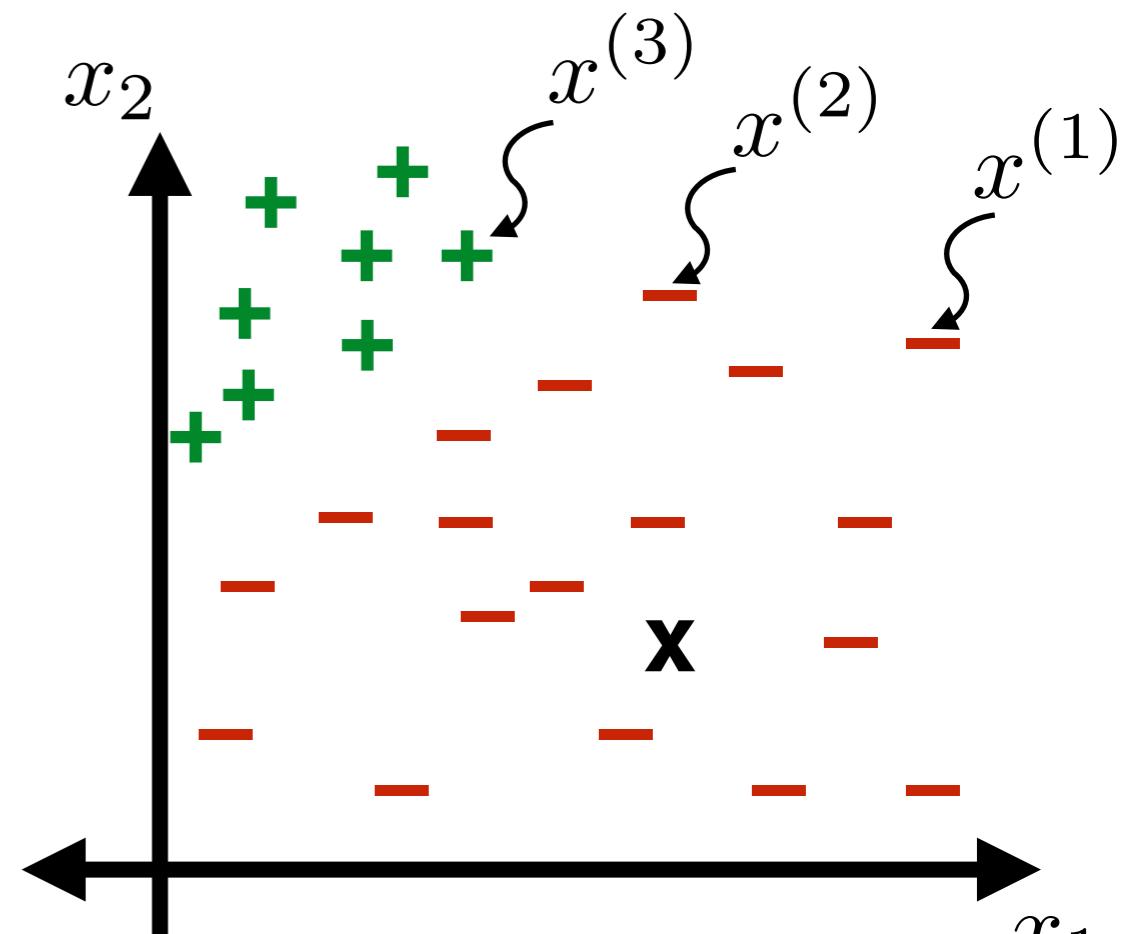


Getting started

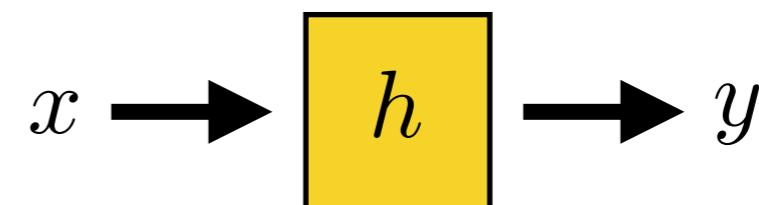
What do we have? (Training) data

- 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 $\mathcal{D}_n = \{(x^{(1)}, y^{(1)}), \dots, (x^{(n)}, y^{(n)})\}$



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

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



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

INTRO TO MACHINE LEARNING

classmate

Date _____

Page _____

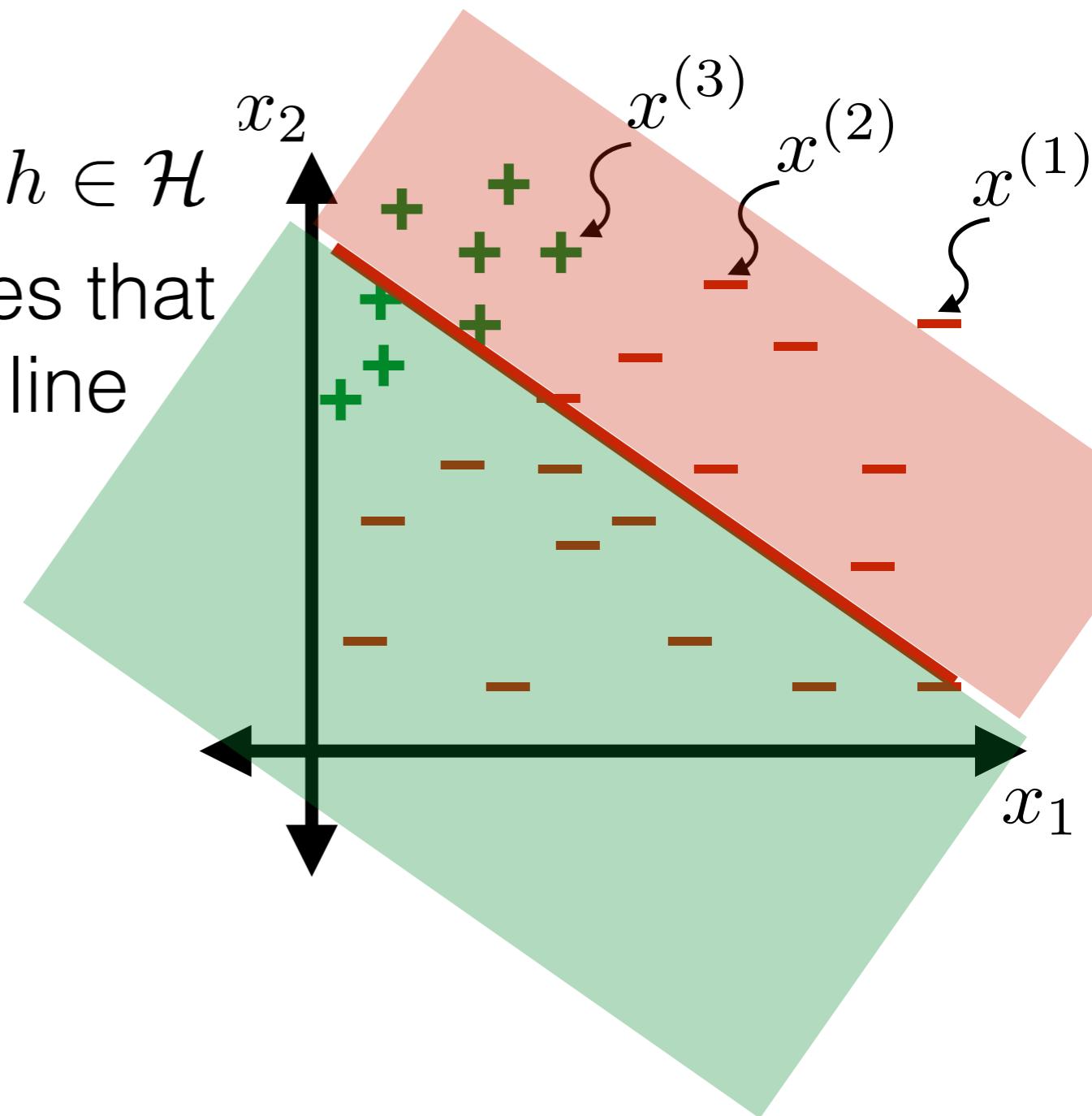
Lecture 1 MIT 6.036: Introduction to Machine Learning (Tamara Broderick)

* Slide 79 (Linear classifiers) (Intro)

- You can assume for example, we want to determine whether whether a new born baby will suffer a seizure after being kept in an incubator.
- Assume we have 'n' babies to study.
For an i^{th} baby, we have two things:
 - Feature vector with 'd' dimensions. These vectors have ~~reasons that the baby~~ parameters such as amount of oxygen in the incubator, temperature of incubator, etc. & they have ~~a~~ values.
 - Label $\in \{-1, 1\}$. This label indicates whether the baby suffer a seizure or not.

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

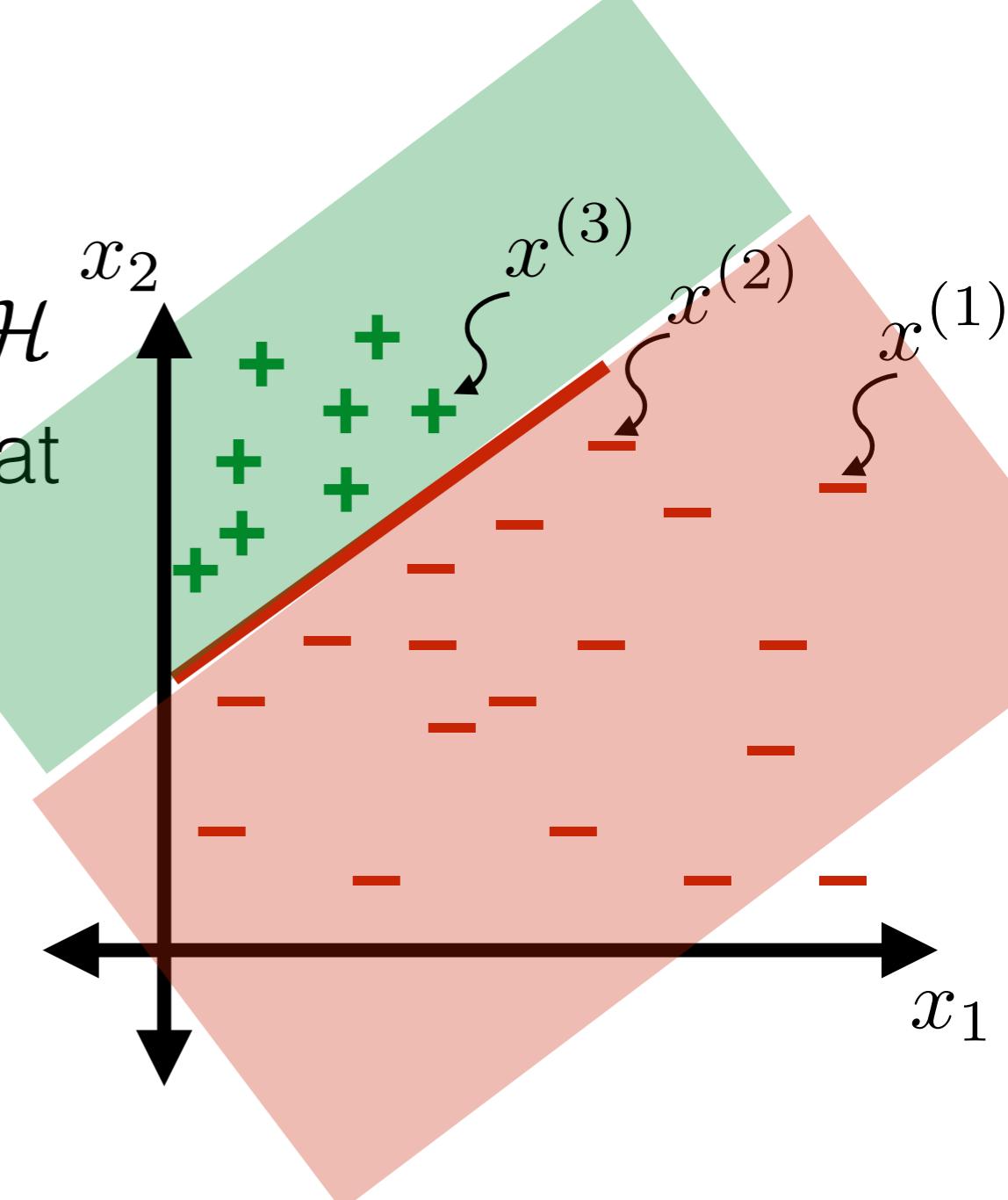
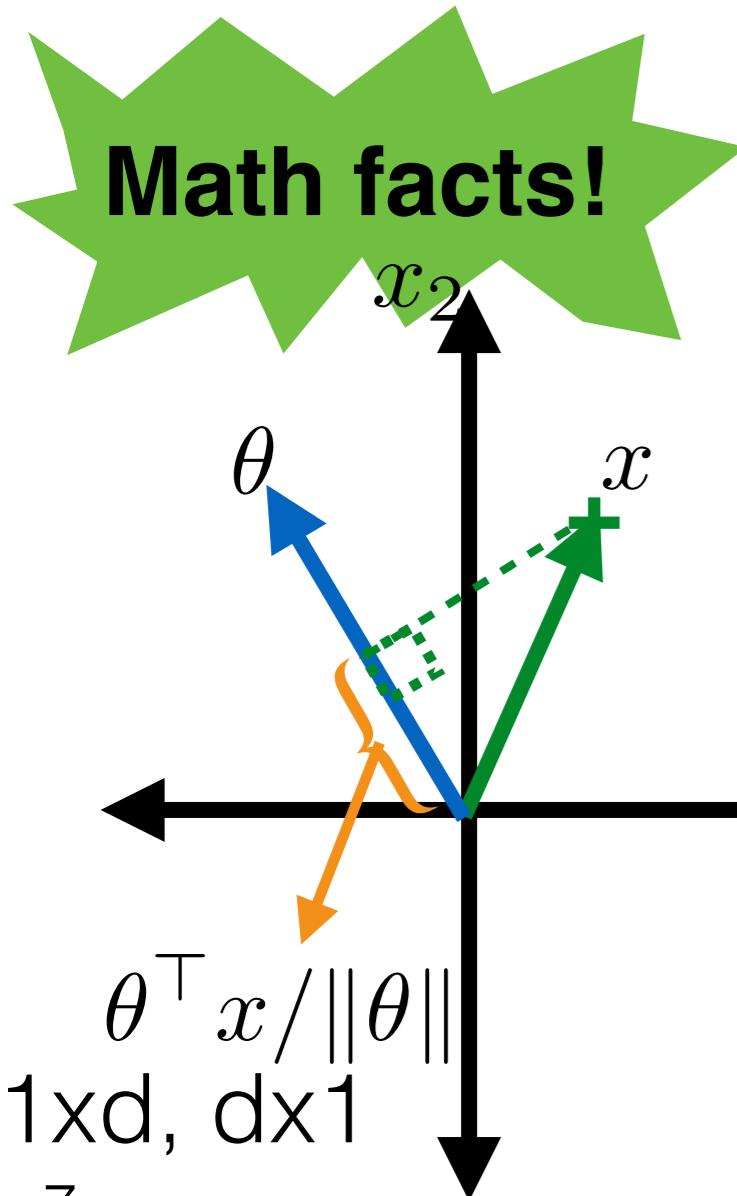


* Slide 90 (Linear Classifiers)

- Hypothesis is a function that for an ~~unknown~~ random new baby with an unknown label; predicts the label from the feature vector.
- Hypothesis class is a class or set of all candidate hypotheses.
- In this case (linear classifier), the hypothesis is a line, which divides the dimensional space.
- This one (slide 90 example) is obviously a bad 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



~~governor~~ ~~kinematics~~ * Slide 110 minimum of midboard II : 280.2 TIM Dowlag

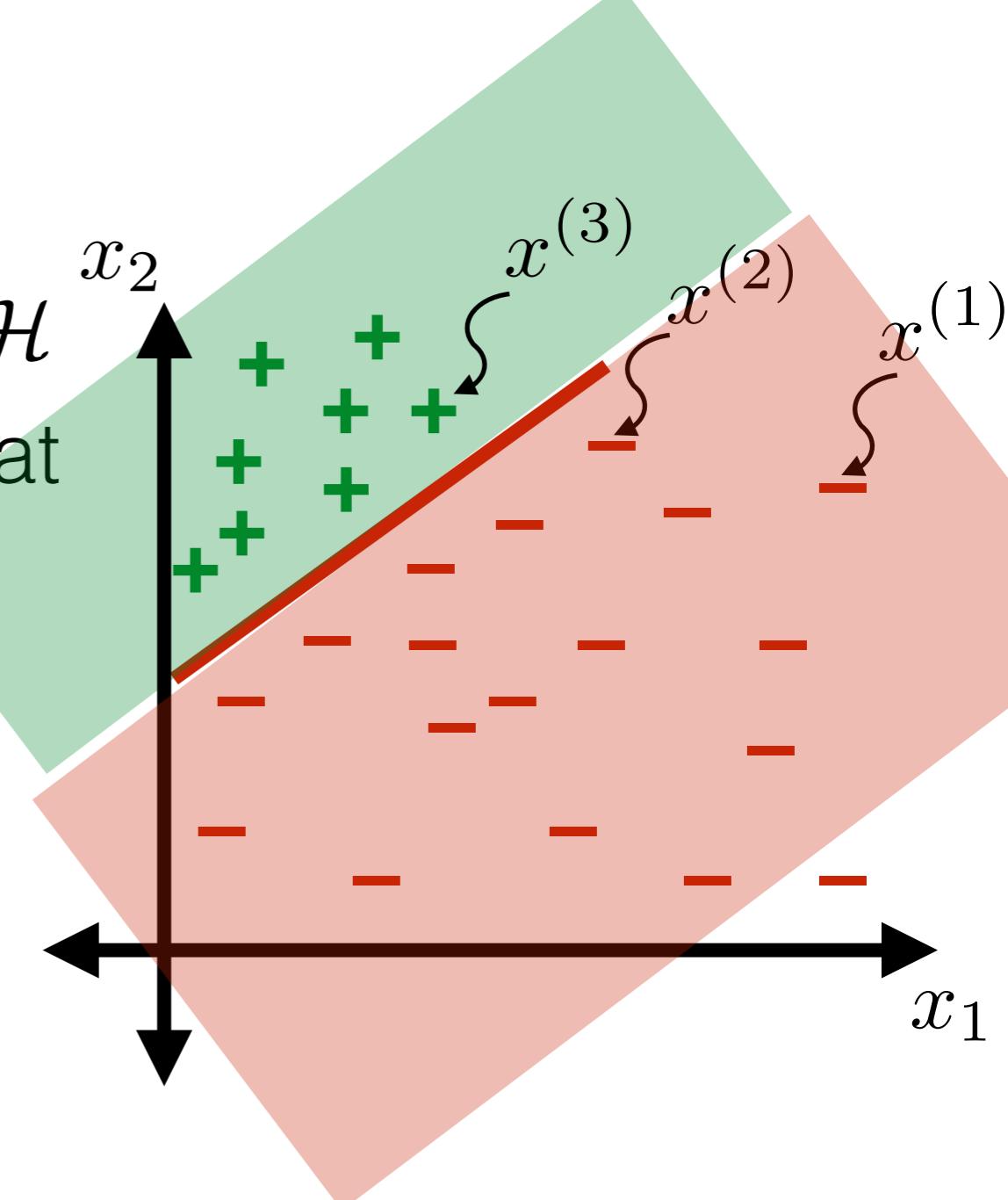
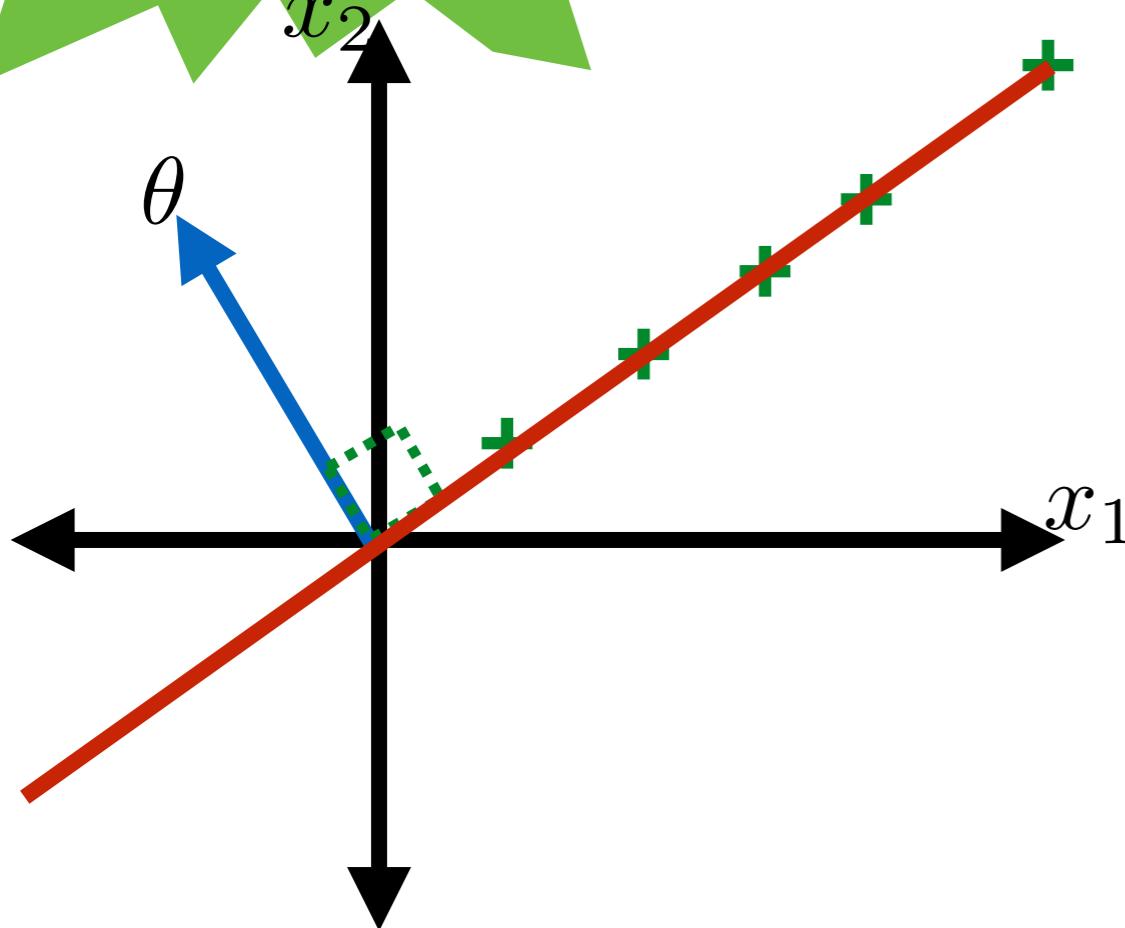
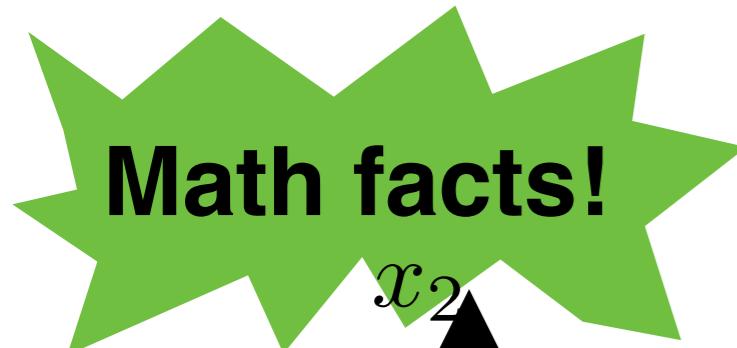
(contd) (midboard point) PT shift *

- Here, θ & x are two column vectors. & the projection of vector x on vector θ is given by
$$\frac{\theta^T x}{|\theta|}$$
 i.e. dot product of θ & x divided by magnitude of θ .
orthogonal no of equal princi
- This projection can tell how much close are the directions in which θ & x are pointing. You can imagine that if x is rotated more in direction of θ , the projection will increase.
- The projection on θ won't change even if θ is scaled upwards.

* Slide 119 & slide 134 Slide 119

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



~~slide 119~~ & slide 134 Slide 119

(normal) vector of slide

- Now, the projecting line will always be perpendicular to θ . All the points lying on this line will have same projection on θ . Hence, we can use this line as a classifier.
- All the points above this line will have projection greater than points lying on the line. & all points below this line will have projection lower than points lying on the line.

We can define this line as:-

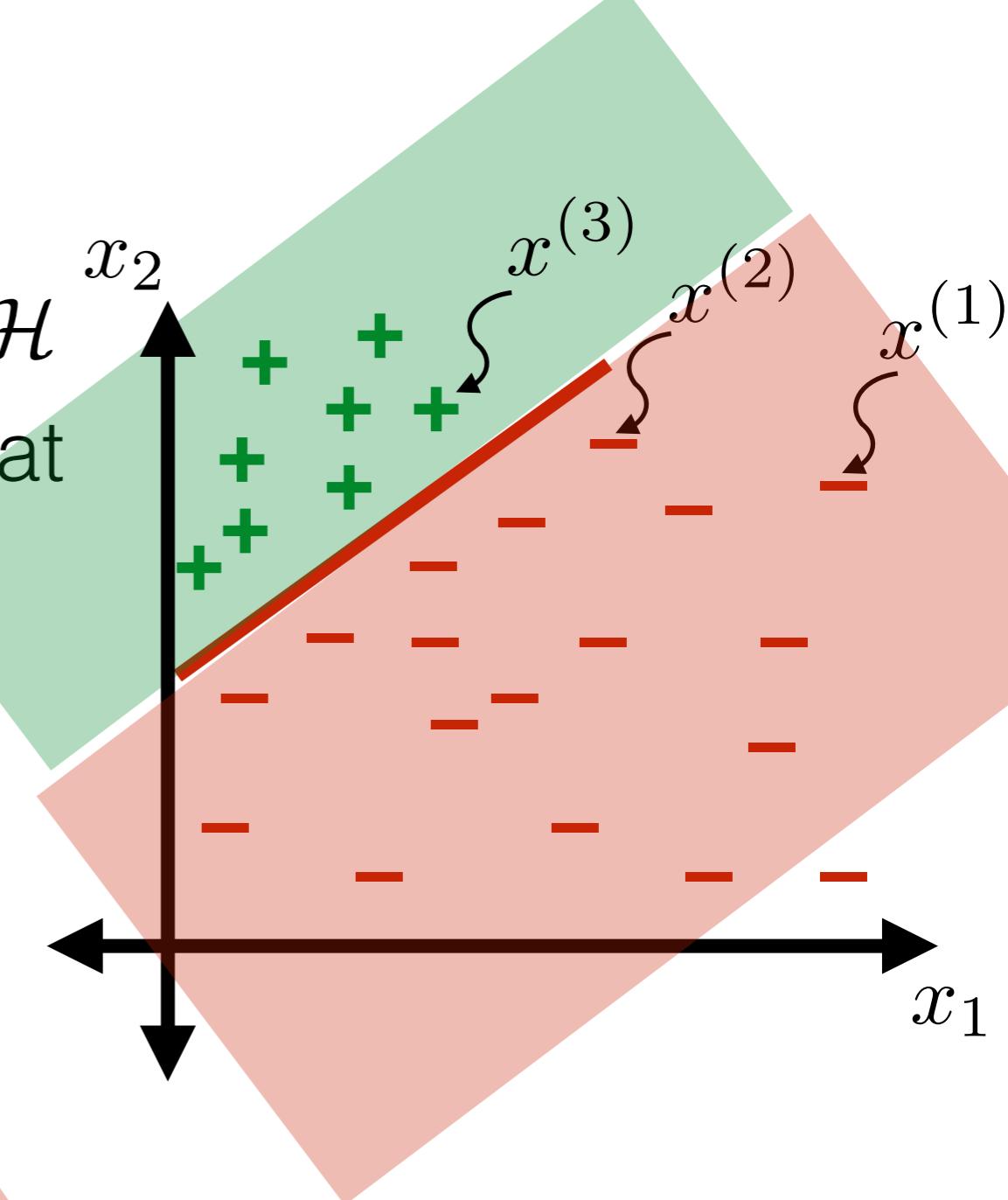
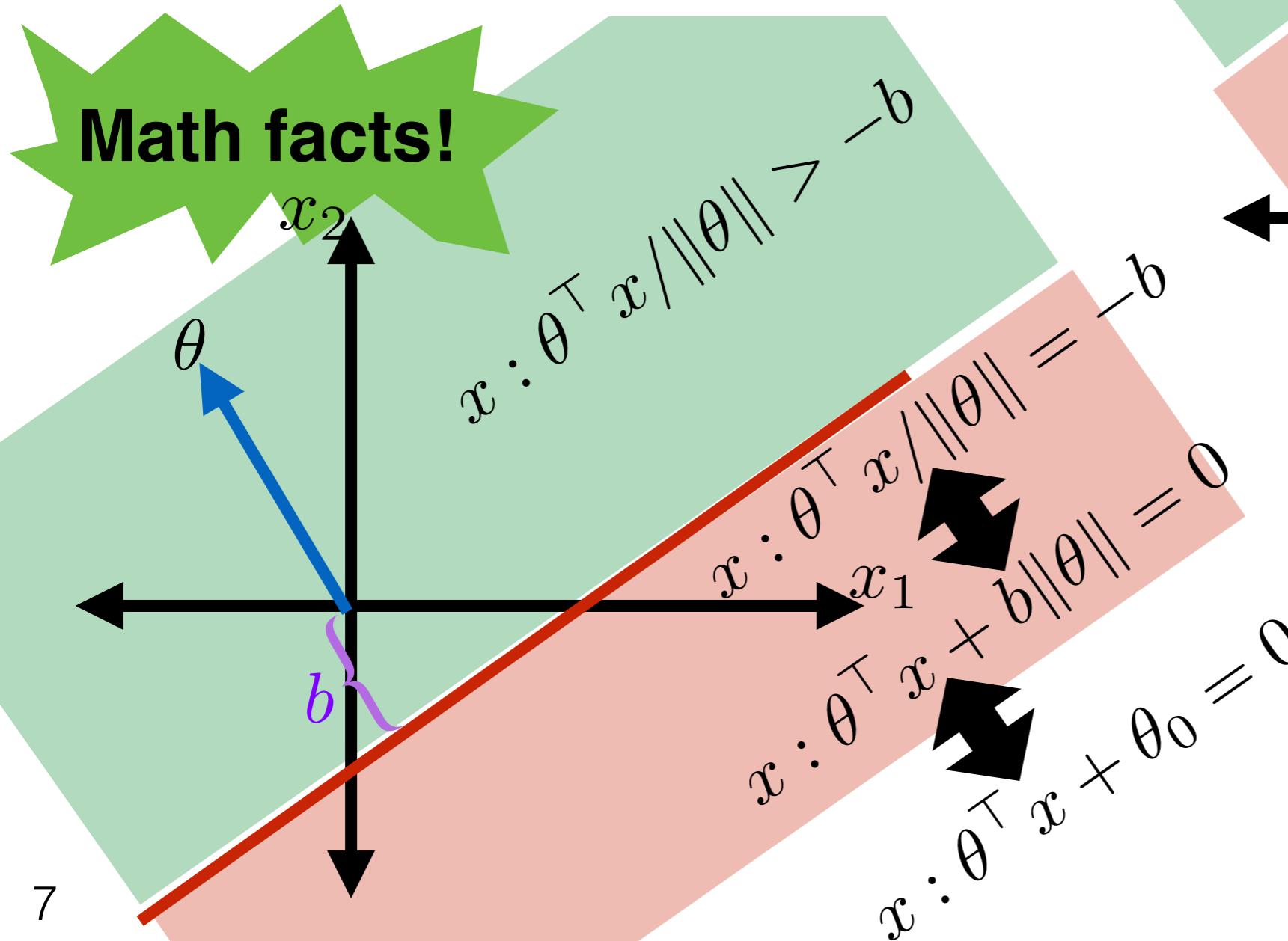
$$x : (\theta^T x) / |\theta| = p$$

i.e. set of all points x such that projection of x on θ is p .

We can alternatively

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



* Slide 134 Slide 135

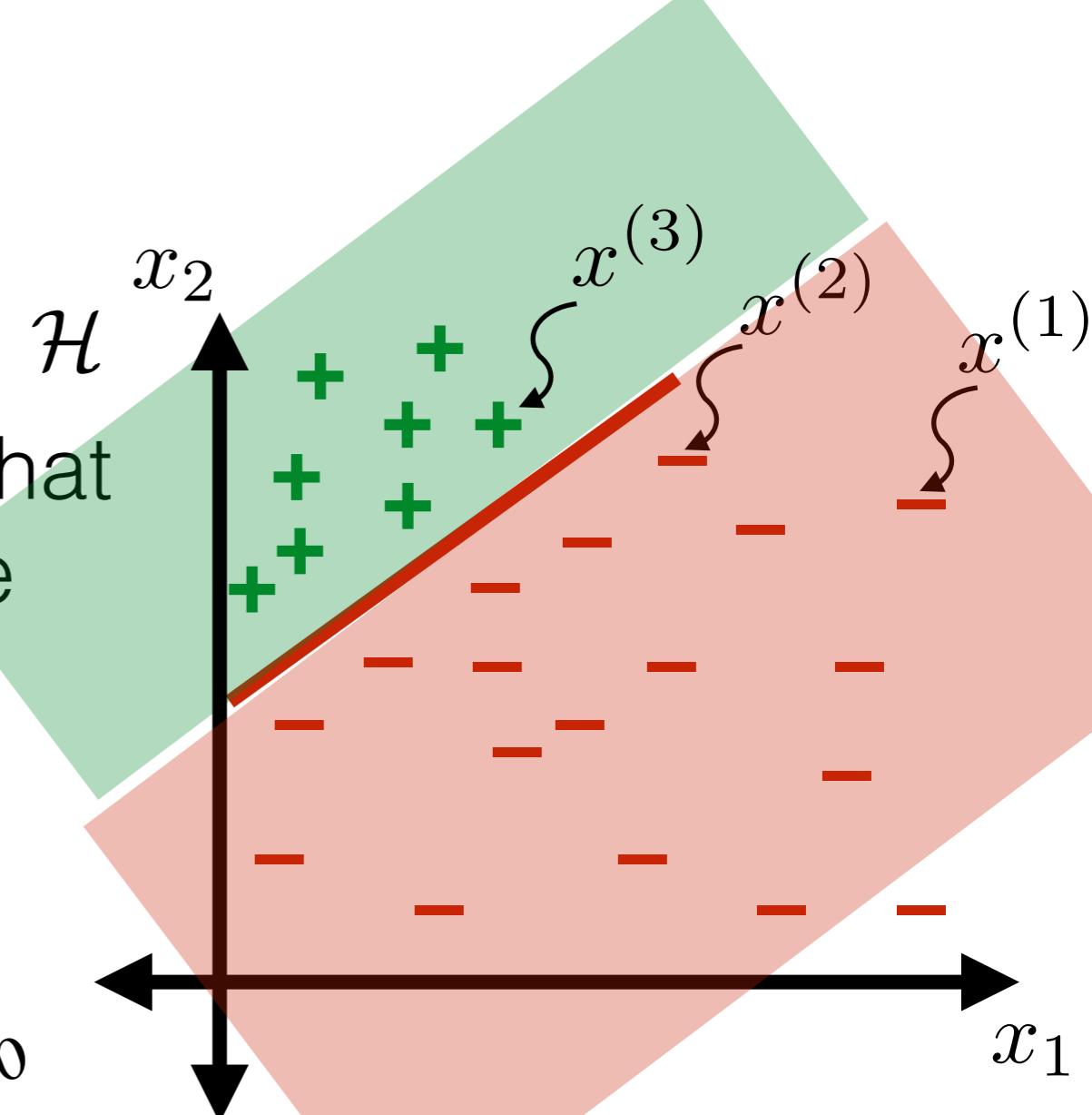
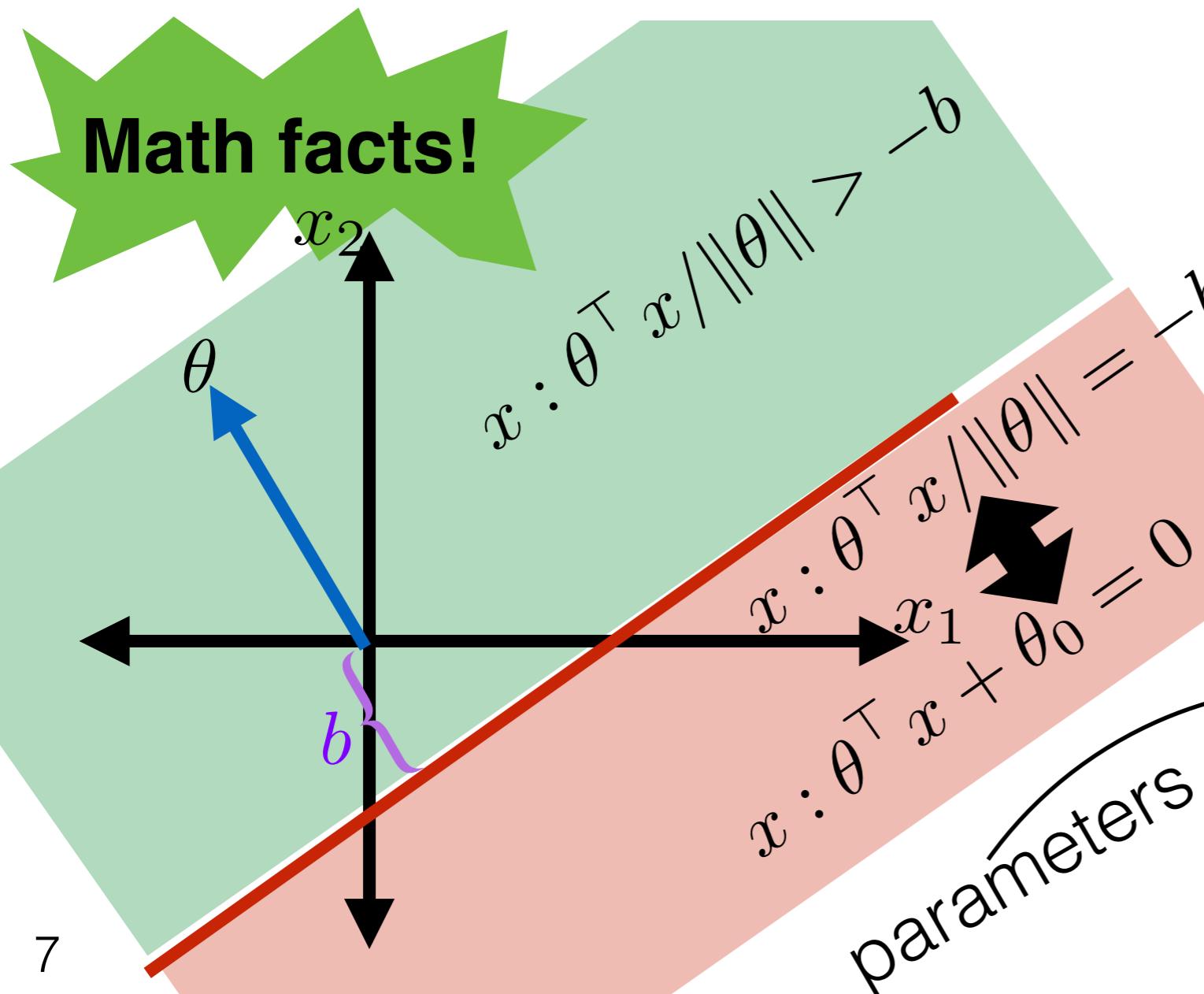
- We can alternatively define the line as:-

$$\mathbf{x} : \boldsymbol{\theta}^T \mathbf{x} + \theta_0 = 0$$

where, $\theta_0 = -p \cdot |\boldsymbol{\theta}|$

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 classifier:
$$h(x; \theta, \theta_0) = \text{sign}(\theta^\top x + \theta_0)$$
$$= \begin{cases} +1 & \text{if } \theta^\top x + \theta_0 > 0 \\ -1 & \text{if } \theta^\top x + \theta_0 \leq 0 \end{cases}$$
- $\mathcal{H} = \text{set of all such } h$

(de)stributional animal

* Slide 146

- The semicolon in the $h(x; \theta, \theta_0)$ distinguishes x from θ, θ_0 & tells that x is an input, but θ & θ_0 are just identifying parameters.
- Just as we use i to denote i^{th} index or value, we use θ & θ_0 here.

How good is a classifier?

- Should predict well on future data

- How good is a classifier at a single point? Loss $L(g, a)$

- Example: 0-1 loss

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

- Example: asymmetric loss

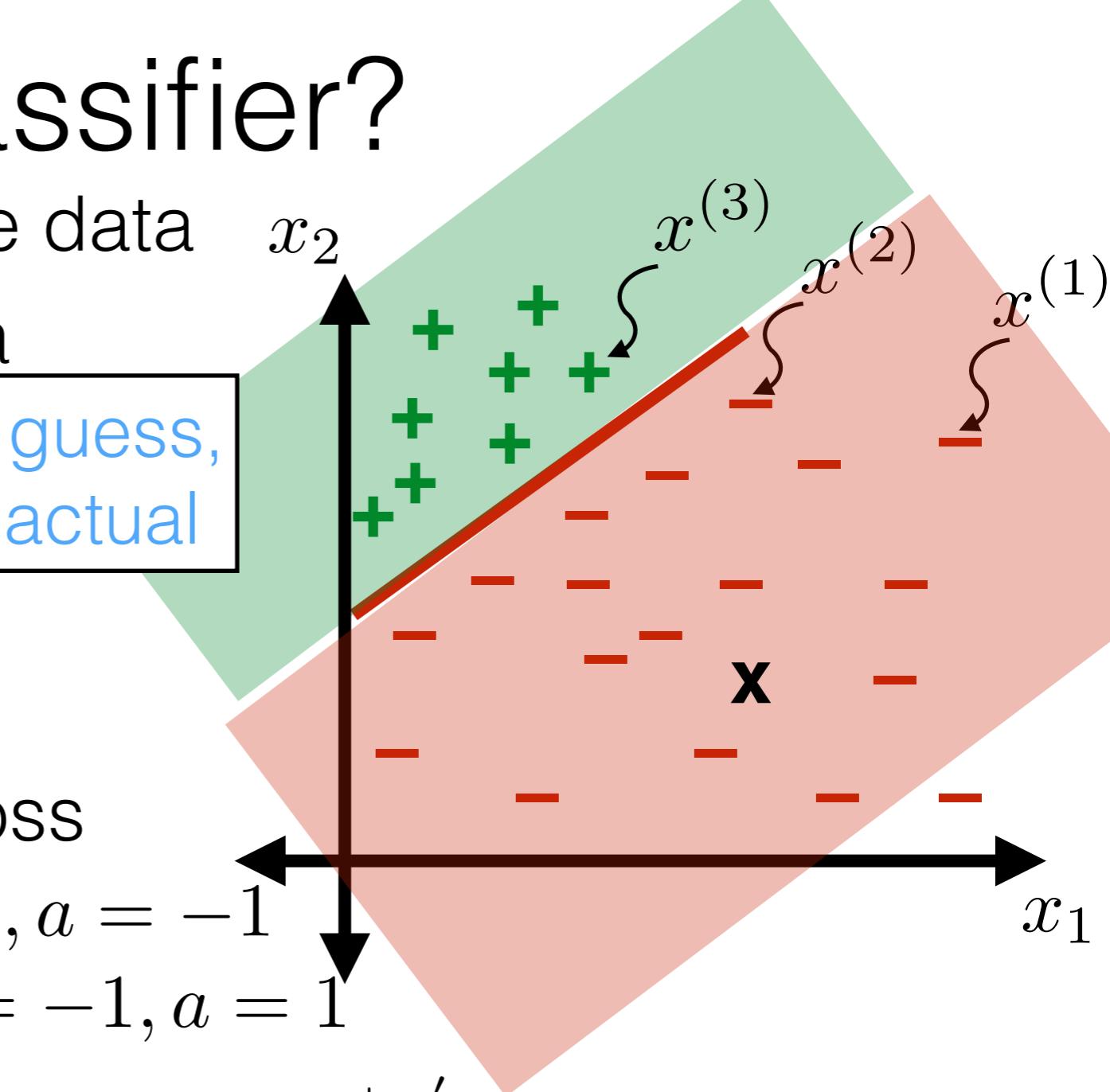
$$L(g, a) = \begin{cases} 1 & \text{if } g = 1, a = -1 \\ 100 & \text{if } g = -1, a = 1 \\ 0 & \text{else} \end{cases}$$

- 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 $\mathcal{E}_n(h) < \mathcal{E}_n(\tilde{h})$

**g: guess,
a: actual**



* Slide 161 (How good is a linear classifier)

- In the case of loss, we want to it is preferred to differentiate losses from each other. We want to differentiate how much ^{one} type of loss affects us & how other type of loss affects us. (~~we can't~~ Here, types of losses does NOT mean 0-1 & assymetric loss)
- So, we prefer assymetric to 0-1.
- In the case of assymetric, we can compare the ($g=1, a=-1$) as this: we ~~predicted~~ determined using our model for a new baby that this baby will have seizure. So, the doctor checks up on it & ~~it~~ doesn't have a seizure. So, everything's fine but we got ~~the~~ loss.

- In case of ($g = -\frac{1}{3}$, $a = 1$) , i.e. we predict that the baby will ~~have a seizure~~ not have a seizure. So, the doctor doesn't check up on it. But the baby does suffer from seizure & there's no one to tend to it. Now, this is a greater loss than previous one. Hence, we associate higher loss value to it.

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

- Example:

for $j = 1, \dots, 1$ trillion

Randomly sample $(\theta^{(j)}, \theta_0^{(j)})$

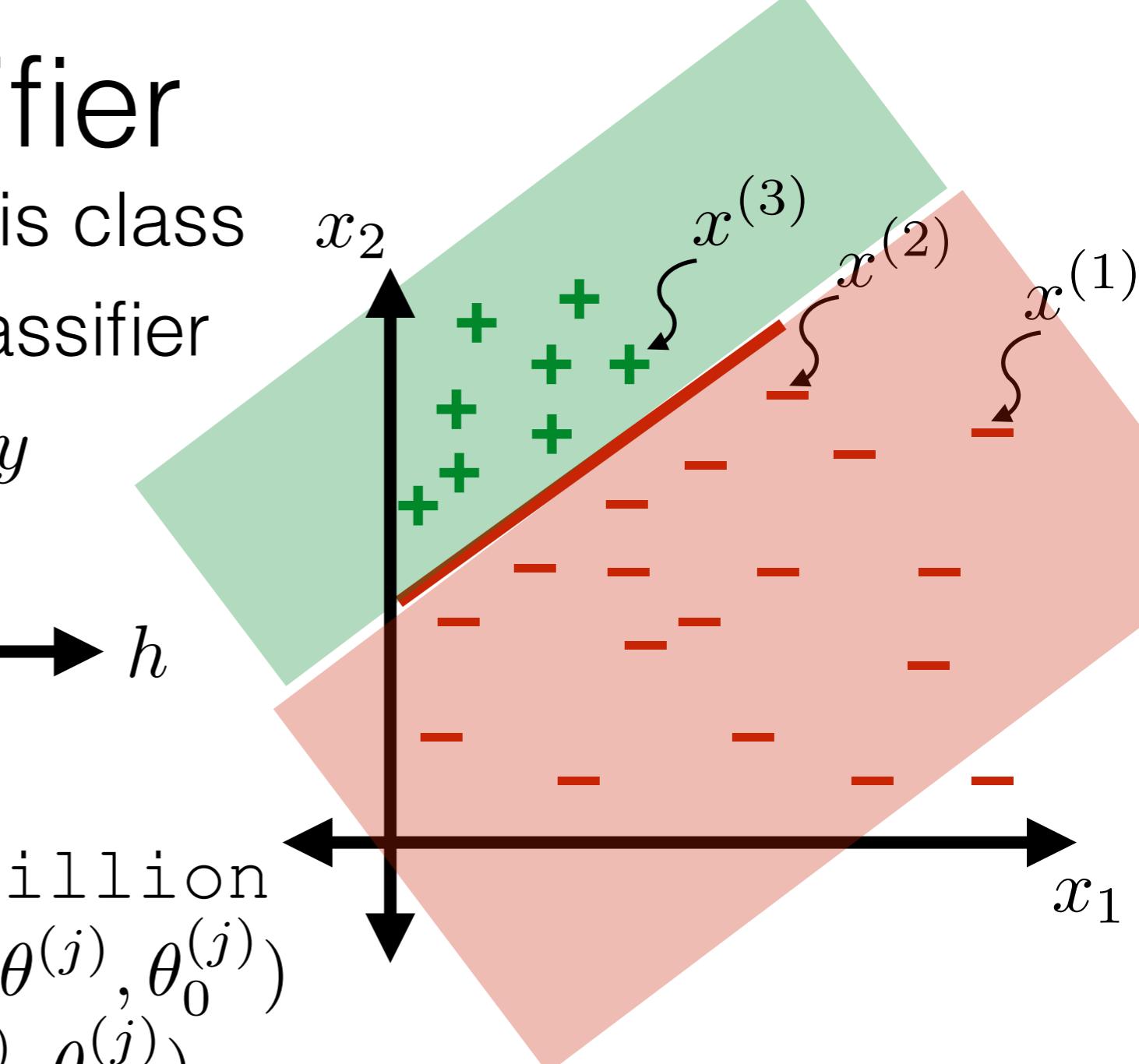
Set $h^{(j)}(x) = h(x; \theta^{(j)}, \theta_0^{(j)})$

Ex_learning_alg(\mathcal{D}_n ; $k < 1$ trillion)

Set $j^* = \operatorname{argmin}_{j \in \{1, \dots, k\}} \mathcal{E}_n(h^{(j)})$

Return $h^{(j^*)}$

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



* Learning a classifier (slide last)

- To answer the last question, the training error of hypothesis returned by $\text{alg}(D_n; 2)$ will always be lower than hypothesis returned by $\text{alg}(D_n; 1)$.
- You can prove that by proving that the inverse can't be true. Suppose that $\text{alg}(D_n, p)$ splits out hypothesis 'a' & $\text{alg}(D_n, q) \cancel{\text{spits out}}$ is taken into consideration where $q > p$. Now, in the 'p' hypothesis, 'a' had the least error. Now when we look for further 'q-p' hypothesis, they may have all hypothesis with error greater than 'a' & will spit out previous result 'a'. If 'q-p' has at least one hypothesis with error less than 'a', it will return that hypothesis. Thus, for ~~for~~ $q > p$, $\text{alg}(D_n, q)$ will spit out hypothesis with error strictly lower than or equal to $\text{alg}(D_n, p)$.