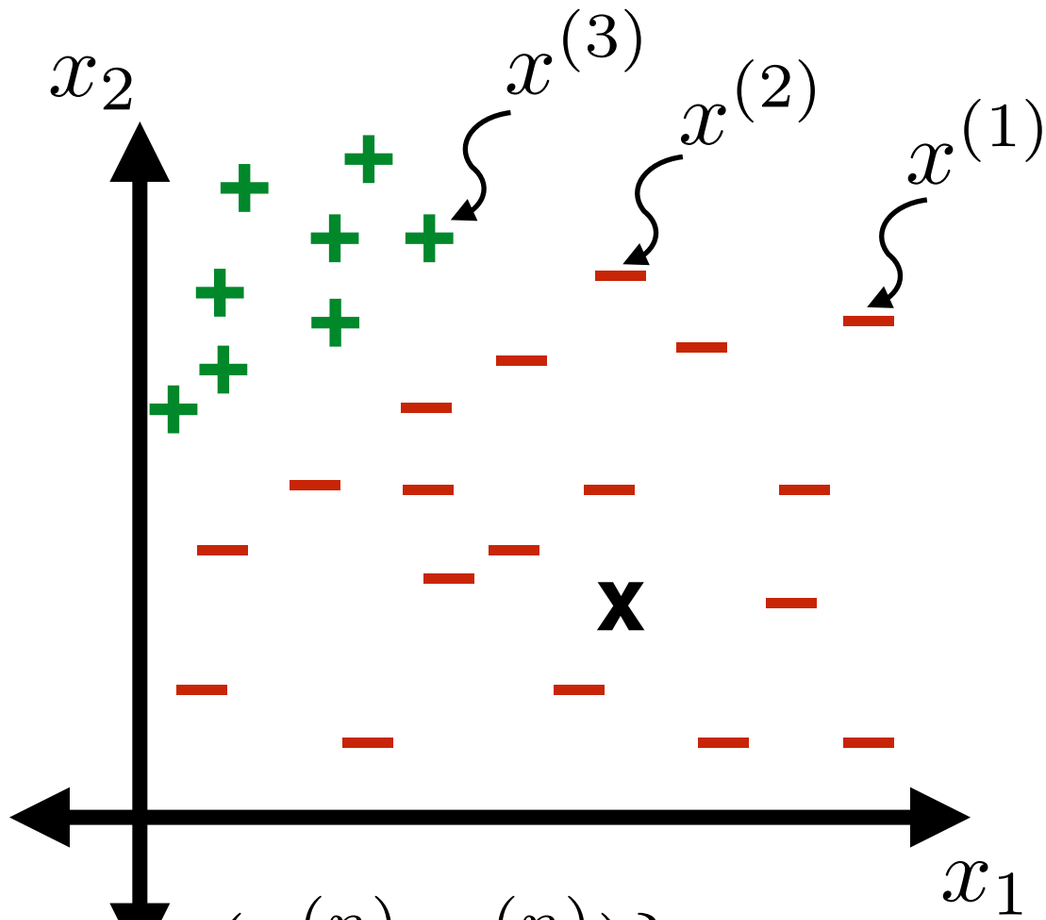
Getting started

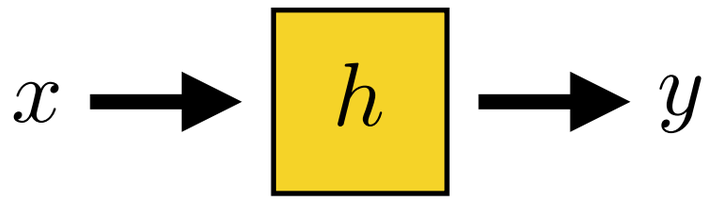
What do we have? (Training) data

- *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

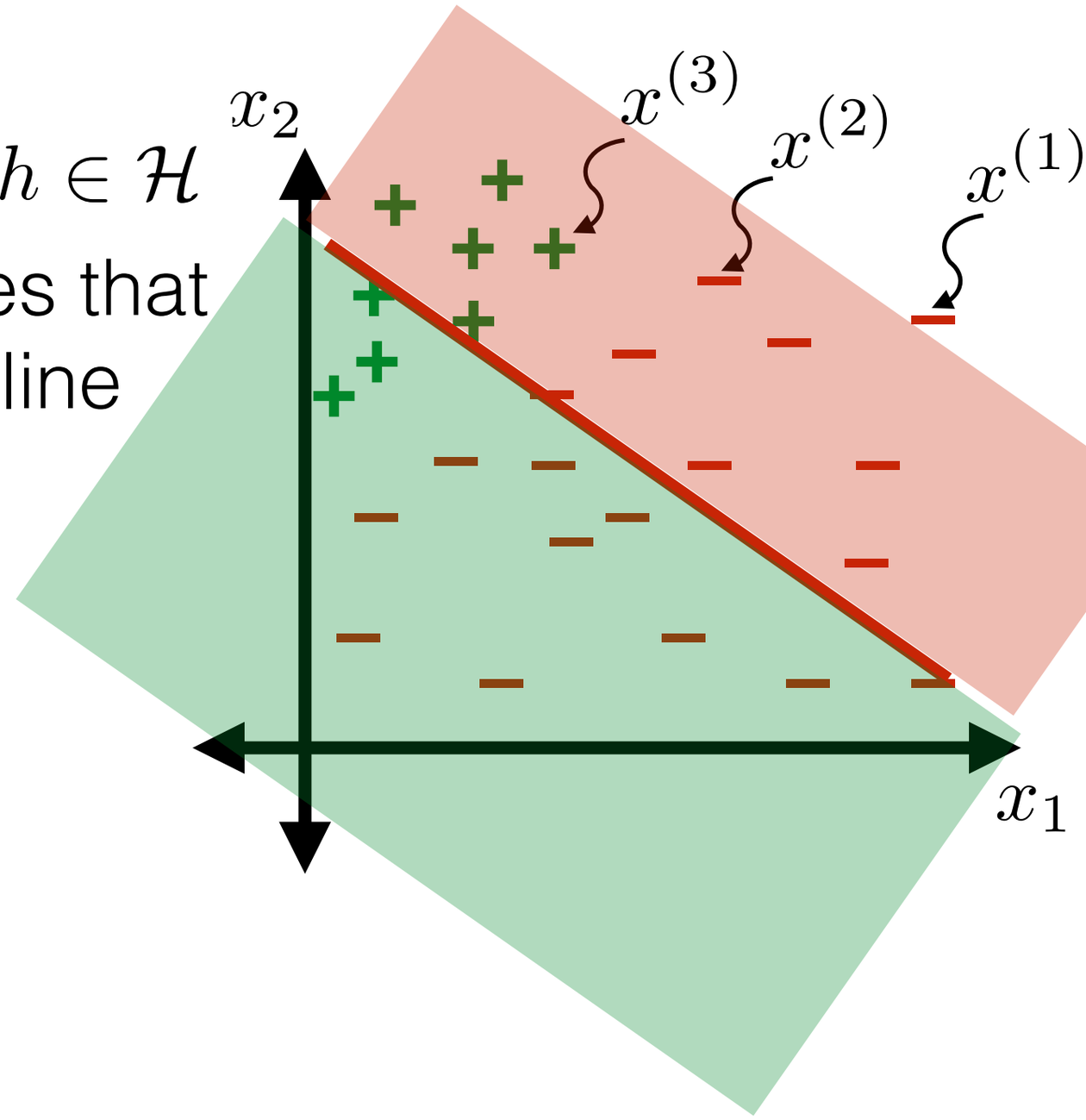
- How to label? Hypothesis $h : \mathbb{R}^d \to \{-1, +1\}$



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

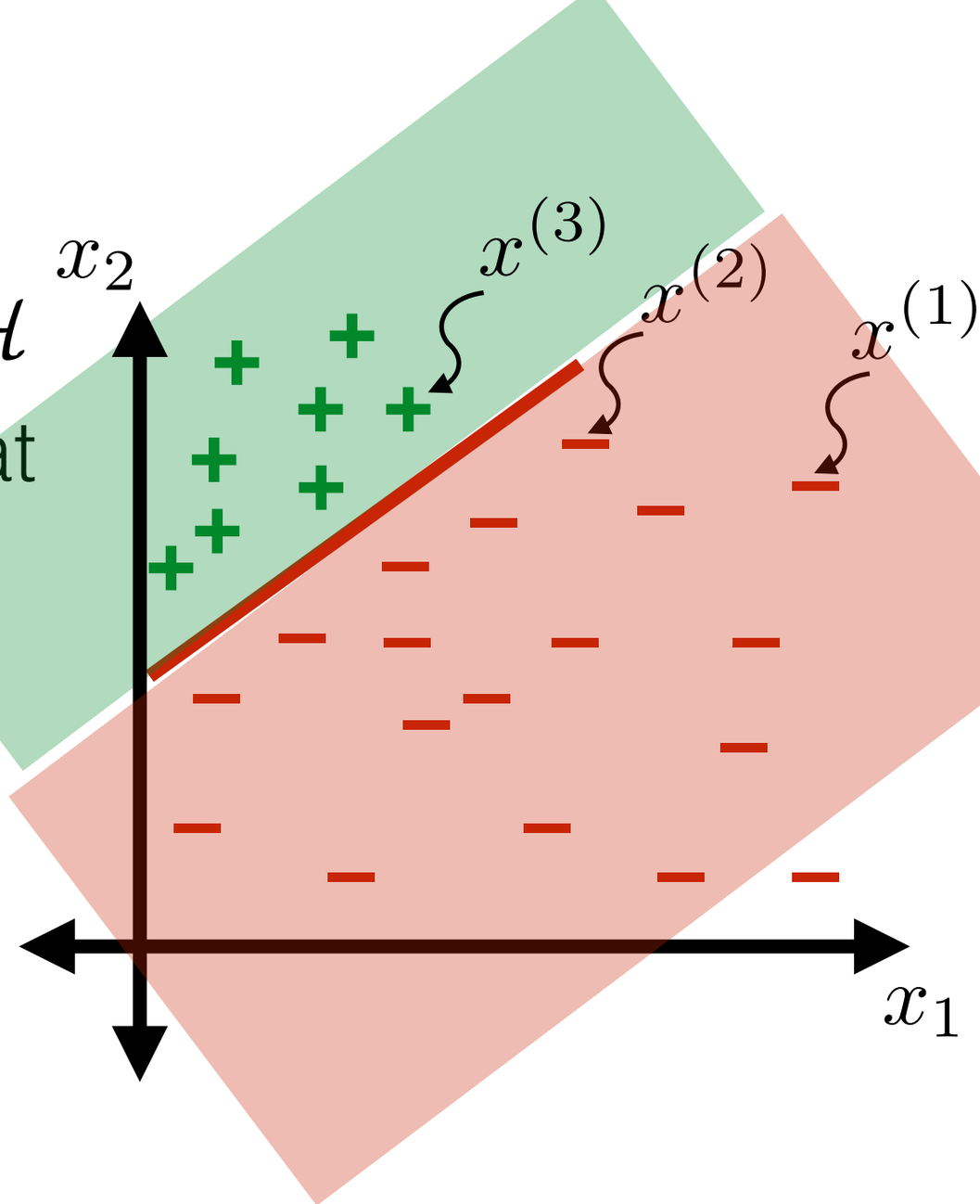
Linear classifiers

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

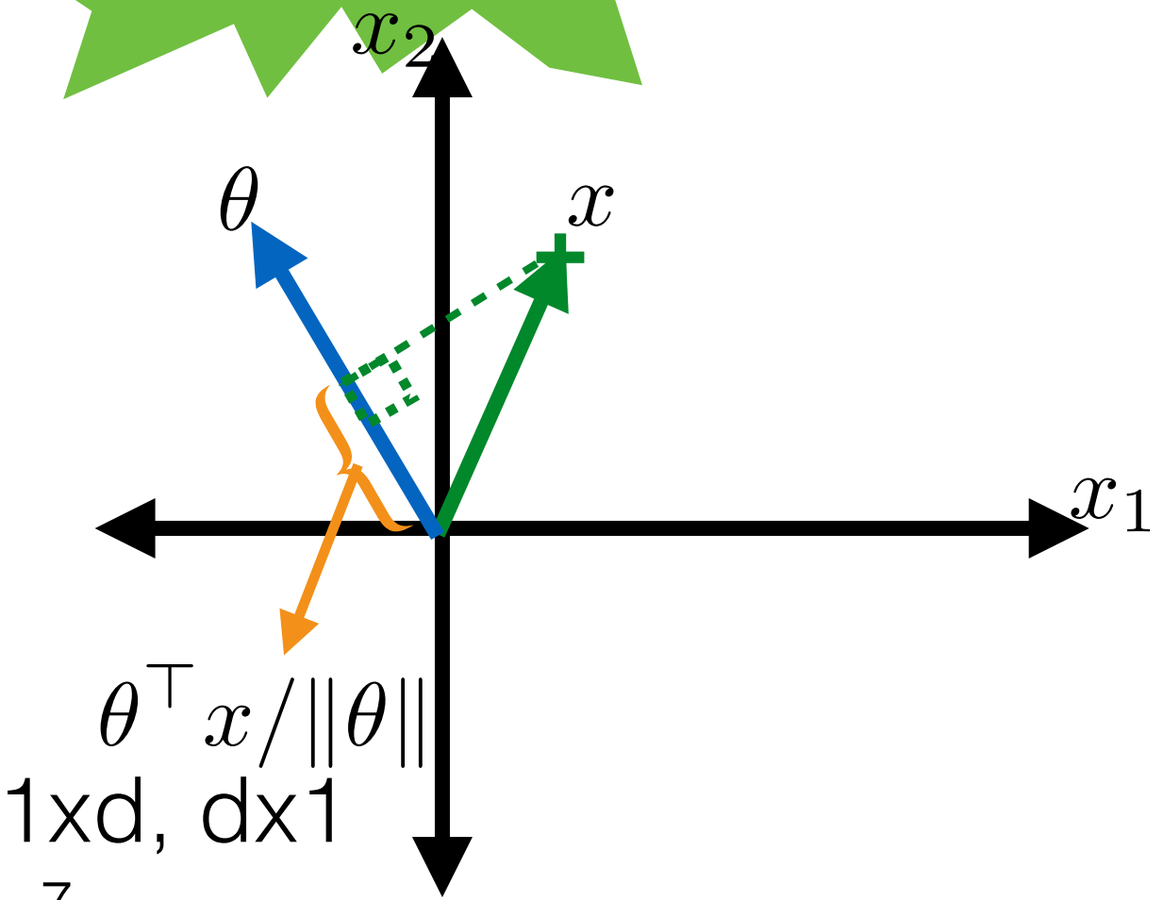


Linear classifiers

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

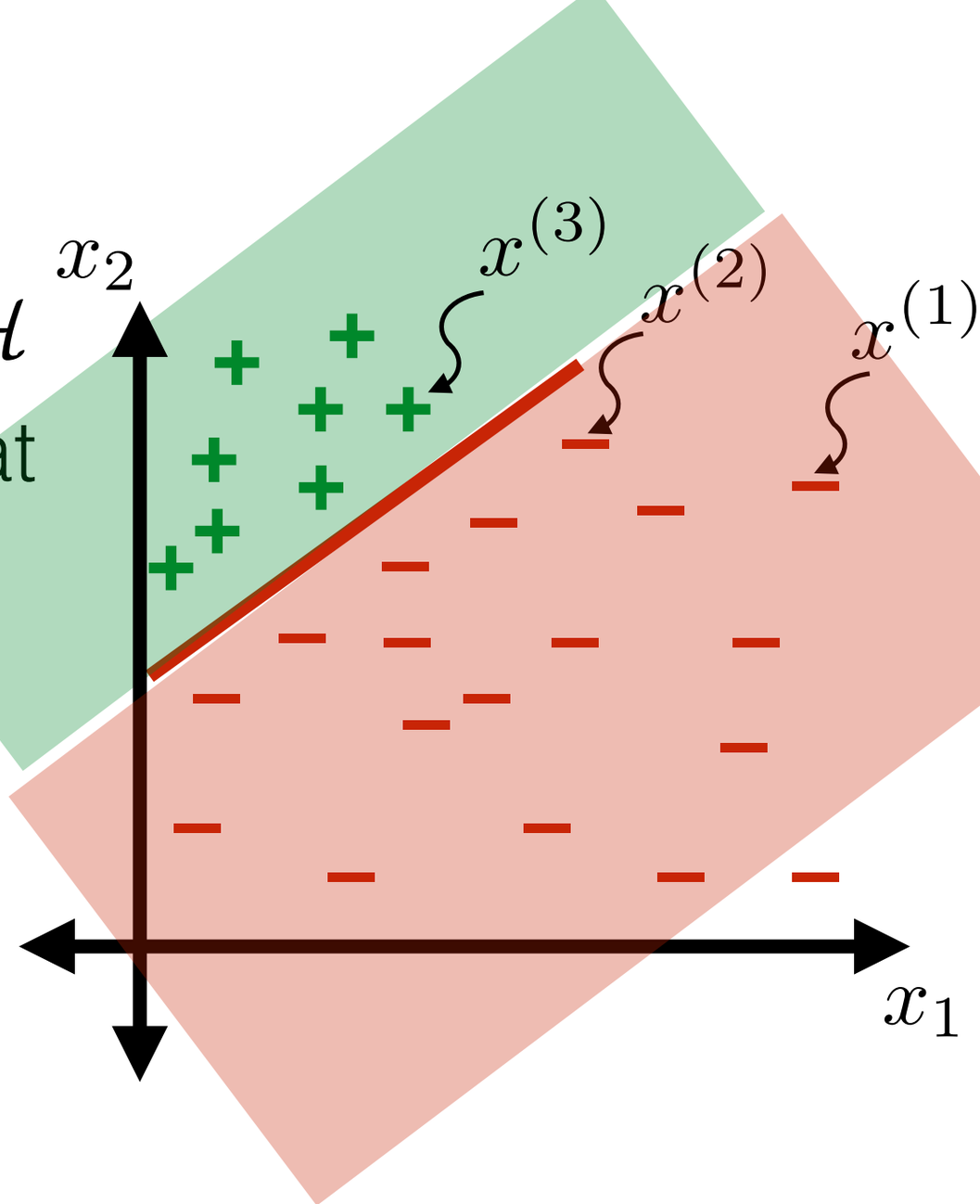


Math facts!

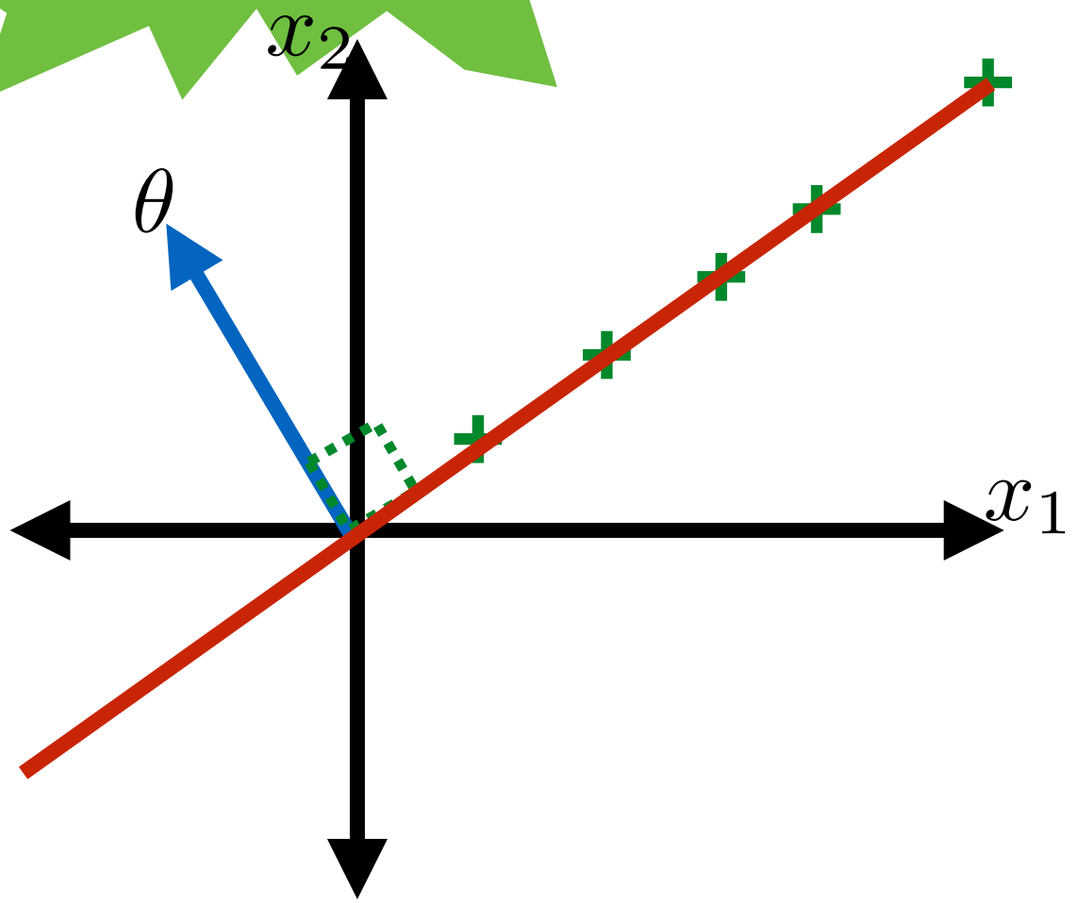


Linear classifiers

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

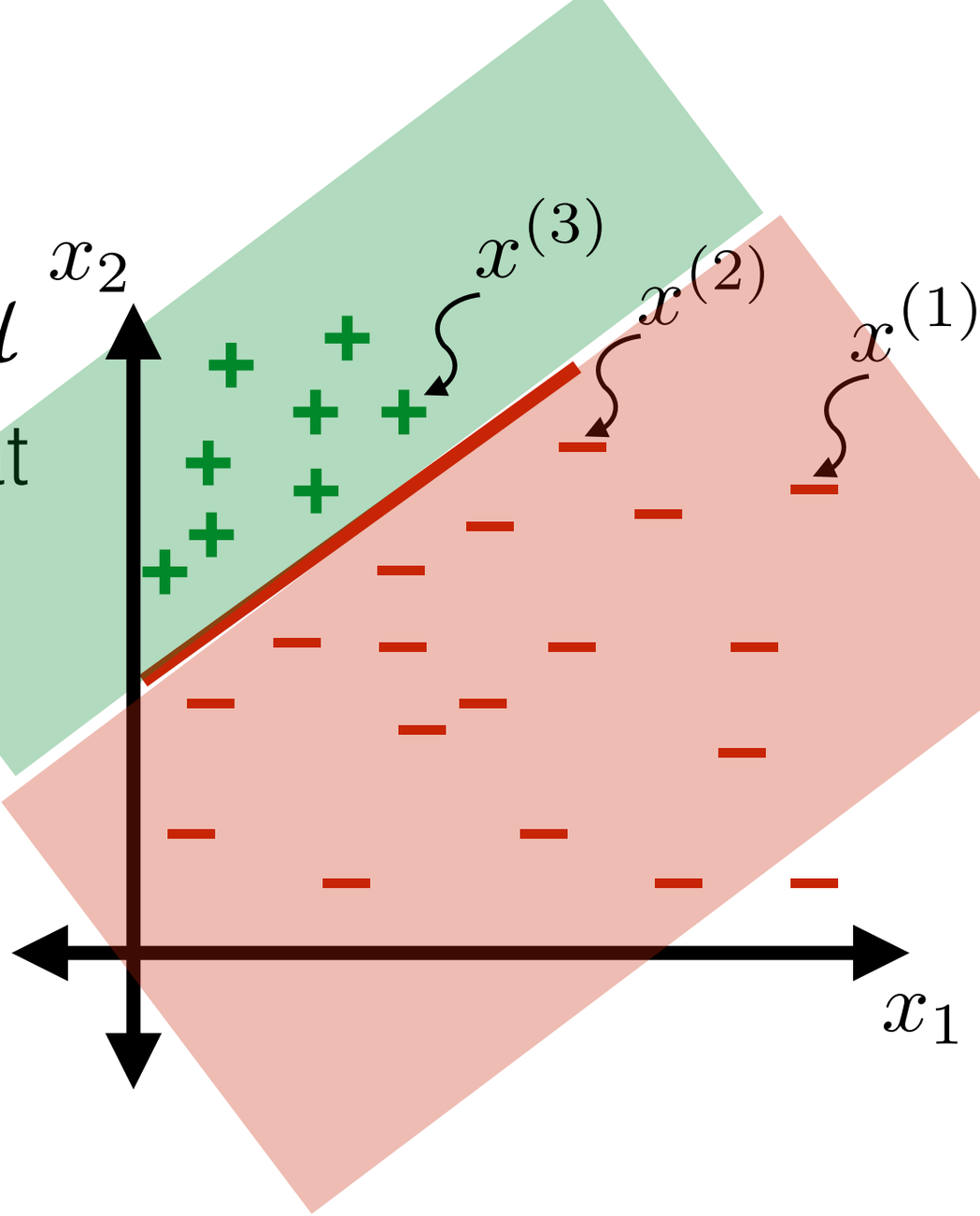


Math facts!

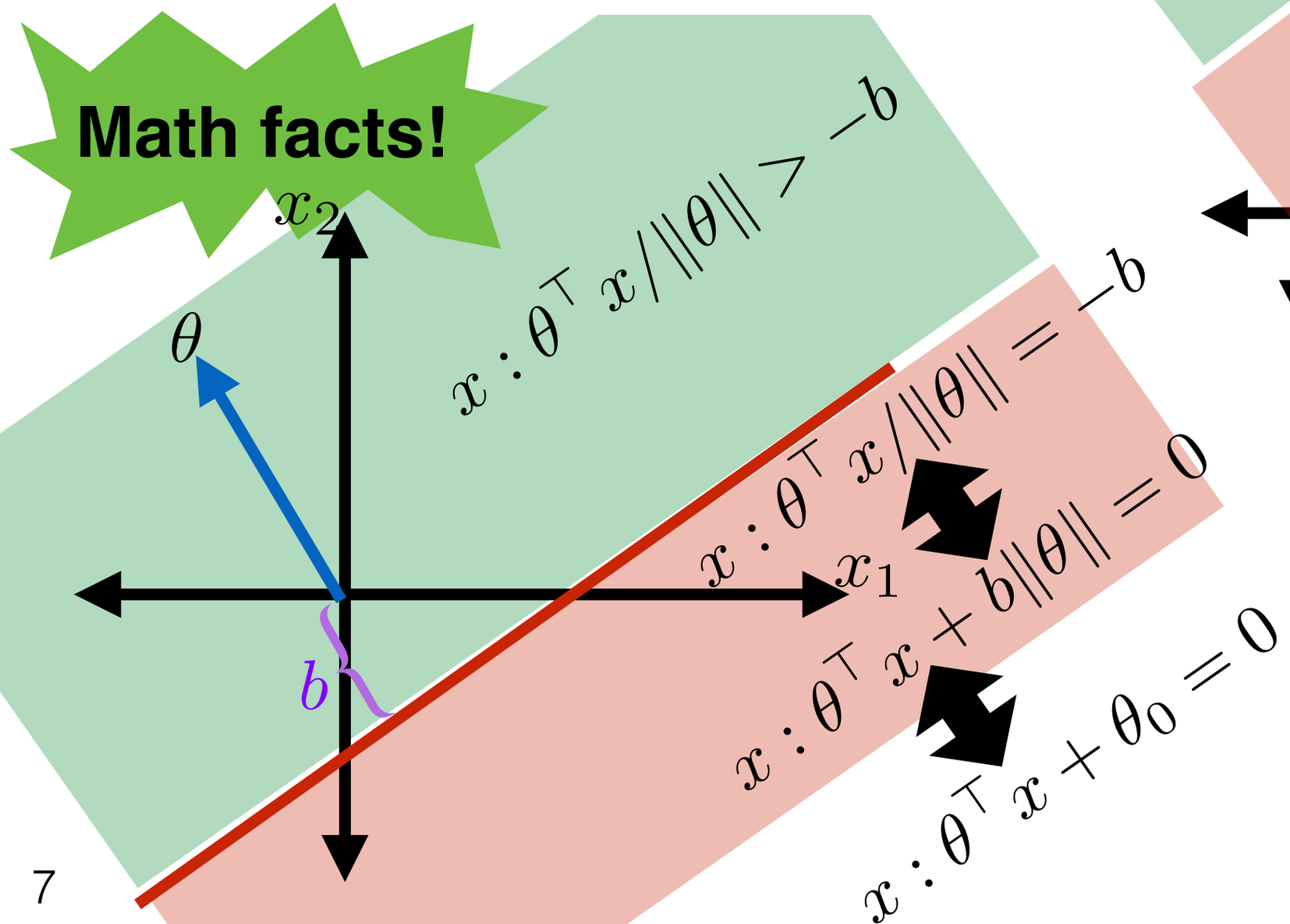


Linear classifiers

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

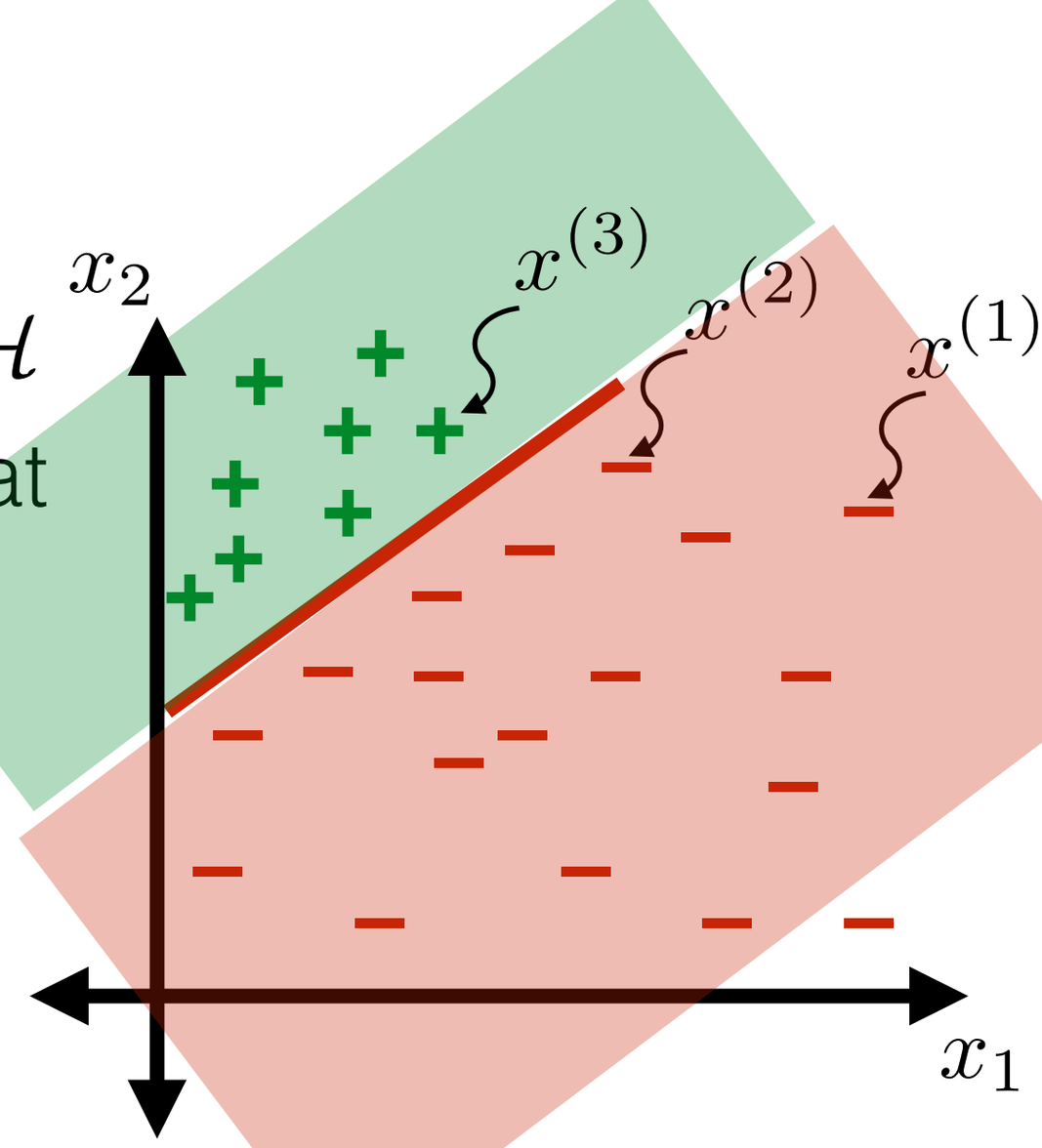


Math facts!

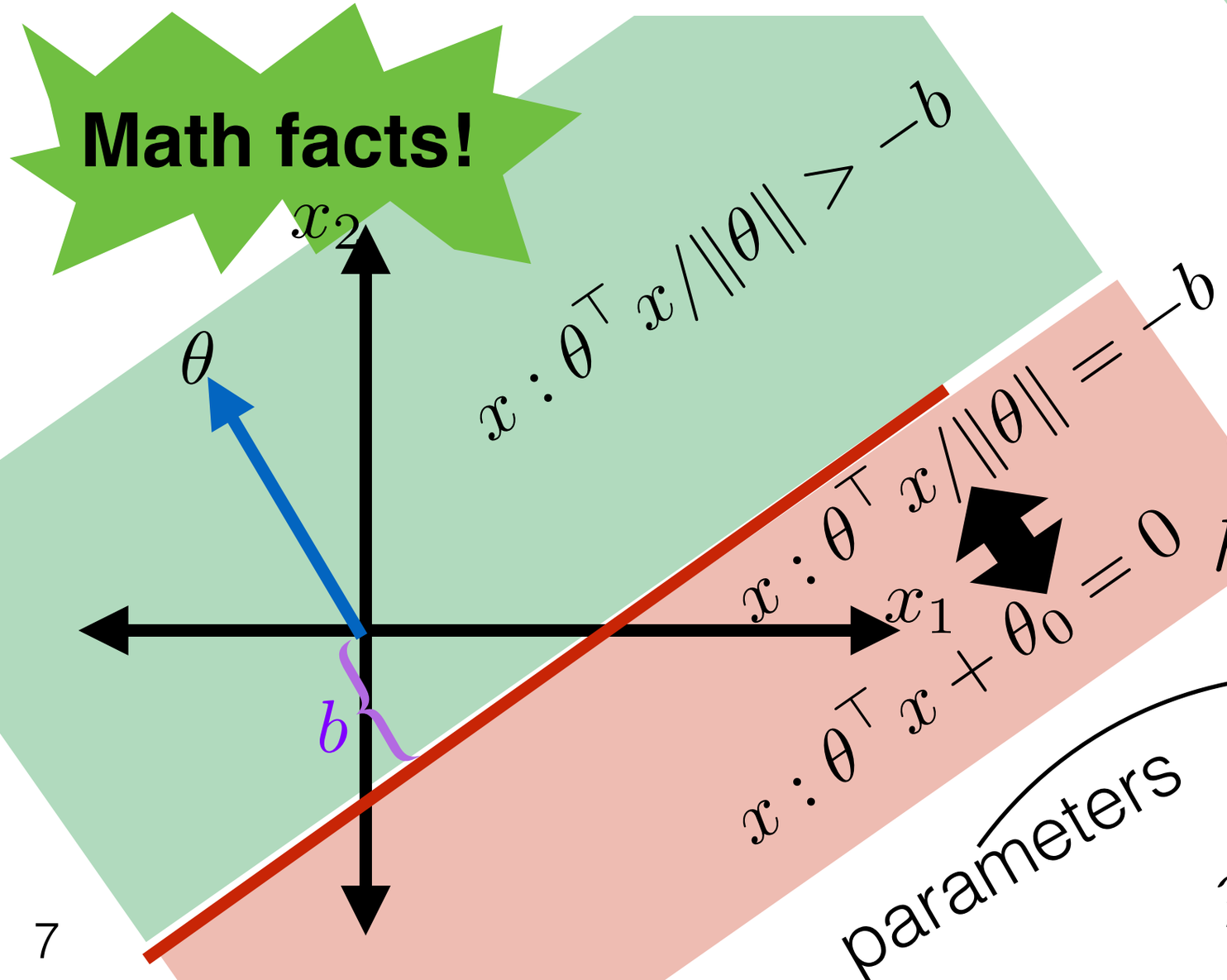


Linear classifiers

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



Math facts!



- Linear classifier:

$$h(x; \theta, \theta_0) = \text{sign}(\theta^\top x + \theta_0)$$

parameters

$$= \begin{cases} +1 \text{ if } \theta^\top x + \theta_0 > 0 \\ -1 \text{ if } \theta^\top x + \theta_0 \leq 0 \end{cases}$$

$\mathcal{H}$ = set of all such h

How good is a classifier?

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

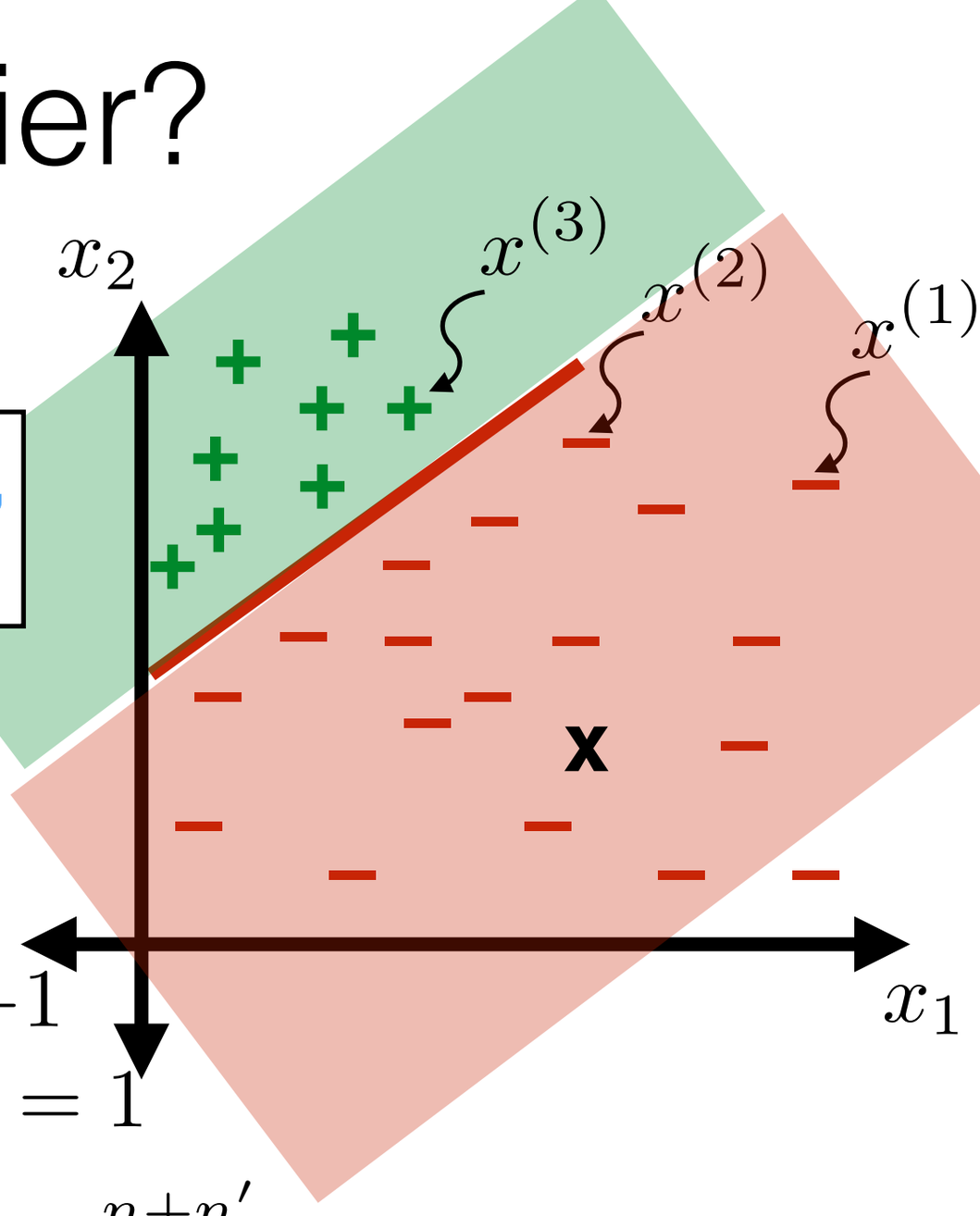
g: guess,
a: actual

  - Example: 0-1 loss

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

  - Example: asymmetric loss

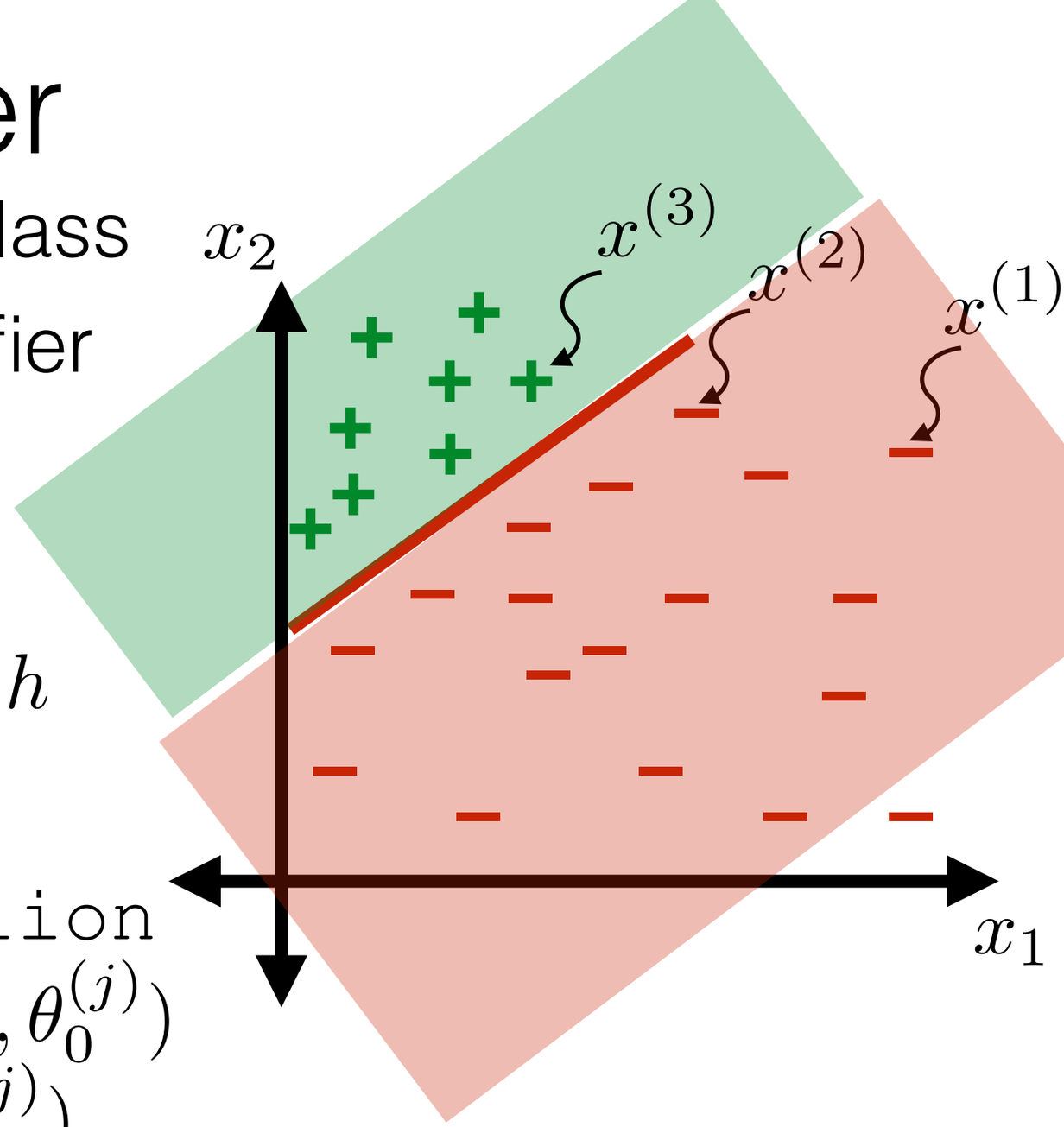
$$L(g,a) = \begin{cases} 1 \text{ if } g = 1, a = -1 \\ 100 \text{ if } g = -1, a = 1 \\ 0 \text{ else} \end{cases}$$



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

Learning a classifier

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



- Example:

```
for j = 1, …, 1 trillion
    Randomly sample (θ^(j), θ_0^(j))
    Set h^(j)(x) = h(x; θ^(j), θ_0^(j))
Ex_learning_alg( D_n ; k < 1 trillion)
    Set j* = argmin_{j∈{1,…,k}} ε_n(h^(j))
    Return h^(j*)
```

hyperparameter

- How does training error of `Ex_learning_alg`$(\mathcal{D}_n;1)$ compare to the training error of `Ex_learning_alg`$(\mathcal{D}_n;2)$?