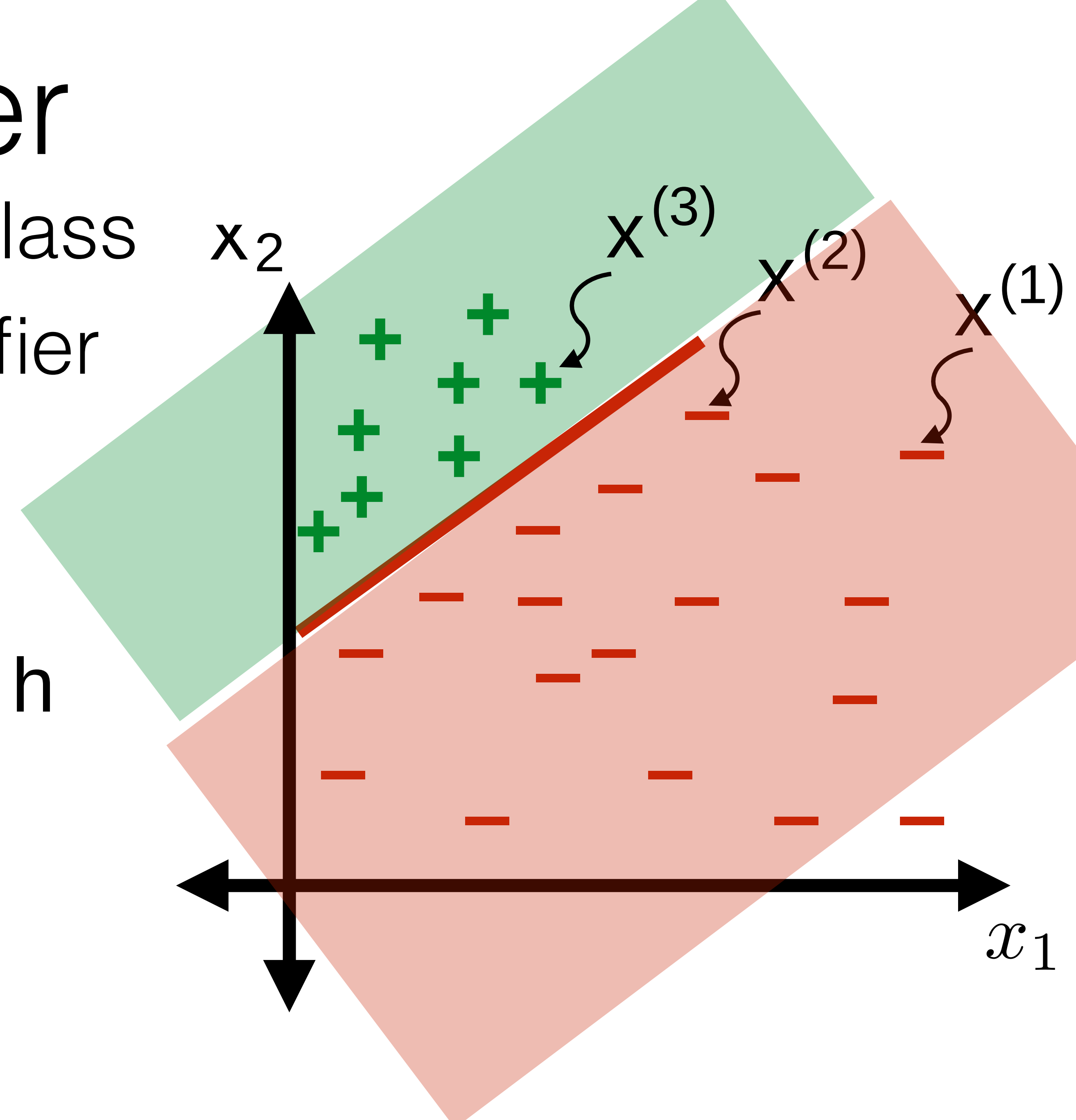
Learning a classifier

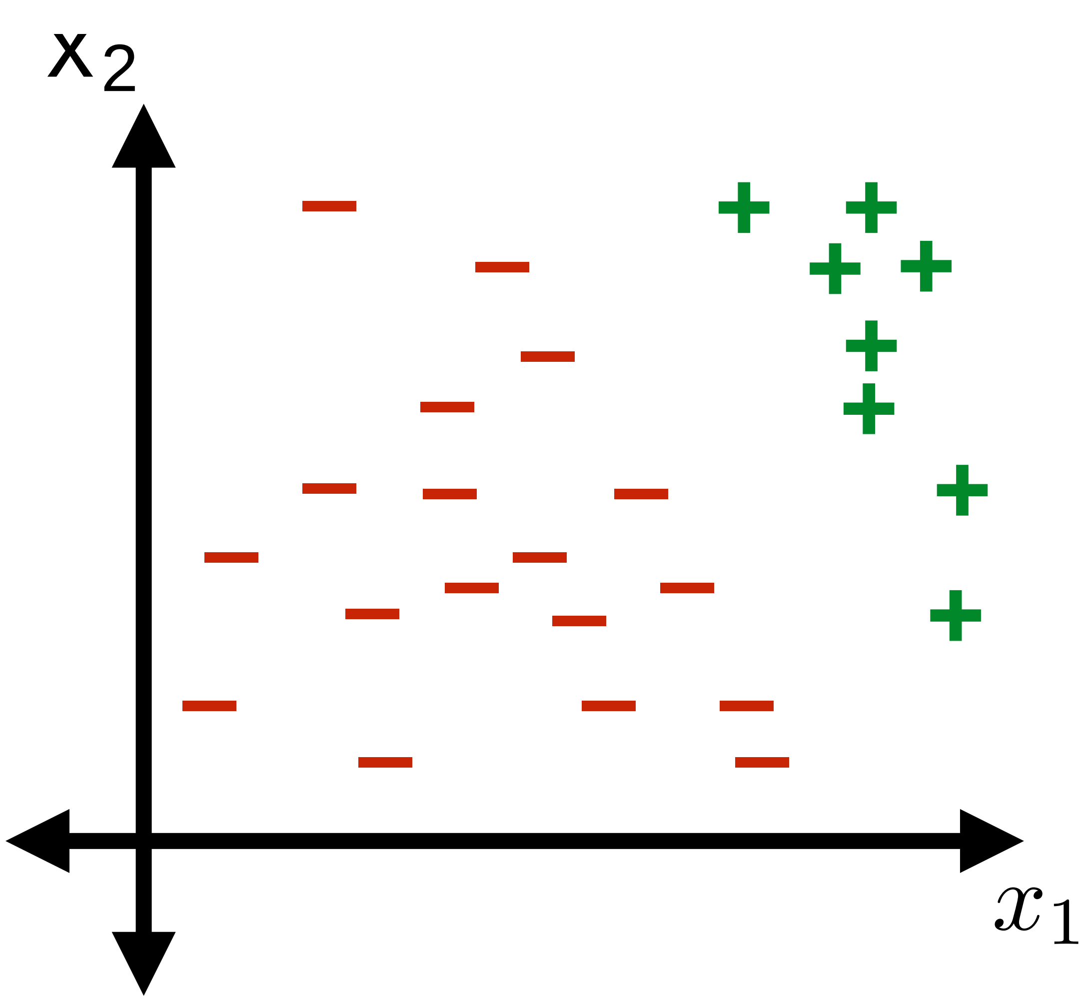
- Have data; have hypothesis class
- Want to choose a good classifier
  - Recall: $x \rightarrow \boxed{h} \rightarrow y$
  - New: $D_n \rightarrow \boxed{\text{learning algorithm}} \rightarrow h$



Learning a classifier

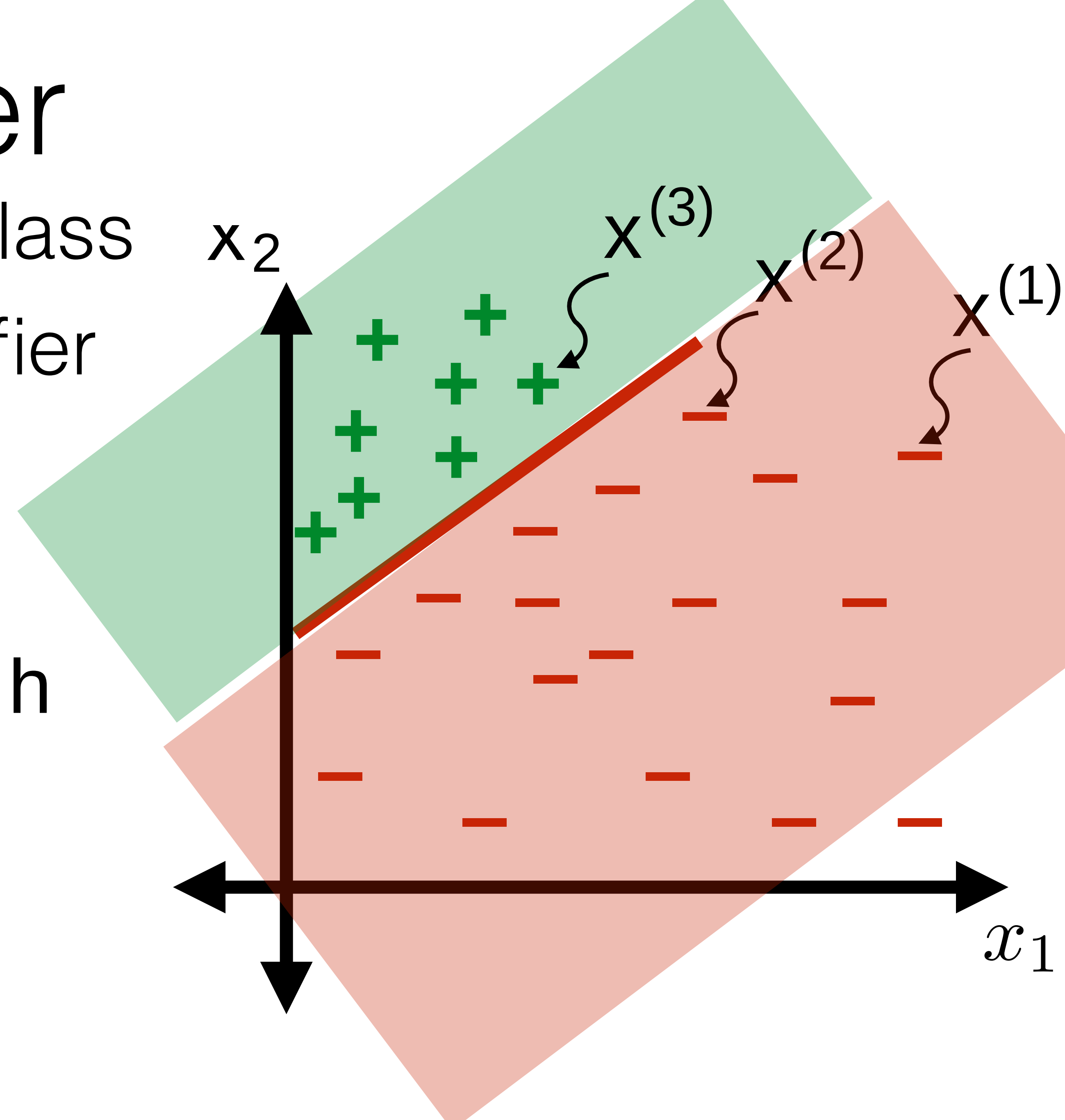
- Have data; have hypothesis class
- Want to choose a good classifier
  - Recall: $x \rightarrow \boxed{h} \rightarrow y$
  - New: $D_n \rightarrow \boxed{\text{learning algorithm}} \rightarrow h$

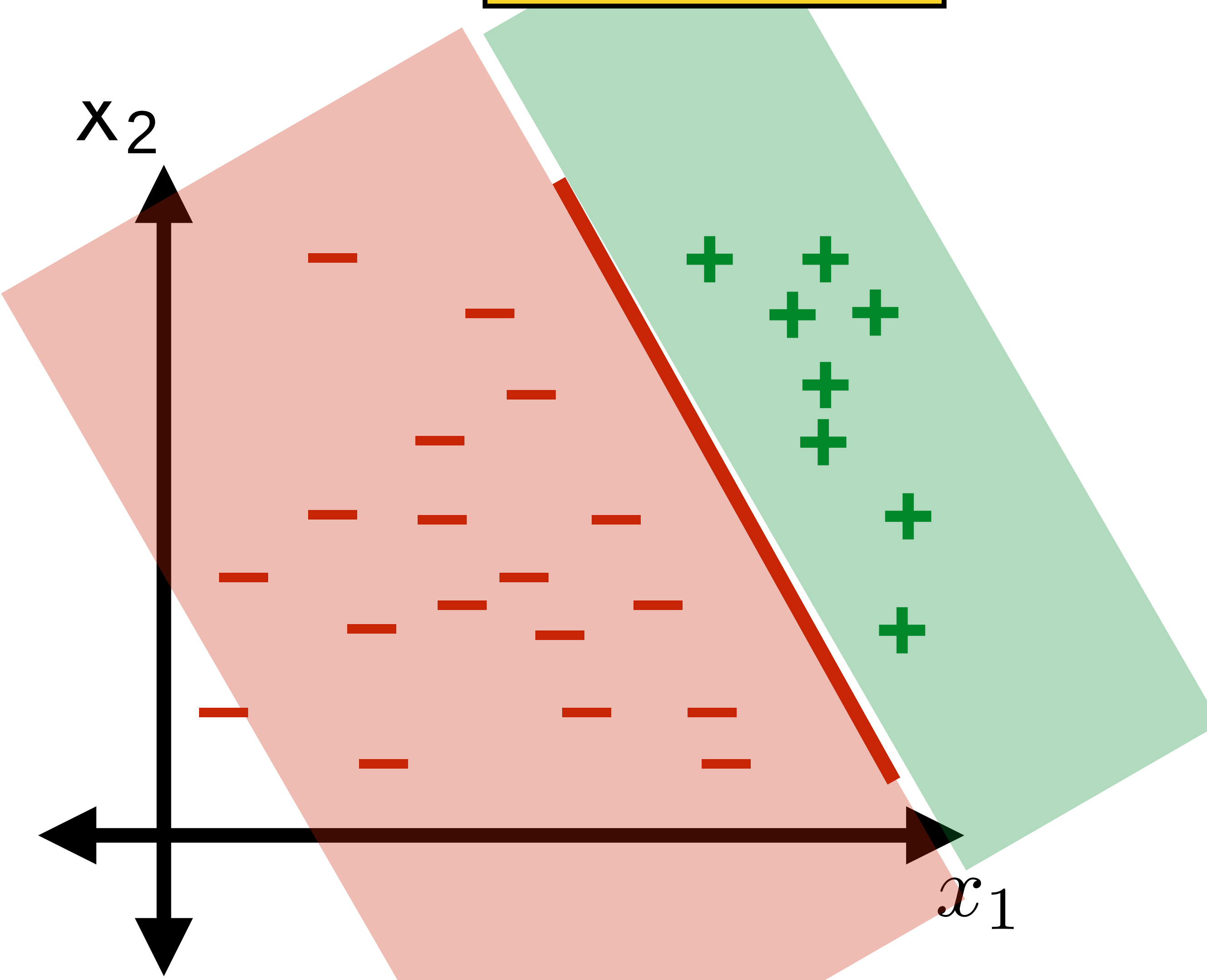




Learning a classifier

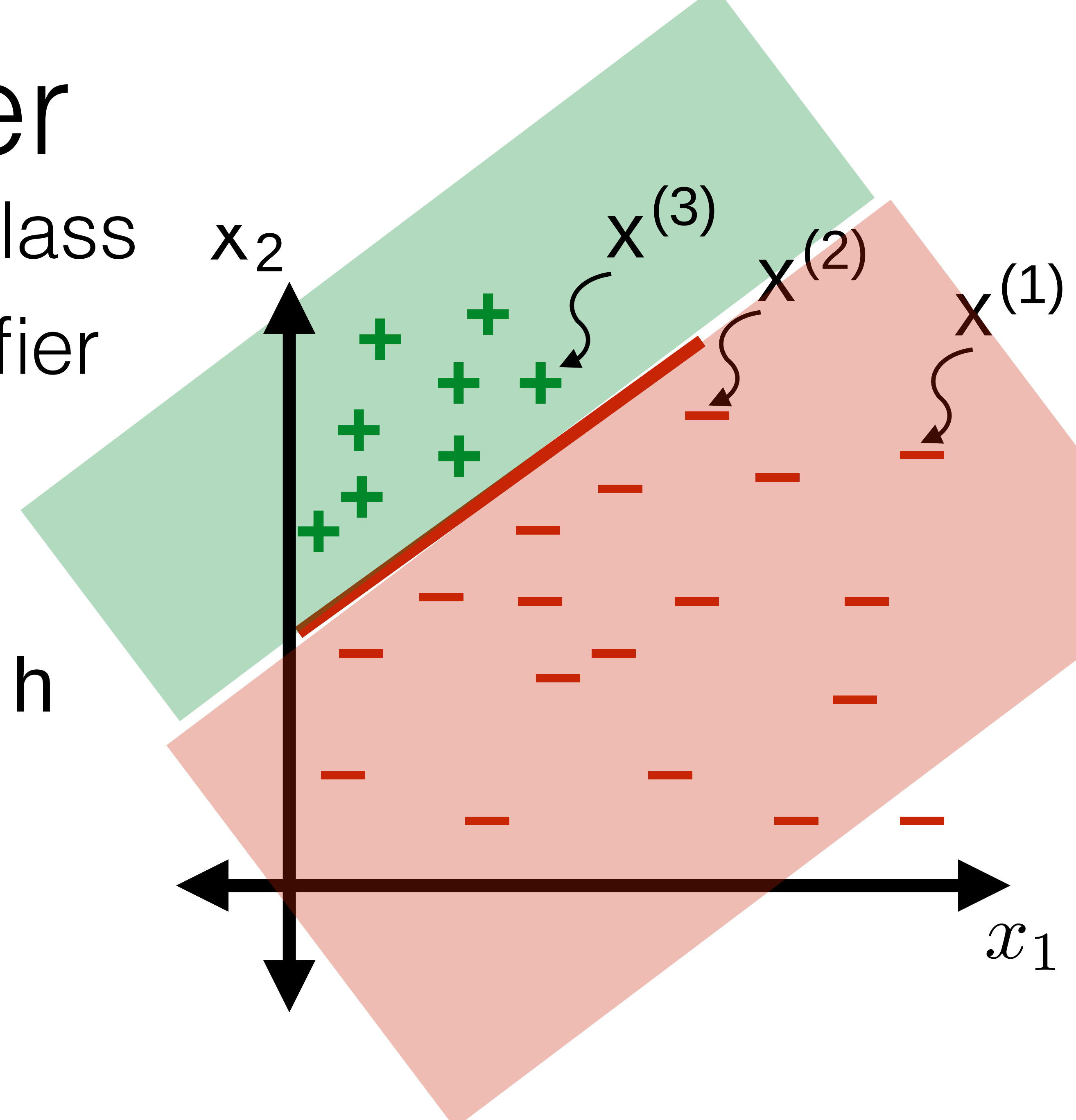
- Have data; have hypothesis class
- Want to choose a good classifier
  - Recall: x → h → y
  - New:
    D_n → learning algorithm → h





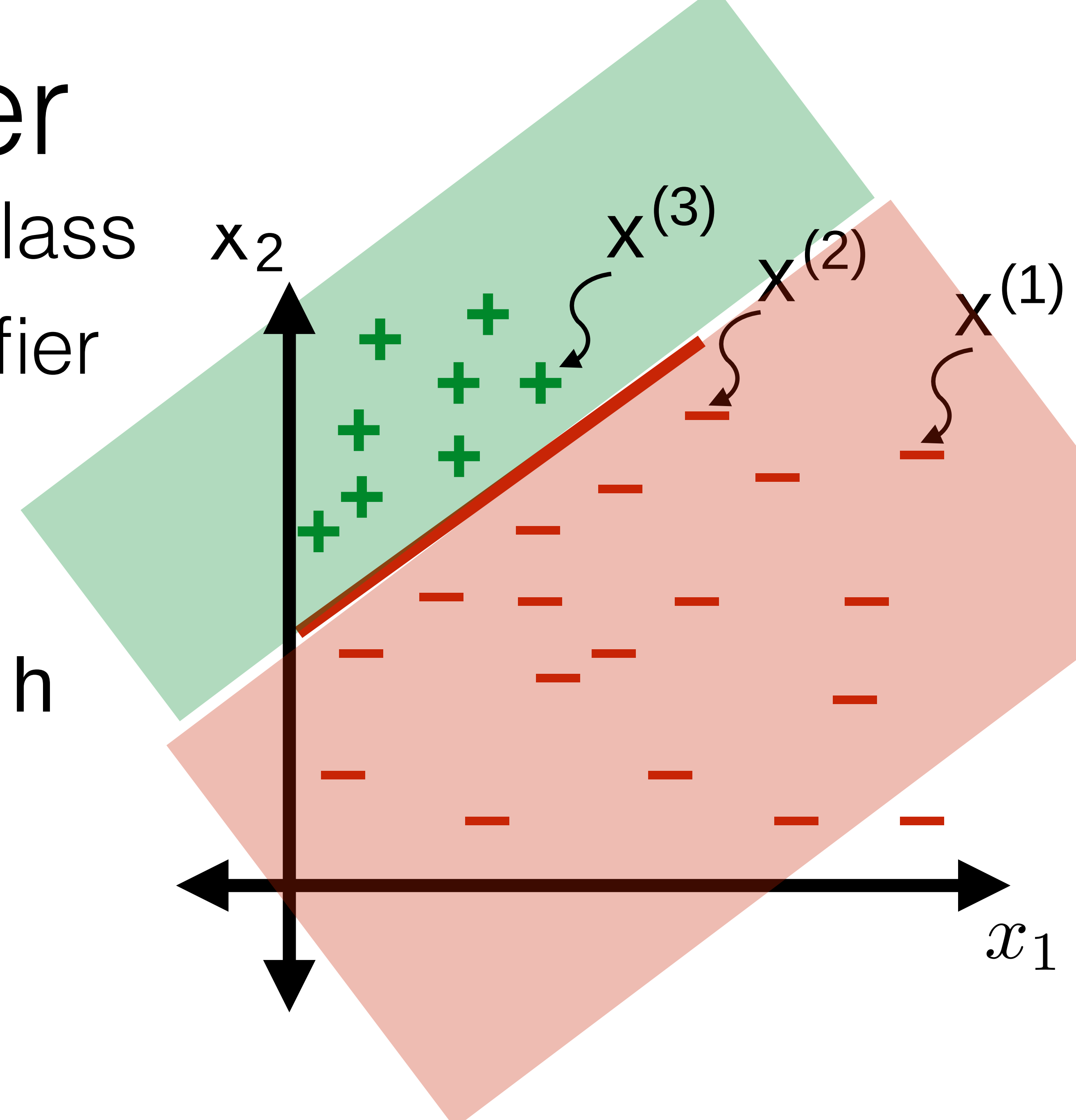
Learning a classifier

- Have data; have hypothesis class
- Want to choose a good classifier
    - Recall: $x \rightarrow \boxed{h} \rightarrow y$
    - New: $D_n \rightarrow \boxed{\text{learning algorithm}} \rightarrow h$



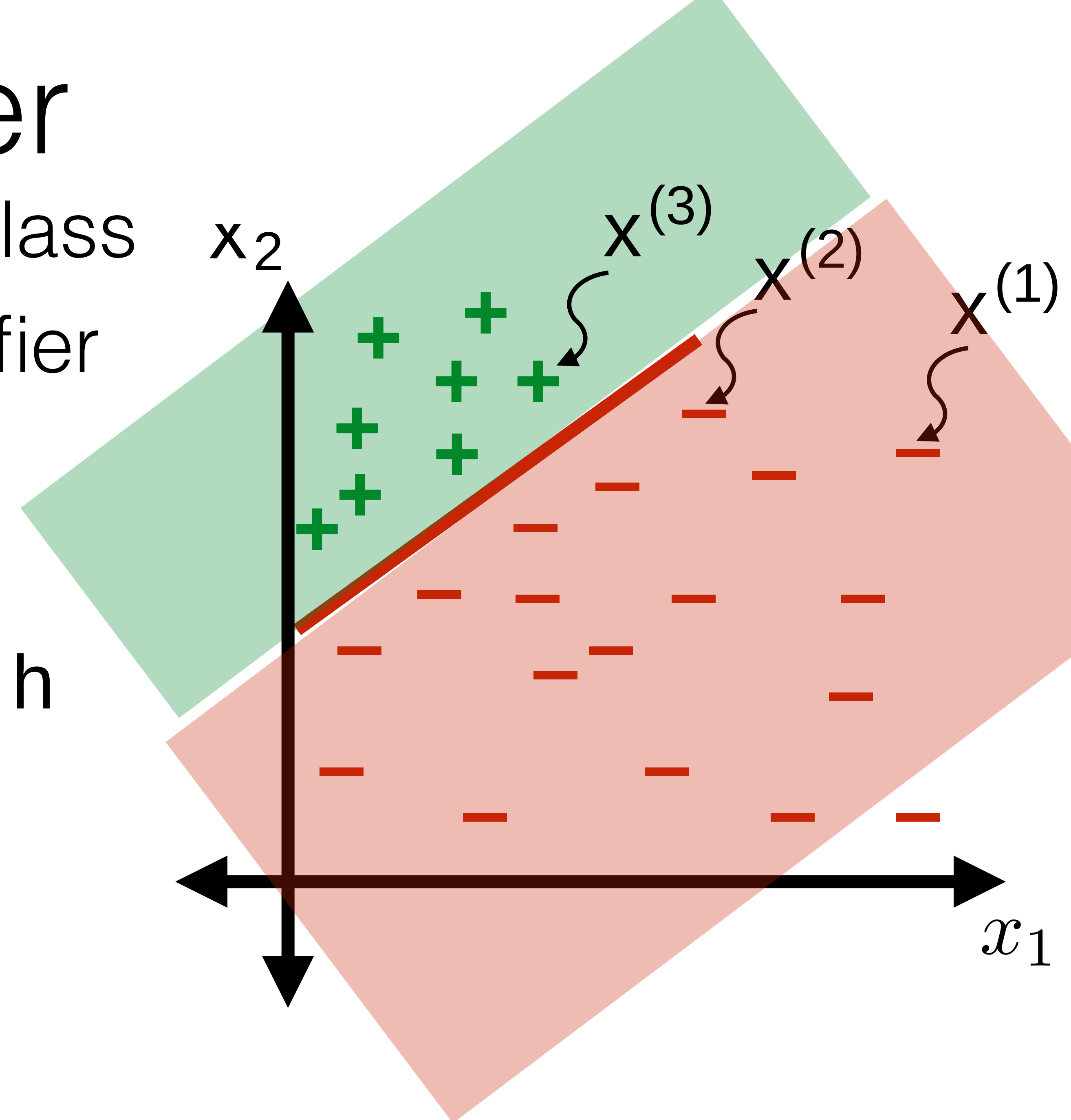
Learning a classifier

- Have data; have hypothesis class
- Want to choose a good classifier
  - Recall: x → h → y
  - New: D_n → learning algorithm → h
- Example:



Learning a classifier

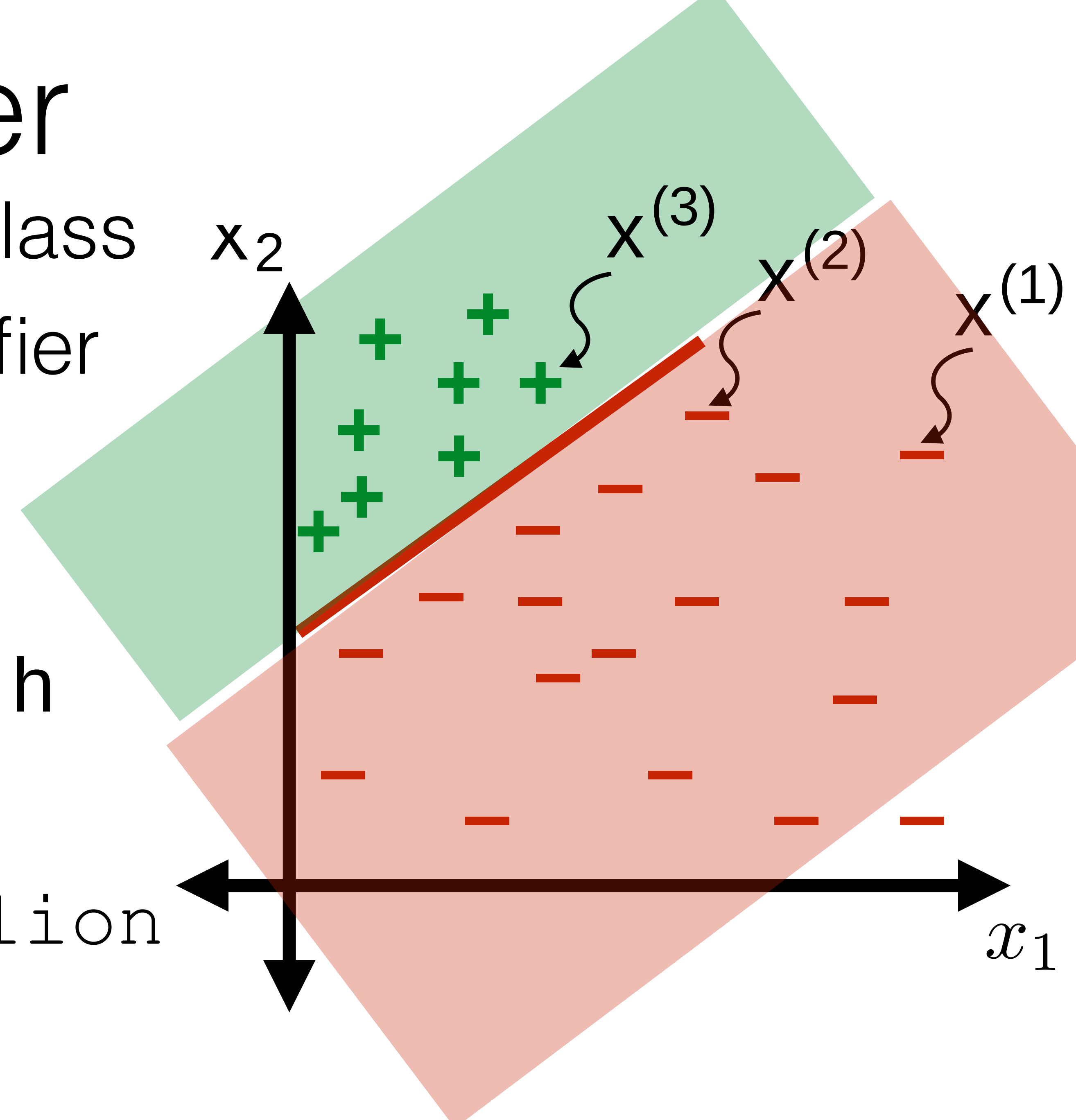
- Have data; have hypothesis class
- Want to choose a good classifier
  - Recall: $x \rightarrow \boxed{h} \rightarrow y$
  - New: $D_n \rightarrow \boxed{\text{learning algorithm}} \rightarrow h$
- Example:



Learning a classifier

- Have data; have hypothesis class
- Want to choose a good classifier
  - Recall: $x \rightarrow$ h $\rightarrow$ y
  - New: $D_n \rightarrow$ learning algorithm $\rightarrow$ h
- Example:

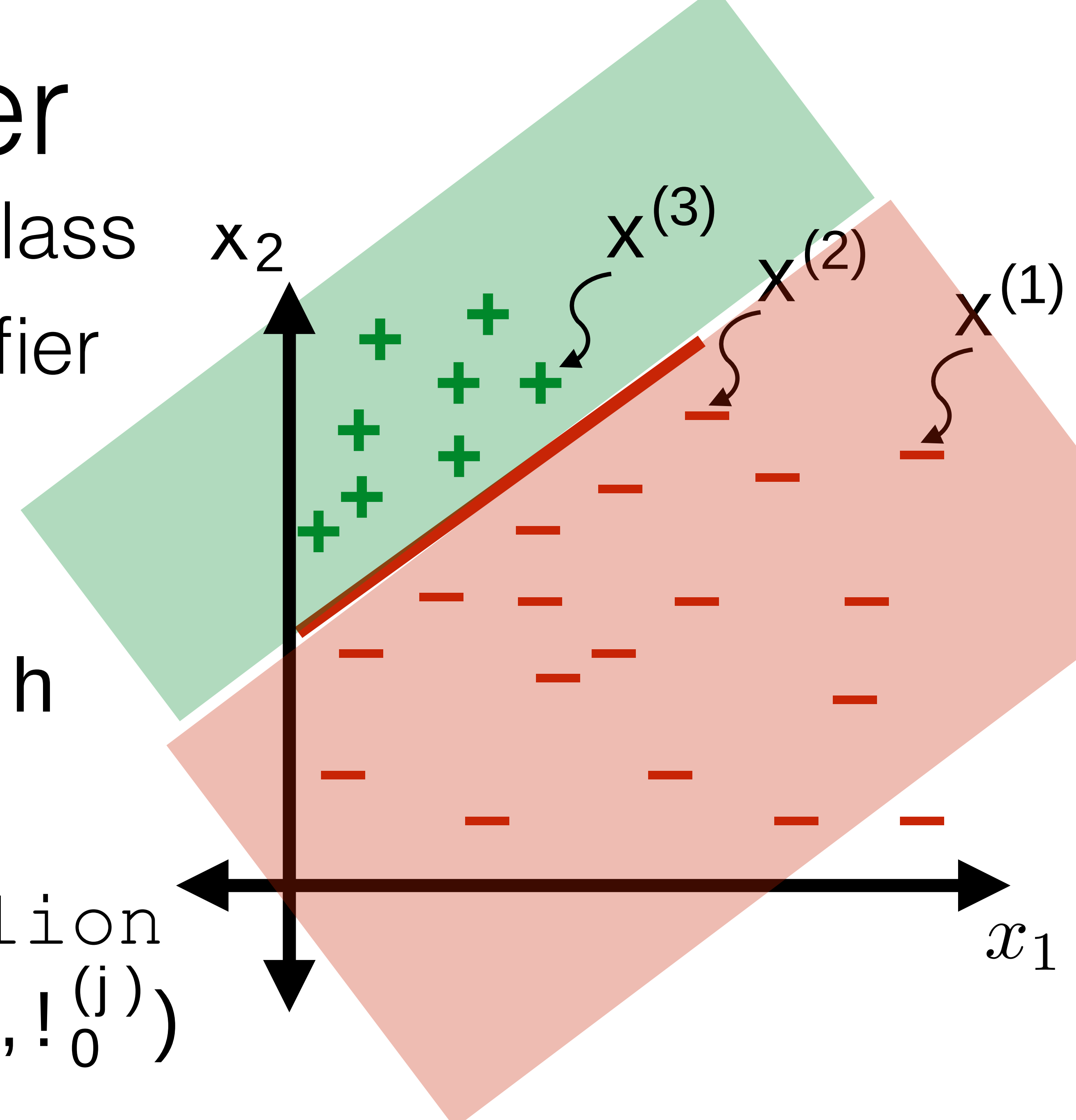
```
for j = 1, …, 1 trillion
```



Learning a classifier

- Have data; have hypothesis class
- Want to choose a good classifier
  - Recall: x → h → y
  - New: D_n → learning algorithm → h
- Example:

```
for j = 1, …, 1 trillion
  Randomly sample (!^(j), !_0^(j))
```



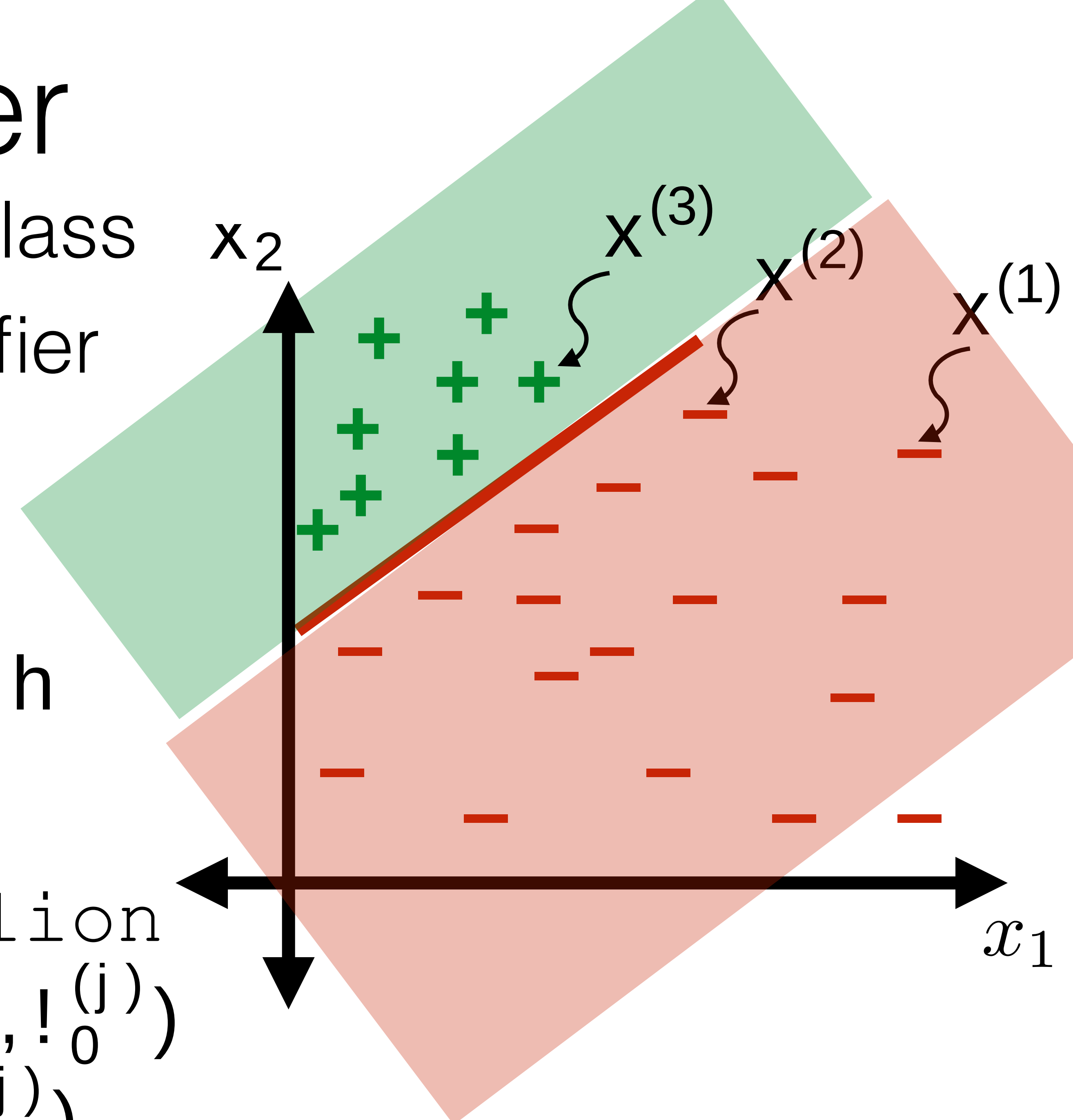
Learning a classifier

- Have data; have hypothesis class
- Want to choose a good classifier
  - Recall: $x \rightarrow \boxed{h} \rightarrow y$
  - New: $D_n \rightarrow \boxed{\text{learning algorithm}} \rightarrow h$
- Example:

```
for j = 1, …, 1 trillion
  Randomly sample (!(j), !0(j))
  Set h(j)(x) = h(x; !(j), !0(j))
```

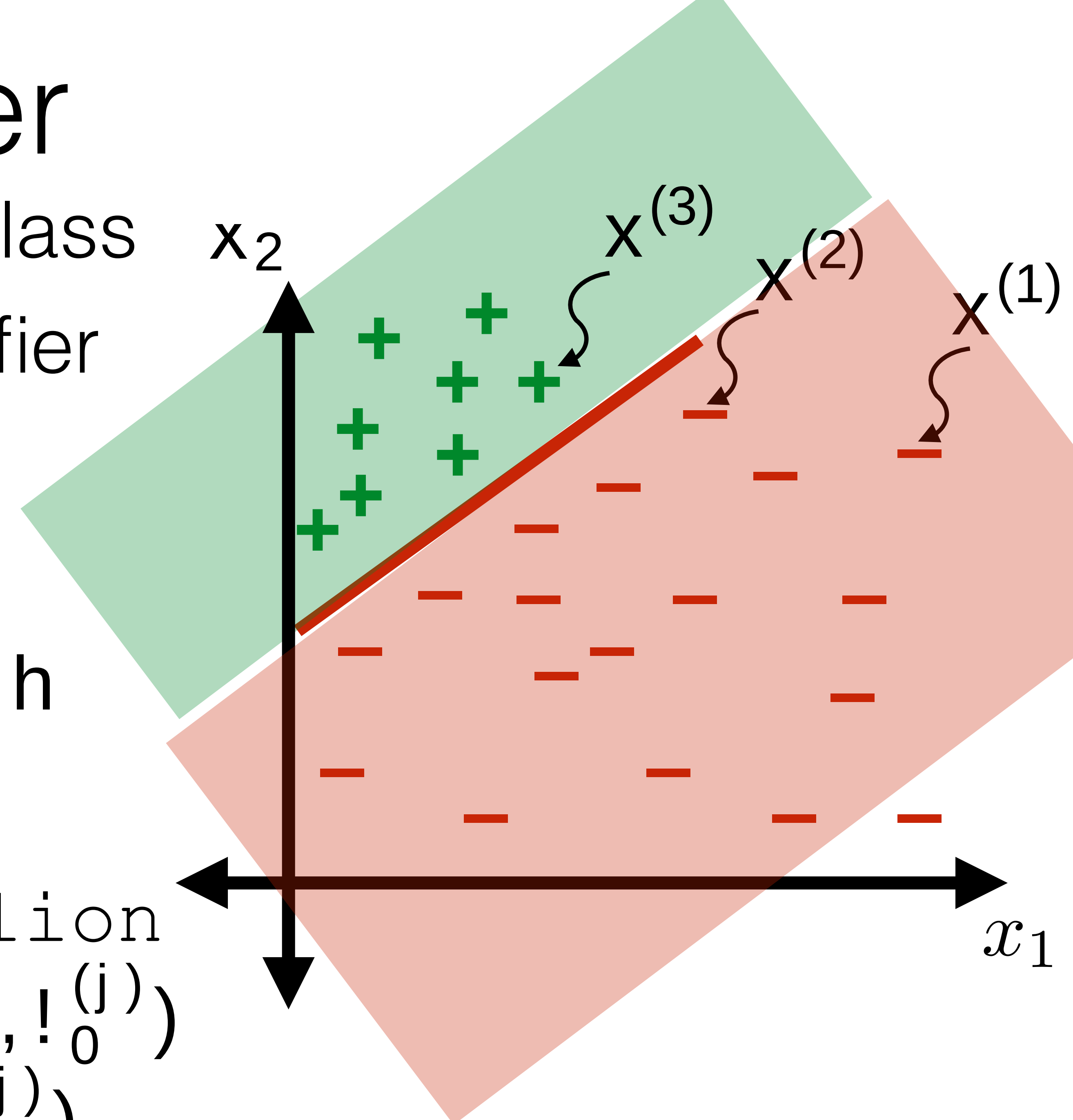
$\text{Randomly sample } (!^{(j)}, !_0^{(j)})$

$\text{Set } h^{(j)}(x) = h(x; !^{(j)}, !_0^{(j)})$



Learning a classifier

- Have data; have hypothesis class
- Want to choose a good classifier
  - Recall: $x \rightarrow$ **h** $\rightarrow$ y
  - New: $D_n \rightarrow$ **learning algorithm** $\rightarrow$ h



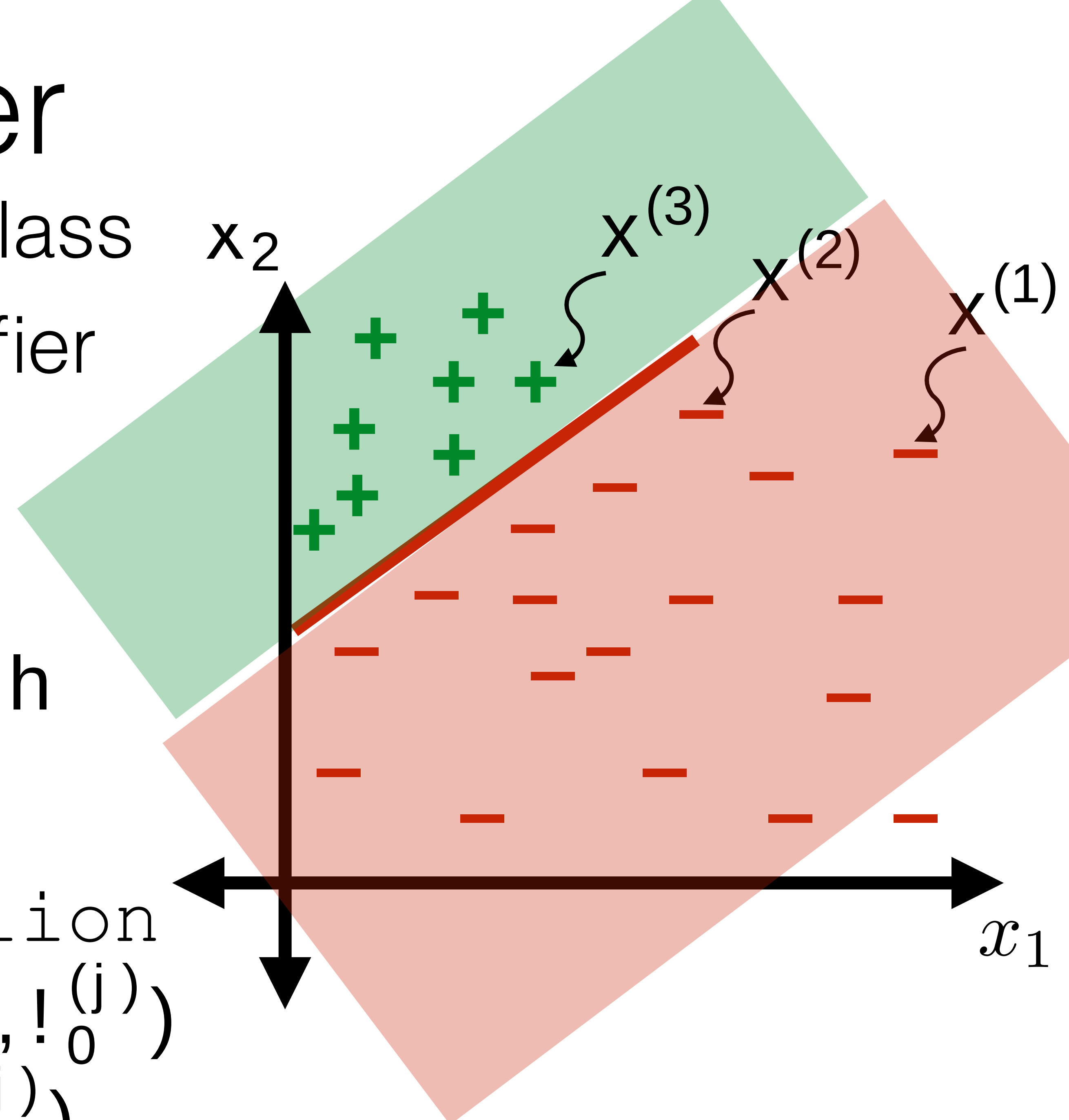
- Example:

```
for j = 1, …, 1 trillion
  Randomly sample (!^(j), !_0^(j))
  Set h^(j)(x) = h(x; !^(j), !_0^(j))
```

Learning a classifier

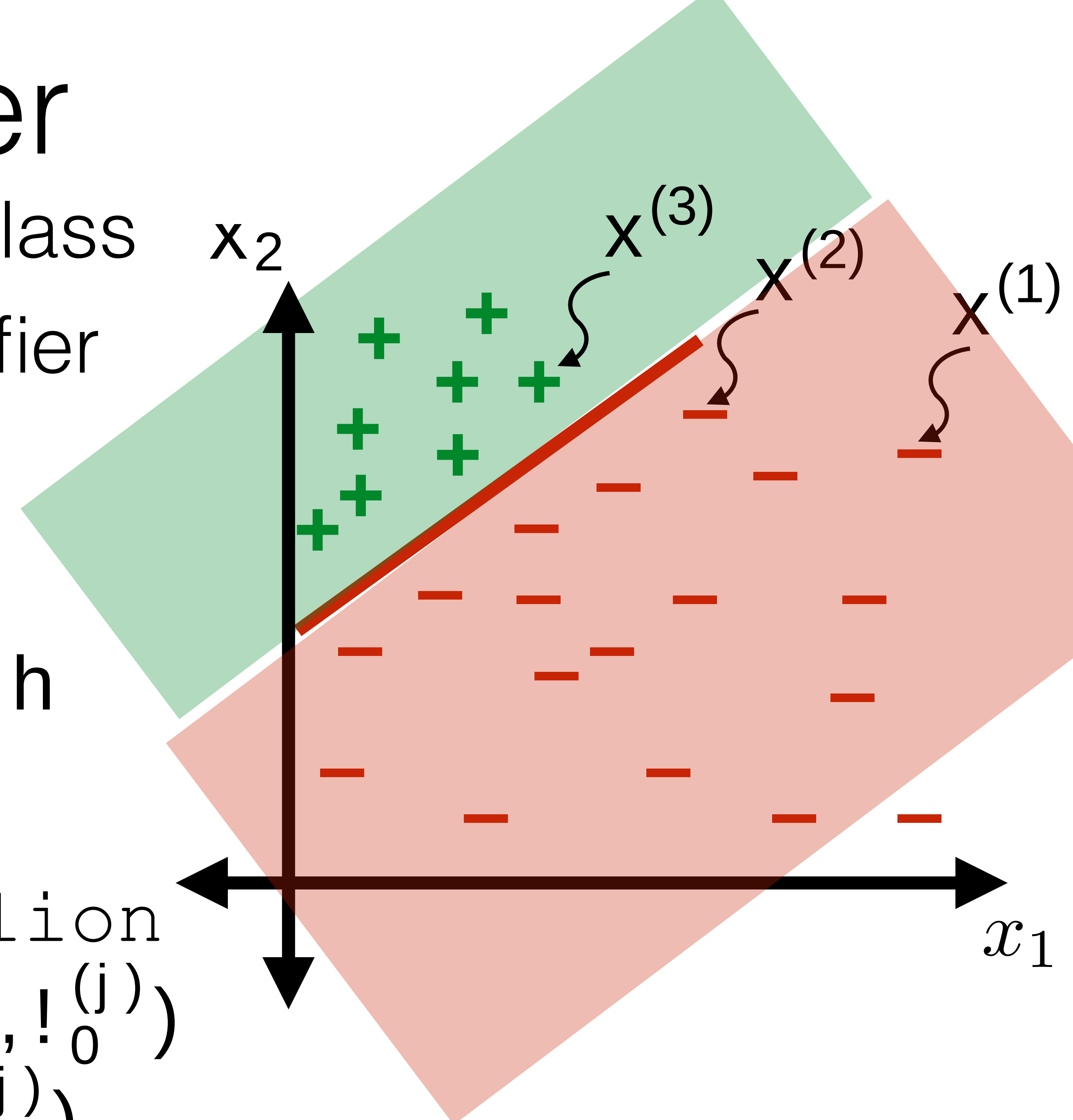
- Have data; have hypothesis class
- Want to choose a good classifier
  - Recall: $x \rightarrow$ h $\rightarrow$ y
  - New: $D_n \rightarrow$ learning algorithm $\rightarrow$ h
- Example:

```
for j = 1, …, 1 trillion
  Randomly sample (!(j),!0(j))
  Set h(j)(x) = h(x;!(j),!0(j))
Ex_learning_alg
```



Learning a classifier

- Have data; have hypothesis class
- Want to choose a good classifier
  - Recall: $x \rightarrow$ h $\rightarrow$ y
  - New: $D_n \rightarrow$ learning algorithm $\rightarrow$ h

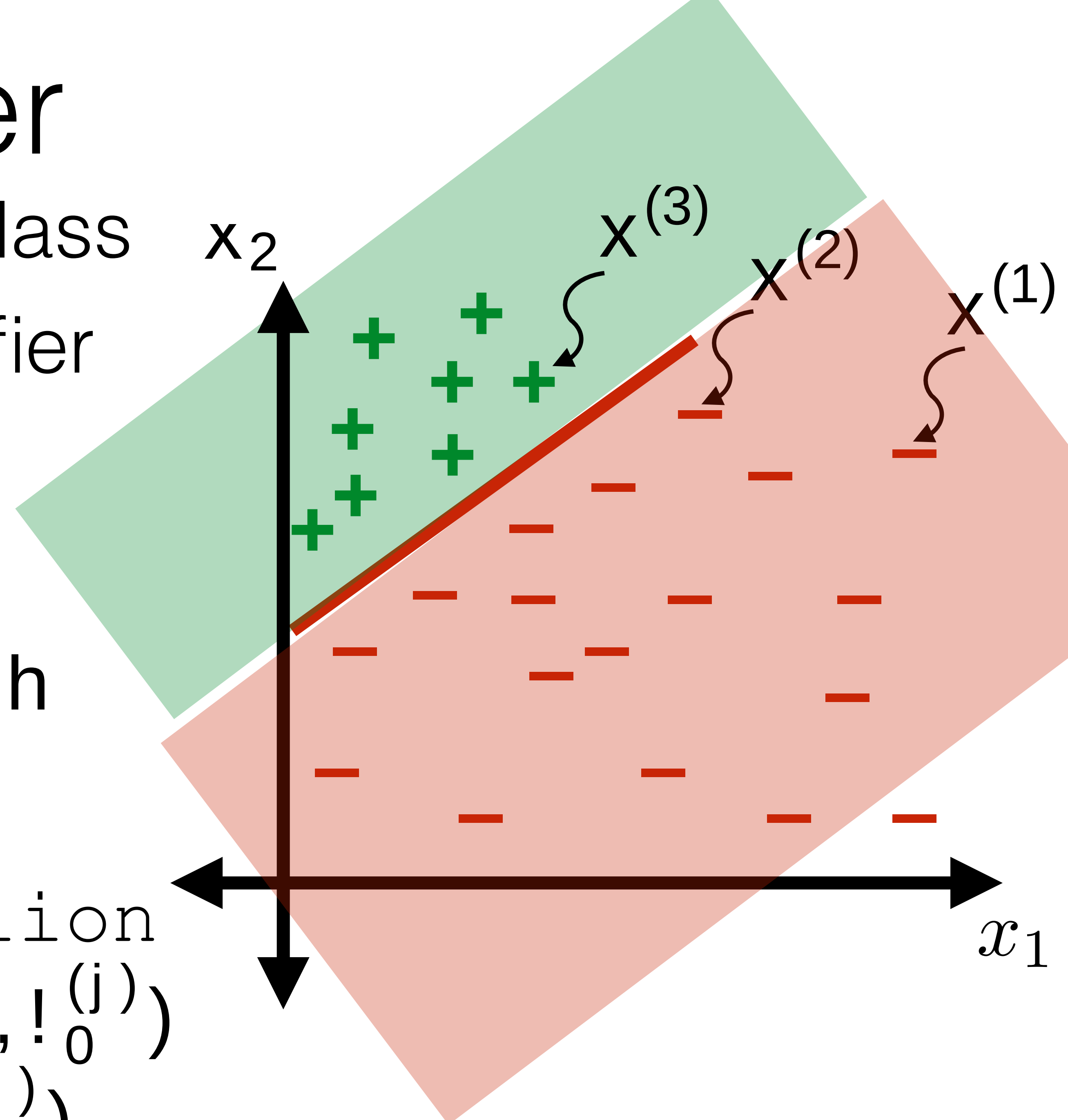


- Example:

```
for j = 1, …, 1 trillion
  Randomly sample (!^(j), !_0^(j))
  Set h^(j)(x) = h(x; !^(j), !_0^(j))
Ex_learning_alg( D_n )
```

Learning a classifier

- Have data; have hypothesis class
- Want to choose a good classifier
  - Recall: $x \rightarrow$ h $\rightarrow$ y
  - New: $D_n \rightarrow$ learning algorithm $\rightarrow$ h



- Example:

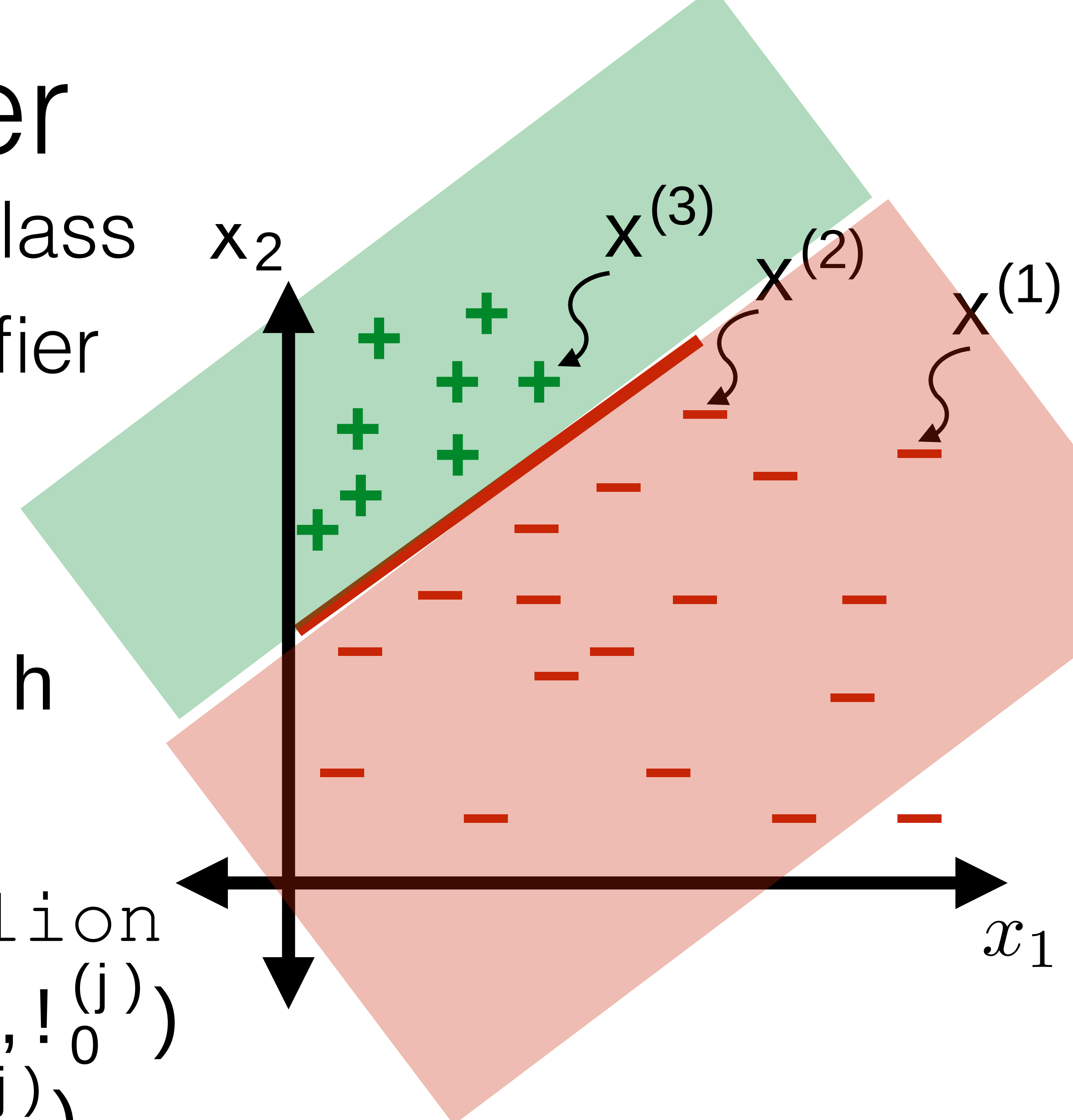
```
for j = 1, …, 1 trillion
   Randomly sample (!(j), !0(j))
   Set h(j)(x) = h(x; !(j), !0(j))
Ex_learning_alg( Dn ; k < 1 trillion)
```

Learning a classifier

- Have data; have hypothesis class
- Want to choose a good classifier
  - Recall: $x \rightarrow$ h $\rightarrow$ y
  - New: $D_n \rightarrow$ learning algorithm $\rightarrow$ h
- Example:

for j = 1, …, 1 trillion
  Randomly sample $(!^{(j)}, !_0^{(j)})$
  Set $h^{(j)}(x) = h(x; !^{(j)}, !_0^{(j)})$
Ex_learning_alg(D_n ; k < 1 trillion)

hyperparameter

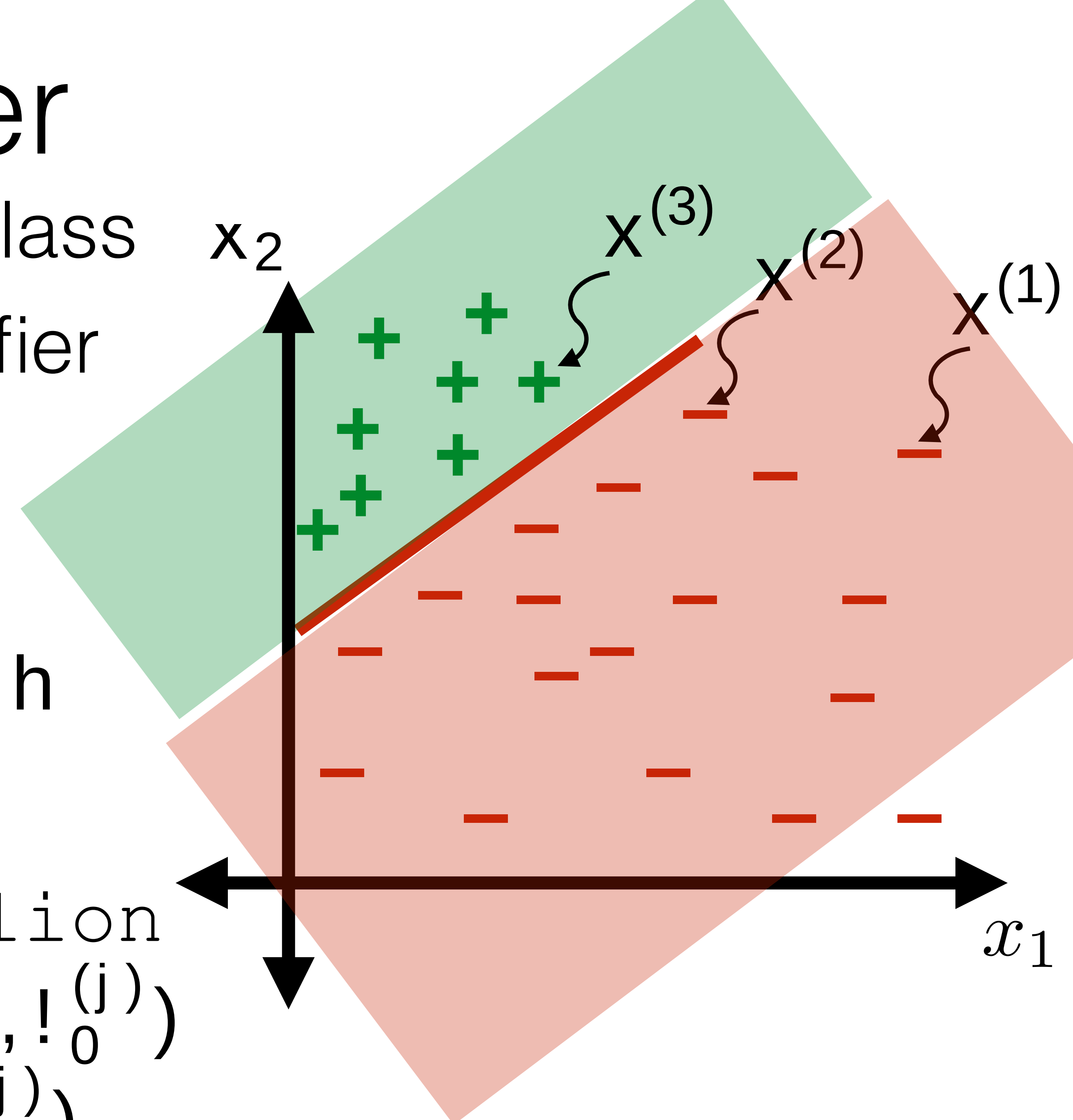


Learning a classifier

- Have data; have hypothesis class
- Want to choose a good classifier
  - Recall: x → h → y
  - New: D_n → learning algorithm → h
- Example:

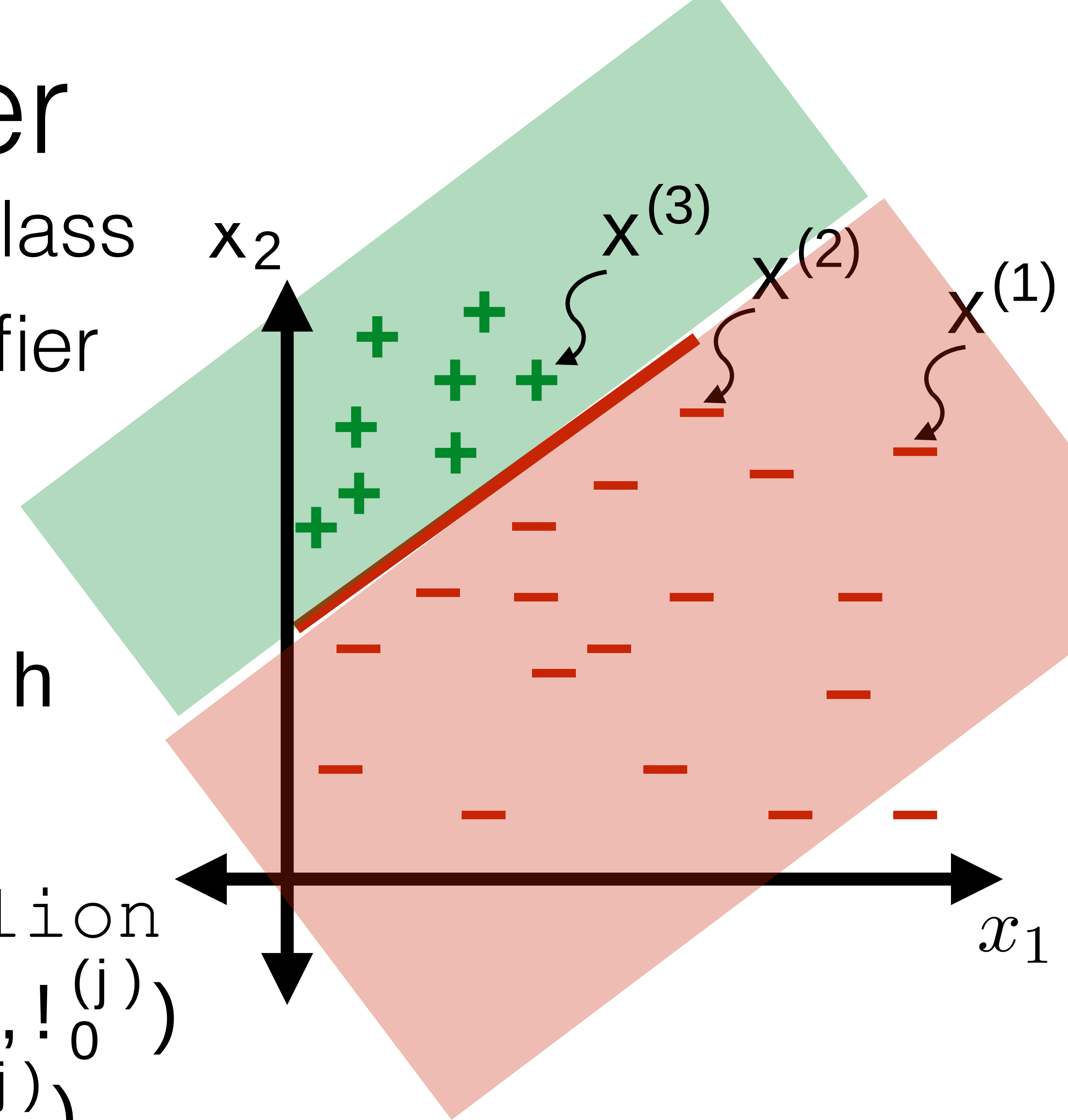
```
for j = 1, …, 1 trillion
  Randomly sample (!^(j), !_0^(j))
  Set h^(j)(x) = h(x; !^(j), !_0^(j))
Ex_learning_alg( D_n ; k < 1 trillion)
  Set j^! = argmin_{j " {1,…,k}} E_n(h^(j))
```

hyperparameter



Learning a classifier

- Have data; have hypothesis class
- Want to choose a good classifier
  - Recall: $x \rightarrow$ h $\rightarrow$ y
  - New: $D_n \rightarrow$ learning algorithm $\rightarrow$ h



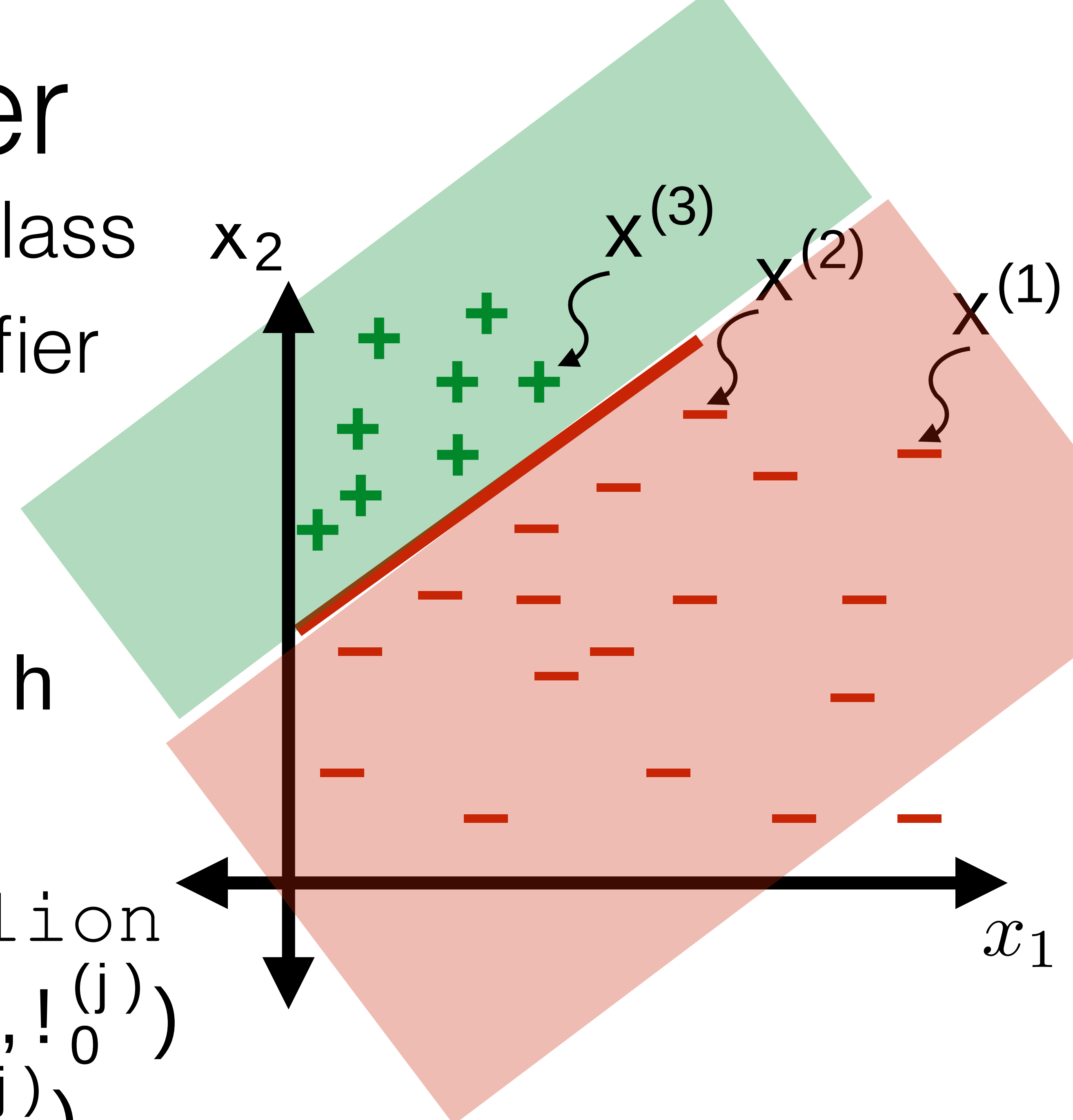
- Example:

```
for j = 1, …, 1 trillion
   Randomly sample (!(j), !0(j))
   Set h(j)(x) = h(x; !(j), !0(j))
Ex_learning_alg( Dn ; k < 1 trillion)
   Set j! = argmin j " {1,...,k} En(h(j))
   Return h(j!)
```

hyperparameter

Learning a classifier

- Have data; have hypothesis class
- Want to choose a good classifier
  - Recall: x → **h** → y
  - New: D_n → **learning algorithm** → h



- Example:

```
for j = 1, …, 1 trillion
   Randomly sample (!(j), !0(j))
   Set h(j)(x) = h(x; !(j), !0(j))
Ex_learning_alg( Dn ; k < 1 trillion)
   Set j! = argmin j " {1,…,k} En(h(j))
   Return h(j!)
```

hyperparameter

- How does training error of `Ex_learning_alg`(D_n;1) compare to the training error of `Ex_learning_alg`(D_n;2)?

MIT · 6.036 | Introduction to Machine Learning (2020)

MIT 6.036(2020)· 课程资料包 @ShowMeAI

视频
中英双语字幕

课件
一键打包下载

笔记
官方笔记翻译

代码
作业项目解析

视频 · B 站 [扫码或点击链接]
https://www.bilibili.com/video/BV1y44y187wN

课件 & 代码 · 博客 [扫码或点击链接]
http://blog.showmeai.tech/mit-6.036

机器学习 循环神经 神经网络 感知器 特征构建 聚类 马尔可夫决策过程 网络 随机森林 决策树 逻辑回归 卷积神经网络 状态机

Awesome AI Courses Notes Cheatsheets 是 ShowMeAI 资料库的分支系列，覆盖最具知名度的 **TOP50+** 门 AI 课程，旨在为读者和学习者提供一整套高品质中文学习笔记和速查表。

点击课程名称，跳转至课程**资料包**页面，**一键下载**课程全部资料！

机器学习	深度学习	自然语言处理	计算机视觉
Stanford · CS229	Stanford · CS230	Stanford · CS224n	Stanford · CS231n
# Awesome AI Courses Notes Cheatsheets· 持续更新中			
知识图谱	**图机器学习**	**深度强化学习**	**自动驾驶**
Stanford · CS520	Stanford · CS224W	UCBerkeley · CS285	MIT · 6.S094

微信公众号

资料下载方式 2：扫码点击底部菜单栏

称为 **AI 内容创作者？** 回复 [添砖加瓦]