Master studije
GEOINFORMATIKA

Mašinsko učenje - Linearni klasifikatori. Percepton.

Tatjana Jakšić Krüger

tatjana@turing.mi.sanu.ac.rs

7. decembar 2021.

Građevinski fakultet Univerziteta u Beogradu

Podsetnik

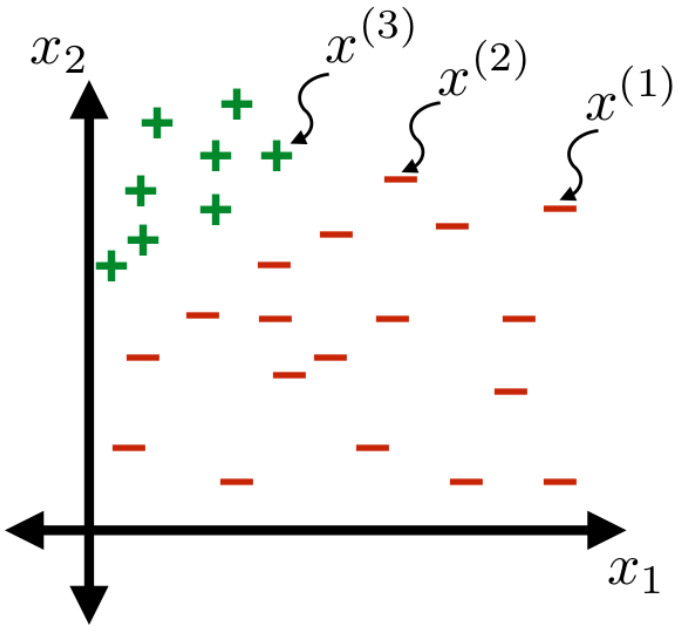
- Uopšteno linearno modelovanje.
- Objasnili smo šta je funkcija veze.
- Opisali logit (sigmoid) funkciju.
- Opisali Njutnovu metodu.
- Primer klasifikacije spam poruka.

Cilj za danas

- Linearni klasifikatori.
- Perceptron.
- Podsećanje na logističku regresiju/linearnu logističku klasifikaciji.

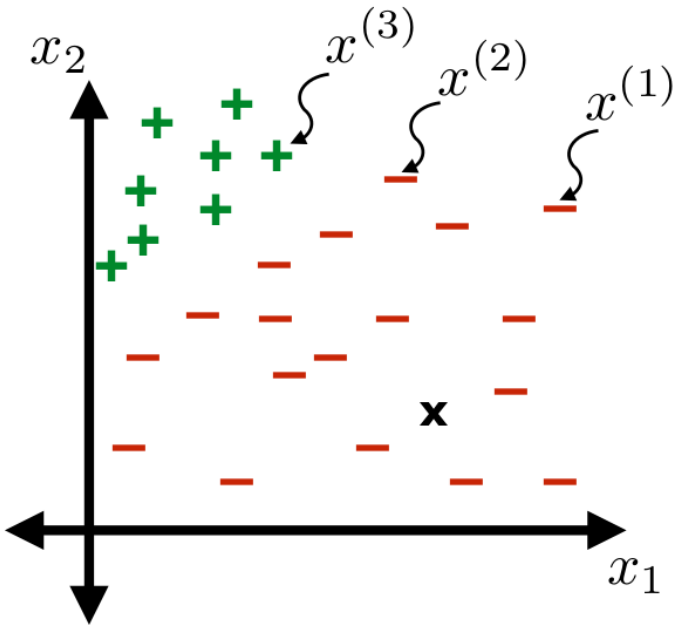
Linearni klasifikator

- Rekli smo da je tipično kodiranje sa 1 i 0. Tipično je i $+1$ i -1.
- Postavka problema:
  - Ulazni podacu su m svojstava: $x^{(i)} = (x_1^{(i)}, \ldots, x_d^{(i)}) \in \mathcal{R}^d$, gde $i \in \{1, \ldots, n\}$.
  - Ishod je binaran i kodiran sa $+1$ i -1.
- Primer. Srčani udar sa svojstvima x_1 (krvni pritisak) i x_2 (starost pacijenta). Obeležje (labela) je $+1$ ako je pacijent imao srčani udari, -1 ukoliko nije imao srčani udar.
- Pretpostavimo da imamo situaciju na slici! Neka je skup za obučavanje skup $D_n = \{(x^{(1)}, y^{(1)}), \ldots, (x^{(n)}, y^{(n)})\}$.
- Definišimo hipotezu $h : \mathcal{R}^d \to \{+1, -1\}$



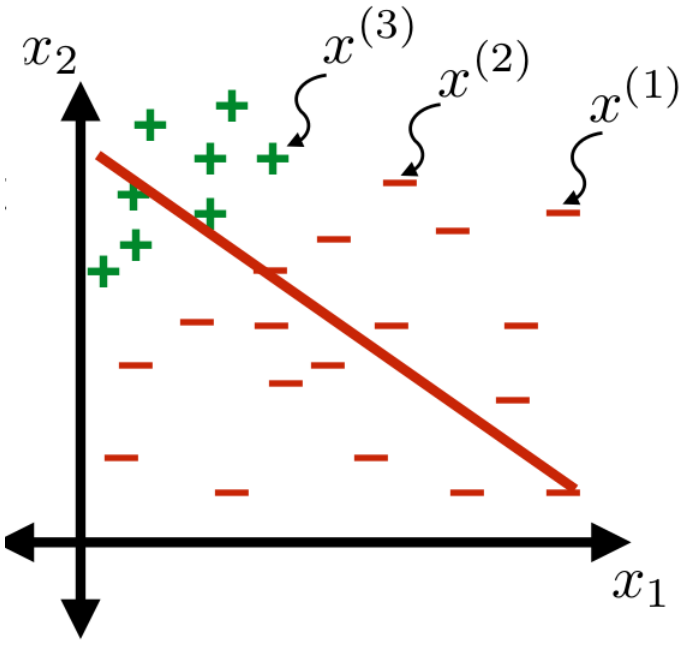
Linearni klasifikator

- Rekli smo da je tipično kodiranje sa 1 i 0. Tipično je i $+1$ i -1.
- Postavka problema:
  - Ulazni podacu su m svojstava: $x^{(i)} = (x_1^{(i)}, \ldots, x_d^{(i)}) \in \mathcal{R}^d$, gde $i \in \{1, \ldots, n\}$.
  - Ishod je binaran i kodiran sa $+1$ i -1.
- Primer. Srčani udar sa svojstvima x_1 (krvni pritisak) i x_2 (starost pacijenta). Obeležje (labela) je $+1$ ako je pacijent imao srčani udari, -1 ukoliko nije imao srčani udar.
- Pretpostavimo da imamo situaciju na slici! Neka je skup za obučavanje skup $D_n = \{(x^{(1)}, y^{(1)}), \ldots, (x^{(n)}, y^{(n)})\}$.
- Definišimo hipotezu $h : \mathcal{R}^d \to \{+1, -1\}$



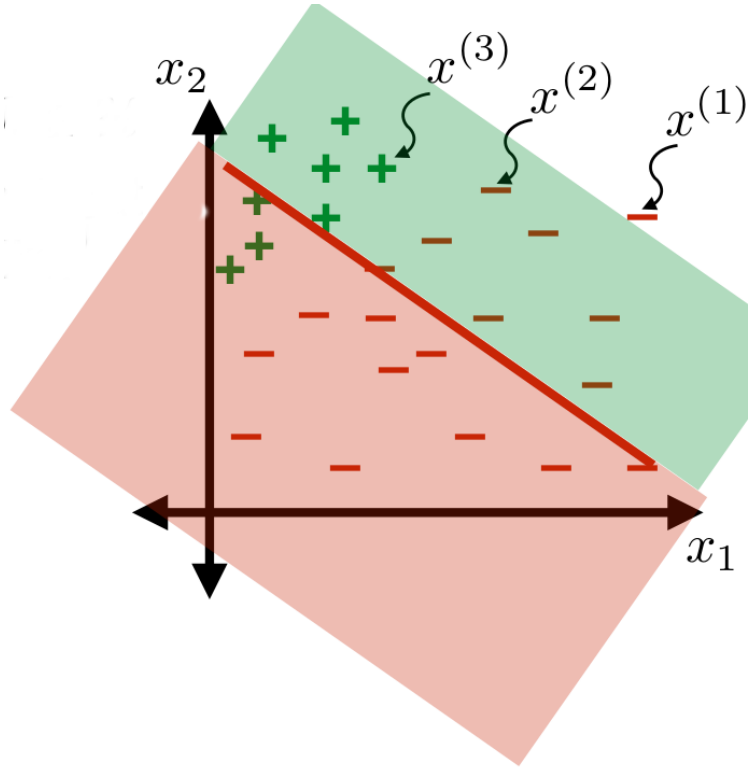
Linearni klasifikator

- Neka je $\mathcal{H}$ klasa hipoteza, odnosno, kolecija funkcija $h \in \mathcal{H}$.
- Na slici je prikazana jedna hipoteza koja pripada klasi gde je oznaka +1 sa jedne strane linije a -1 s druge strane.
- Ostatak prezentacije izveden je iz predavanja prof. Tamara Broderick (https://tamarabroderick.com/files/ml_6036_2020_lectures/broderick_lecture_02.pdf)



Linearni klasifikator

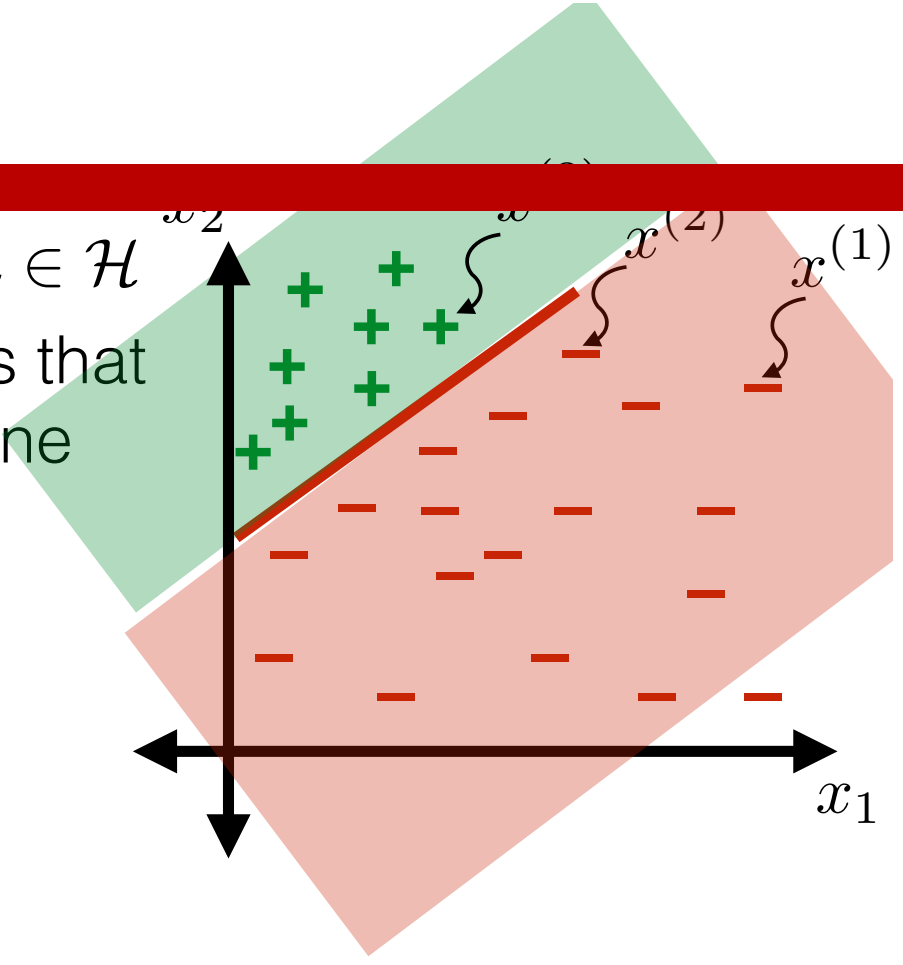
- Neka je $\mathcal{H}$ klasa hipoteza, odnosno, kolecija funkcija $h \in \mathcal{H}$.
- Na slici je prikazana jedna hipoteza koja pripada klasi gde je oznaka +1 sa jedne strane linije a -1 s druge strane.
- Ostatak prezentacije izveden je iz predavanja prof. Tamara Broderick (https://tamarabroderick.com/files/ml_6036_2020_lectures/broderick_lecture_02.pdf)



Linear classifiers

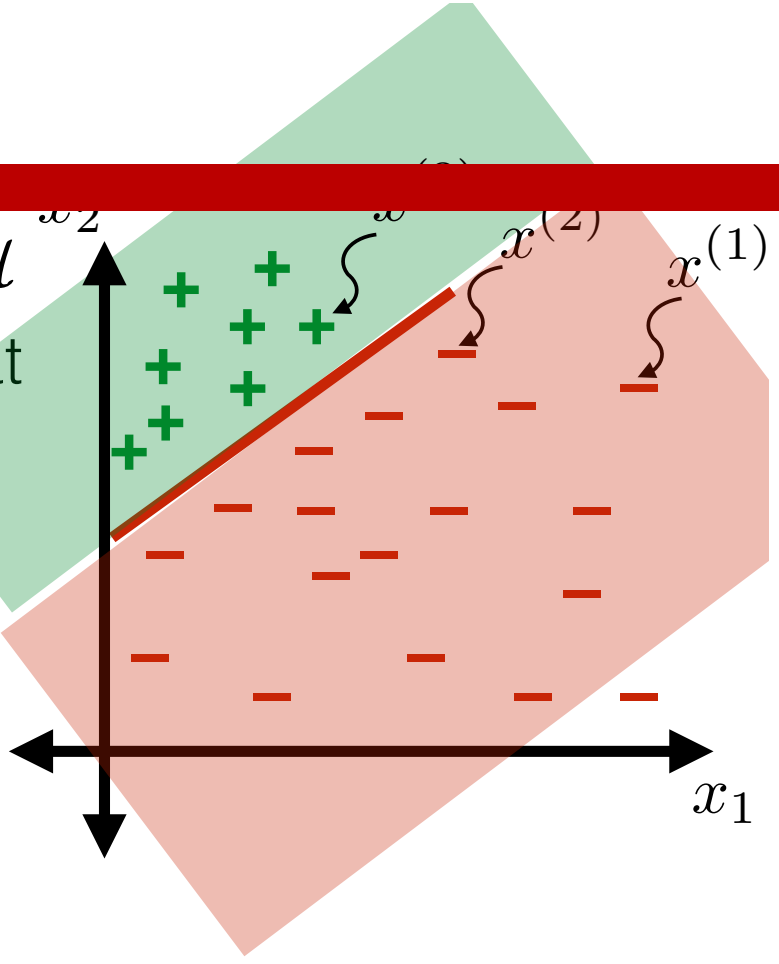
- Hypothesis class $\mathcal{H}$: set of $h \in \mathcal{H}$
  - Example $\mathcal{H}$: All hypotheses that label +1 on one side of a line and -1 on the other side

Math facts!

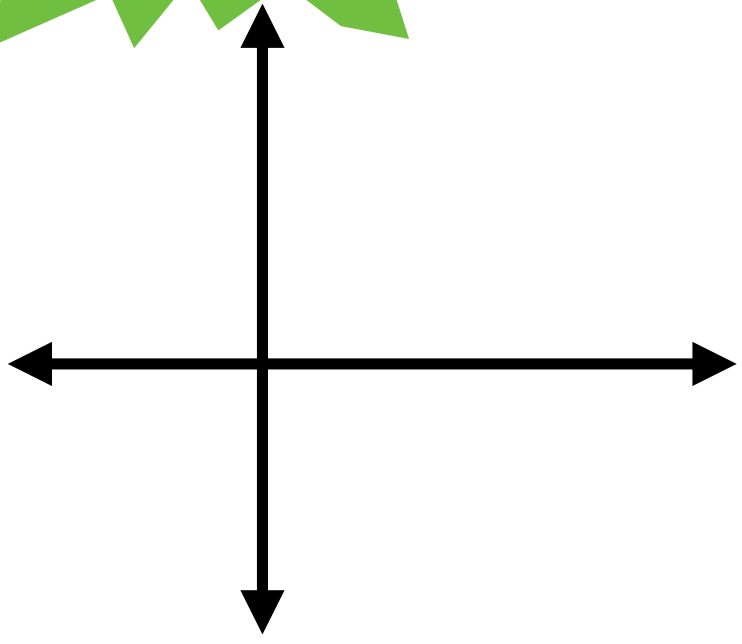


Linear classifiers

- 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

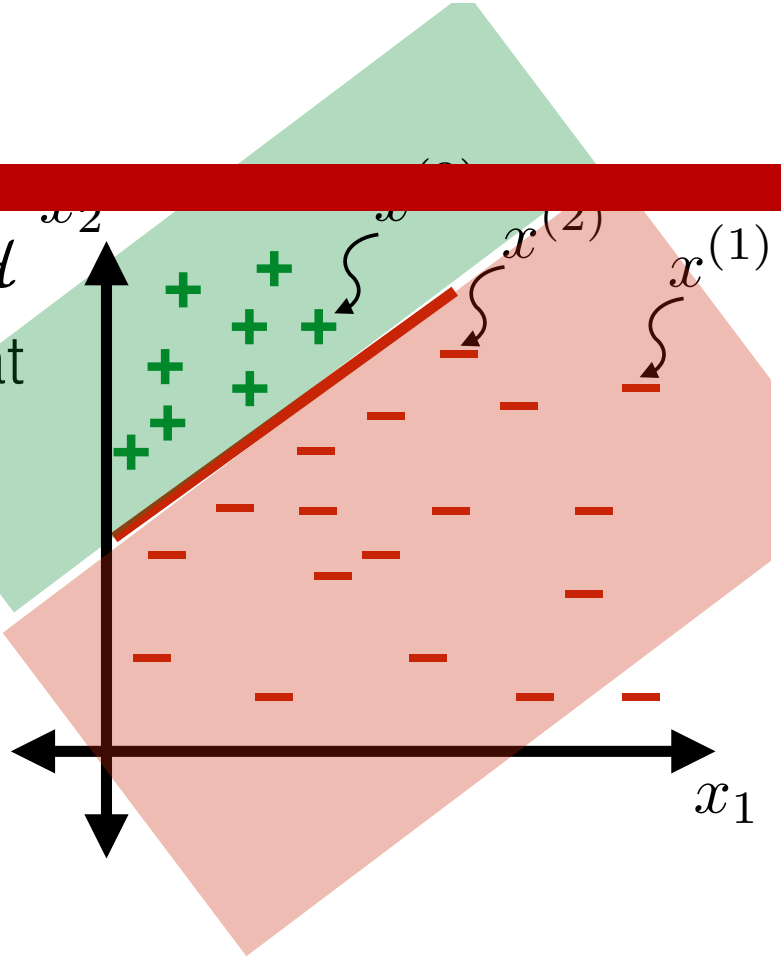


Math facts!

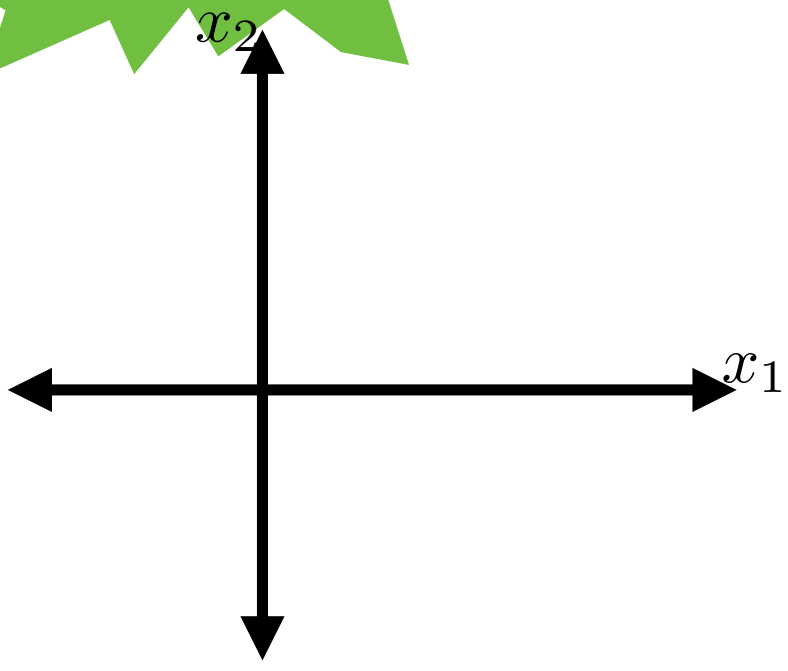


Linear classifiers

- 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

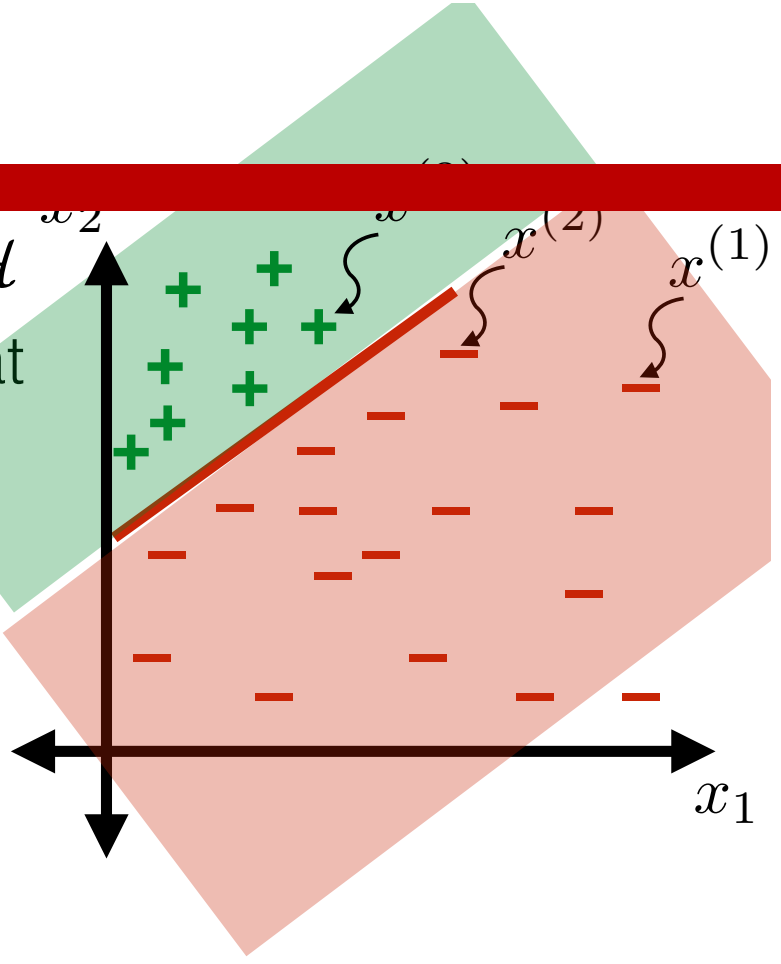


Math facts!

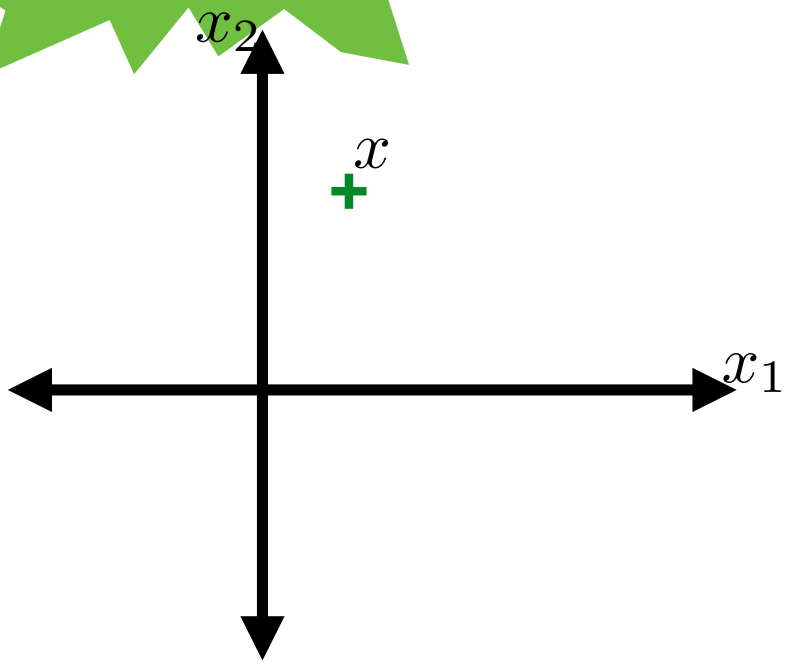


Linear classifiers

- 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

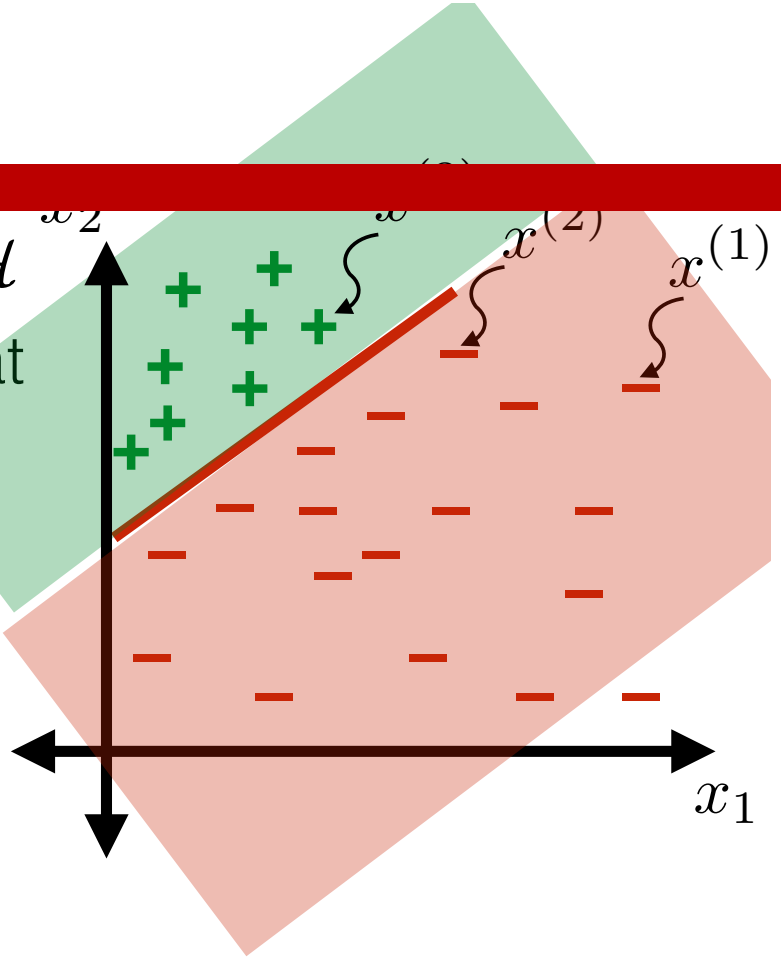


Math facts!

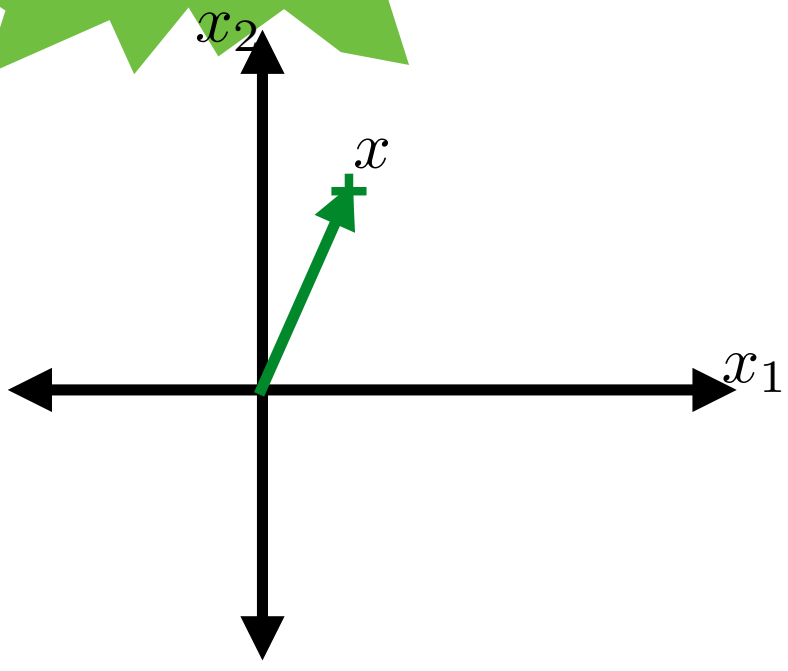


Linear classifiers

- 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

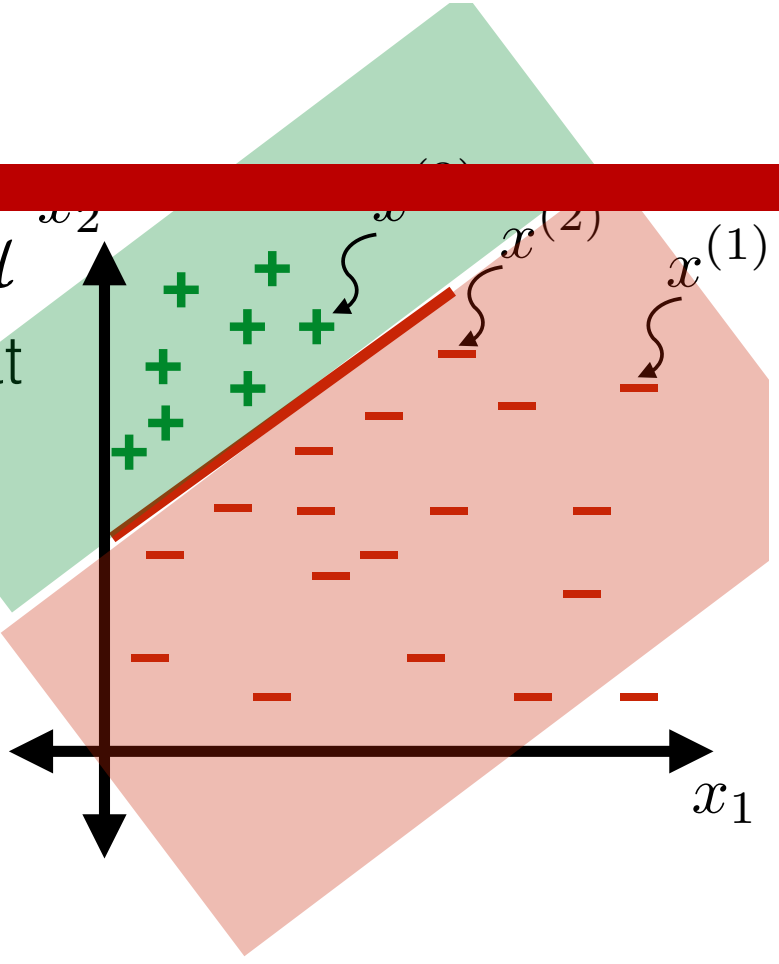


Math facts!

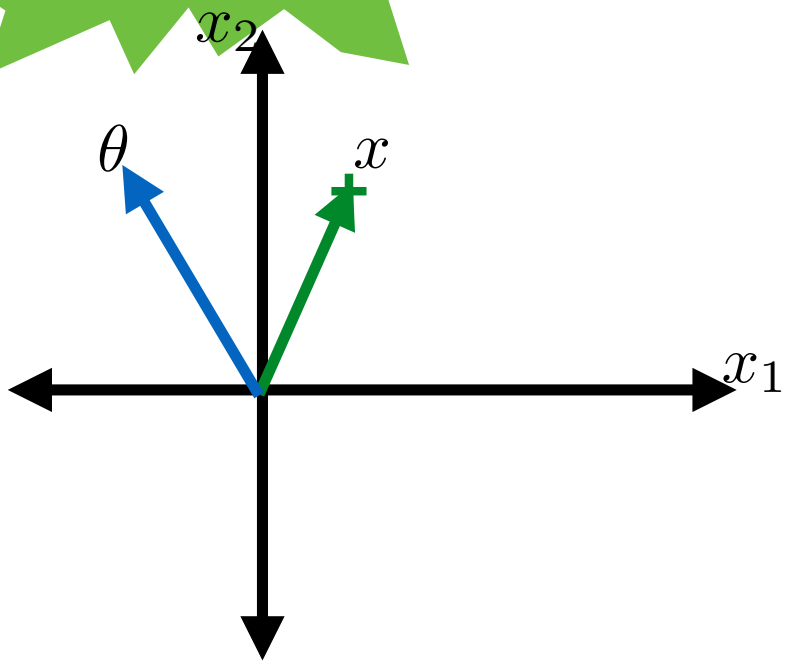


Linear classifiers

- 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

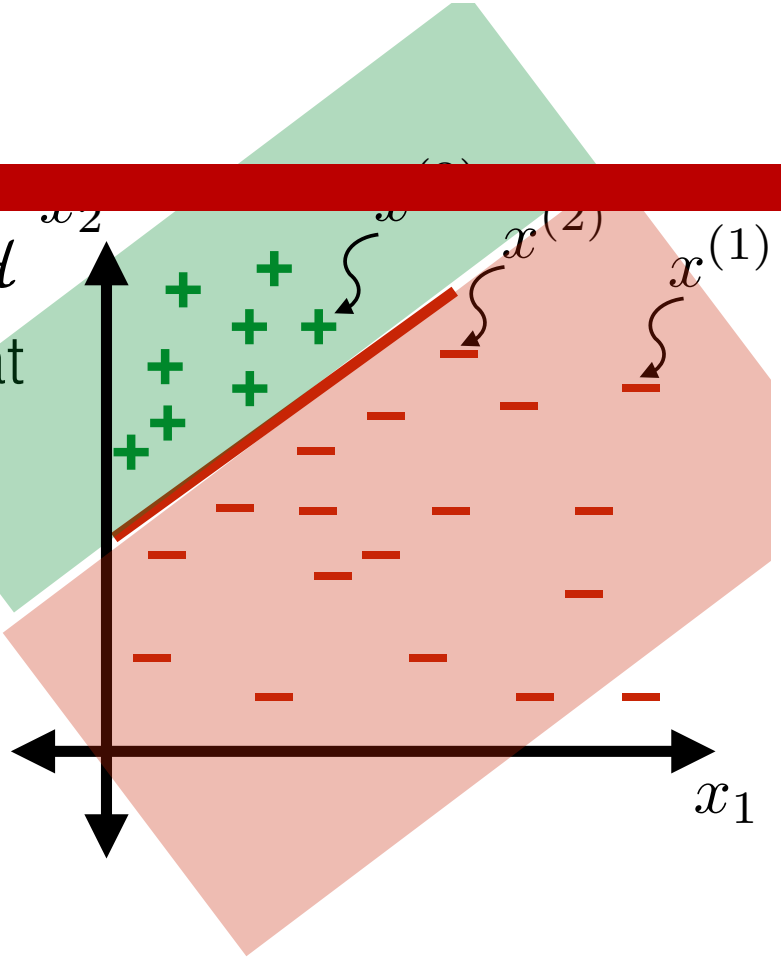


Math facts!

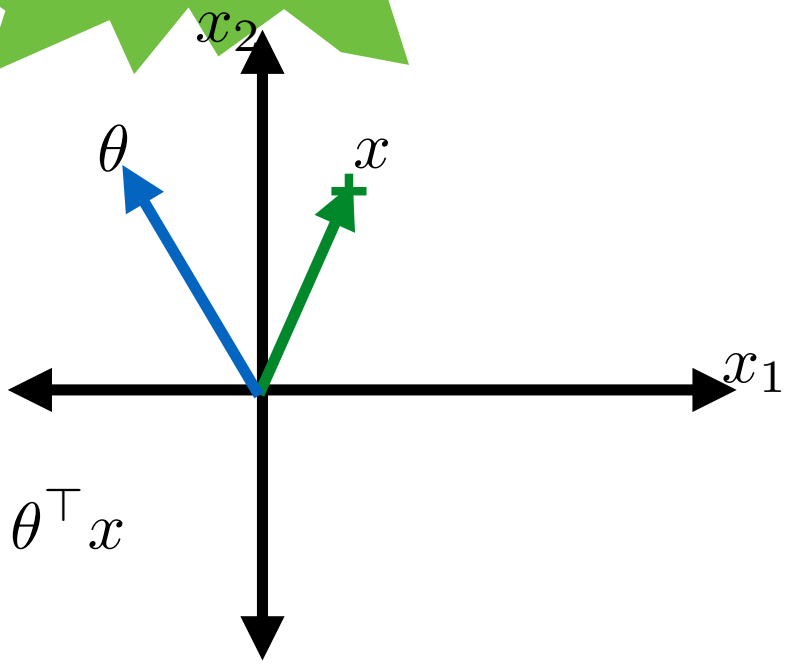


Linear classifiers

- 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

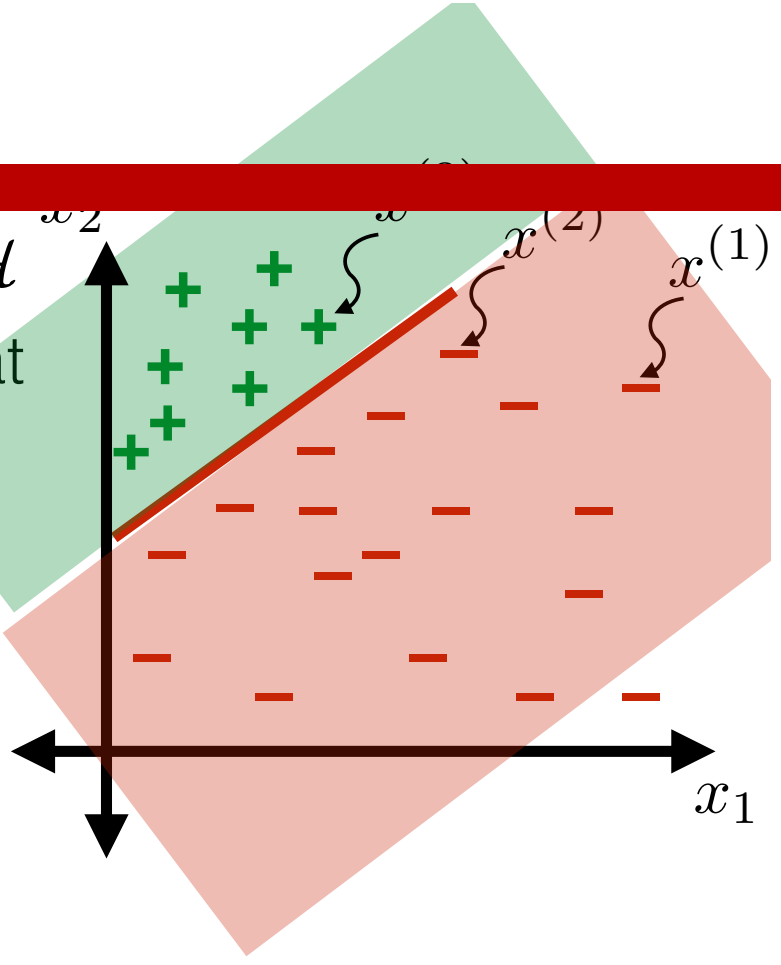


Math facts!

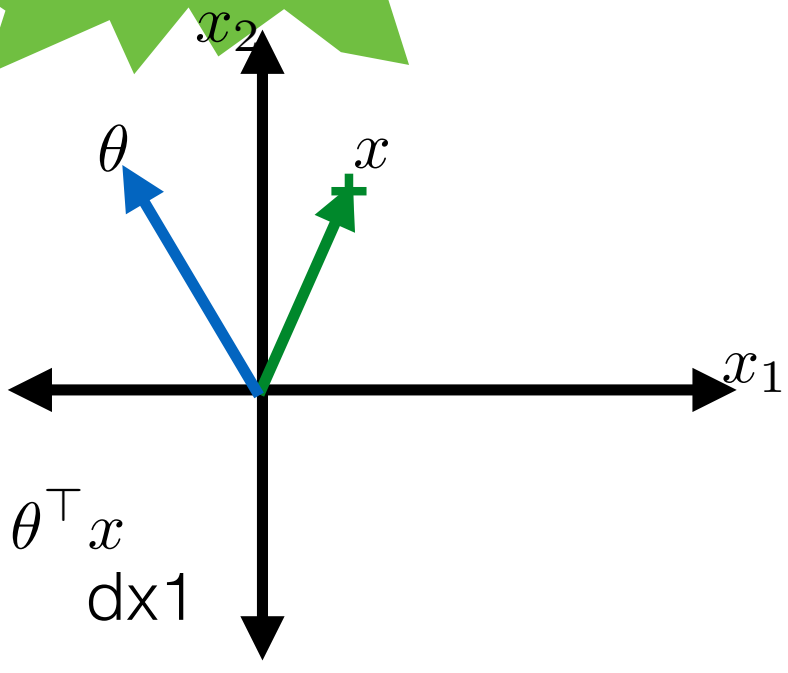


Linear classifiers

- 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

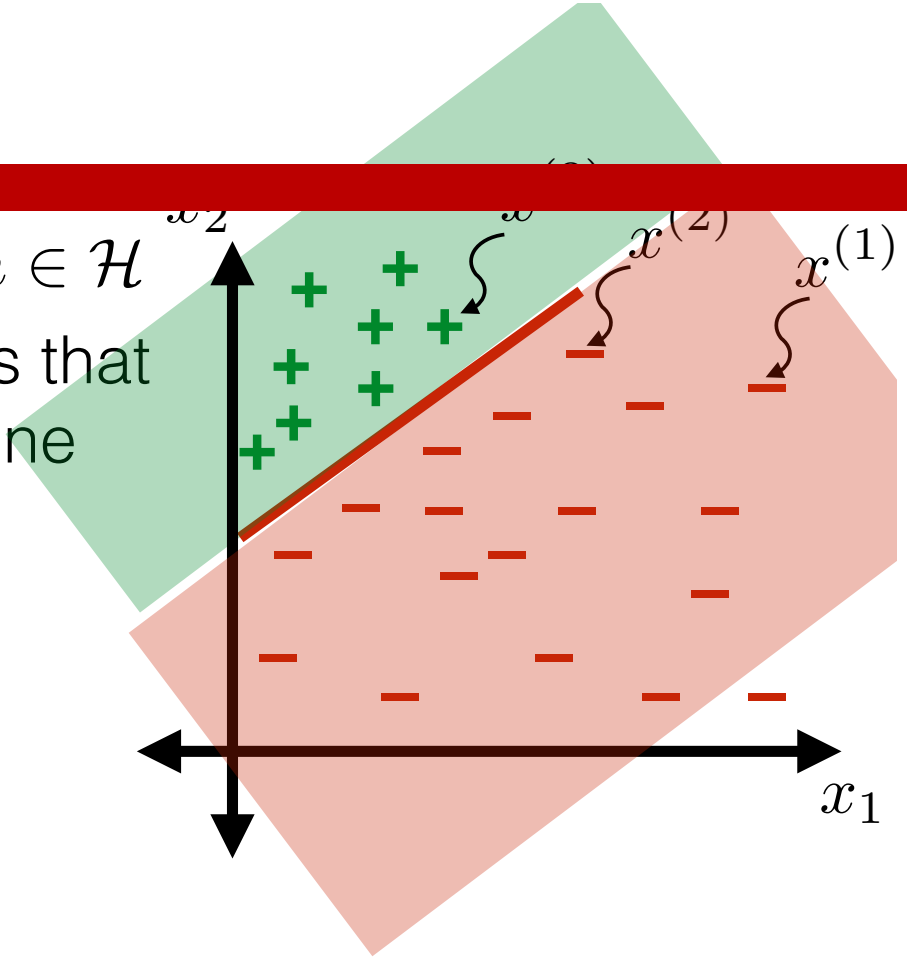


Math facts!

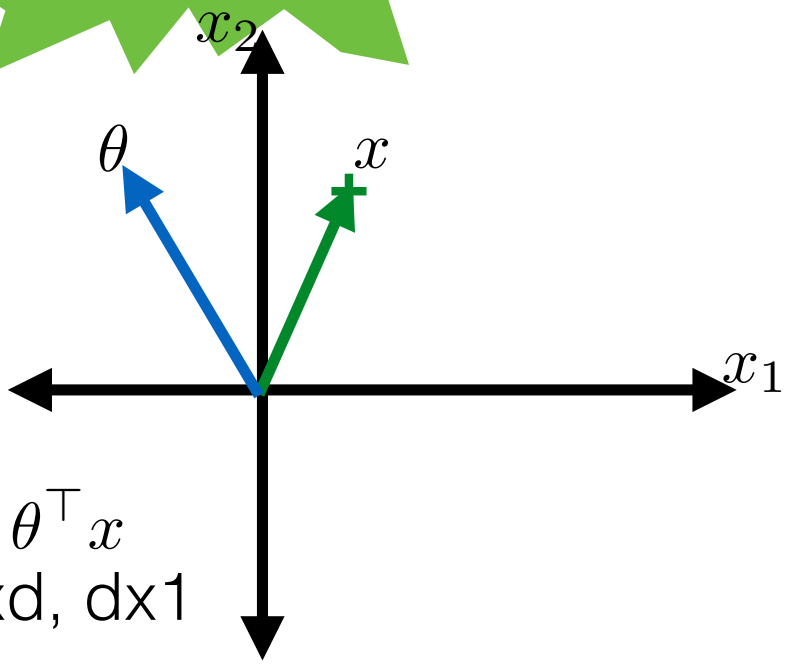


Linear classifiers

- 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!

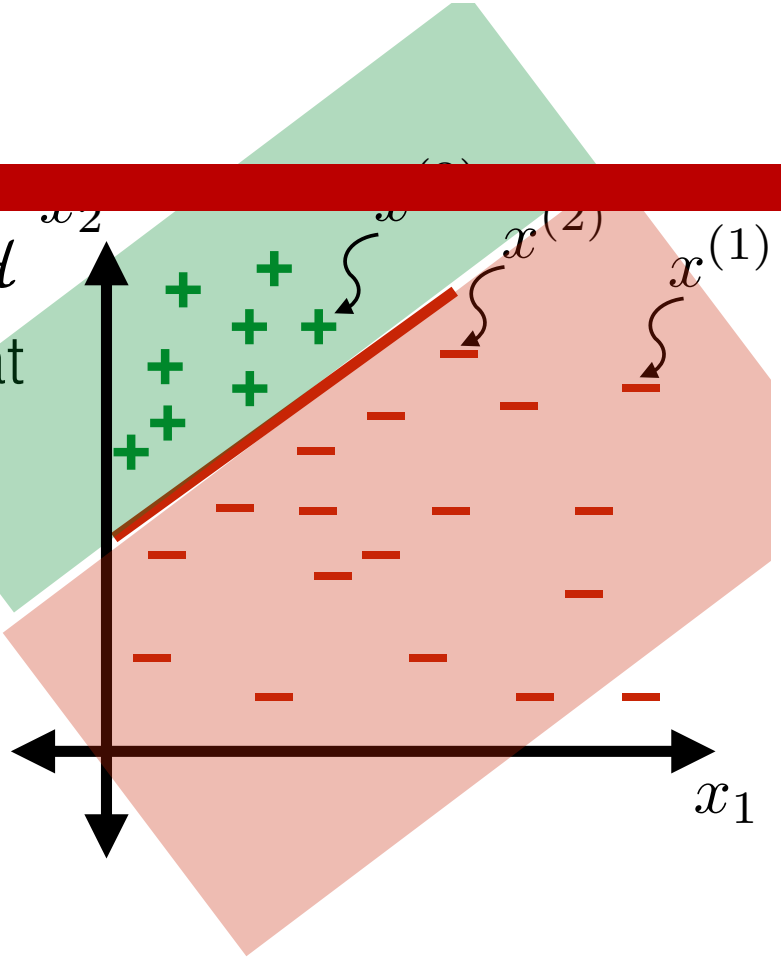


$\theta^\top x$

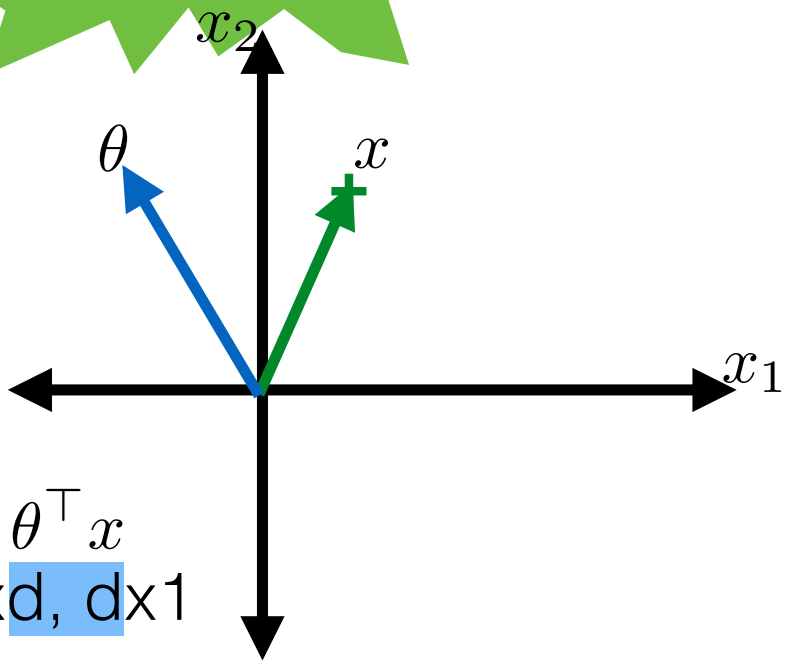
1xd, dx1

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!

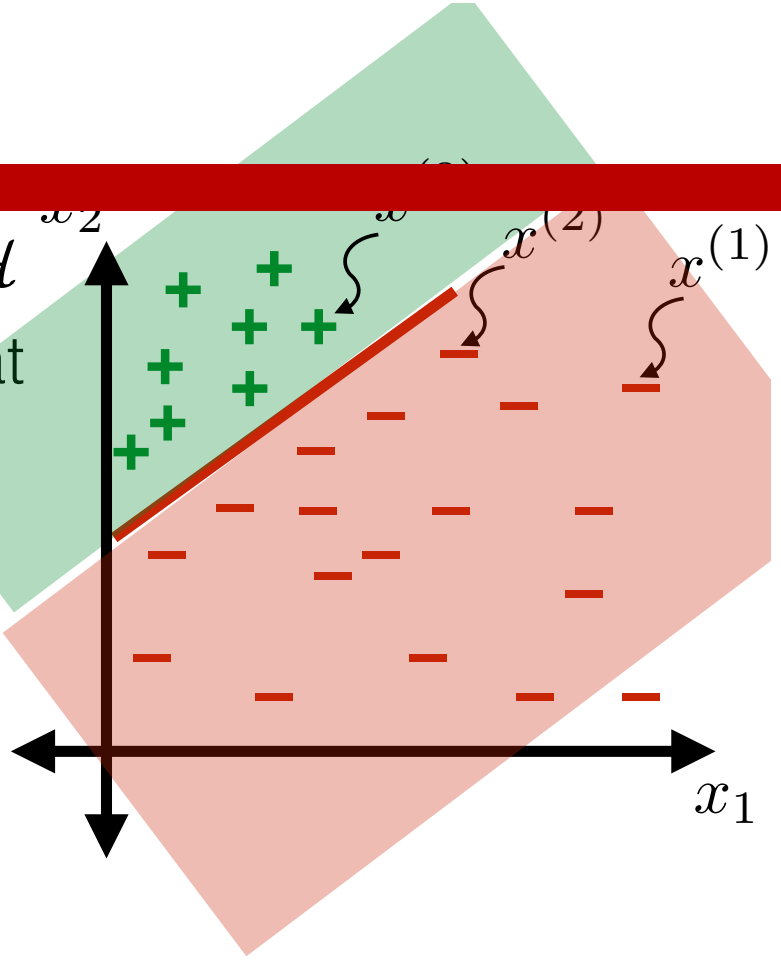


$\theta^\top x$

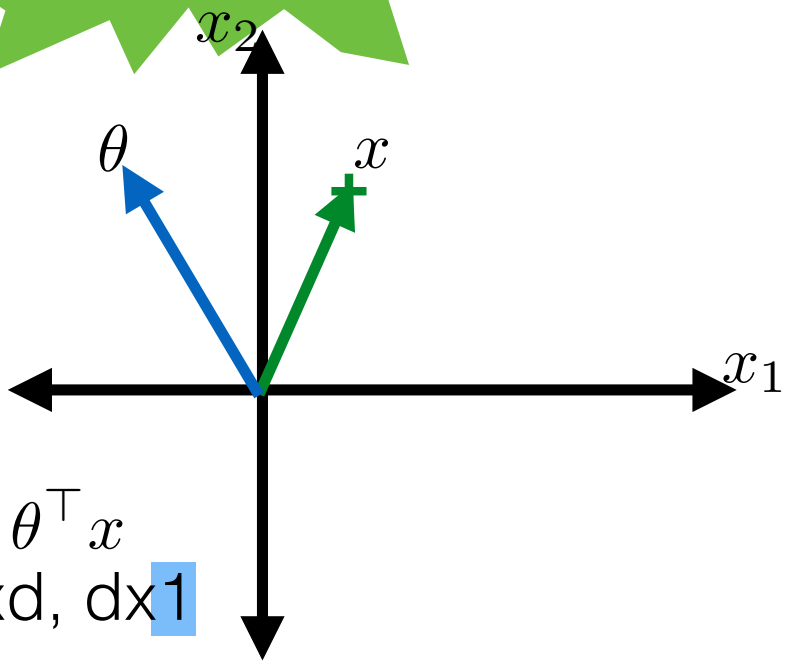
1xd, dx1

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!

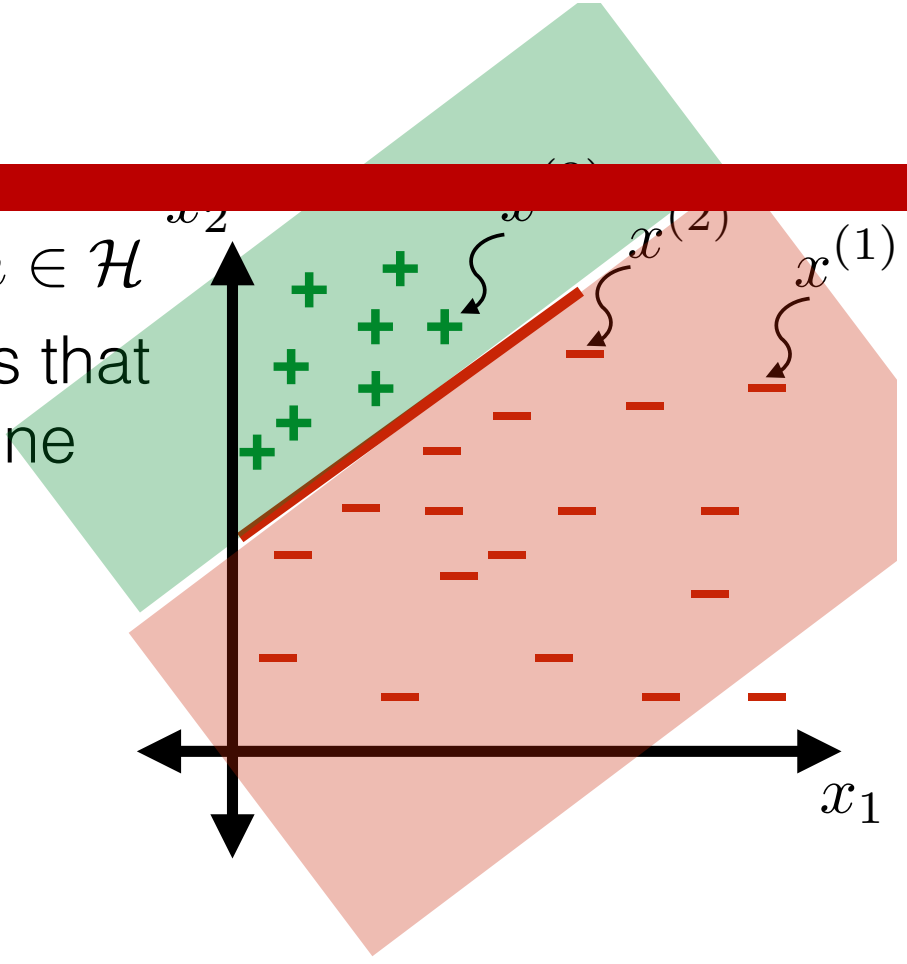


$\theta^\top x$

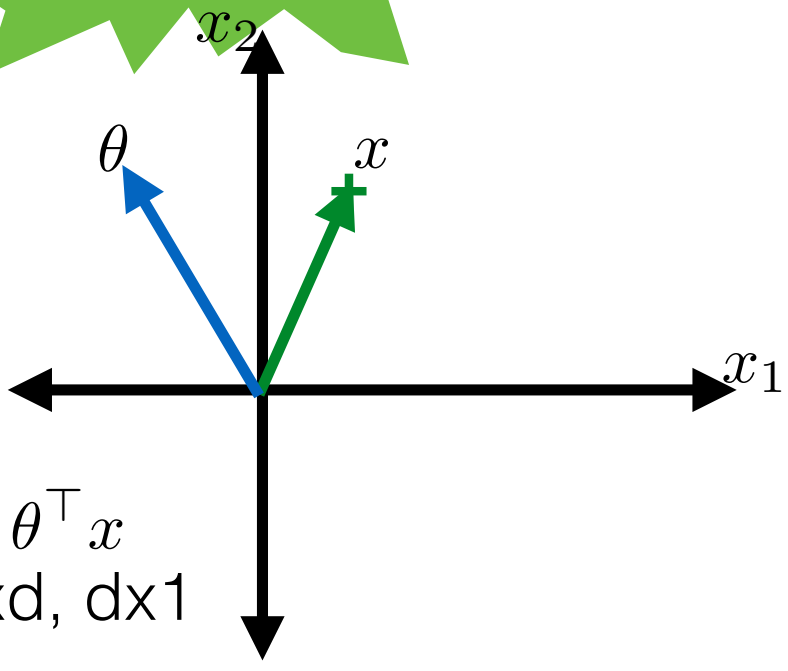
1xd, dx1

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!

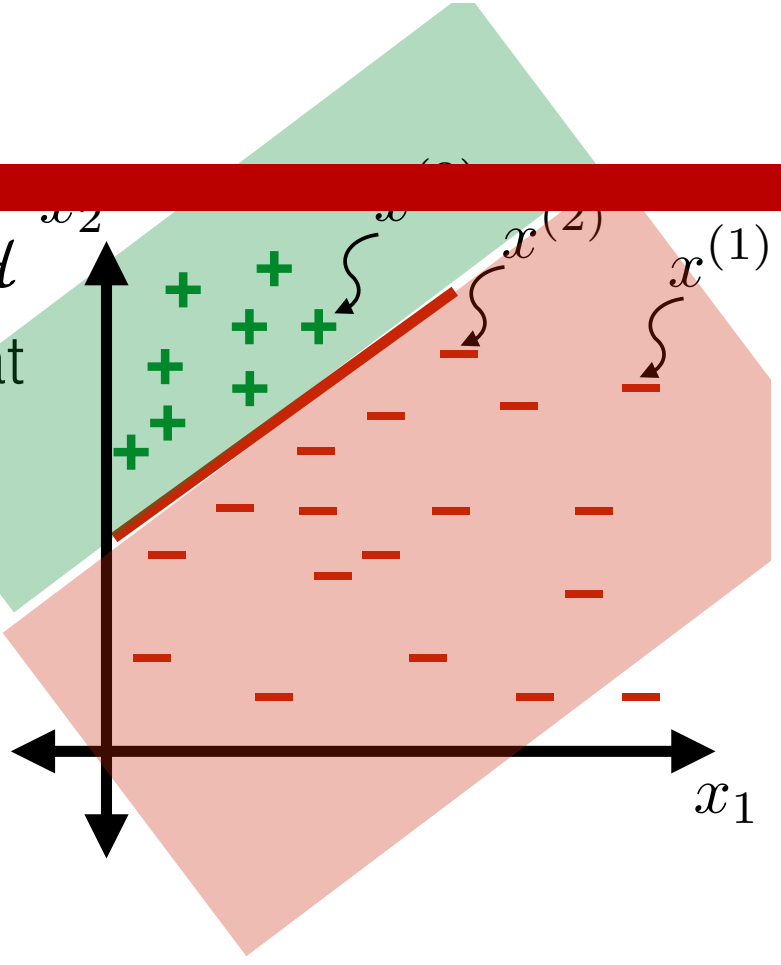


$\theta^\top x$

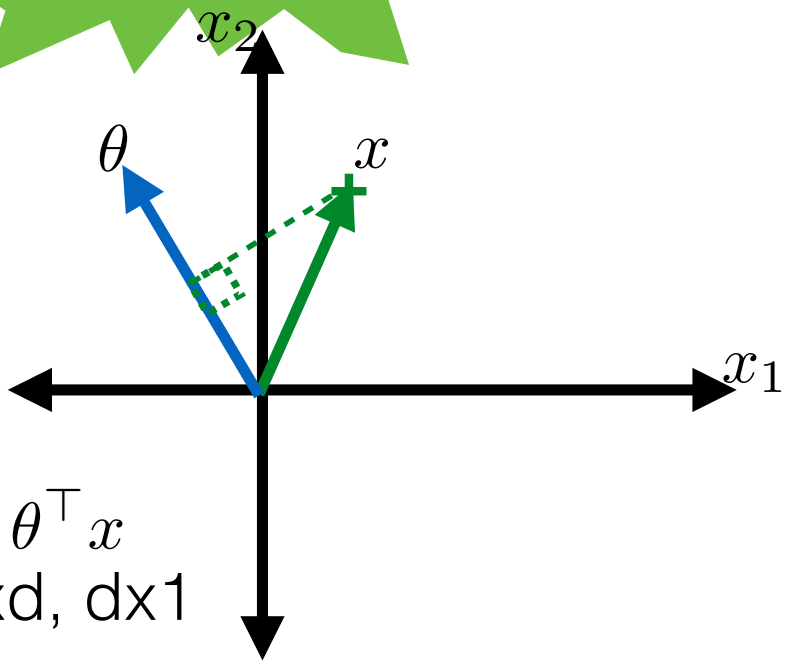
1xd, dx1

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!

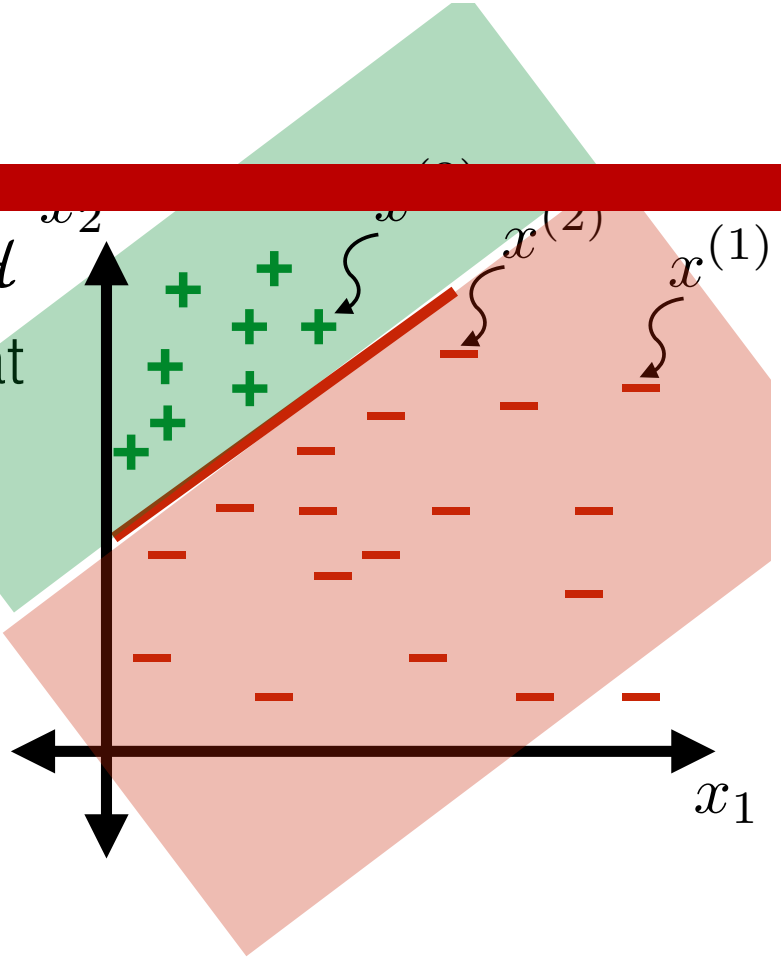


$\theta^\top x$

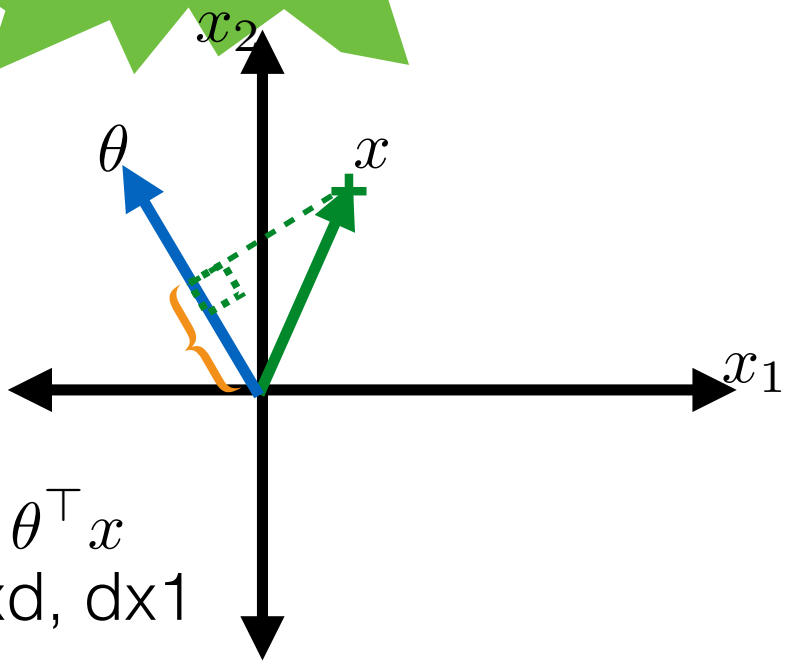
1xd, dx1

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!

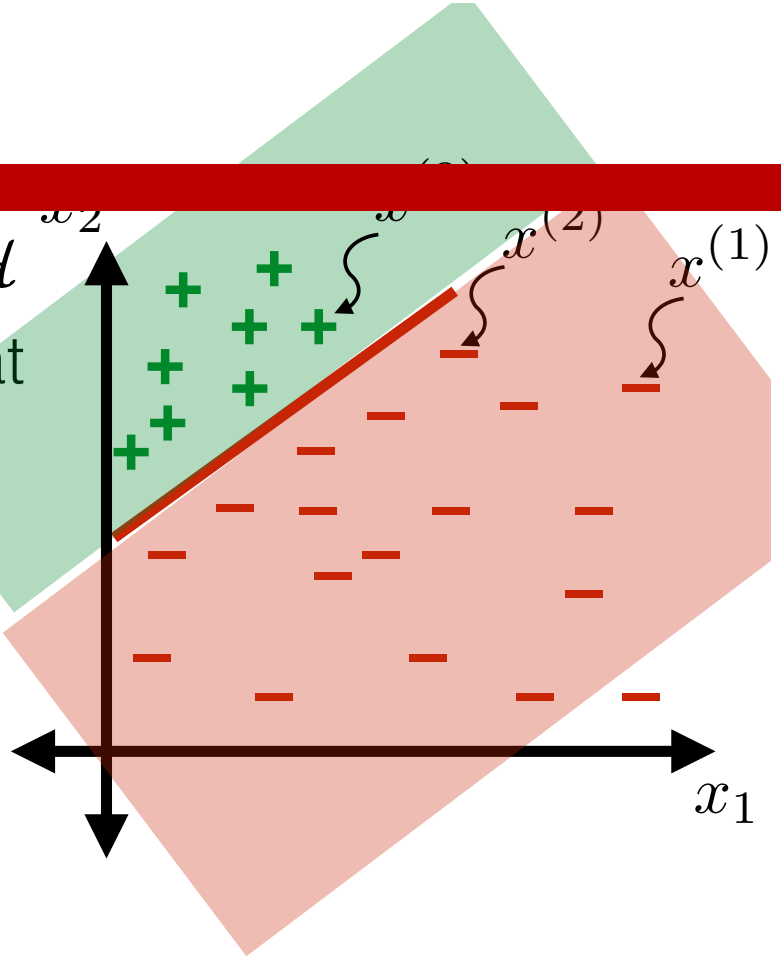


$\theta^\top x$

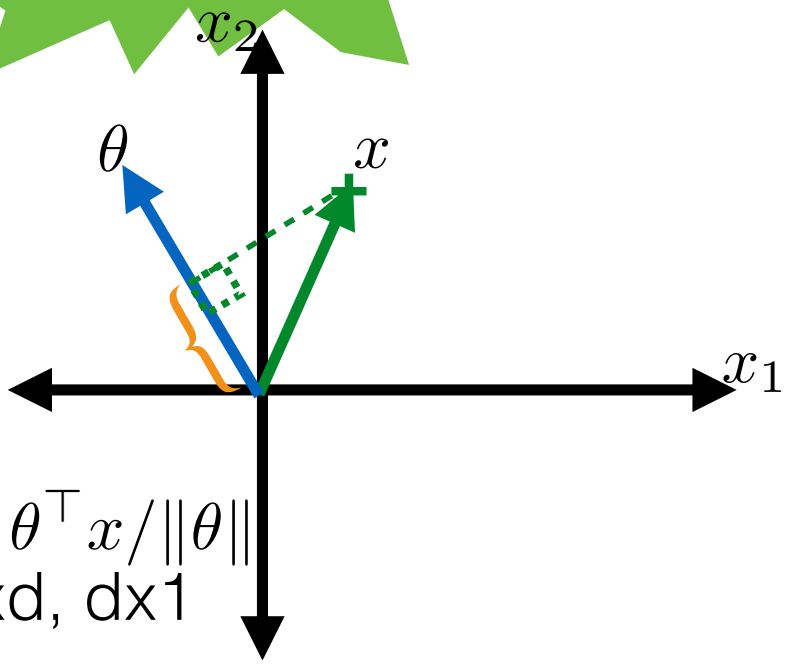
1xd, dx1

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

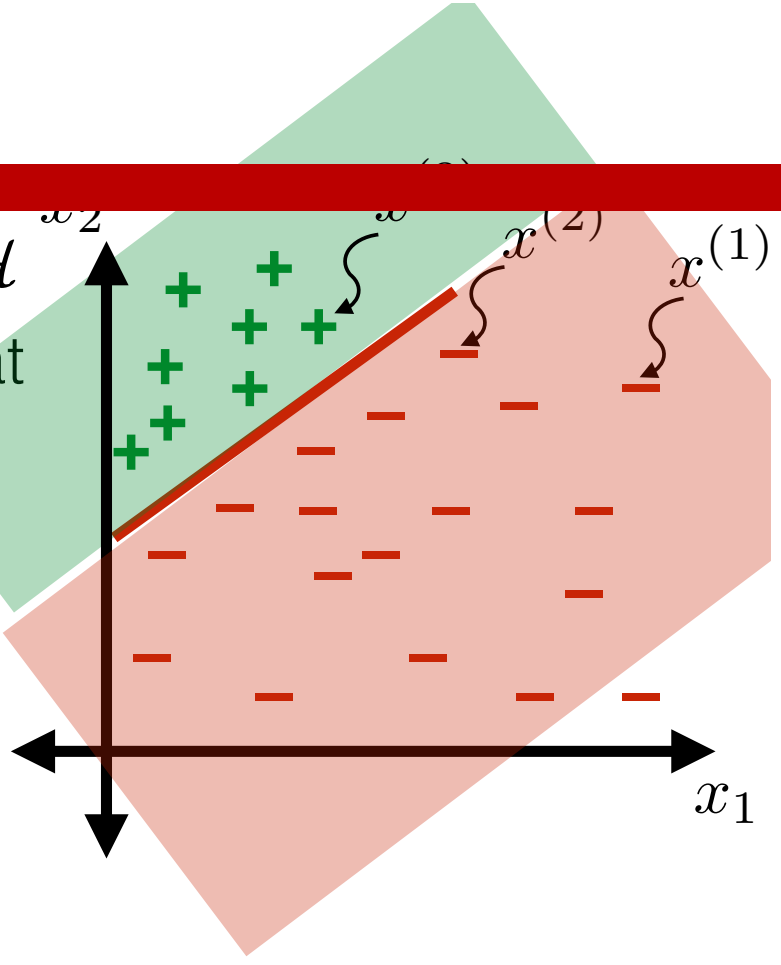


Math facts!

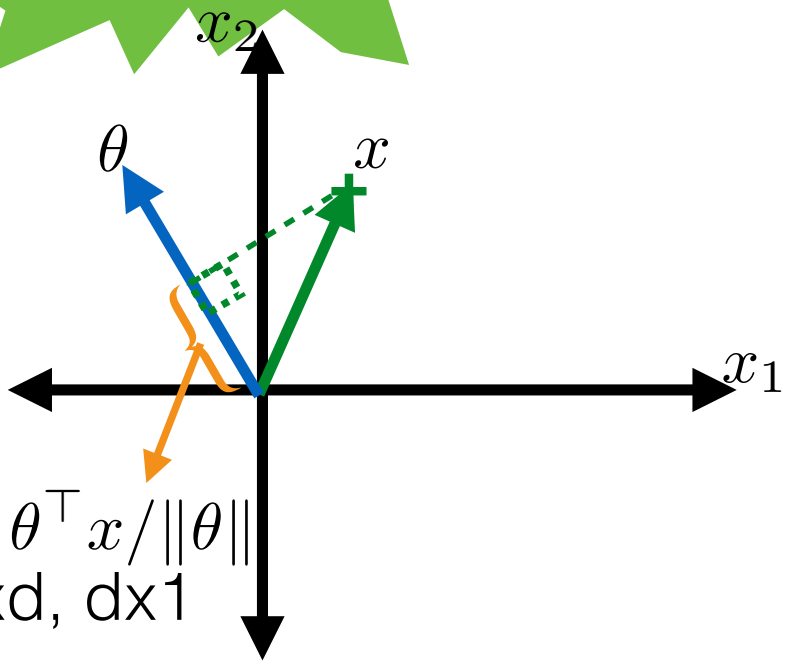


Linear classifiers

- 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

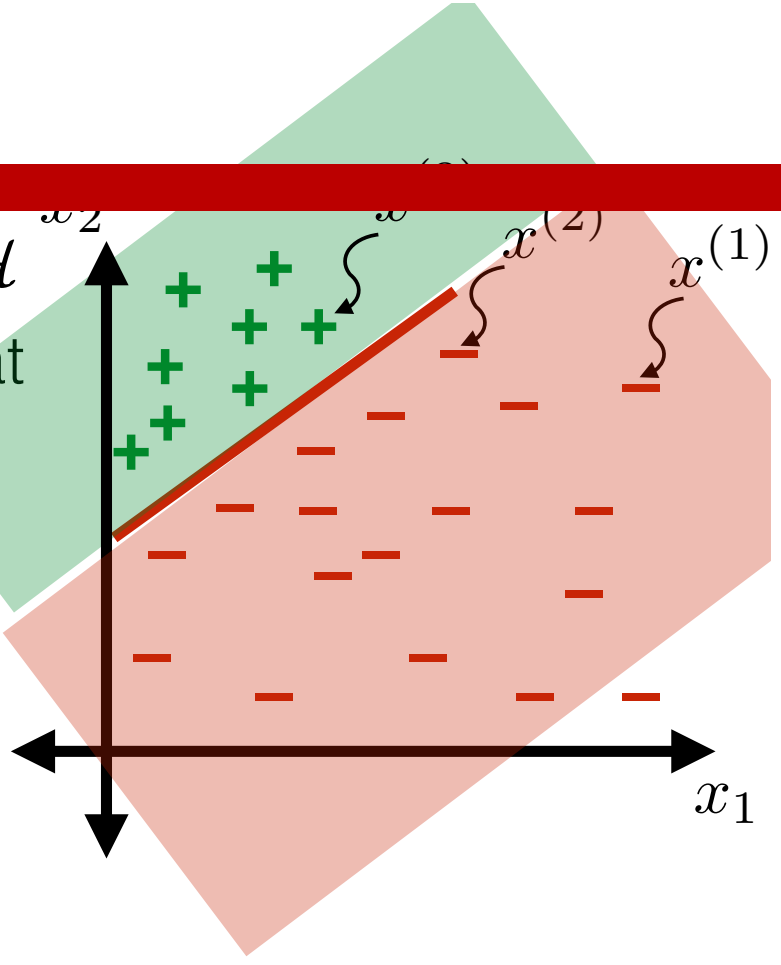


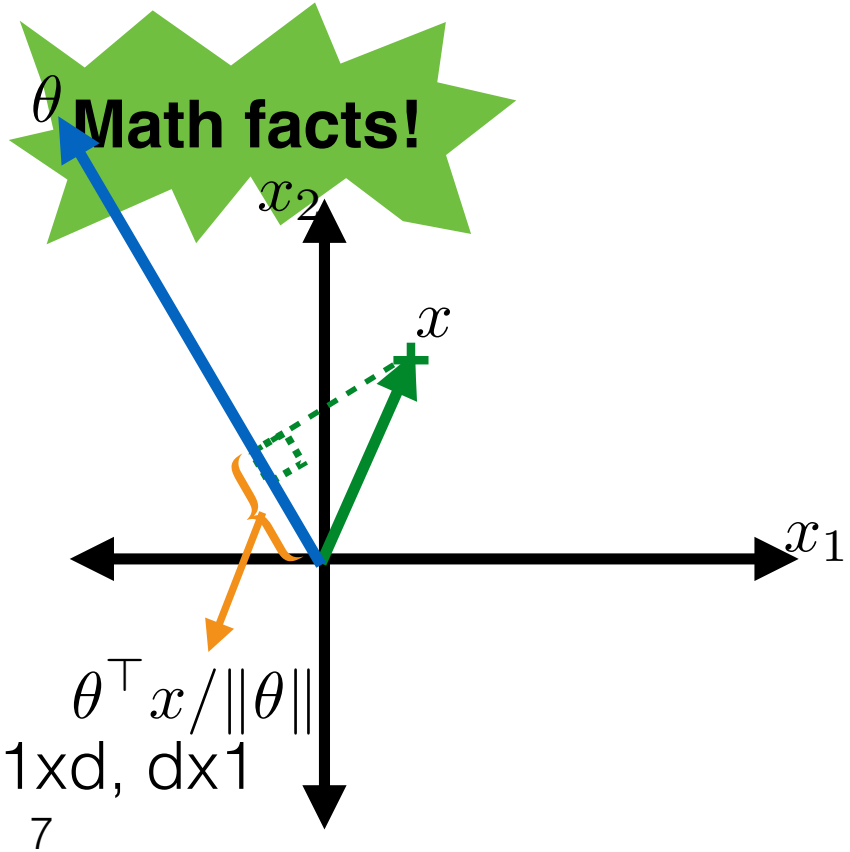
Math facts!



Linear classifiers

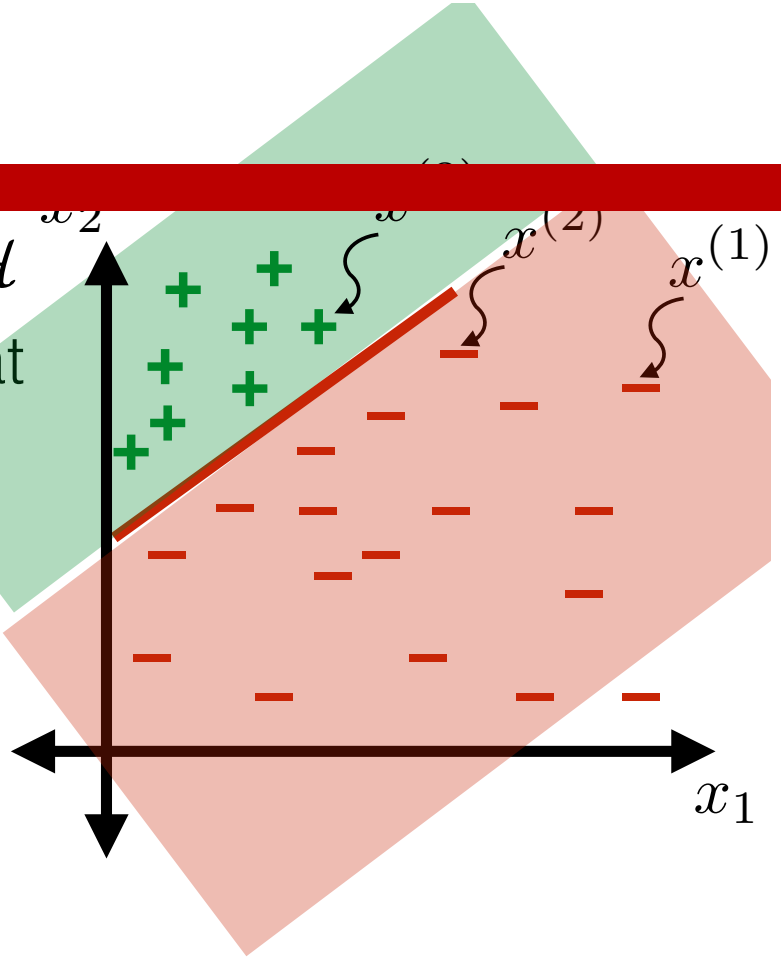
- Hypothesis class $\mathcal{H}$: set of $h \in \mathcal{H}$
  - Example $\mathcal{H}$: All hypotheses that label +1 on one side of a line and -1 on the other side



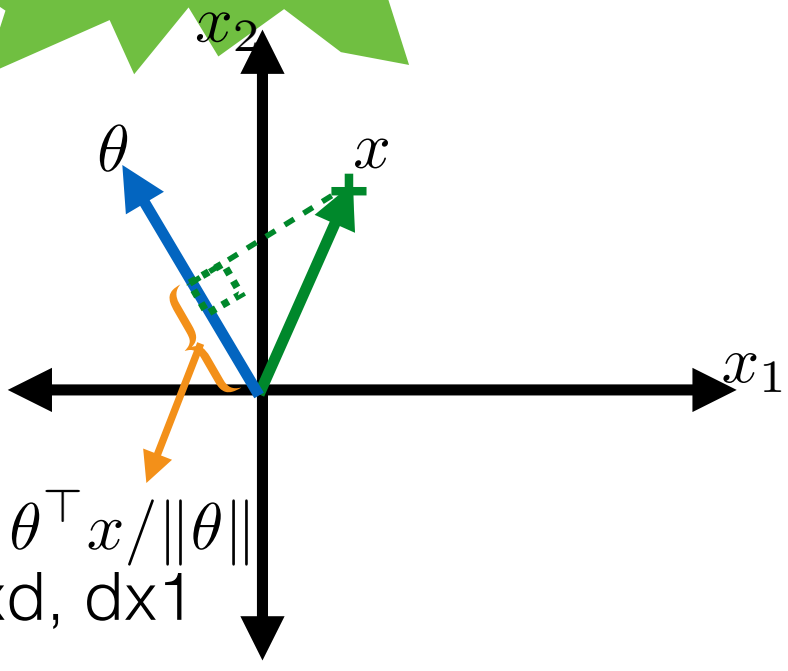


Linear classifiers

- 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

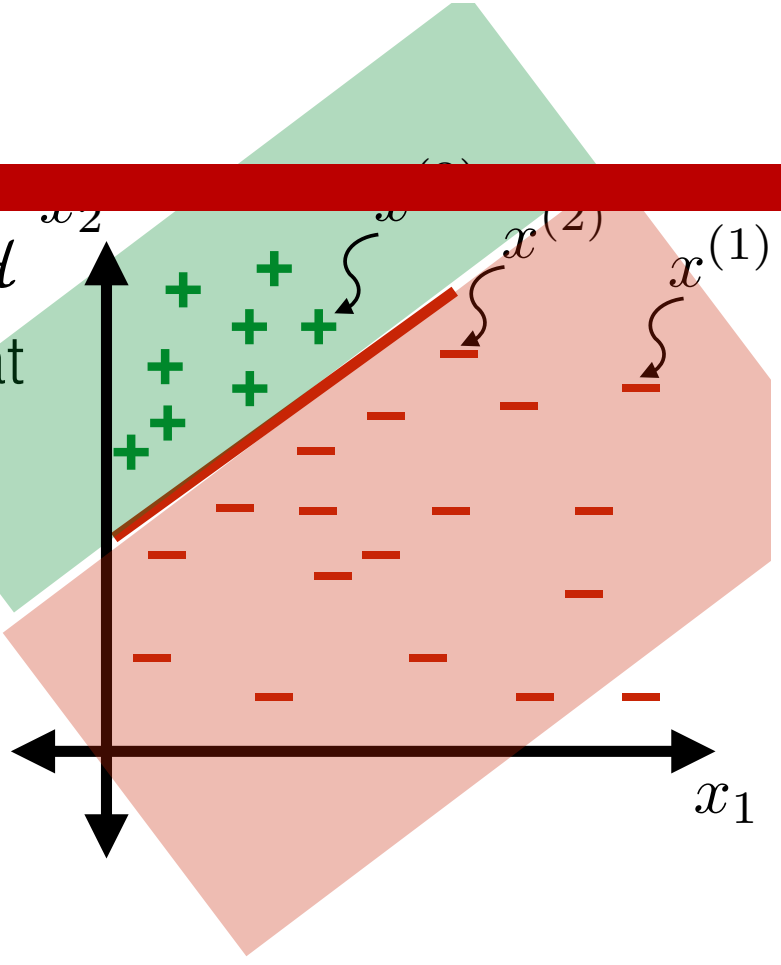


Math facts!

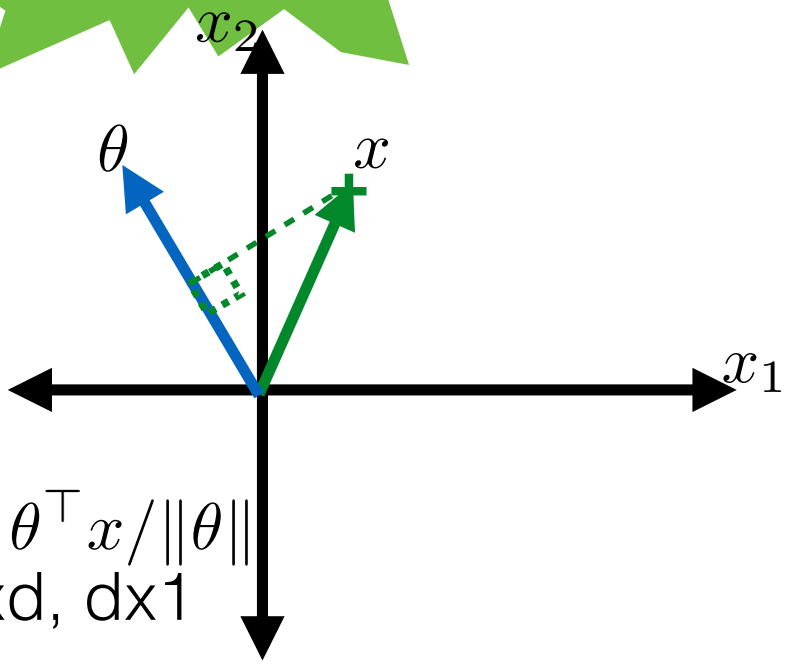


Linear classifiers

- 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

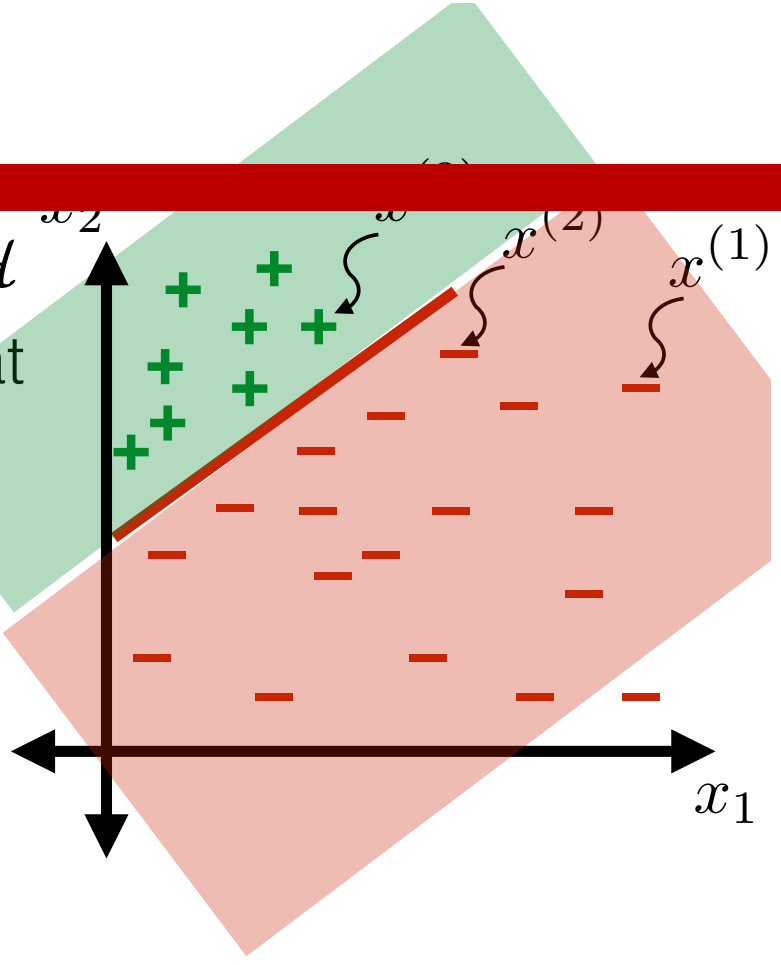


Math facts!

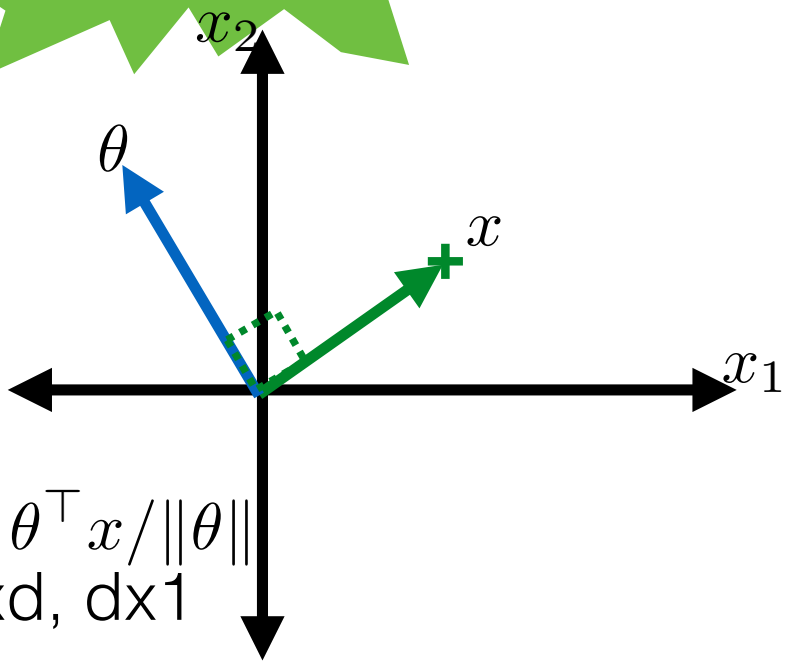


Linear classifiers

- 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

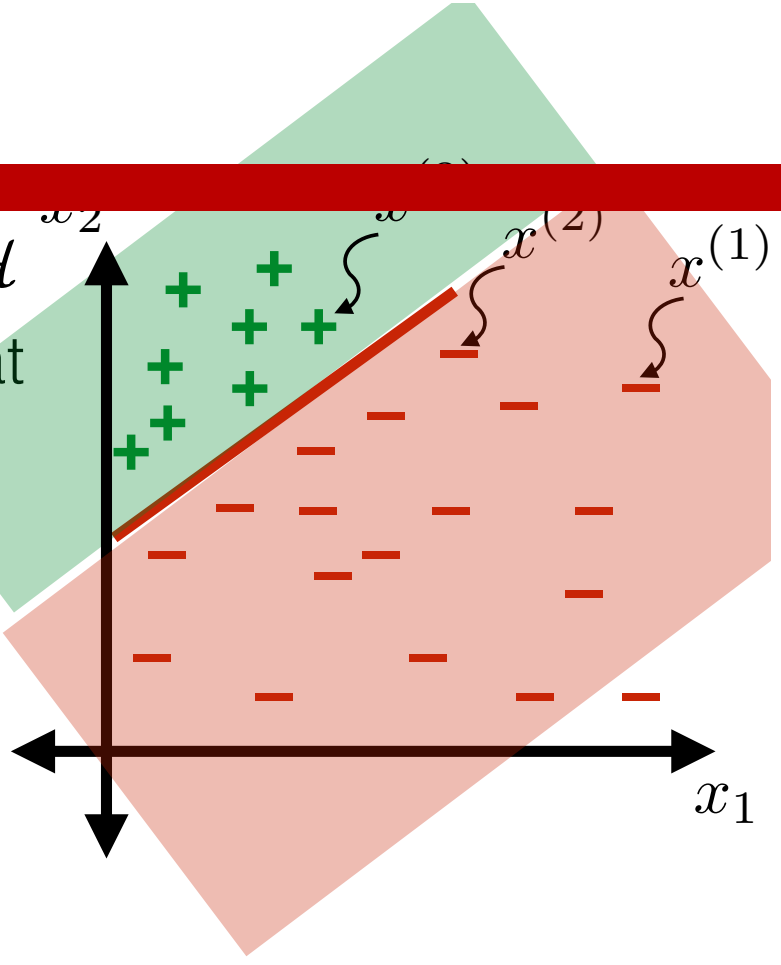


Math facts!

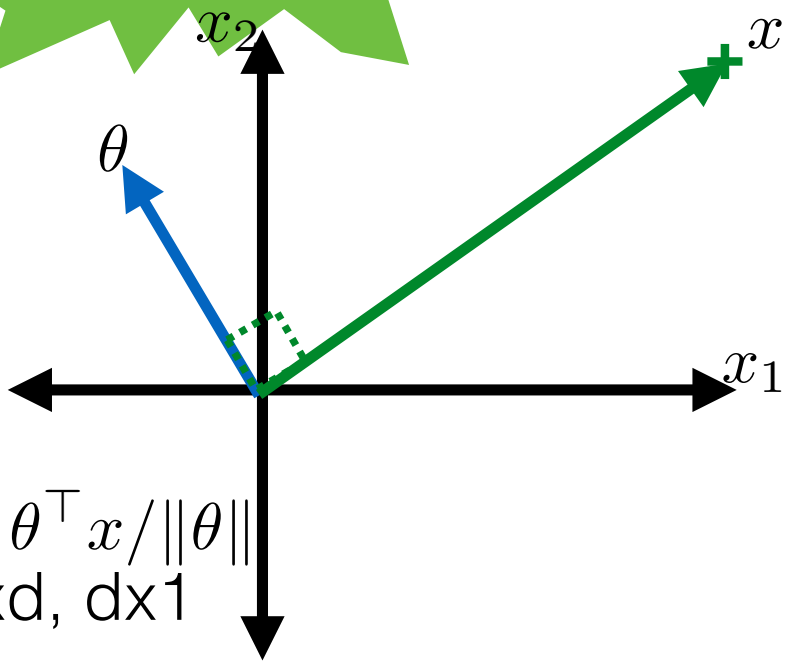


Linear classifiers

- 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

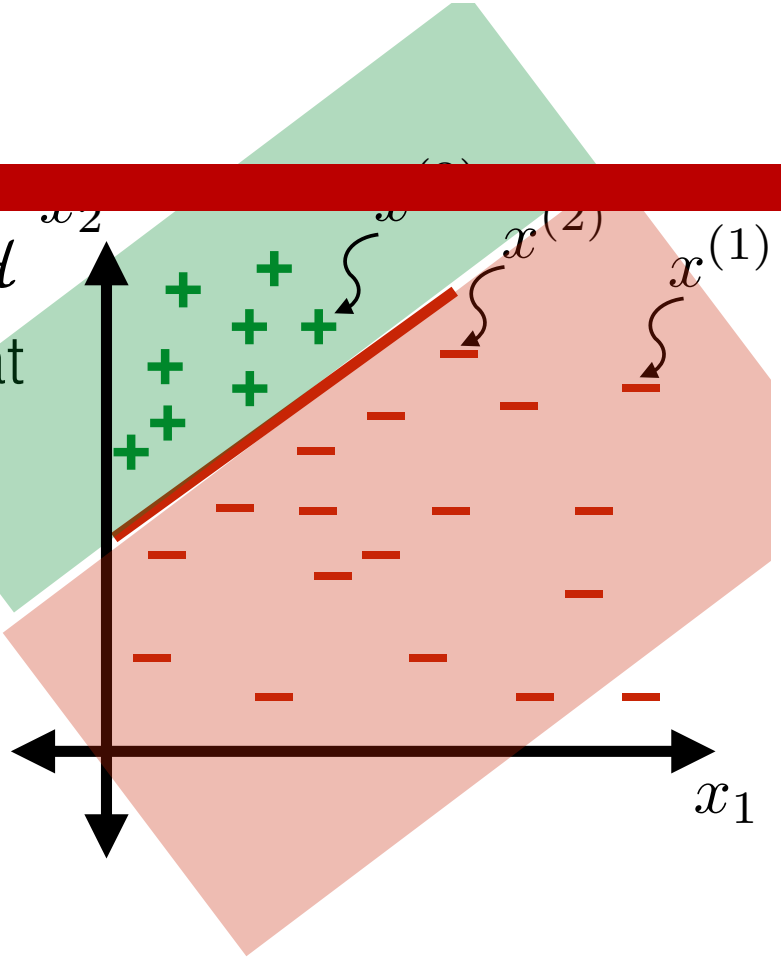


Math facts!

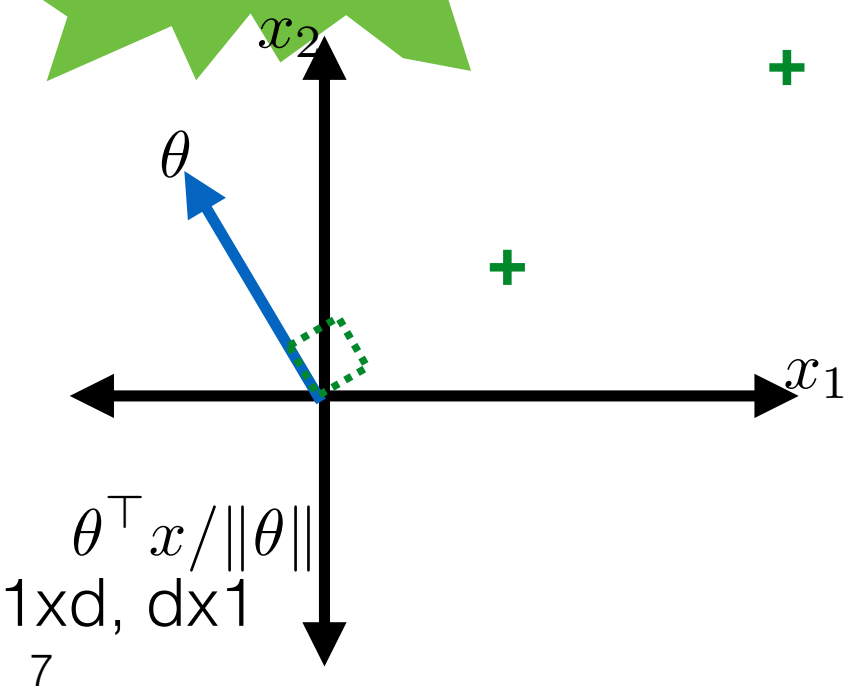


Linear classifiers

- 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

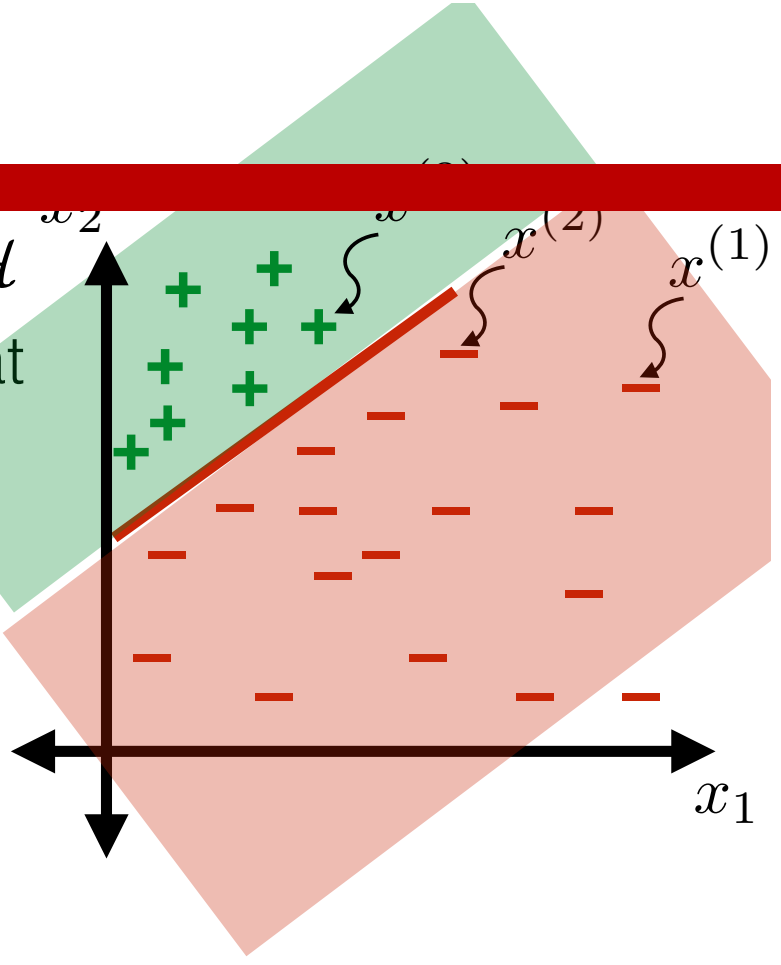


Math facts!

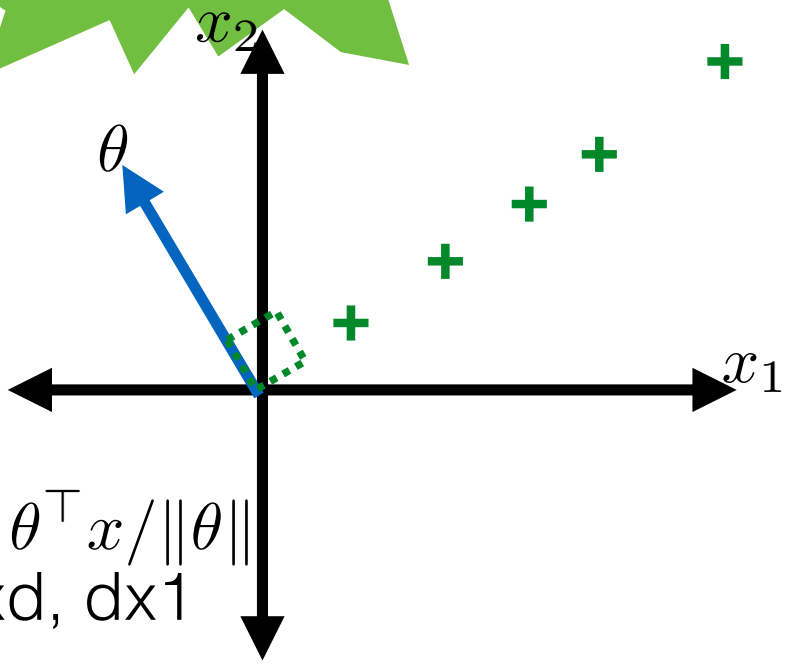


Linear classifiers

- 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

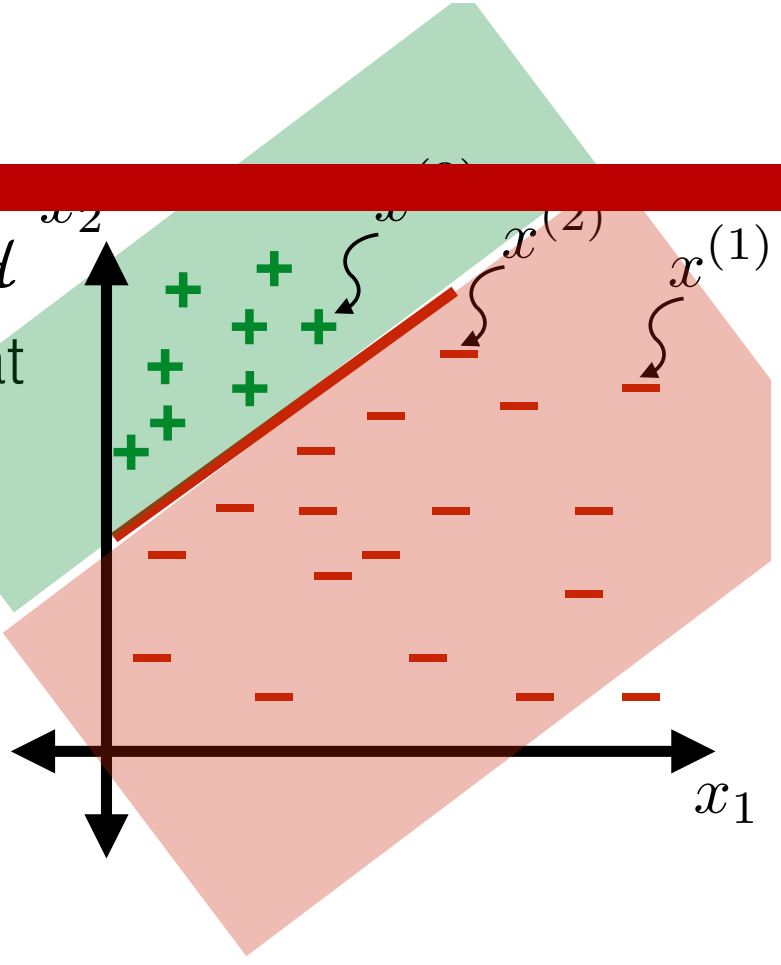


Math facts!

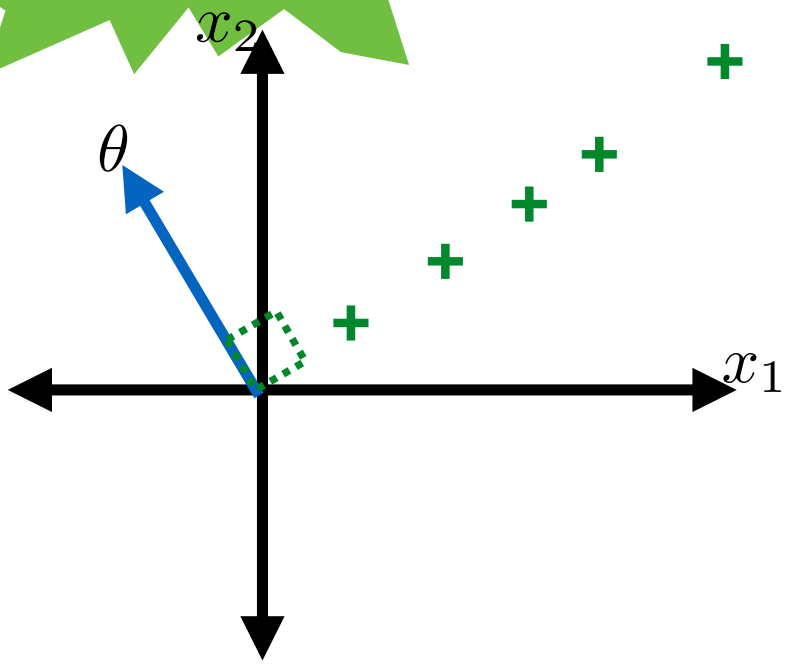


Linear classifiers

- 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

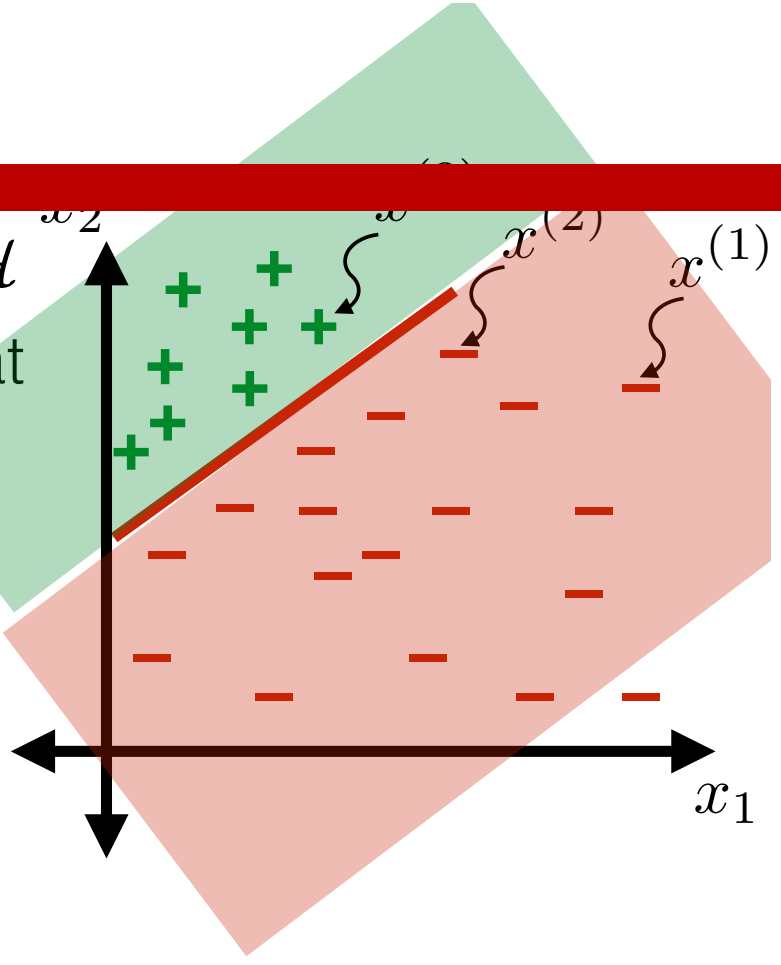


Math facts!

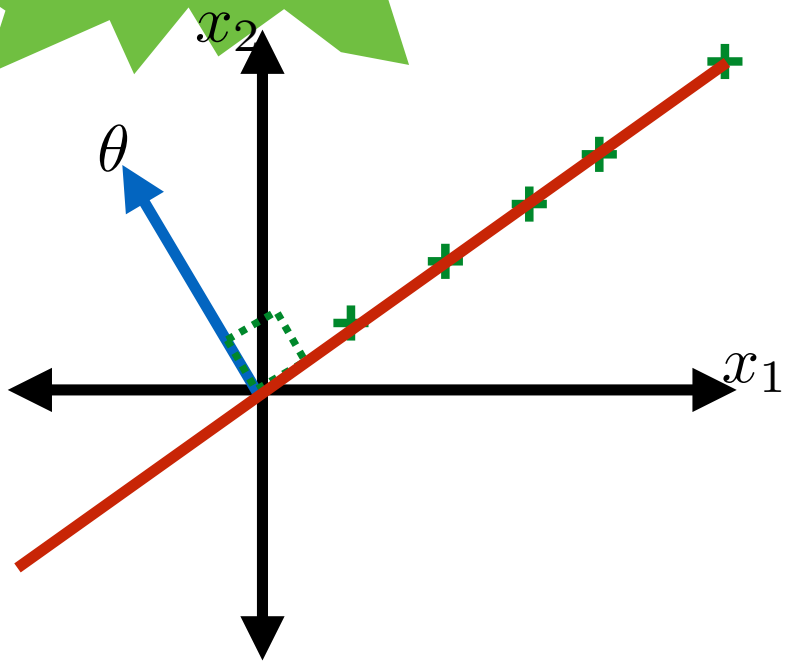


Linear classifiers

- 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

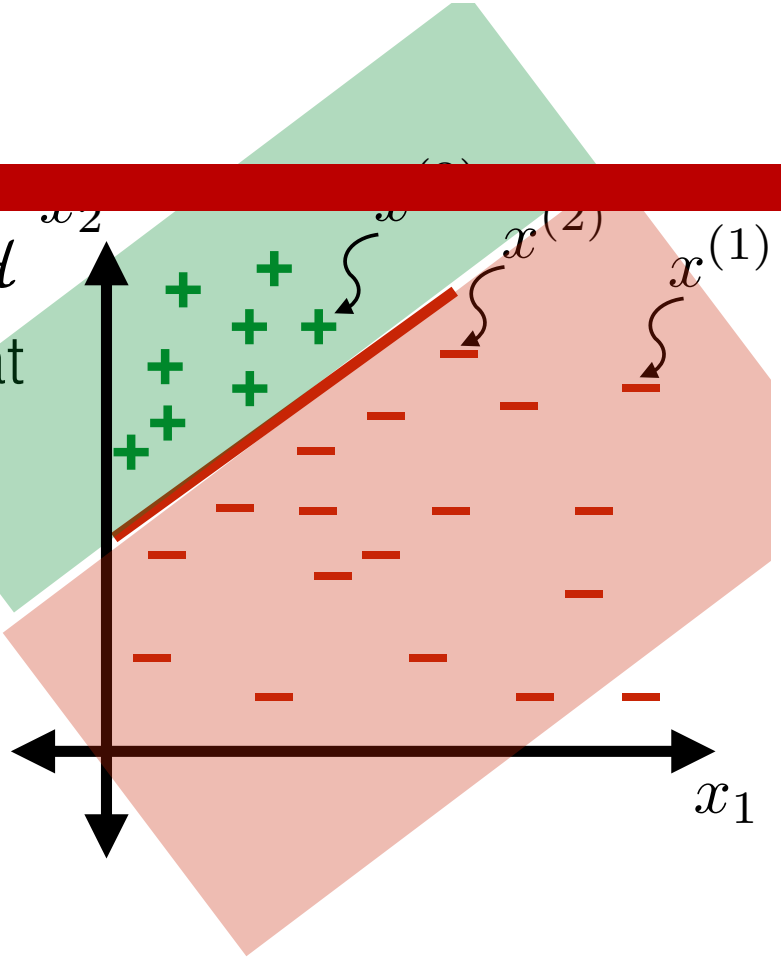


Math facts!

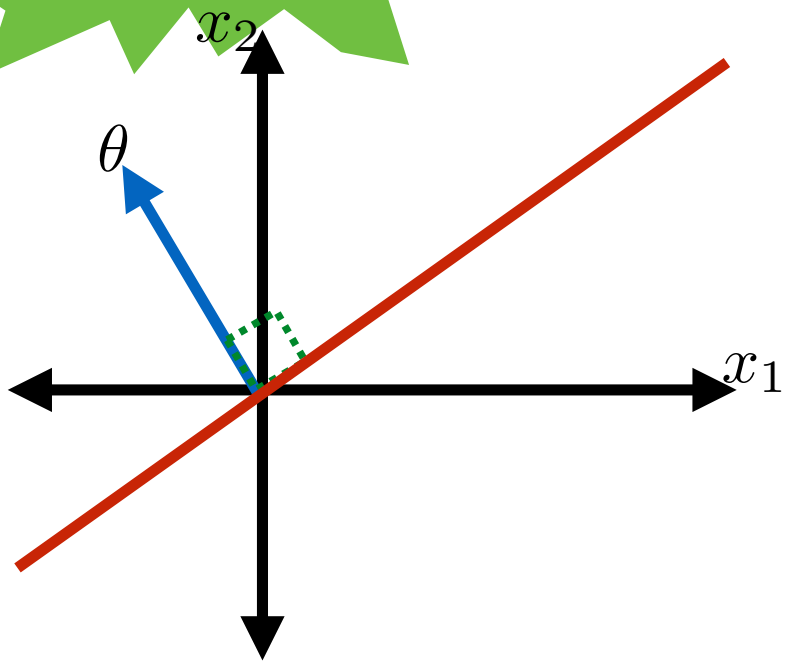


Linear classifiers

- 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

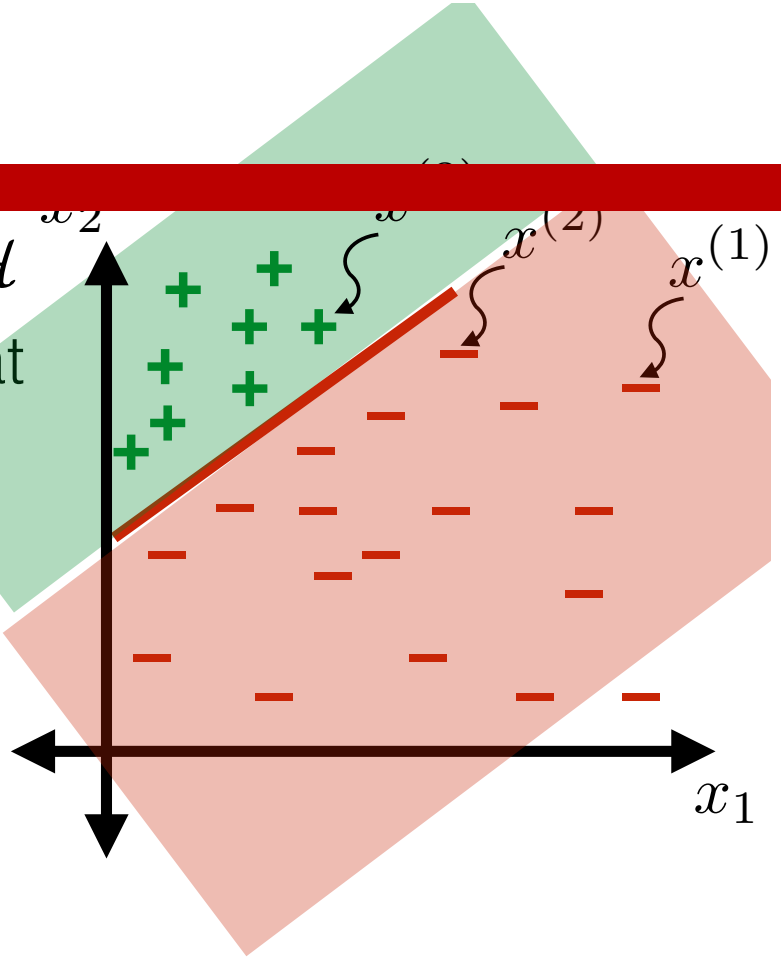


Math facts!

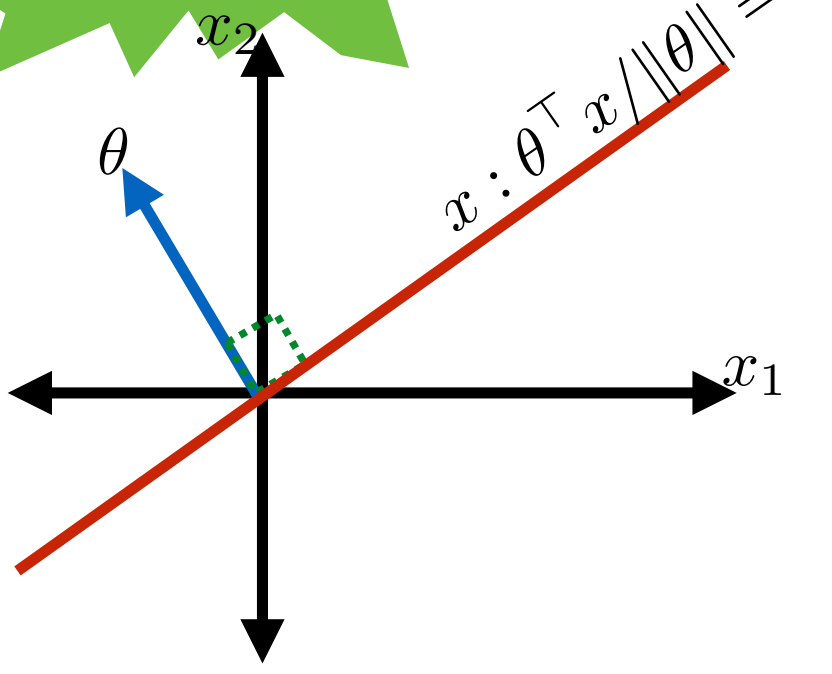


Linear classifiers

- 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

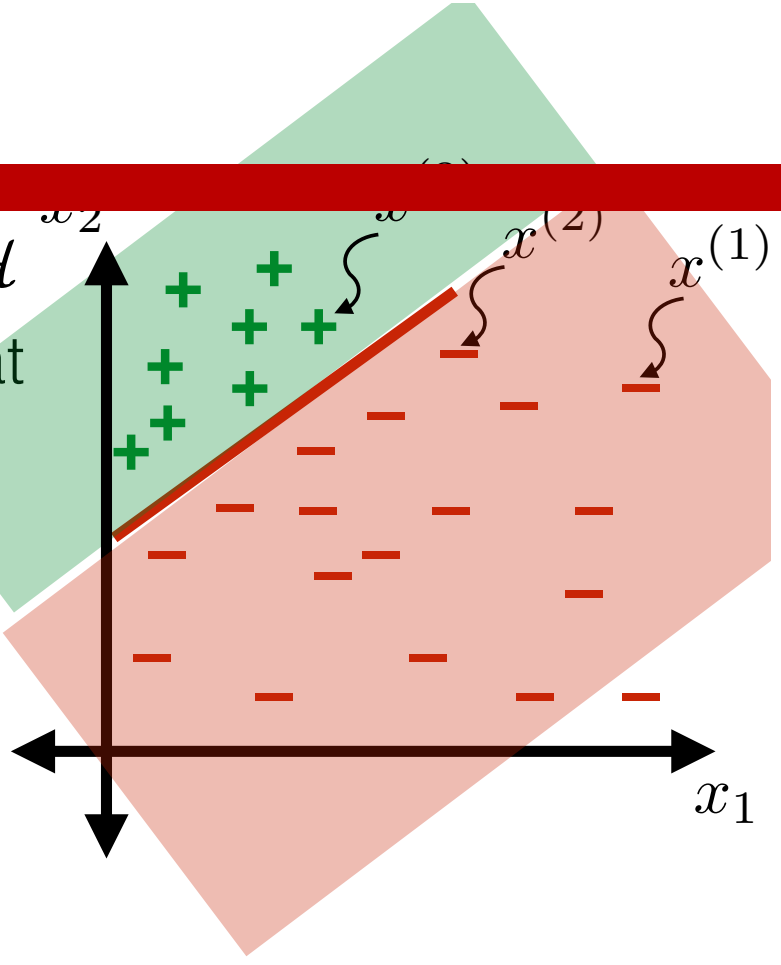


Math facts!

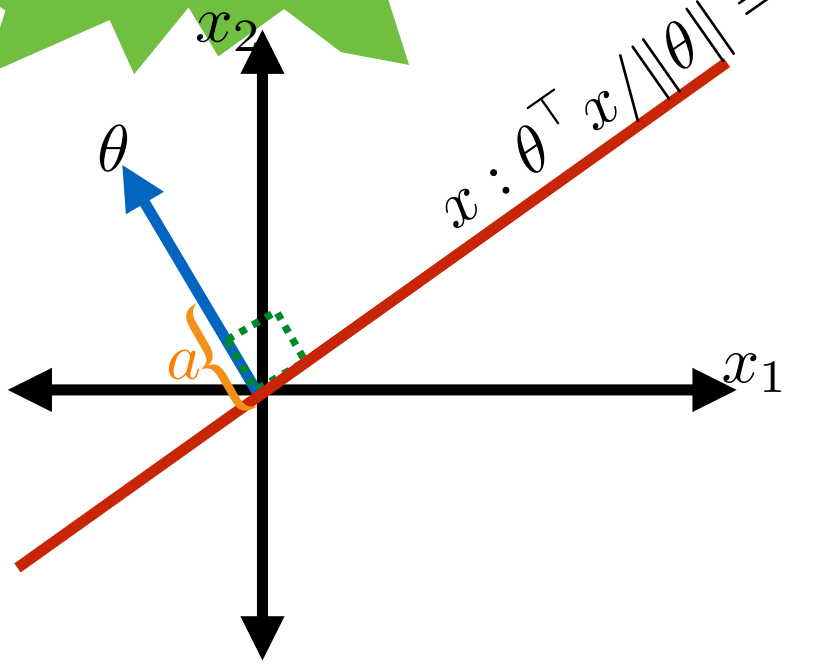


Linear classifiers

- 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

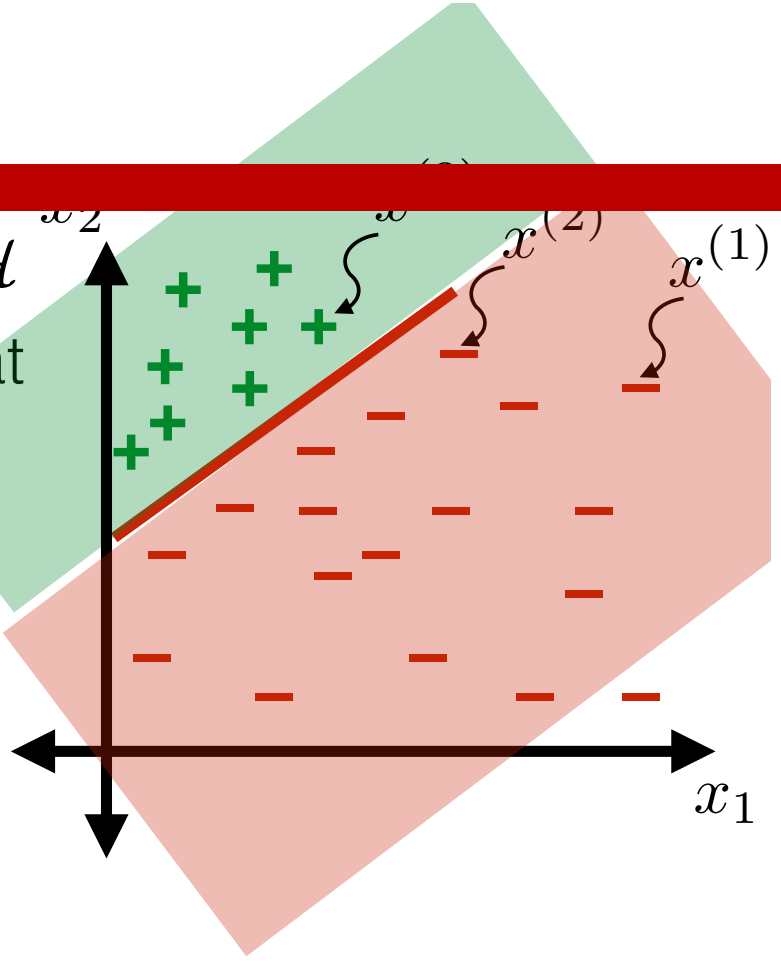


Math facts!

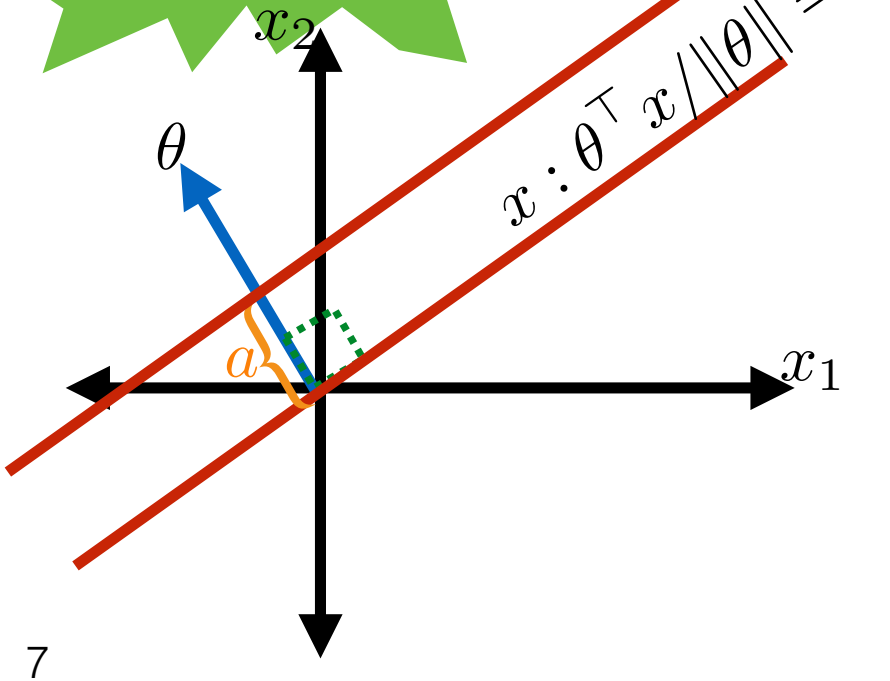


Linear classifiers

- 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

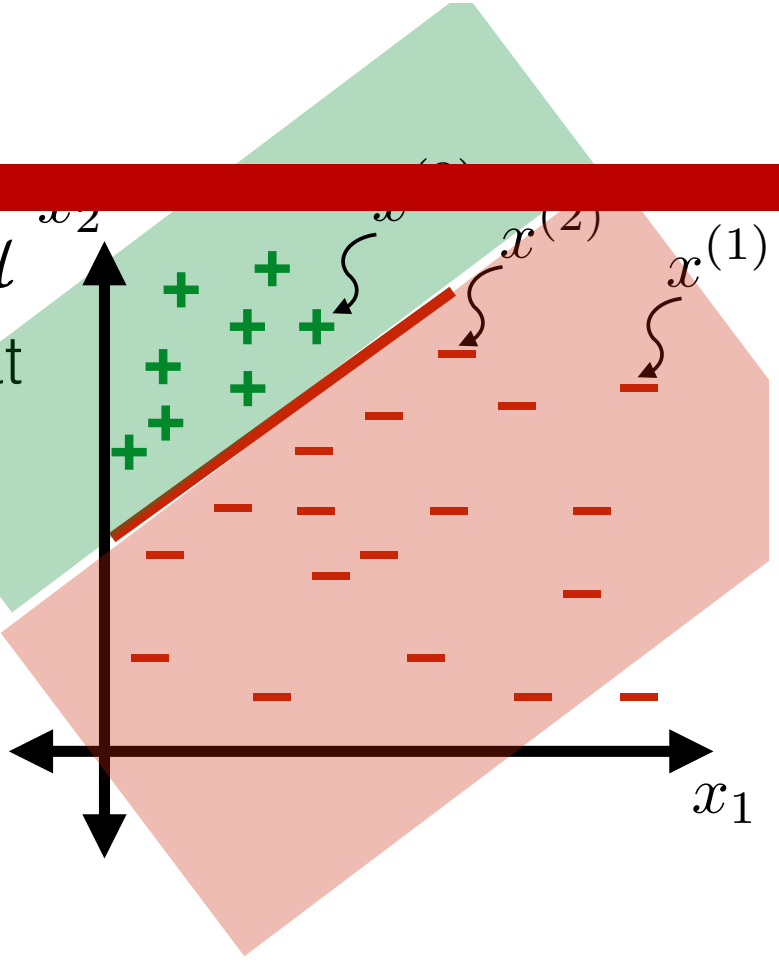


Math facts!

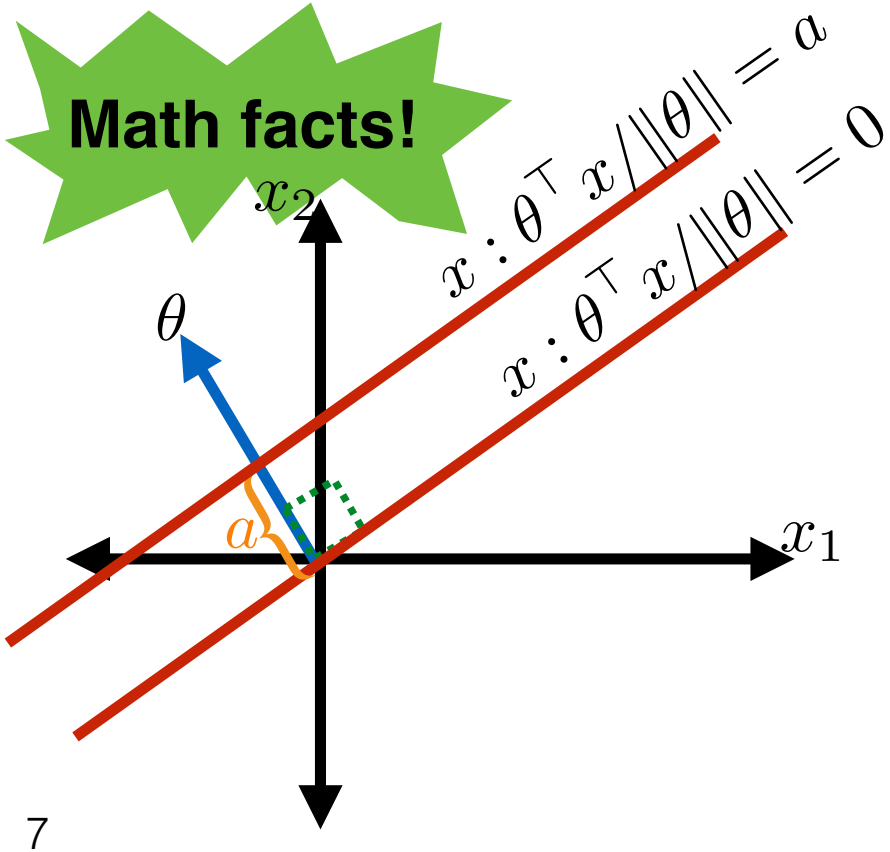


Linear classifiers

- 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

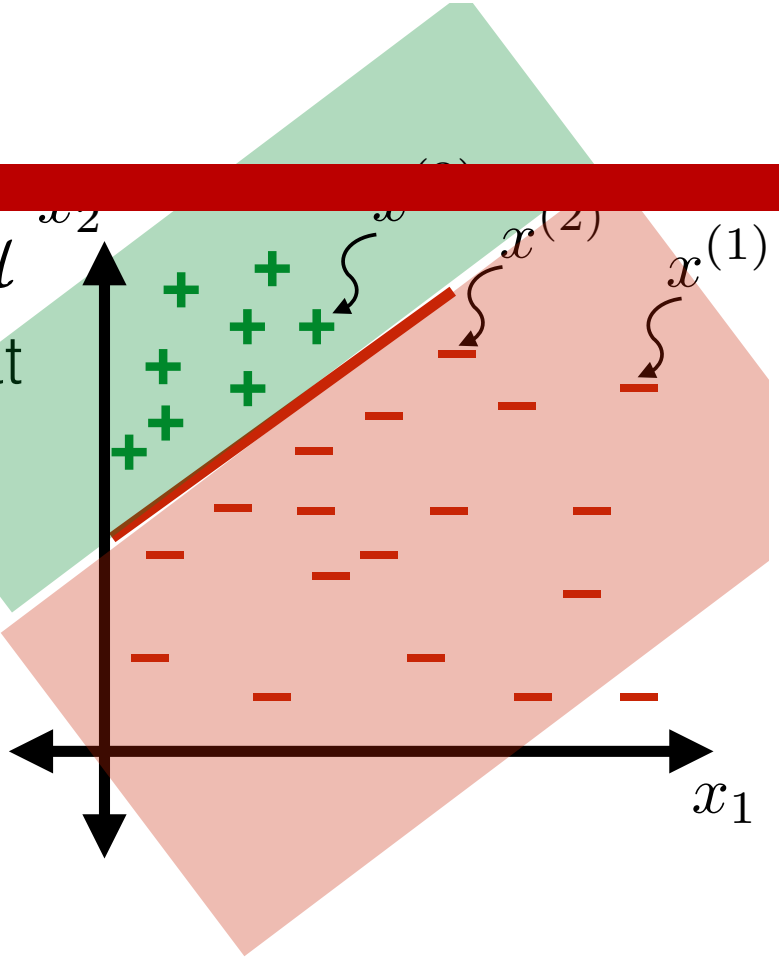


Math facts!

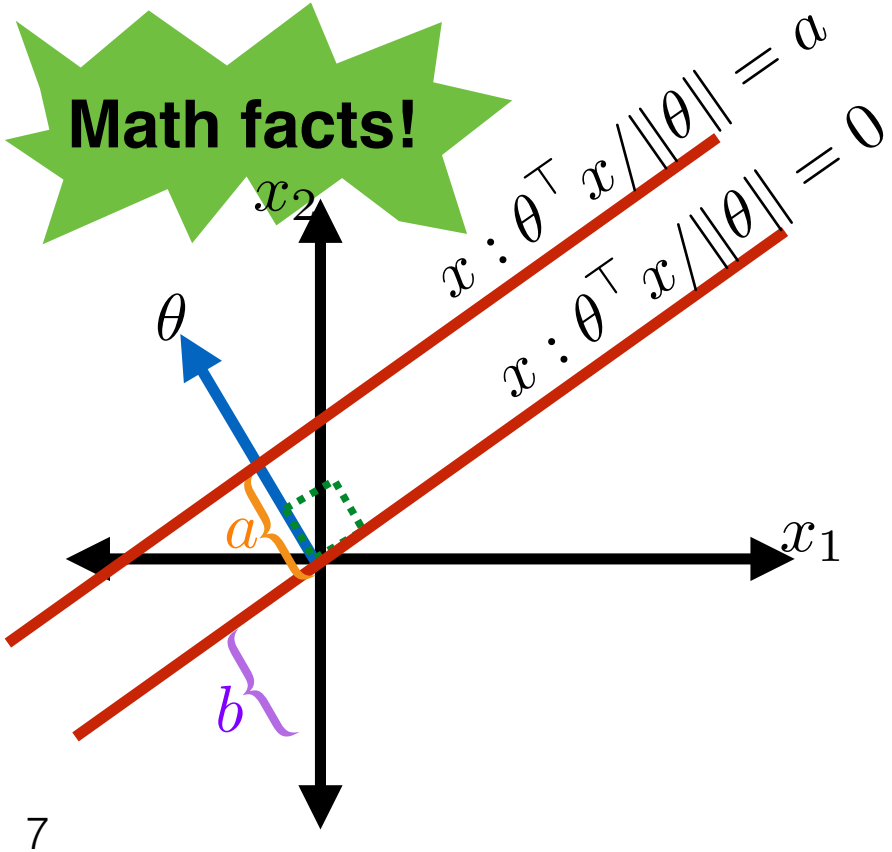


Linear classifiers

- 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

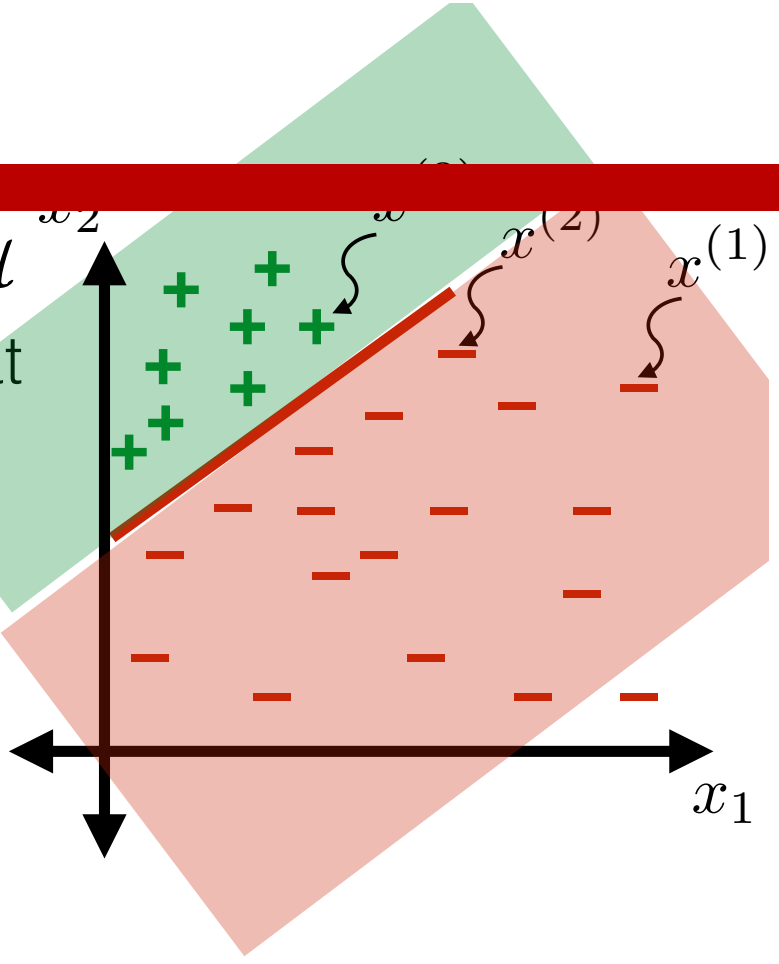


Math facts!

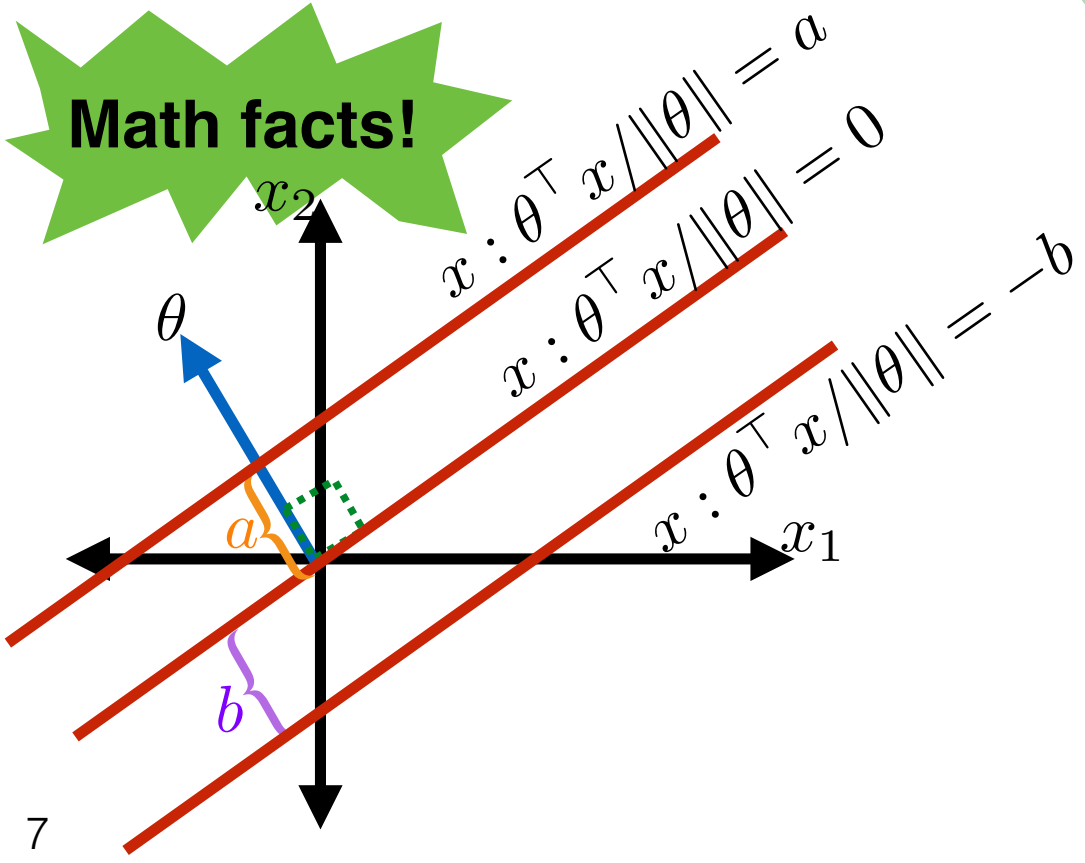


Linear classifiers

- 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

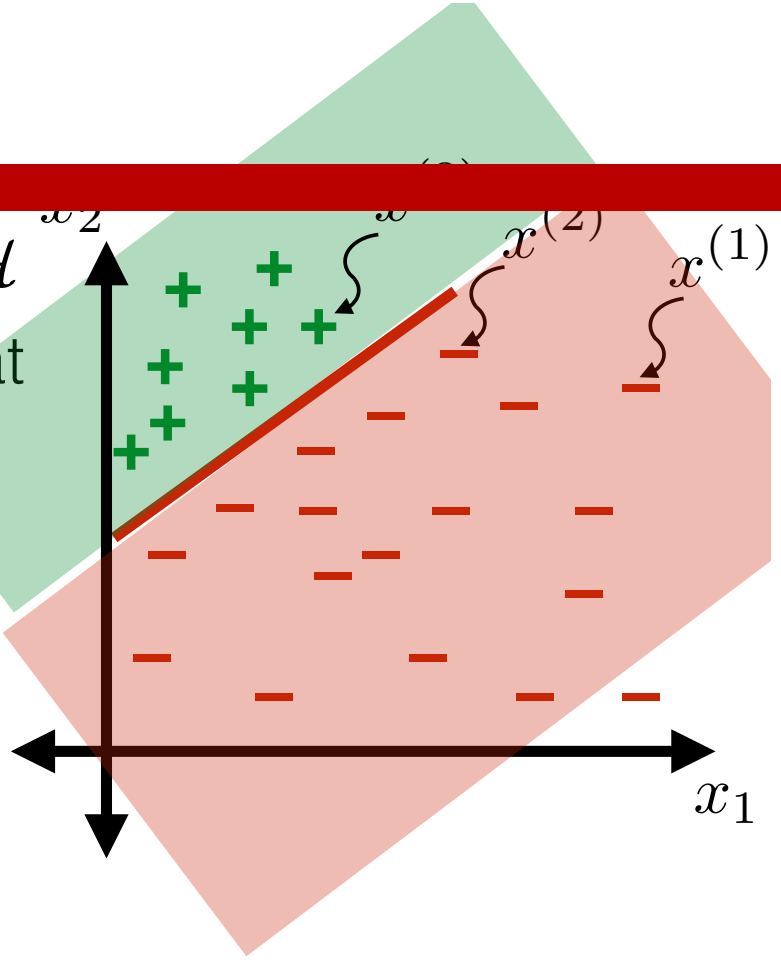


Math facts!

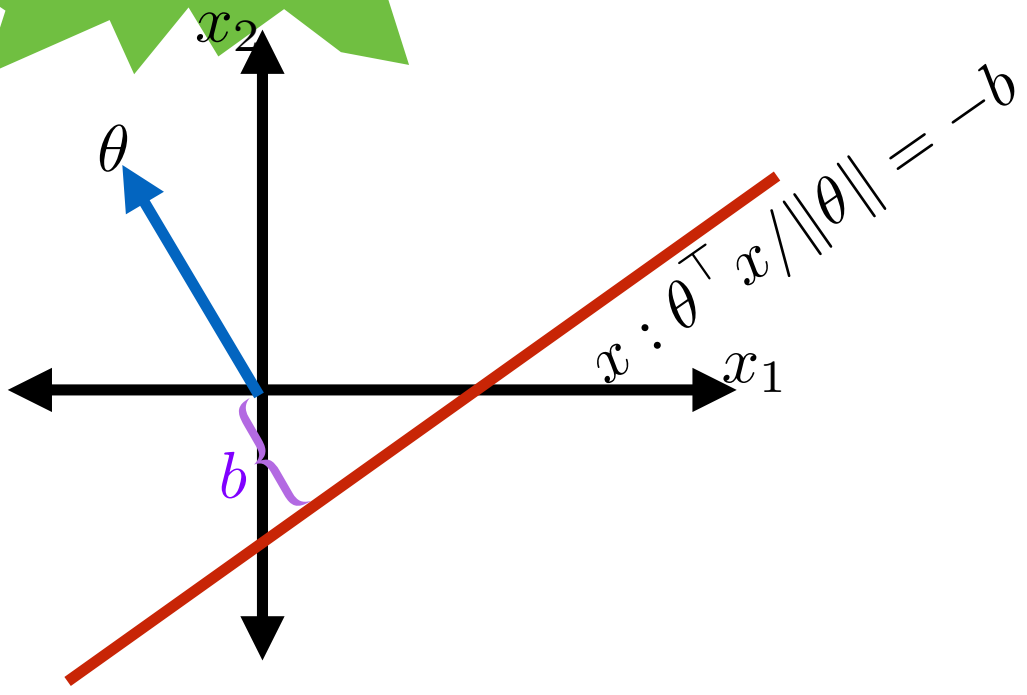


Linear classifiers

- 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

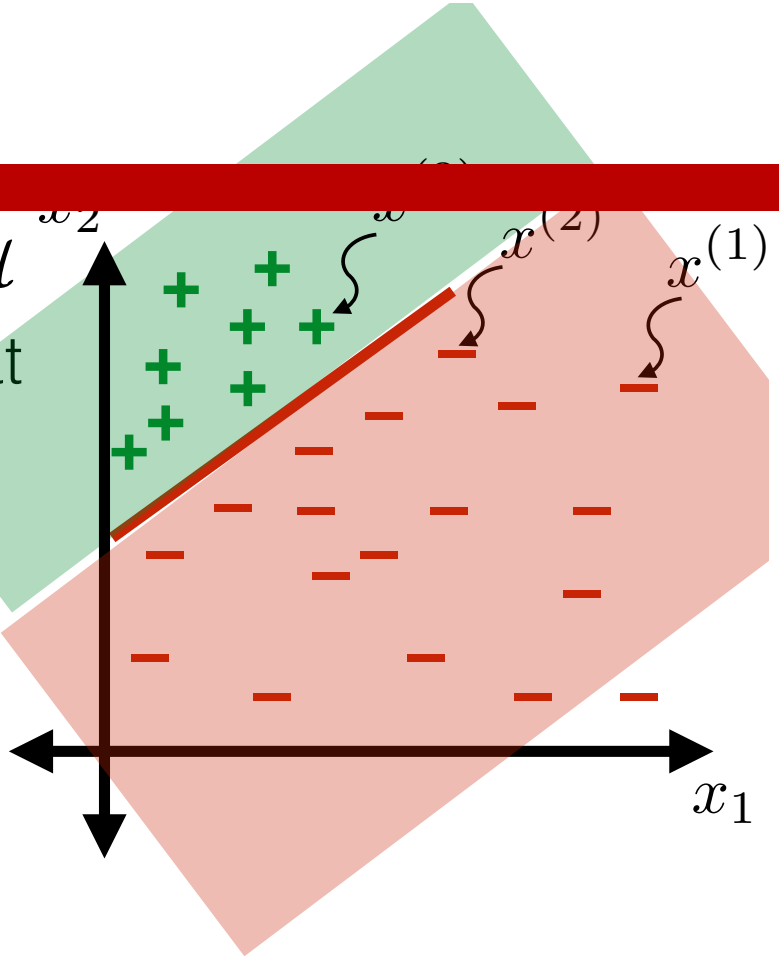


Math facts!

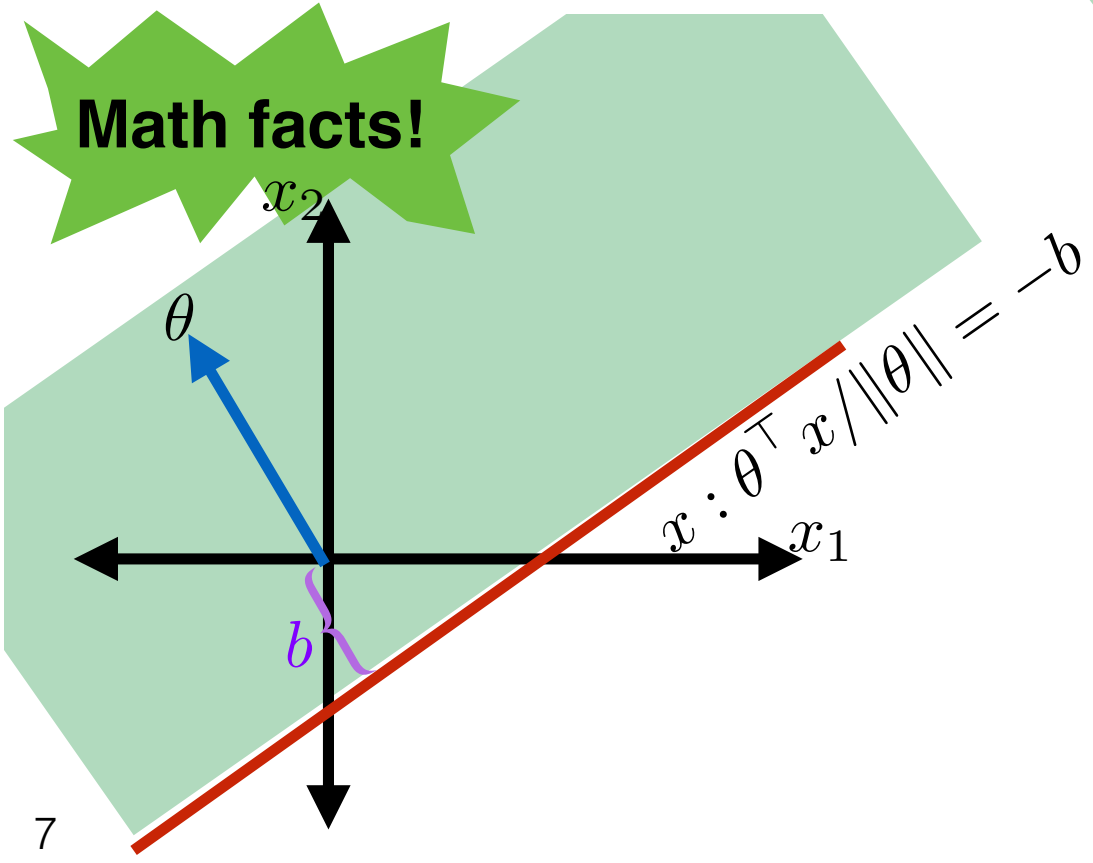


Linear classifiers

- 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

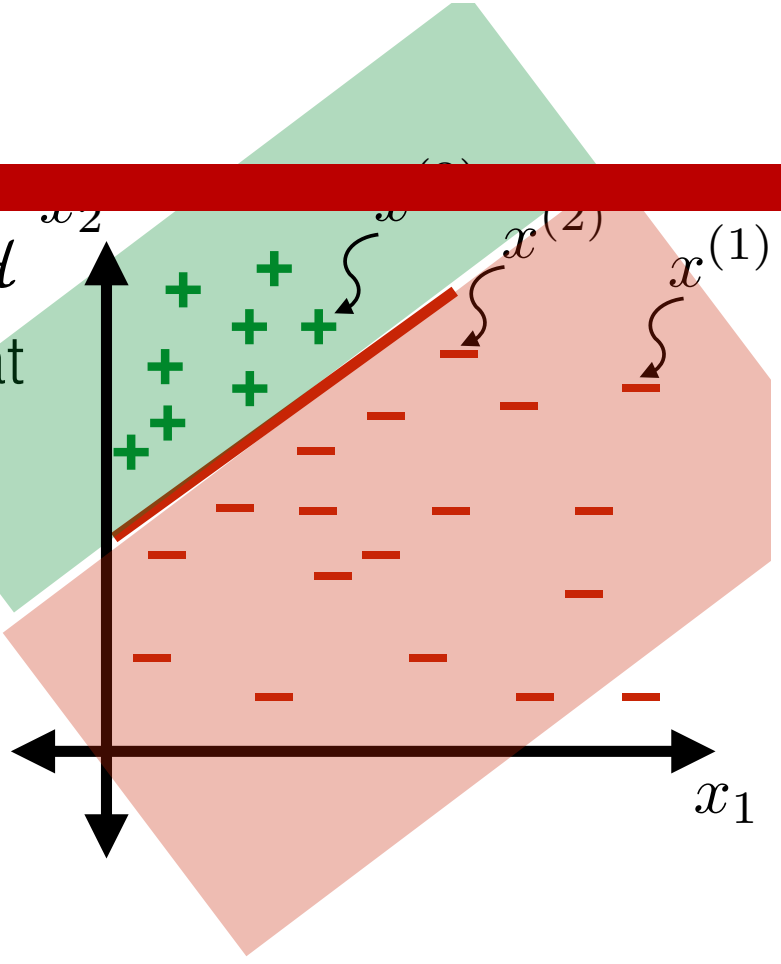


Math facts!

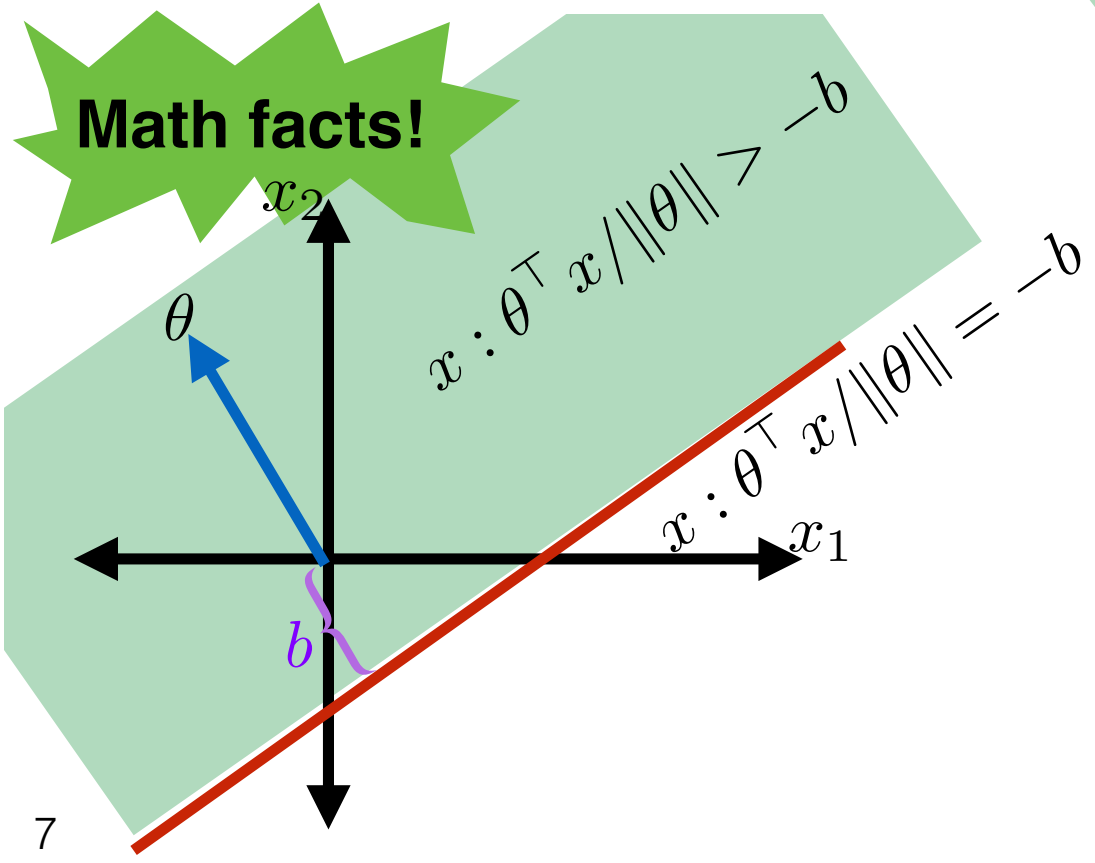


Linear classifiers

- 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

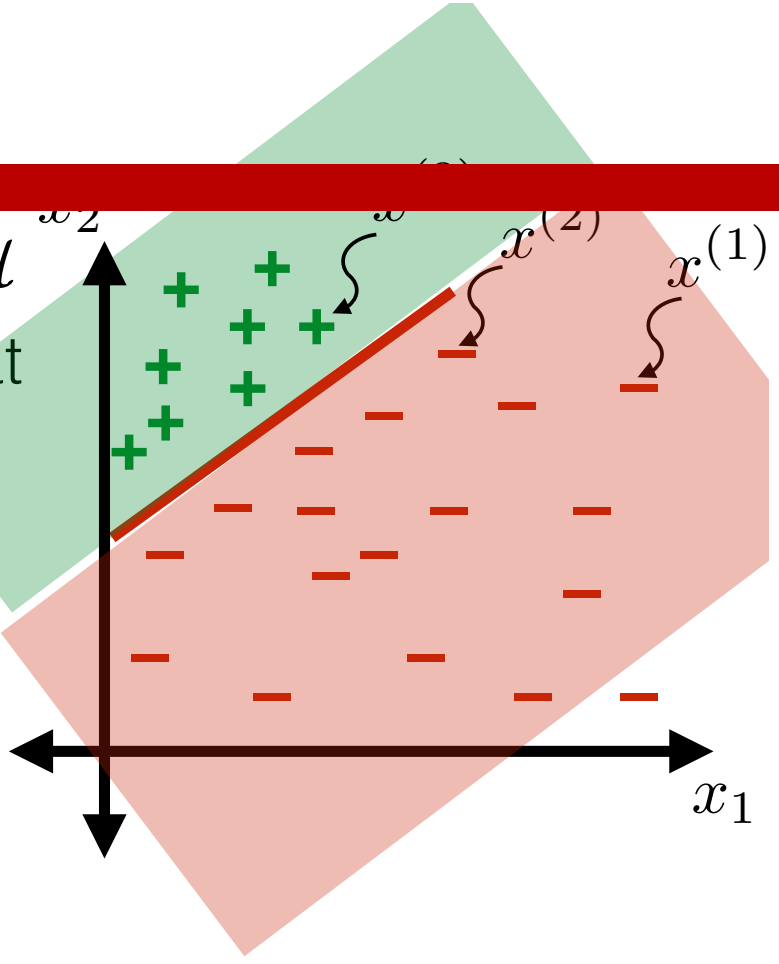


Math facts!

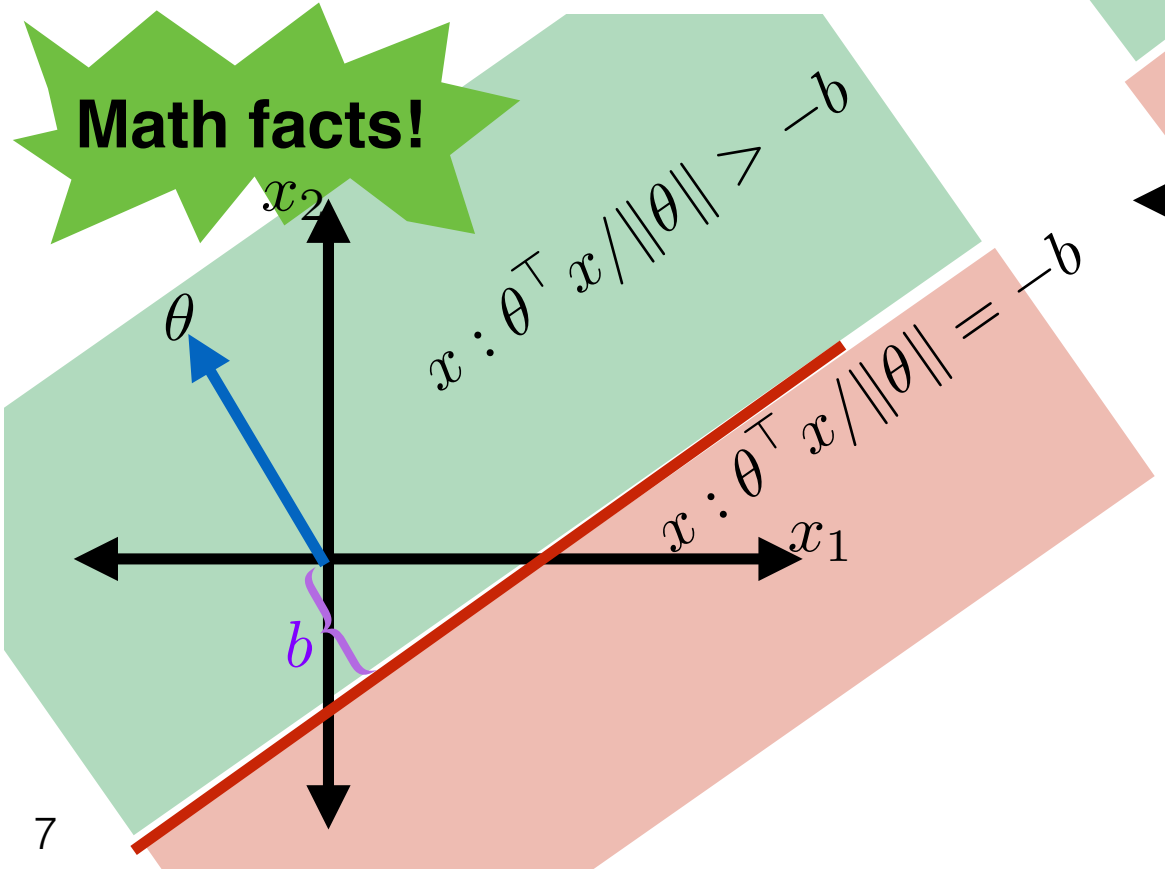


Linear classifiers

- 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

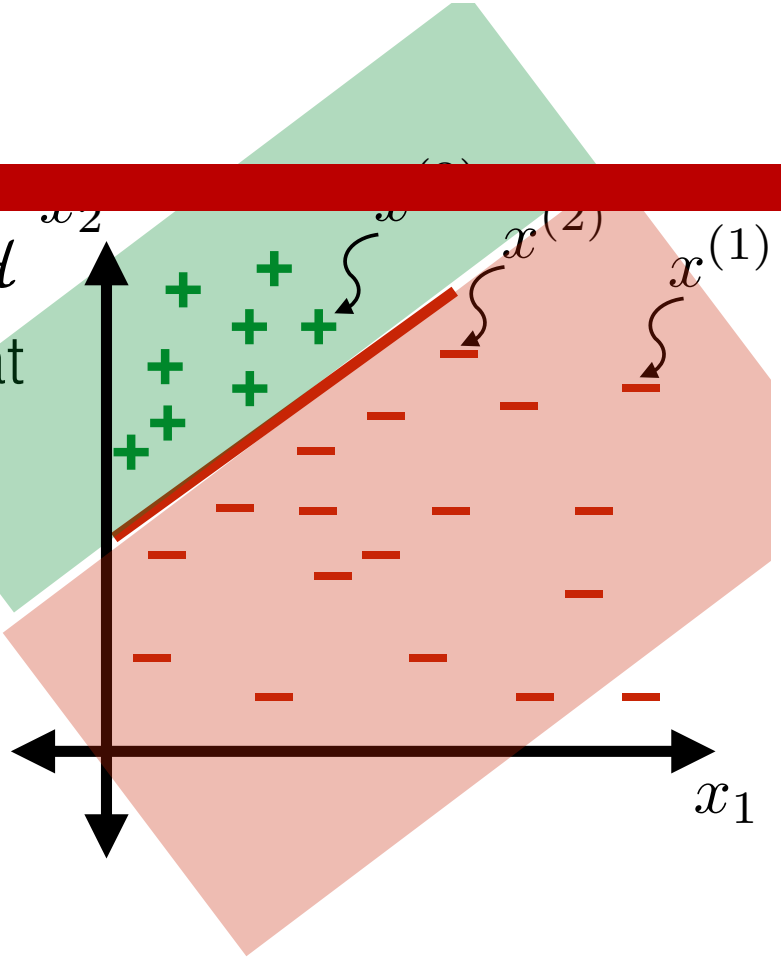


Math facts!

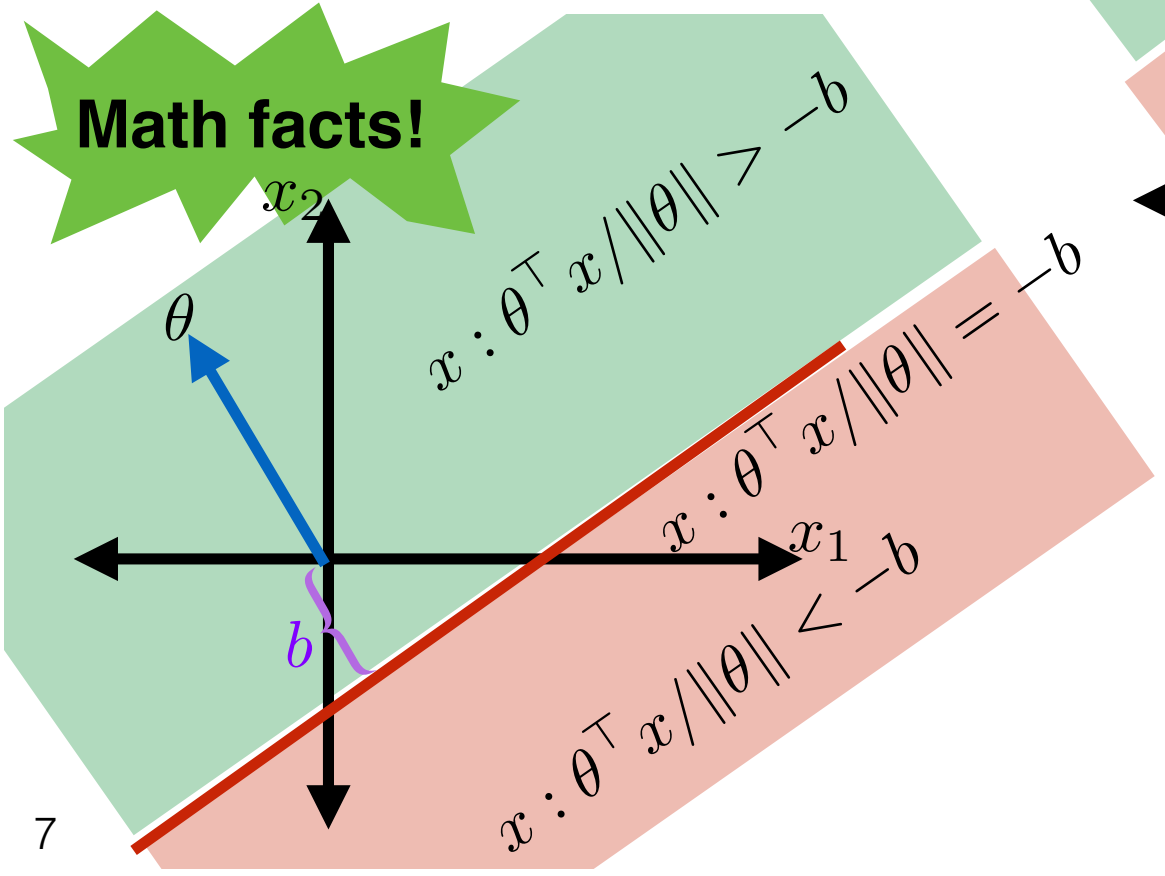


Linear classifiers

- 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

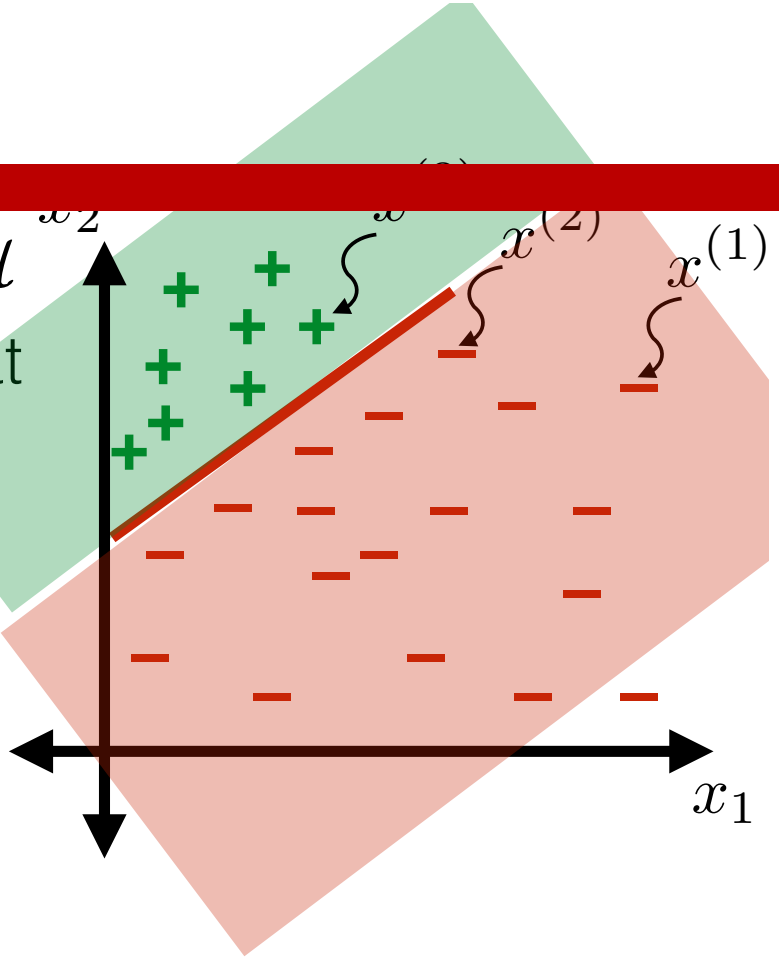


Math facts!

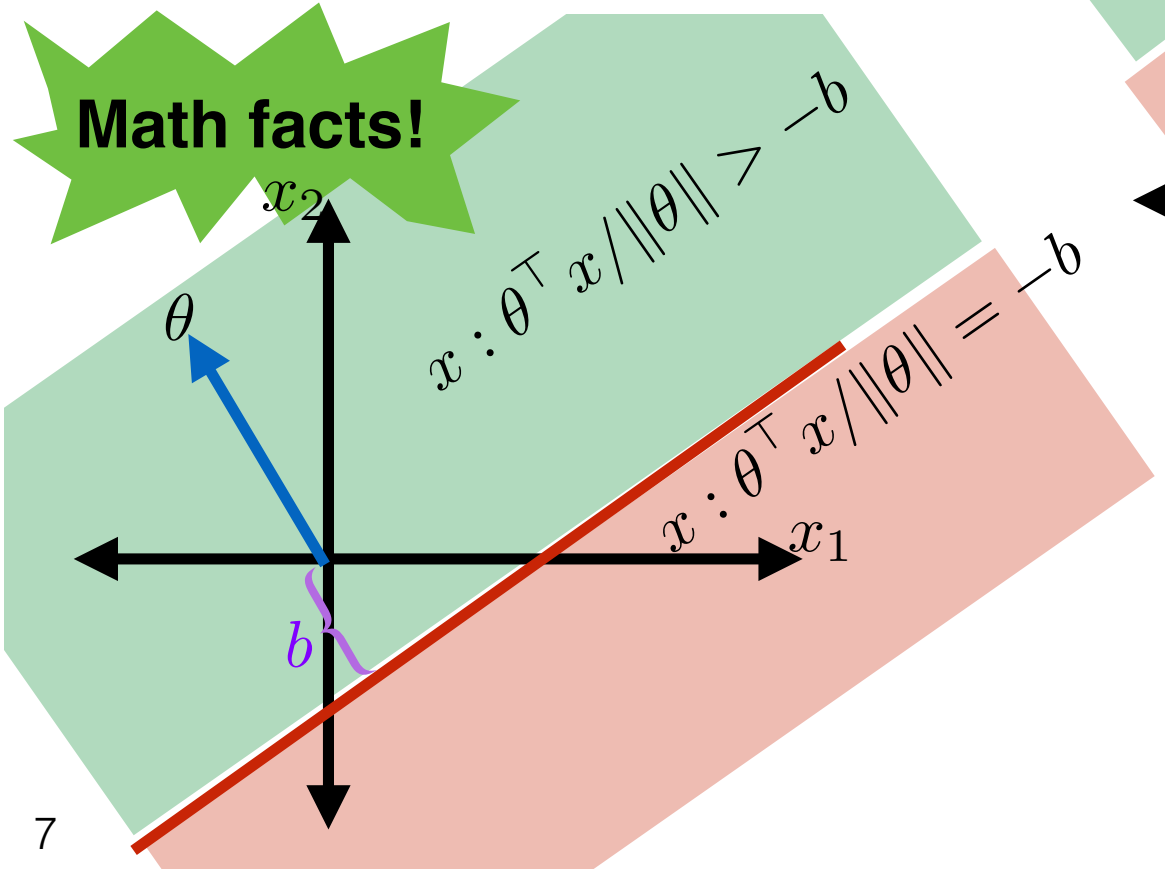


Linear classifiers

- 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

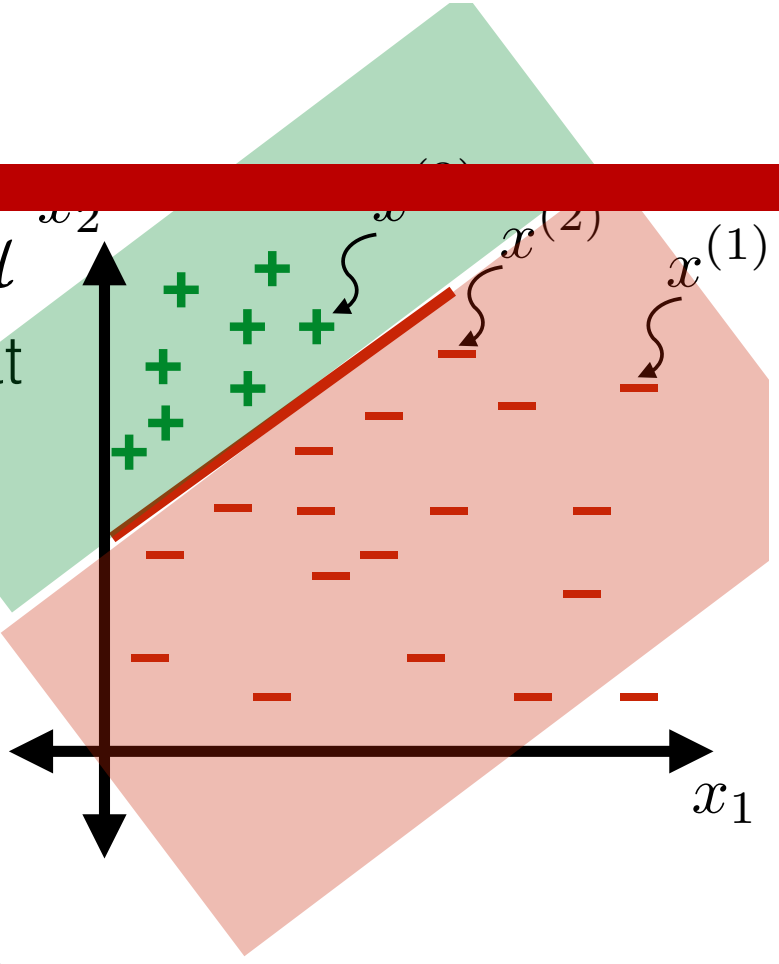


Math facts!

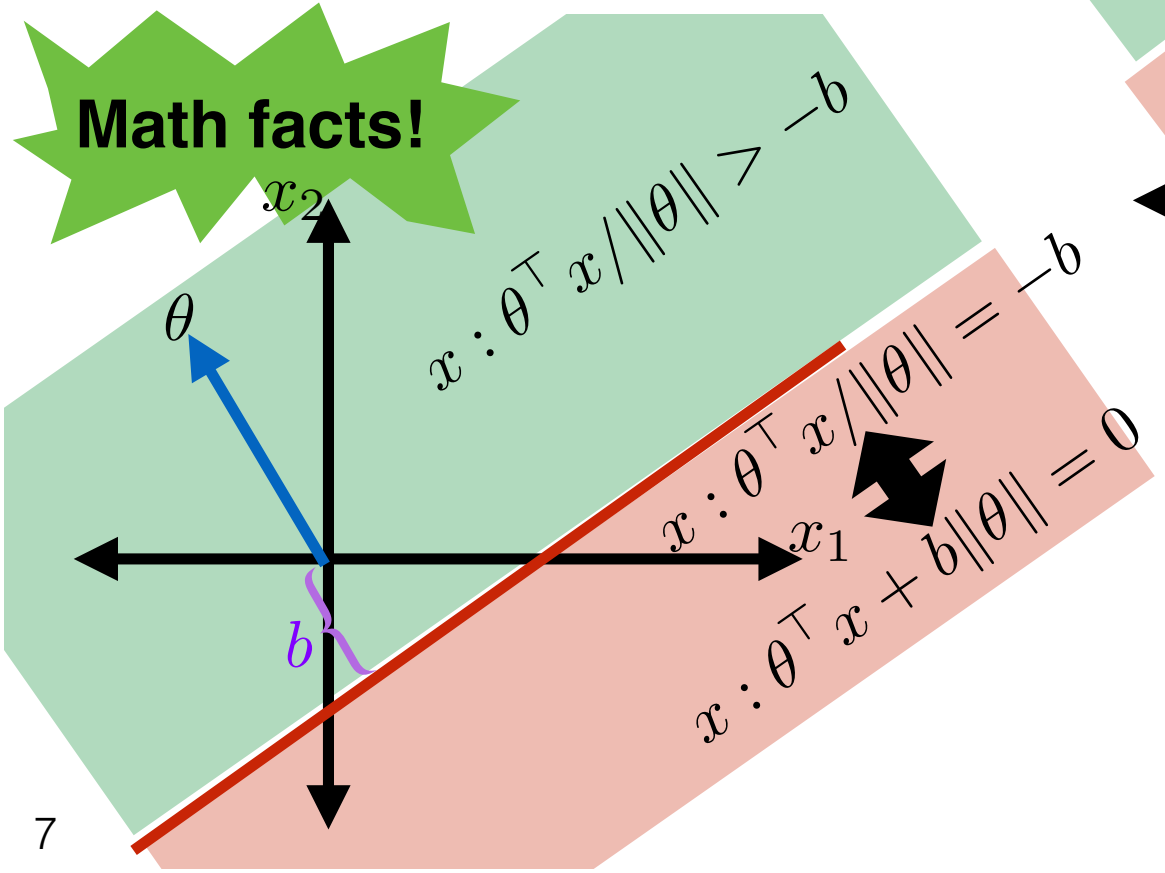


Linear classifiers

- 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

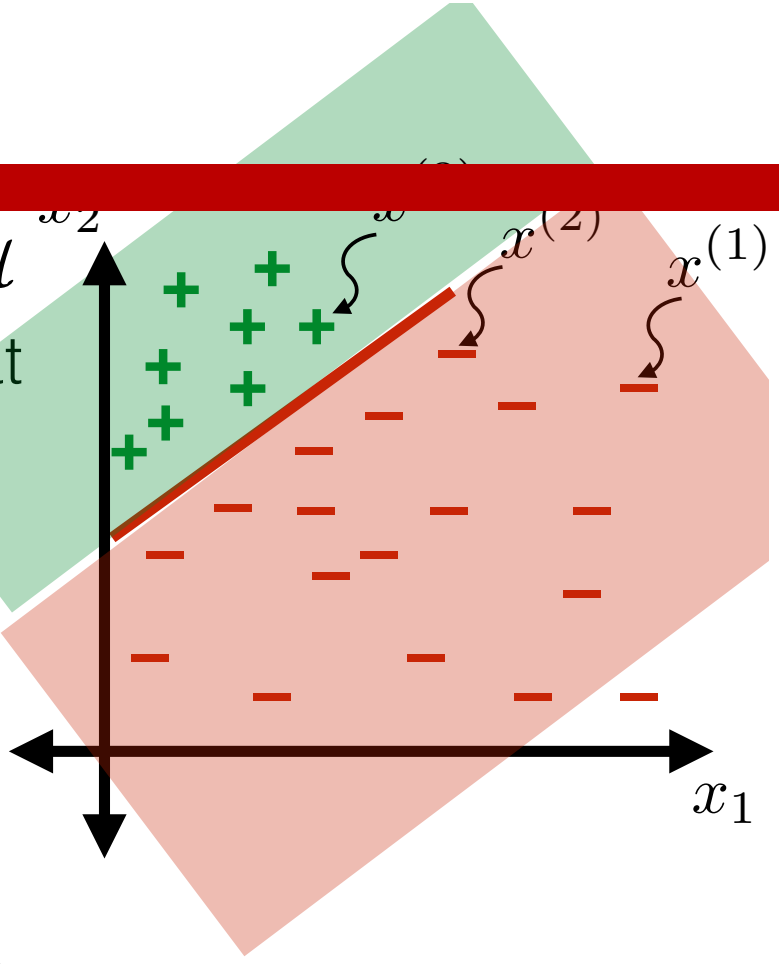


Math facts!

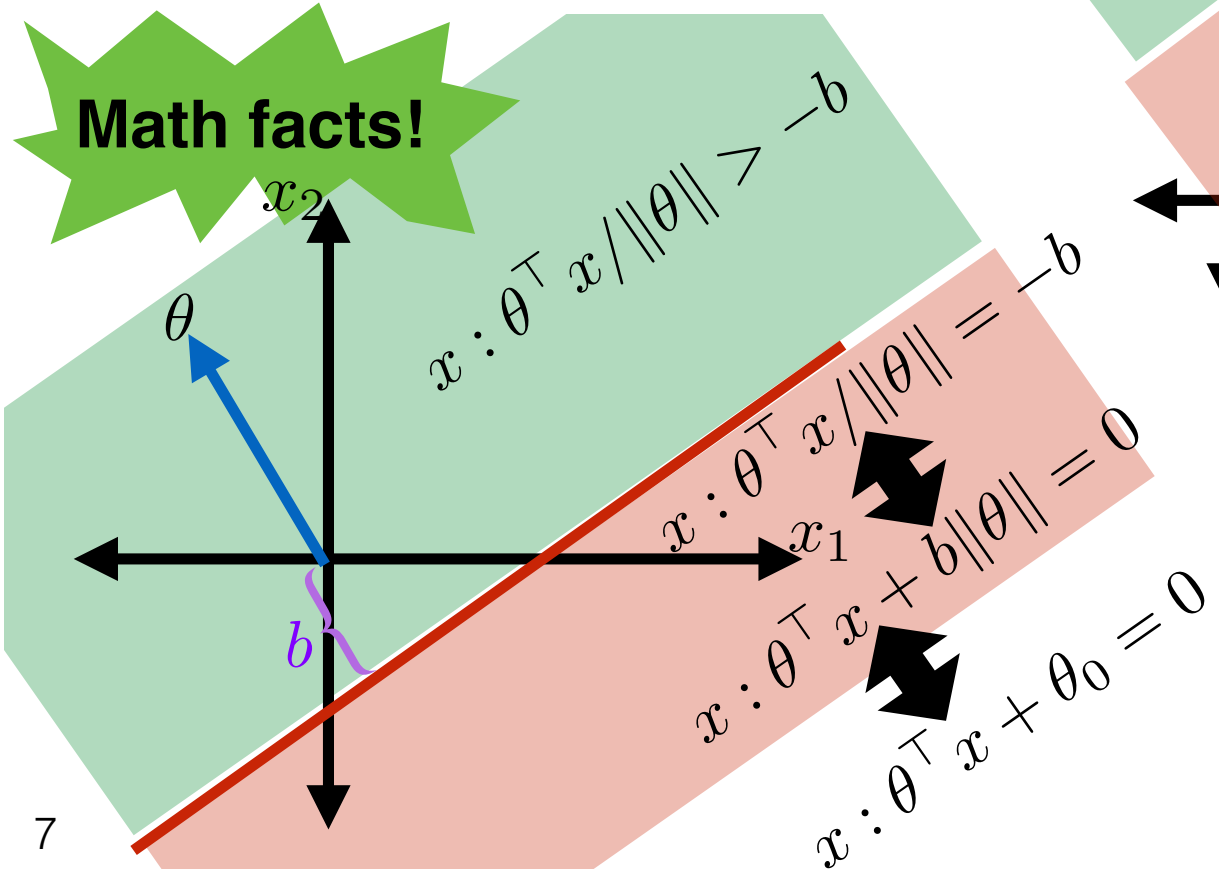


Linear classifiers

- 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

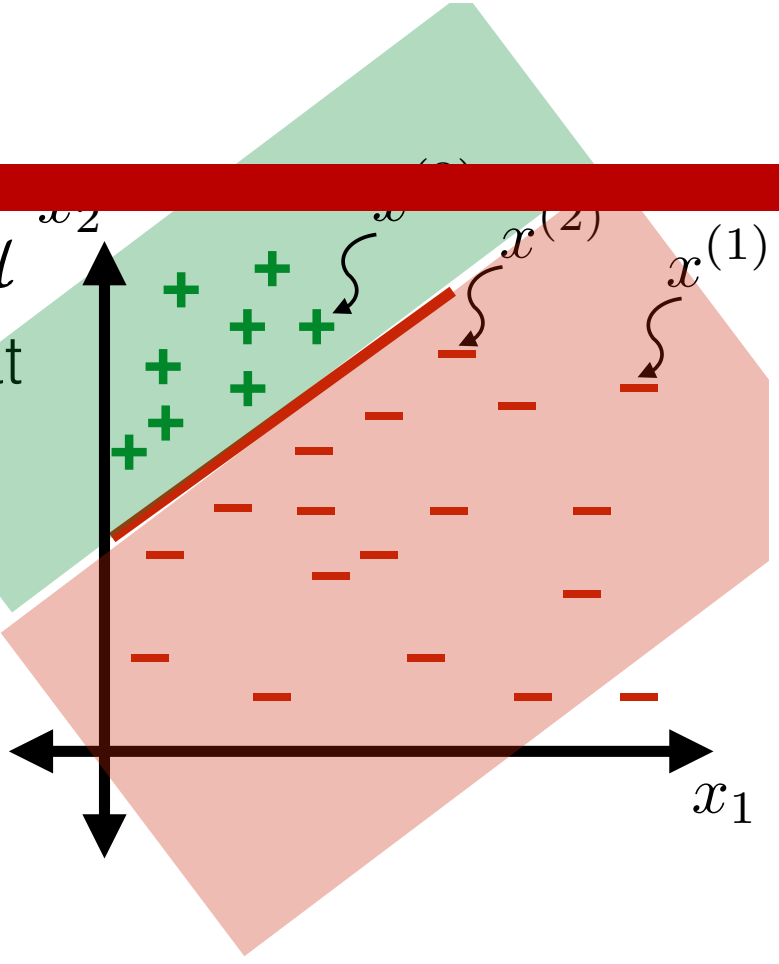


Math facts!

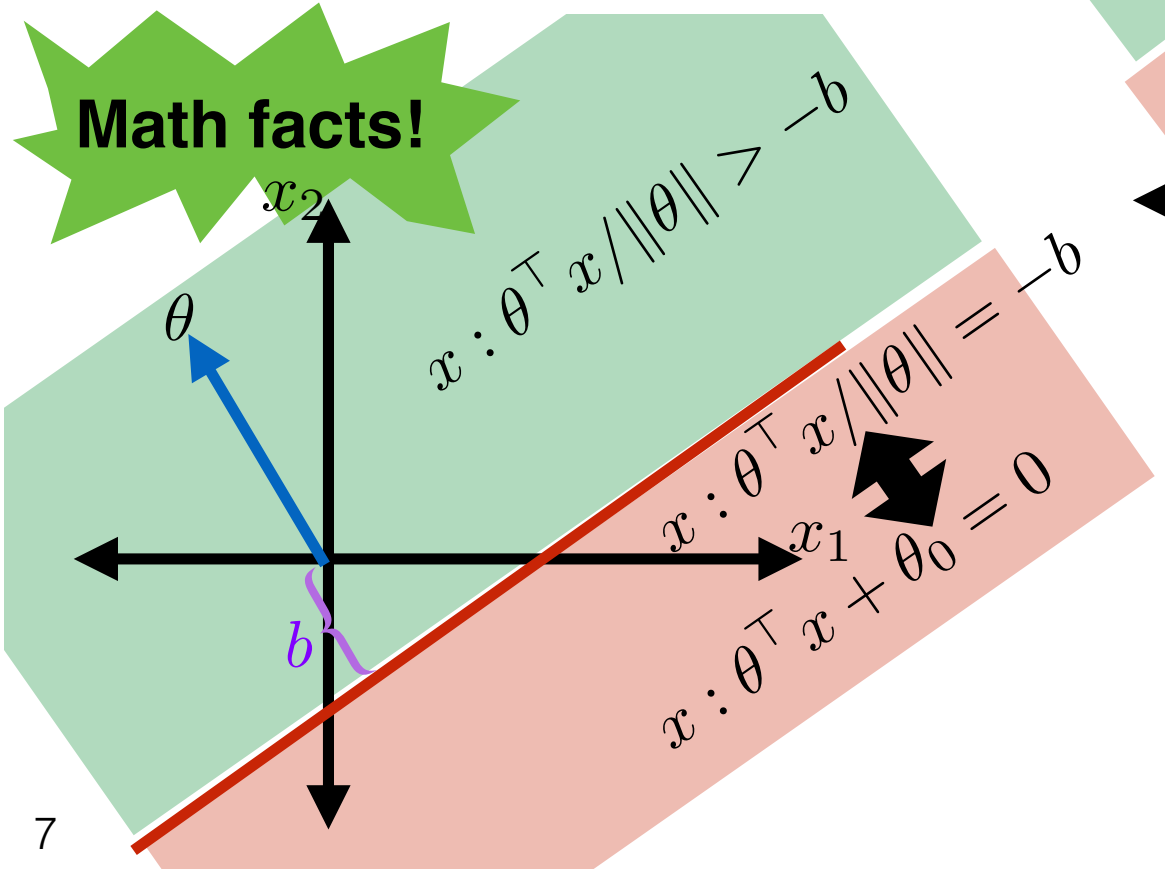


Linear classifiers

- 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

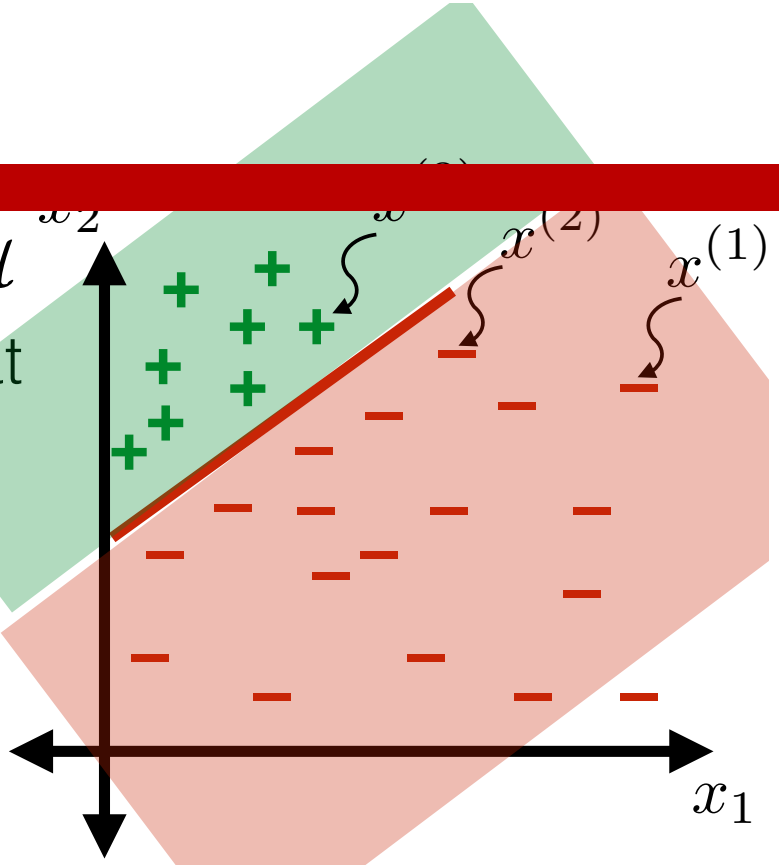


Math facts!

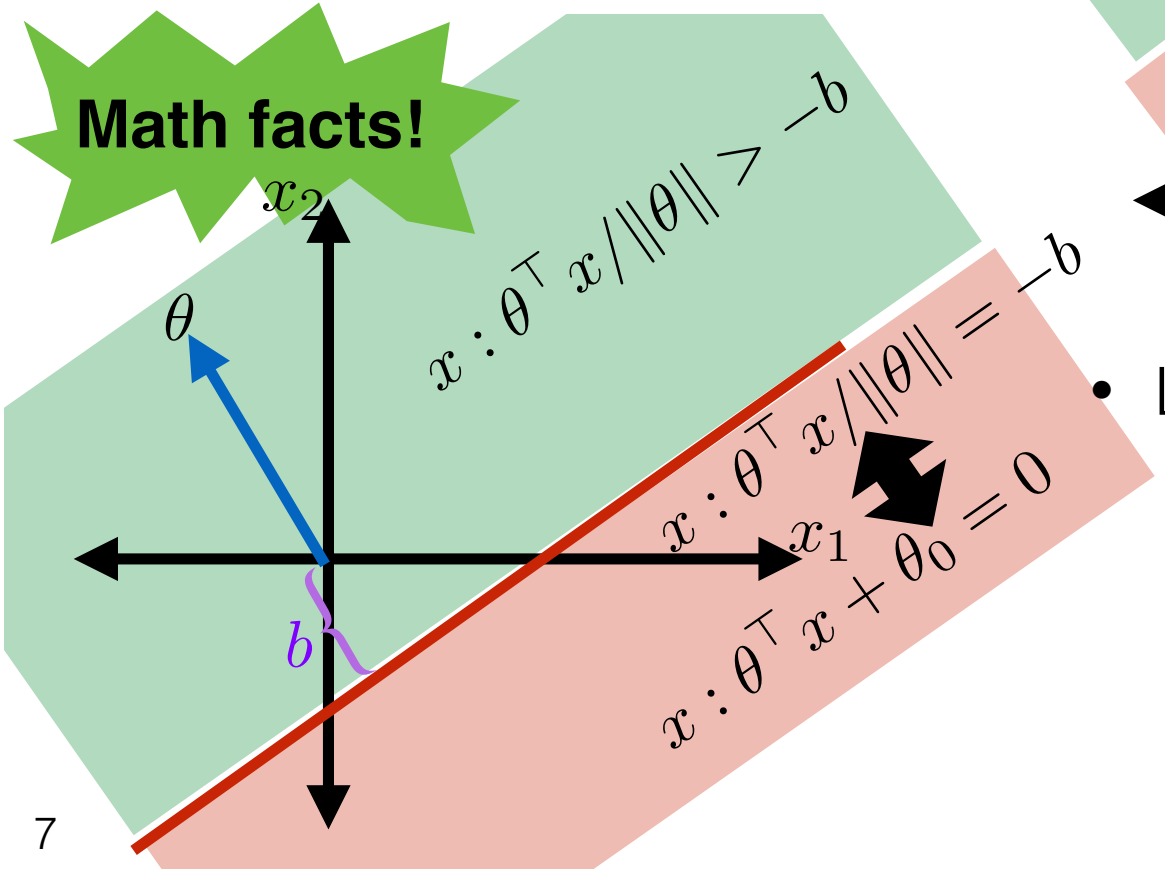


Linear classifiers

- 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



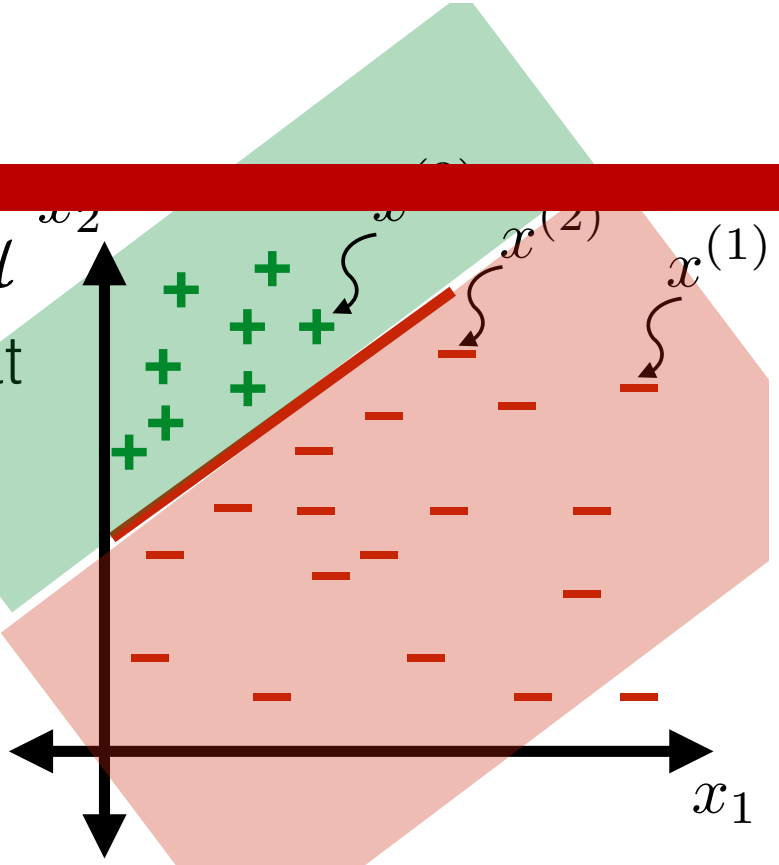
Math facts!



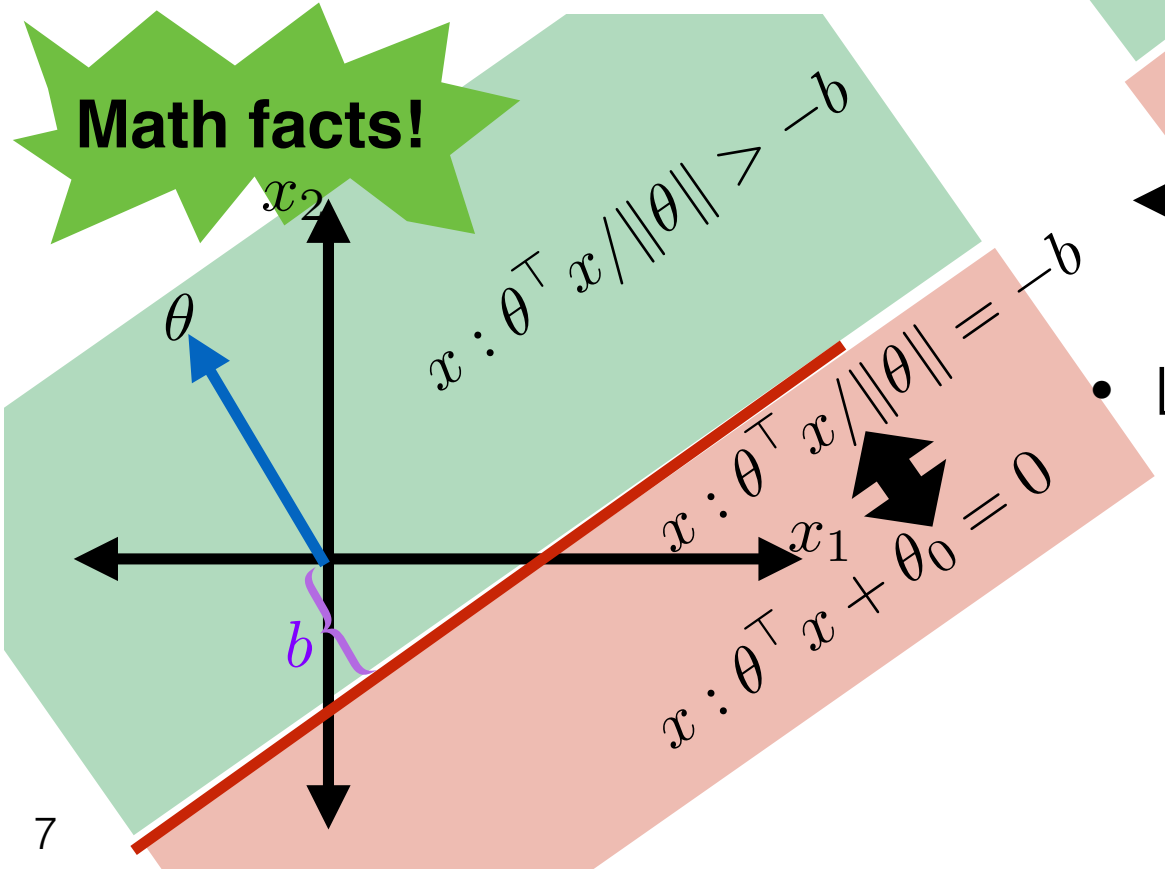
- Linear classifier:

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!

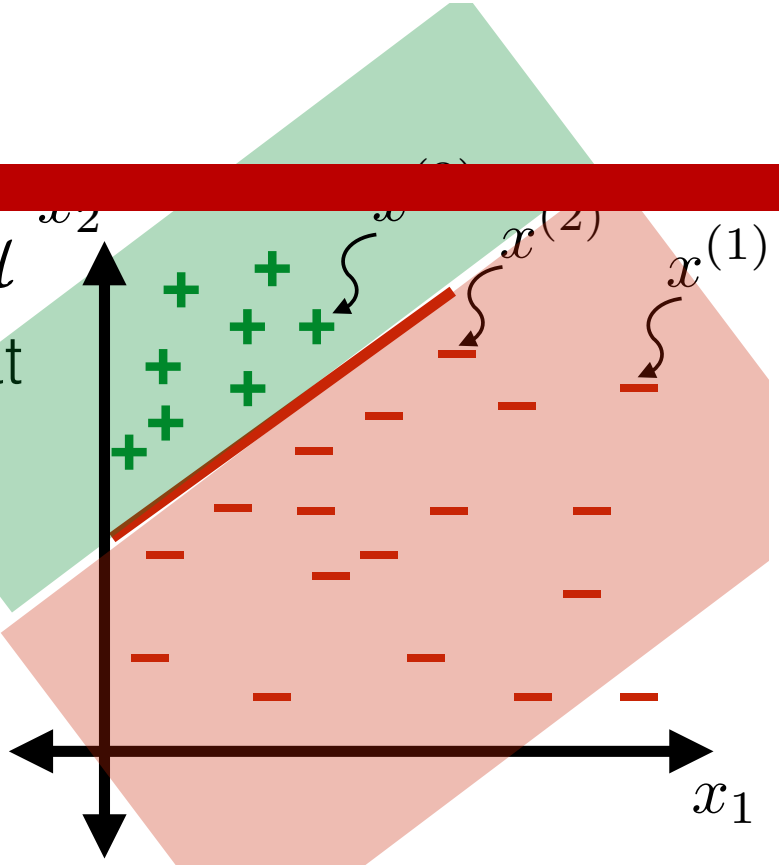


- Linear classifier:

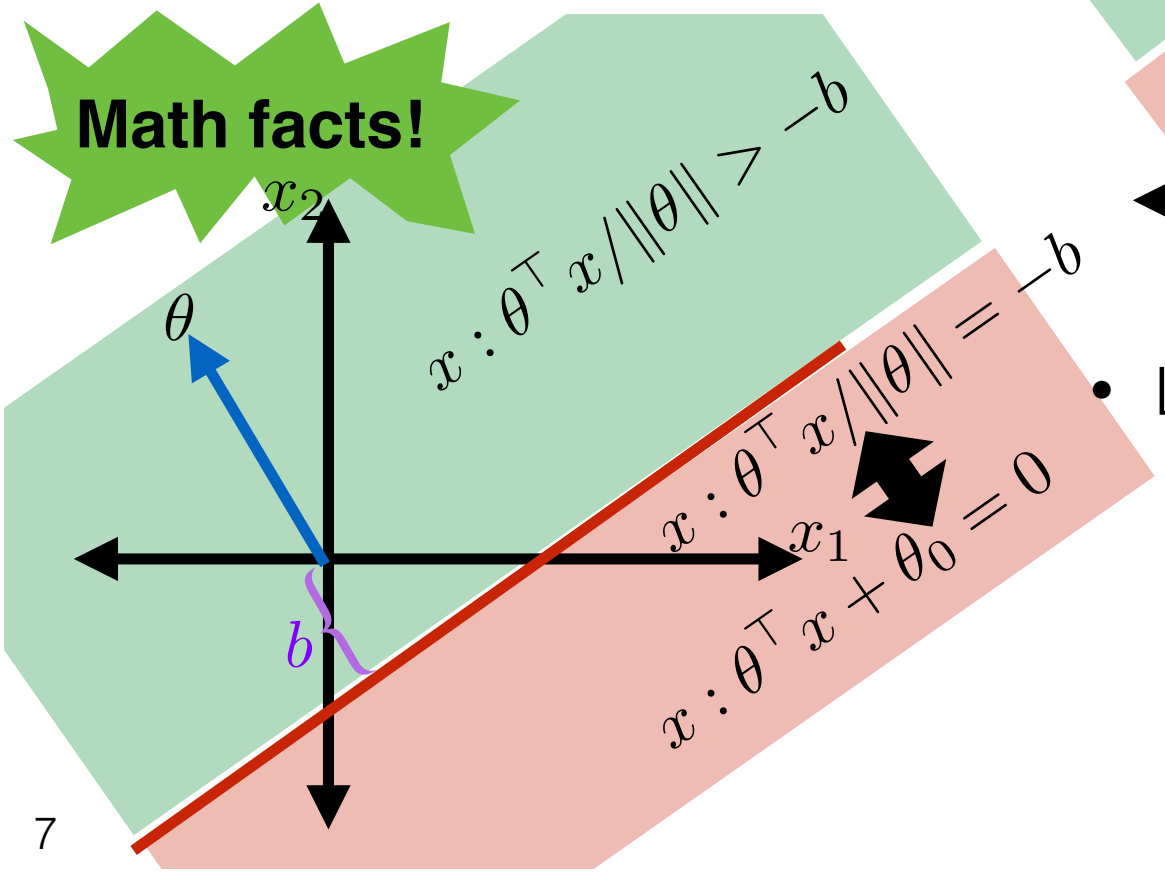
$$h(x) = \text{sign}(\theta^\top x + \theta_0)$$

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!



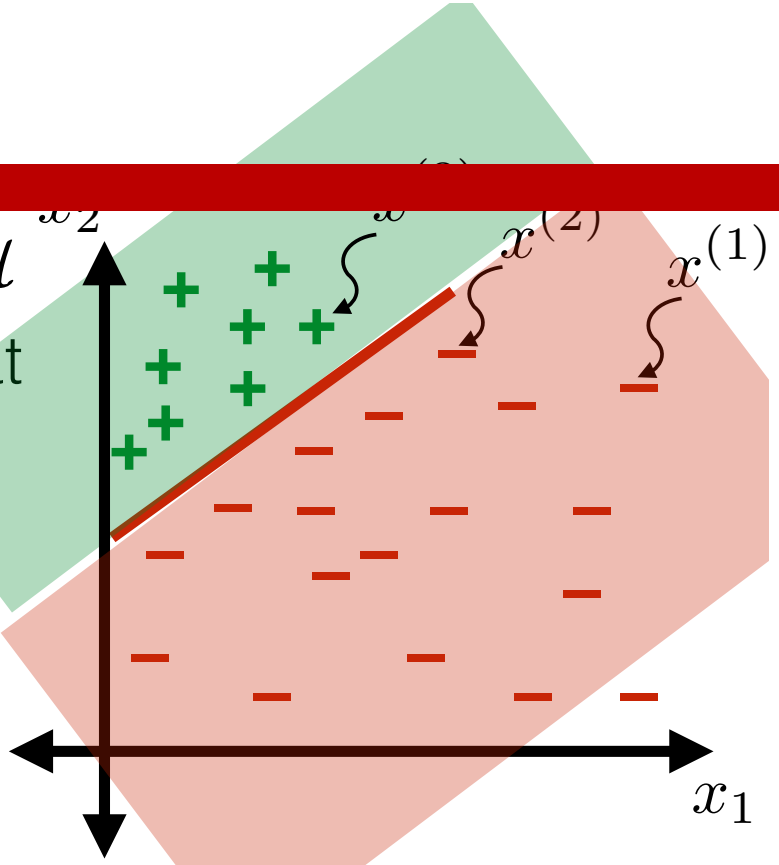
- Linear classifier:

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

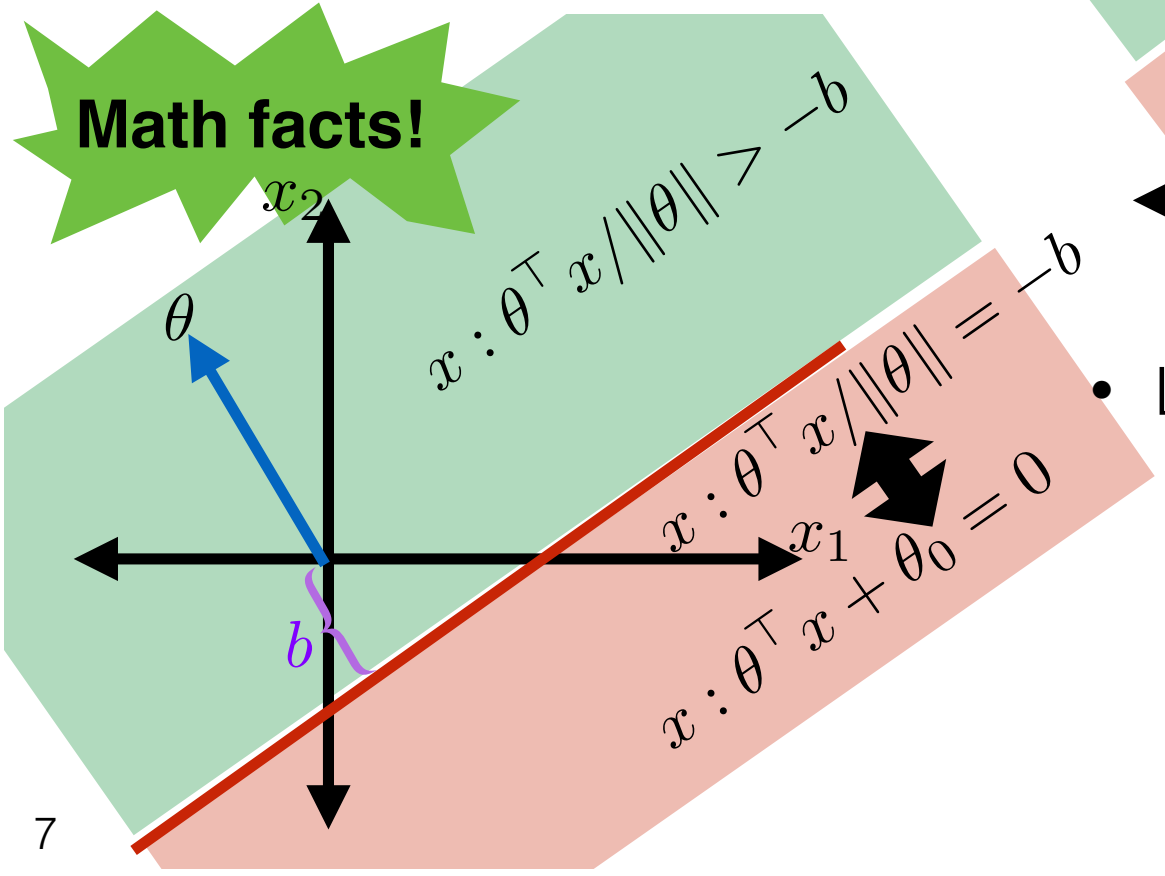
$$= \begin{cases} +1 \text{ if } \theta^\top x + \theta_0 > 0 \\ -1 \text{ if } \theta^\top x + \theta_0 < 0 \end{cases}$$

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!



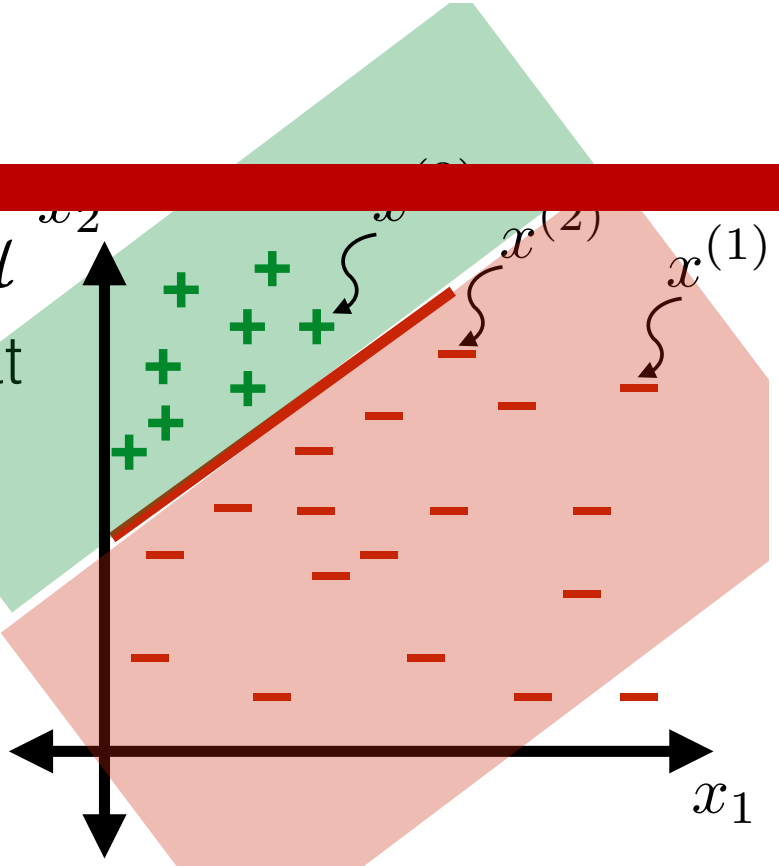
- Linear classifier:

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

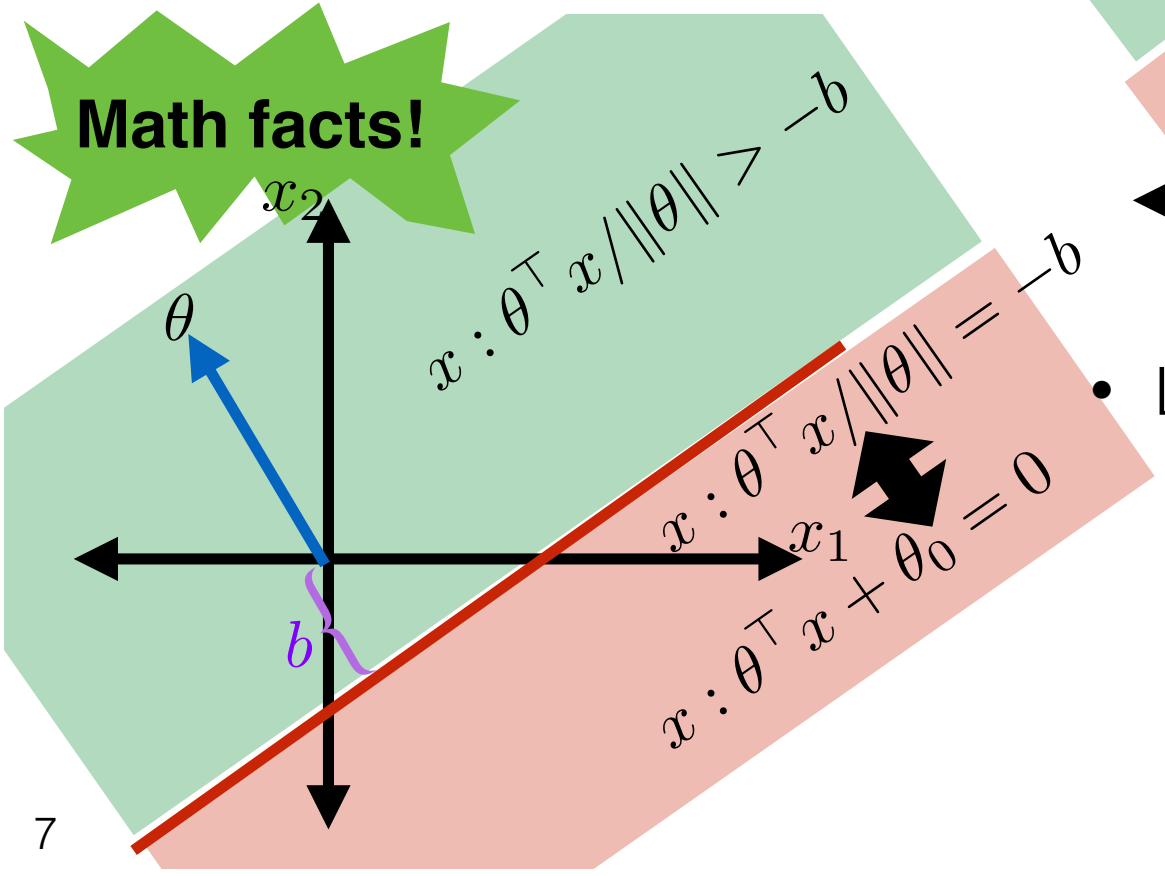
$$= \begin{cases} +1 \text{ if } \theta^\top x + \theta_0 > 0 \\ -1 \text{ if } \theta^\top x + \theta_0 < 0 \end{cases}$$

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!



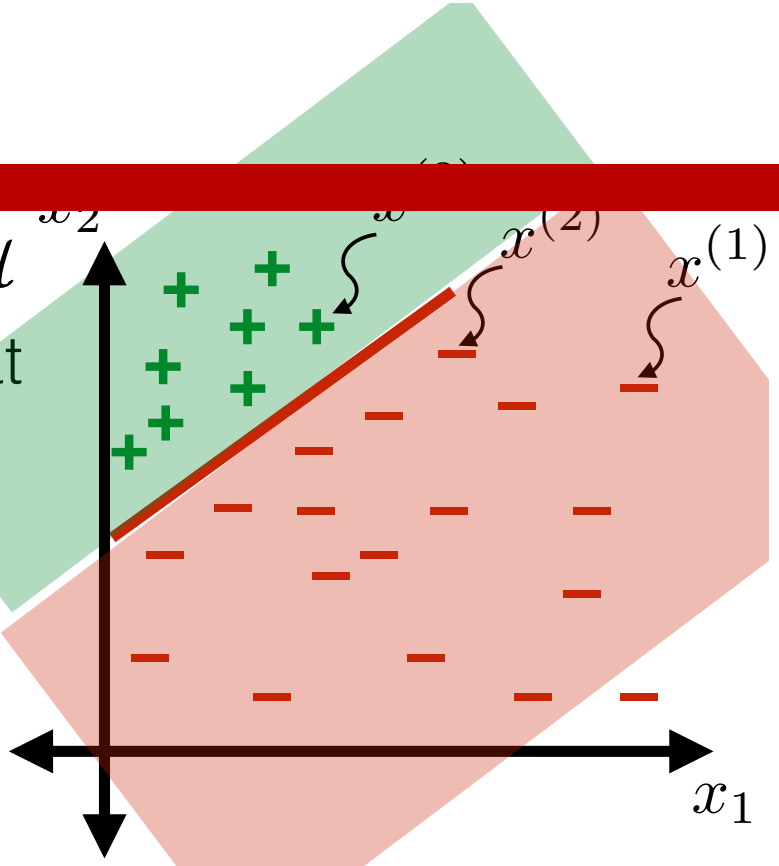
- Linear classifier:

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

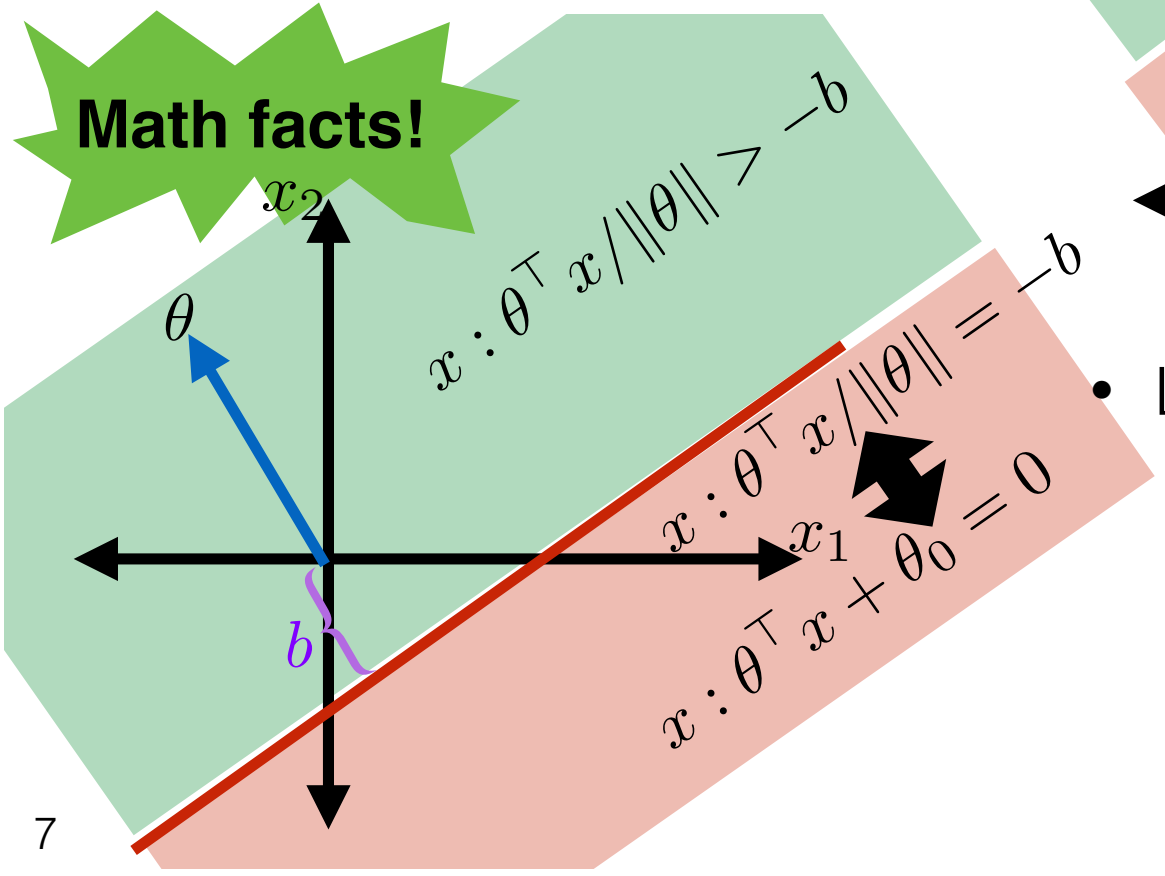
$$= \begin{cases} +1 \text{ if } \theta^\top x + \theta_0 > 0 \\ -1 \text{ if } \theta^\top x + \theta_0 \leq 0 \end{cases}$$

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!



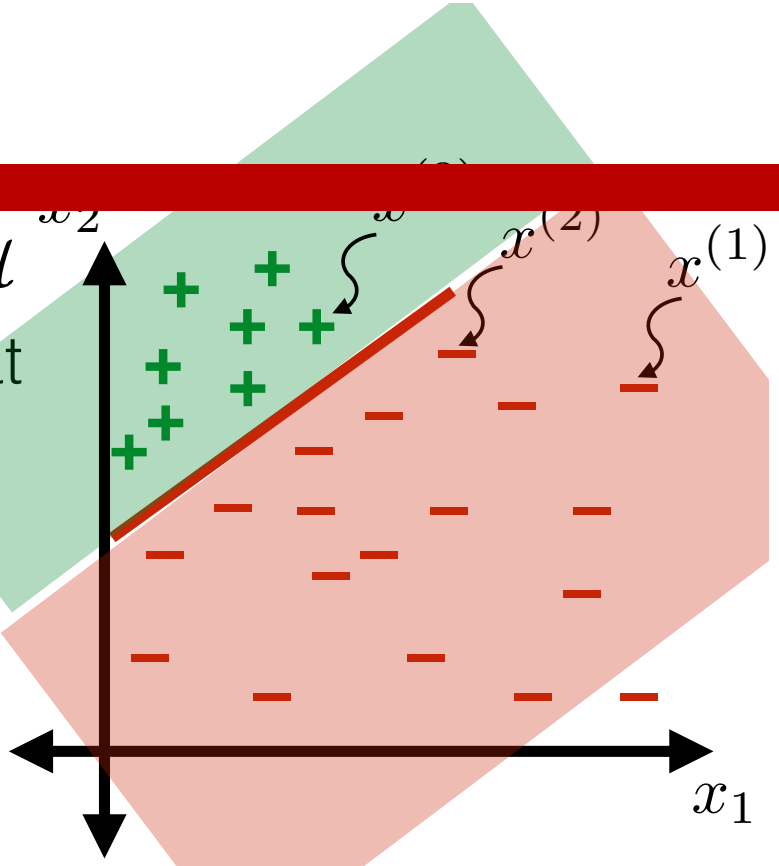
- Linear classifier:

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

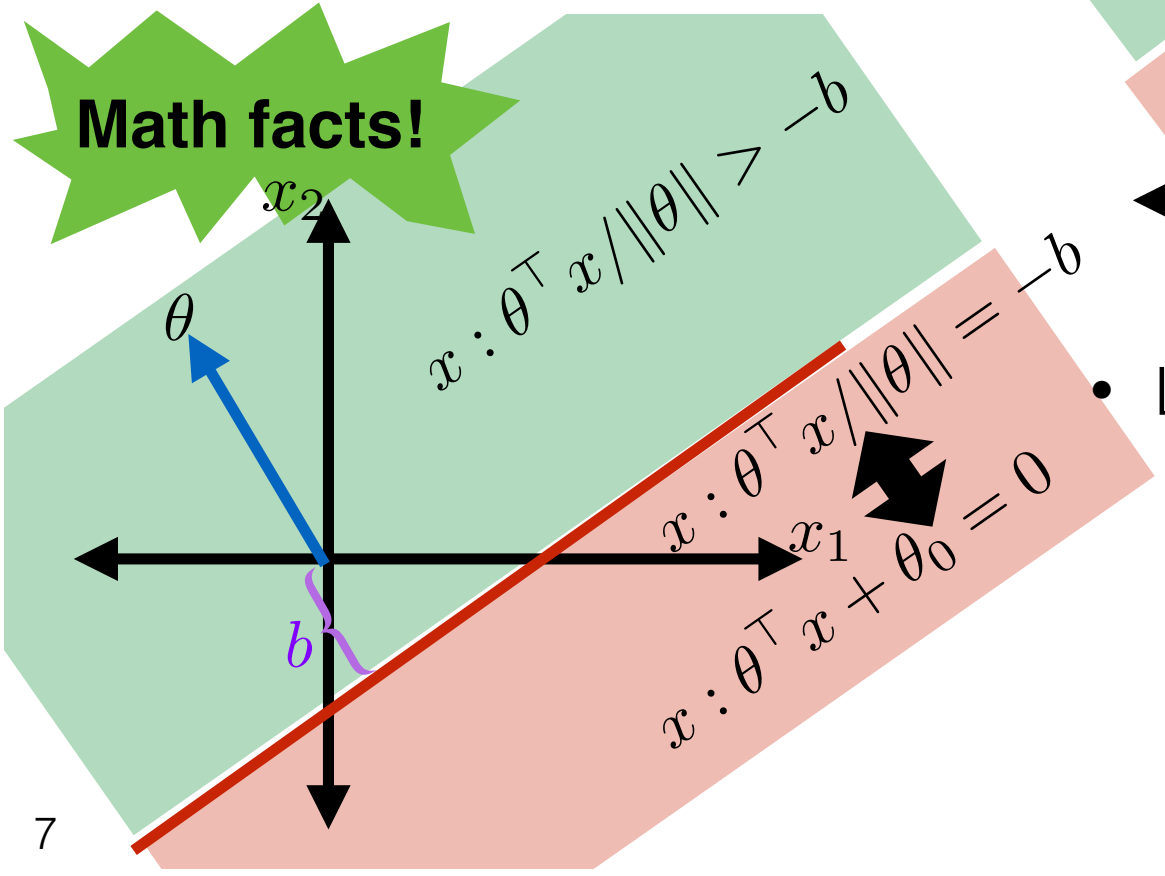
$$= \begin{cases} +1 \text{ if } \theta^\top x + \theta_0 > 0 \\ -1 \text{ if } \theta^\top x + \theta_0 \leq 0 \end{cases}$$

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!



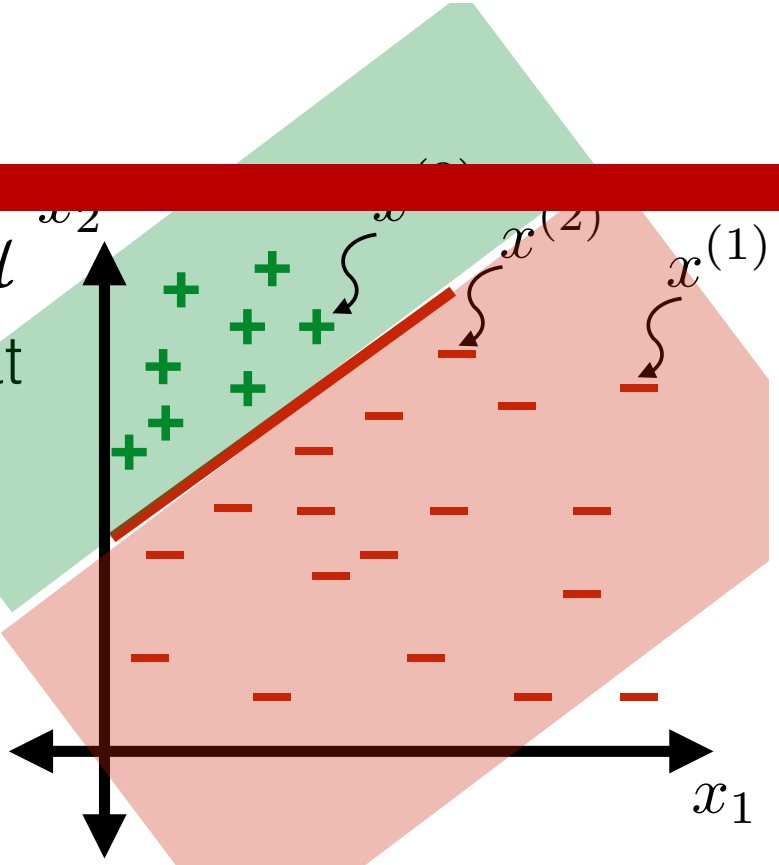
- Linear classifier:

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

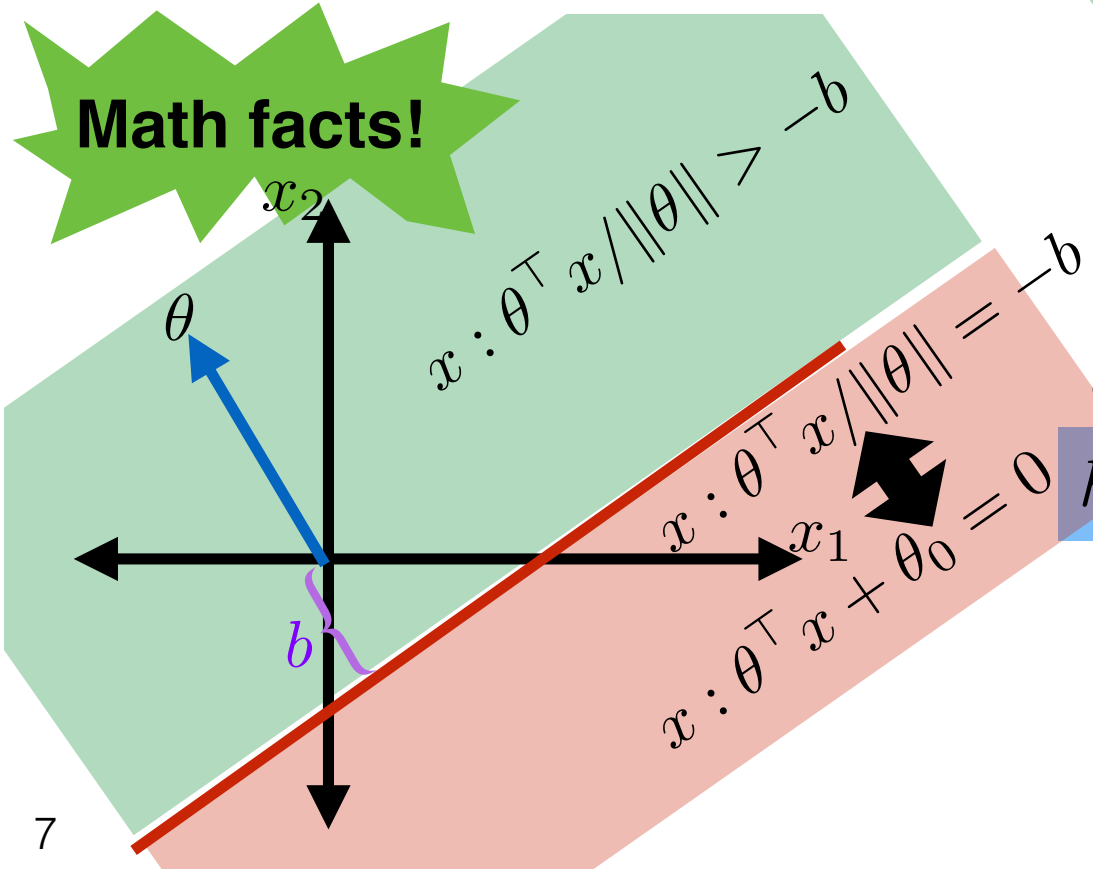
$$= \begin{cases} +1 \text{ if } \theta^\top x + \theta_0 > 0 \\ -1 \text{ if } \theta^\top x + \theta_0 \leq 0 \end{cases}$$

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!



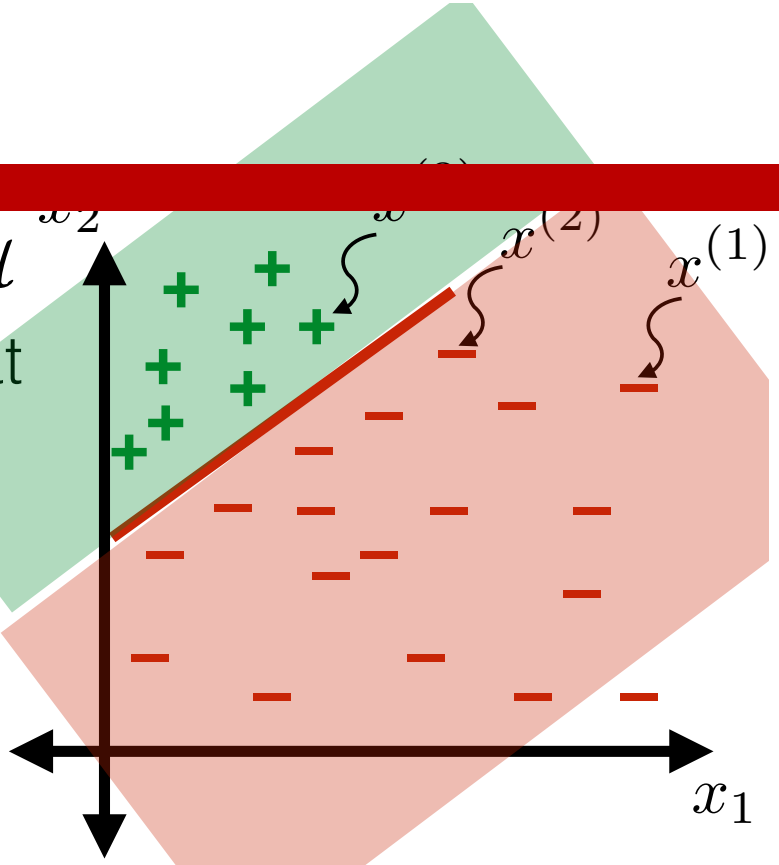
- Linear classifier:

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

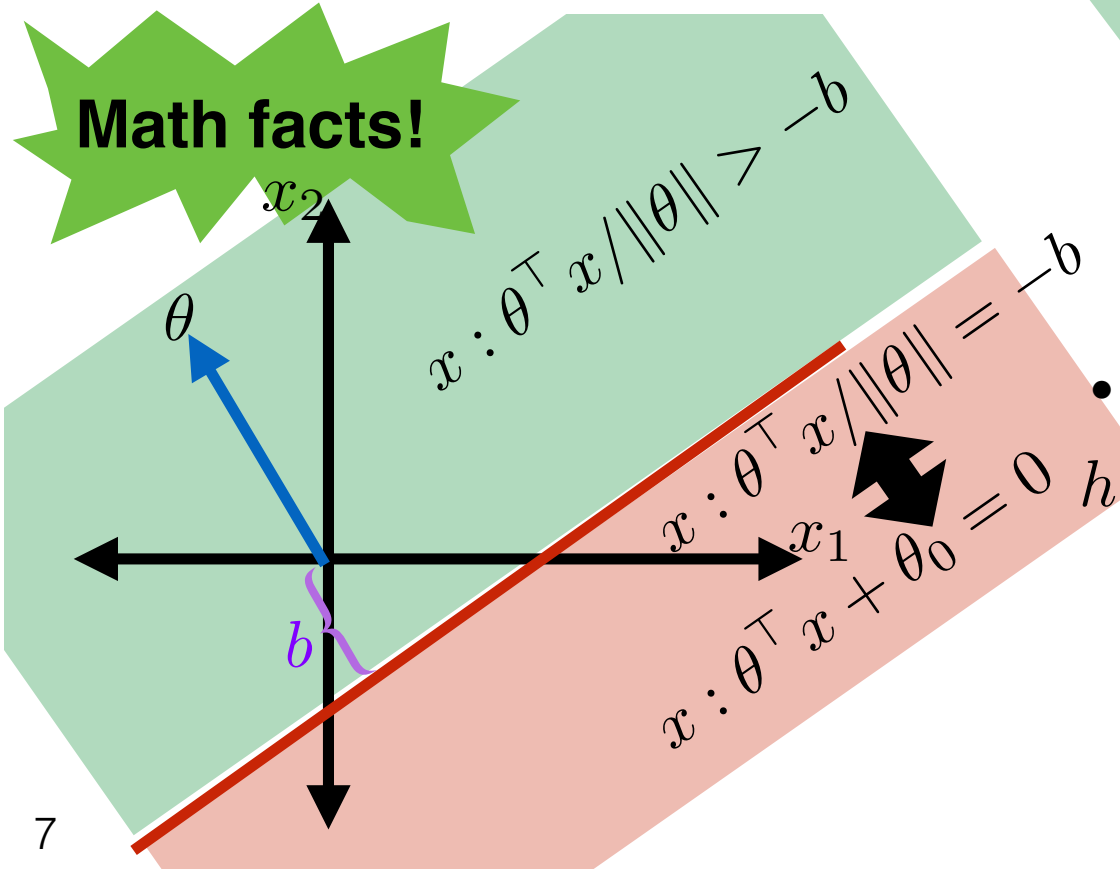
$$= \begin{cases} +1 \text{ if } \theta^\top x + \theta_0 > 0 \\ -1 \text{ if } \theta^\top x + \theta_0 \leq 0 \end{cases}$$

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!



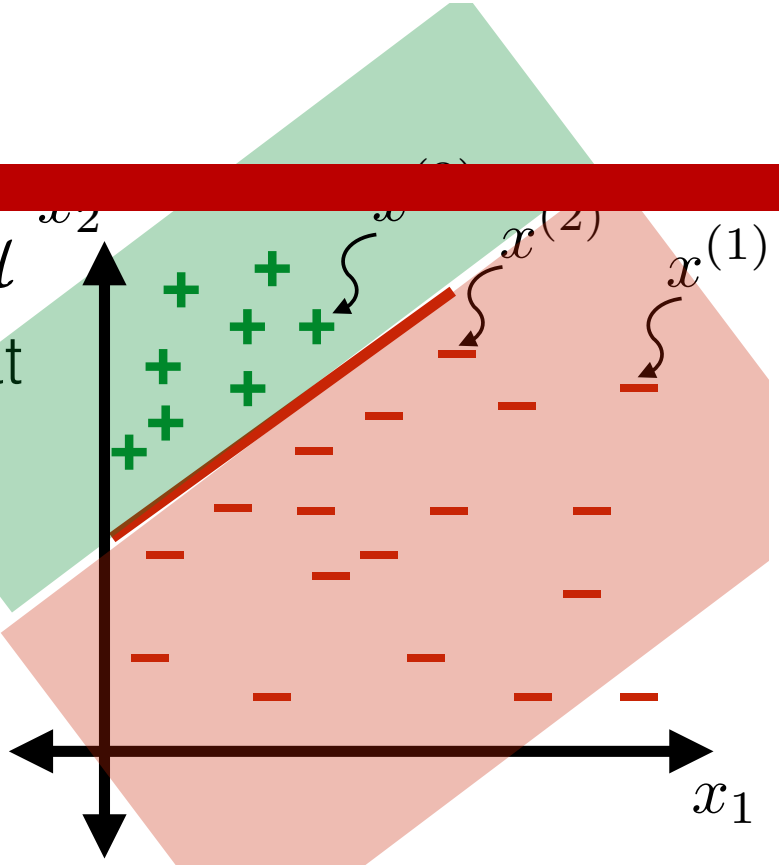
- Linear classifier:

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

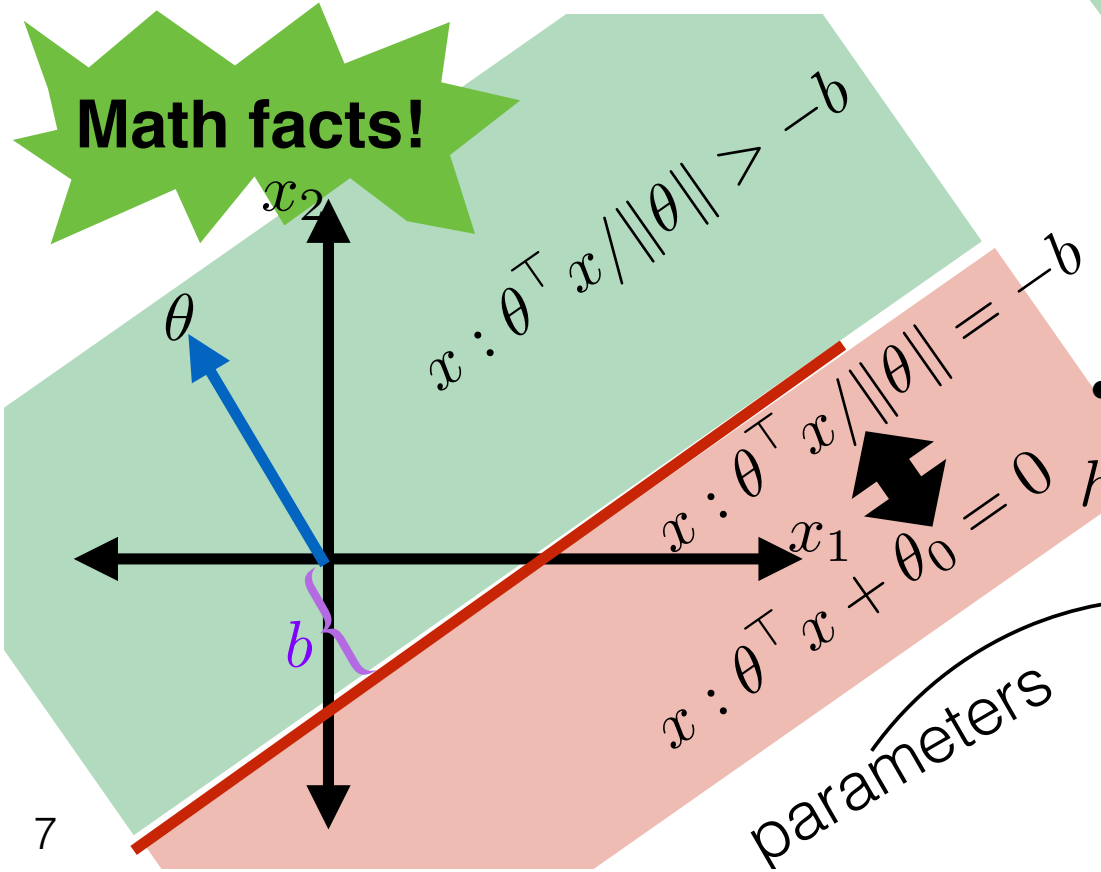
$$= \begin{cases} +1 \text{ if } \theta^\top x + \theta_0 > 0 \\ -1 \text{ if } \theta^\top x + \theta_0 \leq 0 \end{cases}$$

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:

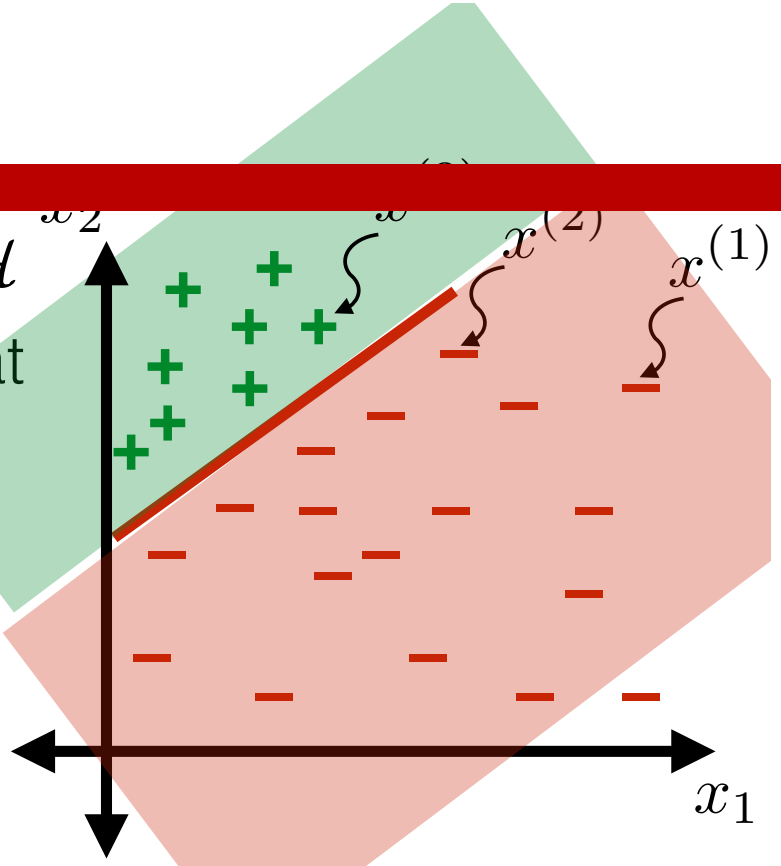
$$h(x; \theta, \theta_0) = \text{sign}(\theta^\top x + \theta_0)$$

parameters

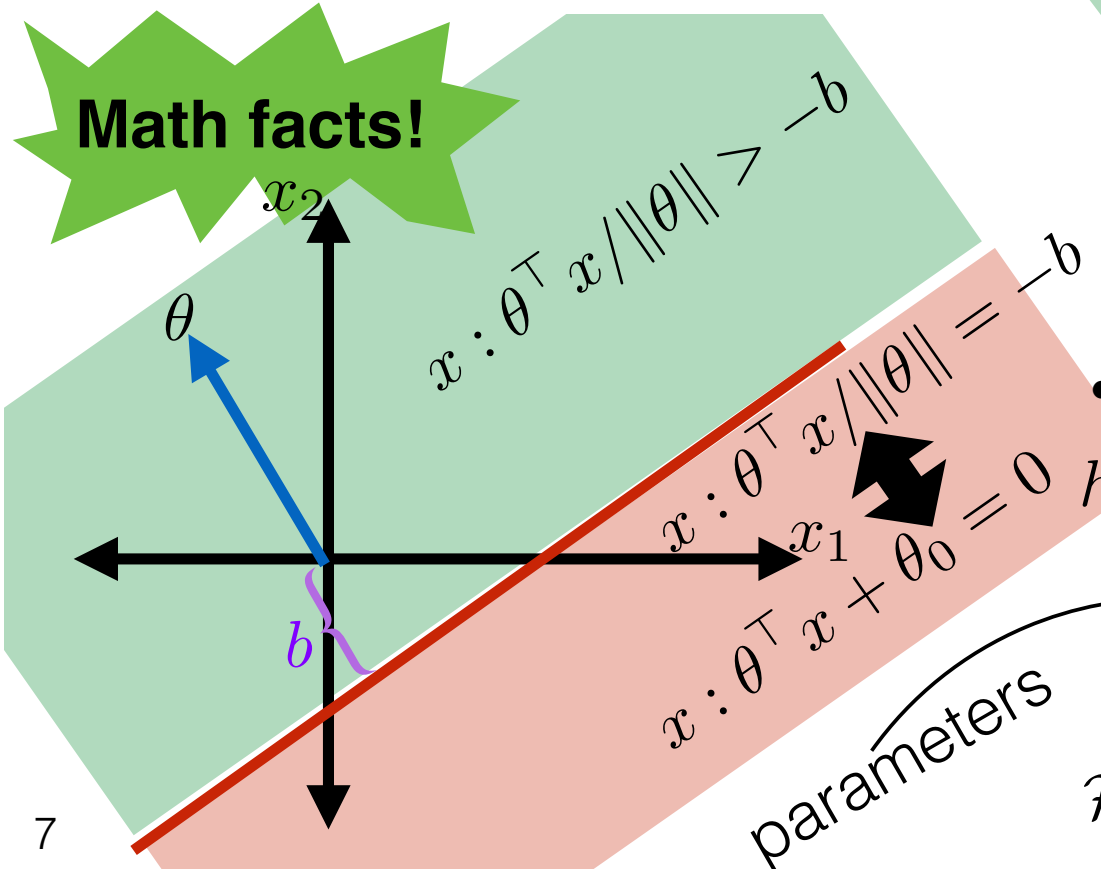
$$= \begin{cases} +1 \text{ if } \theta^\top x + \theta_0 > 0 \\ -1 \text{ if } \theta^\top x + \theta_0 \leq 0 \end{cases}$$

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:

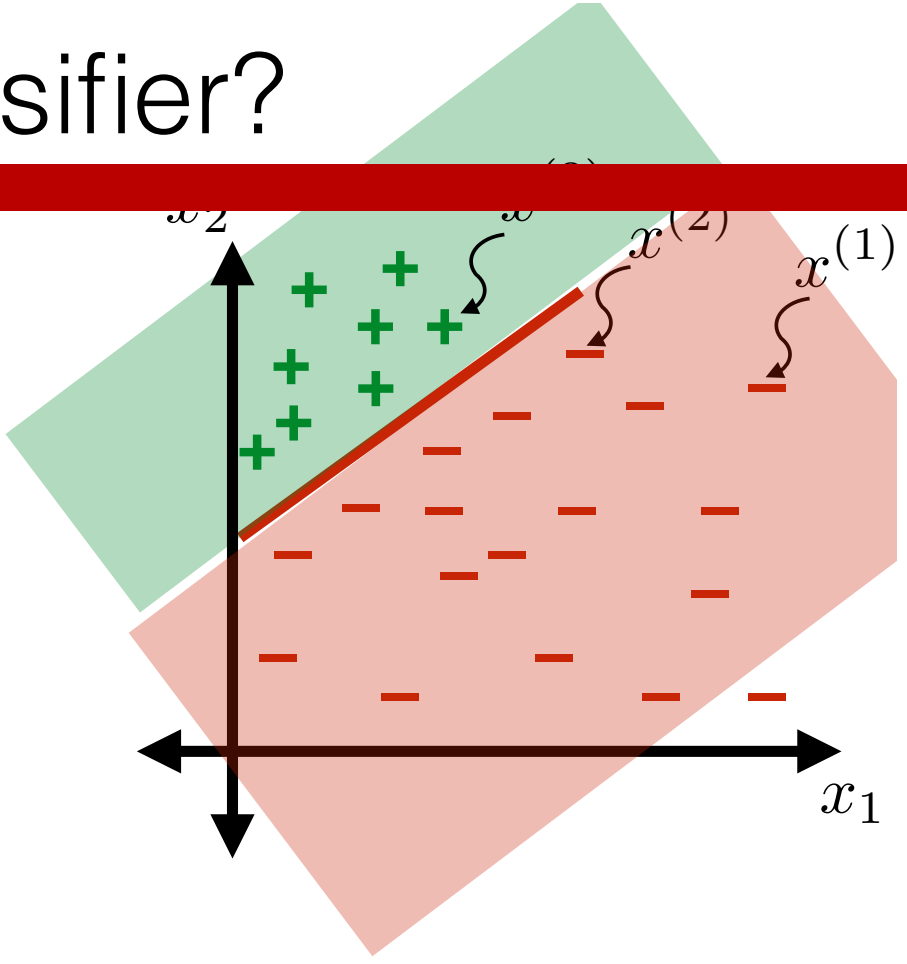
$$h(x; \theta, \theta_0) = \text{sign}(\theta^\top x + \theta_0)$$

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

parameters

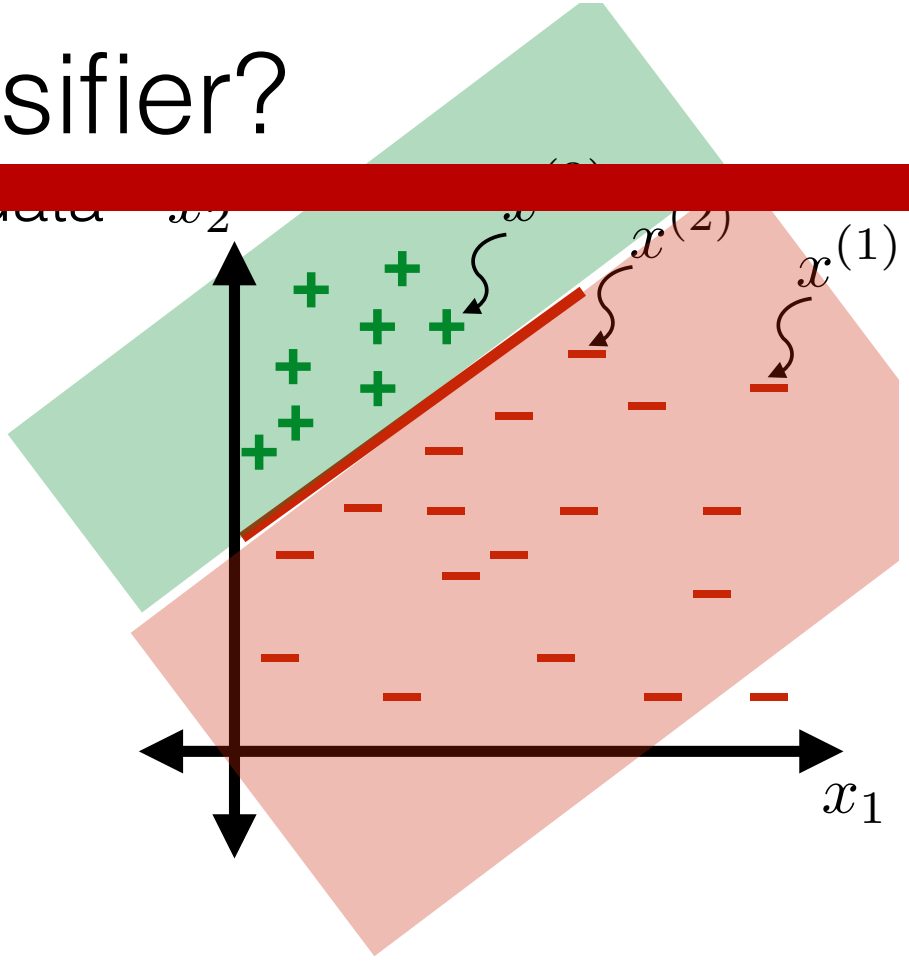
$$\mathcal{H} = \text{set of all such } h$$

How good is a classifier?



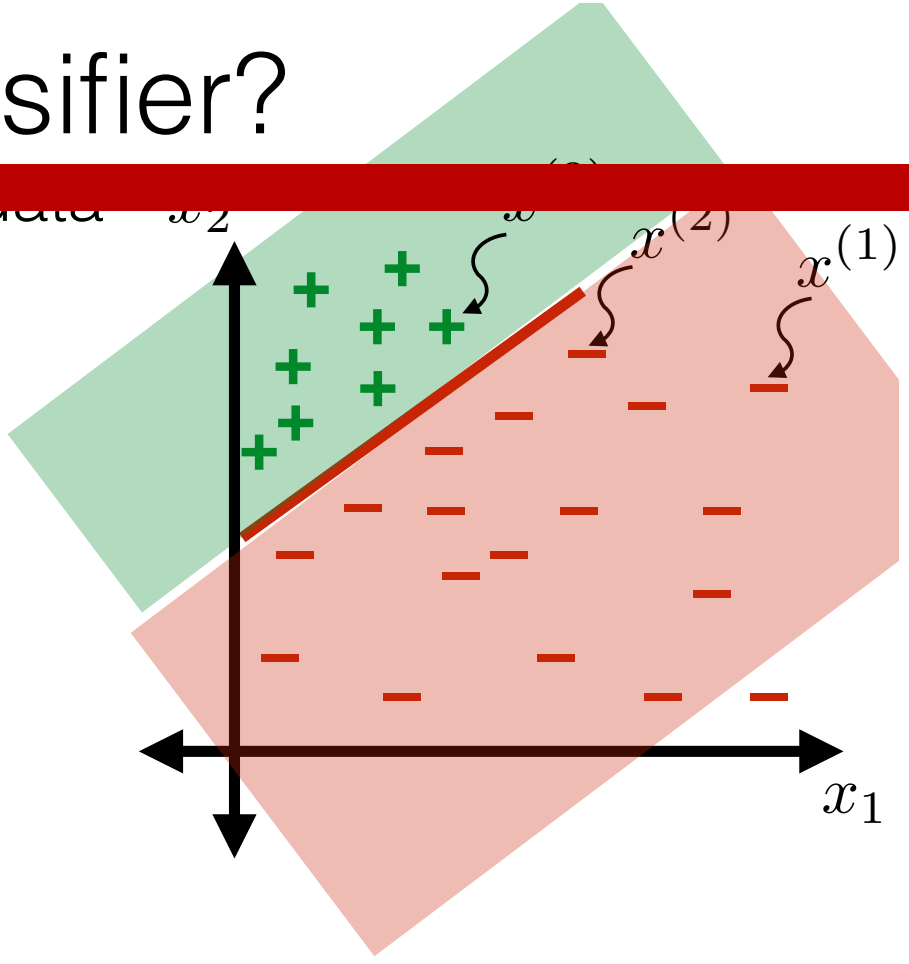
How good is a classifier?

Should predict well on future data



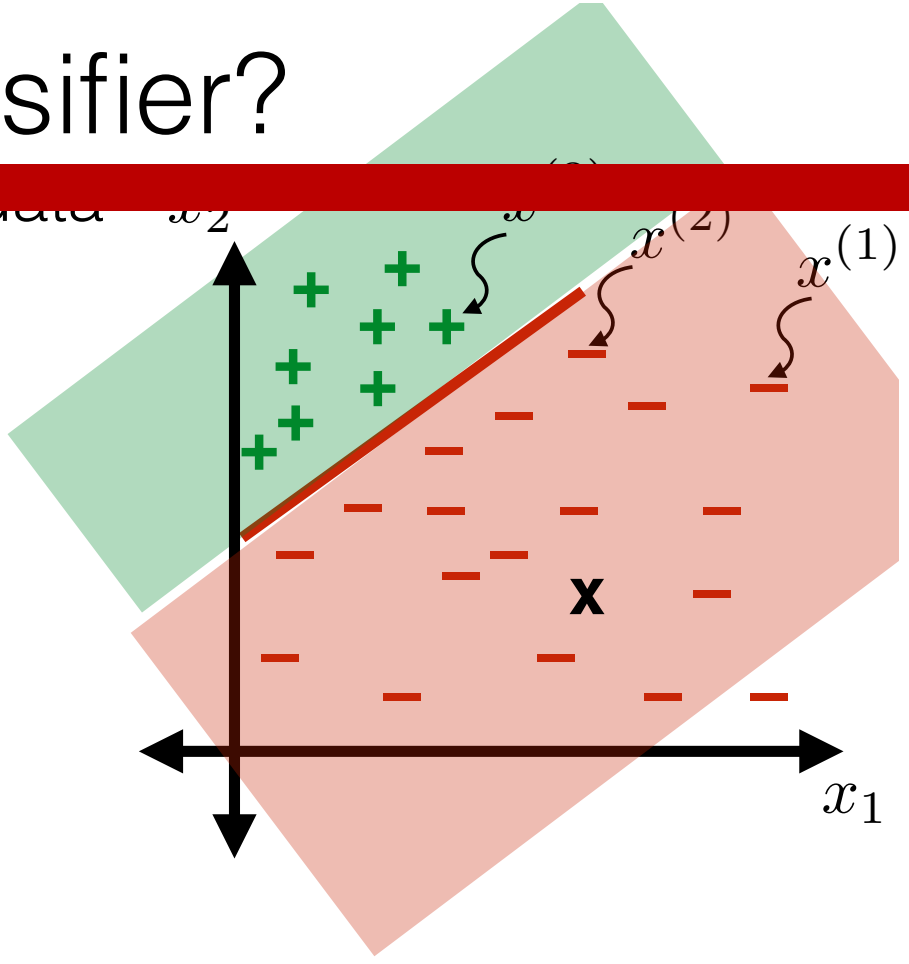
How good is a classifier?

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



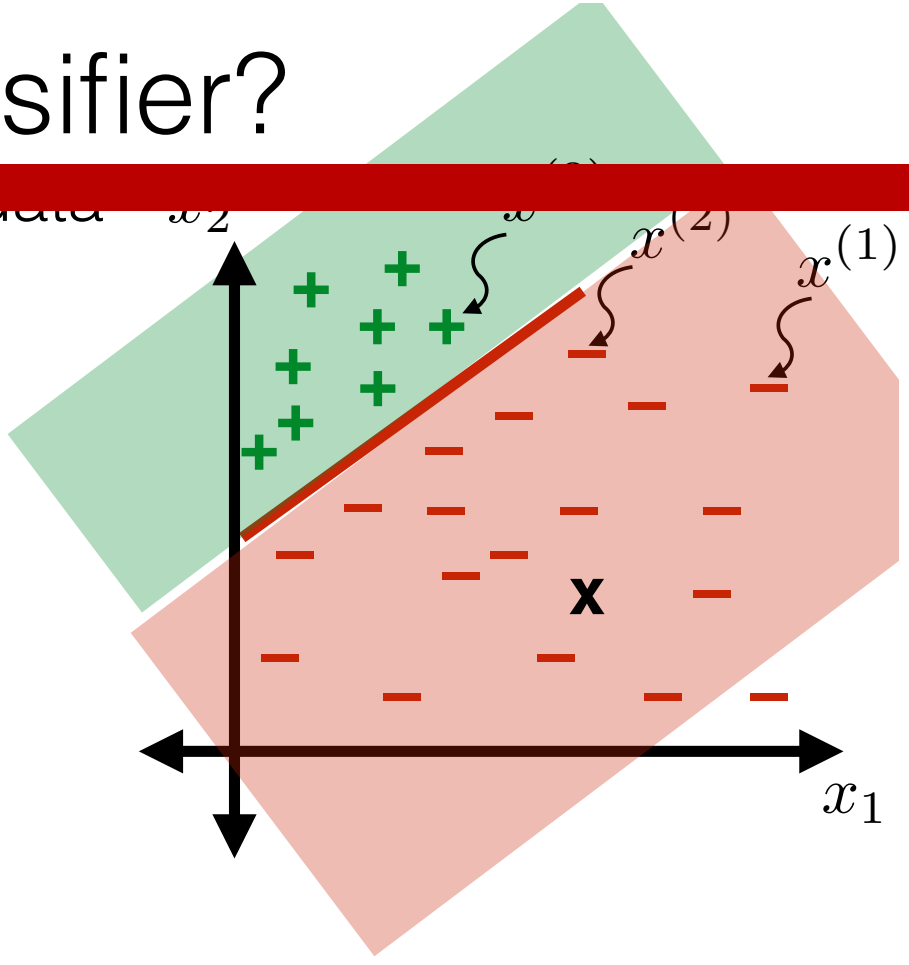
How good is a classifier?

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



How good is a classifier?

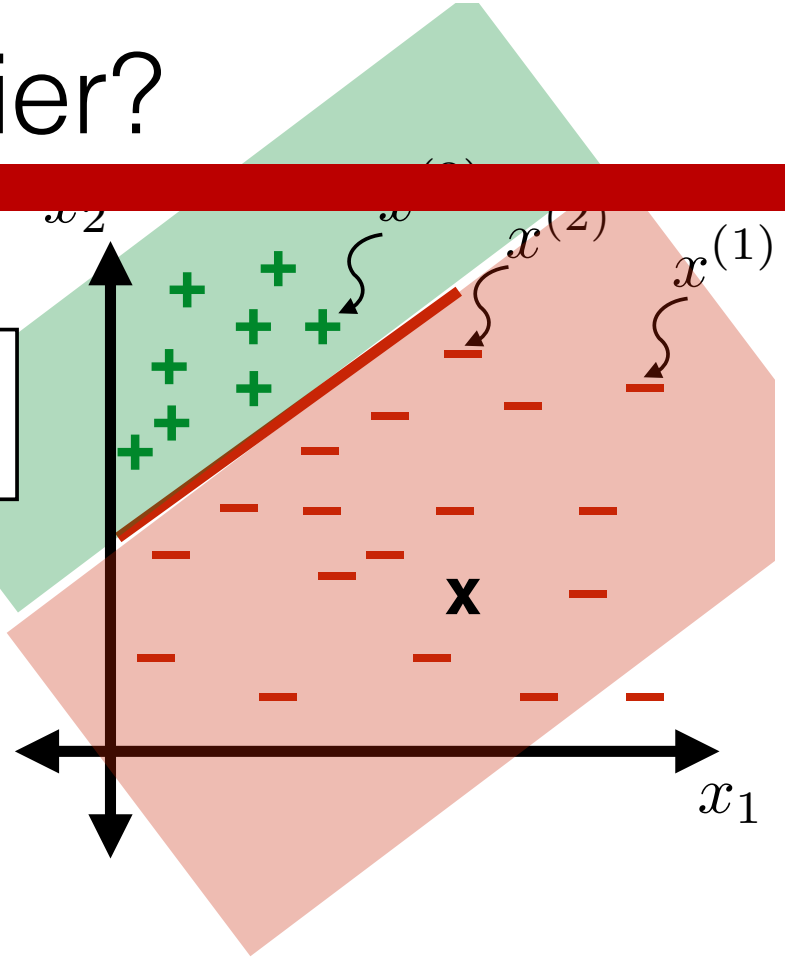
- Should predict well on future data
- How good is a classifier at a single point? Loss $L(g, a)$



How good is a classifier?

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

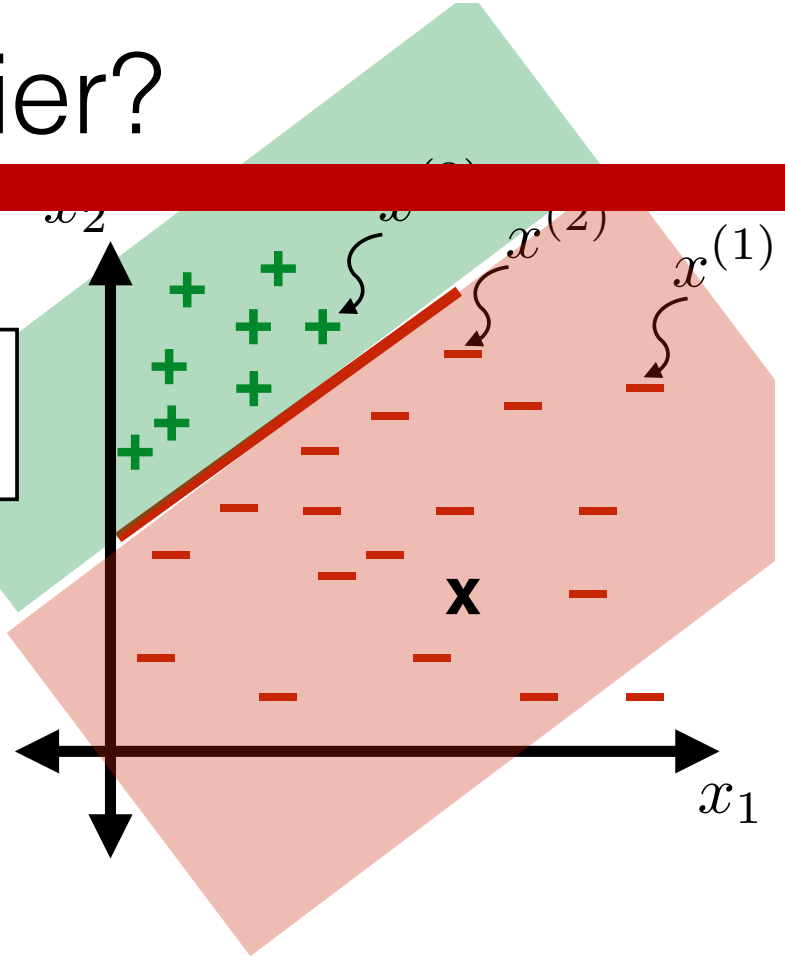
g: guess,
a: actual



How good is a classifier?

- How good is a classifier at a single point? Loss $L(g, a)$
  - Example: 0-1 loss

g: guess,
a: actual



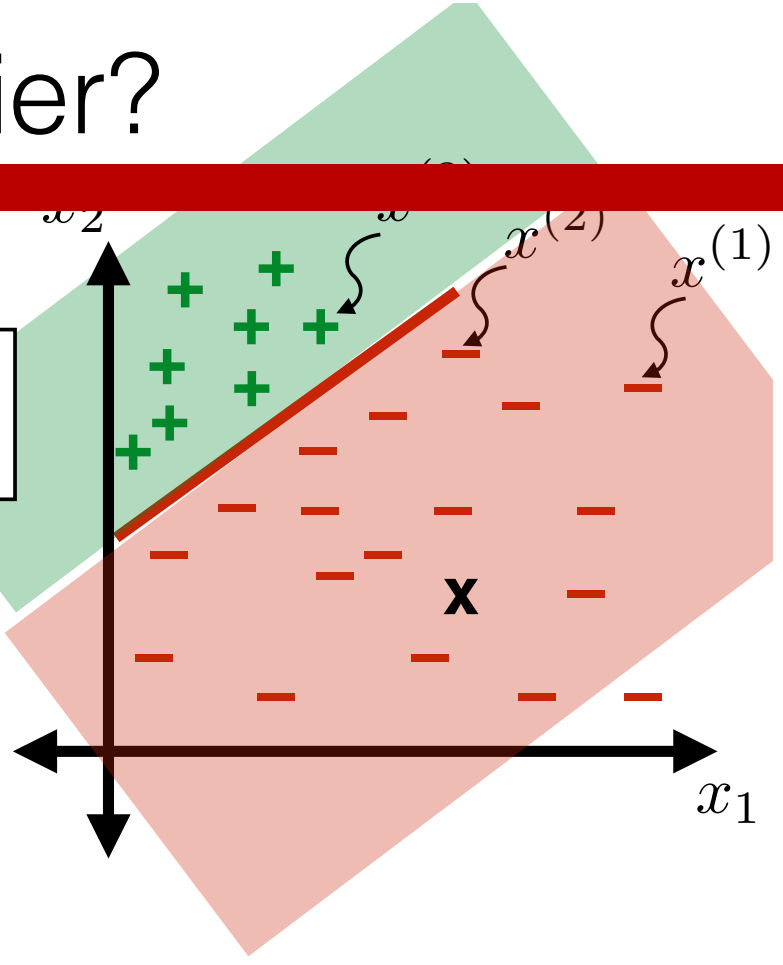
How good is a classifier?

- 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?

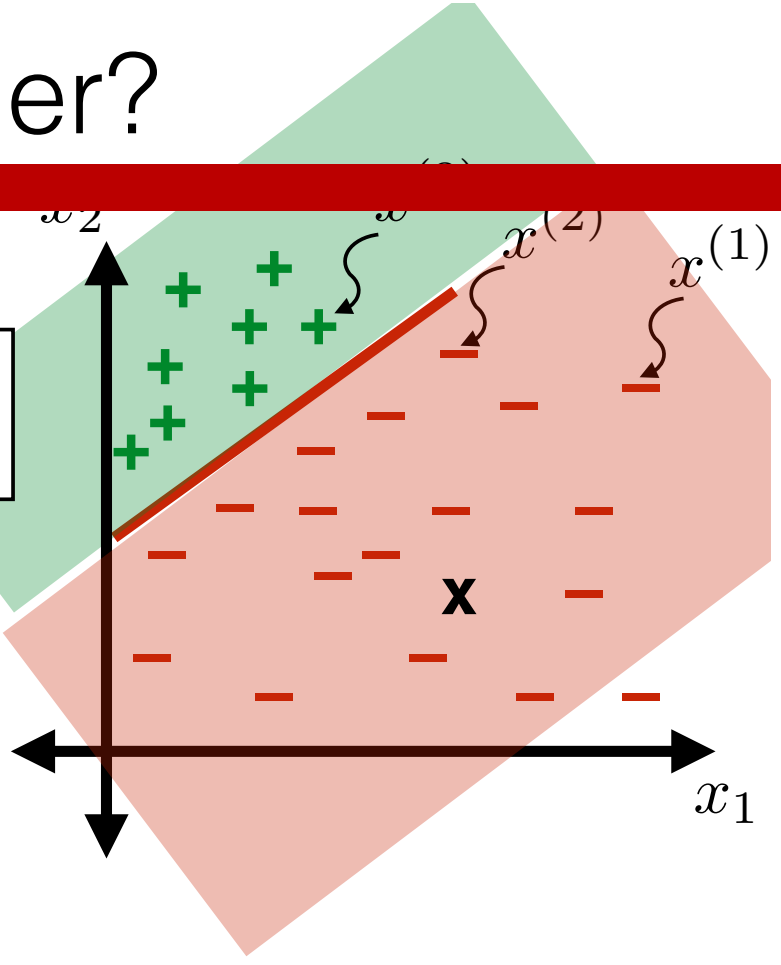
- How good is a classifier at a single point? Loss $L(g, a)$

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?

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

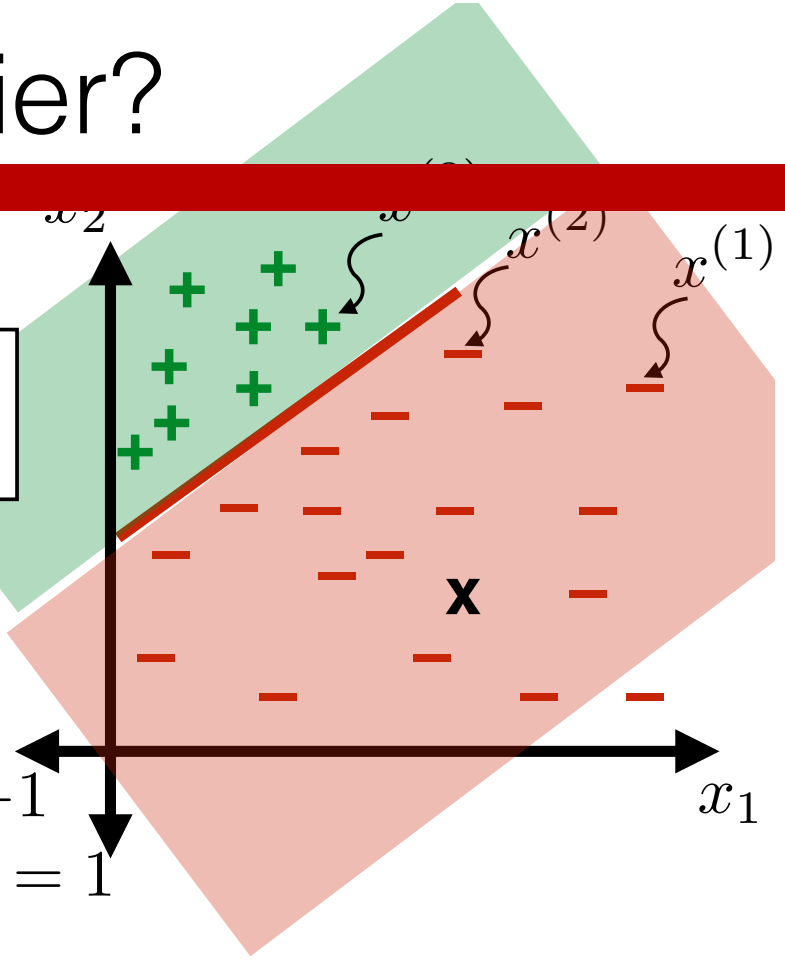
g: guess,
a: actual

  - Example: 0-1 loss

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

  - Example: asymmetric loss

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



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

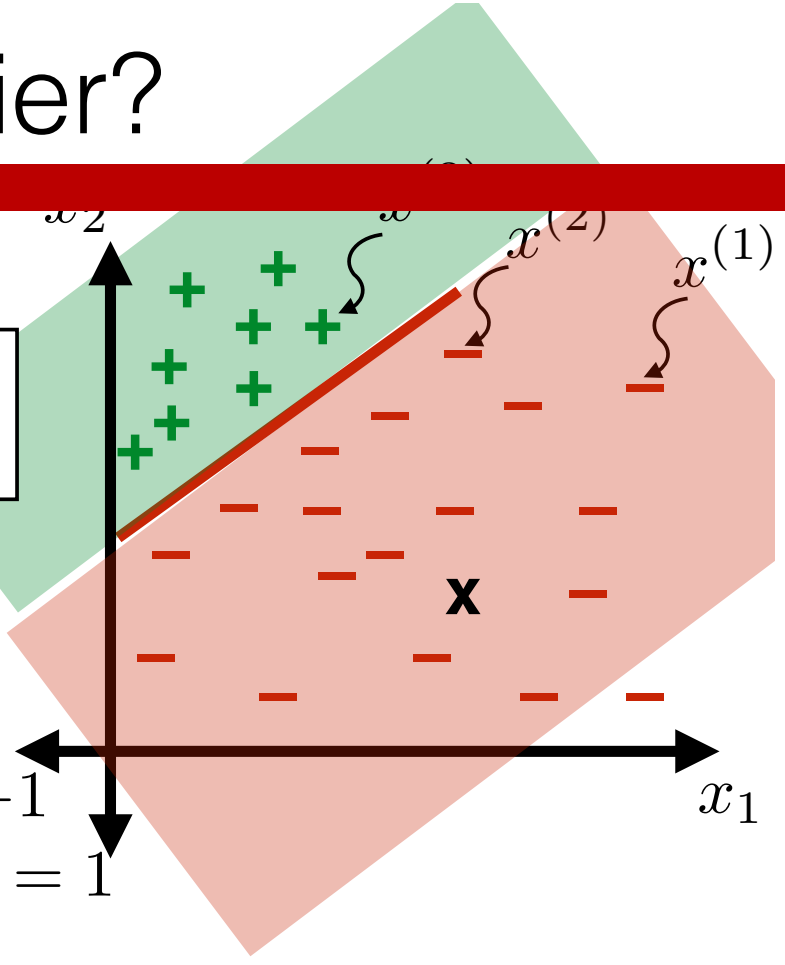
  - Example: 0-1 loss

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

  - 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?

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

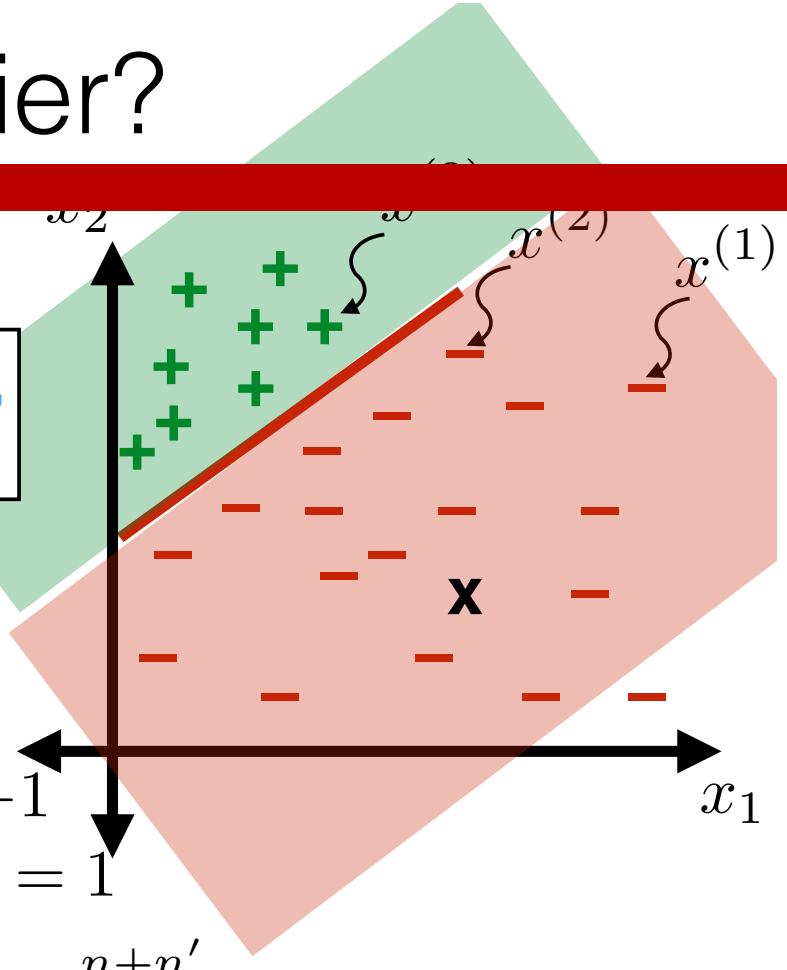
g: guess,
a: actual

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

How good is a classifier?

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

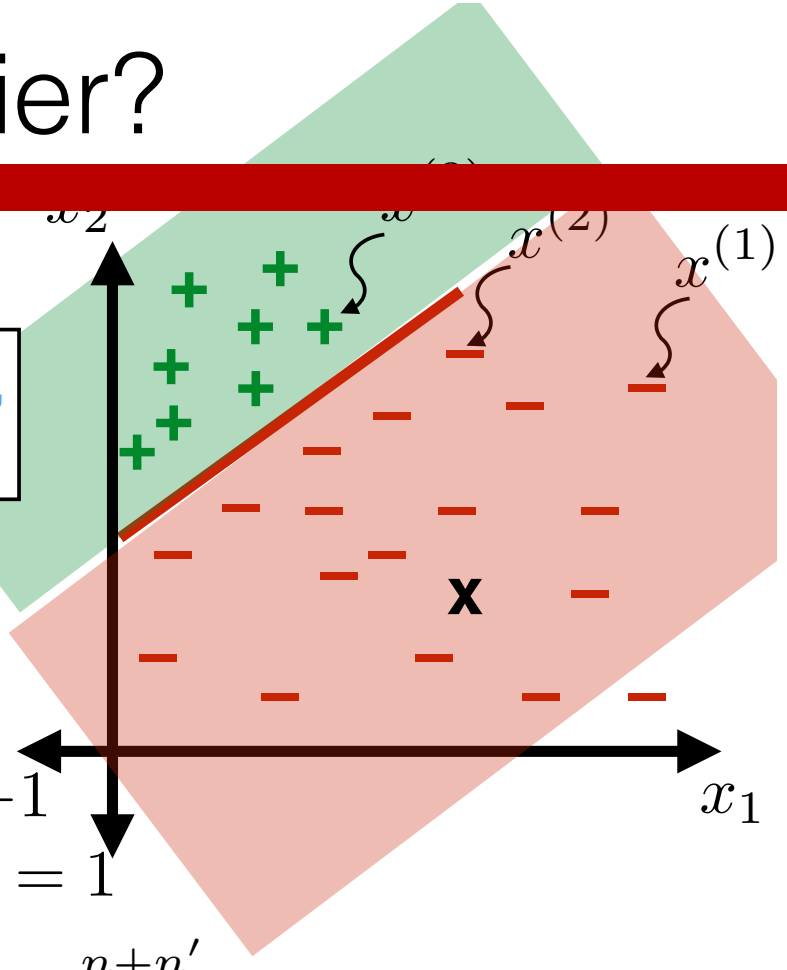
g: guess,
a: actual

  - Example: 0-1 loss

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

  - Example: asymmetric loss

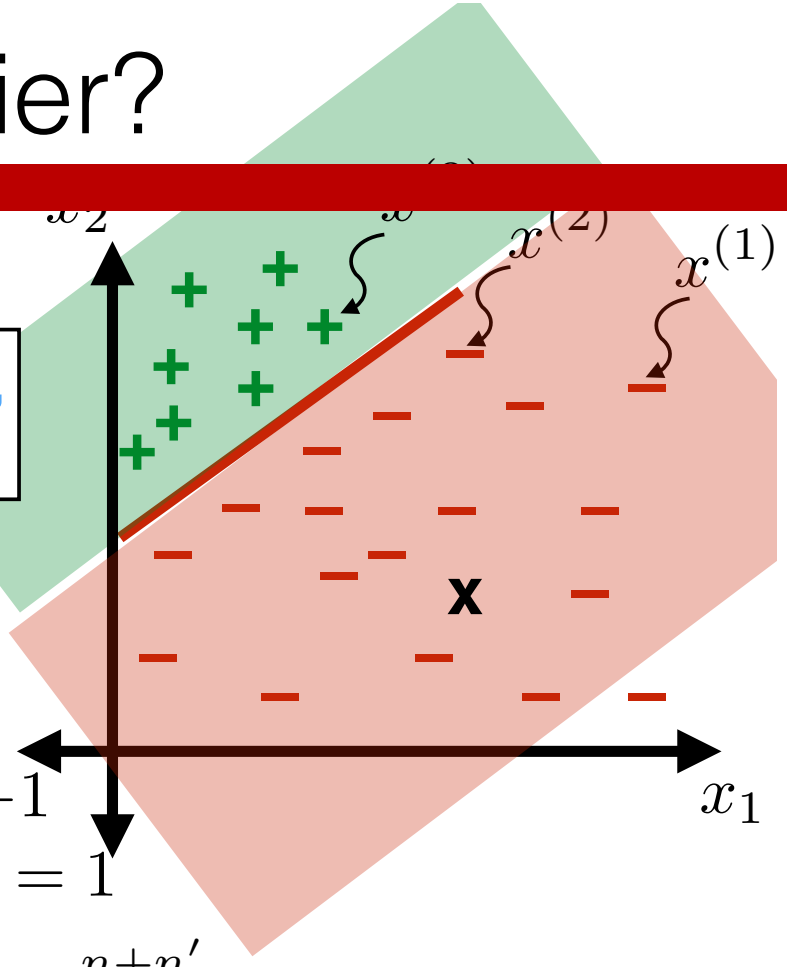
$$L(g,a) = \begin{cases} 1 \text{ if } g = 1, a = -1 \\ 100 \text{ if } g = -1, a = 1 \\ 0 \text{ else} \end{cases}$$



- 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:

How good is a classifier?



- 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: $\mathcal{E}_n(h) = \frac{1}{n} \sum_{i=1}^{n} L(h(x^{(i)}), y^{(i)})$

How good is a classifier?

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

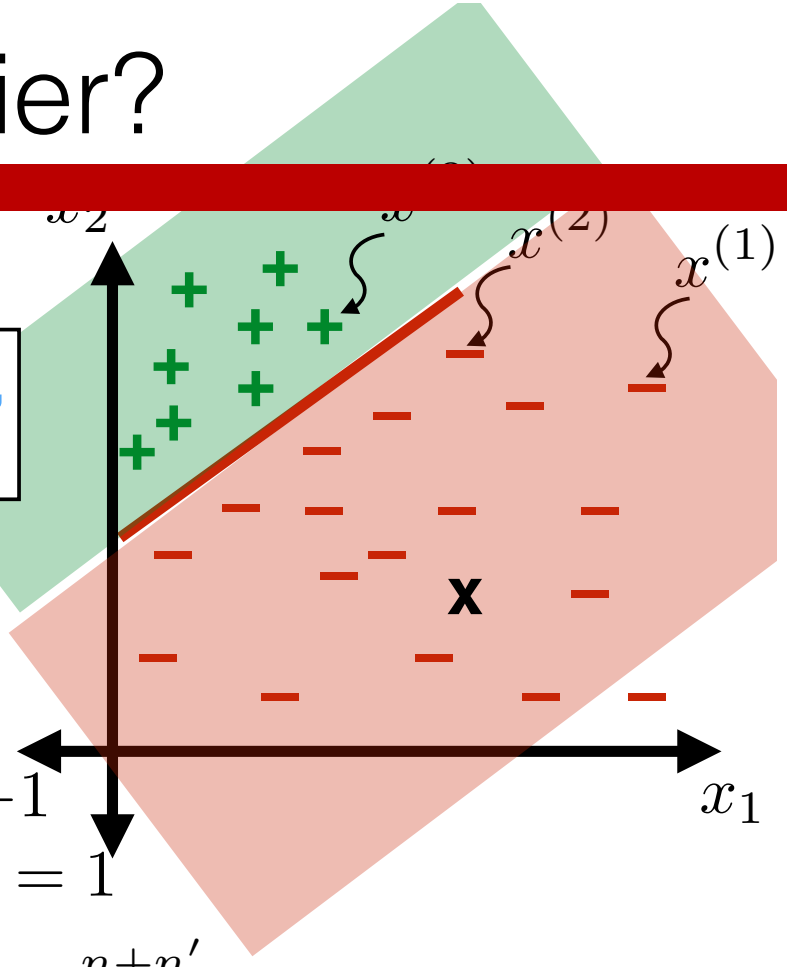
g: guess,
a: actual

  - Example: 0-1 loss

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

  - Example: asymmetric loss

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

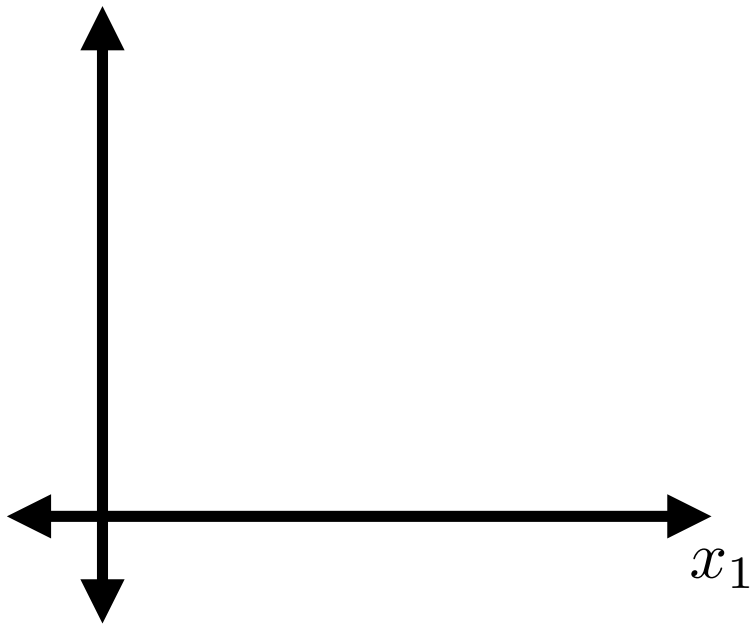


- Test error (*n'* new points): $\mathcal{E}(h) = \frac{1}{n'} \sum_{i=n+1}^{n+n'} L(h(x^{(i)}), y^{(i)})$
- Training error: $\mathcal{E}_n(h) = \frac{1}{n} \sum_{i=1}^{n} L(h(x^{(i)}), y^{(i)})$
- Prefer h to $\tilde{h}$ if $\mathcal{E}_n(h) < \mathcal{E}_n(\tilde{h})$

Learning a classifier

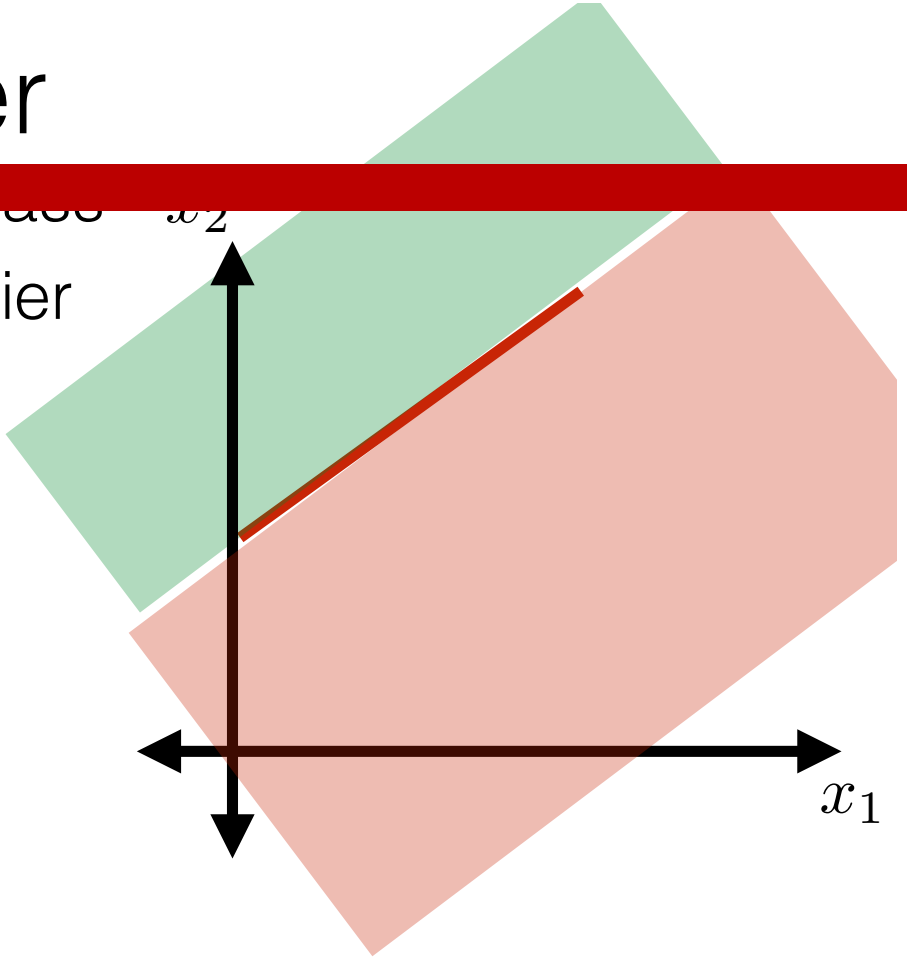
Have data, have hypothesis class

- Want to choose a good classifier
  - Recall: $x \rightarrow \boxed{h} \rightarrow y$



Learning a classifier

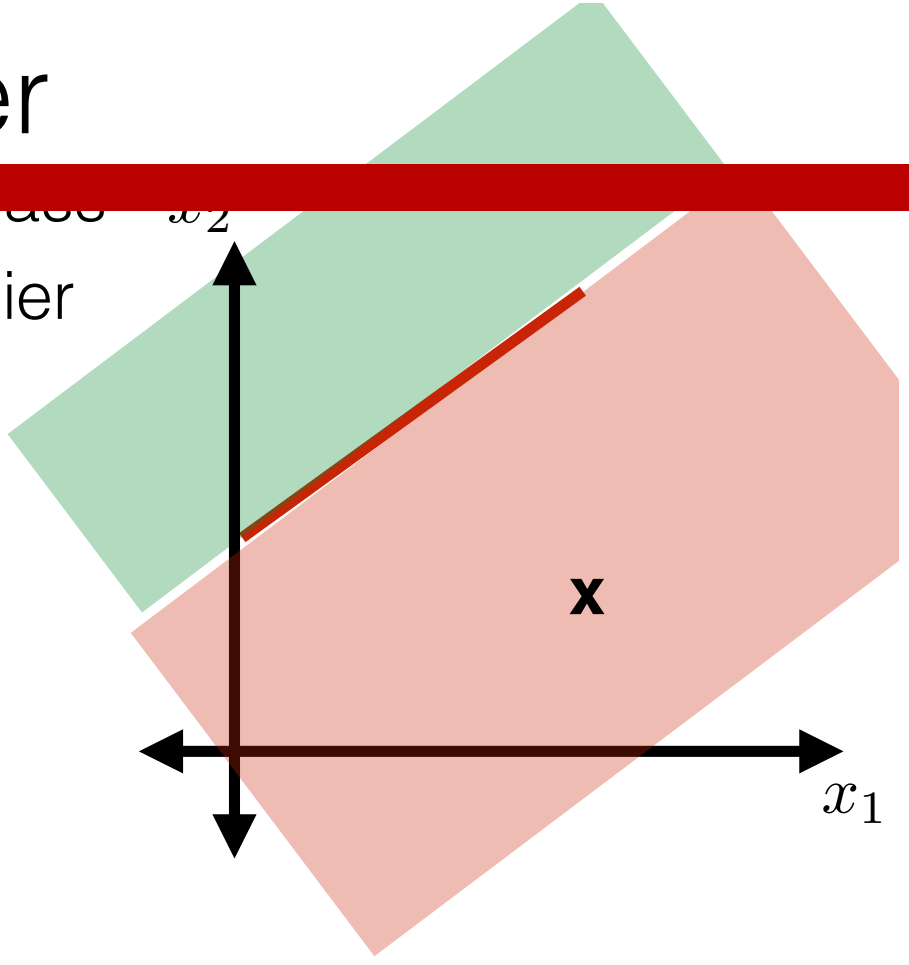
- Have data, have hypothesis class
- Want to choose a good classifier
  - Recall: $x \rightarrow \boxed{h} \rightarrow y$



Learning a classifier

Have data, have hypothesis class

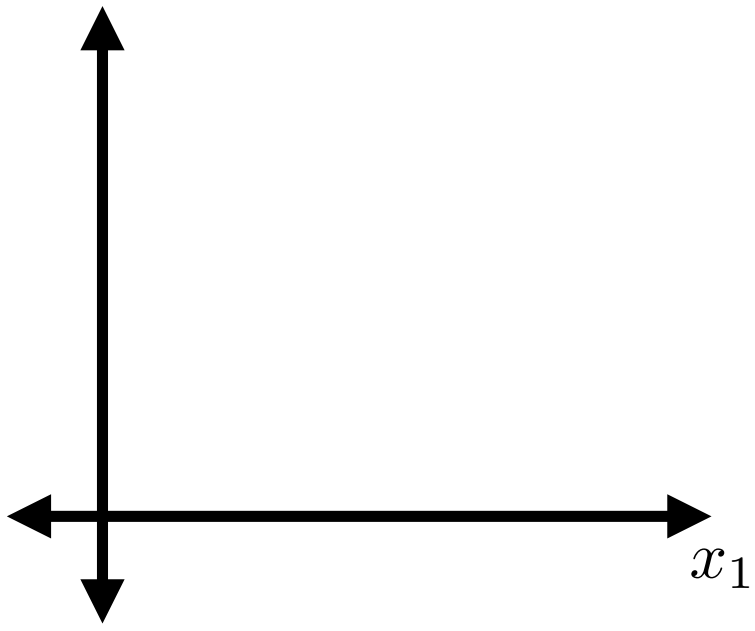
- Want to choose a good classifier
  - Recall: $x \rightarrow \boxed{h} \rightarrow y$



Learning a classifier

Have data, have hypothesis class

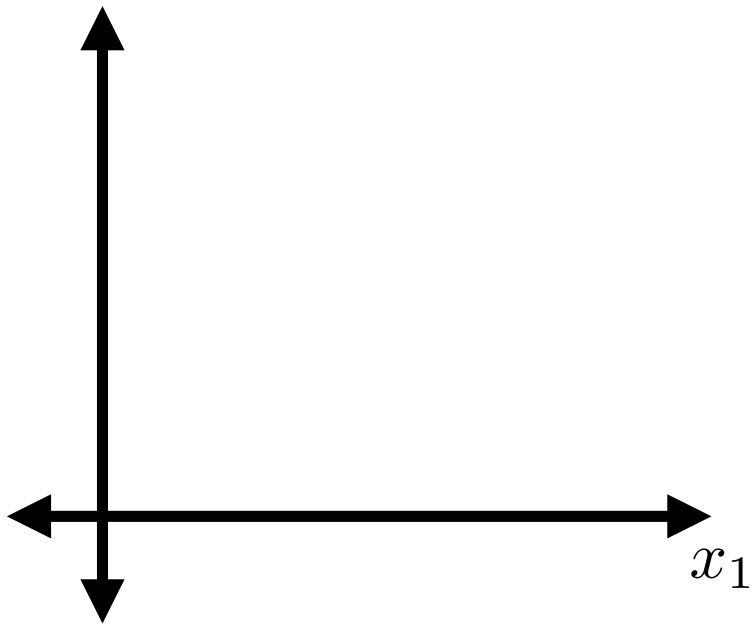
- Want to choose a good classifier
  - Recall: $x \rightarrow \boxed{h} \rightarrow y$



Learning a classifier

Have data, have hypothesis class

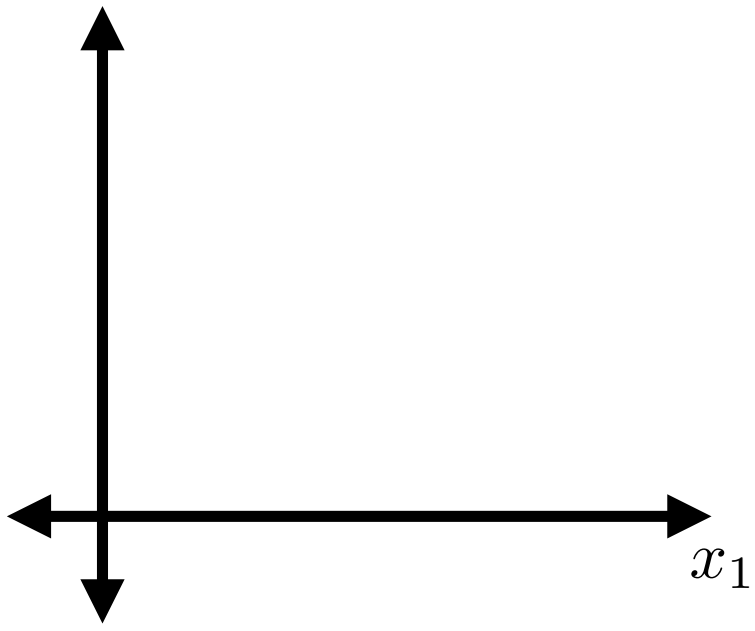
- Want to choose a good classifier
  - Recall: $x \rightarrow \boxed{h} \rightarrow y$
  - New:



Learning a classifier

Have data, have hypothesis class

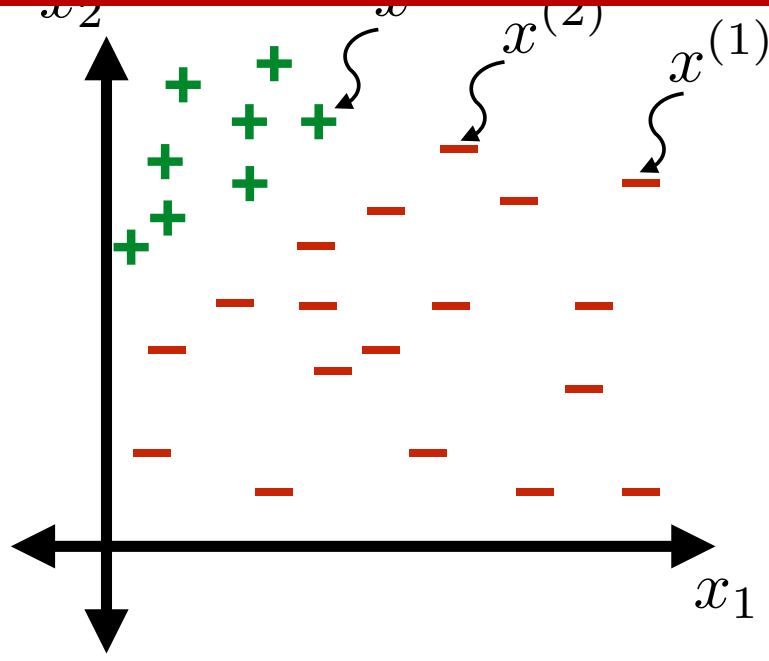
- Want to choose a good classifier
  - Recall: $x \rightarrow h \rightarrow y$
  - New: $\mathcal{D}_n \rightarrow$ learning algorithm $\rightarrow h$



Learning a classifier

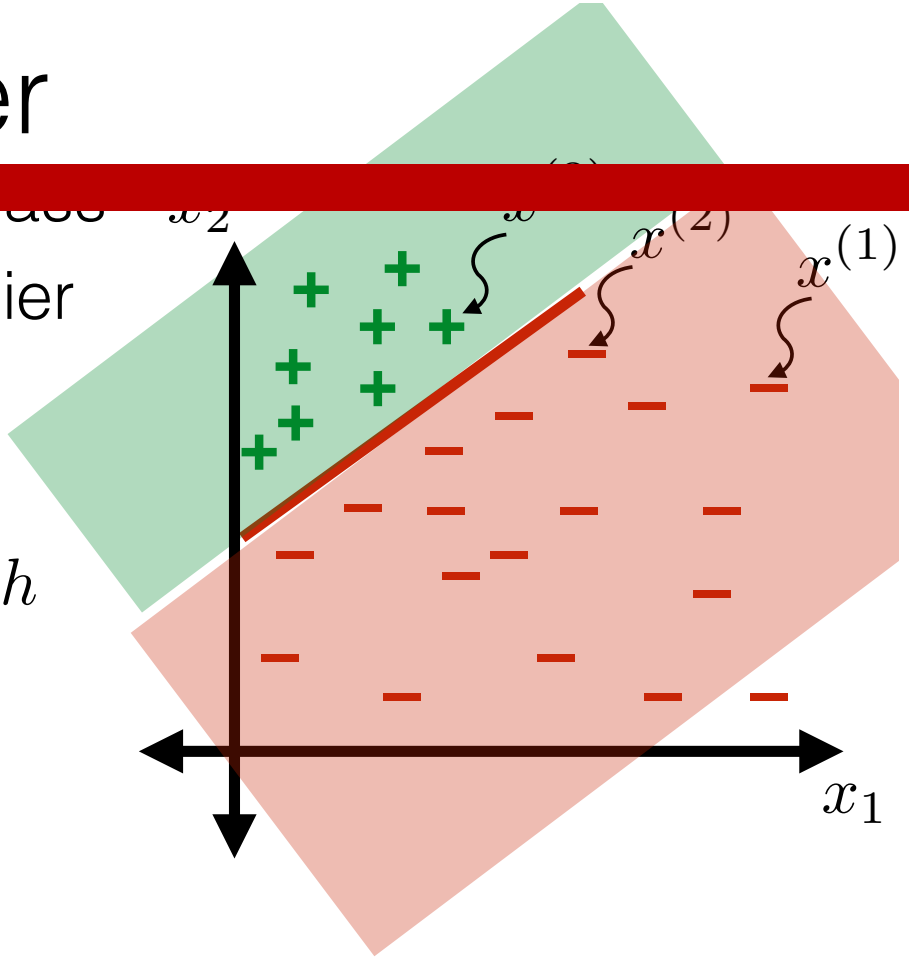
Have data, have hypothesis class

- Want to choose a good classifier
  - Recall: $x \rightarrow h \rightarrow y$
  - New: $\mathcal{D}_n \rightarrow$ learning algorithm $\rightarrow h$



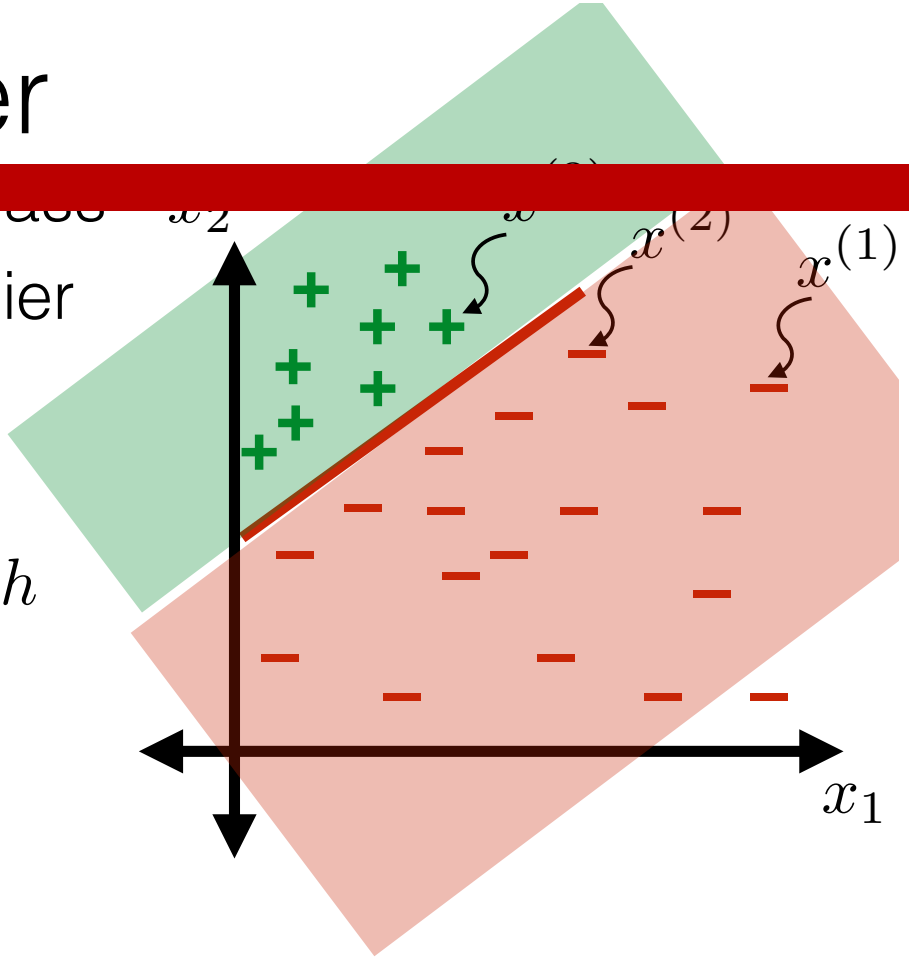
Learning a classifier

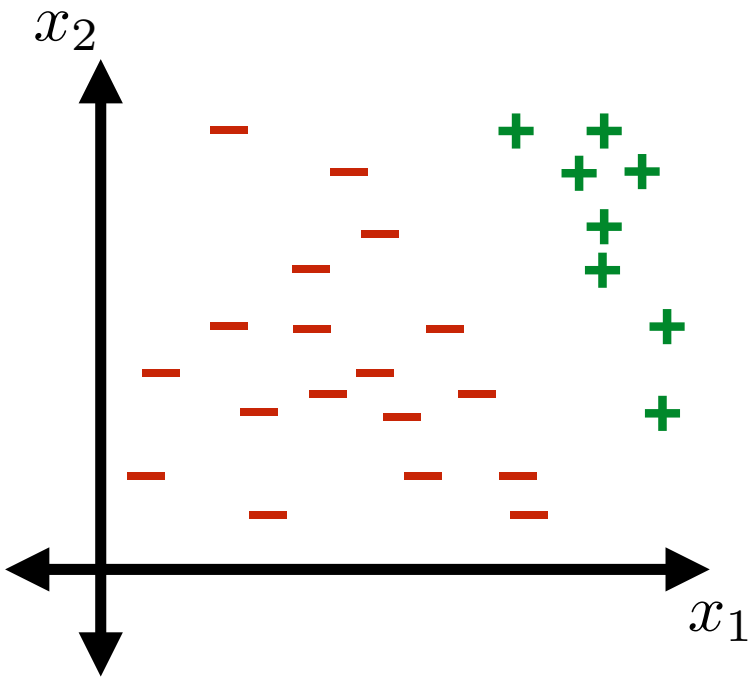
- Have data, have hypothesis class
- Want to choose a good classifier
  - Recall: $x \rightarrow h \rightarrow y$
  - New: $\mathcal{D}_n \rightarrow$ learning algorithm $\rightarrow h$



Learning a classifier

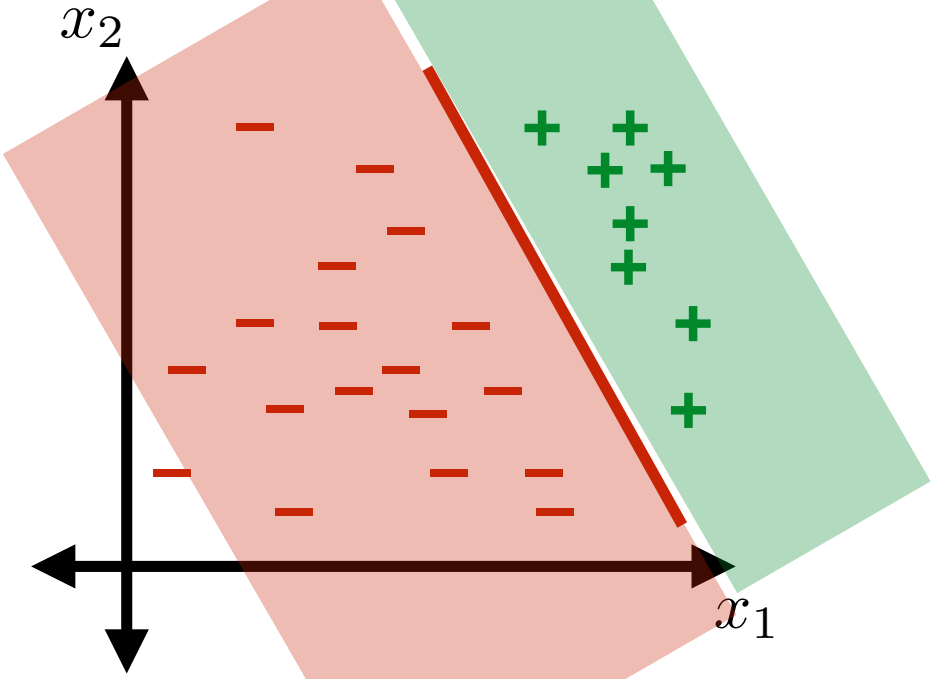
- Want to choose a good classifier
  - Recall: $x \rightarrow h \rightarrow y$
  - New: $\mathcal{D}_n \rightarrow$ learning algorithm $\rightarrow h$

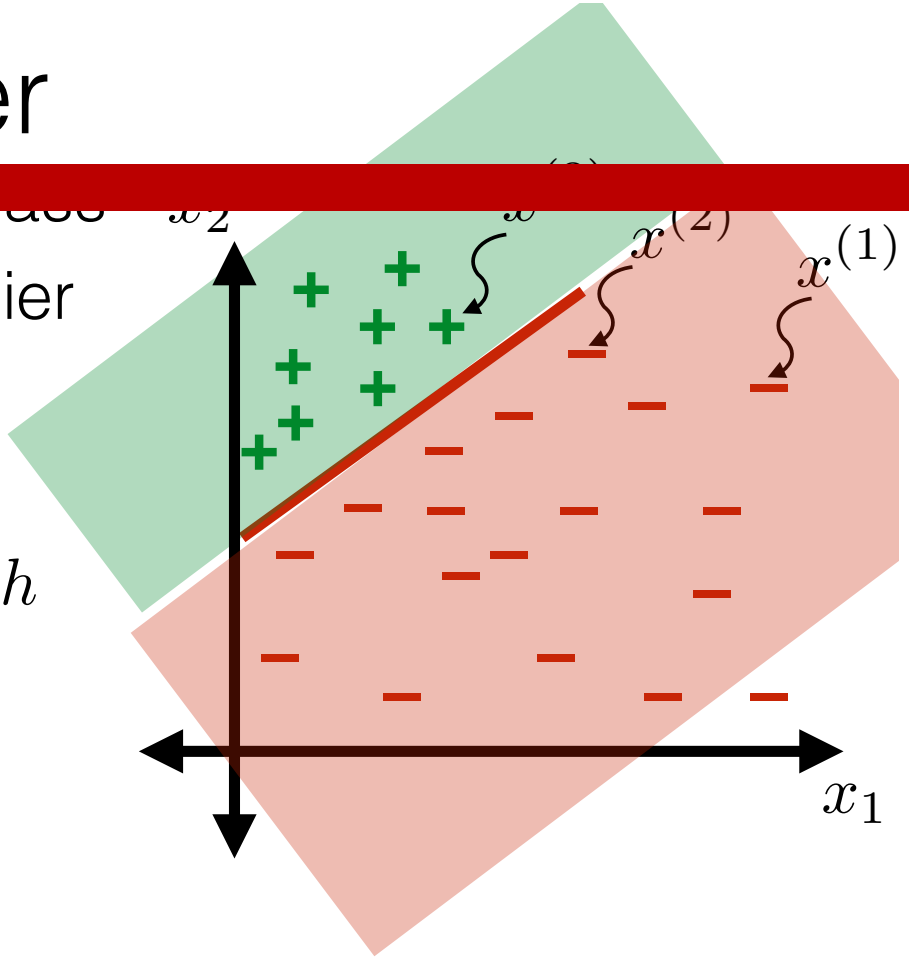




Learning a classifier

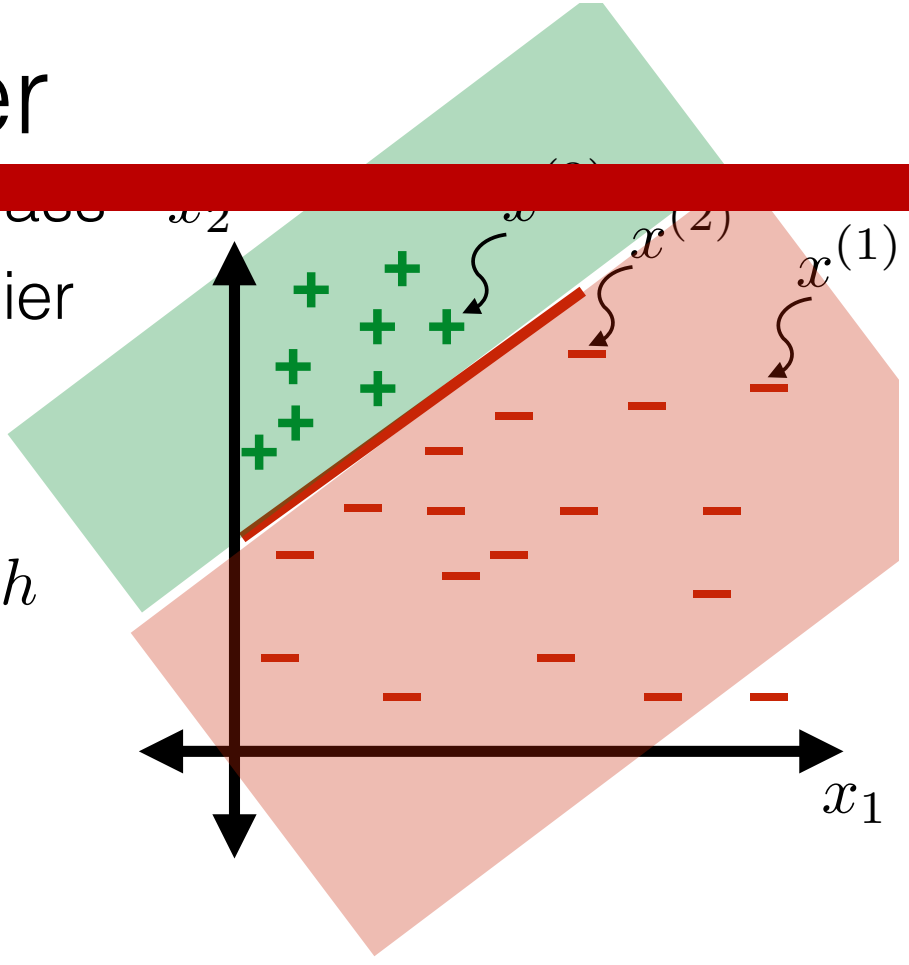
- Want to choose a good classifier
  - Recall: $x \rightarrow h \rightarrow y$
  - New: $\mathcal{D}_n \rightarrow$ learning algorithm $\rightarrow h$





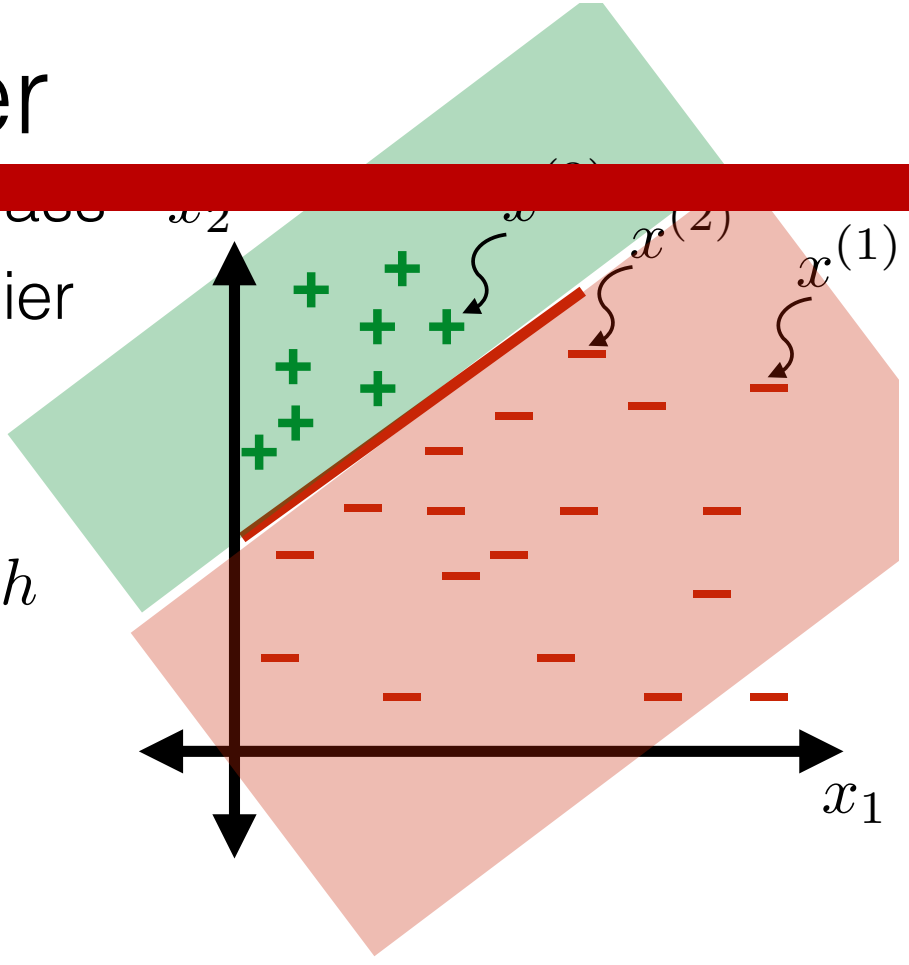
Learning a classifier

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



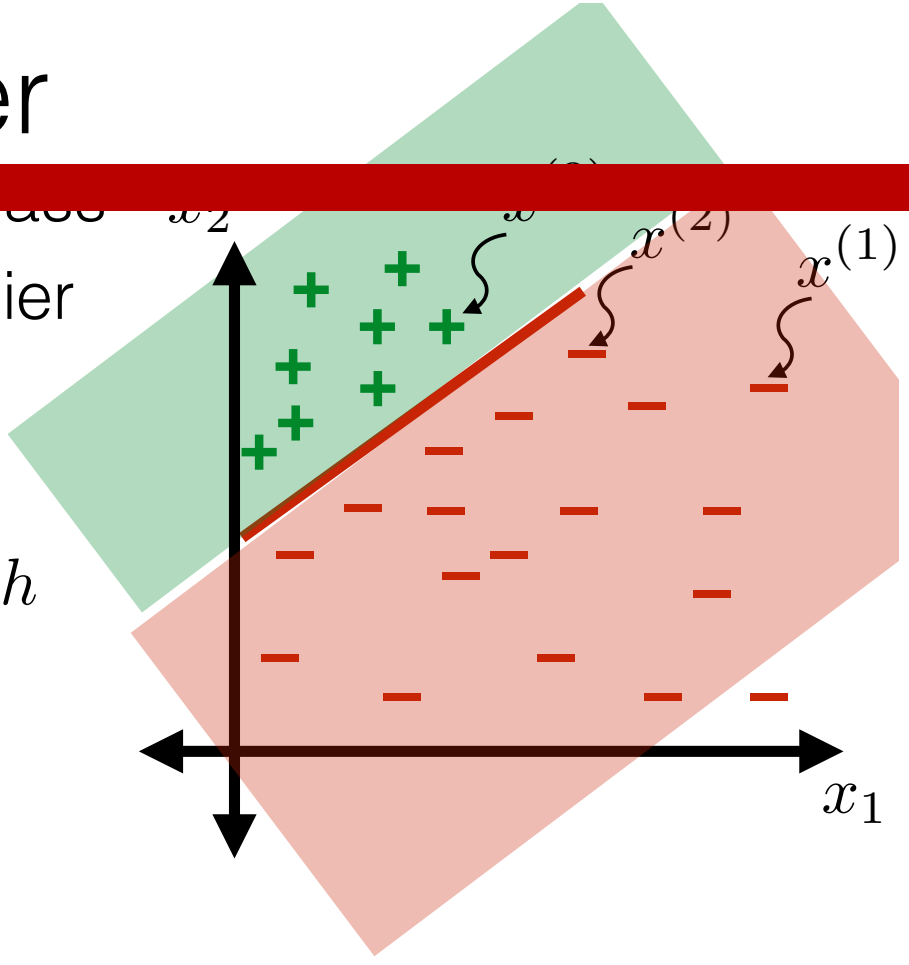
Learning a classifier

- 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:



Learning a classifier

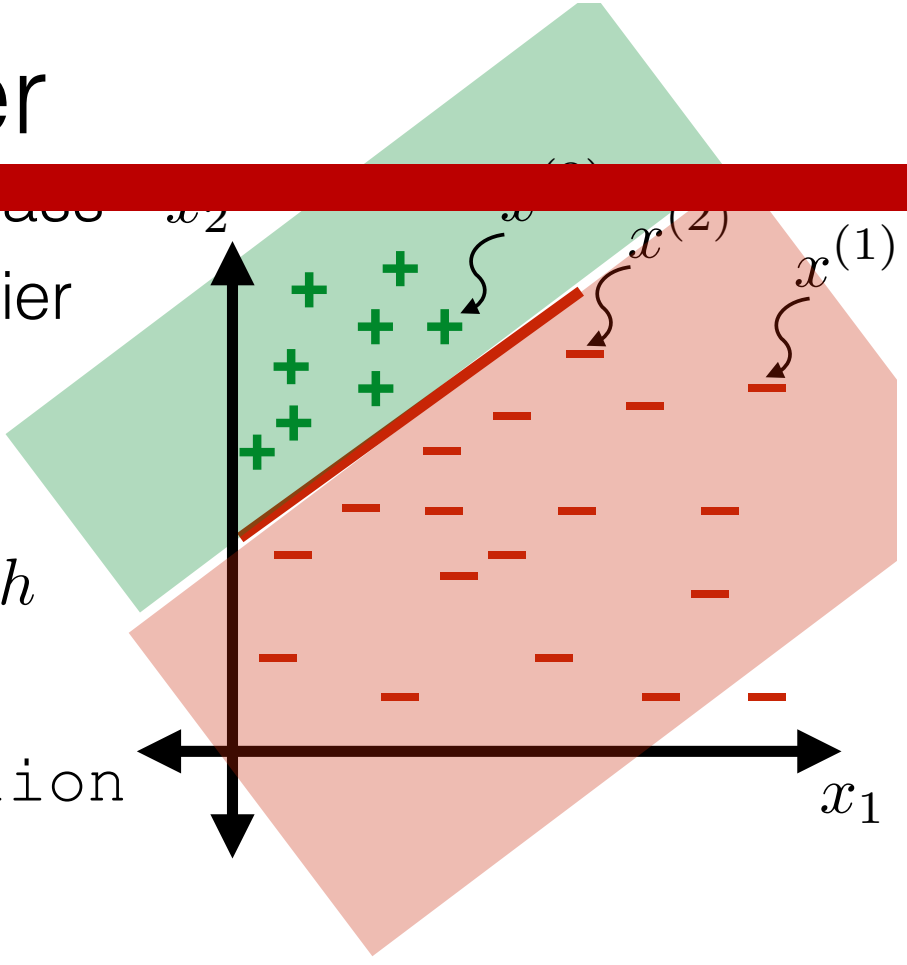
- Have data, have hypothesis class
- Want to choose a good classifier
  - Recall: $x \rightarrow h \rightarrow y$
  - New: $\mathcal{D}_n \rightarrow$ learning algorithm $\rightarrow h$
- Example:



Learning a classifier

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

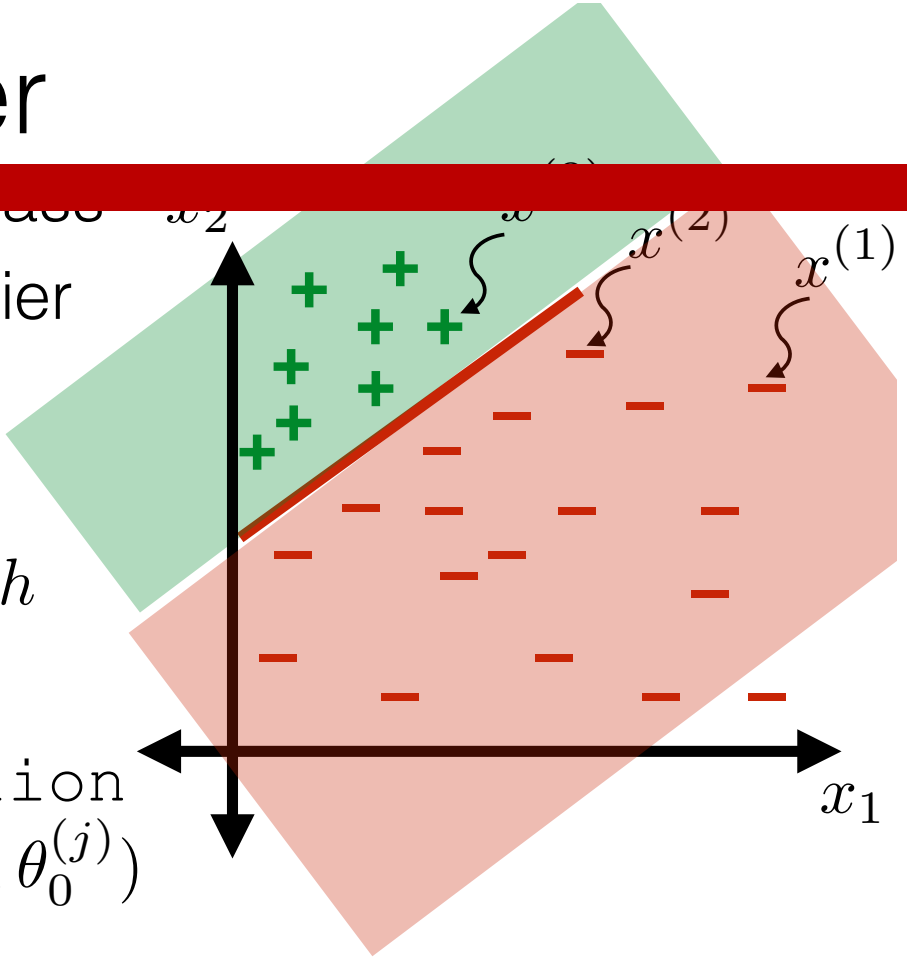
```
for j = 1, …, 1 trillion
```



Learning a classifier

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

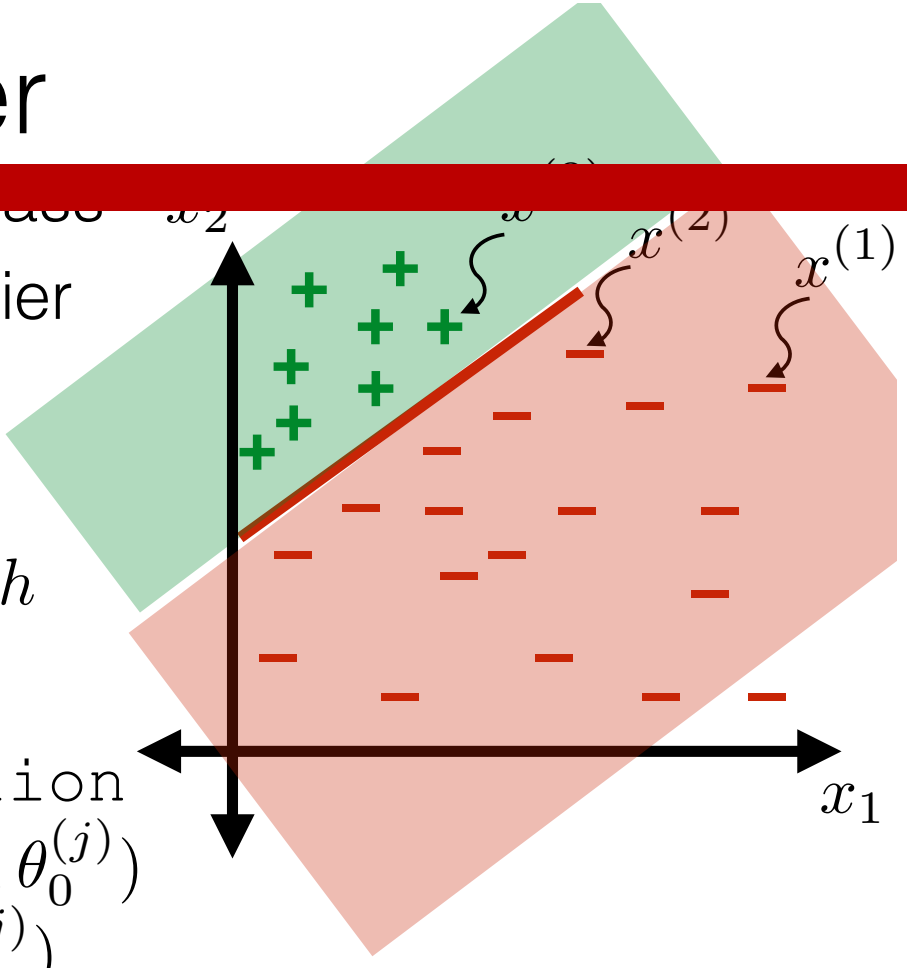
```
for j = 1, …, 1 trillion
  Randomly sample (θ^(j), θ_0^(j))
```



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:

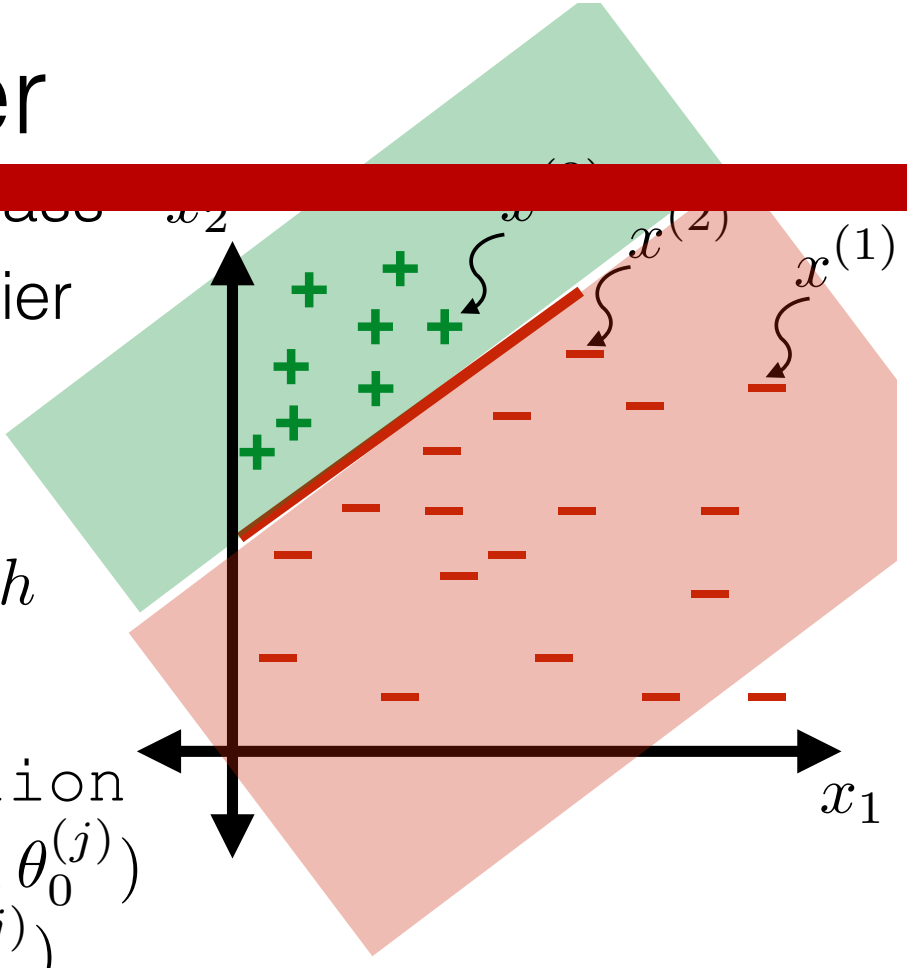
```
for j = 1, …, 1 trillion
  Randomly sample (θ^(j), θ_0^(j))
  Set h^(j)(x) = h(x; θ^(j), θ_0^(j))
```



Learning a classifier

- Want to choose a good classifier
  - Recall: $x \rightarrow h \rightarrow y$
  - New: $\mathcal{D}_n \rightarrow$ learning algorithm $\rightarrow h$
- Example:

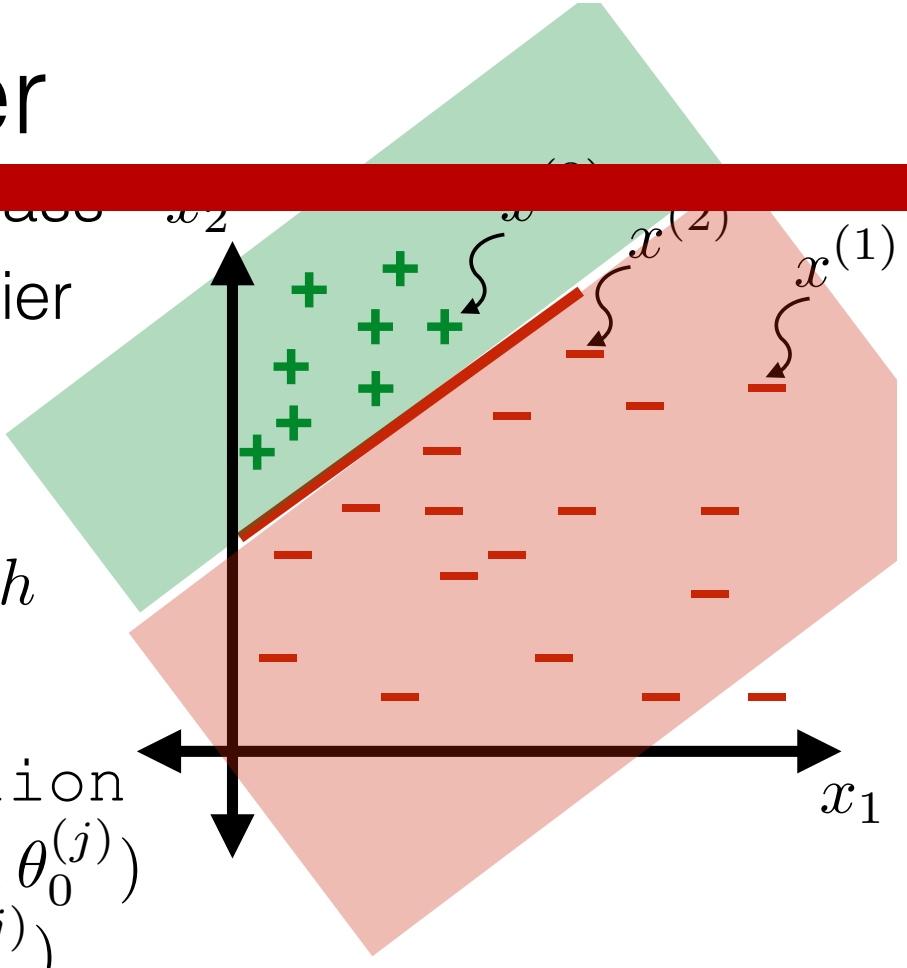
```
for j = 1, …, 1 trillion
  Randomly sample (θ^(j), θ_0^(j))
  Set h^(j)(x) = h(x; θ^(j), θ_0^(j))
```



Learning a classifier

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

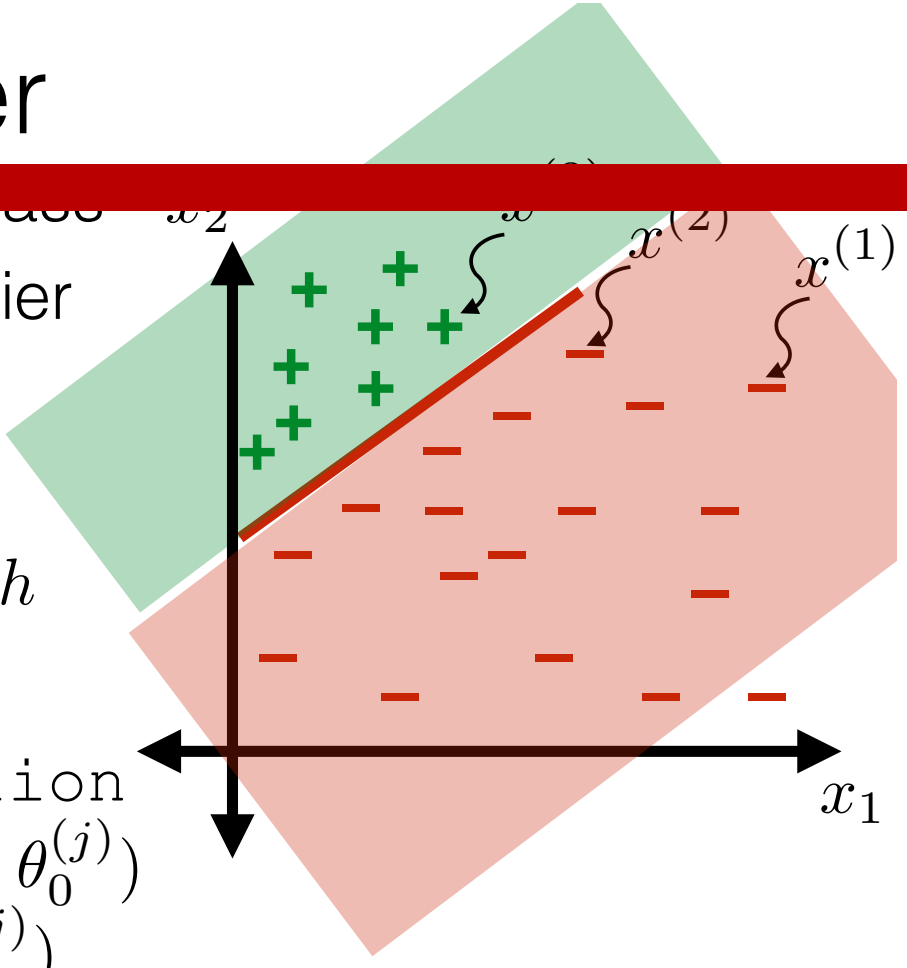
```
for j = 1, …, 1 trillion
  Randomly sample (θ^(j), θ_0^(j))
  Set h^(j)(x) = h(x; θ^(j), θ_0^(j))
Ex_learning_alg
```



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:

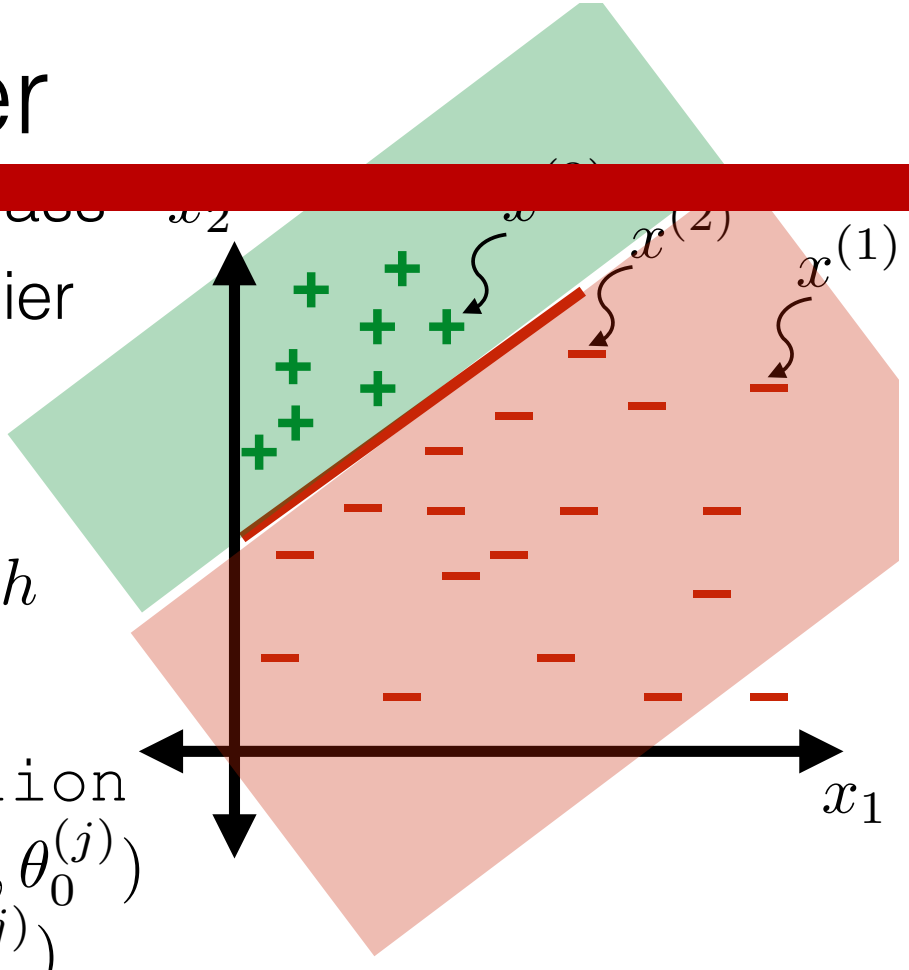
```
for j = 1, …, 1 trillion
  Randomly sample (θ^(j), θ_0^(j))
  Set h^(j)(x) = h(x; θ^(j), θ_0^(j))
Ex_learning_alg( D_n )
```



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)
```

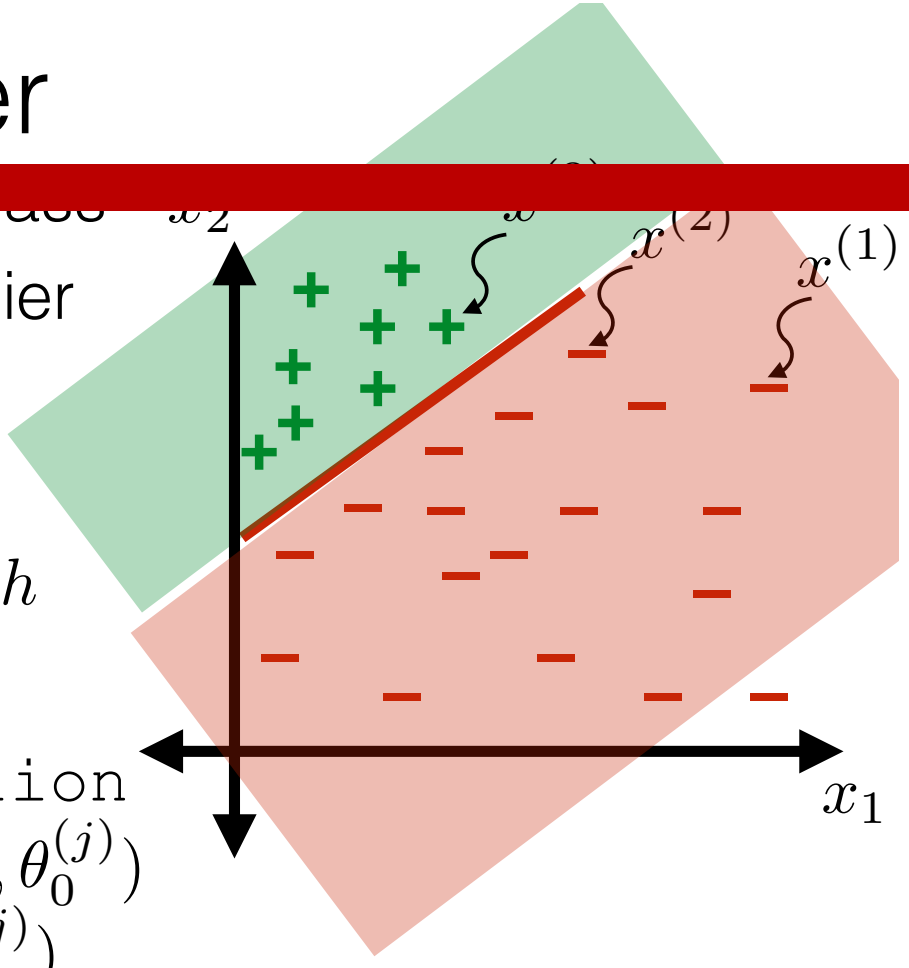


Learning a classifier

- 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)
```

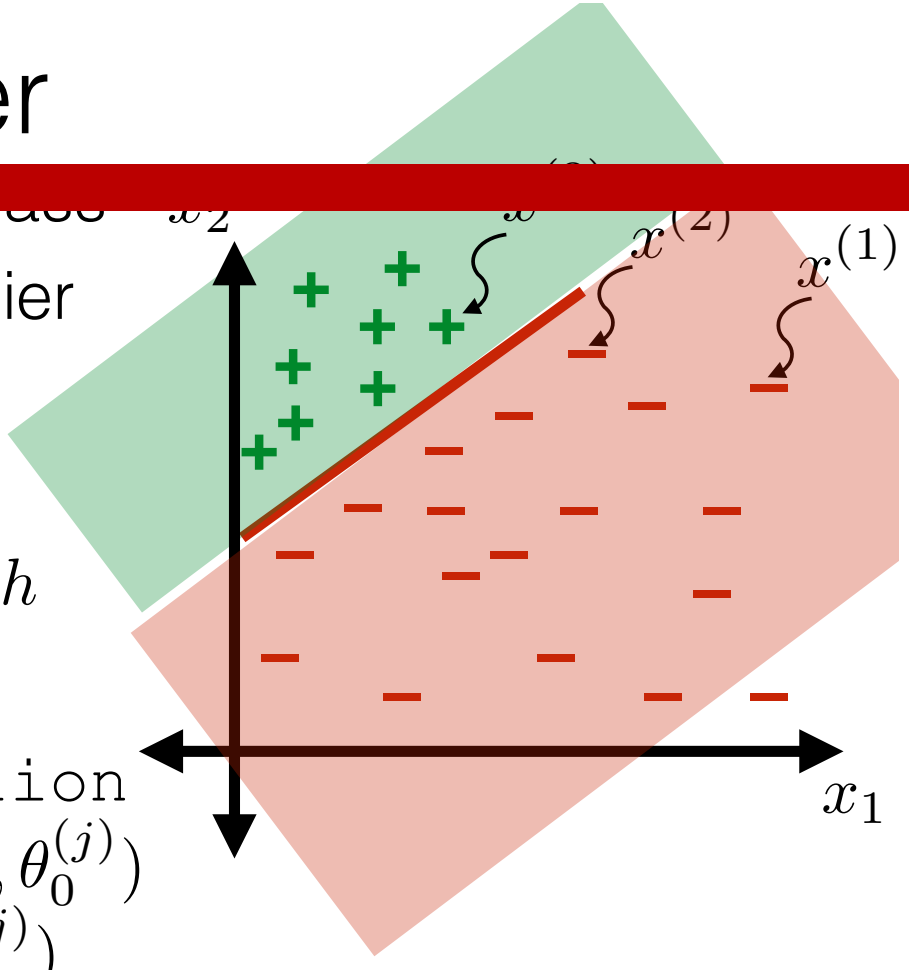
hyperparameter



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}} E_n(h^(j))
```



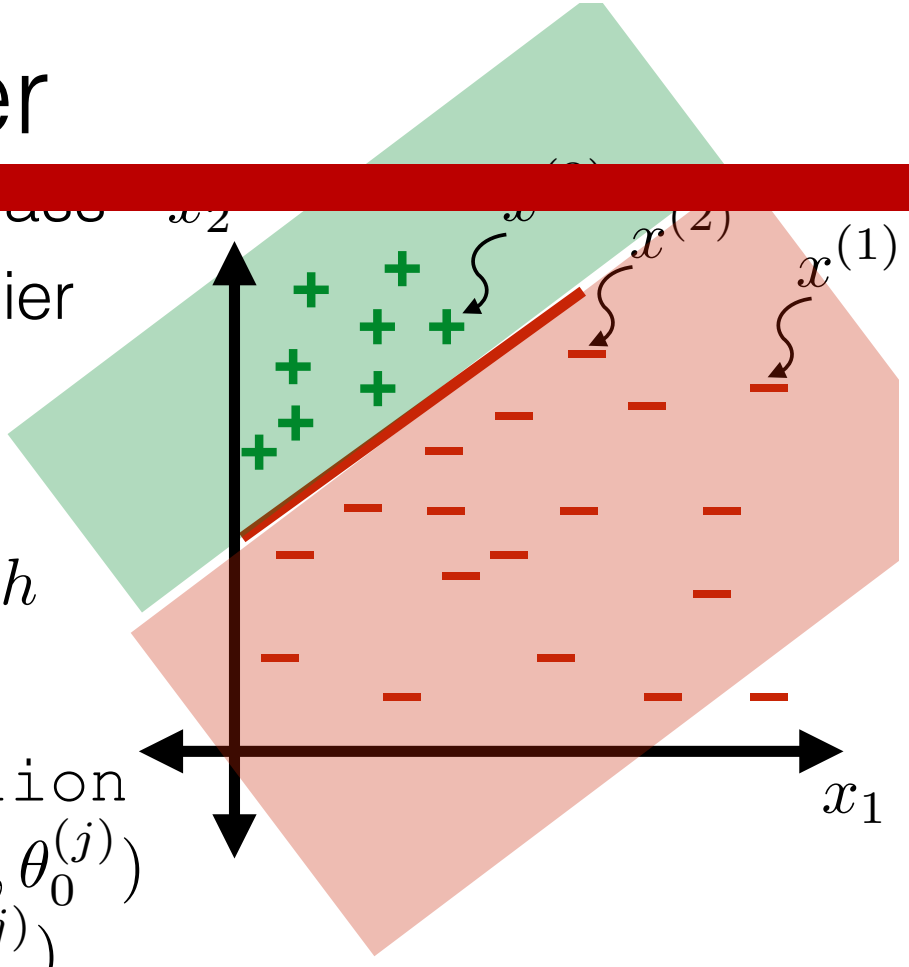
hyperparameter

Learning a classifier

- Have data, have hypothesis class
- Want to choose a good classifier
  - Recall: $x \rightarrow h \rightarrow y$
  - New: $\mathcal{D}_n \rightarrow$ 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}} E_n(h^(j))
    Return h^(j*)
```

hyperparameter

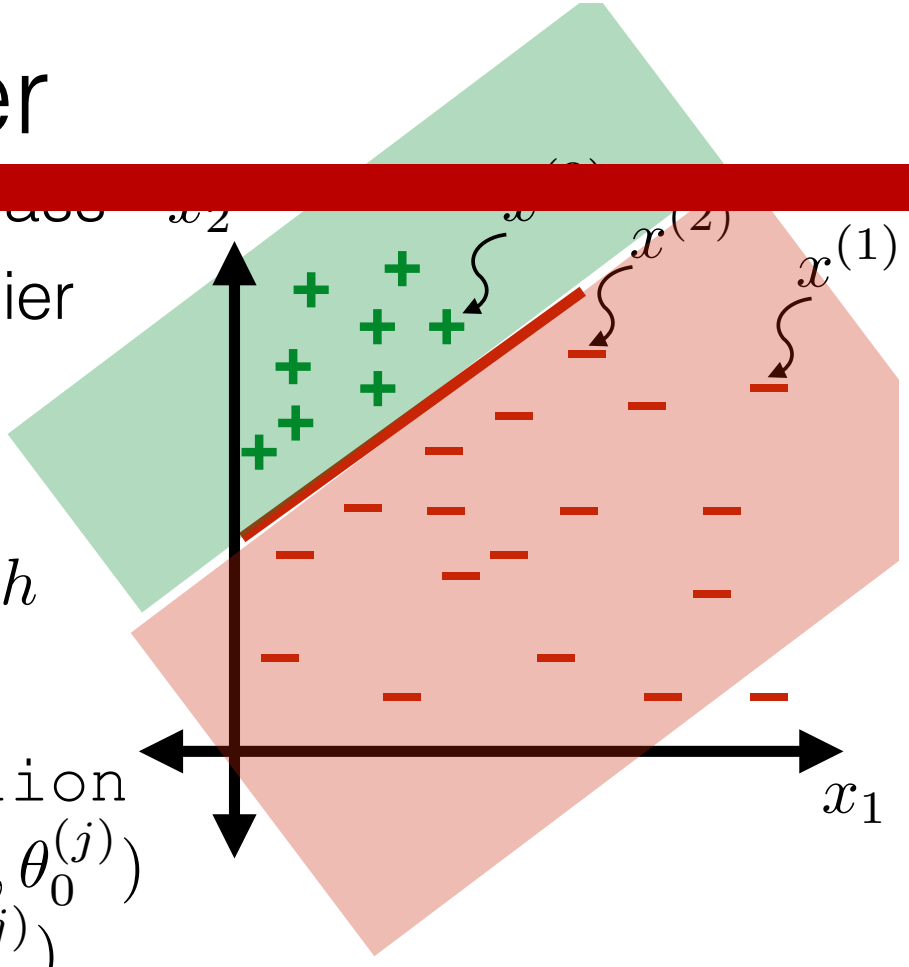


Learning a classifier

- 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}} E_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)`?

Perceptron Algorithm

Perceptron Algorithm

Perceptron

Perceptron Algorithm

Perceptron($\mathcal{D}_n$; τ)

Perceptron Algorithm

Perceptron($\mathcal{D}_n$; τ)

  Initialize $\theta = [0 \ 0 \ \dots \ 0]^\top$

  Initialize $\theta_0 = 0$

Perceptron Algorithm

`Perceptron(` $\mathcal{D}_n$ `;` τ `)`

`Initialize` $\theta = [0 \;\; 0 \;\; \ldots \;\; 0]^\top$ [How many 0s?]

`Initialize` $\theta_0 = 0$

Perceptron Algorithm

Perceptron($\mathcal{D}_n$; τ)

  Initialize $\theta = [0 \ 0 \ \ldots \ 0]^\top$ [How many 0s?]

  Initialize $\theta_0 = 0$

  for t = 1 to τ

Perceptron Algorithm

Perceptron($\mathcal{D}_n$; τ)

  Initialize $\theta = [0 \ 0 \ \ldots \ 0]^\top$ [How many 0s?]

  Initialize $\theta_0 = 0$

  for t = 1 to τ

    changed = False

Perceptron Algorithm

Perceptron($\mathcal{D}_n$; τ)

  Initialize $\theta = [0 \ 0 \ \ldots \ 0]^\top$ [How many 0s?]

  Initialize $\theta_0 = 0$

  for t = 1 to τ

    changed = False

    for i = 1 to n

Perceptron Algorithm

Perceptron($\mathcal{D}_n$; τ)

  Initialize $\theta = [0 \ 0 \ \ldots \ 0]^\top$ [How many 0s?]

  Initialize $\theta_0 = 0$

  for t = 1 to τ

    changed = False

    for i = 1 to n

      if $y^{(i)}(\theta^\top x^{(i)} + \theta_0) \leq 0$

Perceptron Algorithm

Perceptron($\mathcal{D}_n$; τ)

  Initialize $\theta = [0 \ 0 \ \ldots \ 0]^\top$ [How many 0s?]

  Initialize $\theta_0 = 0$

  for t = 1 to τ

    changed = False

    for i = 1 to n

      if $y^{(i)}(\theta^\top x^{(i)} + \theta_0) \leq 0$

Perceptron Algorithm

Perceptron($\mathcal{D}_n$; τ)

  Initialize $\theta = [0 \ 0 \ \ldots \ 0]^\top$ [How many 0s?]

  Initialize $\theta_0 = 0$

  for t = 1 to τ

    changed = False

    for i = 1 to n

      if $y^{(i)}(\theta^\top x^{(i)} + \theta_0) \leq 0$

Perceptron Algorithm

Perceptron($\mathcal{D}_n$; τ)

  Initialize $\theta = [0 \ 0 \ \ldots \ 0]^\top$ [How many 0s?]

  Initialize $\theta_0 = 0$

  for t = 1 to τ

    changed = False

    for i = 1 to n

      if $y^{(i)}(\theta^\top x^{(i)} + \theta_0) \leq 0$

Perceptron Algorithm

Perceptron($\mathcal{D}_n$; τ)

  Initialize $\theta = [0 \ 0 \ \ldots \ 0]^\top$ [How many 0s?]

  Initialize $\theta_0 = 0$

  for t = 1 to τ

    changed = False

    for i = 1 to n

      if $y^{(i)}(\theta^\top x^{(i)} + \theta_0) \leq 0$

Perceptron Algorithm

Perceptron($\mathcal{D}_n$; τ)

  Initialize $\theta = [0 \ 0 \ \ldots \ 0]^\top$ [How many 0s?]

  Initialize $\theta_0 = 0$

  for t = 1 to τ

    changed = False

    for i = 1 to n

      if $y^{(i)}(\theta^\top x^{(i)} + \theta_0) \leq 0$

Perceptron Algorithm

Perceptron($\mathcal{D}_n$; τ)

  Initialize $\theta = [0 \ 0 \ \ldots \ 0]^\top$

  Initialize $\theta_0 = 0$

  for t = 1 to τ

    changed = False

    for i = 1 to n

      if $y^{(i)}(\theta^\top x^{(i)} + \theta_0) \leq 0$

[How many 0s?]

[i.e. True if either:

Perceptron Algorithm

```
Perceptron( D_n ; τ )
  Initialize  θ = [0  0  ...  0]^⊤
  Initialize  θ_0 = 0
  for t = 1 to τ
    changed = False
    for i = 1 to n
      if  y^(i)(θ^⊤ x^(i) + θ_0) ≤ 0
```

[How many 0s?]

[i.e. True if either:

A. point is not on the line & prediction is wrong

Perceptron Algorithm

```
Perceptron( D_n ; τ )
  Initialize θ = [0 0 ... 0]^⊤
  Initialize θ_0 = 0
  for t = 1 to τ
    changed = False
    for i = 1 to n
      if y^(i)(θ^⊤ x^(i) + θ_0) ≤ 0
```

[How many 0s?]

[i.e. True if either:

A. point is not on the line & prediction is wrong
B. point is on the line

Perceptron Algorithm

Perceptron($\mathcal{D}_n$; τ)

  Initialize $\theta = [0 \ 0 \ \ldots \ 0]^\top$

  Initialize $\theta_0 = 0$

  for t = 1 to τ

    changed = False

    for i = 1 to n

      if $y^{(i)}(\theta^\top x^{(i)} + \theta_0) \leq 0$

[How many 0s?]

[i.e. True if either:

A. point is not on the line & prediction is wrong
B. point is on the line

Perceptron Algorithm

```
Perceptron( D_n ; τ )
  Initialize  θ = [0  0  ...  0]^⊤
  Initialize  θ_0 = 0
  for t = 1 to τ
    changed = False
    for i = 1 to n
      if  y^(i)(θ^⊤ x^(i) + θ_0) ≤ 0
```

[How many 0s?]

[i.e. True if either:

A. point is not on the line & prediction is wrong
B. point is on the line
C. initial step]

Perceptron Algorithm

```
Perceptron( D_n ; τ )
  Initialize  θ = [0 0 ... 0]^⊤
  Initialize  θ_0 = 0
  for t = 1 to τ
    changed = False
    for i = 1 to n
      if  y^(i) (θ^⊤ x^(i) + θ_0) ≤ 0
```

[How many 0s?]

[i.e. True if either:

A. point is not on the line & prediction is wrong
B. point is on the line
C. initial step]

Perceptron Algorithm

Perceptron($\mathcal{D}_n$; τ)

  Initialize $\theta = [0 \ 0 \ \ldots \ 0]^\top$

  Initialize $\theta_0 = 0$

  for t = 1 to τ

    changed = False

    for i = 1 to n

      if $y^{(i)}(\theta^\top x^{(i)} + \theta_0) \leq 0$

        Set $\theta = \theta + y^{(i)} x^{(i)}$

        Set $\theta_0 = \theta_0 + y^{(i)}$

[How many 0s?]

[i.e. True if either:

A. point is not on the line & prediction is wrong
B. point is on the line
C. initial step]

Perceptron Algorithm

Perceptron($\mathcal{D}_n$; τ)

  Initialize $\theta = [0 \ 0 \ \ldots \ 0]^\top$

  Initialize $\theta_0 = 0$

  for t = 1 to τ

    changed = False

    for i = 1 to n

      if $y^{(i)}(\theta^\top x^{(i)} + \theta_0) \leq 0$

        Set $\theta = \theta + y^{(i)} x^{(i)}$

        Set $\theta_0 = \theta_0 + y^{(i)}$

        changed = True

[How many 0s?]

[i.e. True if either:

A. point is not on the line & prediction is wrong
B. point is on the line
C. initial step]

Perceptron Algorithm

```
Perceptron( D_n ; τ )
  Initialize  θ = [0 0 ... 0]^⊤
  Initialize  θ_0 = 0
  for t = 1 to τ
    changed = False
    for i = 1 to n
      if y^(i)(θ^⊤ x^(i) + θ_0) ≤ 0
        Set  θ = θ + y^(i) x^(i)
        Set  θ_0 = θ_0 + y^(i)
        changed = True
    if not changed
      break
```

[How many 0s?]

[i.e. True if either:

A. point is not on the line & prediction is wrong
B. point is on the line
C. initial step]

Perceptron Algorithm

```
Perceptron( D_n ; τ )
  Initialize  θ = [0 0 ... 0]^⊤
  Initialize  θ_0 = 0
  for t = 1 to τ
    changed = False
    for i = 1 to n
      if y^(i)(θ^⊤ x^(i) + θ_0) ≤ 0
        Set  θ = θ + y^(i) x^(i)
        Set  θ_0 = θ_0 + y^(i)
        changed = True
    if not changed
      break
  Return  θ, θ_0
```

[How many 0s?]

[i.e. True if either:

A. point is not on the line & prediction is wrong
B. point is on the line
C. initial step]

Perceptron Algorithm

Perceptron($\mathcal{D}_n$; τ)

  Initialize $\theta = [0 \ 0 \ \ldots \ 0]^\top$

  Initialize $\theta_0 = 0$

  for t = 1 to τ

    changed = False

    for i = 1 to n

      if $y^{(i)}(\theta^\top x^{(i)} + \theta_0) \leq 0$

        Set $\theta = \theta + y^{(i)} x^{(i)}$

        Set $\theta_0 = \theta_0 + y^{(i)}$

        changed = True

    if not changed

      break

  Return θ, θ_0

[How many 0s?]

[i.e. True if either:

A. point is not on the line & prediction is wrong
B. point is on the line
C. initial step]

Perceptron Algorithm

```
Perceptron( D_n ; τ )
  Initialize  θ = [0 0 ... 0]^T
  Initialize  θ_0 = 0
  for t = 1 to τ
    changed = False
    for i = 1 to n
      if y^(i)(θ^T x^(i) + θ_0) ≤ 0
          Set  θ = θ + y^(i) x^(i)
          Set  θ_0 = θ_0 + y^(i)
          changed = True
    if not changed
      break
  Return  θ, θ_0
```

[How many 0s?]

[i.e. True if either:

A. point is not on the line & prediction is wrong
B. point is on the line
C. initial step]

What does an update do?

Perceptron Algorithm

```
Perceptron( D_n ; τ )
  Initialize  θ = [0 0 ... 0]^T
  Initialize  θ_0 = 0
  for t = 1 to τ
    changed = False
    for i = 1 to n
      if y^(i)(θ^T x^(i) + θ_0) ≤ 0
        Set  θ = θ + y^(i) x^(i)
        Set  θ_0 = θ_0 + y^(i)
        changed = True
    if not changed
      break
  Return  θ, θ_0
```

[How many 0s?]

[i.e. True if either:

A. point is not on the line & prediction is wrong
B. point is on the line
C. initial step]

What does an update do?

Perceptron Algorithm

Perceptron($\mathcal{D}_n$; τ)

  Initialize $\theta = [0 \ 0 \ \ldots \ 0]^\top$

  Initialize $\theta_0 = 0$

  for t = 1 to τ

    changed = False

    for i = 1 to n

      if $y^{(i)}(\theta^\top x^{(i)} + \theta_0) \leq 0$

        Set $\theta = \theta + y^{(i)} x^{(i)}$

        Set $\theta_0 = \theta_0 + y^{(i)}$

        changed = True

    if not changed

      break

  Return θ, θ_0

[How many 0s?]

[i.e. True if either:

A. point is not on the line & prediction is wrong
B. point is on the line
C. initial step]

What does an update do?

Perceptron Algorithm

```
Perceptron( D_n ; τ )
  Initialize  θ = [0 0 ... 0]^⊤
  Initialize  θ_0 = 0
  for t = 1 to τ
    changed = False
    for i = 1 to n
      if y^(i)(θ^⊤ x^(i) + θ_0) ≤ 0
        Set  θ = θ + y^(i) x^(i)
        Set  θ_0 = θ_0 + y^(i)
        changed = True
    if not changed
      break
  Return  θ, θ_0
```

[How many 0s?]

[i.e. True if either:

A. point is not on the line & prediction is wrong
B. point is on the line
C. initial step]

What does an update do?

$$y^{(i)}\left(\theta_{\text{updated}}^{\top} x^{(i)} + \theta_{0,\text{updated}}\right)$$

Perceptron Algorithm

Perceptron($\mathcal{D}_n$; τ)

  Initialize $\theta = [0 \;\; 0 \;\; \ldots \;\; 0]^\top$

  Initialize $\theta_0 = 0$

  for t = 1 to τ

    changed = False

    for i = 1 to n

      if $y^{(i)}(\theta^\top x^{(i)} + \theta_0) \leq 0$

        Set $\theta = \theta + y^{(i)} x^{(i)}$

        Set $\theta_0 = \theta_0 + y^{(i)}$

        changed = True

    if not changed

      break

  Return θ, θ_0

[How many 0s?]

[i.e. True if either:

A. point is not on the line & prediction is wrong
B. point is on the line
C. initial step]

What does an update do?

$$y^{(i)}\left((\theta + y^{(i)}x^{(i)})^\top x^{(i)} + (\theta_0 + y^{(i)})\right)$$

Perceptron Algorithm

Perceptron($\mathcal{D}_n$; τ)

  Initialize $\theta = [0 \ 0 \ \ldots \ 0]^\top$

  Initialize $\theta_0 = 0$

  for t = 1 to τ

    changed = False

    for i = 1 to n

      if $y^{(i)}(\theta^\top x^{(i)} + \theta_0) \leq 0$

        Set $\theta = \theta + y^{(i)} x^{(i)}$

        Set $\theta_0 = \theta_0 + y^{(i)}$

        changed = True

    if not changed

      break

  Return θ, θ_0

[How many 0s?]

[i.e. True if either:

A. point is not on the line & prediction is wrong
B. point is on the line
C. initial step]

What does an update do?

$$y^{(i)}\left((\theta + y^{(i)}x^{(i)})^\top x^{(i)} + (\theta_0 + y^{(i)})\right)$$
$$= y^{(i)}(\theta^\top x^{(i)} + \theta_0) + (y^{(i)})^2(x^{(i)\top} x^{(i)} + 1)$$

Perceptron Algorithm

Perceptron($\mathcal{D}_n$; τ)

  Initialize $\theta = [0 \ 0 \ \ldots \ 0]^\top$

  Initialize $\theta_0 = 0$

  for t = 1 to τ

    changed = False

    for i = 1 to n

      if $y^{(i)}(\theta^\top x^{(i)} + \theta_0) \leq 0$

        Set $\theta = \theta + y^{(i)} x^{(i)}$

        Set $\theta_0 = \theta_0 + y^{(i)}$

        changed = True

    if not changed

      break

  Return θ, θ_0

[How many 0s?]

[i.e. True if either:

A. point is not on the line & prediction is wrong
B. point is on the line
C. initial step]

What does an update do?

$$y^{(i)}\left((\theta + y^{(i)}x^{(i)})^\top x^{(i)} + (\theta_0 + y^{(i)})\right)$$
$$= y^{(i)}(\theta^\top x^{(i)} + \theta_0) + (y^{(i)})^2(x^{(i)\top} x^{(i)} + 1)$$

Perceptron Algorithm

Perceptron($\mathcal{D}_n$; τ)

  Initialize $\theta = [0 \ 0 \ \ldots \ 0]^\top$

  Initialize $\theta_0 = 0$

  for t = 1 to τ

    changed = False

    for i = 1 to n

      if $y^{(i)}(\theta^\top x^{(i)} + \theta_0) \leq 0$

        Set $\theta = \theta + y^{(i)} x^{(i)}$

        Set $\theta_0 = \theta_0 + y^{(i)}$

        changed = True

    if not changed

      break

  Return θ, θ_0

[How many 0s?]

[i.e. True if either:

A. point is not on the line & prediction is wrong
B. point is on the line
C. initial step]

What does an update do?

$$y^{(i)}\left((\theta + y^{(i)}x^{(i)})^\top x^{(i)} + (\theta_0 + y^{(i)})\right)$$

$$= y^{(i)}(\theta^\top x^{(i)} + \theta_0) + (y^{(i)})^2(x^{(i)\top}x^{(i)} + 1)$$

Perceptron Algorithm

Perceptron($\mathcal{D}_n$; τ)

  Initialize $\theta = [0 \ 0 \ \ldots \ 0]^\top$

  Initialize $\theta_0 = 0$

  for t = 1 to τ

    changed = False

    for i = 1 to n

      if $y^{(i)}(\theta^\top x^{(i)} + \theta_0) \leq 0$

        Set $\theta = \theta + y^{(i)} x^{(i)}$

        Set $\theta_0 = \theta_0 + y^{(i)}$

        changed = True

    if not changed

      break

  Return θ, θ_0

[How many 0s?]

[i.e. True if either:

A. point is not on the line & prediction is wrong
B. point is on the line
C. initial step]

What does an update do?

$$y^{(i)}\left((\theta + y^{(i)}x^{(i)})^\top x^{(i)} + (\theta_0 + y^{(i)})\right)$$
$$= y^{(i)}(\theta^\top x^{(i)} + \theta_0) + (y^{(i)})^2(x^{(i)\top} x^{(i)} + 1)$$

Perceptron Algorithm

Perceptron($\mathcal{D}_n$; τ)

  Initialize $\theta = [0 \ 0 \ \ldots \ 0]^\top$

  Initialize $\theta_0 = 0$

  for t = 1 to τ

    changed = False

    for i = 1 to n

      if $y^{(i)}(\theta^\top x^{(i)} + \theta_0) \leq 0$

        Set $\theta = \theta + y^{(i)} x^{(i)}$

        Set $\theta_0 = \theta_0 + y^{(i)}$

        changed = True

    if not changed

      break

  Return θ, θ_0

[How many 0s?]

[i.e. True if either:

A. point is not on the line & prediction is wrong
B. point is on the line
C. initial step]

What does an update do?

$$y^{(i)}\left((\theta + y^{(i)}x^{(i)})^\top x^{(i)} + (\theta_0 + y^{(i)})\right)$$
$$= y^{(i)}(\theta^\top x^{(i)} + \theta_0) + (y^{(i)})^2(x^{(i)\top} x^{(i)} + 1)$$

Perceptron Algorithm

Perceptron($\mathcal{D}_n$; τ)

  Initialize $\theta = [0 \ 0 \ \ldots \ 0]^\top$

  Initialize $\theta_0 = 0$

  for t = 1 to τ

    changed = False

    for i = 1 to n

      if $y^{(i)}(\theta^\top x^{(i)} + \theta_0) \leq 0$

        Set $\theta = \theta + y^{(i)} x^{(i)}$

        Set $\theta_0 = \theta_0 + y^{(i)}$

        changed = True

    if not changed

      break

  Return θ, θ_0

[How many 0s?]

[i.e. True if either:

A. point is not on the line & prediction is wrong
B. point is on the line
C. initial step]

What does an update do?

$$y^{(i)}\left((\theta + y^{(i)}x^{(i)})^\top x^{(i)} + (\theta_0 + y^{(i)})\right)$$
$$= y^{(i)}(\theta^\top x^{(i)} + \theta_0) + (y^{(i)})^2(x^{(i)\top} x^{(i)} + 1)$$

Perceptron Algorithm

Perceptron($\mathcal{D}_n$; τ)

  Initialize $\theta = [0 \ 0 \ \ldots \ 0]^\top$

  Initialize $\theta_0 = 0$

  for t = 1 to τ

    changed = False

    for i = 1 to n

      if $y^{(i)}(\theta^\top x^{(i)} + \theta_0) \leq 0$

        Set $\theta = \theta + y^{(i)} x^{(i)}$

        Set $\theta_0 = \theta_0 + y^{(i)}$

        changed = True

    if not changed

      break

  Return θ, θ_0

[How many 0s?]

[i.e. True if either:

A. point is not on the line & prediction is wrong
B. point is on the line
C. initial step]

What does an update do?

$$y^{(i)}\left((\theta + y^{(i)}x^{(i)})^\top x^{(i)} + (\theta_0 + y^{(i)})\right)$$
$$= y^{(i)}(\theta^\top x^{(i)} + \theta_0) + (y^{(i)})^2(x^{(i)\top} x^{(i)} + 1)$$

Perceptron Algorithm

Perceptron($\mathcal{D}_n$; τ)

  Initialize $\theta = [0 \ 0 \ \ldots \ 0]^\top$

  Initialize $\theta_0 = 0$

  for t = 1 to τ

    changed = False

    for i = 1 to n

      if $y^{(i)}(\theta^\top x^{(i)} + \theta_0) \leq 0$

        Set $\theta = \theta + y^{(i)} x^{(i)}$

        Set $\theta_0 = \theta_0 + y^{(i)}$

        changed = True

    if not changed

      break

  Return θ, θ_0

[How many 0s?]

[i.e. True if either:

A. point is not on the line & prediction is wrong
B. point is on the line
C. initial step]

What does an update do?

$$y^{(i)}\left((\theta + y^{(i)}x^{(i)})^\top x^{(i)} + (\theta_0 + y^{(i)})\right)$$
$$= y^{(i)}(\theta^\top x^{(i)} + \theta_0) + (y^{(i)})^2(x^{(i)\top} x^{(i)} + 1)$$

Perceptron Algorithm

Perceptron($\mathcal{D}_n$; τ)

  Initialize $\theta = [0 \ 0 \ \ldots \ 0]^\top$

  Initialize $\theta_0 = 0$

  for t = 1 to τ

    changed = False

    for i = 1 to n

      if $y^{(i)}(\theta^\top x^{(i)} + \theta_0) \leq 0$

        Set $\theta = \theta + y^{(i)} x^{(i)}$

        Set $\theta_0 = \theta_0 + y^{(i)}$

        changed = True

    if not changed

      break

  Return θ, θ_0

[How many 0s?]

[i.e. True if either:

A. point is not on the line & prediction is wrong
B. point is on the line
C. initial step]

What does an update do?

$$y^{(i)}\left((\theta + y^{(i)}x^{(i)})^\top x^{(i)} + (\theta_0 + y^{(i)})\right)$$
$$= y^{(i)}(\theta^\top x^{(i)} + \theta_0) + (y^{(i)})^2(x^{(i)\top} x^{(i)} + 1)$$

Perceptron Algorithm

Perceptron($\mathcal{D}_n$; τ)

  Initialize $\theta = [0 \ 0 \ \ldots \ 0]^\top$

  Initialize $\theta_0 = 0$

  for t = 1 to τ

    changed = False

    for i = 1 to n

      if $y^{(i)}(\theta^\top x^{(i)} + \theta_0) \leq 0$

        Set $\theta = \theta + y^{(i)} x^{(i)}$

        Set $\theta_0 = \theta_0 + y^{(i)}$

        changed = True

    if not changed

      break

  Return θ, θ_0

[How many 0s?]

[i.e. True if either:

A. point is not on the line & prediction is wrong
B. point is on the line
C. initial step]

What does an update do?

$$y^{(i)}\left((\theta + y^{(i)}x^{(i)})^\top x^{(i)} + (\theta_0 + y^{(i)})\right)$$

$$= y^{(i)}(\theta^\top x^{(i)} + \theta_0) + (y^{(i)})^2(x^{(i)\top} x^{(i)} + 1)$$

Perceptron Algorithm

Perceptron($\mathcal{D}_n$; τ)

  Initialize $\theta = [0 \ 0 \ \ldots \ 0]^\top$

  Initialize $\theta_0 = 0$

  for t = 1 to τ

    changed = False

    for i = 1 to n

      if $y^{(i)}(\theta^\top x^{(i)} + \theta_0) \leq 0$

        Set $\theta = \theta + y^{(i)} x^{(i)}$

        Set $\theta_0 = \theta_0 + y^{(i)}$

        changed = True

    if not changed

      break

  Return θ, θ_0

[How many 0s?]

[i.e. True if either:

A. point is not on the line & prediction is wrong
B. point is on the line
C. initial step]

What does an update do?

$$y^{(i)}\left((\theta + y^{(i)}x^{(i)})^\top x^{(i)} + (\theta_0 + y^{(i)})\right)$$

$$= y^{(i)}(\theta^\top x^{(i)} + \theta_0) + (y^{(i)})^2(x^{(i)\top} x^{(i)} + 1)$$

Perceptron Algorithm

Perceptron($\mathcal{D}_n$; τ)

  Initialize $\theta = [0 \ 0 \ \ldots \ 0]^\top$

  Initialize $\theta_0 = 0$

  for t = 1 to τ

    changed = False

    for i = 1 to n

      if $y^{(i)}(\theta^\top x^{(i)} + \theta_0) \leq 0$

        Set $\theta = \theta + y^{(i)} x^{(i)}$

        Set $\theta_0 = \theta_0 + y^{(i)}$

        changed = True

    if not changed

      break

  Return θ, θ_0

[How many 0s?]

[i.e. True if either:

A. point is not on the line & prediction is wrong
B. point is on the line
C. initial step]

What does an update do?

$$y^{(i)}\left((\theta + y^{(i)}x^{(i)})^\top x^{(i)} + (\theta_0 + y^{(i)})\right)$$
$$= y^{(i)}(\theta^\top x^{(i)} + \theta_0) + (y^{(i)})^2(x^{(i)\top} x^{(i)} + 1)$$

Perceptron Algorithm

Perceptron($\mathcal{D}_n$; τ)

  Initialize $\theta = [0 \ 0 \ \ldots \ 0]^\top$

  Initialize $\theta_0 = 0$

  for t = 1 to τ

    changed = False

    for i = 1 to n

      if $y^{(i)}(\theta^\top x^{(i)} + \theta_0) \leq 0$

        Set $\theta = \theta + y^{(i)} x^{(i)}$

        Set $\theta_0 = \theta_0 + y^{(i)}$

        changed = True

    if not changed

      break

  Return θ, θ_0

[How many 0s?]

[i.e. True if either:

A. point is not on the line & prediction is wrong
B. point is on the line
C. initial step]

What does an update do?

$$y^{(i)}\left((\theta + y^{(i)}x^{(i)})^\top x^{(i)} + (\theta_0 + y^{(i)})\right)$$
$$= y^{(i)}(\theta^\top x^{(i)} + \theta_0) + (y^{(i)})^2(x^{(i)\top} x^{(i)} + 1)$$

Perceptron Algorithm

Perceptron($\mathcal{D}_n$; τ)

  Initialize $\theta = [0 \ 0 \ \ldots \ 0]^\top$

  Initialize $\theta_0 = 0$

  for t = 1 to τ

    changed = False

    for i = 1 to n

      if $y^{(i)}(\theta^\top x^{(i)} + \theta_0) \leq 0$

        Set $\theta = \theta + y^{(i)} x^{(i)}$

        Set $\theta_0 = \theta_0 + y^{(i)}$

        changed = True

    if not changed

      break

  Return θ, θ_0

[How many 0s?]

[i.e. True if either:

A. point is not on the line & prediction is wrong
B. point is on the line
C. initial step]

What does an update do?

$$y^{(i)}\left((\theta + y^{(i)}x^{(i)})^\top x^{(i)} + (\theta_0 + y^{(i)})\right)$$
$$= y^{(i)}(\theta^\top x^{(i)} + \theta_0) + (y^{(i)})^2(x^{(i)\top} x^{(i)} + 1)$$

Perceptron Algorithm

Perceptron($\mathcal{D}_n$; τ)

  Initialize $\theta = [0 \ 0 \ \ldots \ 0]^\top$

  Initialize $\theta_0 = 0$

  for t = 1 to τ

    changed = False

    for i = 1 to n

      if $y^{(i)}(\theta^\top x^{(i)} + \theta_0) \leq 0$

        Set $\theta = \theta + y^{(i)} x^{(i)}$

        Set $\theta_0 = \theta_0 + y^{(i)}$

        changed = True

    if not changed

      break

  Return θ, θ_0

[How many 0s?]

[i.e. True if either:

A. point is not on the line & prediction is wrong
B. point is on the line
C. initial step]

What does an update do?

$$y^{(i)}\left((\theta + y^{(i)}x^{(i)})^\top x^{(i)} + (\theta_0 + y^{(i)})\right)$$
$$= y^{(i)}(\theta^\top x^{(i)} + \theta_0) + (y^{(i)})^2(x^{(i)\top} x^{(i)} + 1)$$
$$= y^{(i)}(\theta^\top x^{(i)} + \theta_0) + (\|x^{(i)}\|^2 + 1)$$

Perceptron Algorithm

Perceptron($\mathcal{D}_n$; τ)

  Initialize $\theta = [0 \ 0 \ \ldots \ 0]^\top$

  Initialize $\theta_0 = 0$

  for t = 1 to τ

    changed = False

    for i = 1 to n

      if $y^{(i)}(\theta^\top x^{(i)} + \theta_0) \leq 0$

        Set $\theta = \theta + y^{(i)} x^{(i)}$

        Set $\theta_0 = \theta_0 + y^{(i)}$

        changed = True

    if not changed

      break

  Return θ, θ_0

[How many 0s?]

[i.e. True if either:

A. point is not on the line & prediction is wrong
B. point is on the line
C. initial step]

What does an update do?

$$y^{(i)}\left((\theta + y^{(i)}x^{(i)})^\top x^{(i)} + (\theta_0 + y^{(i)})\right)$$
$$= y^{(i)}(\theta^\top x^{(i)} + \theta_0) + (y^{(i)})^2(x^{(i)\top}x^{(i)} + 1)$$
$$= y^{(i)}(\theta^\top x^{(i)} + \theta_0) + (\|x^{(i)}\|^2 + 1)$$

Perceptron Algorithm

Perceptron($\mathcal{D}_n$; τ)

  Initialize $\theta = [0 \ 0 \ \ldots \ 0]^\top$

  Initialize $\theta_0 = 0$

  for t = 1 to τ

    changed = False

    for i = 1 to n

      if $y^{(i)}(\theta^\top x^{(i)} + \theta_0) \leq 0$

        Set $\theta = \theta + y^{(i)} x^{(i)}$

        Set $\theta_0 = \theta_0 + y^{(i)}$

        changed = True

    if not changed

      break

  Return θ, θ_0

[How many 0s?]

[i.e. True if either:

A. point is not on the line & prediction is wrong
B. point is on the line
C. initial step]

What does an update do?

$$y^{(i)}\left((\theta + y^{(i)}x^{(i)})^\top x^{(i)} + (\theta_0 + y^{(i)})\right)$$
$$= y^{(i)}(\theta^\top x^{(i)} + \theta_0) + (y^{(i)})^2(x^{(i)\top} x^{(i)} + 1)$$
$$= y^{(i)}(\theta^\top x^{(i)} + \theta_0) + (\|x^{(i)}\|^2 + 1)$$

Perceptron Algorithm

```
Perceptron( D_n ; τ )
  Initialize  θ = [0 0 ... 0]^T
  Initialize  θ_0 = 0
  for t = 1 to τ
    changed = False
    for i = 1 to n
      if y^(i)(θ^T x^(i) + θ_0) ≤ 0
        Set  θ = θ + y^(i) x^(i)
        Set  θ_0 = θ_0 + y^(i)
        changed = True
    if not changed
      break
  Return  θ, θ_0
```

[How many 0s?]

[i.e. True if either:

A. point is not on the line & prediction is wrong
B. point is on the line
C. initial step]

What does an update do?

$$y^{(i)}\left((\theta + y^{(i)}x^{(i)})^\top x^{(i)} + (\theta_0 + y^{(i)})\right)$$
$$= y^{(i)}(\theta^\top x^{(i)} + \theta_0) + (y^{(i)})^2(x^{(i)\top}x^{(i)} + 1)$$
$$= y^{(i)}(\theta^\top x^{(i)} + \theta_0) + (\|x^{(i)}\|^2 + 1)$$

Perceptron Algorithm

Perceptron($\mathcal{D}_n$; τ)

  Initialize $\theta = [0 \ 0 \ \ldots \ 0]^\top$

  Initialize $\theta_0 = 0$

  for t = 1 to τ

    changed = False

    for i = 1 to n

      if $y^{(i)}(\theta^\top x^{(i)} + \theta_0) \leq 0$

        Set $\theta = \theta + y^{(i)} x^{(i)}$

        Set $\theta_0 = \theta_0 + y^{(i)}$

        changed = True

    if not changed

      break

  Return θ, θ_0

[How many 0s?]

[i.e. True if either:

A. point is not on the line & prediction is wrong
B. point is on the line
C. initial step]

What does an update do?

$$y^{(i)}\left((\theta + y^{(i)}x^{(i)})^\top x^{(i)} + (\theta_0 + y^{(i)})\right)$$
$$= y^{(i)}(\theta^\top x^{(i)} + \theta_0) + (y^{(i)})^2(x^{(i)\top} x^{(i)} + 1)$$
$$= y^{(i)}(\theta^\top x^{(i)} + \theta_0) + (\|x^{(i)}\|^2 + 1)$$

Perceptron Algorithm

Perceptron($\mathcal{D}_n$; τ)

  Initialize $\theta = [0 \ 0 \ \ldots \ 0]^\top$

  Initialize $\theta_0 = 0$

  for t = 1 to τ

    changed = False

    for i = 1 to n

      if $y^{(i)}(\theta^\top x^{(i)} + \theta_0) \leq 0$

        Set $\theta = \theta + y^{(i)} x^{(i)}$

        Set $\theta_0 = \theta_0 + y^{(i)}$

        changed = True

    if not changed

      break

  Return θ, θ_0

[demo]

[How many 0s?]

[i.e. True if either:

A. point is not on the line & prediction is wrong
B. point is on the line
C. initial step]

What does an update do?

$$y^{(i)}\left((\theta + y^{(i)}x^{(i)})^\top x^{(i)} + (\theta_0 + y^{(i)})\right)$$
$$= y^{(i)}(\theta^\top x^{(i)} + \theta_0) + (y^{(i)})^2(x^{(i)\top} x^{(i)} + 1)$$
$$= y^{(i)}(\theta^\top x^{(i)} + \theta_0) + (\|x^{(i)}\|^2 + 1)$$

Let's Talk About Classifier Quality

Let's Talk About Classifier Quality

- *Definition*: A training set $\mathcal{D}_n$ is **linearly separable** if there exist θ, θ_0 such that, for every point index $i \in \{1, \ldots, n\}$, we have

$$y^{(i)}(\theta^\top x^{(i)} + \theta_0) > 0$$

Let's Talk About Classifier Quality

- *Definition*: A training set $\mathcal{D}_n$ is **linearly separable** if there exist θ, θ_0 such that, for every point index $i \in \{1, \ldots, n\}$, we have

$$y^{(i)}(\theta^\top x^{(i)} + \theta_0) > 0$$

Let's Talk About Classifier Quality

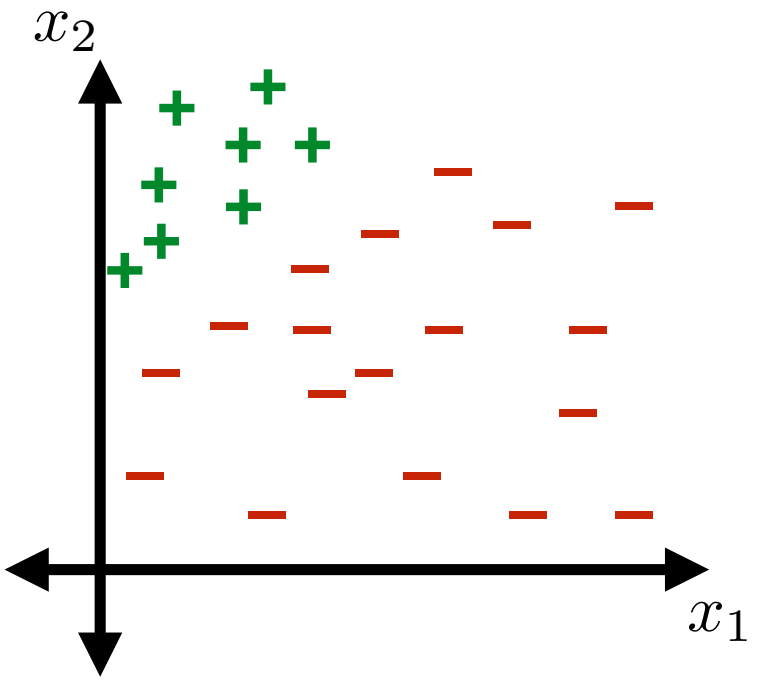
- *Definition*: A training set $\mathcal{D}_n$ is **linearly separable** if there exist θ, θ_0 such that, for every point index $i \in \{1, \ldots, n\}$, we have

$$y^{(i)}(\theta^\top x^{(i)} + \theta_0) > 0$$

Let's Talk About Classifier Quality

- *Definition*: A training set $\mathcal{D}_n$ is **linearly separable** if there exist θ, θ_0 such that, for every point index $i \in \{1, \ldots, n\}$, we have

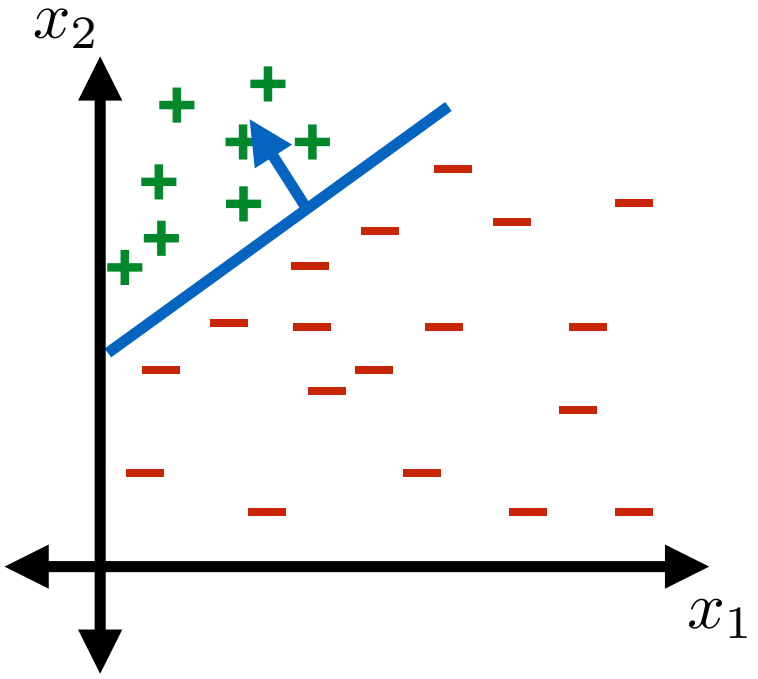
$$y^{(i)}(\theta^\top x^{(i)} + \theta_0) > 0$$



Let's Talk About Classifier Quality

- *Definition*: A training set $\mathcal{D}_n$ is **linearly separable** if there exist θ, θ_0 such that, for every point index $i \in \{1, \ldots, n\}$, we have

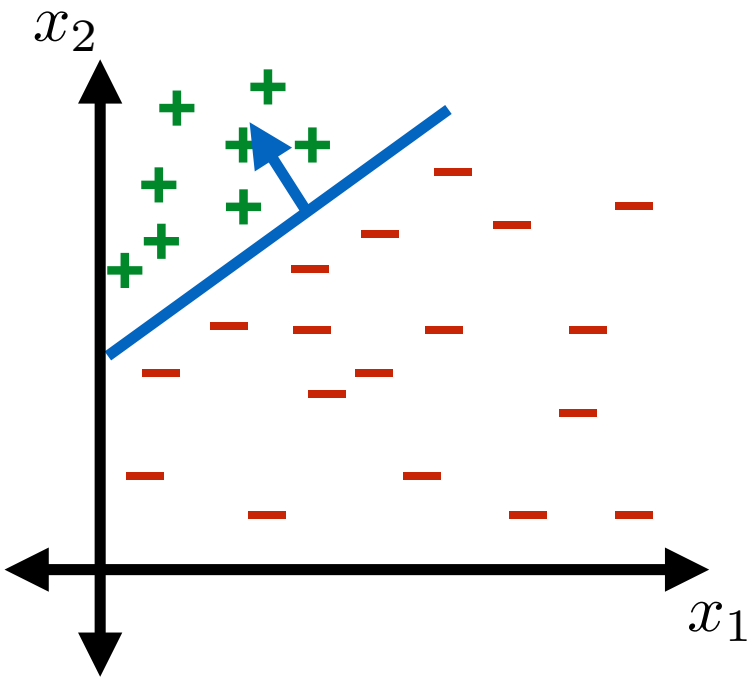
$$y^{(i)}(\theta^\top x^{(i)} + \theta_0) > 0$$

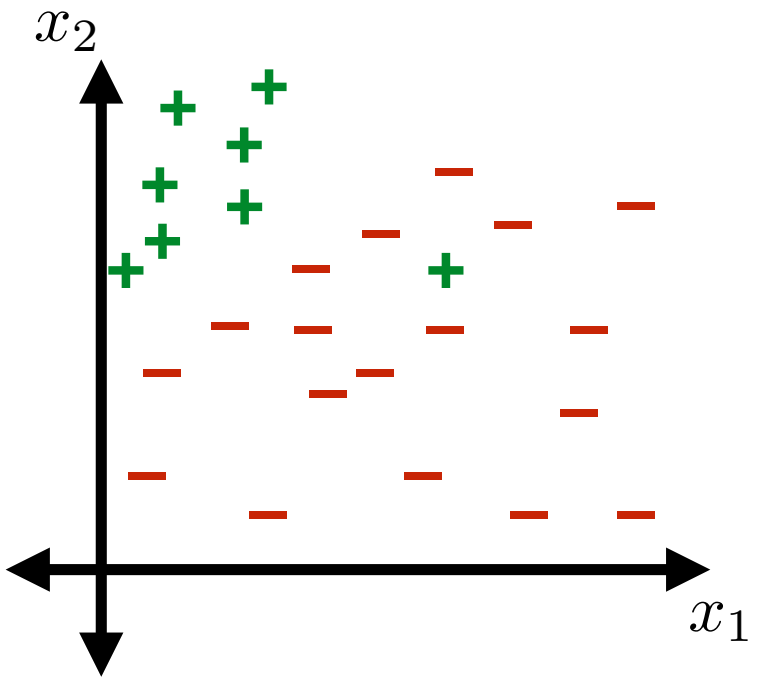


Let's Talk About Classifier Quality

- *Definition*: A training set $\mathcal{D}_n$ is **linearly separable** if there exist θ, θ_0 such that, for every point index $i \in \{1, \ldots, n\}$, we have

$$y^{(i)}(\theta^\top x^{(i)} + \theta_0) > 0$$

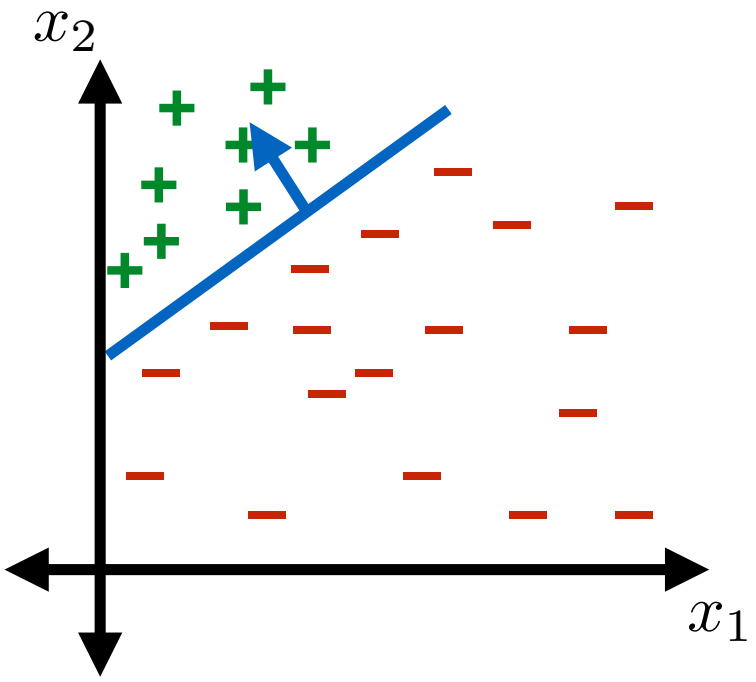


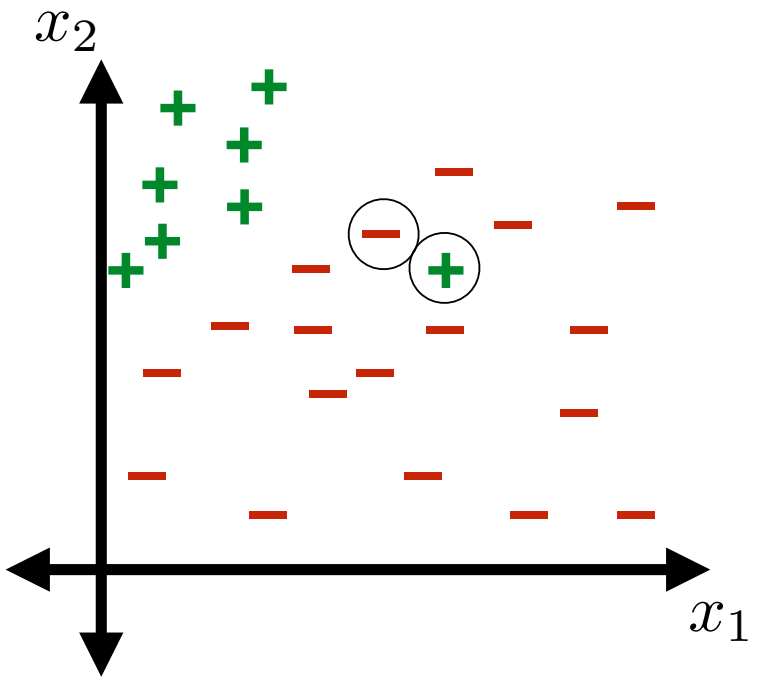


Let's Talk About Classifier Quality

- *Definition*: A training set $\mathcal{D}_n$ is **linearly separable** if there exist θ, θ_0 such that, for every point index $i \in \{1, \ldots, n\}$, we have

$$y^{(i)}(\theta^\top x^{(i)} + \theta_0) > 0$$

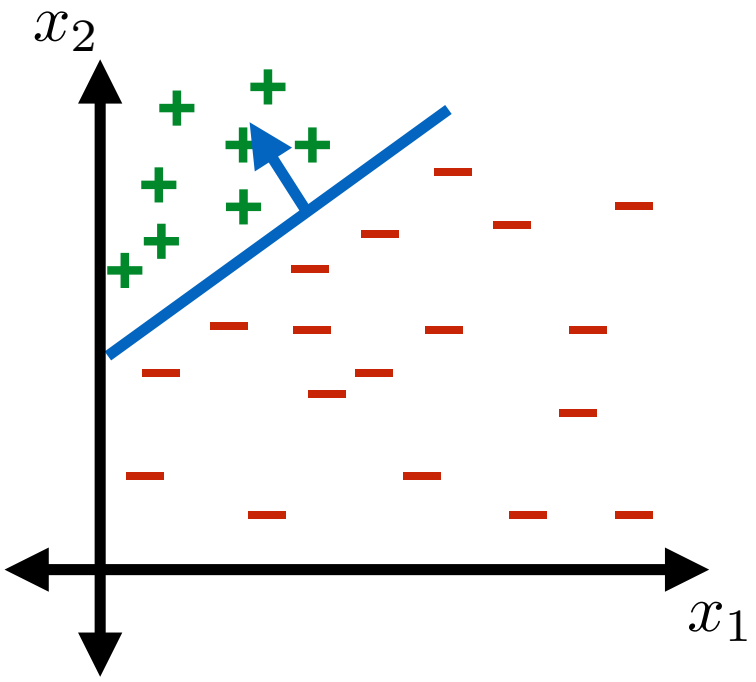


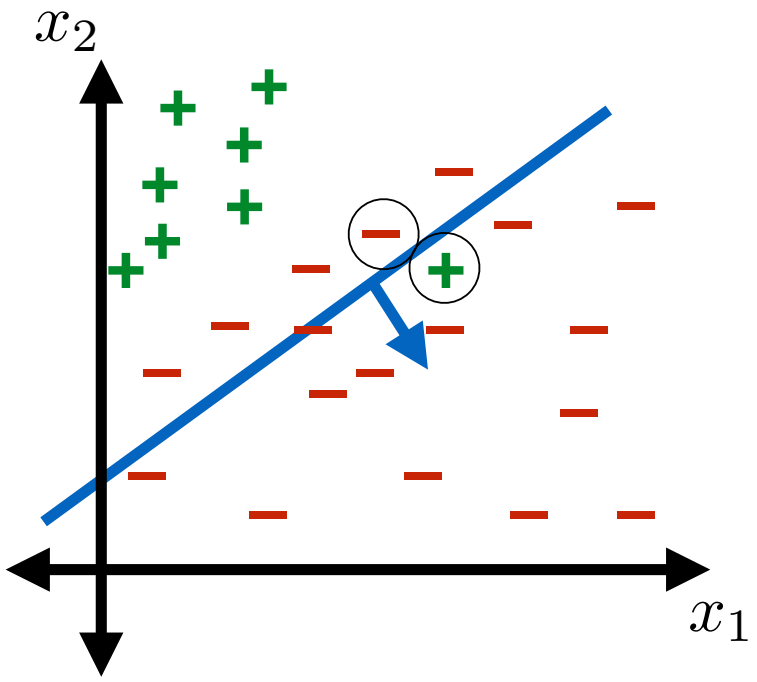


Let's Talk About Classifier Quality

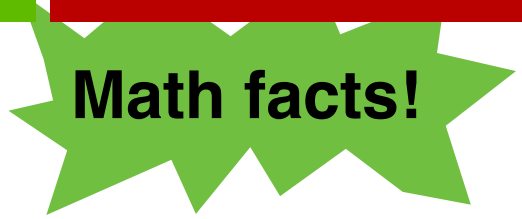
- *Definition*: A training set $\mathcal{D}_n$ is **linearly separable** if there exist θ, θ_0 such that, for every point index $i \in \{1, \ldots, n\}$, we have

$$y^{(i)}(\theta^\top x^{(i)} + \theta_0) > 0$$



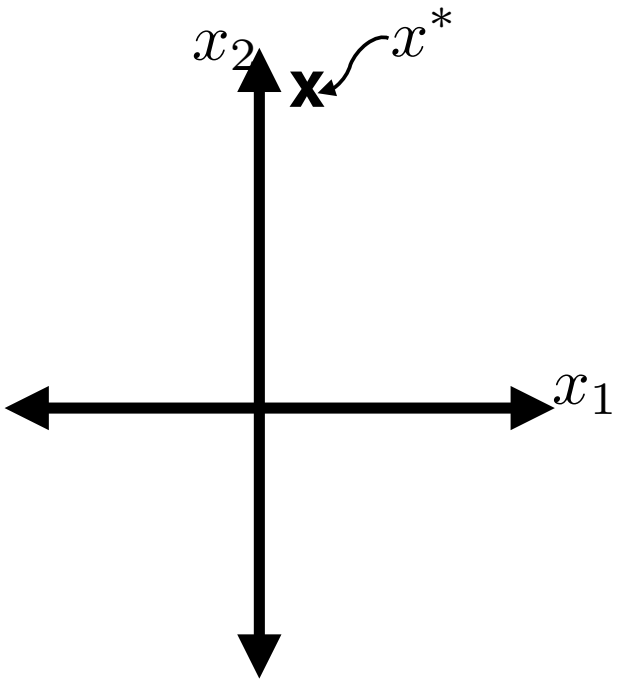


Let's Talk About Classifier Quality



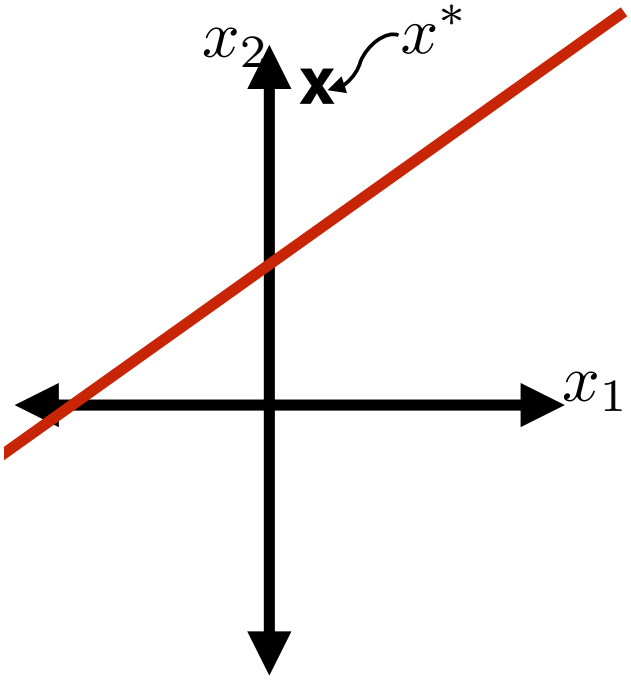
Let's Talk About Classifier Quality

Math facts!



Let's Talk About Classifier Quality

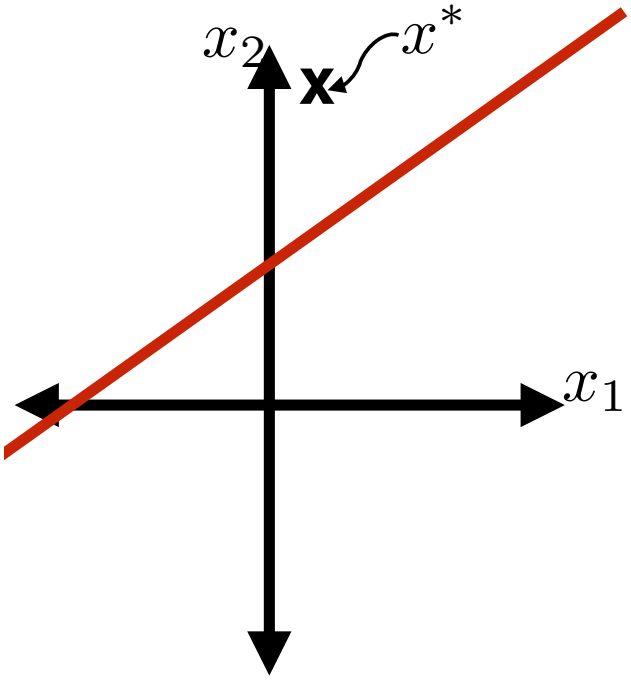
Math facts!



Let's Talk About Classifier Quality

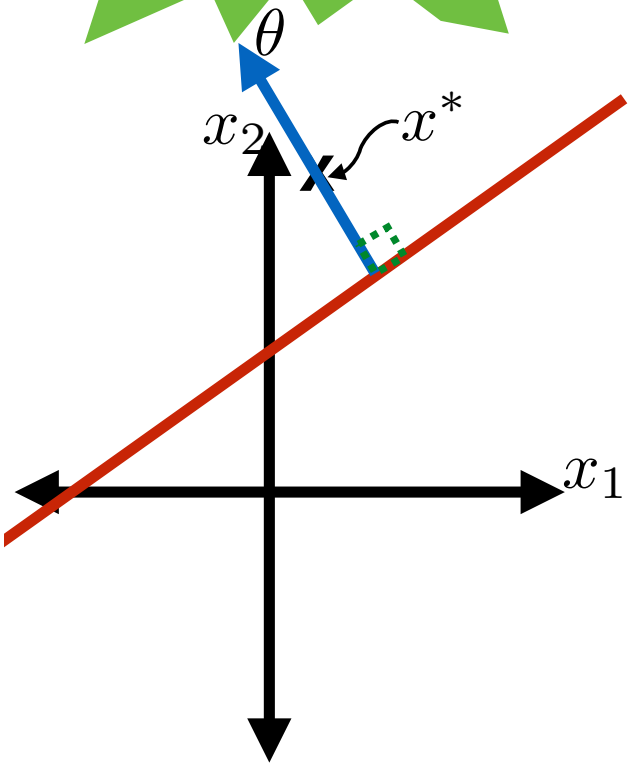
Math facts!

The signed distance from a hyperplane defined by θ, θ_0 to a point x^* is:



Let's Talk About Classifier Quality

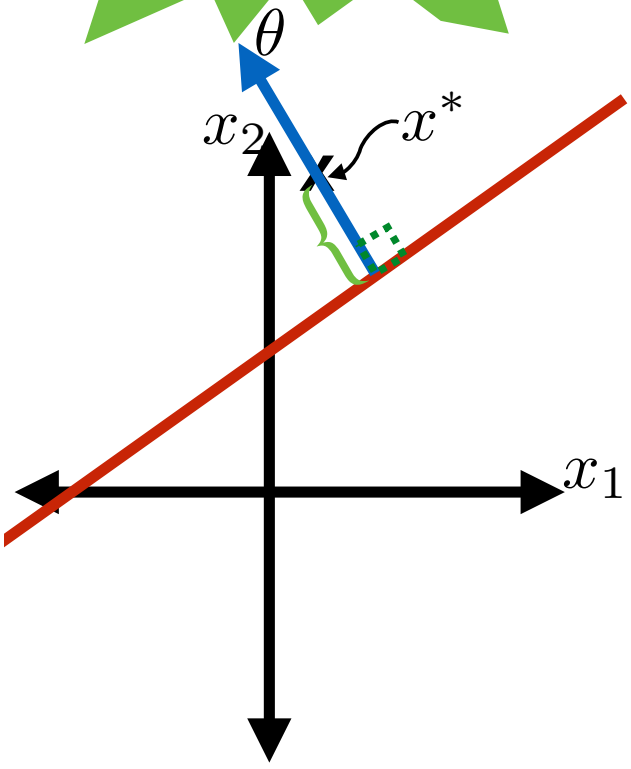
Math facts!



The signed distance from a hyperplane defined by θ, θ_0 to a point x^* is:

Let's Talk About Classifier Quality

Math facts!

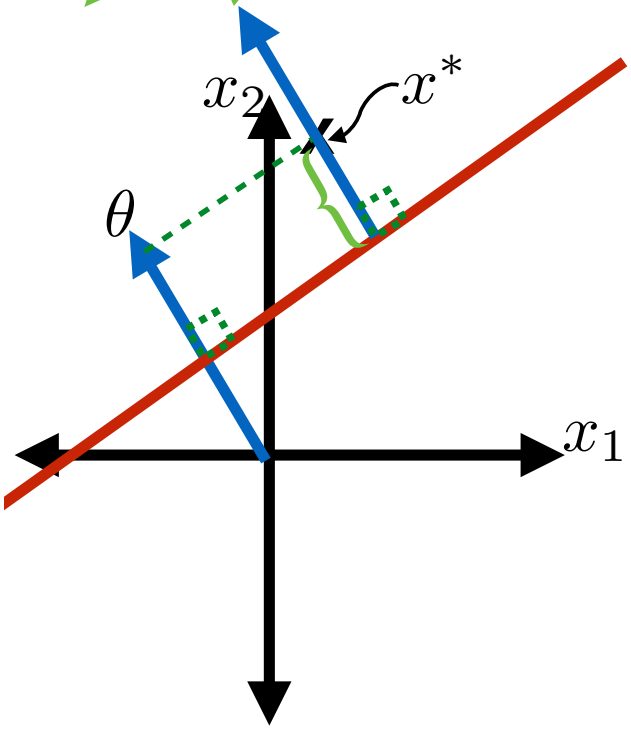


The signed distance from a hyperplane defined by θ, θ_0 to a point x^* is:

Let's Talk About Classifier Quality

Math facts!

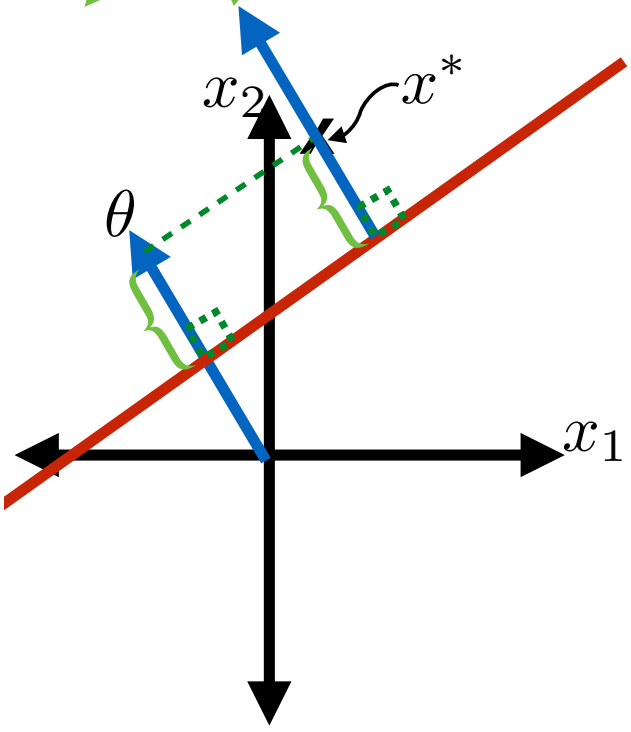
The signed distance from a hyperplane defined by θ, θ_0 to a point x^* is:



Let's Talk About Classifier Quality

Math facts!

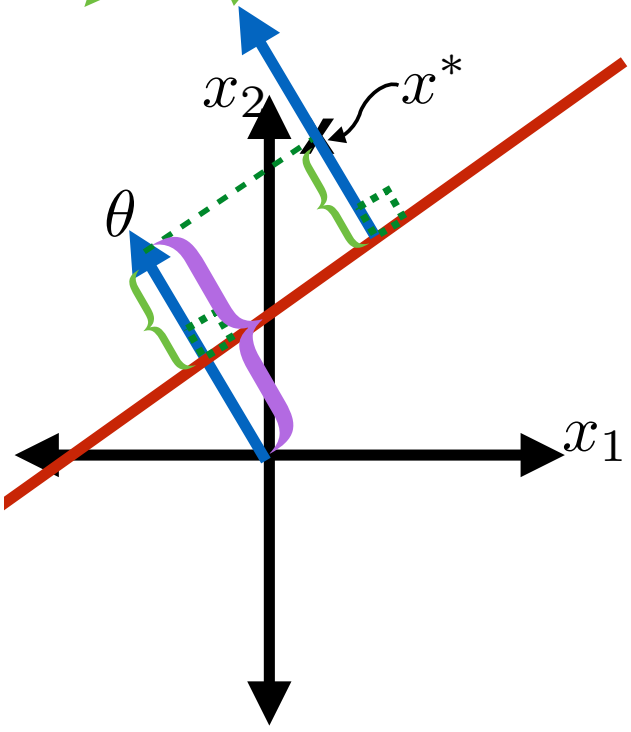
The signed distance from a hyperplane defined by θ, θ_0 to a point x^* is:



Let's Talk About Classifier Quality

Math facts!

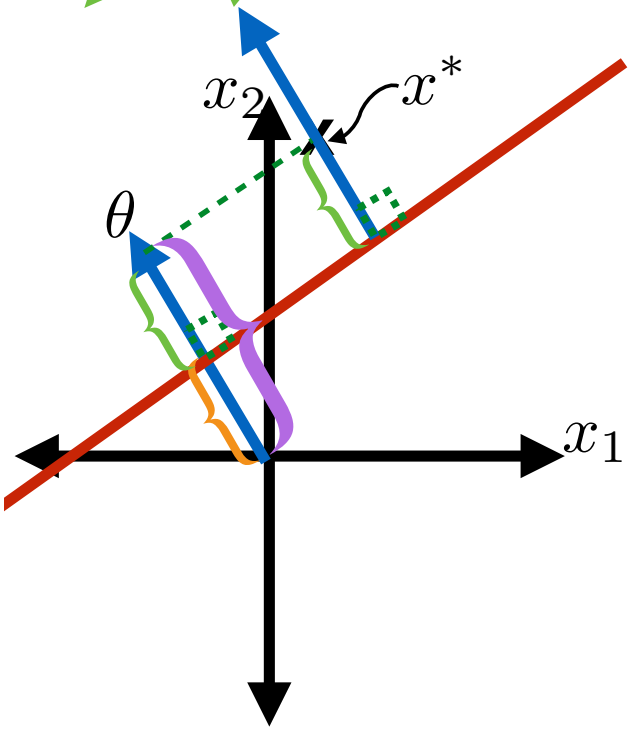
The signed distance from a hyperplane defined by θ, θ_0 to a point x^* is:



Let's Talk About Classifier Quality

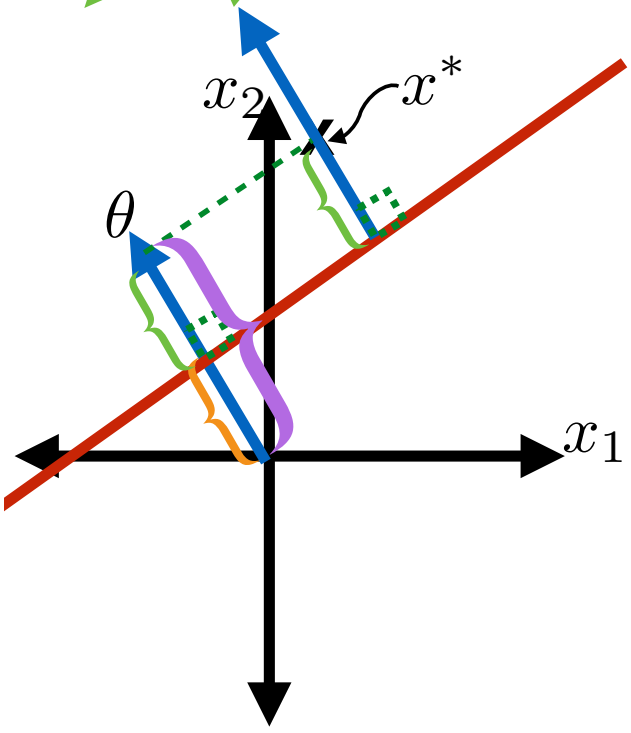
Math facts!

The signed distance from a hyperplane defined by θ, θ_0 to a point x^* is:



Let's Talk About Classifier Quality

Math facts!

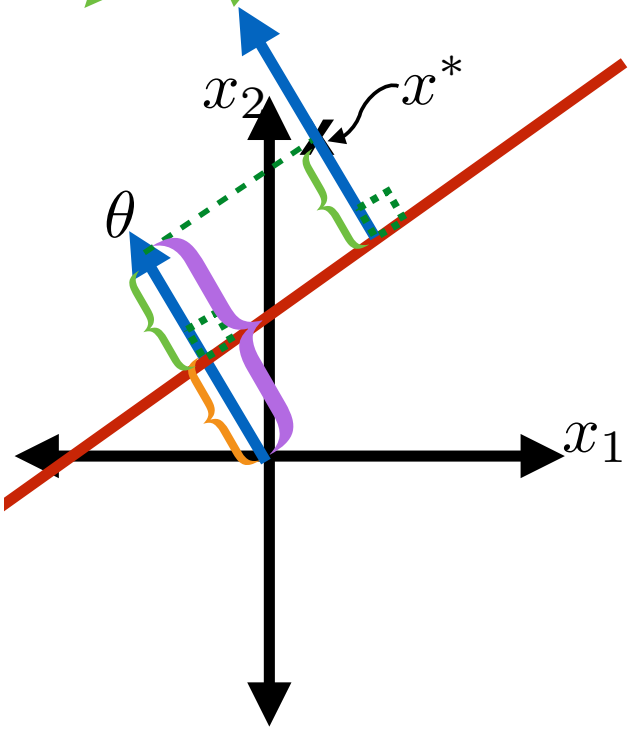


The signed distance from a hyperplane defined by θ, θ_0 to a point x^* is:

= projection of x^* on θ

Let's Talk About Classifier Quality

Math facts!



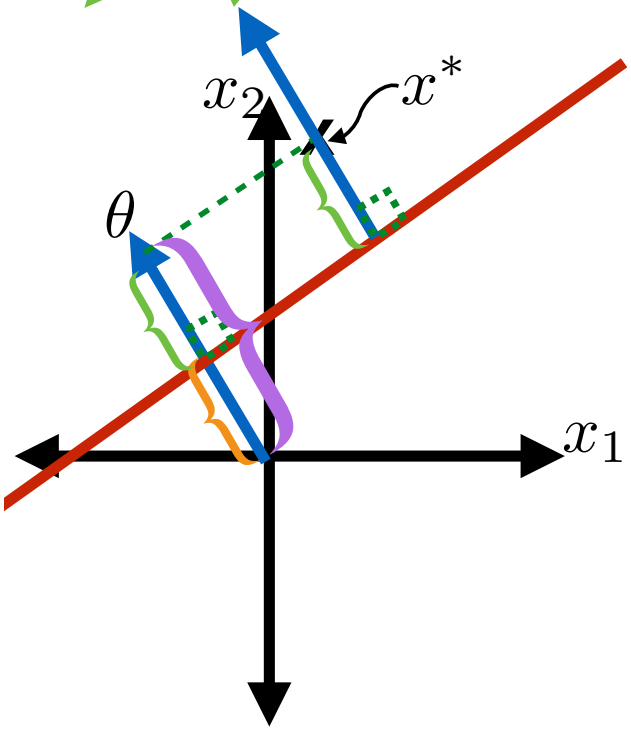
The signed distance from a hyperplane defined by θ, θ_0 to a point x^* is:

= projection of x^* on θ

– signed distance of line to origin

Let's Talk About Classifier Quality

Math facts!



The signed distance from a hyperplane defined by θ, θ_0 to a point x^* is:

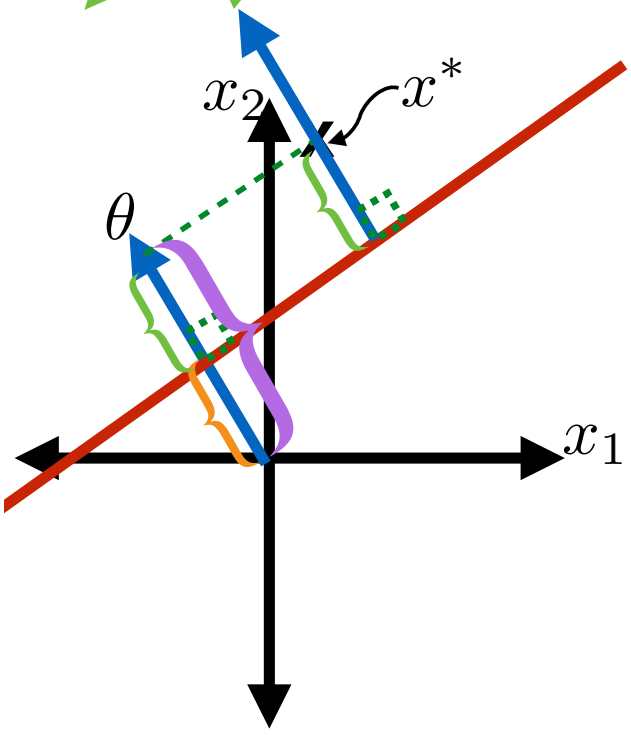
= projection of x^* on θ

– signed distance of line to origin

$$= \frac{\theta^\top x^*}{\|\theta\|}$$

Let's Talk About Classifier Quality

Math facts!



The signed distance from a hyperplane defined by θ, θ_0 to a point x^* is:

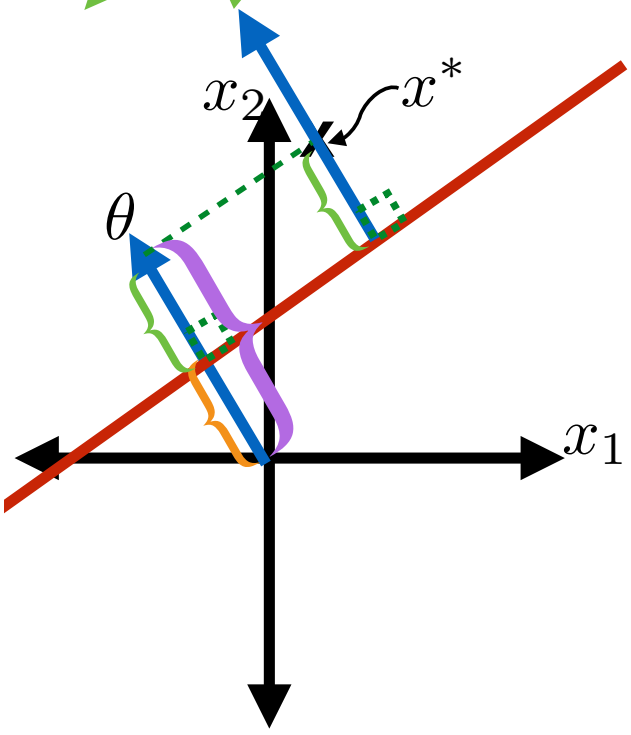
= projection of x^* on θ

− signed distance of line to origin

$$= \frac{\theta^\top x^*}{\|\theta\|} - \frac{-\theta_0}{\|\theta\|}$$

Let's Talk About Classifier Quality

Math facts!



The signed distance from a hyperplane defined by θ, θ_0 to a point x^* is:

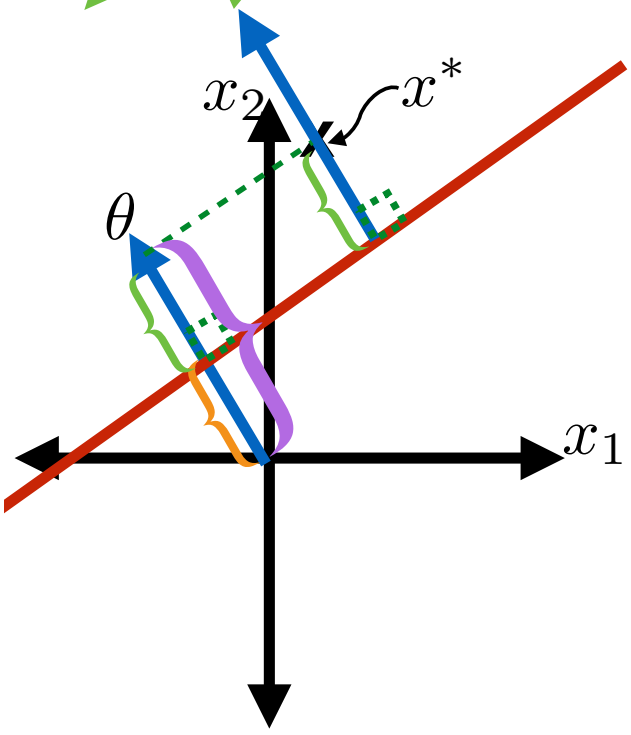
= projection of x^* on θ

– signed distance of line to origin

$$= \frac{\theta^\top x^*}{\|\theta\|} - \frac{-\theta_0}{\|\theta\|} = \frac{\theta^\top x^* + \theta_0}{\|\theta\|}$$

Let's Talk About Classifier Quality

Math facts!



The signed distance from a hyperplane defined by θ, θ_0 to a point x^* is:

= projection of x^* on θ

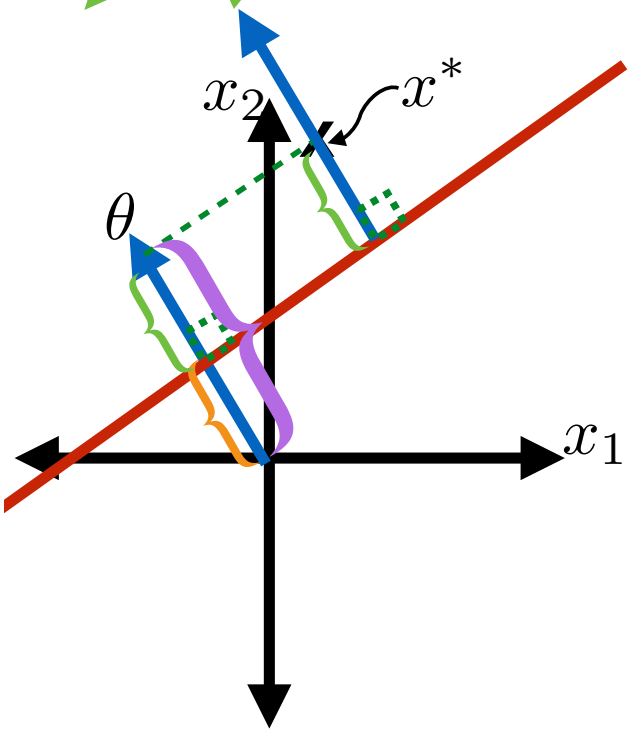
– signed distance of line to origin

$$= \frac{\theta^\top x^*}{\|\theta\|} - \frac{-\theta_0}{\|\theta\|} = \frac{\theta^\top x^* + \theta_0}{\|\theta\|}$$

- *Definition*: The **margin of the labelled point** (x^*, y^*) with respect to the hyperplane defined by θ, θ_0 is:

Let's Talk About Classifier Quality

Math facts!



The signed distance from a hyperplane defined by θ, θ_0 to a point x^* is:

= projection of x^* on θ

– signed distance of line to origin

$$= \frac{\theta^\top x^*}{\|\theta\|} - \frac{-\theta_0}{\|\theta\|} = \frac{\theta^\top x^* + \theta_0}{\|\theta\|}$$

- *Definition*: The **margin of the labelled point** (x^*, y^*) with respect to the hyperplane defined by θ, θ_0 is:

$$y^* \left(\frac{\theta^\top x^* + \theta_0}{\|\theta\|} \right)$$

Let's Talk About Classifier Quality

Math facts!

The signed distance from a hyperplane defined by θ, θ_0 to a point x^* is:

= projection of x^* on θ

− signed distance of line to origin

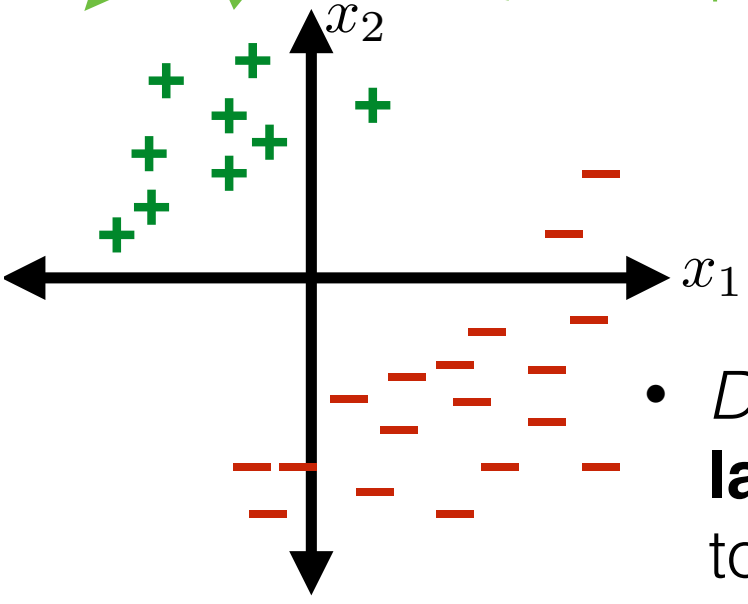
$$= \frac{\theta^\top x^*}{\|\theta\|} - \frac{-\theta_0}{\|\theta\|} = \frac{\theta^\top x^* + \theta_0}{\|\theta\|}$$

- *Definition*: The **margin of the labelled point** (x^*, y^*) with respect to the hyperplane defined by θ, θ_0 is:

$$y^* \left(\frac{\theta^\top x^* + \theta_0}{\|\theta\|} \right)$$

Let's Talk About Classifier Quality

Math facts!



The signed distance from a hyperplane defined by θ, θ_0 to a point x^* is:

= projection of x^* on θ

− signed distance of line to origin

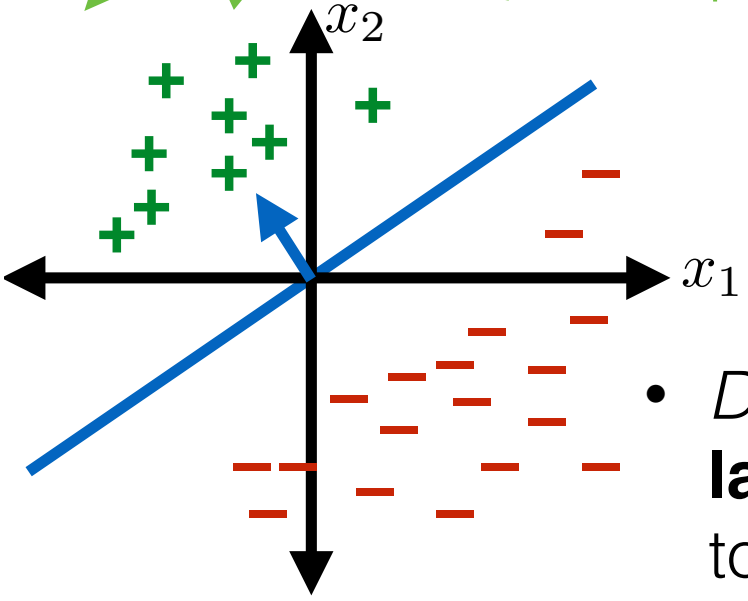
$$= \frac{\theta^\top x^*}{\|\theta\|} - \frac{-\theta_0}{\|\theta\|} = \frac{\theta^\top x^* + \theta_0}{\|\theta\|}$$

- *Definition*: The **margin of the labelled point** (x^*, y^*) with respect to the hyperplane defined by θ, θ_0 is:

$$y^* \left(\frac{\theta^\top x^* + \theta_0}{\|\theta\|} \right)$$

Let's Talk About Classifier Quality

Math facts!



The signed distance from a hyperplane defined by θ, θ_0 to a point x^* is:

= projection of x^* on θ

− signed distance of line to origin

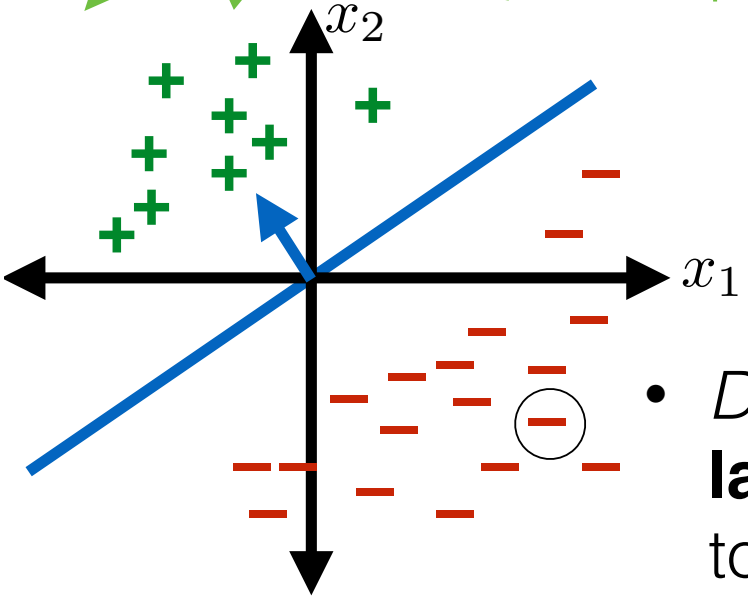
$$= \frac{\theta^\top x^*}{\|\theta\|} - \frac{-\theta_0}{\|\theta\|} = \frac{\theta^\top x^* + \theta_0}{\|\theta\|}$$

- *Definition*: The **margin of the labelled point** (x^*, y^*) with respect to the hyperplane defined by θ, θ_0 is:

$$y^* \left(\frac{\theta^\top x^* + \theta_0}{\|\theta\|} \right)$$

Let's Talk About Classifier Quality

Math facts!



The signed distance from a hyperplane defined by θ, θ_0 to a point x^* is:

= projection of x^* on θ

− signed distance of line to origin

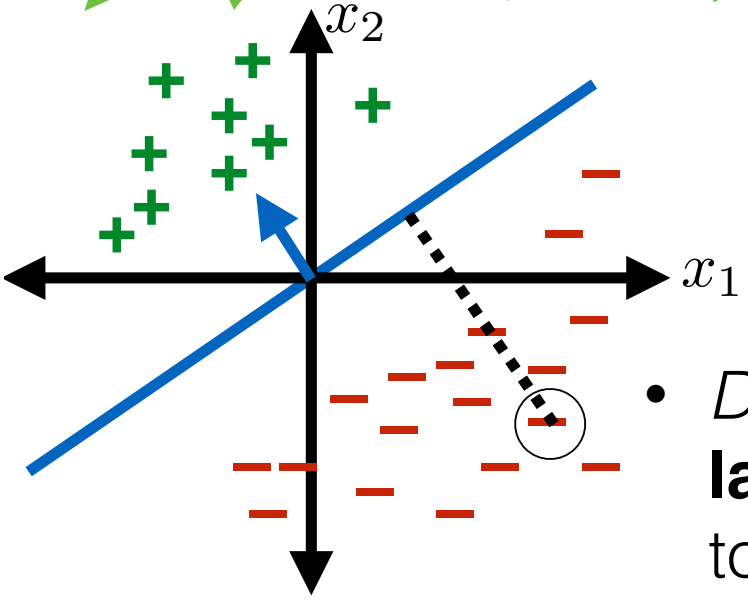
$$= \frac{\theta^\top x^*}{\|\theta\|} - \frac{-\theta_0}{\|\theta\|} = \frac{\theta^\top x^* + \theta_0}{\|\theta\|}$$

- *Definition*: The **margin of the labelled point** (x^*, y^*) with respect to the hyperplane defined by θ, θ_0 is:

$$y^* \left(\frac{\theta^\top x^* + \theta_0}{\|\theta\|} \right)$$

Let's Talk About Classifier Quality

Math facts!



The signed distance from a hyperplane defined by θ, θ_0 to a point x^* is:

= projection of x^* on θ

− signed distance of line to origin

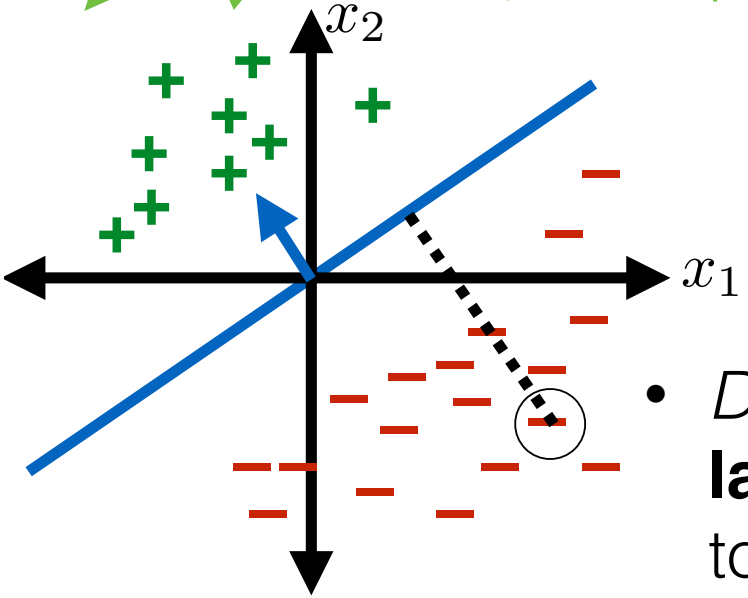
$$= \frac{\theta^\top x^*}{\|\theta\|} - \frac{-\theta_0}{\|\theta\|} = \frac{\theta^\top x^* + \theta_0}{\|\theta\|}$$

- *Definition*: The **margin of the labelled point** (x^*, y^*) with respect to the hyperplane defined by θ, θ_0 is:

$$y^* \left(\frac{\theta^\top x^* + \theta_0}{\|\theta\|} \right)$$

Let's Talk About Classifier Quality

Math facts!



The signed distance from a hyperplane defined by θ, θ_0 to a point x^* is:

= projection of x^* on θ

− signed distance of line to origin

$$= \frac{\theta^\top x^*}{\|\theta\|} - \frac{-\theta_0}{\|\theta\|} = \frac{\theta^\top x^* + \theta_0}{\|\theta\|}$$

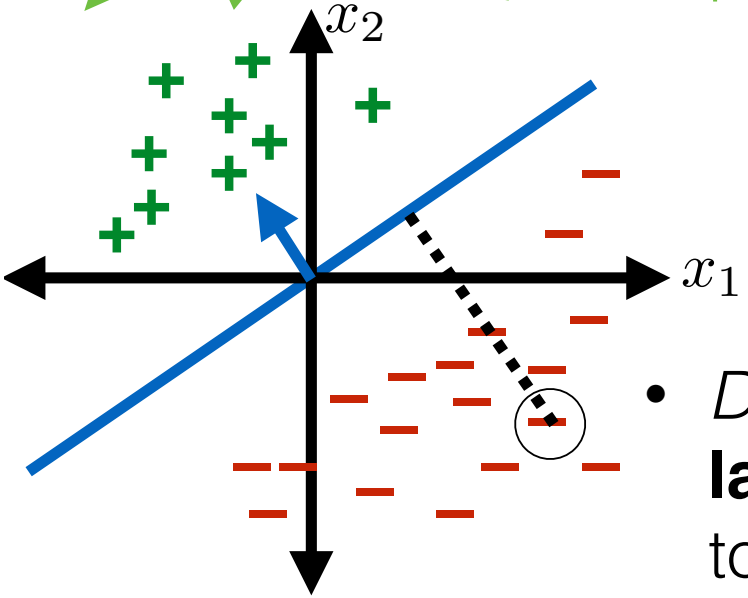
- *Definition*: The **margin of the labelled point** (x^*, y^*) with respect to the hyperplane defined by θ, θ_0 is:

$$y^* \left(\frac{\theta^\top x^* + \theta_0}{\|\theta\|} \right)$$

- *Definition*: The **margin of the training set** $\mathcal{D}_n$ with respect to the hyperplane defined by θ, θ_0 is:

Let's Talk About Classifier Quality

Math facts!



~~The signed distance from a~~ hyperplane defined by θ, θ_0 to a point x^* is:

= projection of x^* on θ

– signed distance of line to origin

$$= \frac{\theta^\top x^*}{\|\theta\|} - \frac{-\theta_0}{\|\theta\|} = \frac{\theta^\top x^* + \theta_0}{\|\theta\|}$$

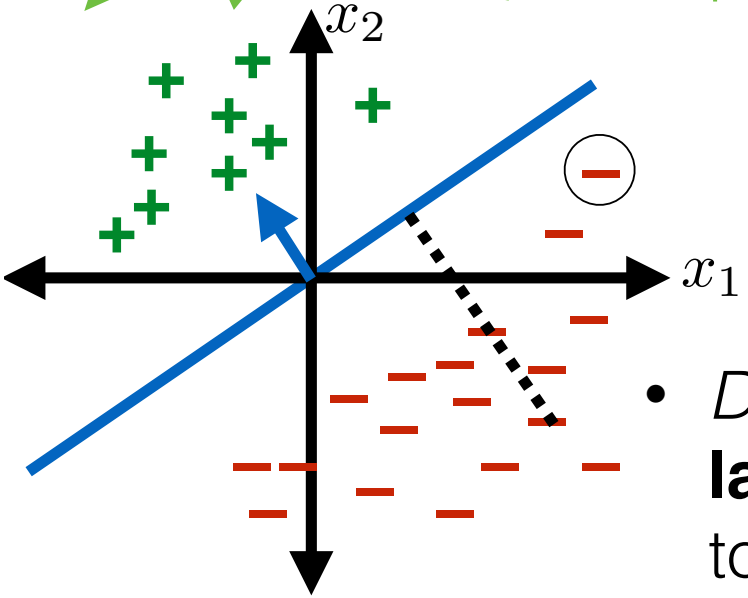
- *Definition*: The **margin of the labelled point** (x^*, y^*) with respect to the hyperplane defined by θ, θ_0 is:

$$y^* \left(\frac{\theta^\top x^* + \theta_0}{\|\theta\|} \right)$$

- *Definition*: The **margin of the training set** $\mathcal{D}_n$ with respect to the hyperplane defined by θ, θ_0 is: $\min_{i \in \{1,\ldots,n\}} y^{(i)} \left(\frac{\theta^\top x^{(i)} + \theta_0}{\|\theta\|} \right)$

Let's Talk About Classifier Quality

Math facts!



The signed distance from a hyperplane defined by θ, θ_0 to a point x^* is:

= projection of x^* on θ

− signed distance of line to origin

$$= \frac{\theta^\top x^*}{\|\theta\|} - \frac{-\theta_0}{\|\theta\|} = \frac{\theta^\top x^* + \theta_0}{\|\theta\|}$$

- *Definition*: The **margin of the labelled point** (x^*, y^*) with respect to the hyperplane defined by θ, θ_0 is:

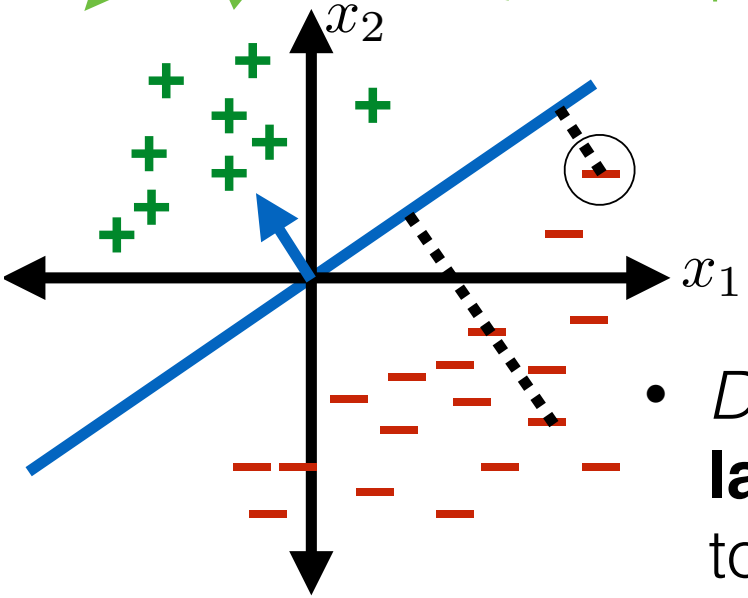
$$y^* \left(\frac{\theta^\top x^* + \theta_0}{\|\theta\|} \right)$$

- *Definition*: The **margin of the training set** $\mathcal{D}_n$ with respect to the hyperplane defined by θ, θ_0 is:

$$\min_{i \in \{1,\ldots,n\}} y^{(i)} \left(\frac{\theta^\top x^{(i)} + \theta_0}{\|\theta\|} \right)$$

Let's Talk About Classifier Quality

Math facts!



The signed distance from a hyperplane defined by θ, θ_0 to a point x^* is:

= projection of x^* on θ

− signed distance of line to origin

$$= \frac{\theta^\top x^*}{\|\theta\|} - \frac{-\theta_0}{\|\theta\|} = \frac{\theta^\top x^* + \theta_0}{\|\theta\|}$$

- *Definition*: The **margin of the labelled point** (x^*, y^*) with respect to the hyperplane defined by θ, θ_0 is:

$$y^* \left(\frac{\theta^\top x^* + \theta_0}{\|\theta\|} \right)$$

- *Definition*: The **margin of the training set** $\mathcal{D}_n$ with respect to the hyperplane defined by θ, θ_0 is:

$$\min_{i \in \{1,\ldots,n\}} y^{(i)} \left(\frac{\theta^\top x^{(i)} + \theta_0}{\|\theta\|} \right)$$

Why classifiers through the origin?

Why classifiers through the origin?

- If we're clever, we don't lose any flexibility

Why classifiers through the origin?

- If we're clever, we don't lose any flexibility
  - Classifier with offset

Why classifiers through the origin?

- If we're clever, we don't lose any flexibility
  - Classifier with offset

$$x \in \mathbb{R}^d, \theta \in \mathbb{R}^d, \theta_0 \in \mathbb{R}$$

$$x : \theta^\top x + \theta_0 = 0$$

Why classifiers through the origin?

- If we're clever, we don't lose any flexibility
  - Classifier with offset

$$x \in \mathbb{R}^d, \theta \in \mathbb{R}^d, \theta_0 \in \mathbb{R}$$

$$x : \theta^\top x + \theta_0 = 0$$

  - Classifier without offset

Why classifiers through the origin?

- If we're clever, we don't lose any flexibility
  - Classifier with offset

$$x \in \mathbb{R}^d, \theta \in \mathbb{R}^d, \theta_0 \in \mathbb{R}$$

$$x : \theta^\top x + \theta_0 = 0$$

  - Classifier without offset

$$x_{\text{new}} \in \mathbb{R}^{d+1}, \theta_{\text{new}} \in \mathbb{R}^{d+1}$$

$$x_{\text{new}} = [x_1, x_2, \ldots, x_d, 1]^\top, \theta_{\text{new}} = [\theta_1, \theta_2, \ldots, \theta_d, \theta_0]^\top$$

Why classifiers through the origin?

- If we're clever, we don't lose any flexibility
  - Classifier with offset

$$x \in \mathbb{R}^d, \theta \in \mathbb{R}^d, \theta_0 \in \mathbb{R}$$

$$x : \theta^\top x + \theta_0 = 0$$

  - Classifier without offset

$$x_{\text{new}} \in \mathbb{R}^{d+1}, \theta_{\text{new}} \in \mathbb{R}^{d+1}$$

$$x_{\text{new}} = [x_1, x_2, \ldots, x_d, 1]^\top, \theta_{\text{new}} = [\theta_1, \theta_2, \ldots, \theta_d, \theta_0]^\top$$

$$x_{\text{new},1:d} : \theta_{\text{new}}^\top x_{\text{new}} = 0$$

Why classifiers through the origin?

- If we're clever, we don't lose any flexibility
  - Classifier with offset

$$x \in \mathbb{R}^d, \theta \in \mathbb{R}^d, \theta_0 \in \mathbb{R}$$

$$x : \theta^\top x + \theta_0 = 0$$

  - Classifier without offset

$$x_{\text{new}} \in \mathbb{R}^{d+1}, \theta_{\text{new}} \in \mathbb{R}^{d+1}$$

$$x_{\text{new}} = [x_1, x_2, \ldots, x_d, 1]^\top, \theta_{\text{new}} = [\theta_1, \theta_2, \ldots, \theta_d, \theta_0]^\top$$

$$x_{\text{new},1:d} : \theta_{\text{new}}^\top x_{\text{new}} = 0$$

Why classifiers through the origin?

- If we're clever, we don't lose any flexibility
  - Classifier with offset

$$x \in \mathbb{R}^d, \theta \in \mathbb{R}^d, \theta_0 \in \mathbb{R}$$

$$x : \theta^\top x + \theta_0 \leq 0$$

  - Classifier without offset

$$x_{\text{new}} \in \mathbb{R}^{d+1}, \theta_{\text{new}} \in \mathbb{R}^{d+1}$$

$$x_{\text{new}} = [x_1, x_2, \ldots, x_d, 1]^\top, \theta_{\text{new}} = [\theta_1, \theta_2, \ldots, \theta_d, \theta_0]^\top$$

$$x_{\text{new},1:d} : \theta_{\text{new}}^\top x_{\text{new}} \leq 0$$

Why classifiers through the origin?

- If we're clever, we don't lose any flexibility
  - Classifier with offset

$$x \in \mathbb{R}^d, \theta \in \mathbb{R}^d, \theta_0 \in \mathbb{R}$$

$$x : \theta^\top x + \theta_0 \overset{<}{=} 0$$
$$>$$

  - Classifier without offset

$$x_{\text{new}} \in \mathbb{R}^{d+1}, \theta_{\text{new}} \in \mathbb{R}^{d+1}$$

$$x_{\text{new}} = [x_1, x_2, \ldots, x_d, 1]^\top, \theta_{\text{new}} = [\theta_1, \theta_2, \ldots, \theta_d, \theta_0]^\top$$

$$x_{\text{new},1:d} : \theta_{\text{new}}^\top x_{\text{new}} \overset{<}{=} 0$$
$$>$$

Why classifiers through the origin?

- If we're clever, we don't lose any flexibility
  - Classifier with offset

$$x \in \mathbb{R}^d, \theta \in \mathbb{R}^d, \theta_0 \in \mathbb{R}$$

$$x : \theta^\top x + \theta_0 \begin{matrix} < \\ = \\ > \end{matrix} 0$$

  - Classifier without offset

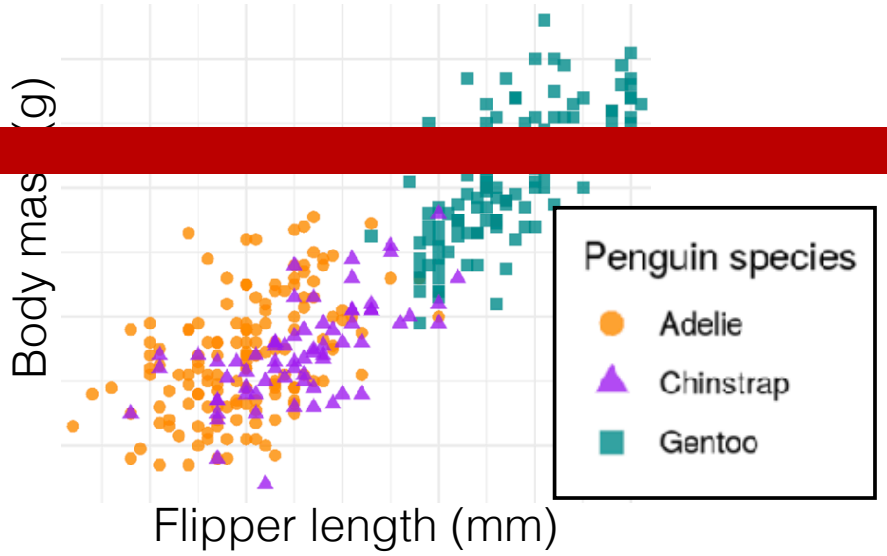
$$x_{\text{new}} \in \mathbb{R}^{d+1}, \theta_{\text{new}} \in \mathbb{R}^{d+1}$$

$$x_{\text{new}} = [x_1, x_2, \ldots, x_d, 1]^\top, \theta_{\text{new}} = [\theta_1, \theta_2, \ldots, \theta_d, \theta_0]^\top$$

$$x_{\text{new},1:d} : \theta_{\text{new}}^\top x_{\text{new}} \begin{matrix} < \\ = \\ > \end{matrix} 0$$

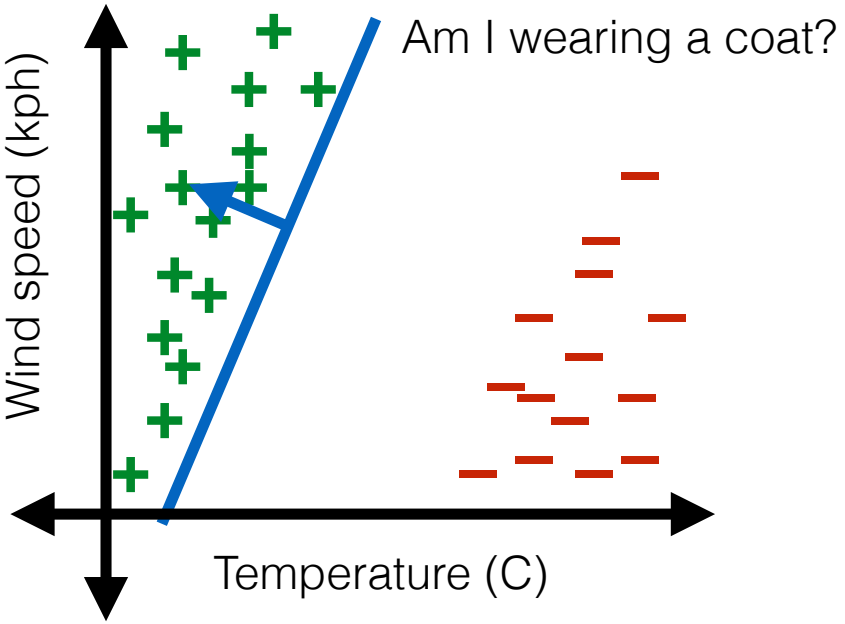
Recall

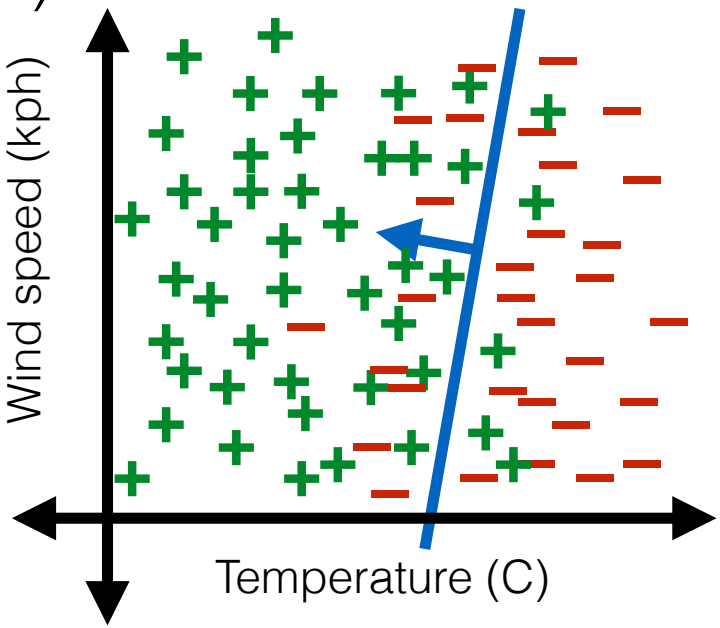
- Perceptron struggles with data that's not linearly separable



Notice

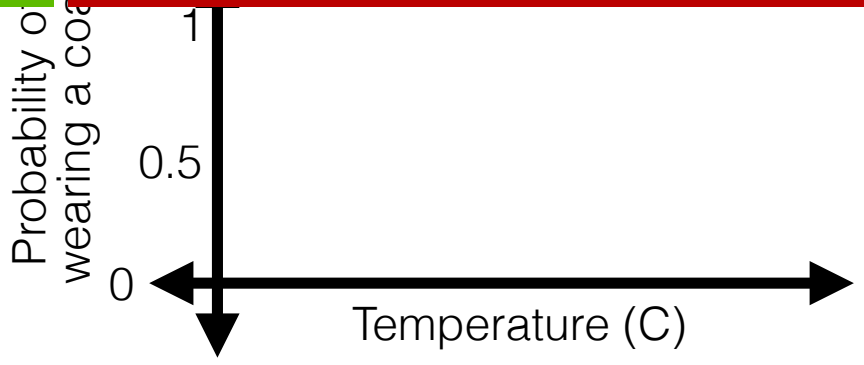
- Perceptron doesn't have a notion of uncertainty (how well do we know what we know?)



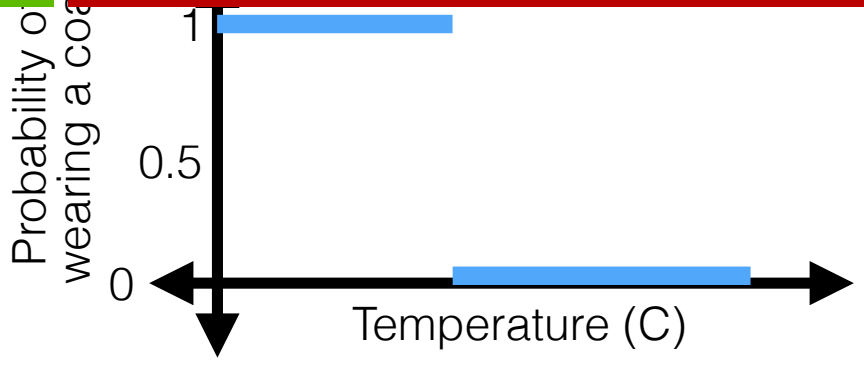


Capturing uncertainty

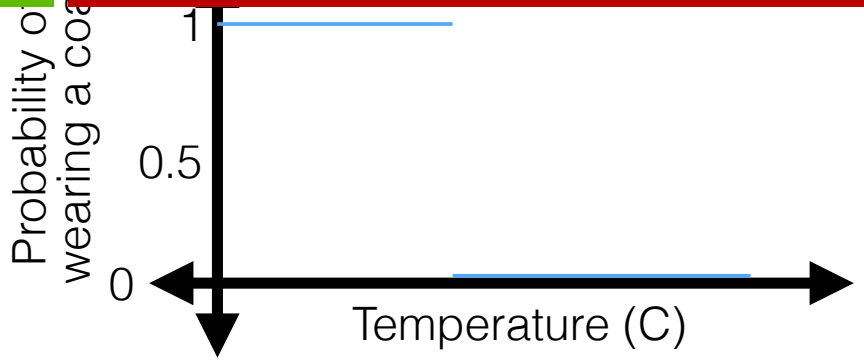
Capturing uncertainty



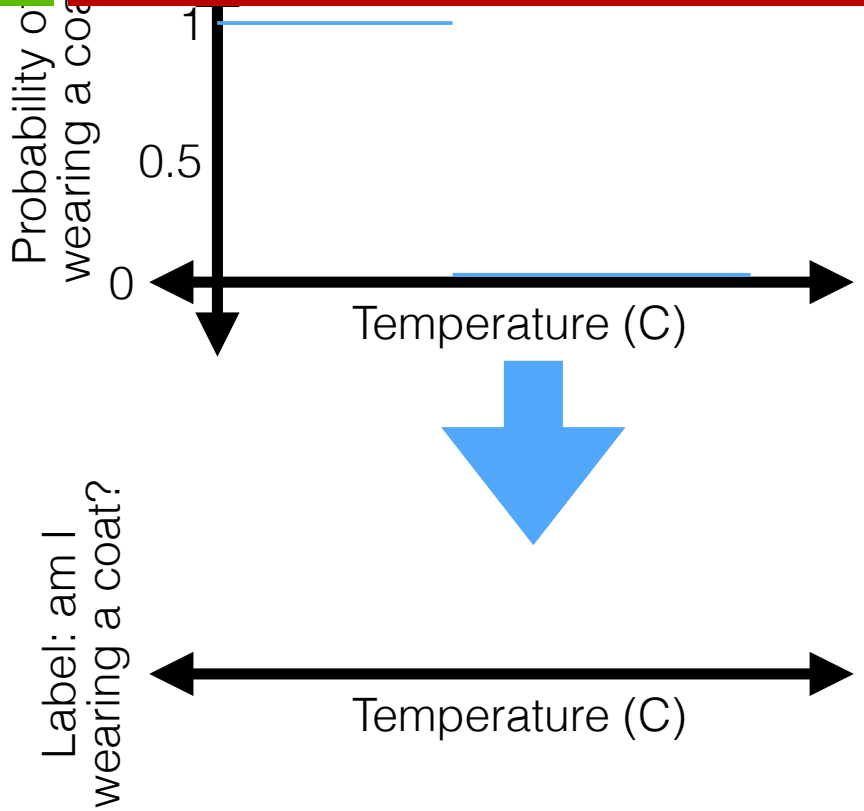
Capturing uncertainty



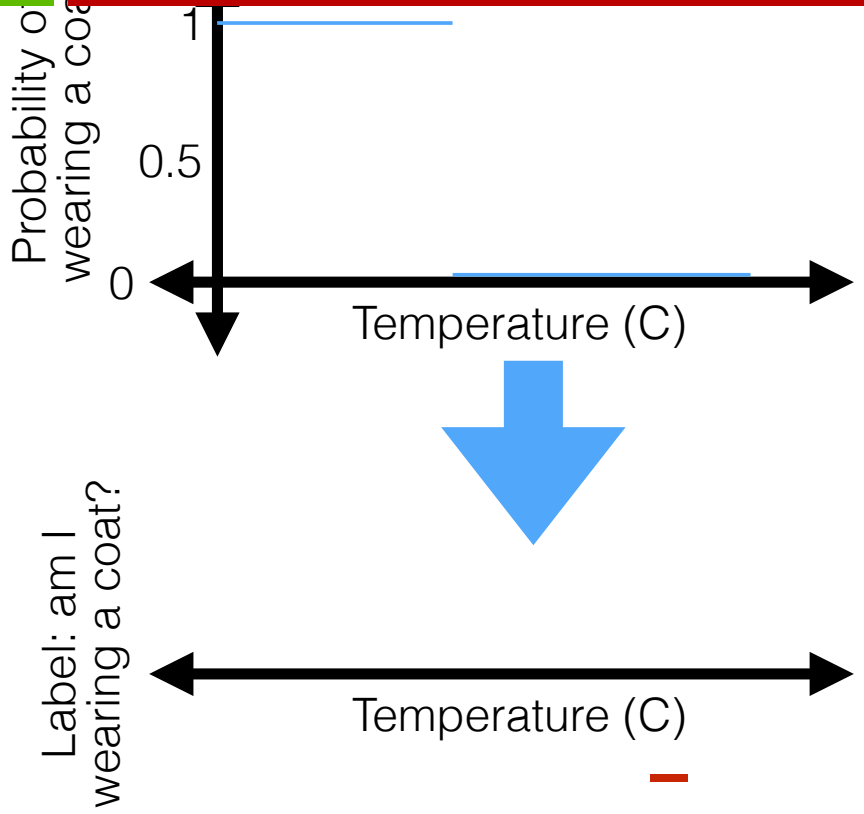
Capturing uncertainty



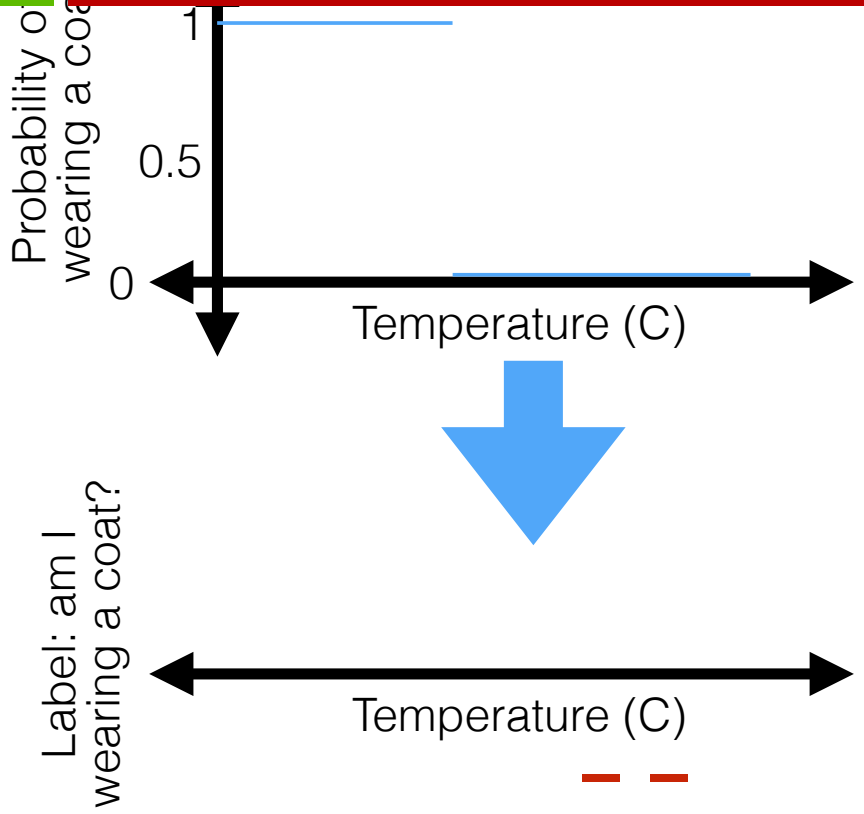
Capturing uncertainty



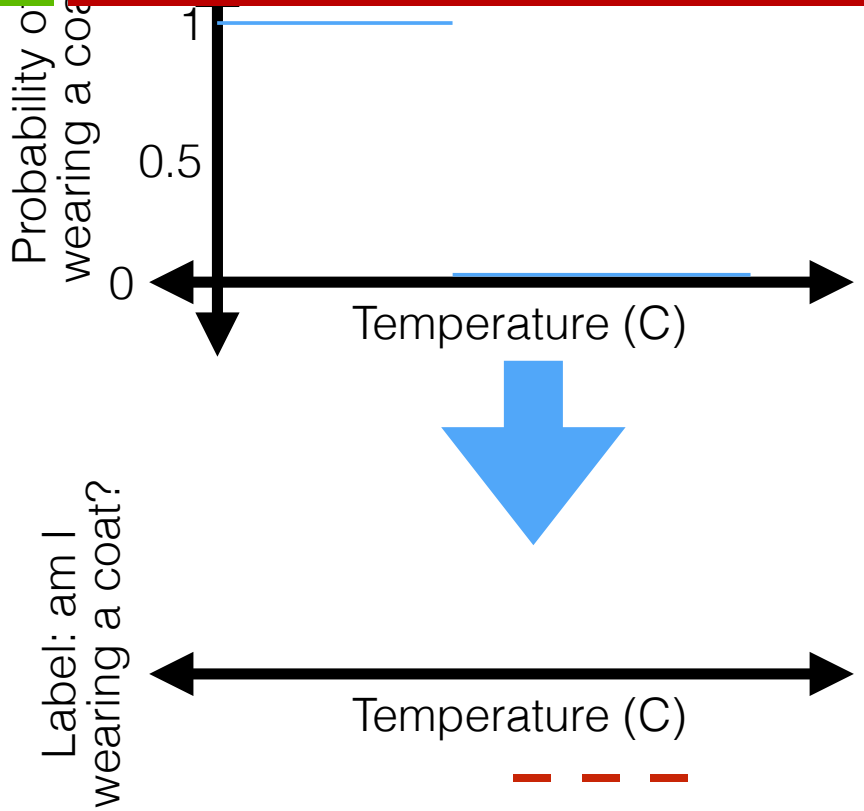
Capturing uncertainty



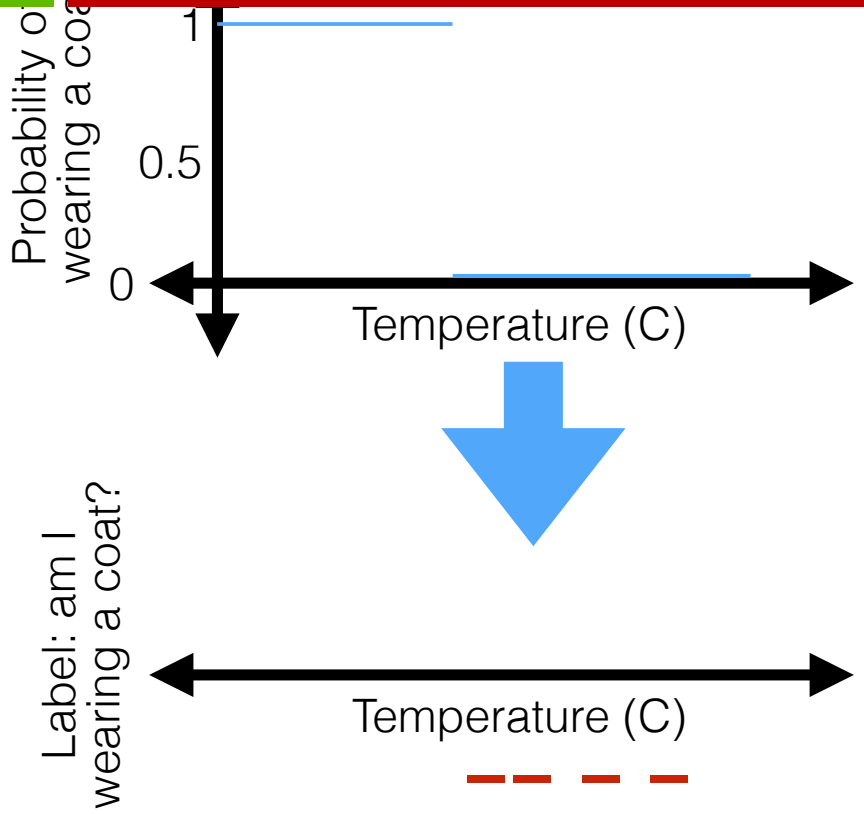
Capturing uncertainty



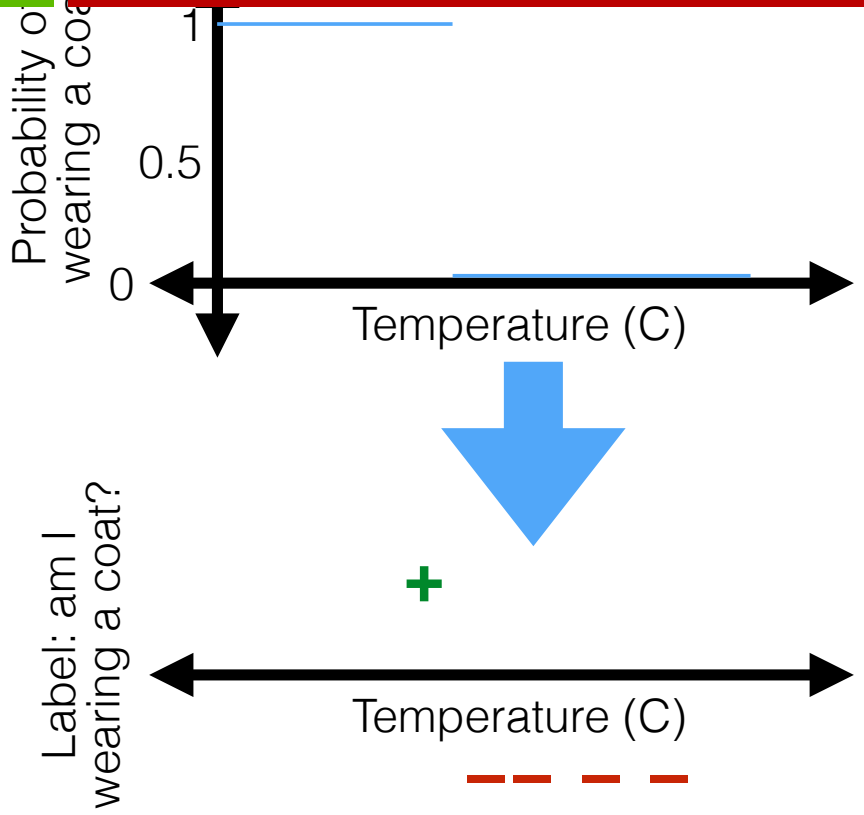
Capturing uncertainty



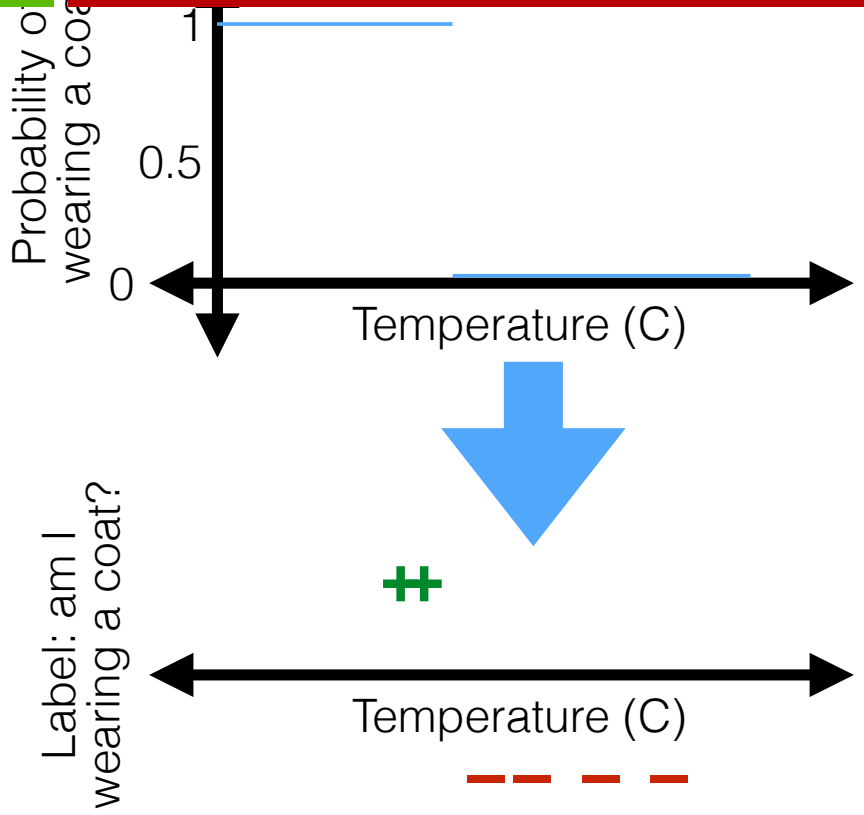
Capturing uncertainty



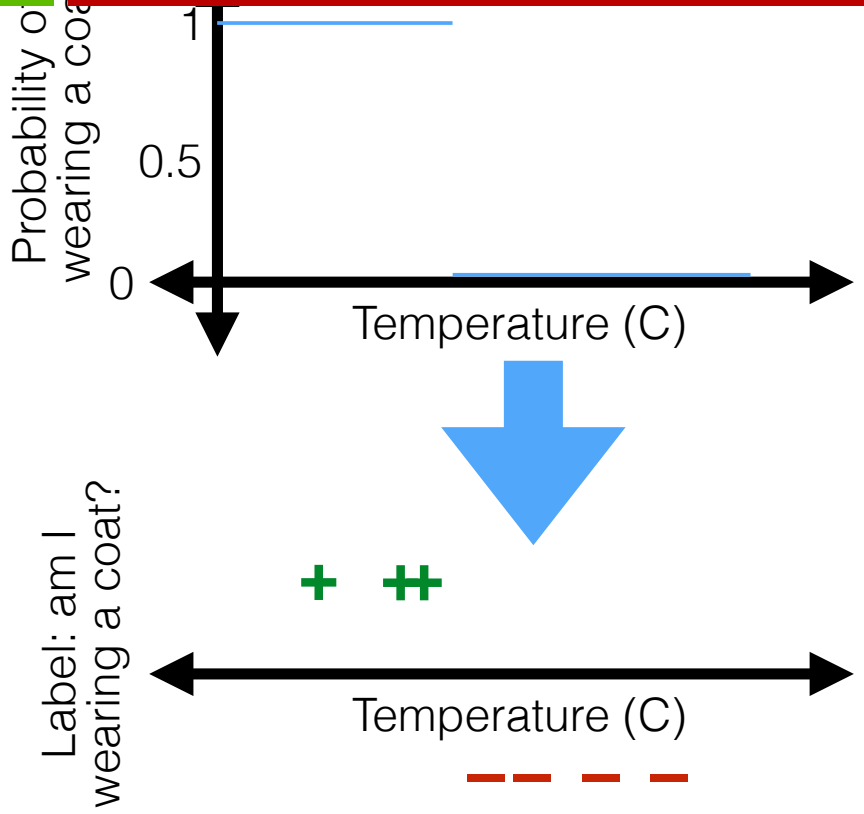
Capturing uncertainty



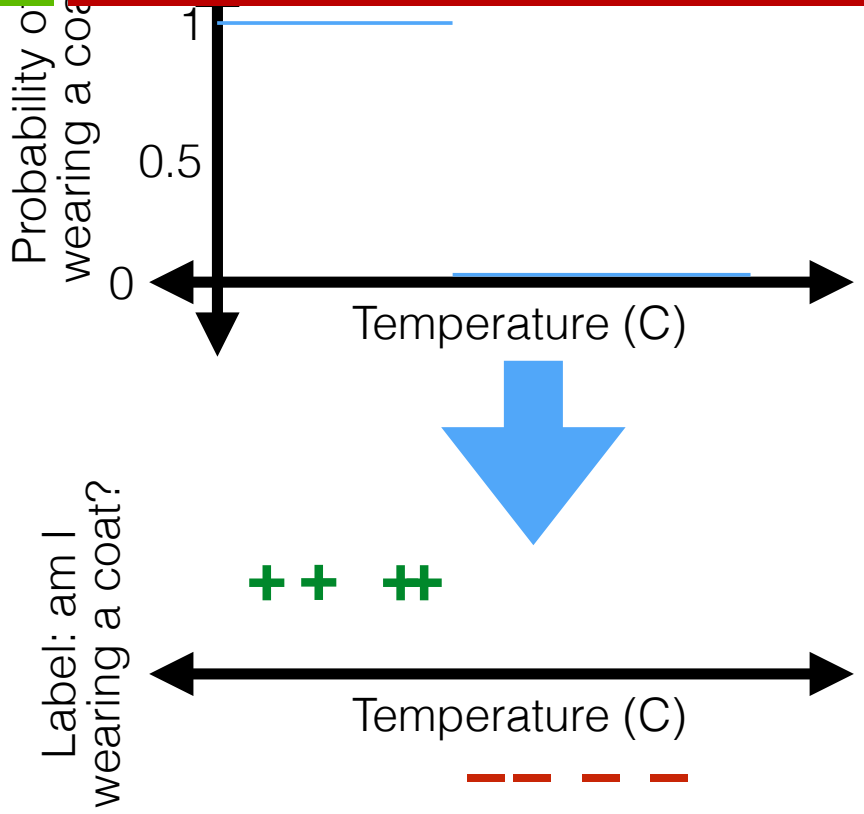
Capturing uncertainty



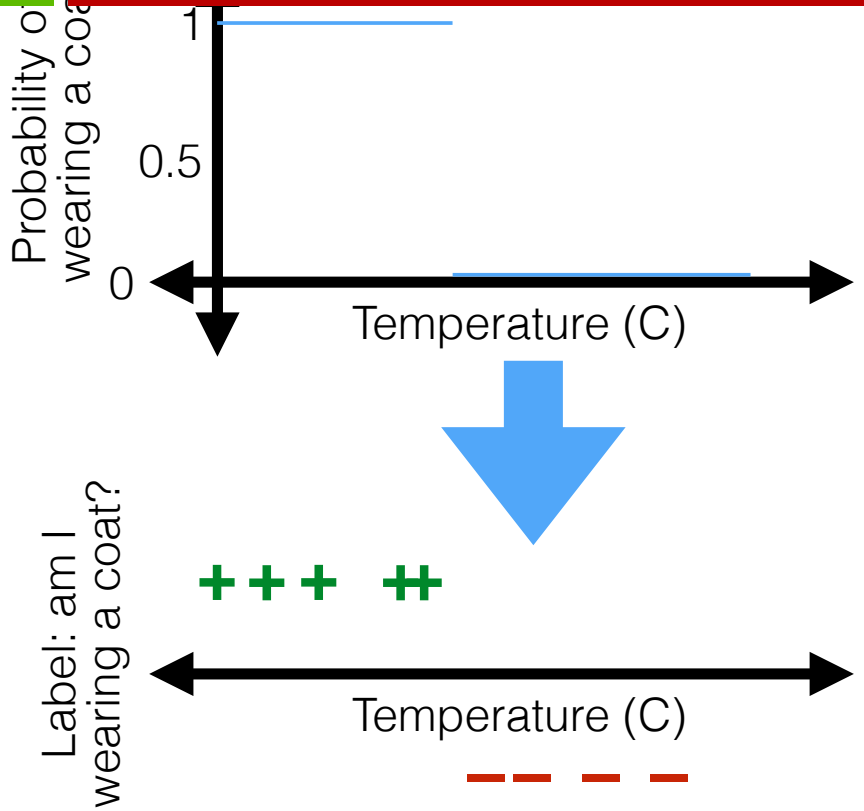
Capturing uncertainty



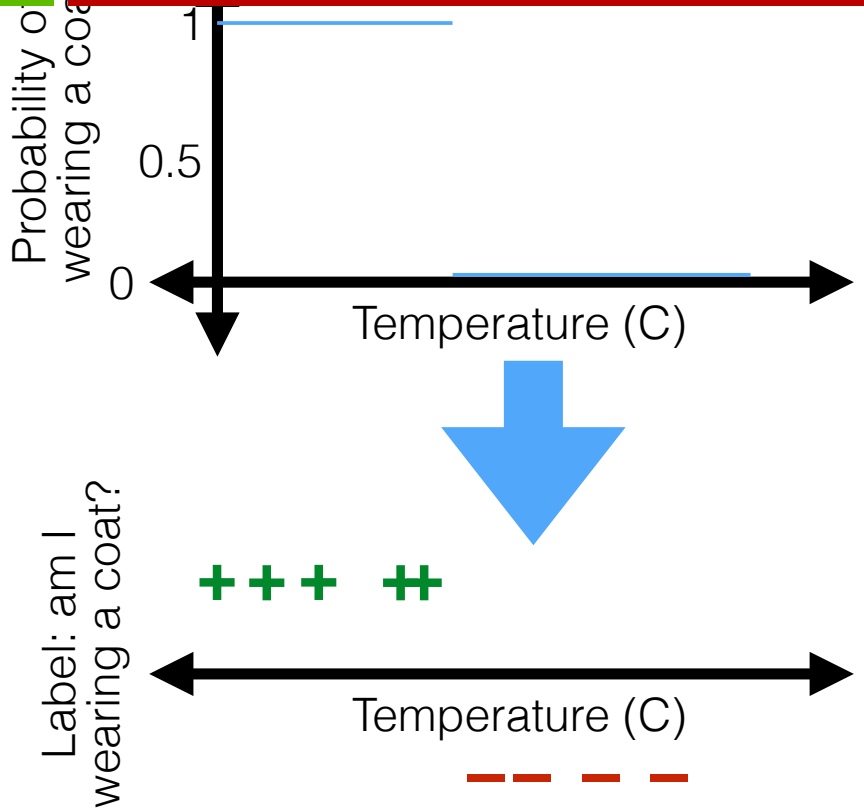
Capturing uncertainty



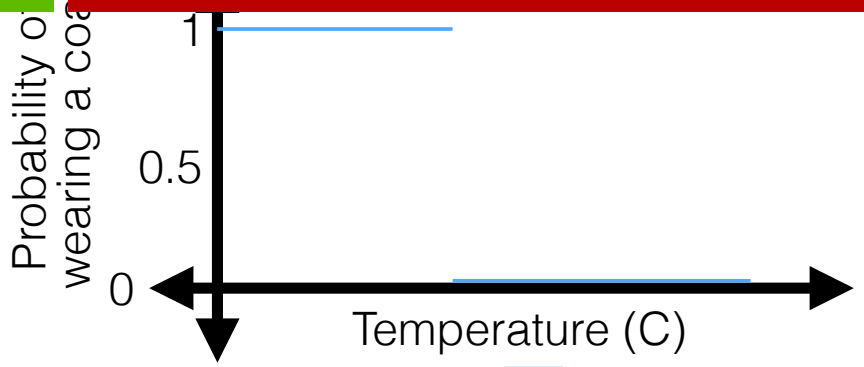
Capturing uncertainty

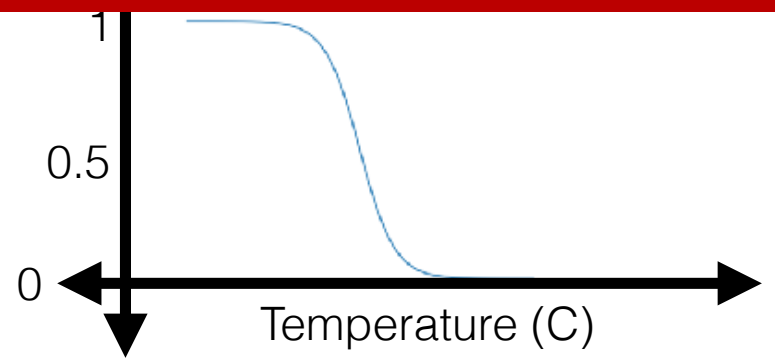


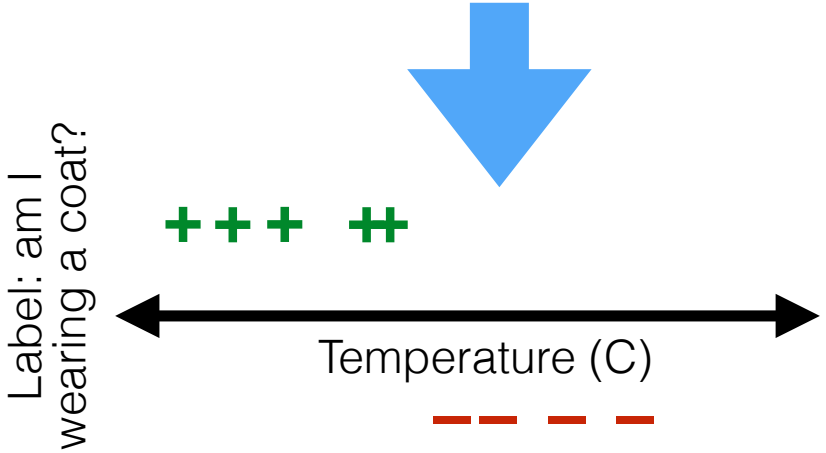
Capturing uncertainty



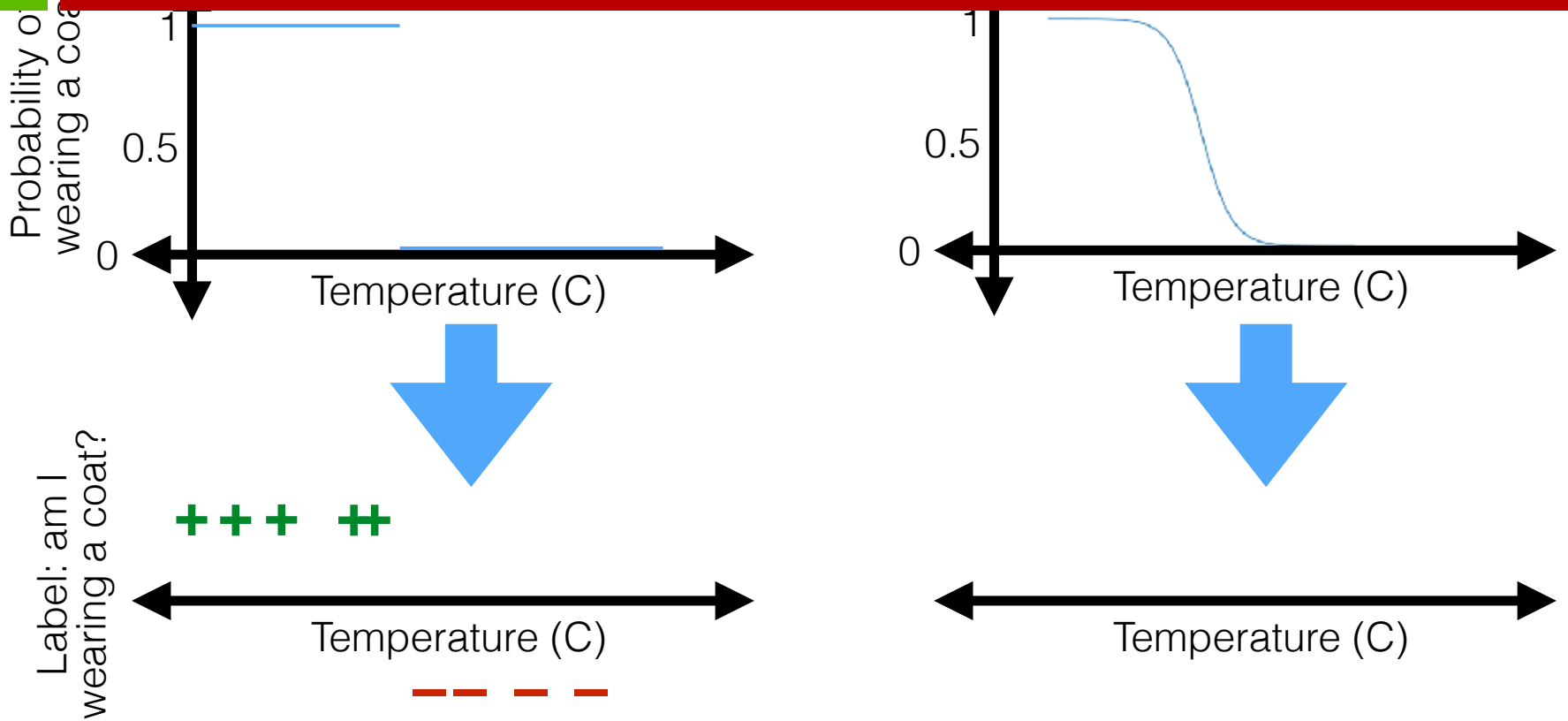
Capturing uncertainty



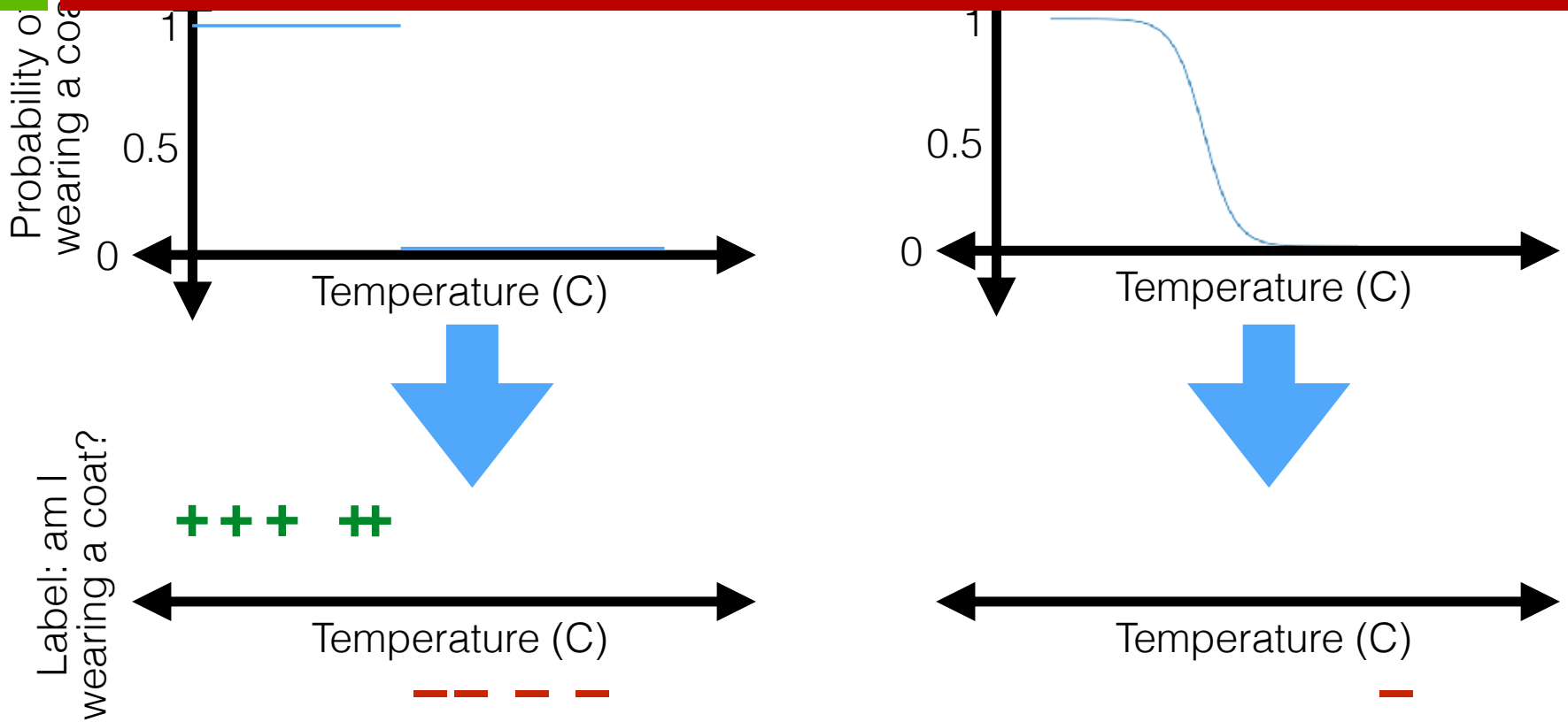




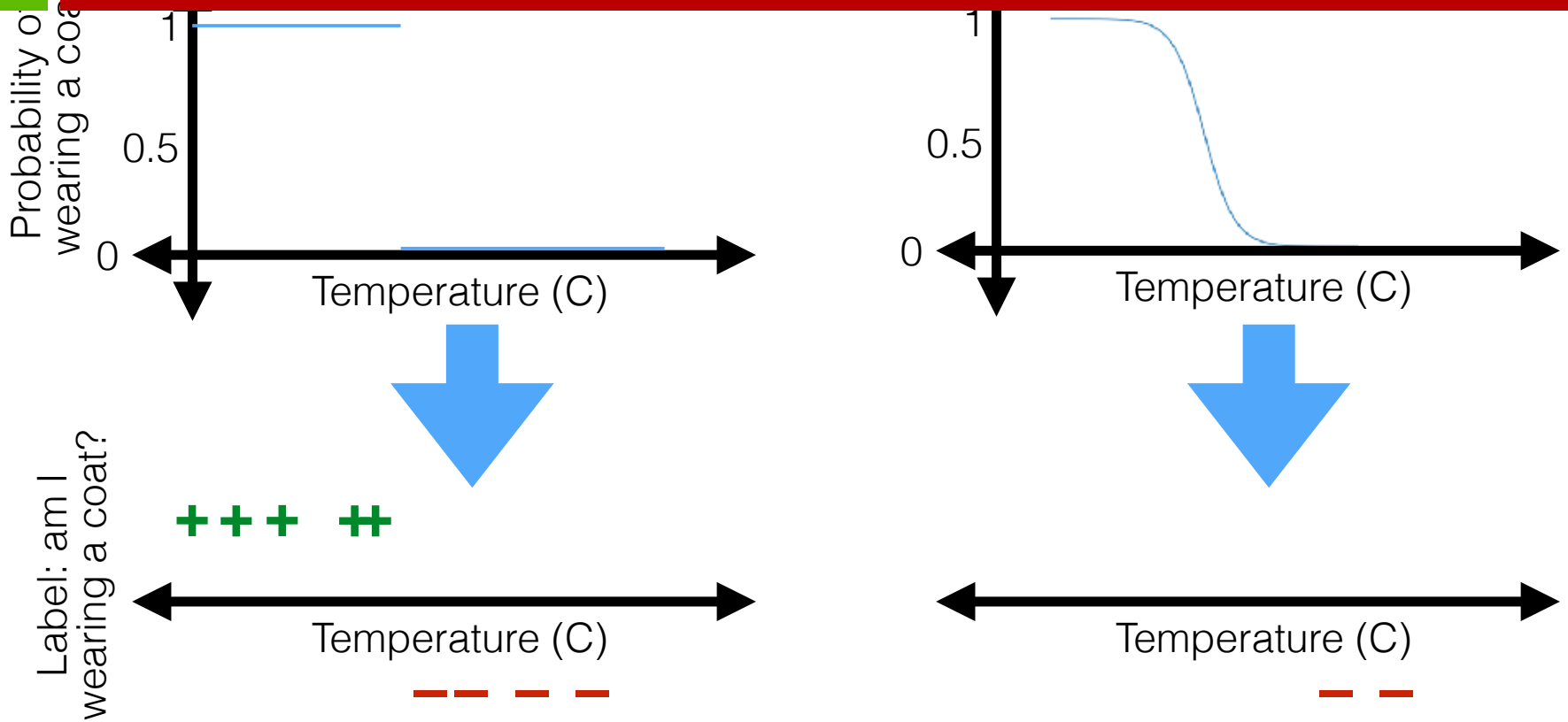
Capturing uncertainty



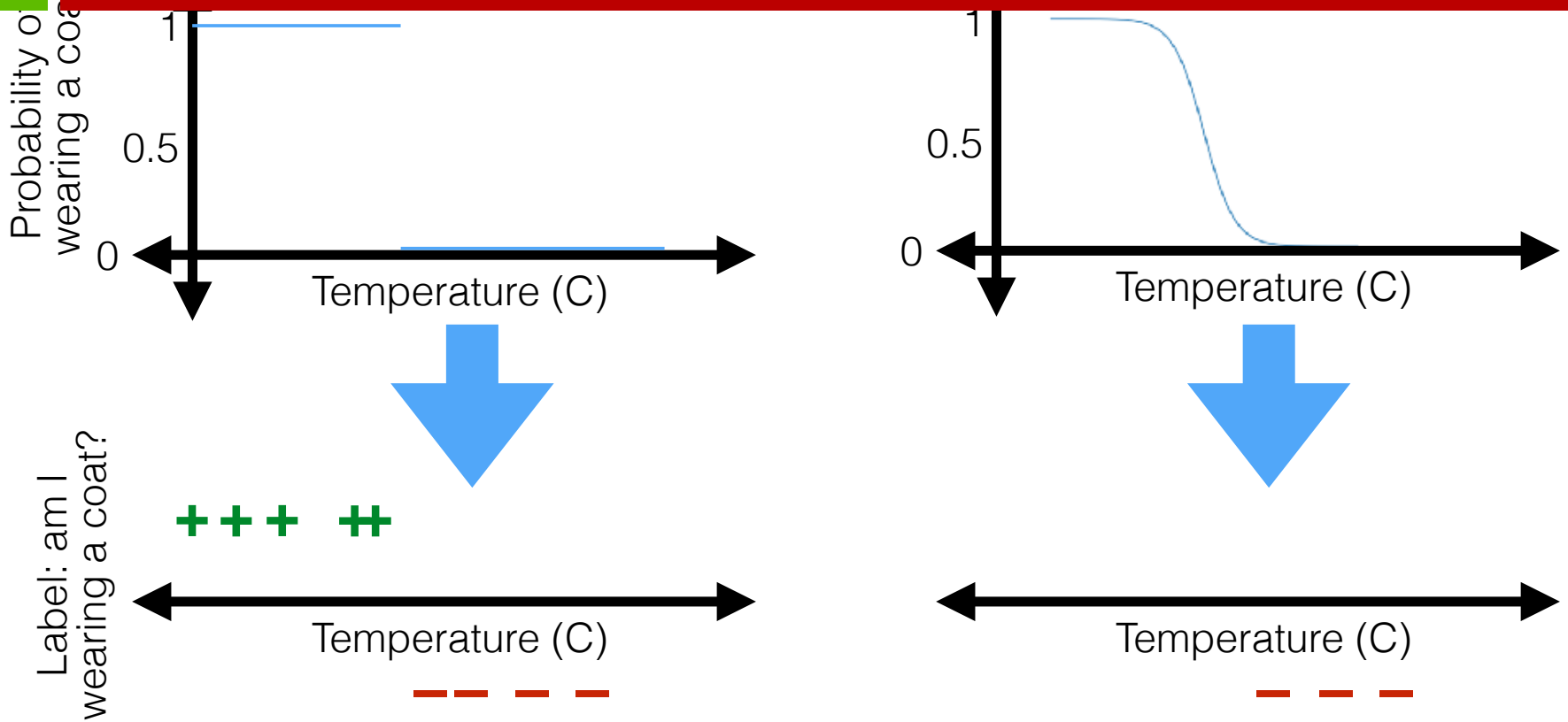
Capturing uncertainty



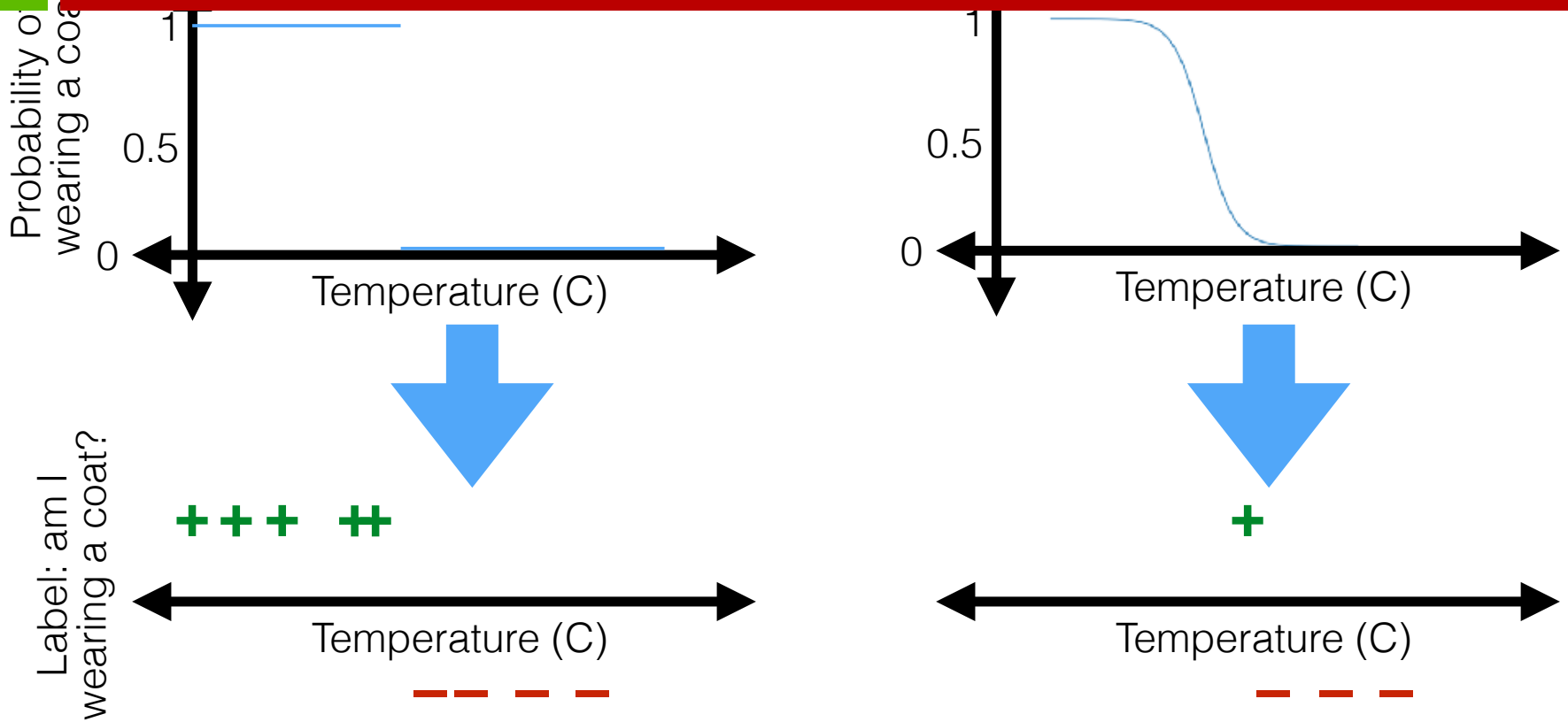
Capturing uncertainty



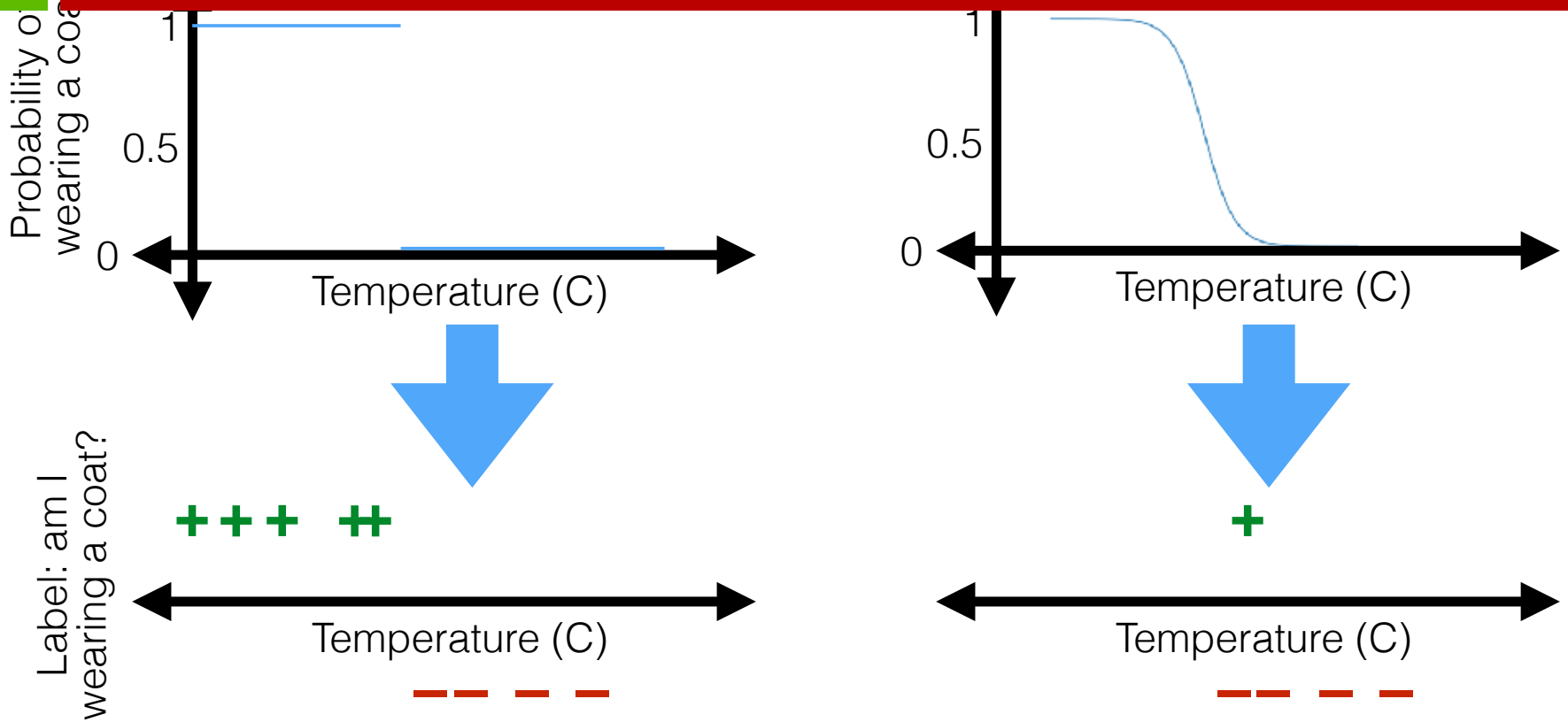
Capturing uncertainty



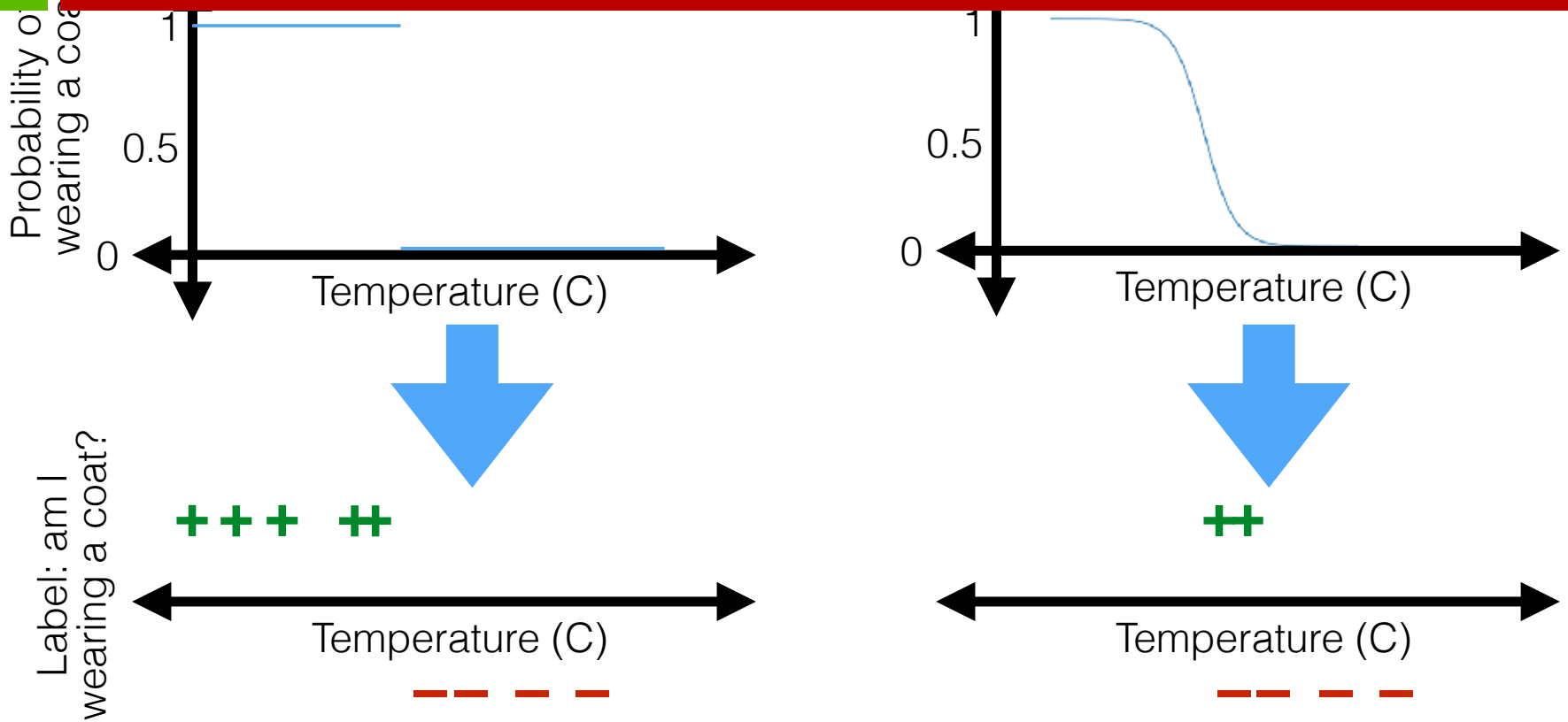
Capturing uncertainty



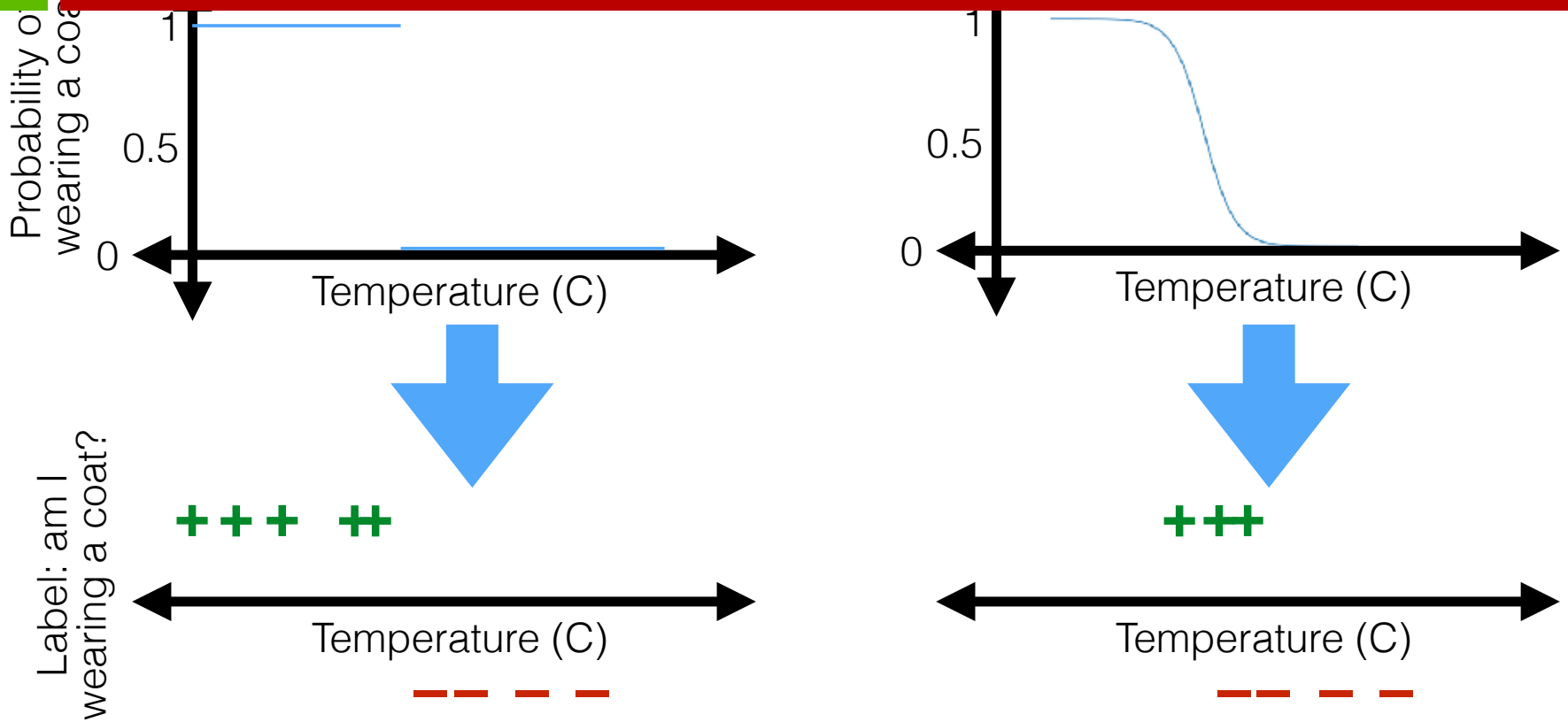
Capturing uncertainty



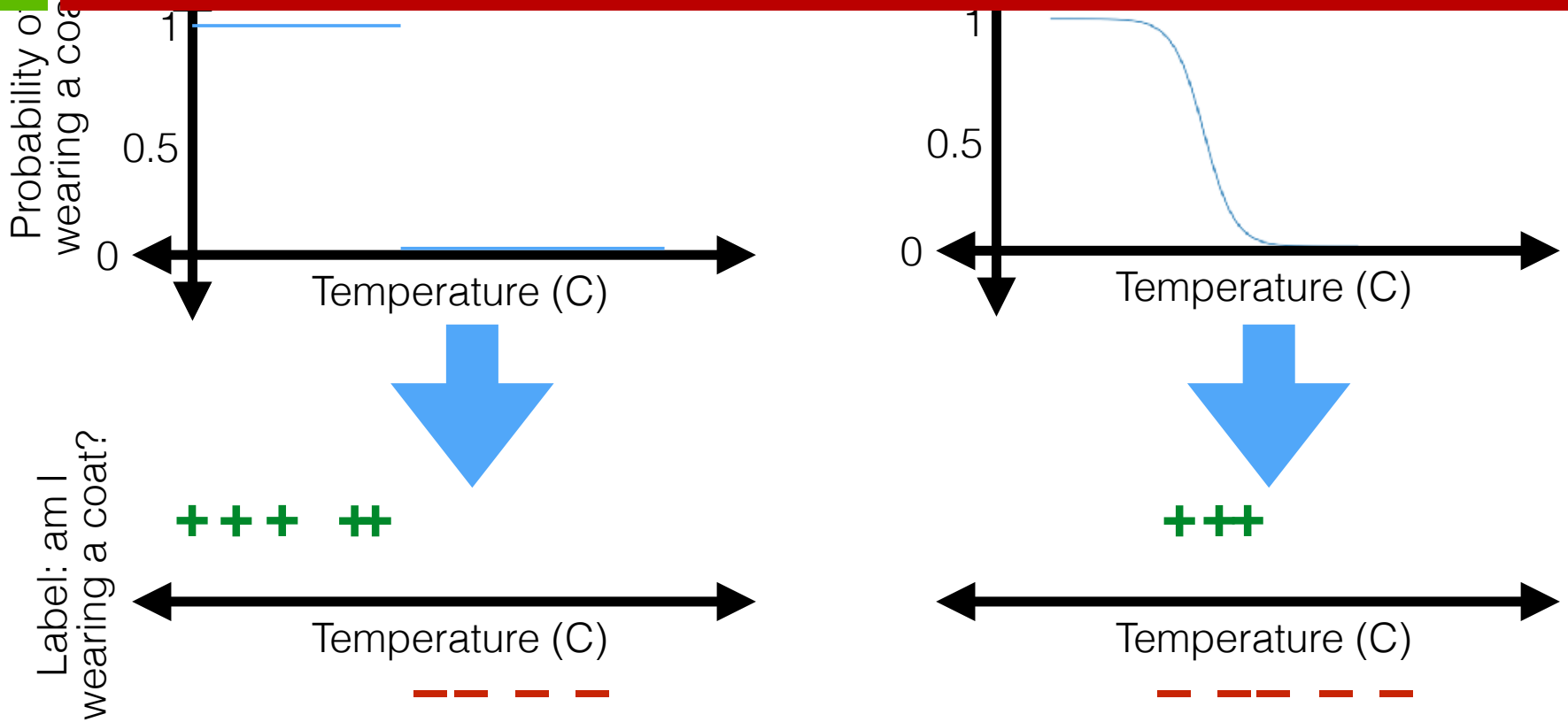
Capturing uncertainty



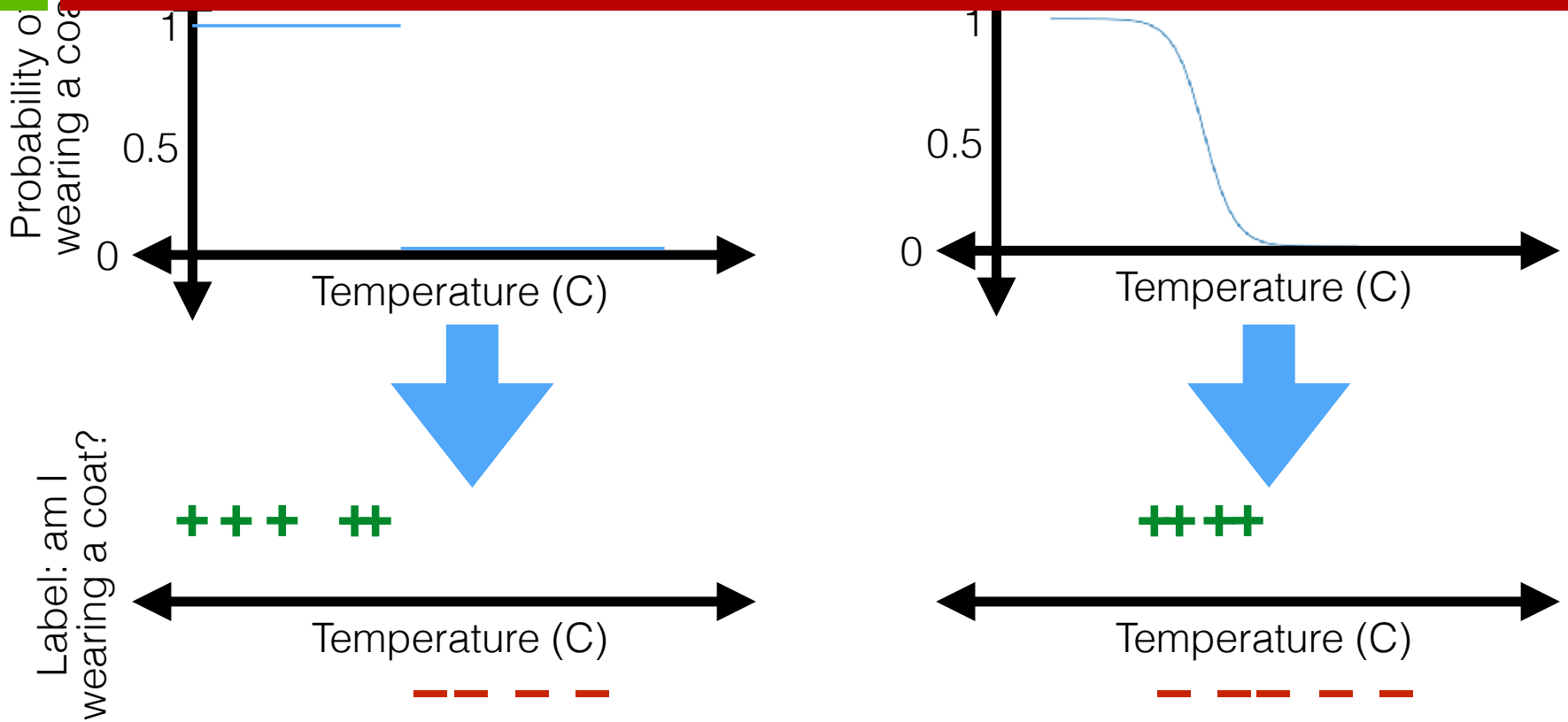
Capturing uncertainty



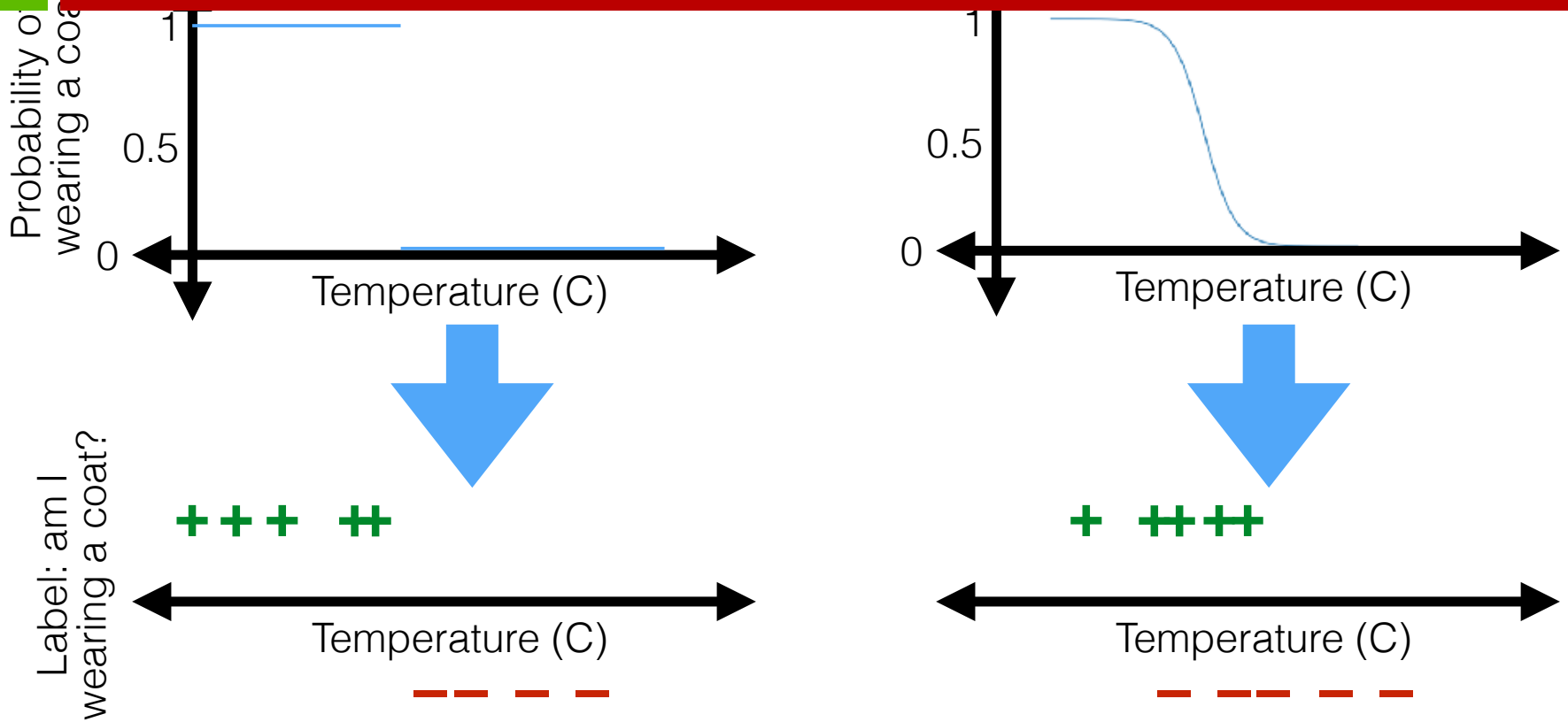
Capturing uncertainty



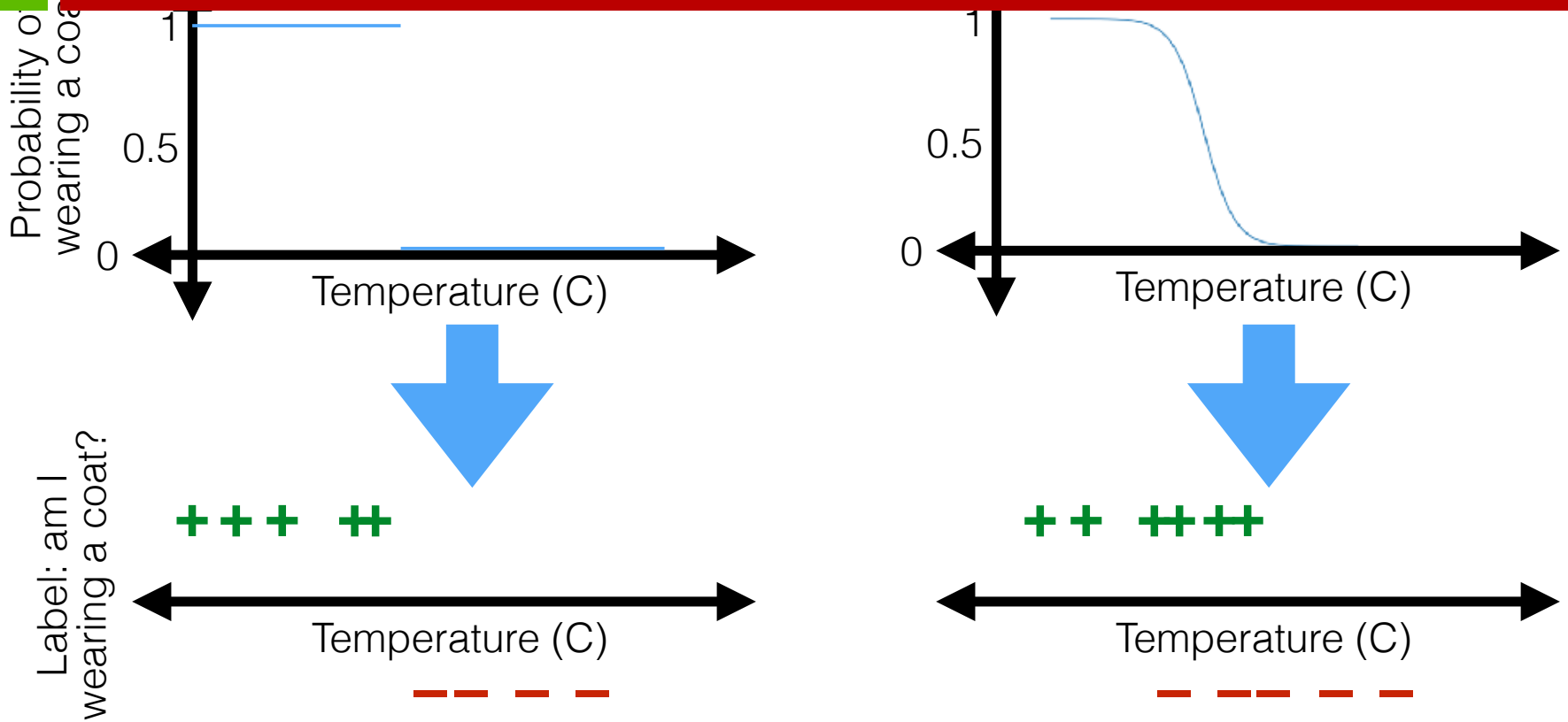
Capturing uncertainty



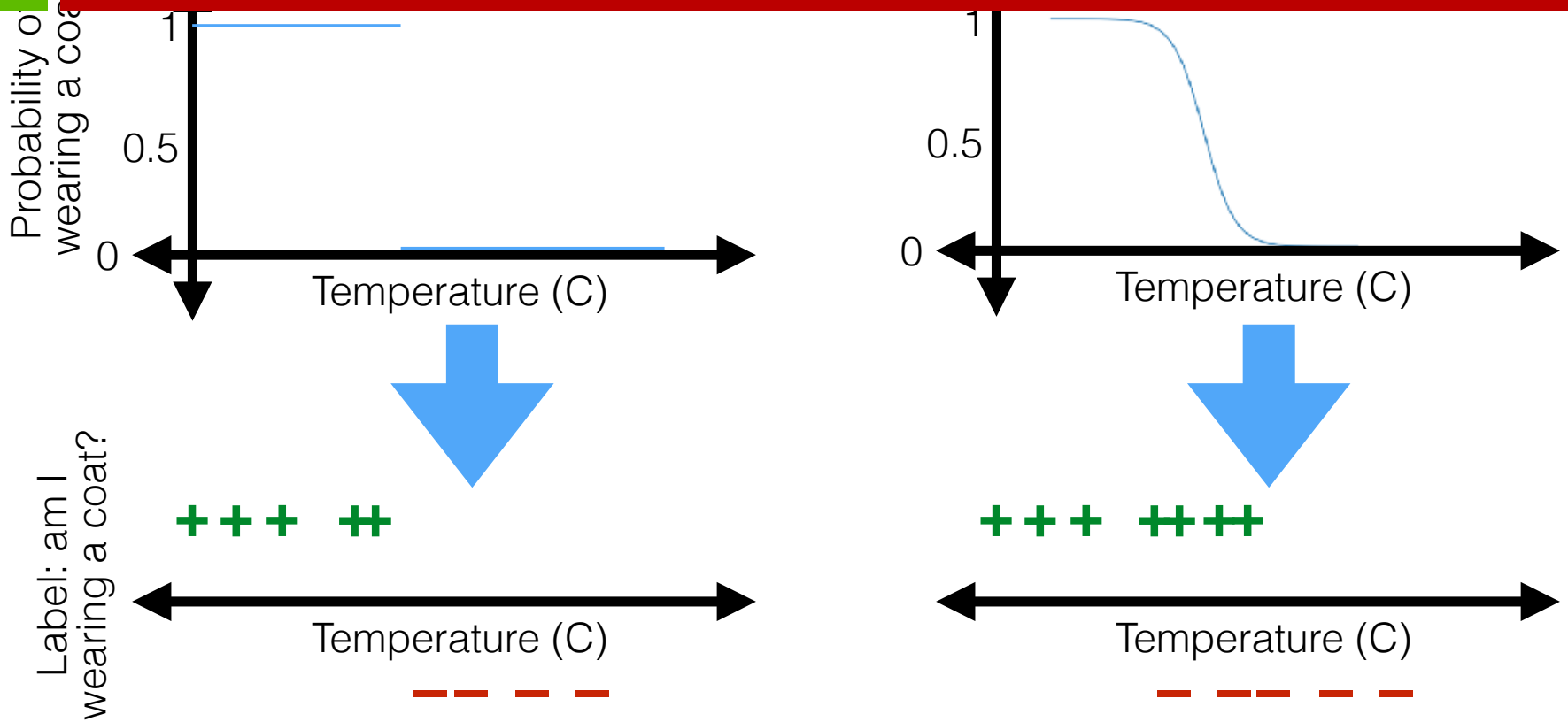
Capturing uncertainty



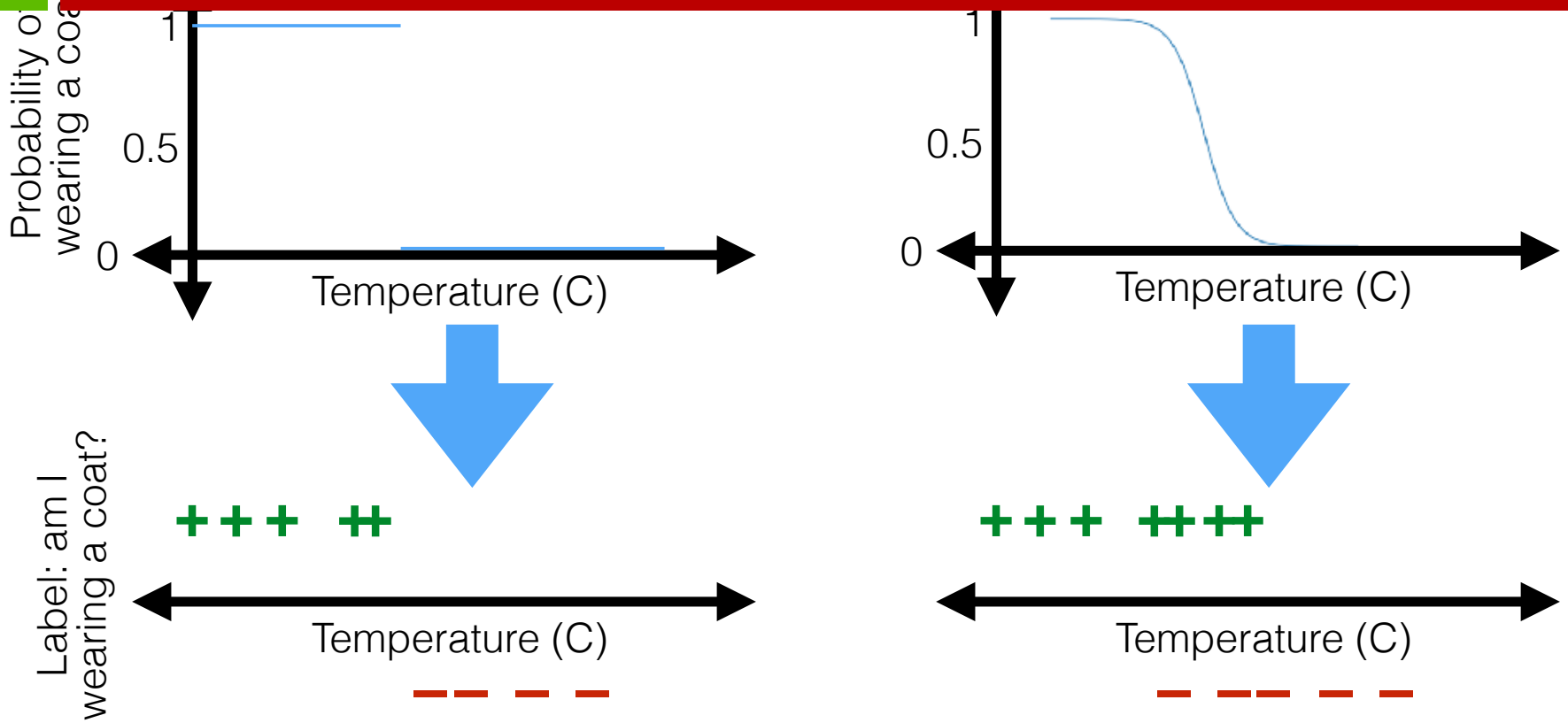
Capturing uncertainty



Capturing uncertainty

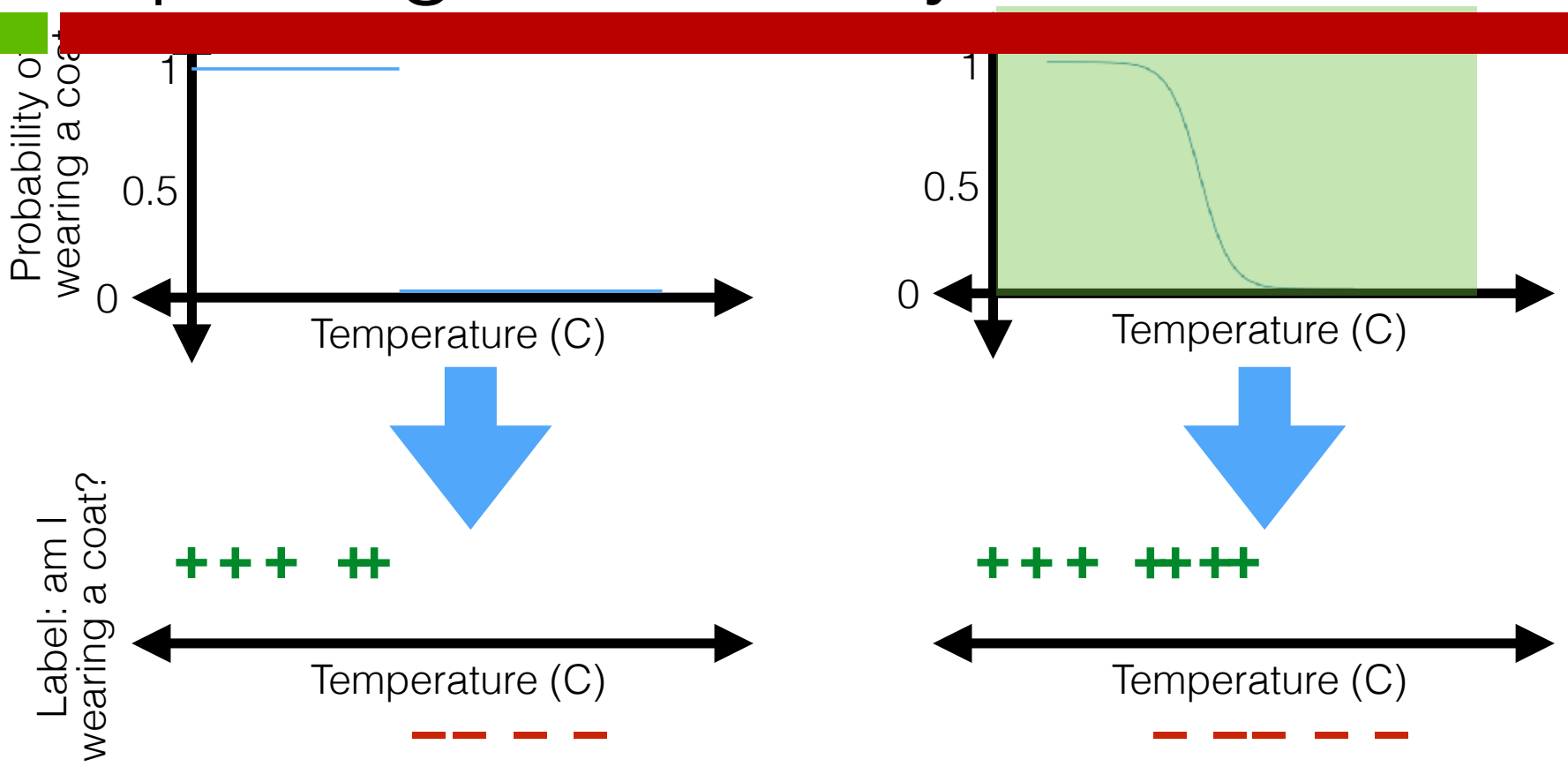


Capturing uncertainty



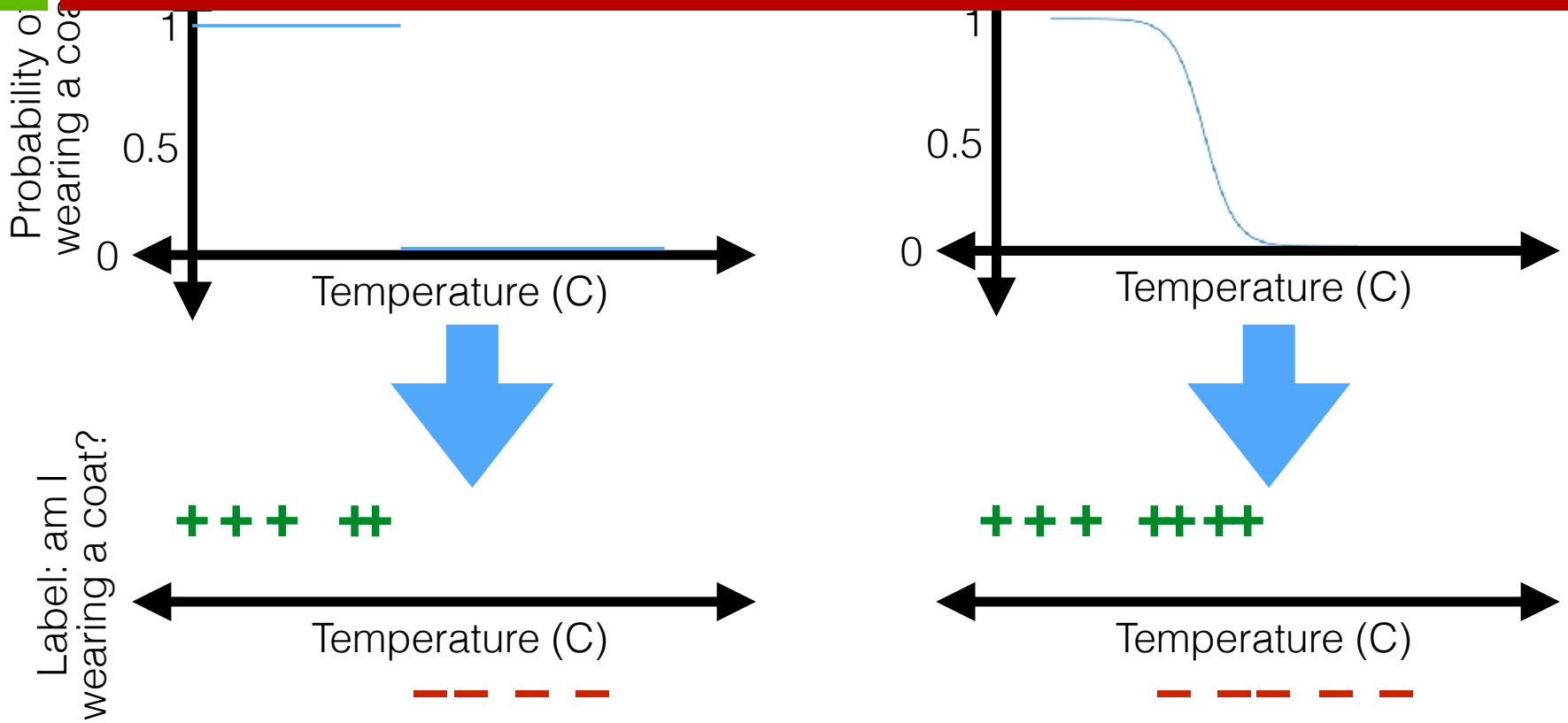
- How to make this shape?

Capturing uncertainty



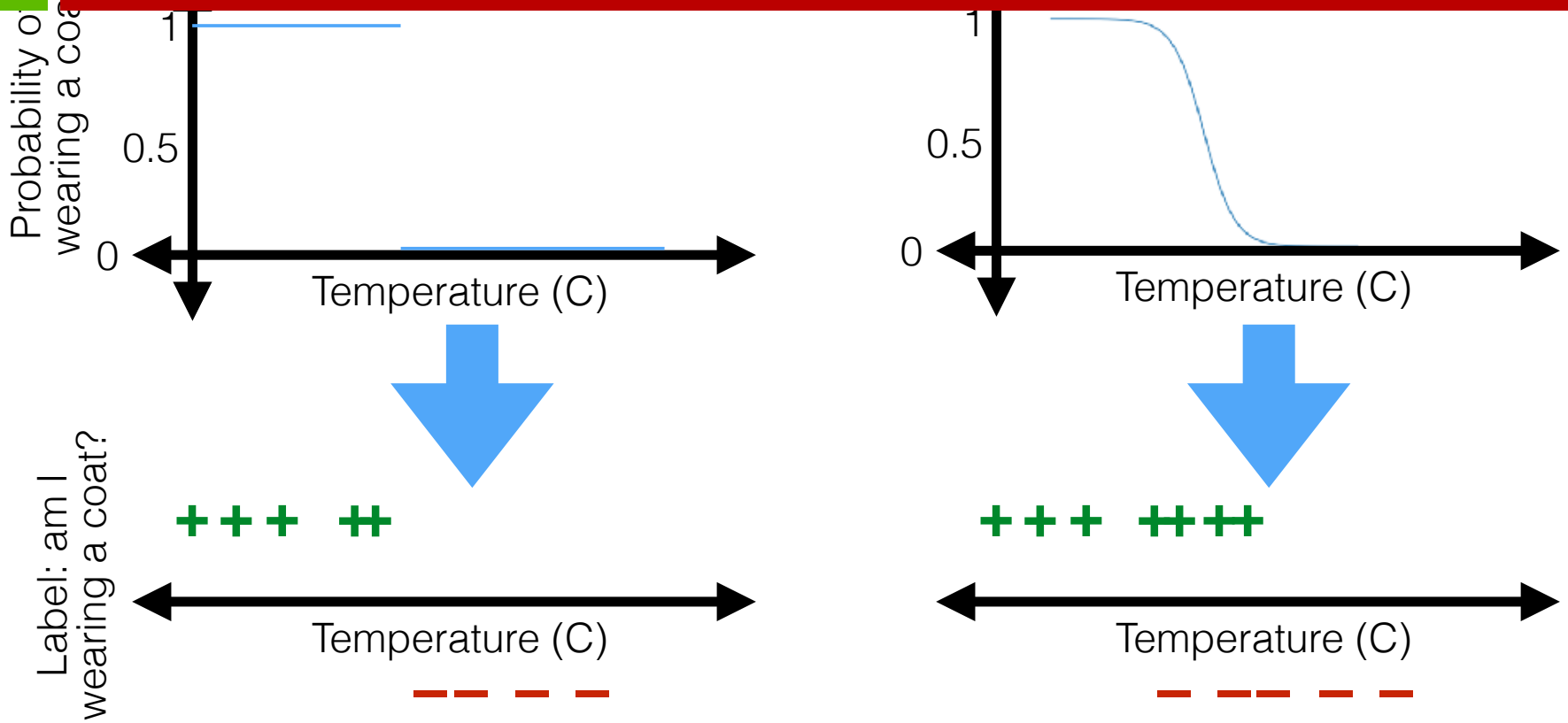
- How to make this shape?

Capturing uncertainty



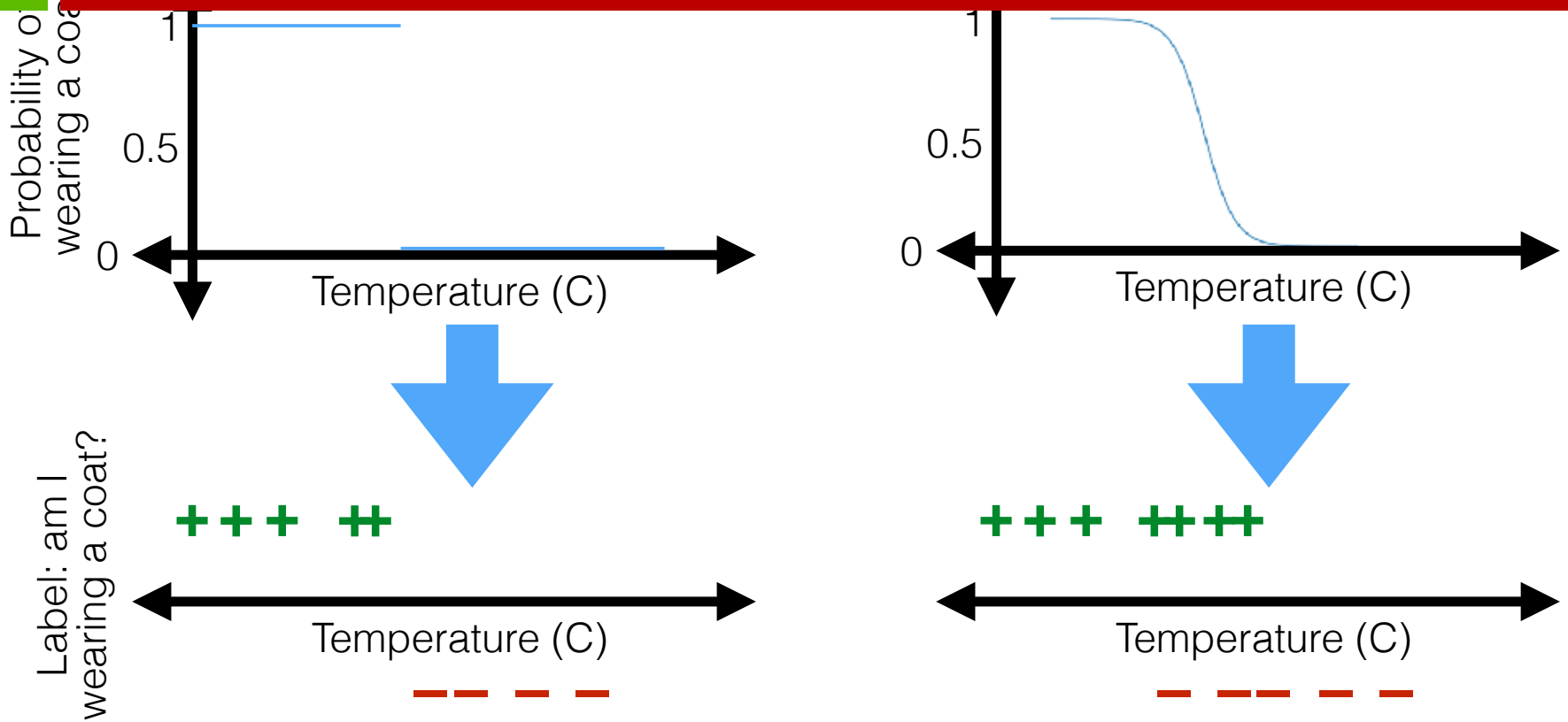
- How to make this shape?

Capturing uncertainty



- How to make this shape?
  - Sigmoid/logistic function

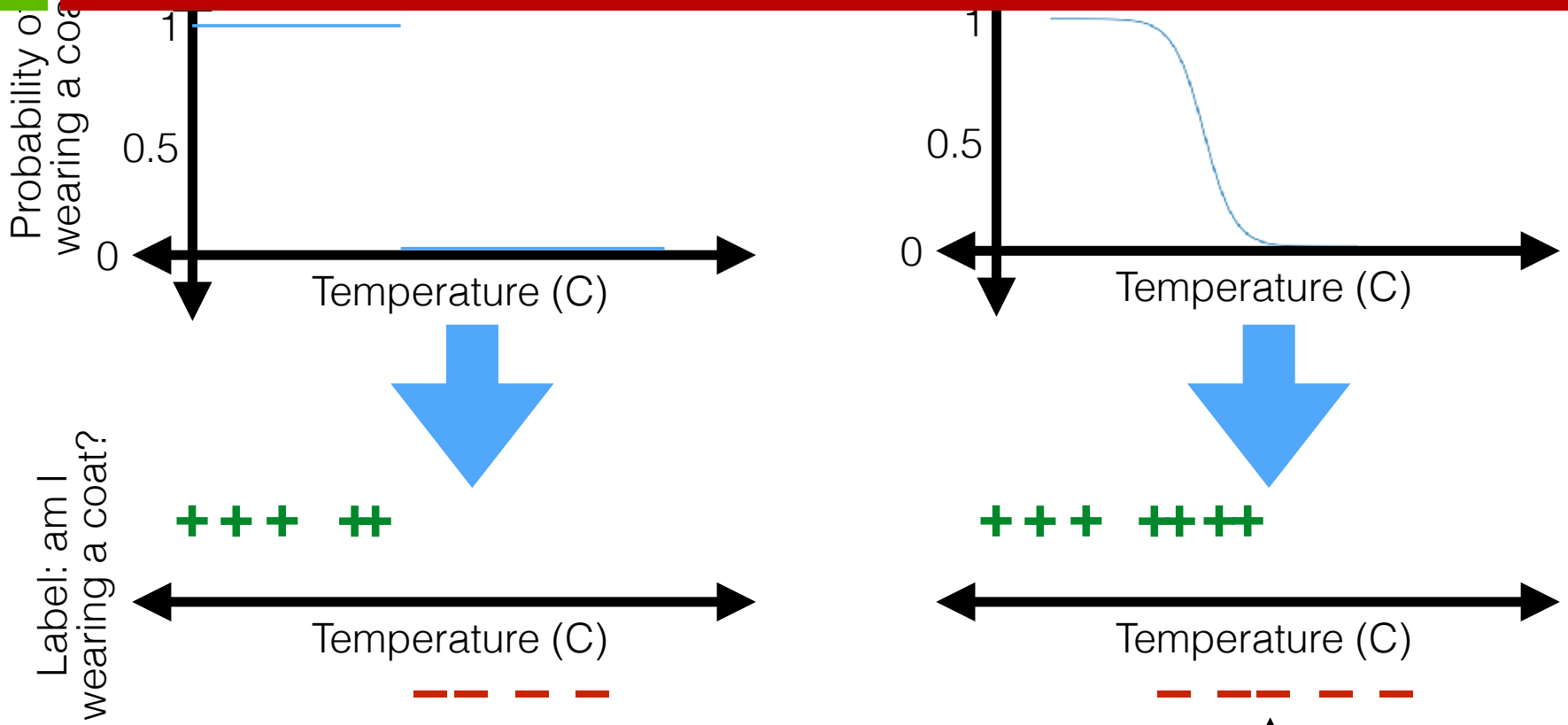
Capturing uncertainty



- How to make this shape?
  - Sigmoid/logistic function

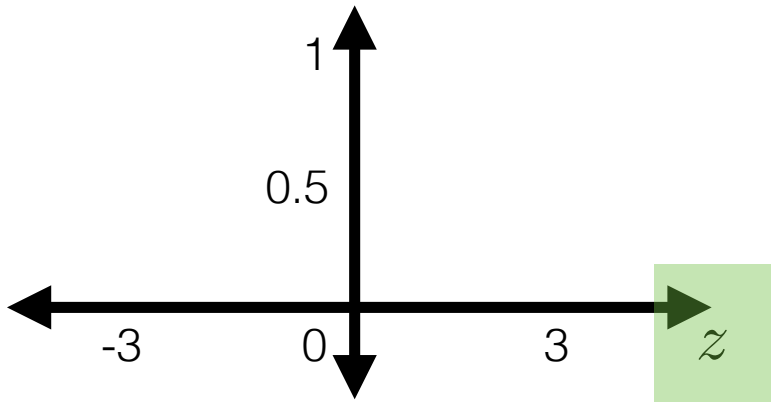
$$\sigma(z) = \frac{1}{1 + \exp(-z)}$$

Capturing uncertainty

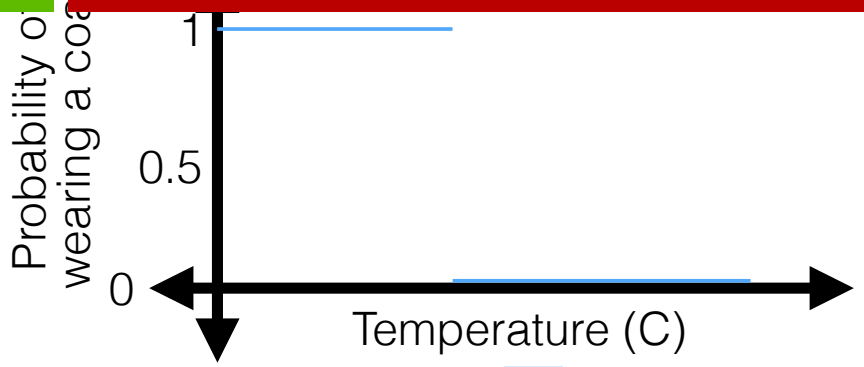


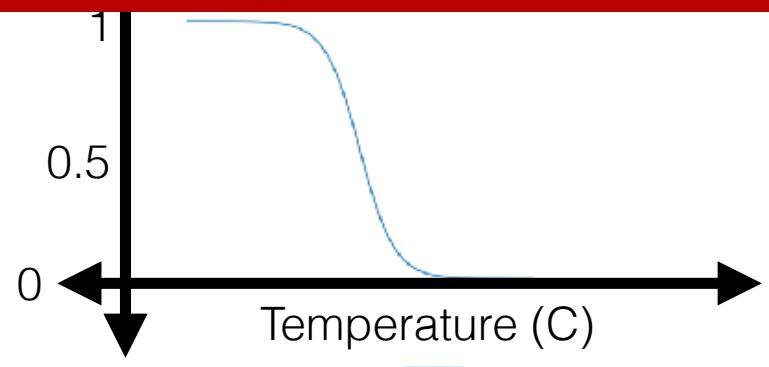
- How to make this shape?
  - Sigmoid/logistic function

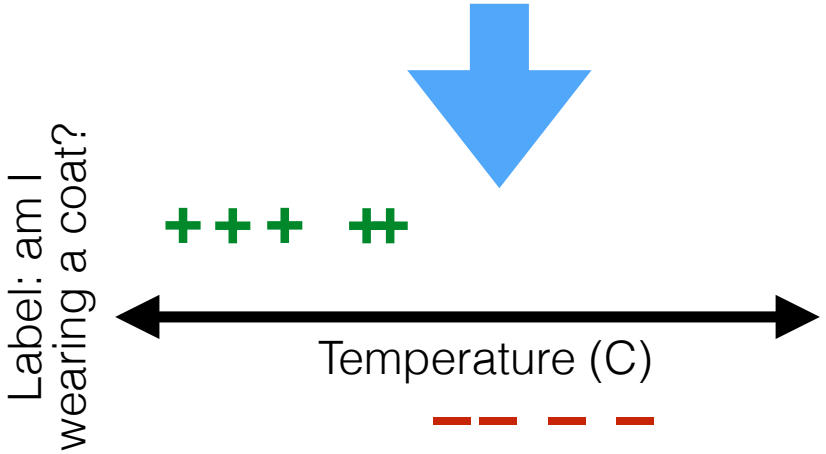
$$\sigma(z) = \frac{1}{1 + \exp(-z)}$$

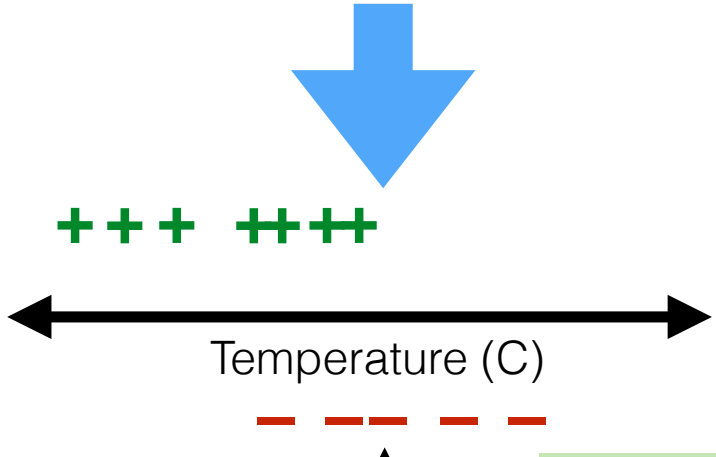


Capturing uncertainty



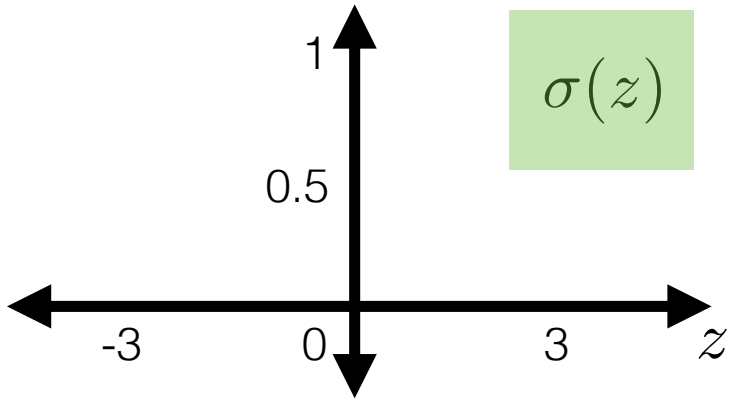






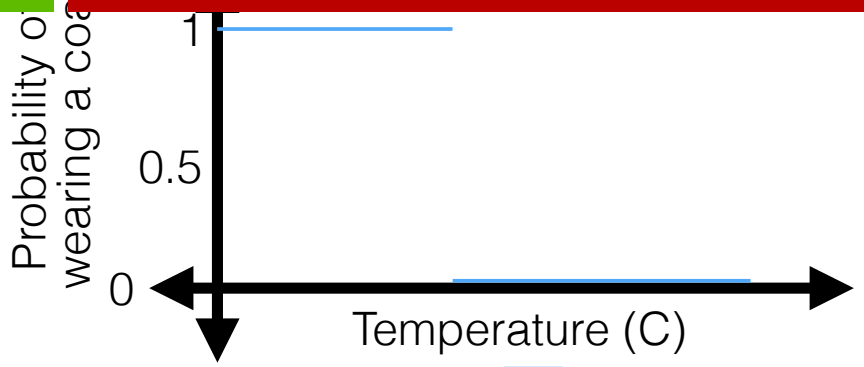
- How to make this shape?
  - Sigmoid/logistic function

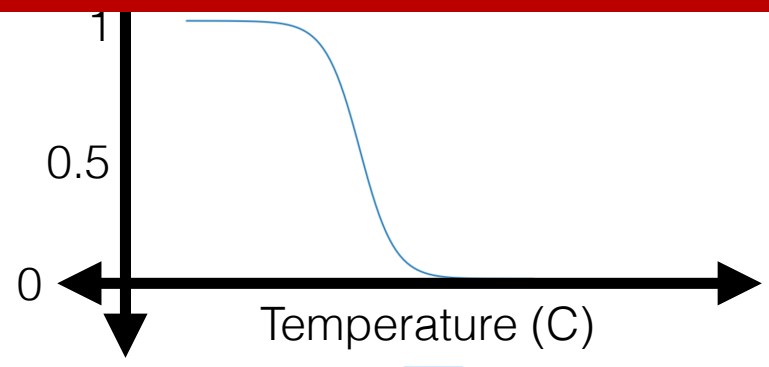
$$\sigma(z) = \frac{1}{1 + \exp(-z)}$$

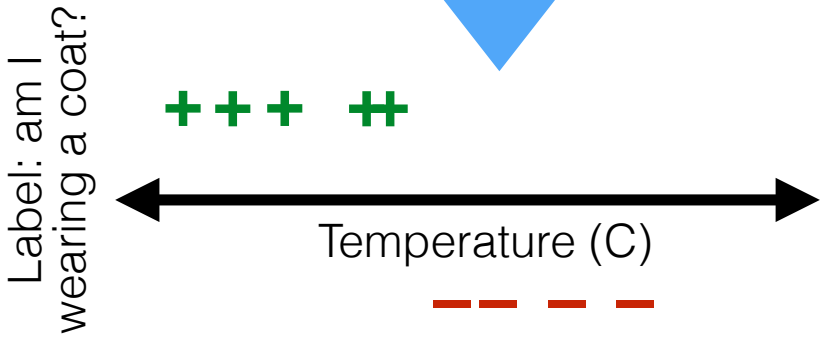


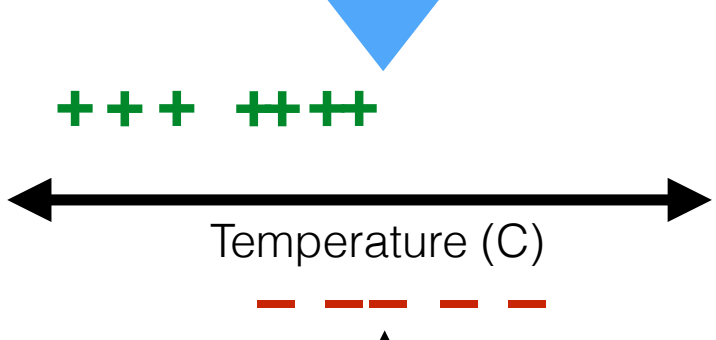
$\sigma(z)$

Capturing uncertainty



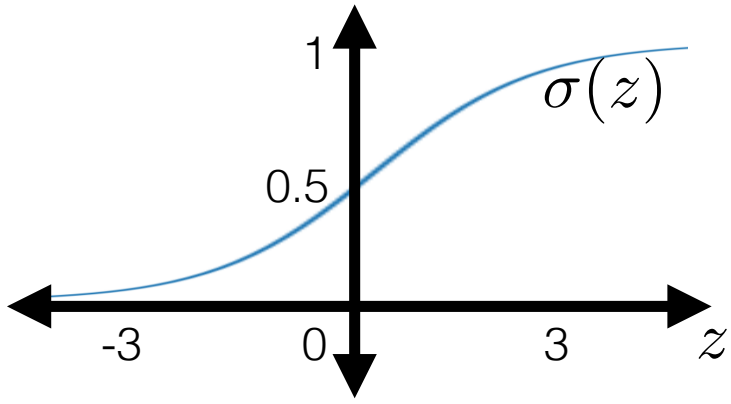




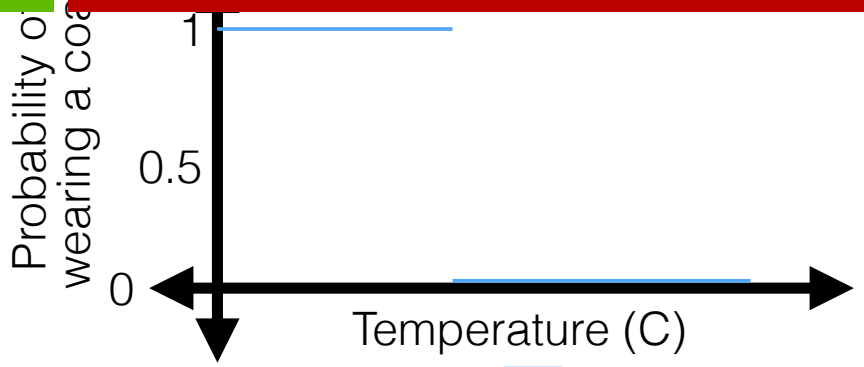


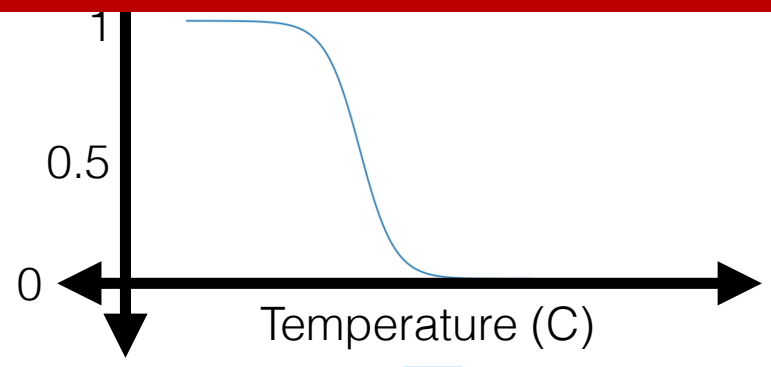
- How to make this shape?
  - Sigmoid/logistic function

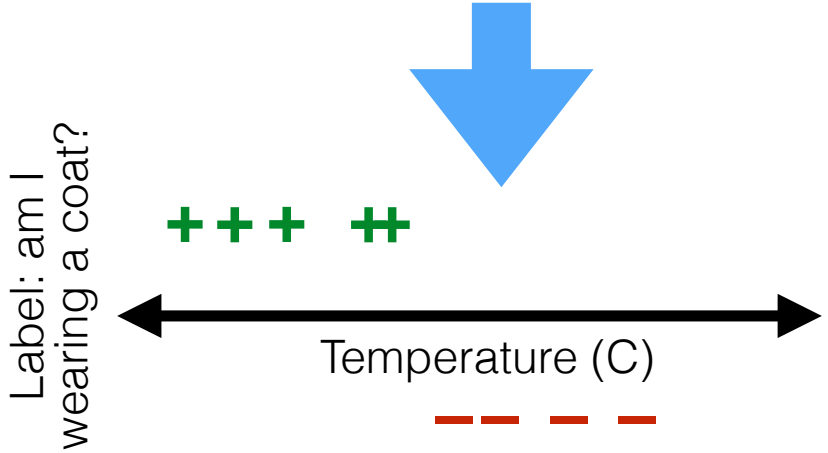
$$\sigma(z) = \frac{1}{1 + \exp(-z)}$$

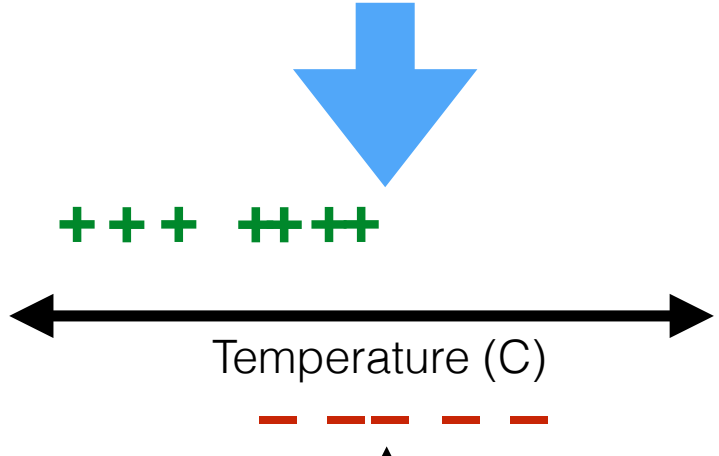


Capturing uncertainty



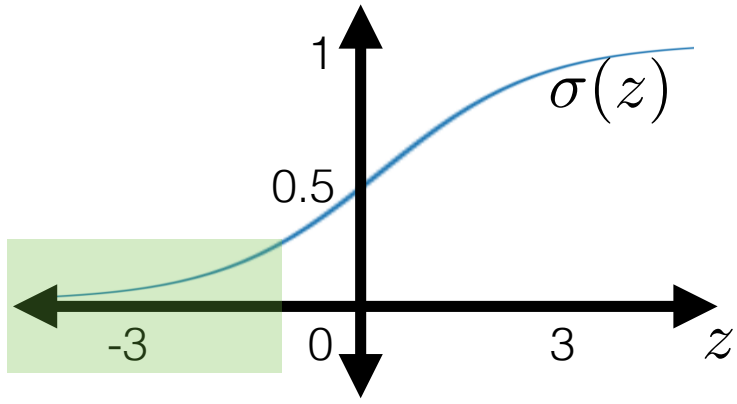




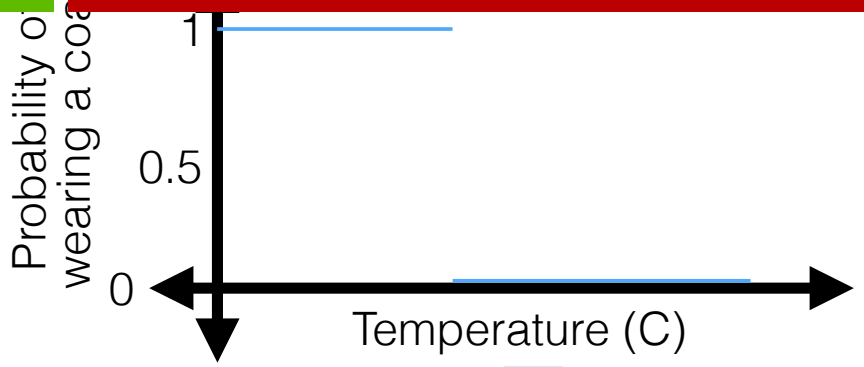


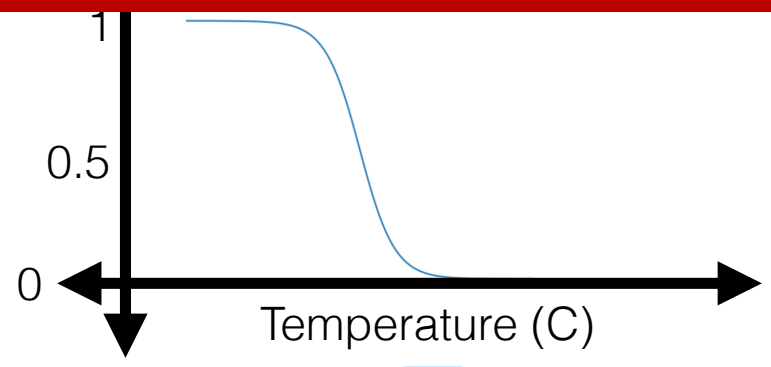
- How to make this shape?
  - Sigmoid/logistic function

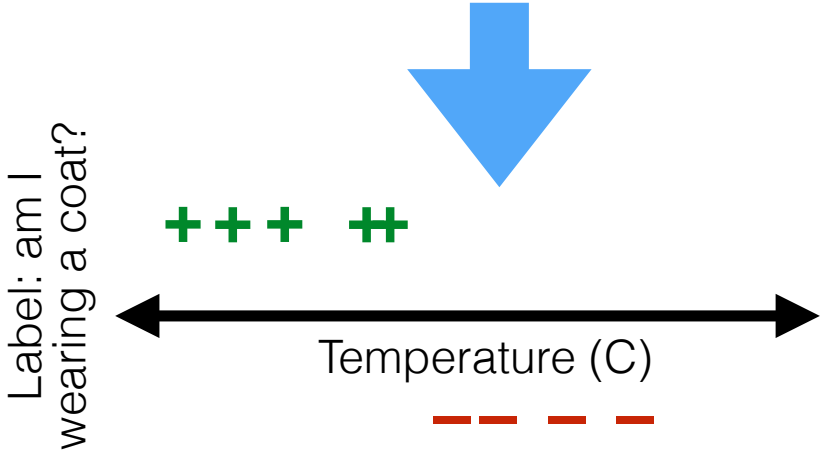
$$\sigma(z) = \frac{1}{1 + \exp(-z)}$$

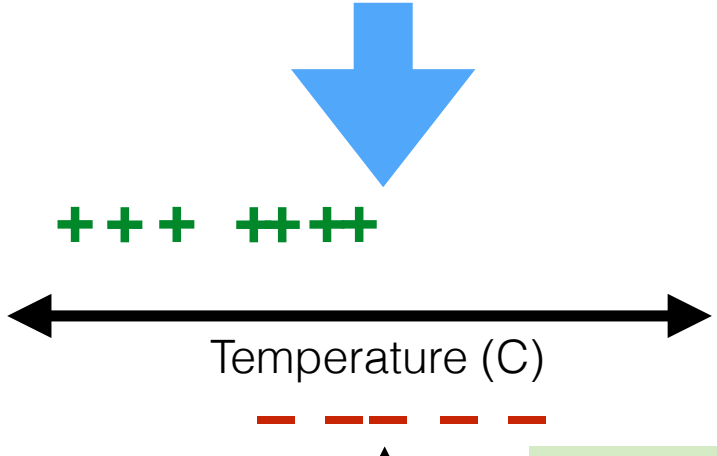


Capturing uncertainty



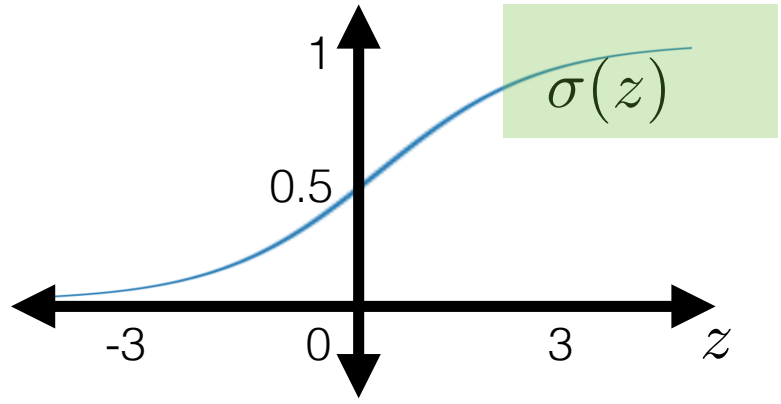




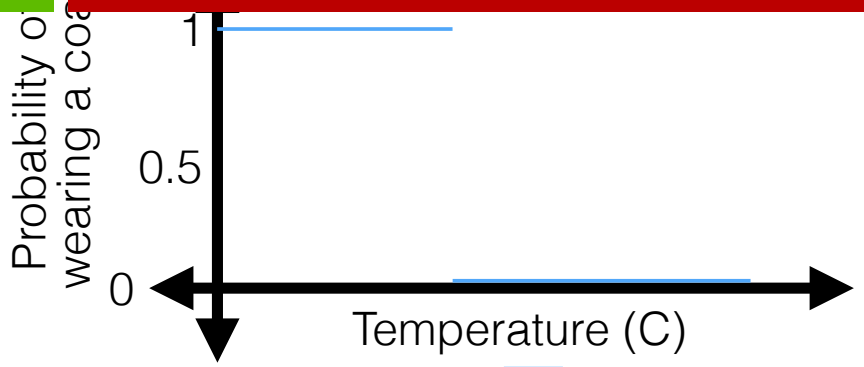


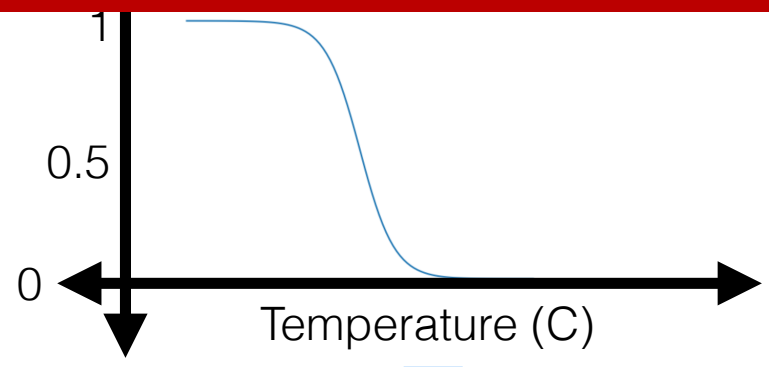
- How to make this shape?
  - Sigmoid/logistic function

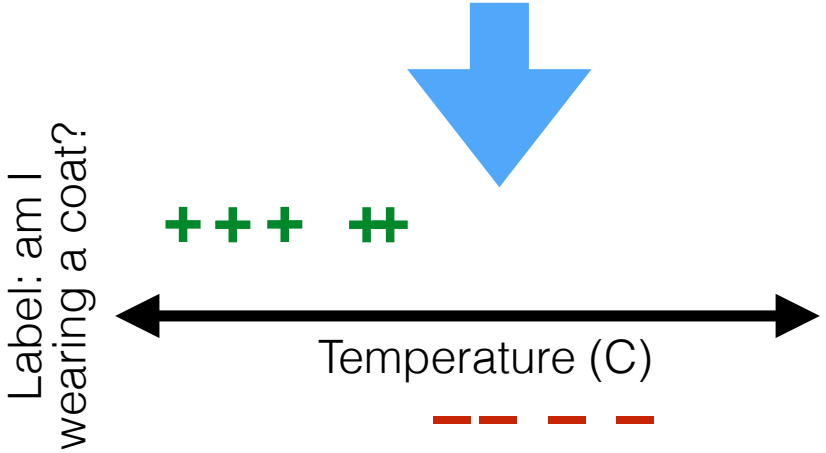
$$\sigma(z) = \frac{1}{1 + \exp(-z)}$$

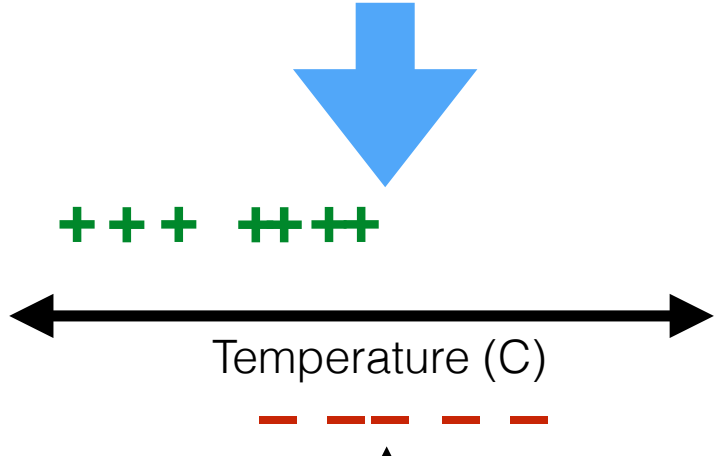


Capturing uncertainty



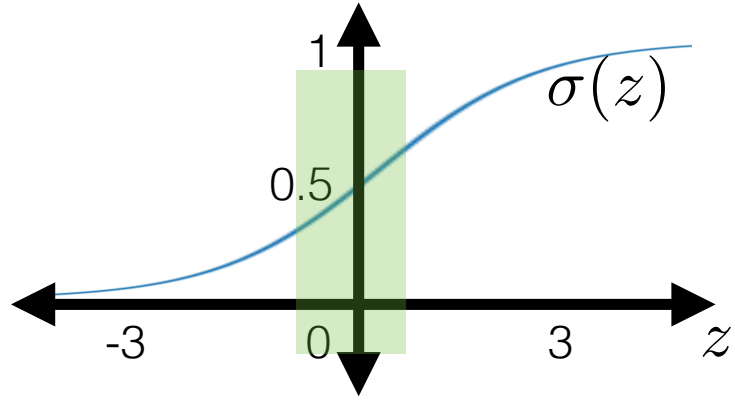




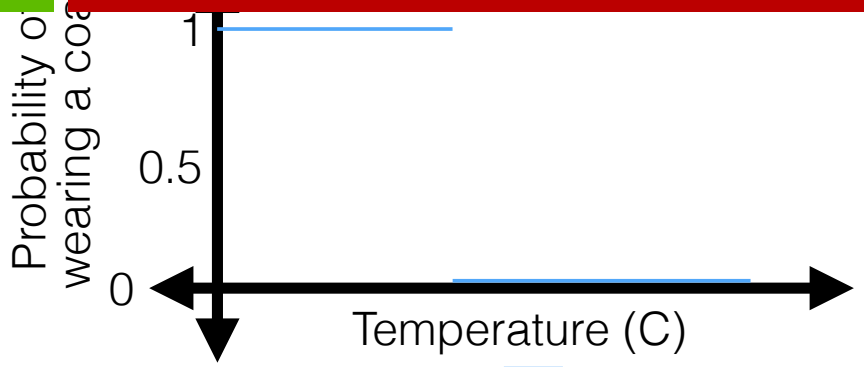


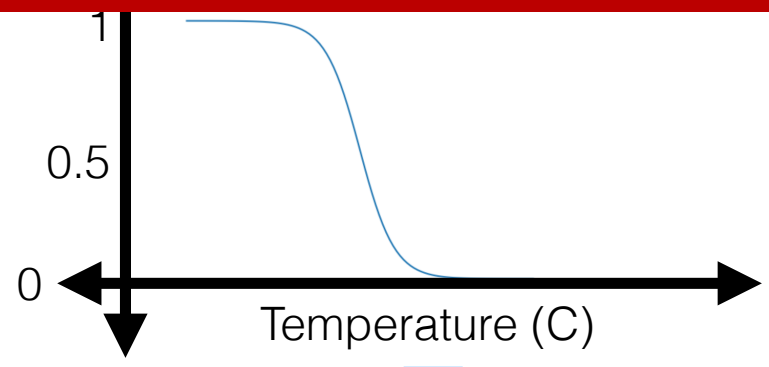
- How to make this shape?
  - Sigmoid/logistic function

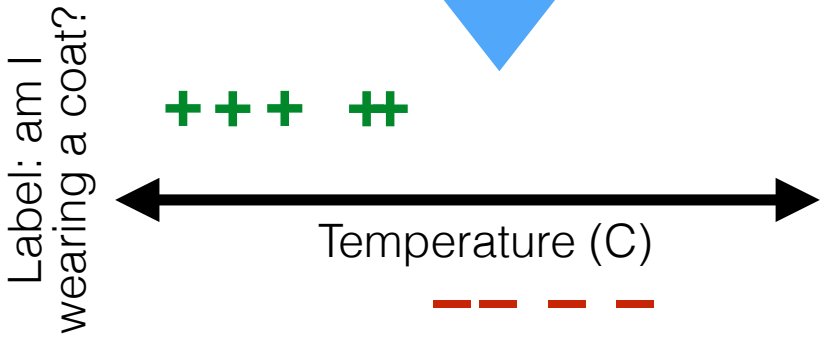
$$\sigma(z) = \frac{1}{1 + \exp(-z)}$$

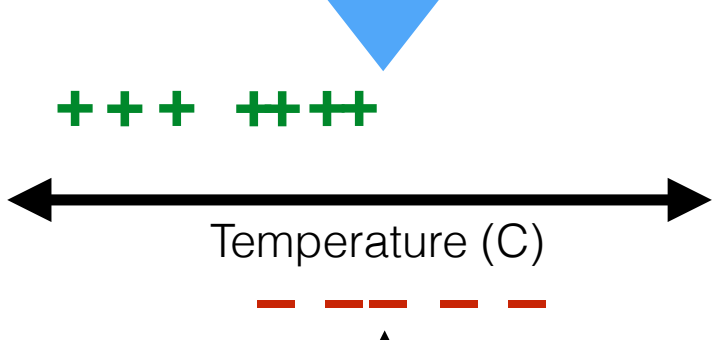


Capturing uncertainty



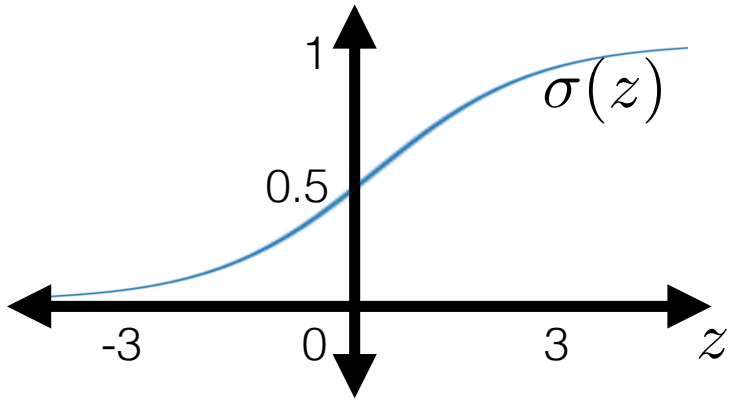




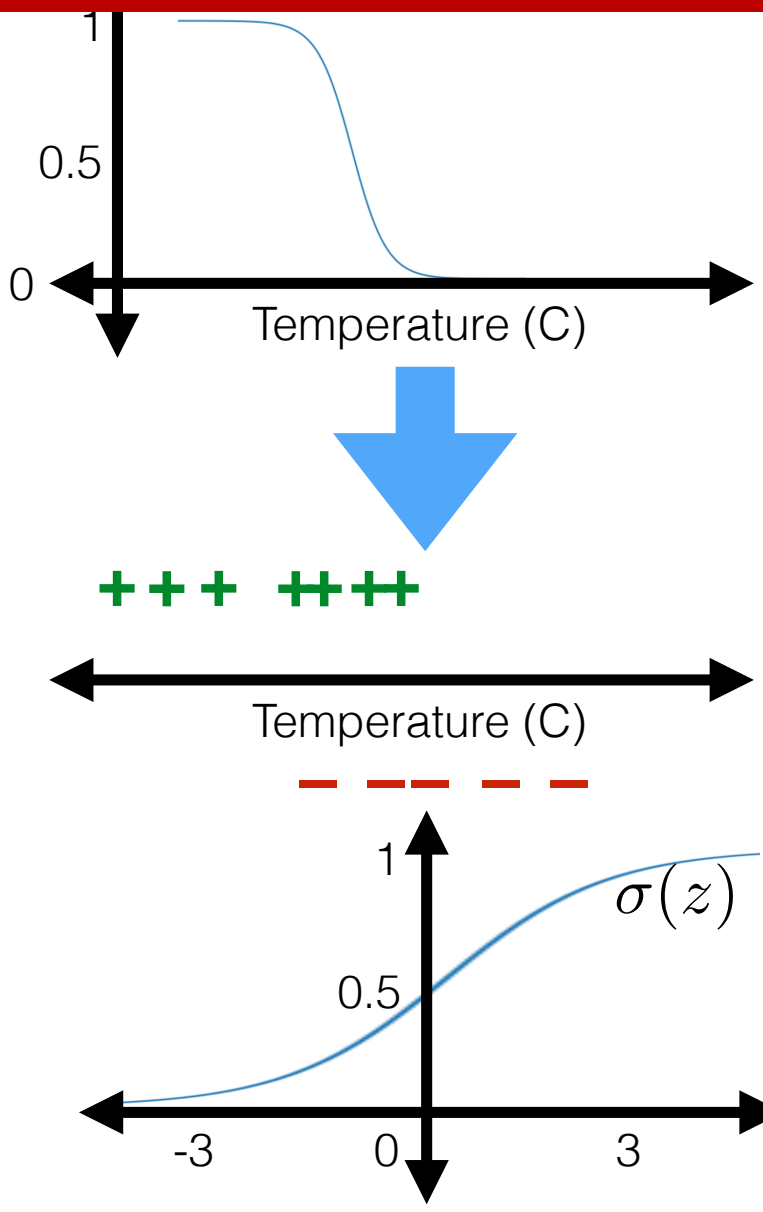


- How to make this shape?
  - Sigmoid/logistic function

$$\sigma(z) = \frac{1}{1 + \exp(-z)}$$



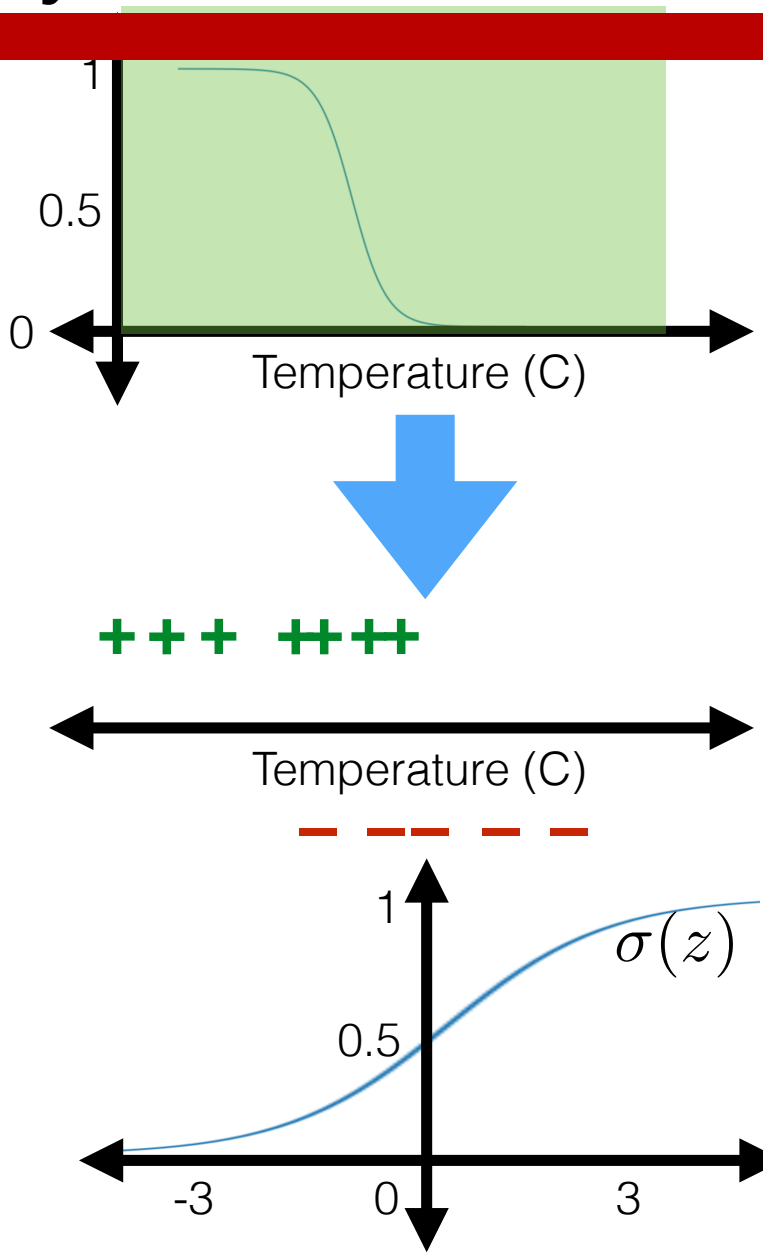
Capturing uncertainty



- How to make this shape?
  - Sigmoid/logistic function

$$\sigma(z) = \frac{1}{1 + \exp(-z)}$$

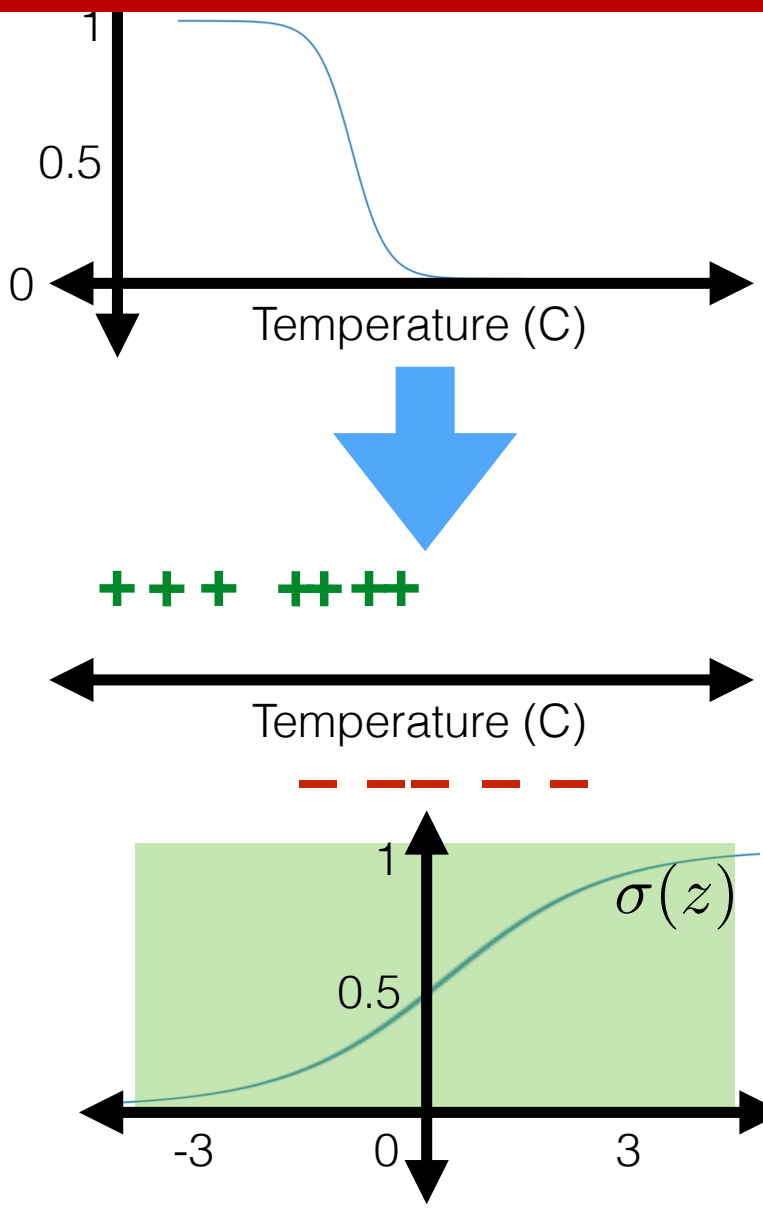
Capturing uncertainty



- How to make this shape?
  - Sigmoid/logistic function

$$\sigma(z) = \frac{1}{1 + \exp(-z)}$$

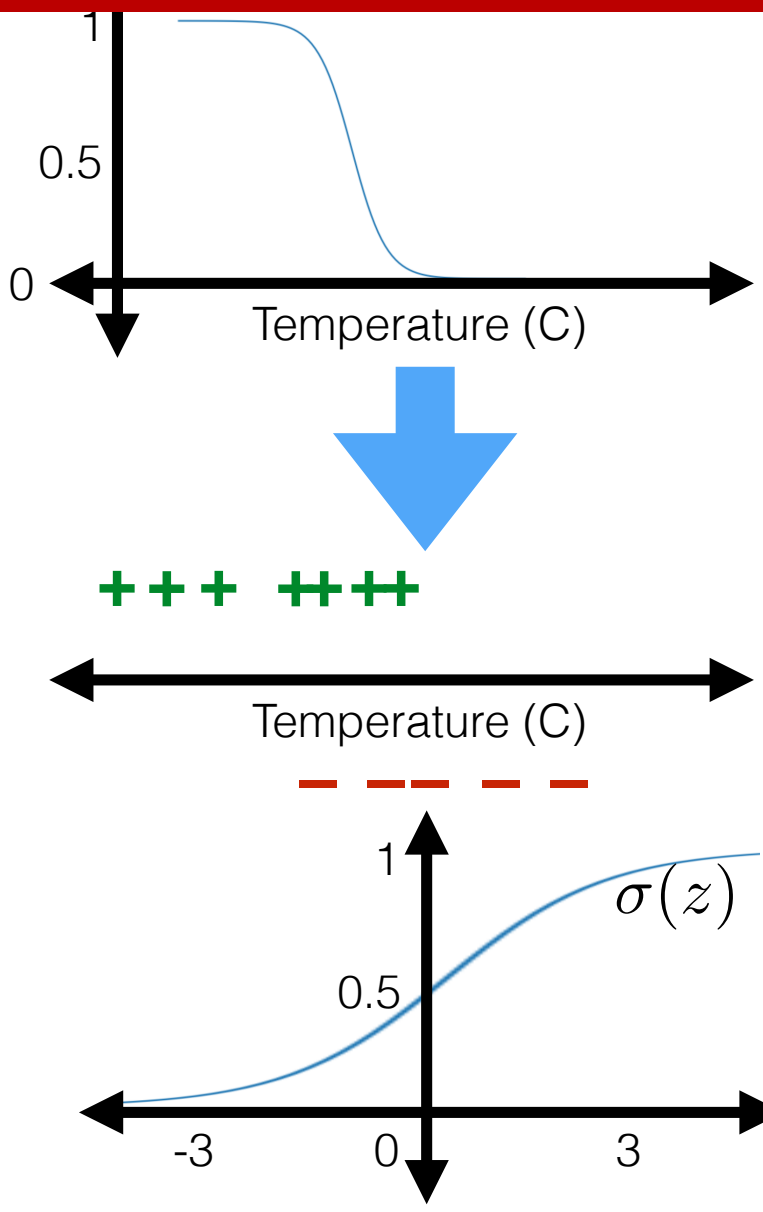
Capturing uncertainty



- How to make this shape?
  - Sigmoid/logistic function

$$\sigma(z) = \frac{1}{1 + \exp(-z)}$$

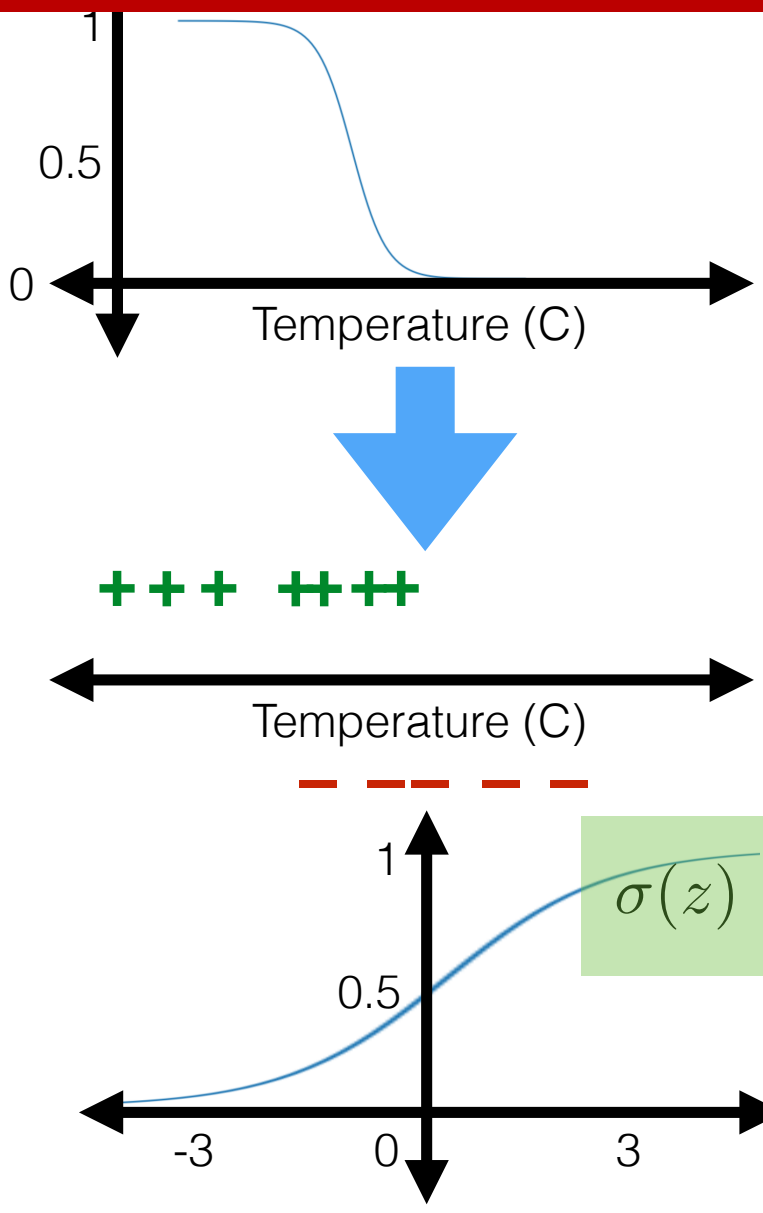
Capturing uncertainty



- How to make this shape?
  - Sigmoid/logistic function

$$\sigma(z) = \frac{1}{1 + \exp(-z)}$$

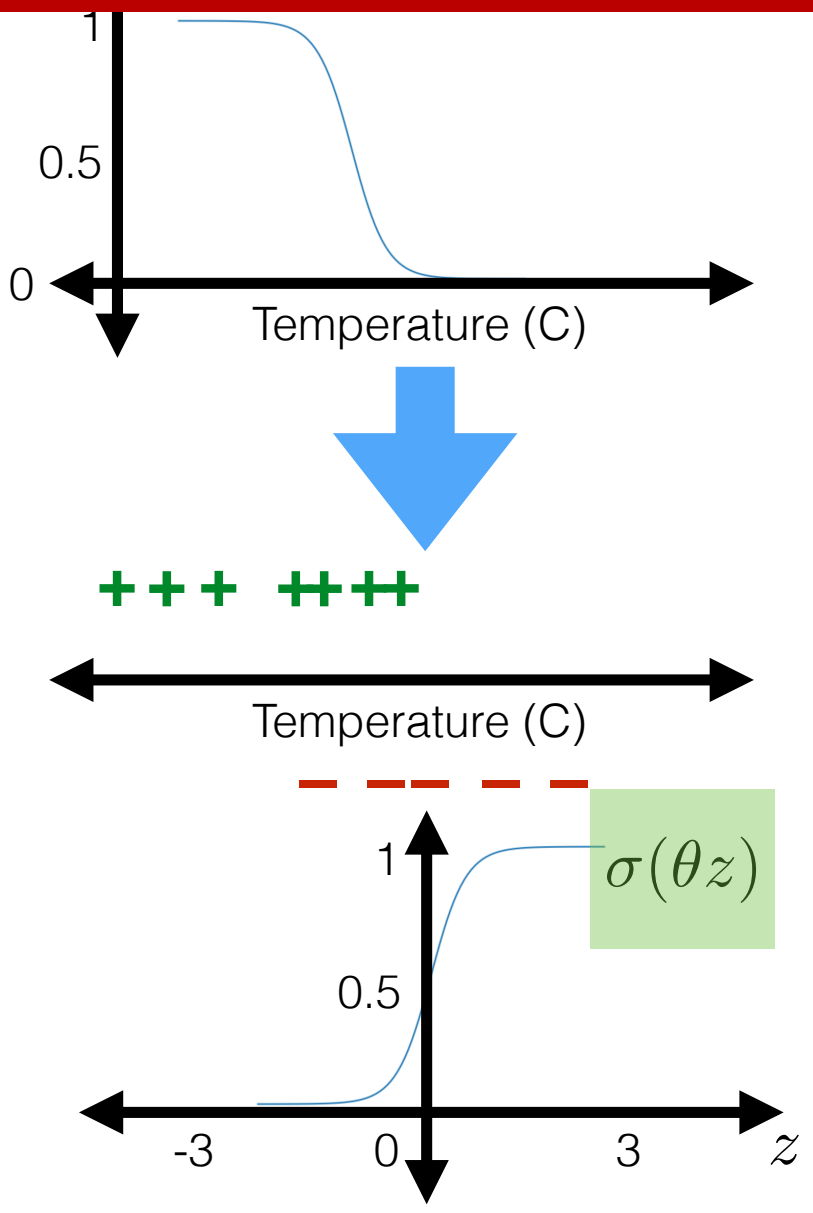
Capturing uncertainty



- How to make this shape?
  - Sigmoid/logistic function

$$\sigma(z) = \frac{1}{1 + \exp(-z)}$$

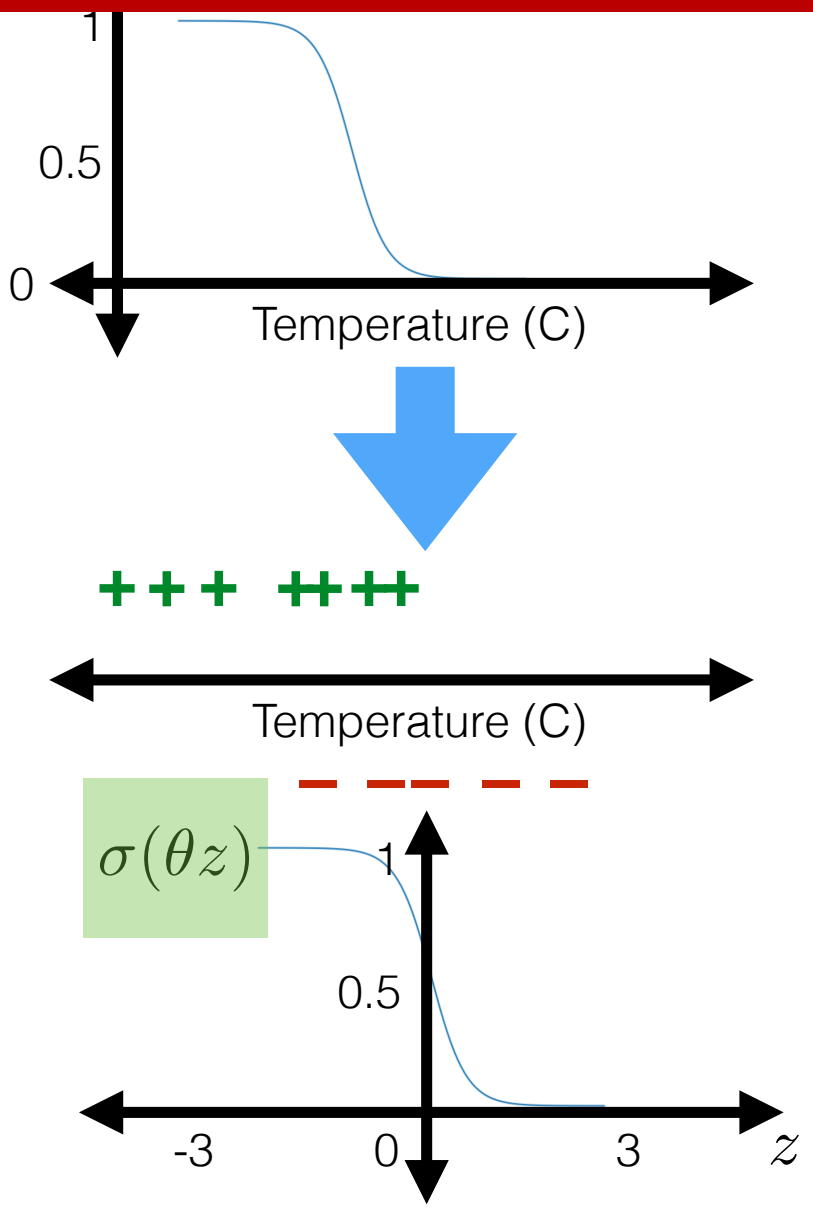
Capturing uncertainty



- How to make this shape?
  - Sigmoid/logistic function

$$\sigma(z) = \frac{1}{1 + \exp(-z)}$$

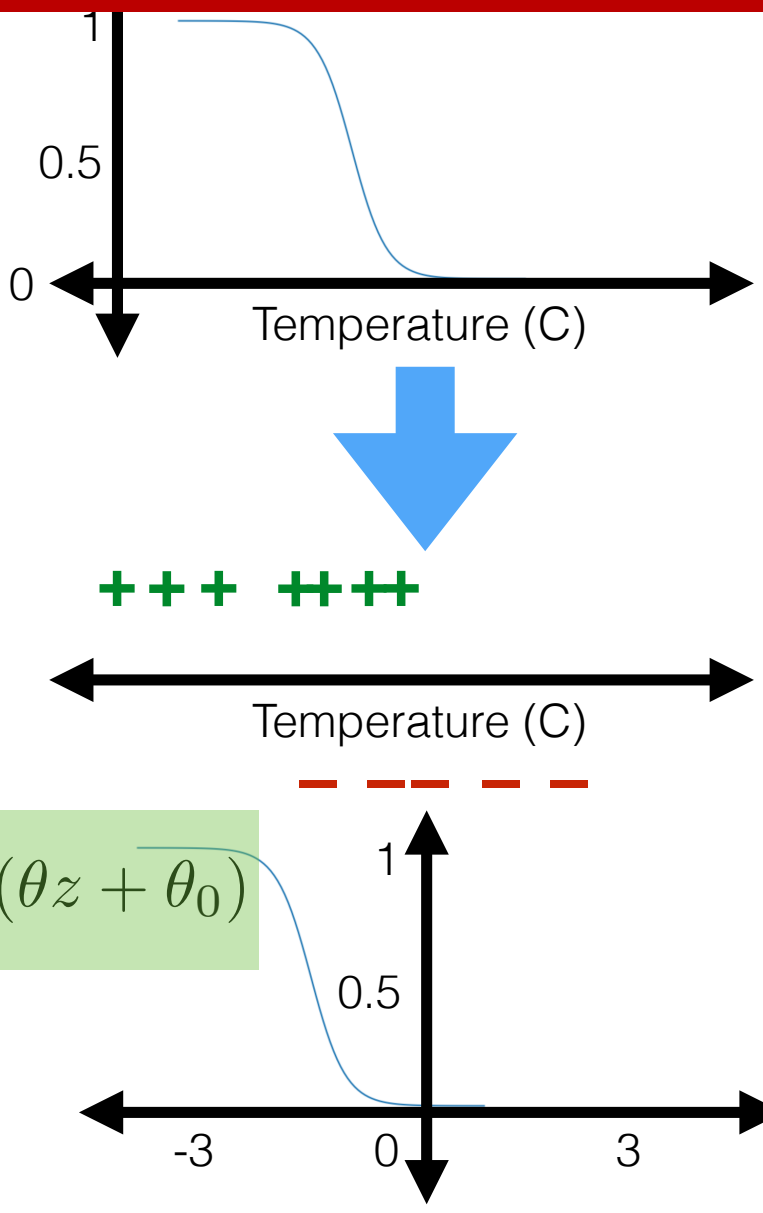
Capturing uncertainty



- How to make this shape?
  - Sigmoid/logistic function

$$\sigma(z) = \frac{1}{1 + \exp(-z)}$$

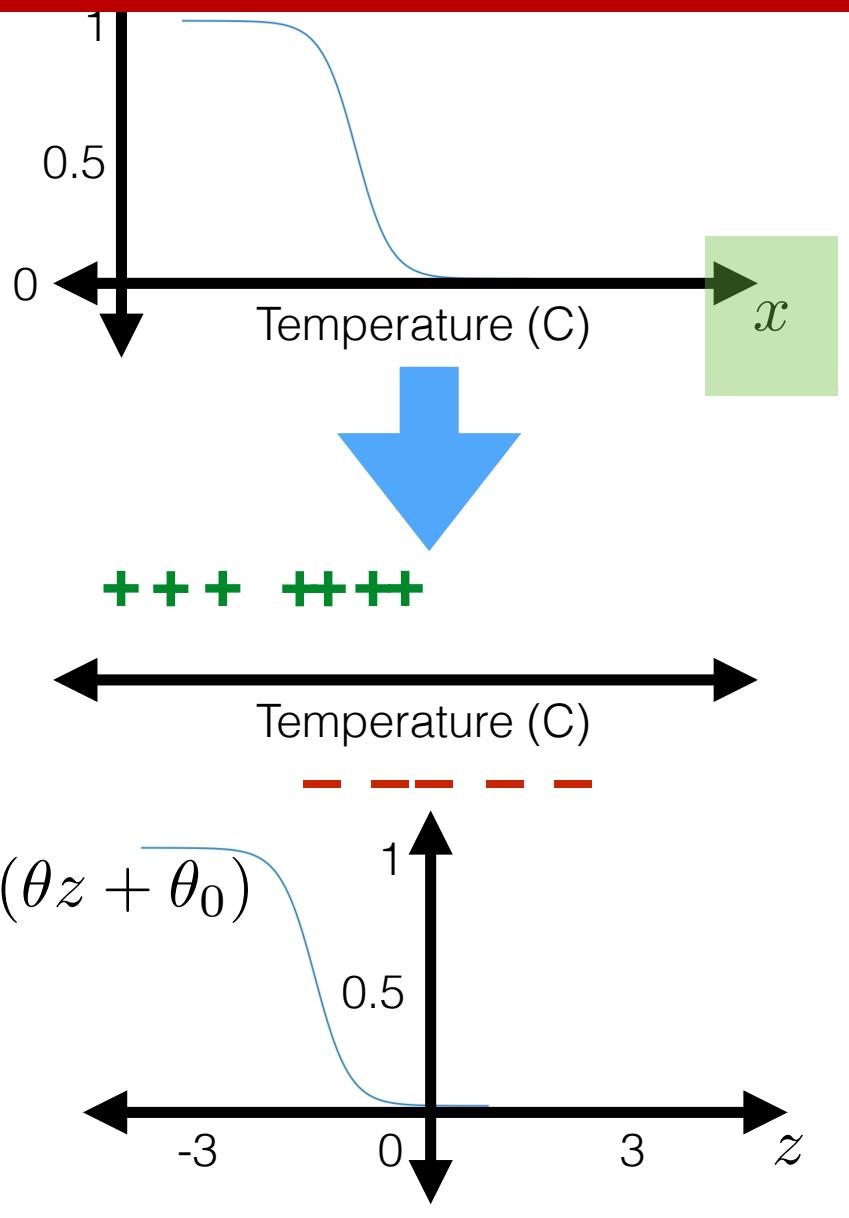
Capturing uncertainty



- How to make this shape?
  - Sigmoid/logistic function

$$\sigma(z) = \frac{1}{1 + \exp(-z)}$$

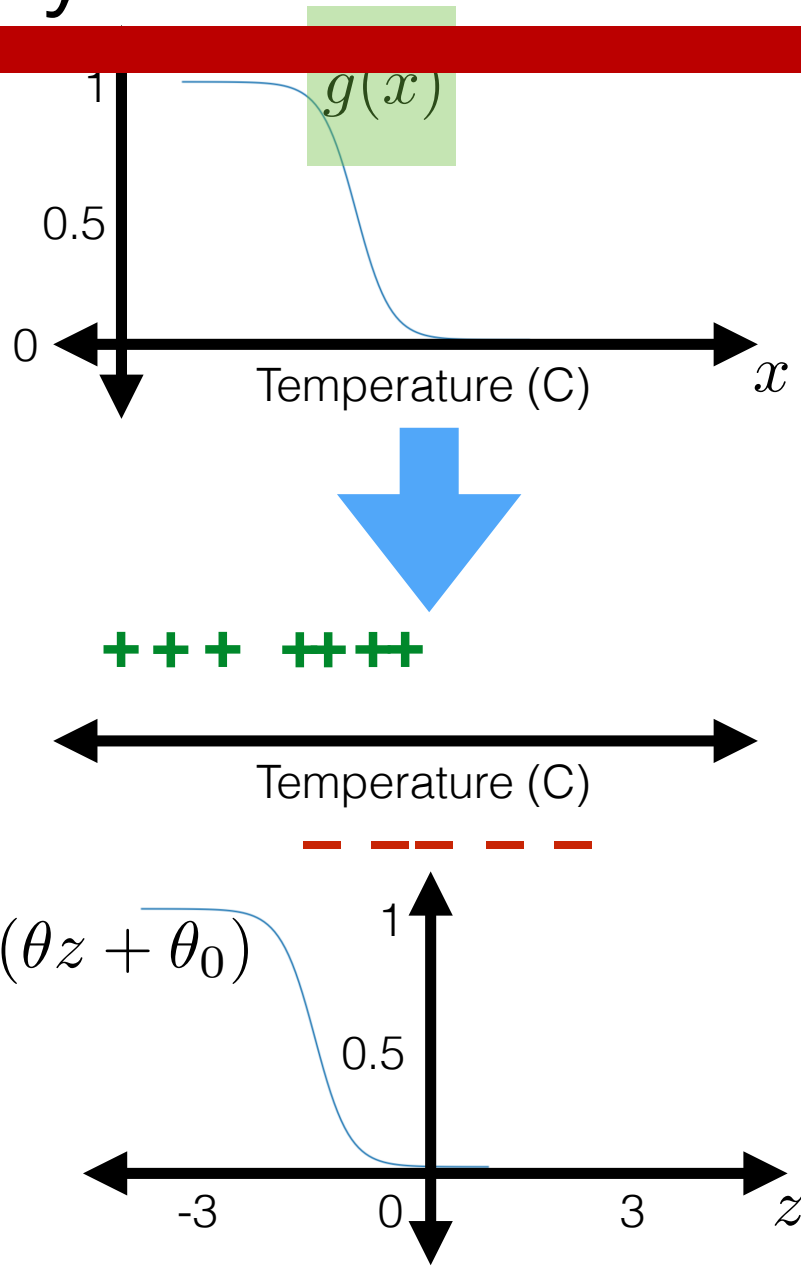
Capturing uncertainty



- How to make this shape?
  - Sigmoid/logistic function

$$\sigma(z) = \frac{1}{1 + \exp(-z)}$$

Capturing uncertainty

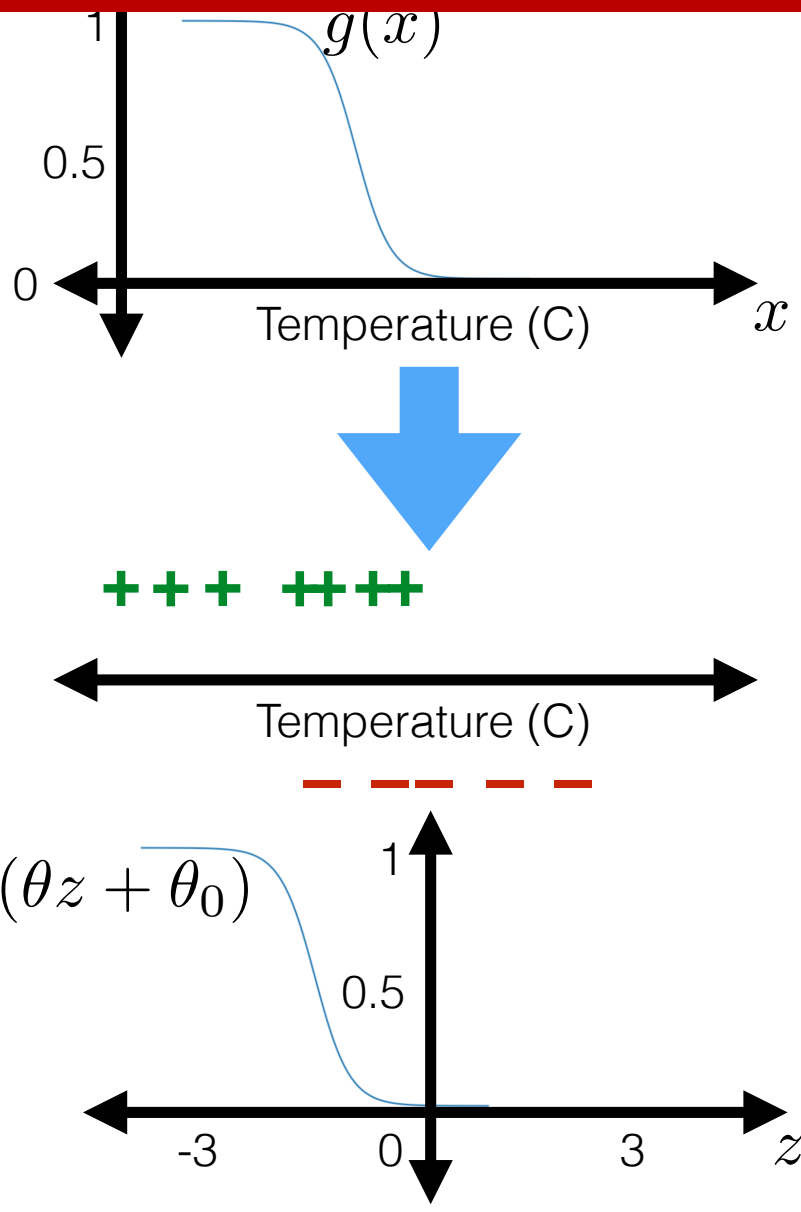


- How to make this shape?
  - Sigmoid/logistic function

$$\sigma(z) = \frac{1}{1 + \exp(-z)}$$

Capturing uncertainty

$$g(x) = \sigma(\theta x + \theta_0)$$

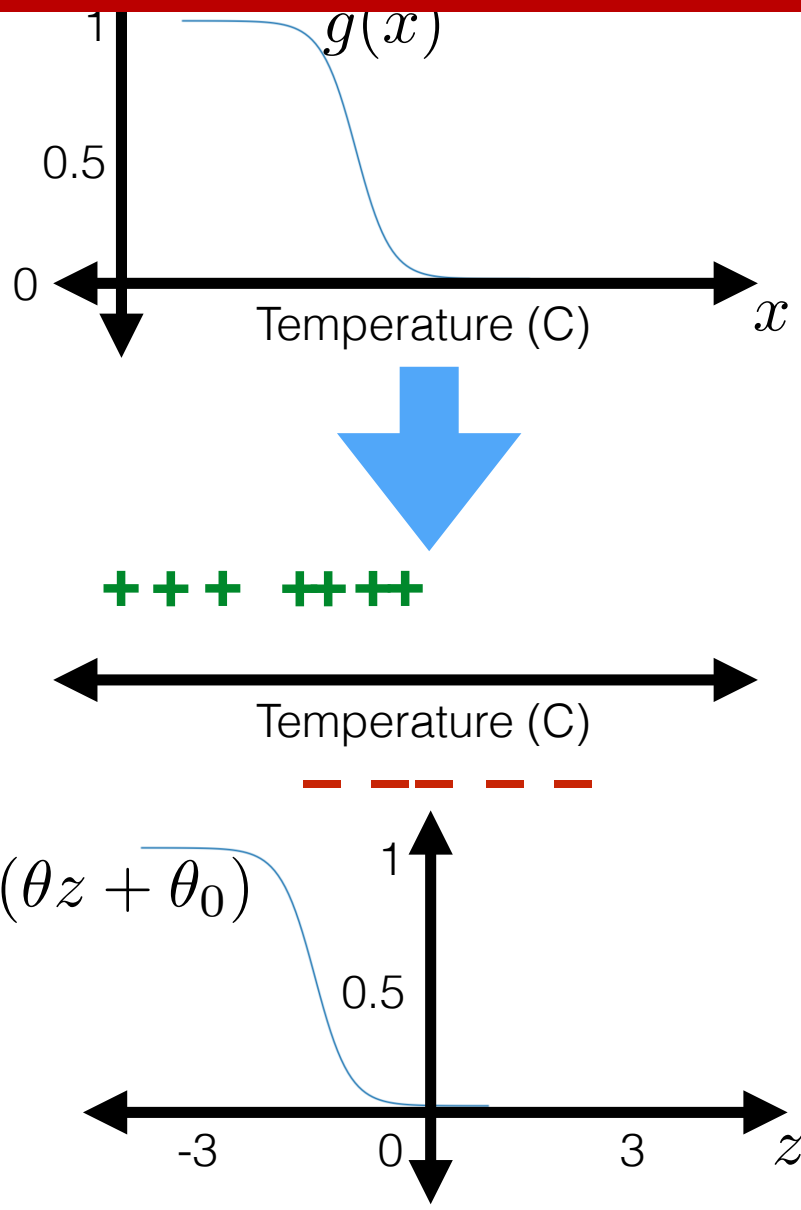


- How to make this shape?
  - Sigmoid/logistic function

$$\sigma(z) = \frac{1}{1 + \exp(-z)}$$

Capturing uncertainty

$$g(x) = \sigma(\theta x + \theta_0)$$
$$= \frac{1}{1 + \exp\{-(\theta x + \theta_0)\}}$$



- How to make this shape?
  - Sigmoid/logistic function

$$\sigma(z) = \frac{1}{1 + \exp(-z)}$$

Capturing uncertainty

Capturing uncertainty

Capturing uncertainty

1 feature:

$$g(x) = \sigma(\theta x + \theta_0)$$

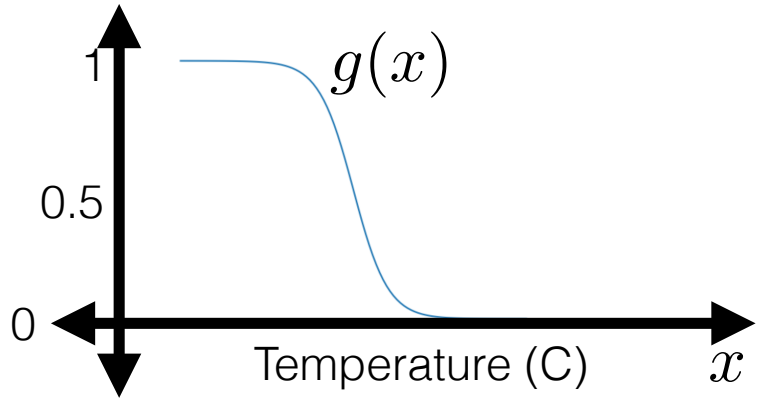
$$= \frac{1}{1 + \exp\{-(\theta x + \theta_0)\}}$$

Capturing uncertainty

Feature:

$$g(x) = \sigma(\theta x + \theta_0)$$

$$= \frac{1}{1 + \exp\{-(\theta x + \theta_0)\}}$$

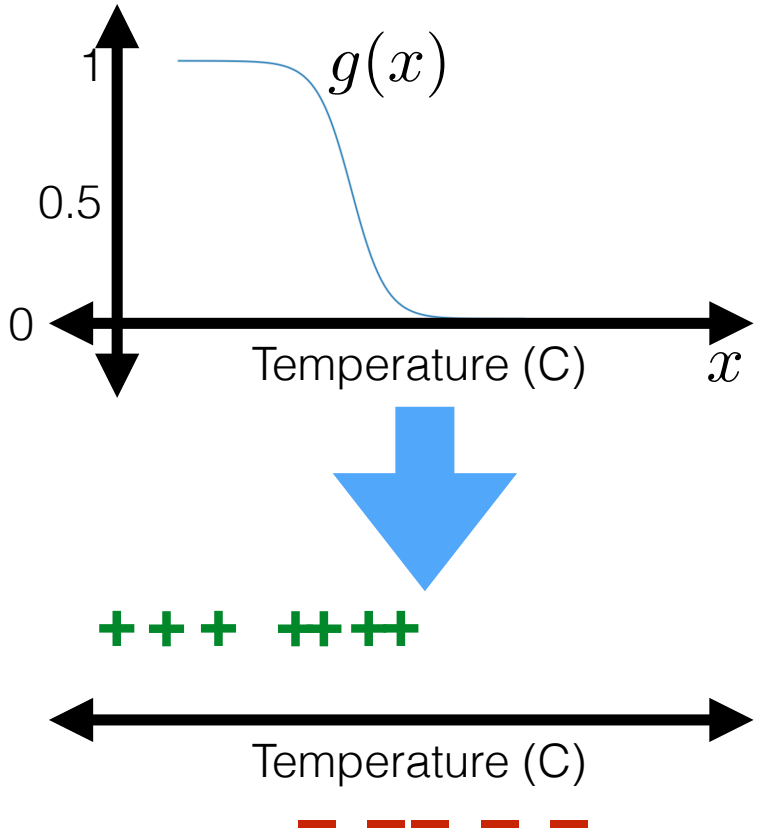


Capturing uncertainty

Feature:

$$g(x) = \sigma(\theta x + \theta_0)$$

$$= \frac{1}{1 + \exp\{-(\theta x + \theta_0)\}}$$



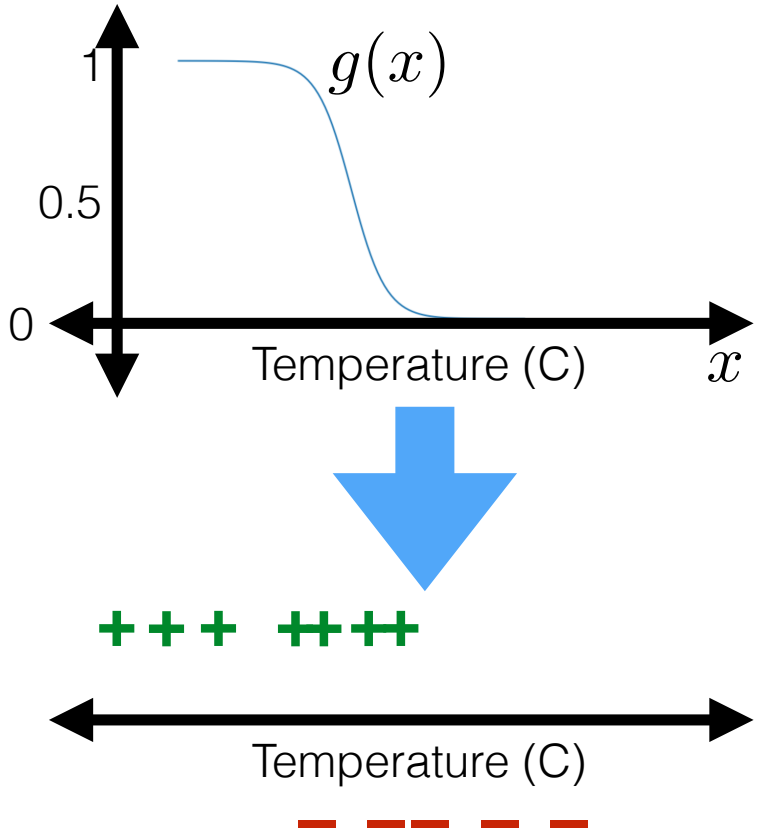
Capturing uncertainty

2 features:

1 feature:

$$g(x) = \sigma(\theta x + \theta_0)$$

$$= \frac{1}{1 + \exp\{-(\theta x + \theta_0)\}}$$



Capturing uncertainty

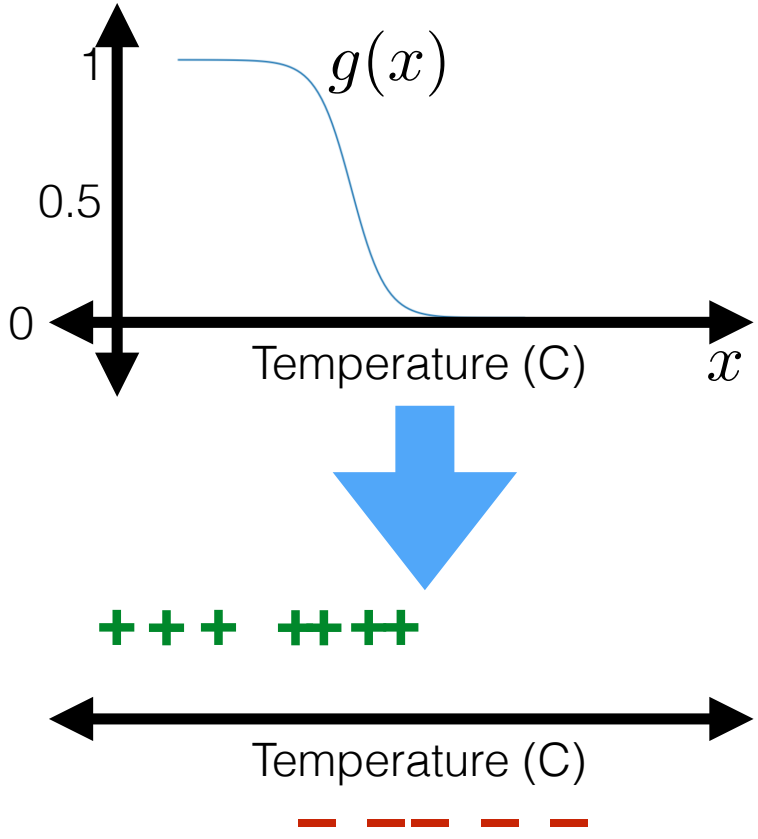
1 feature:

$$g(x) = \sigma(\theta x + \theta_0)$$

$$= \frac{1}{1 + \exp\{-(\theta x + \theta_0)\}}$$

2 features:

$$= \frac{1}{1 + \exp\{-(\theta^\top x + \theta_0)\}}$$

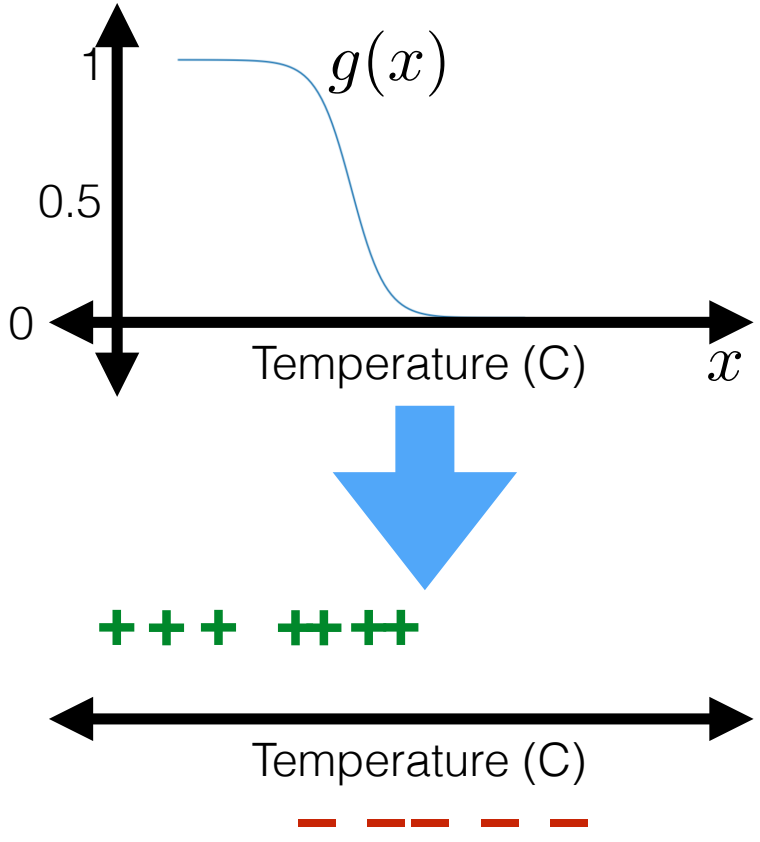


Capturing uncertainty

1 feature:

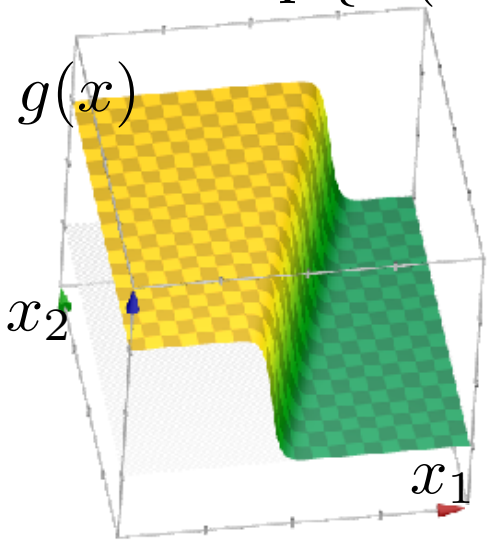
$$g(x) = \sigma(\theta x + \theta_0)$$

$$= \frac{1}{1 + \exp\{-(\theta x + \theta_0)\}}$$



2 features:

$$= \frac{1}{1 + \exp\{-(\theta^\top x + \theta_0)\}}$$

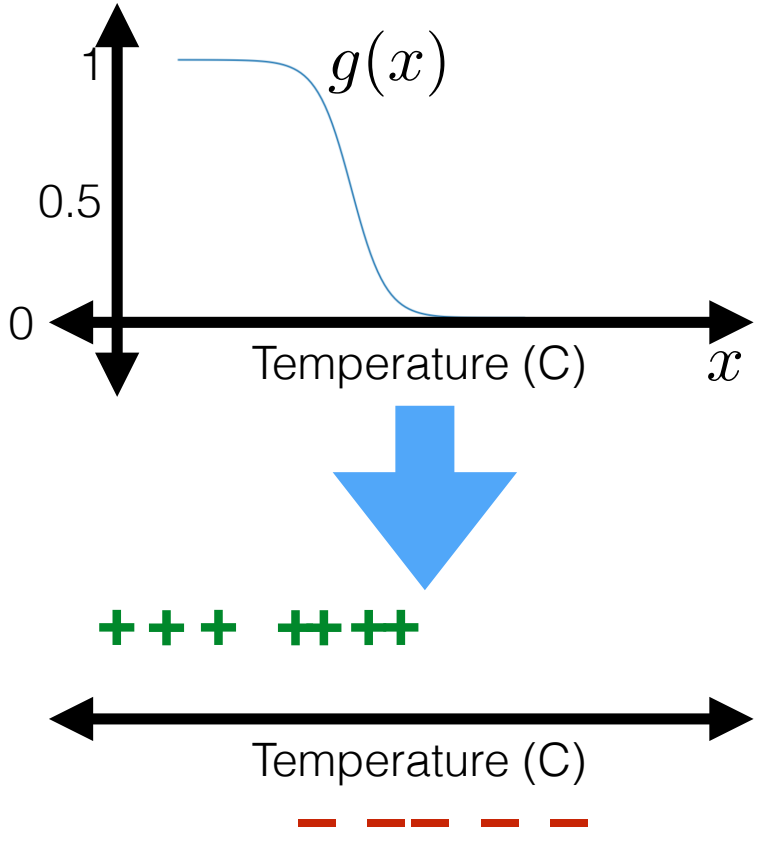


Capturing uncertainty

1 feature:

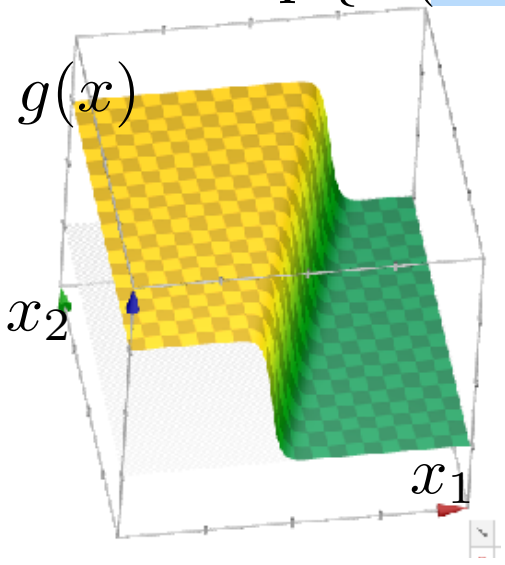
$$g(x) = \sigma(\theta x + \theta_0)$$

$$= \frac{1}{1 + \exp\{-(\theta x + \theta_0)\}}$$



2 features:

$$= \frac{1}{1 + \exp\{-(\theta^\top x + \theta_0)\}}$$

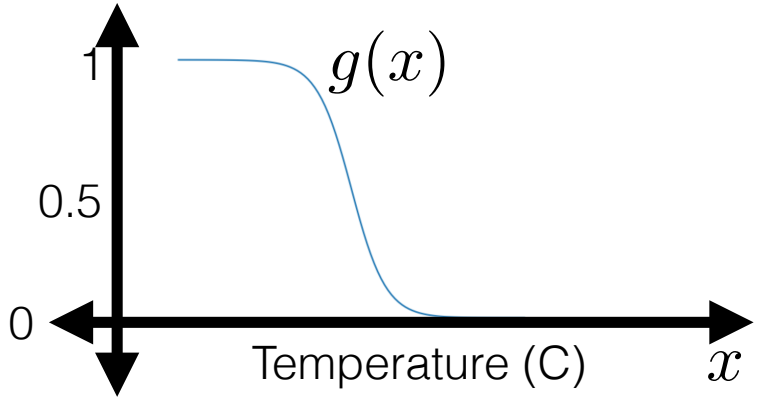


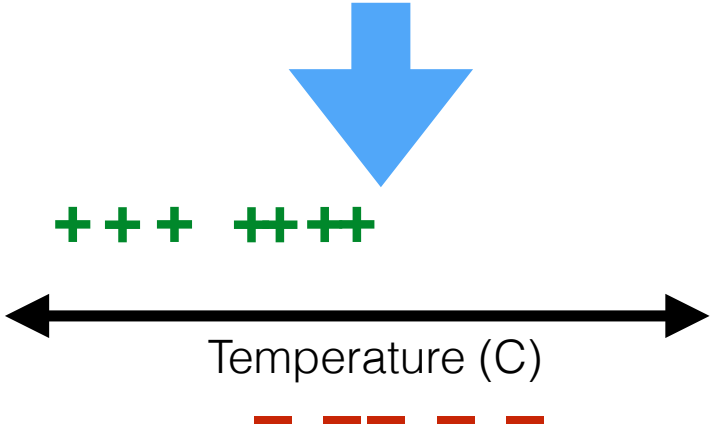
Capturing uncertainty

1 feature:

$$g(x) = \sigma(\theta x + \theta_0)$$

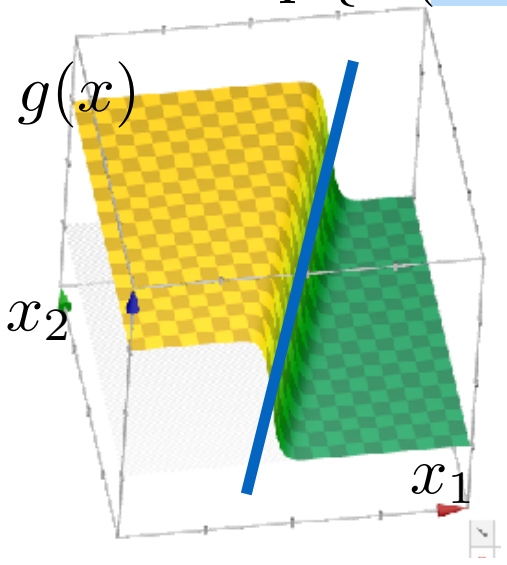
$$= \frac{1}{1 + \exp\{-(\theta x + \theta_0)\}}$$





2 features:

$$= \frac{1}{1 + \exp\{-(\theta^\top x + \theta_0)\}}$$

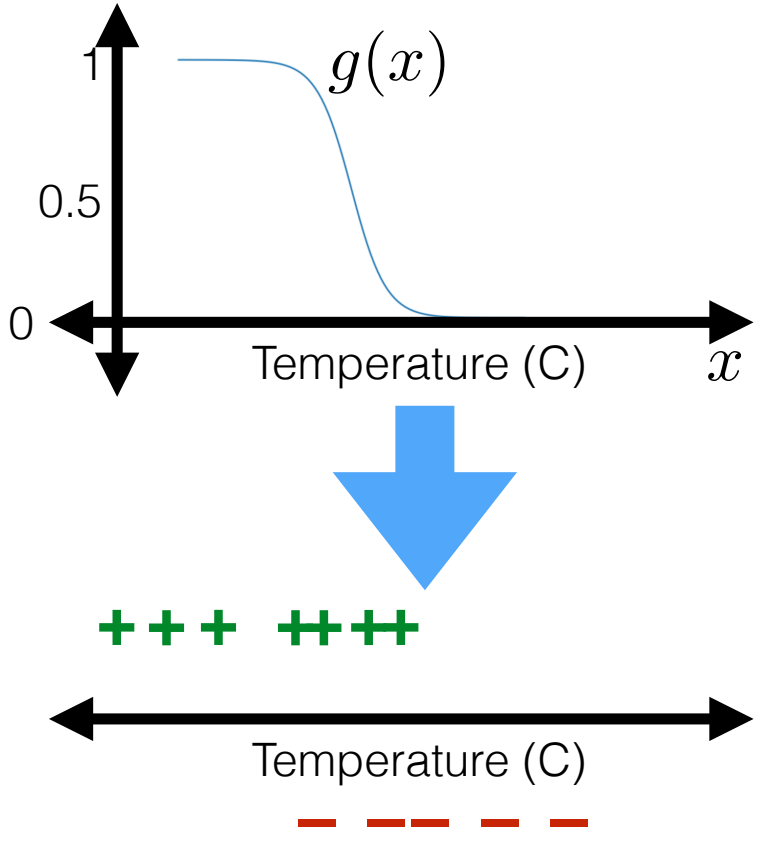


Capturing uncertainty

1 feature:

$$g(x) = \sigma(\theta x + \theta_0)$$

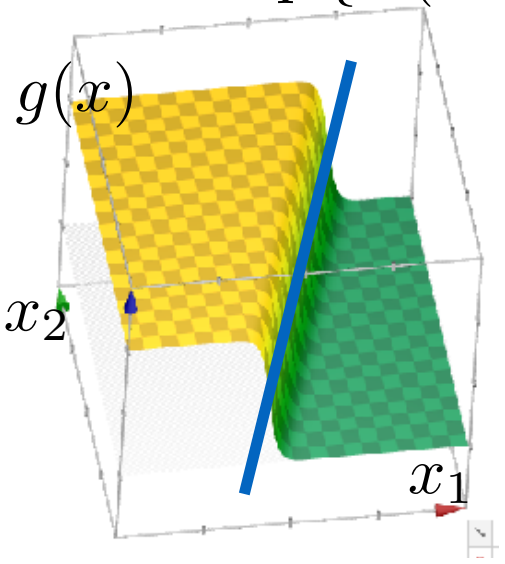
$$= \frac{1}{1 + \exp\{-(\theta x + \theta_0)\}}$$



2 features:

$$g(x) = \sigma(\theta^\top x + \theta_0)$$

$$= \frac{1}{1 + \exp\{-(\theta^\top x + \theta_0)\}}$$

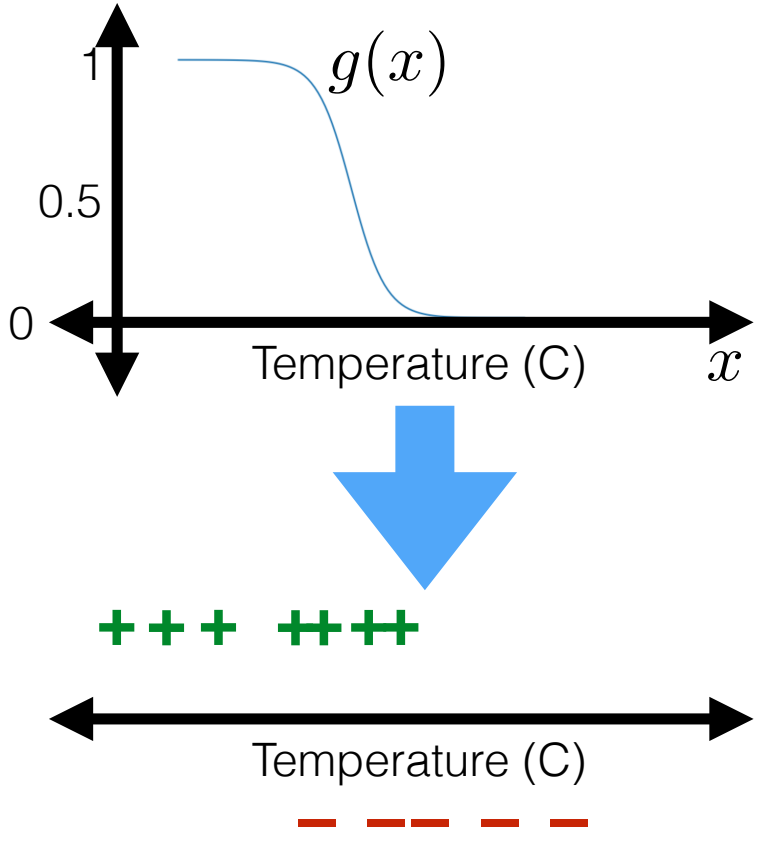


Capturing uncertainty

1 feature:

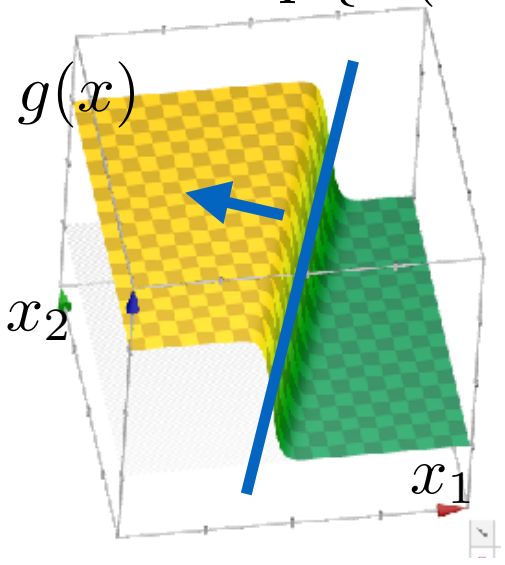
$$g(x) = \sigma(\theta x + \theta_0)$$

$$= \frac{1}{1 + \exp\{-(\theta x + \theta_0)\}}$$



2 features:

$$= \frac{1}{1 + \exp\{-(\theta^\top x + \theta_0)\}}$$

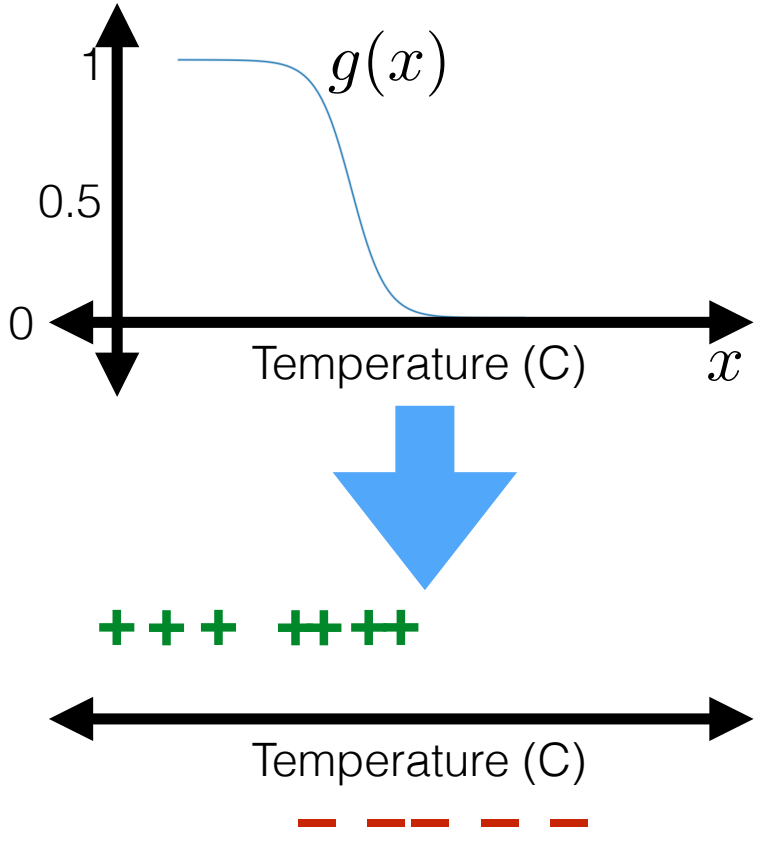


Capturing uncertainty

1 feature:

$$g(x) = \sigma(\theta x + \theta_0)$$

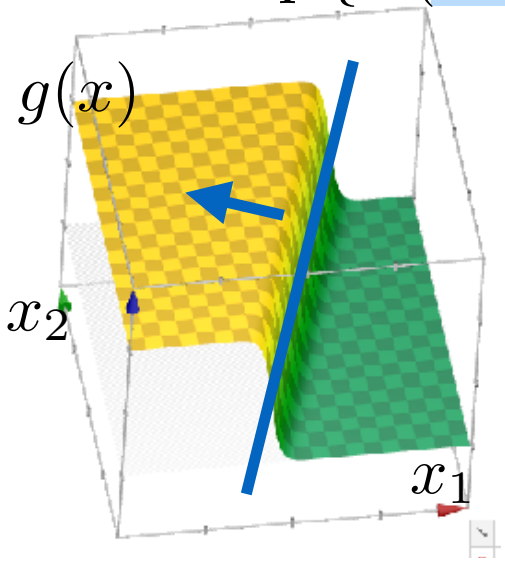
$$= \frac{1}{1 + \exp\{-(\theta x + \theta_0)\}}$$



2 features:

$$g(x) = \sigma(\theta^\top x + \theta_0)$$

$$= \frac{1}{1 + \exp\{-(\theta^\top x + \theta_0)\}}$$

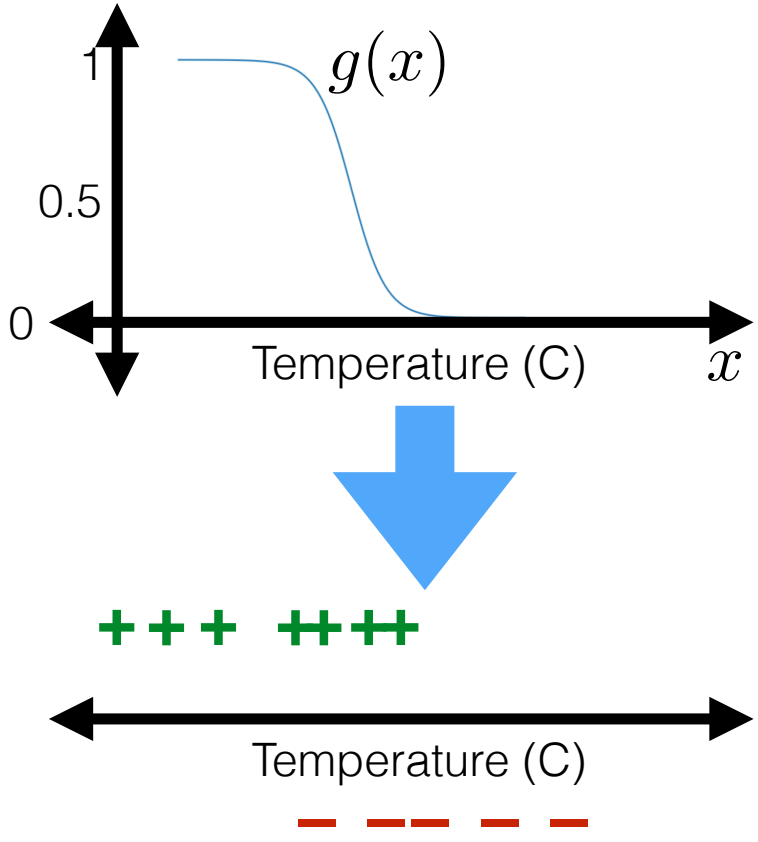


Capturing uncertainty

1 feature:

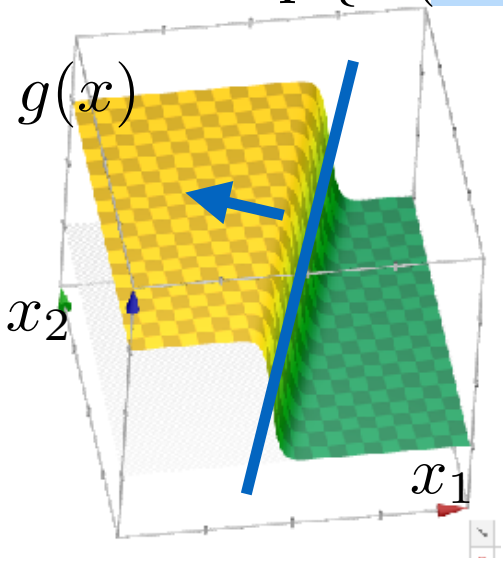
$$g(x) = \sigma(\theta x + \theta_0)$$

$$g(x) = \frac{1}{1 + \exp\{-(\theta x + \theta_0)\}}$$



2 features:

$$g(x) = \frac{1}{1 + \exp\{-(\theta^\top x + \theta_0)\}}$$

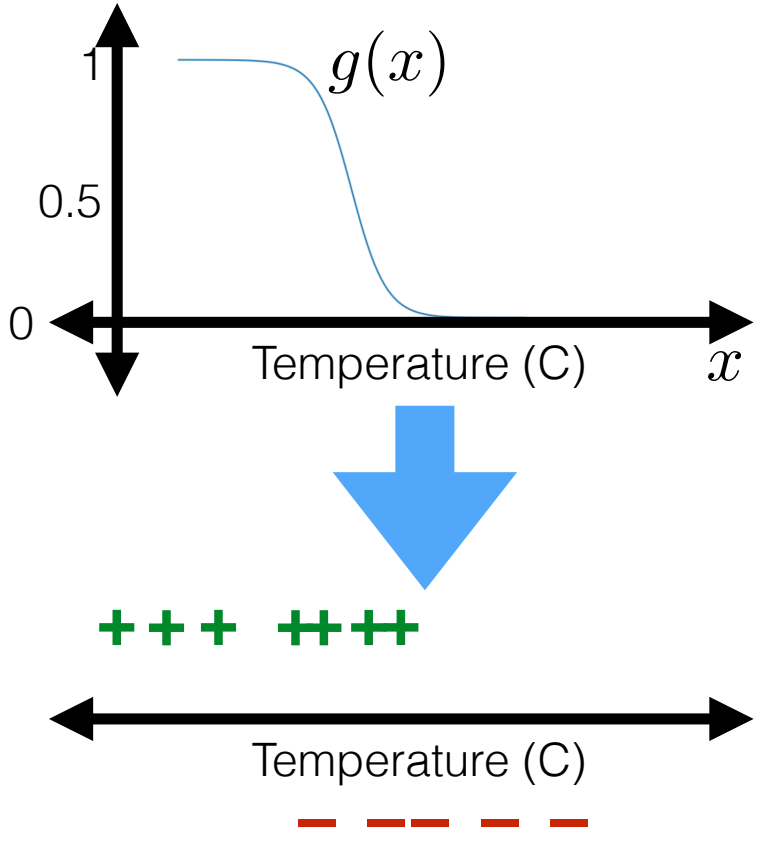


Capturing uncertainty

1 feature:

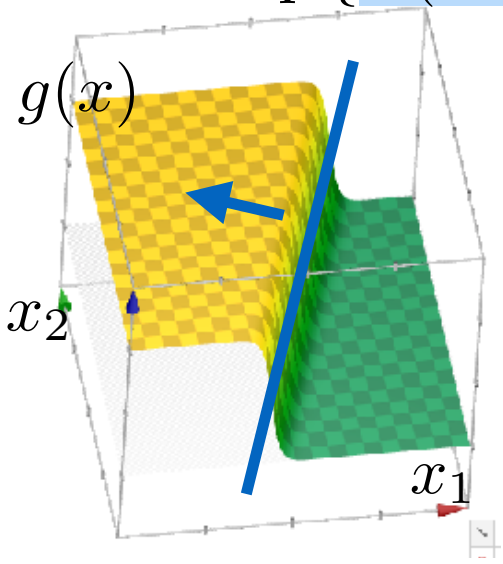
$$g(x) = \sigma(\theta x + \theta_0)$$

$$= \frac{1}{1 + \exp\{-(\theta x + \theta_0)\}}$$



2 features:

$$= \frac{1}{1 + \exp\{-(\theta^\top x + \theta_0)\}}$$

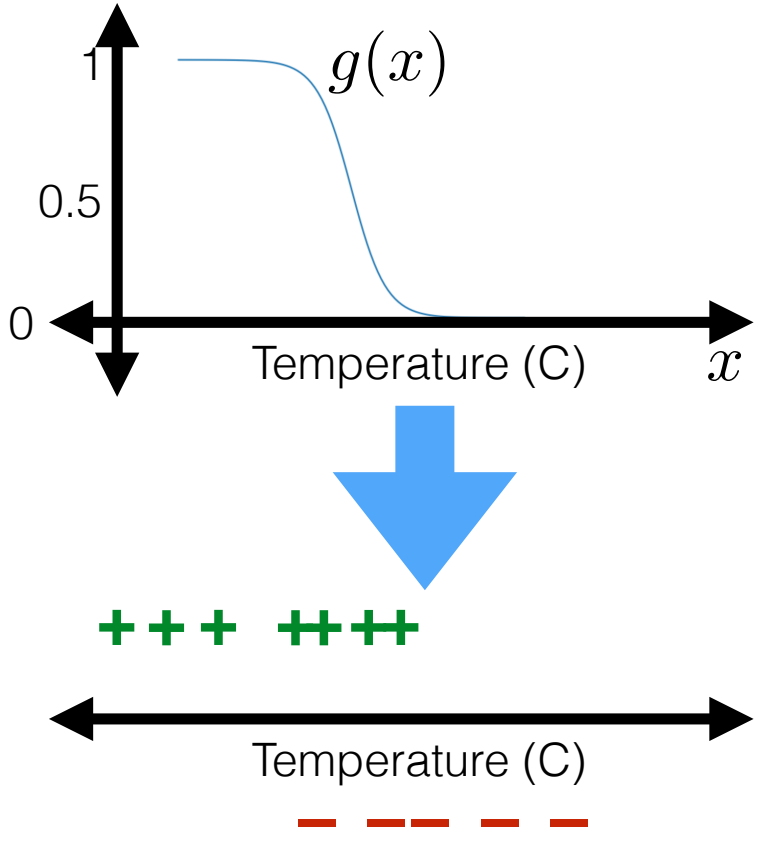


Capturing uncertainty

1 feature:

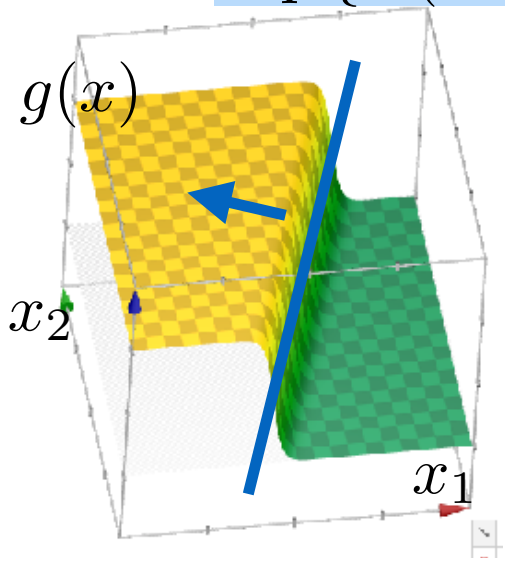
$$g(x) = \sigma(\theta x + \theta_0)$$

$$= \frac{1}{1 + \exp\{-(\theta x + \theta_0)\}}$$



2 features:

$$= \frac{1}{1 + \exp\{-(\theta^\top x + \theta_0)\}}$$

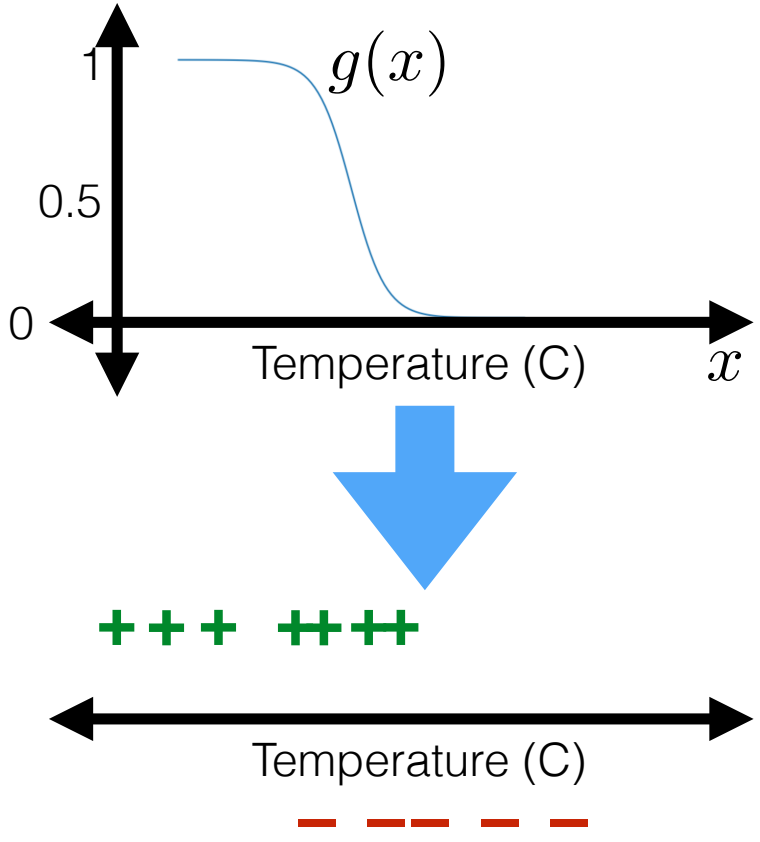


Capturing uncertainty

1 feature:

$$g(x) = \sigma(\theta x + \theta_0)$$

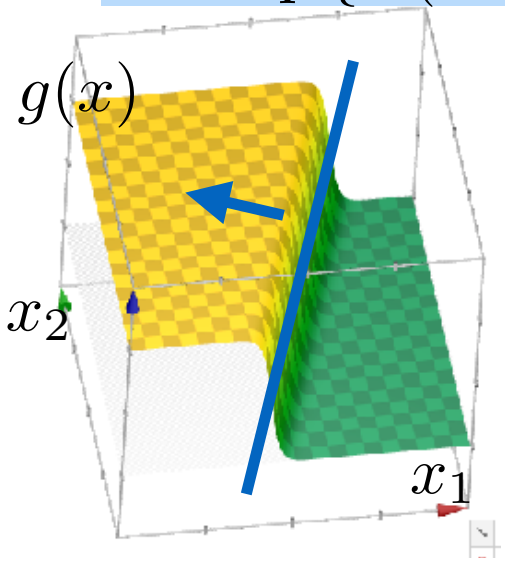
$$= \frac{1}{1 + \exp\{-(\theta x + \theta_0)\}}$$



2 features:

$$g(x) = \sigma(\theta^\top x + \theta_0)$$

$$= \frac{1}{1 + \exp\{-(\theta^\top x + \theta_0)\}}$$

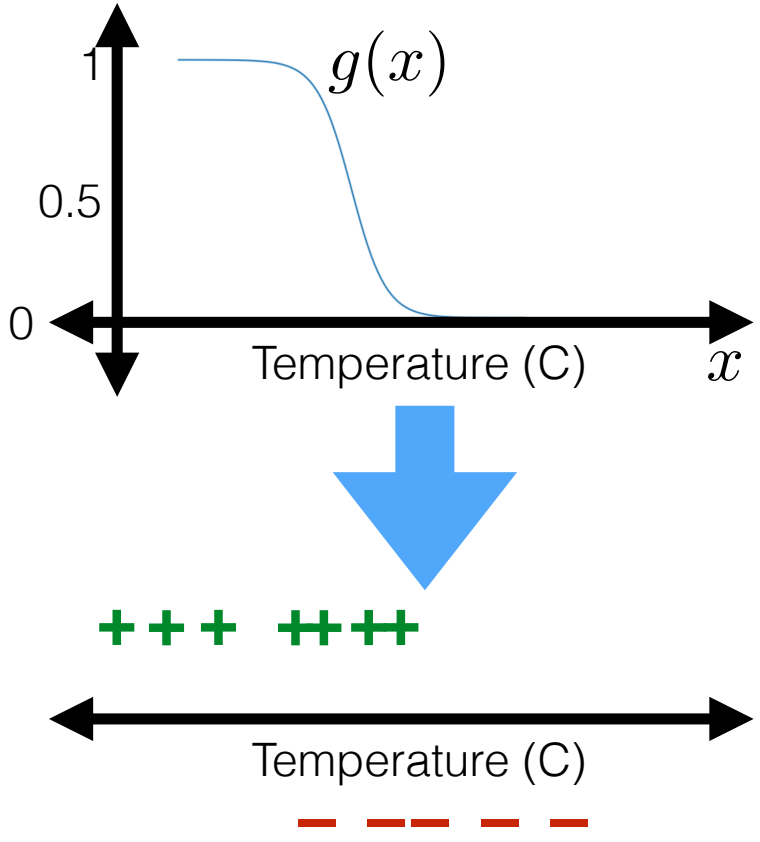


Capturing uncertainty

1 feature:

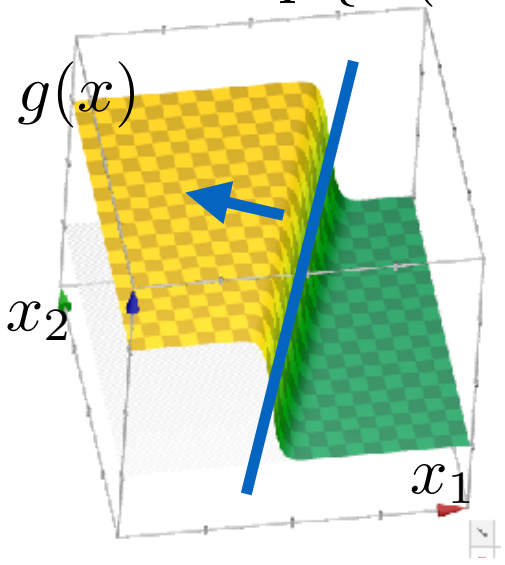
$$g(x) = \sigma(\theta x + \theta_0)$$

$$= \frac{1}{1 + \exp\{-(\theta x + \theta_0)\}}$$



2 features:

$$= \frac{1}{1 + \exp\{-(\theta^\top x + \theta_0)\}}$$

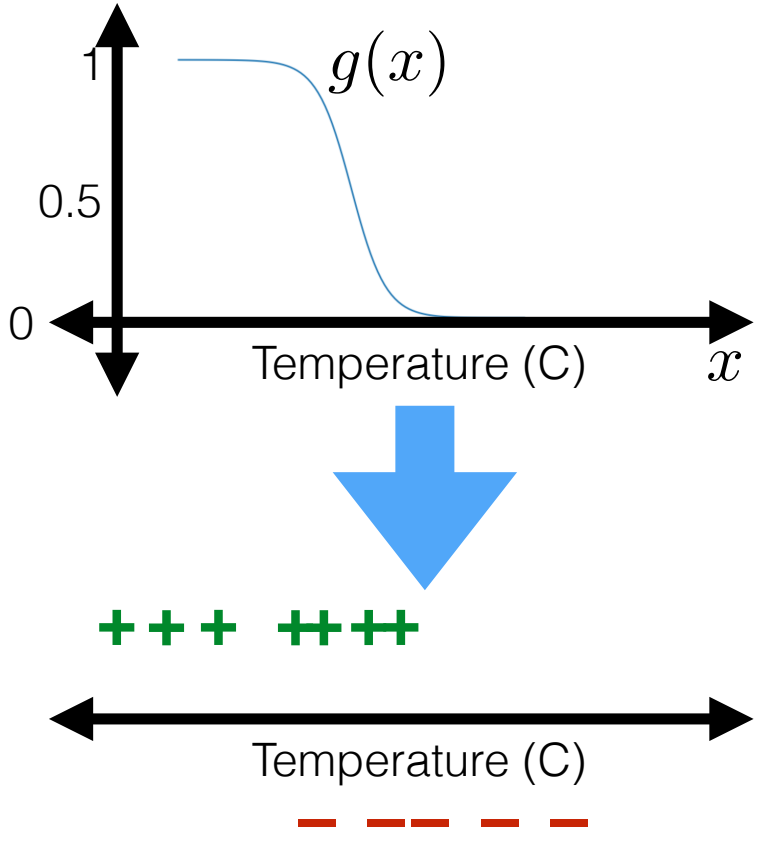


Capturing uncertainty

1 feature:

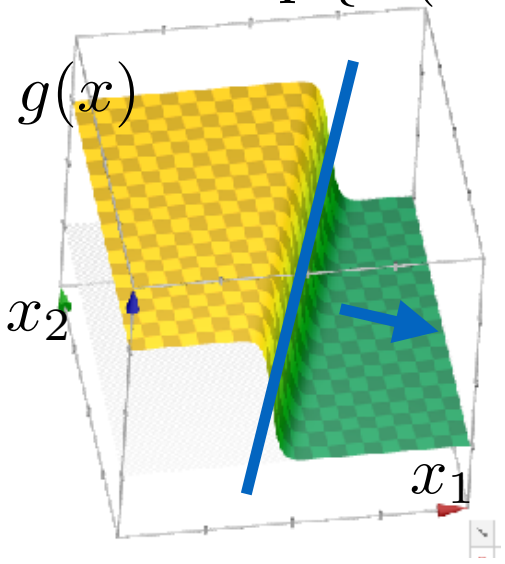
$$g(x) = \sigma(\theta x + \theta_0)$$

$$= \frac{1}{1 + \exp\{-(\theta x + \theta_0)\}}$$



2 features:

$$= \frac{1}{1 + \exp\{-(\theta^\top x + \theta_0)\}}$$

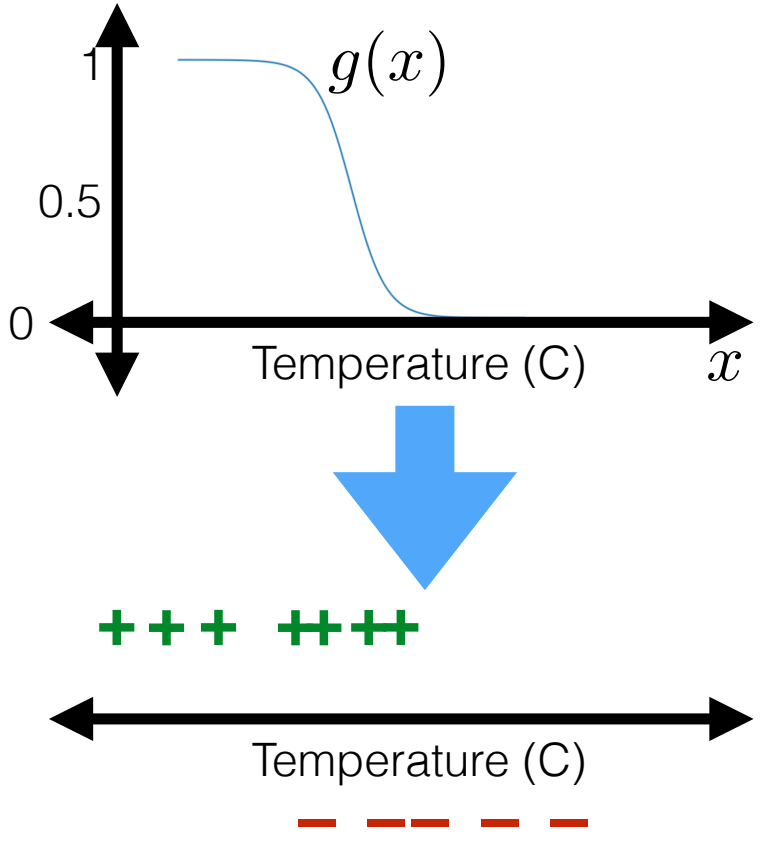


Capturing uncertainty

1 feature:

$$g(x) = \sigma(\theta x + \theta_0)$$

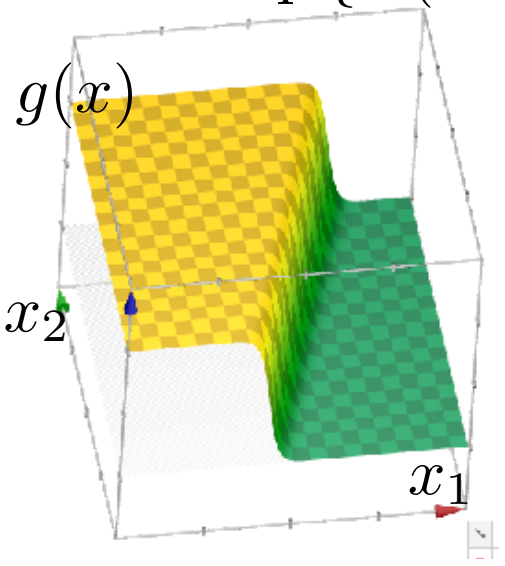
$$= \frac{1}{1 + \exp\{-(\theta x + \theta_0)\}}$$



2 features:

$$g(x) = \sigma(\theta^\top x + \theta_0)$$

$$= \frac{1}{1 + \exp\{-(\theta^\top x + \theta_0)\}}$$

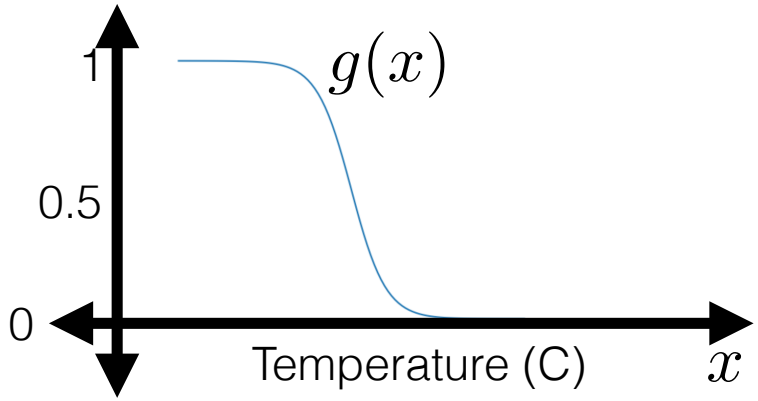


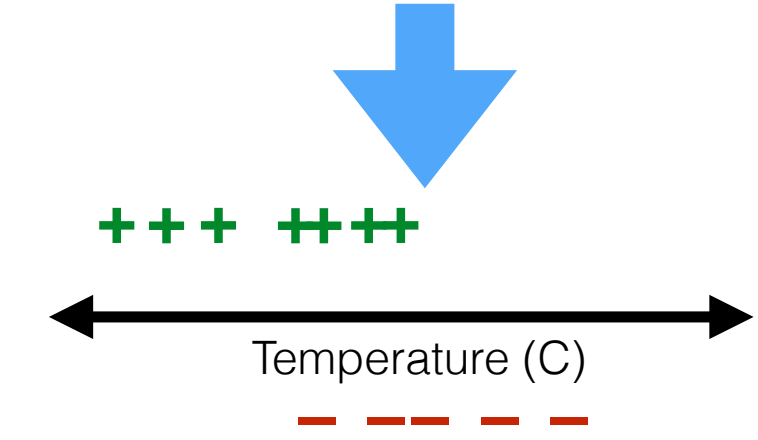
Capturing uncertainty

1 feature:

$$g(x) = \sigma(\theta x + \theta_0)$$

$$= \frac{1}{1 + \exp\{-(\theta x + \theta_0)\}}$$

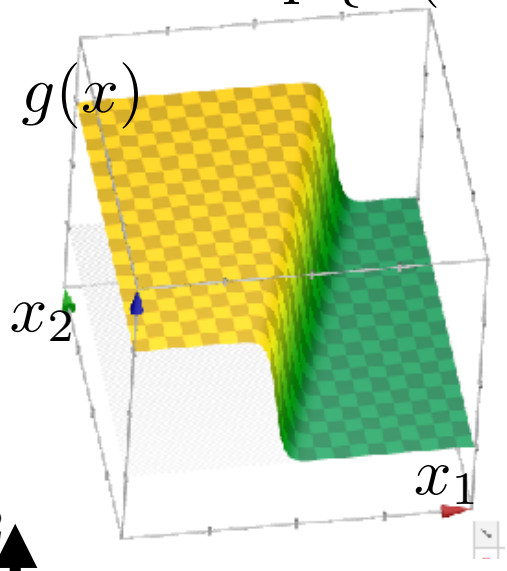


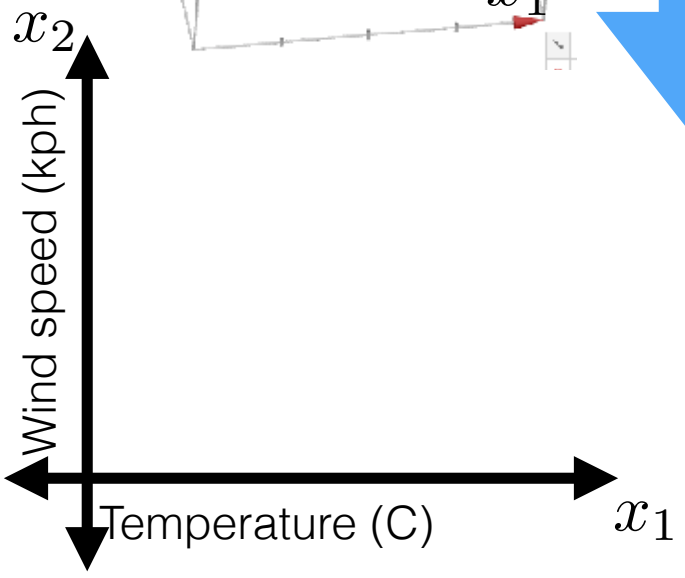


2 features:

$$g(x) = \sigma(\theta^\top x + \theta_0)$$

$$= \frac{1}{1 + \exp\{-(\theta^\top x + \theta_0)\}}$$



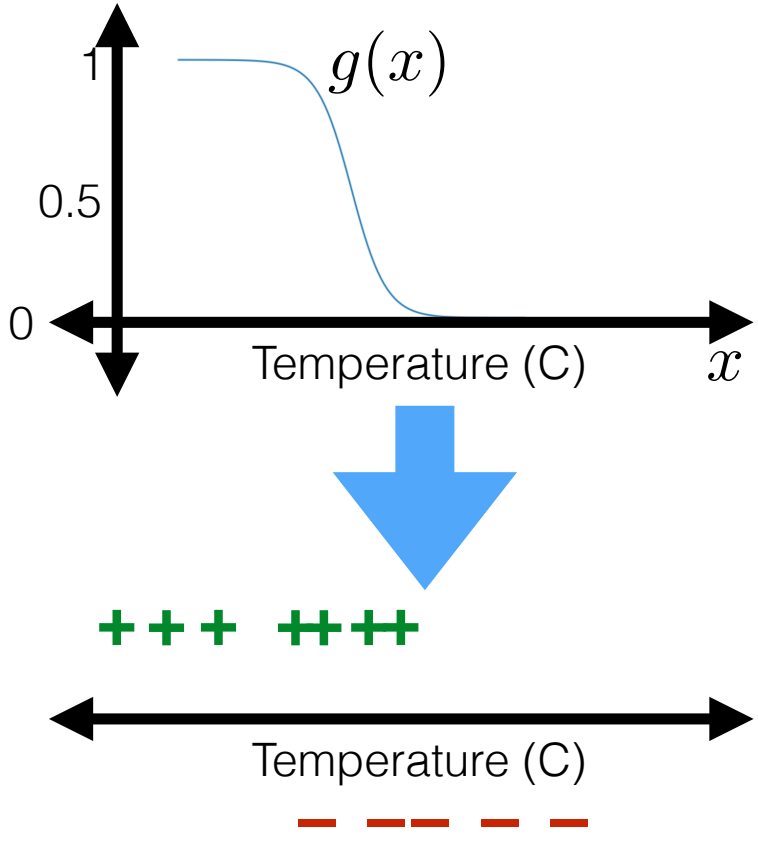


Capturing uncertainty

1 feature:

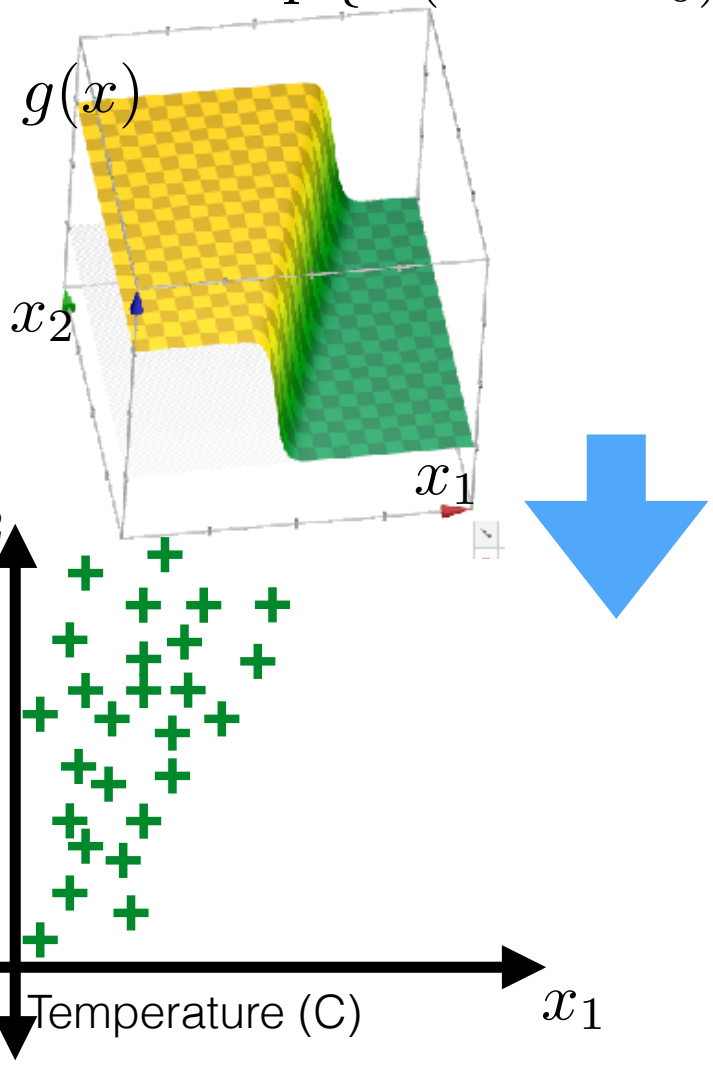
$$g(x) = \sigma(\theta x + \theta_0)$$

$$= \frac{1}{1 + \exp\{-(\theta x + \theta_0)\}}$$



2 features:

$$= \frac{1}{1 + \exp\{-(\theta^\top x + \theta_0)\}}$$

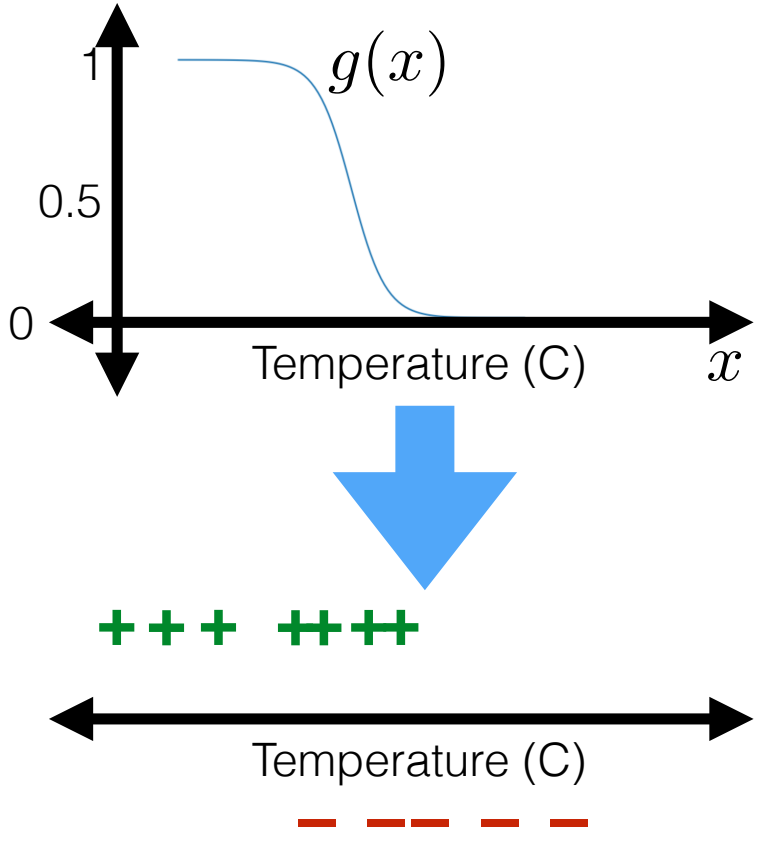


Capturing uncertainty

1 feature:

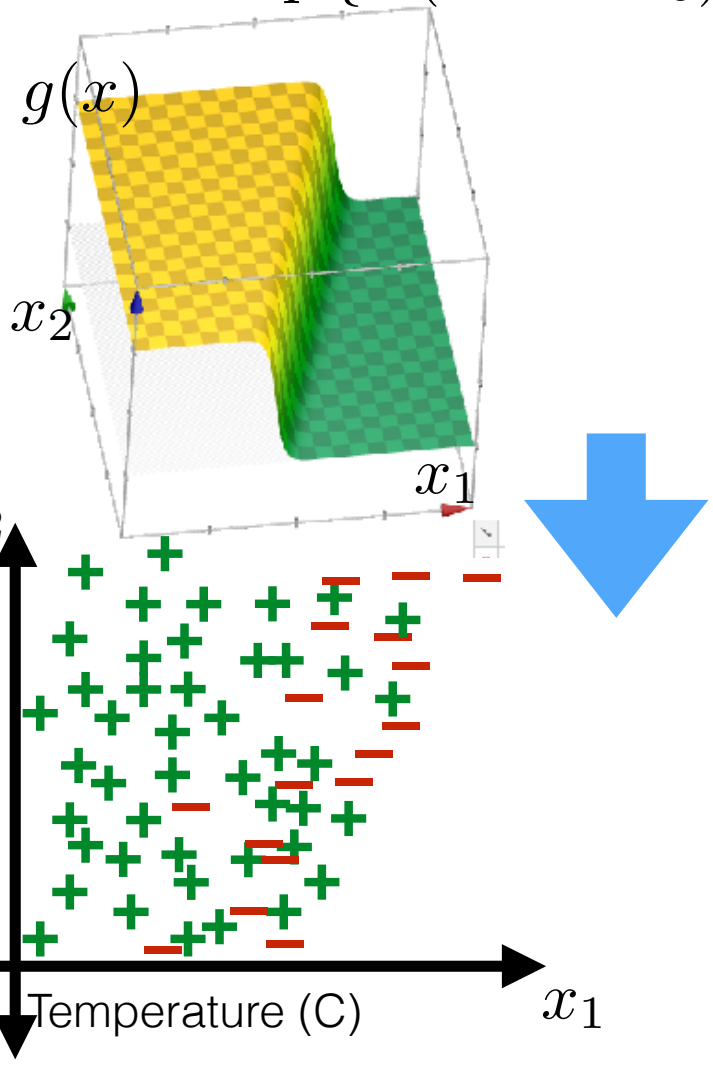
$$g(x) = \sigma(\theta x + \theta_0)$$

$$= \frac{1}{1 + \exp\{-(\theta x + \theta_0)\}}$$



2 features:

$$= \frac{1}{1 + \exp\{-(\theta^\top x + \theta_0)\}}$$

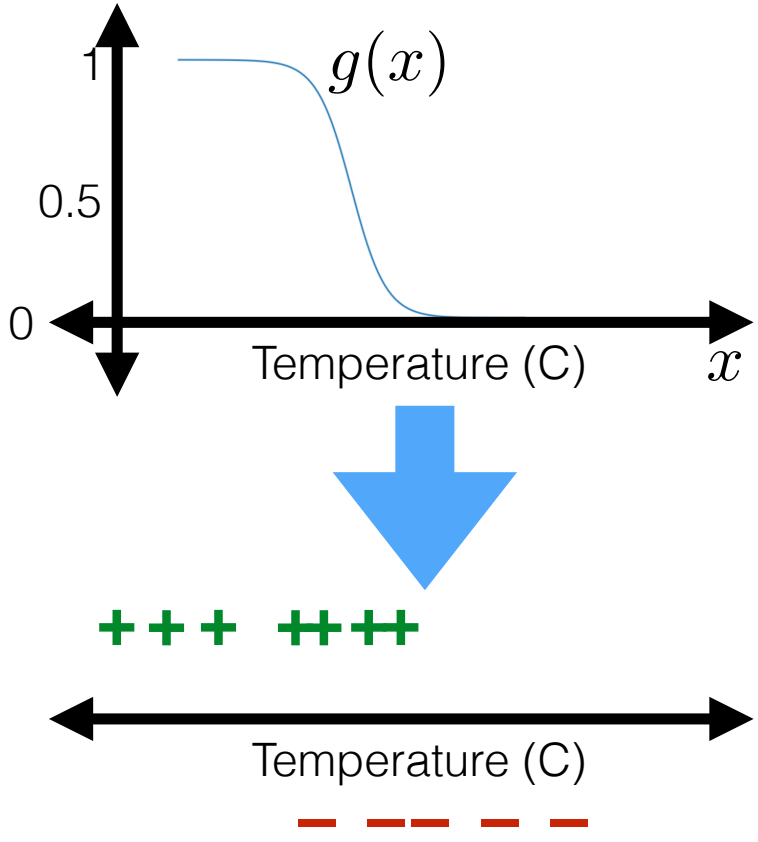


Capturing uncertainty

1 feature:

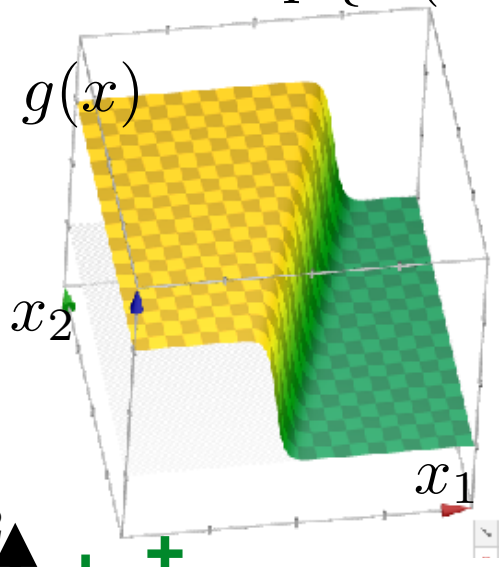
$$g(x) = \sigma(\theta x + \theta_0)$$

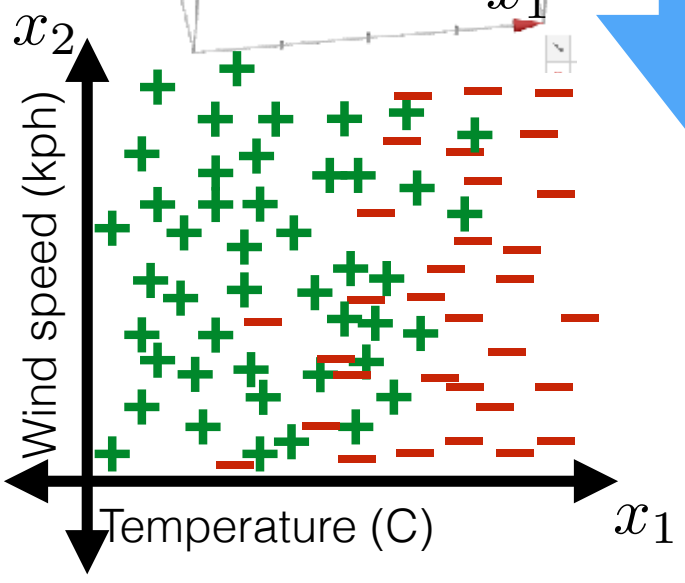
$$= \frac{1}{1 + \exp\{-(\theta x + \theta_0)\}}$$



2 features:

$$= \frac{1}{1 + \exp\{-(\theta^\top x + \theta_0)\}}$$





Linear logistic classification

aka logistic regression

Linear logistic classification

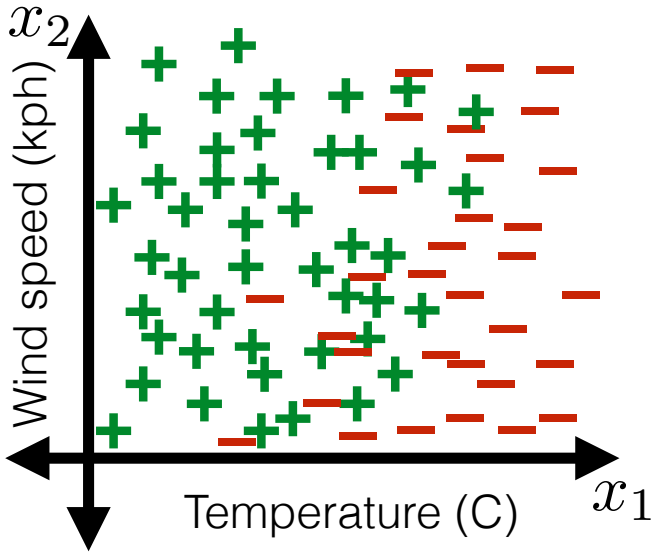
aka logistic regression

- How do we learn a classifier (i.e. learn θ, θ_0)?

Linear logistic classification

aka logistic regression

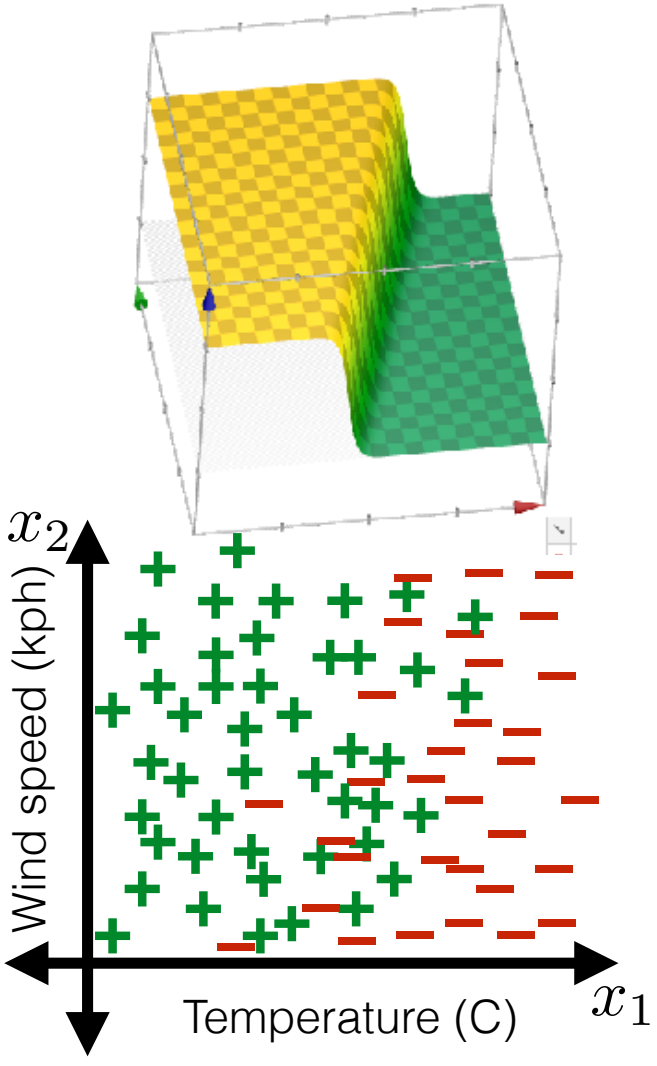
- How do we learn a classifier (i.e. learn θ, θ_0)?



Linear logistic classification

aka logistic reg[illegible]ion

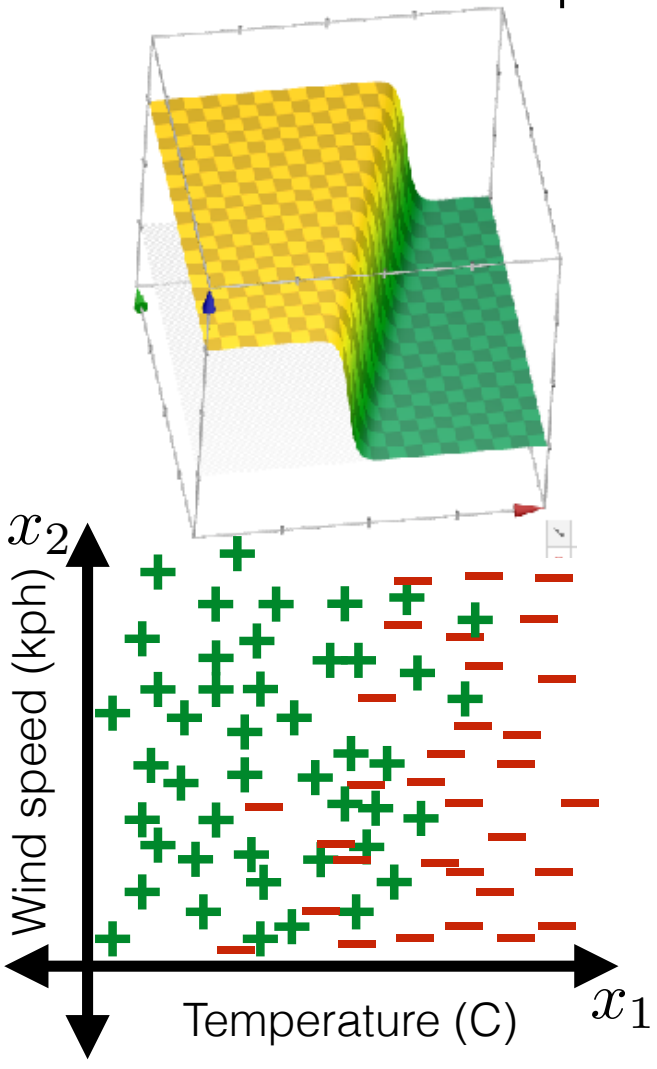
- How do we learn a classifier (i.e. learn θ, θ_0)?



Linear logistic classification

aka logistic reg[illegible]ion

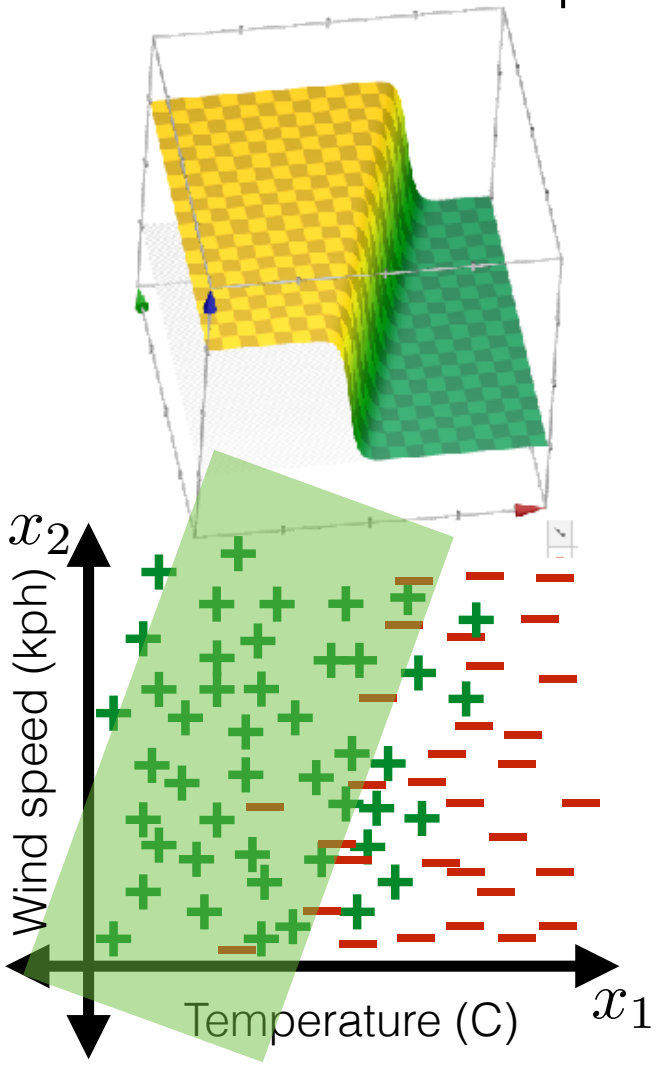
- How do we learn a classifier (i.e. learn θ, θ_0)?
- How do we make predictions?



Linear logistic classification

aka logistic reg...sion

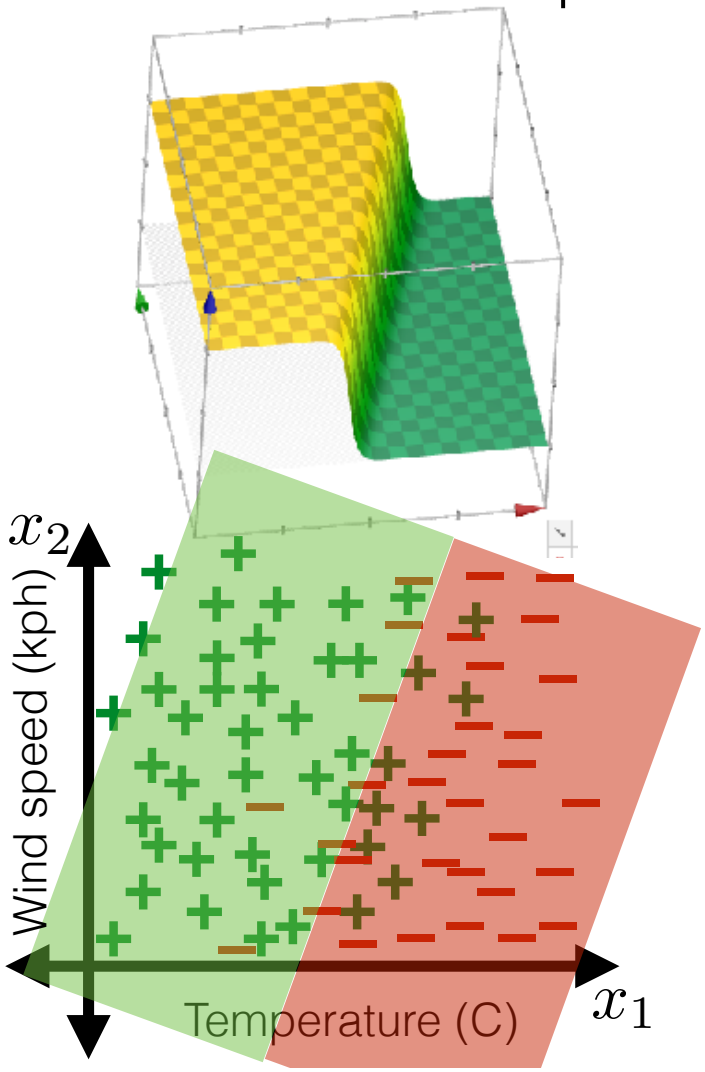
- How do we learn a classifier (i.e. learn θ, θ_0)?
- How do we make predictions?



Linear logistic classification

aka logistic regression

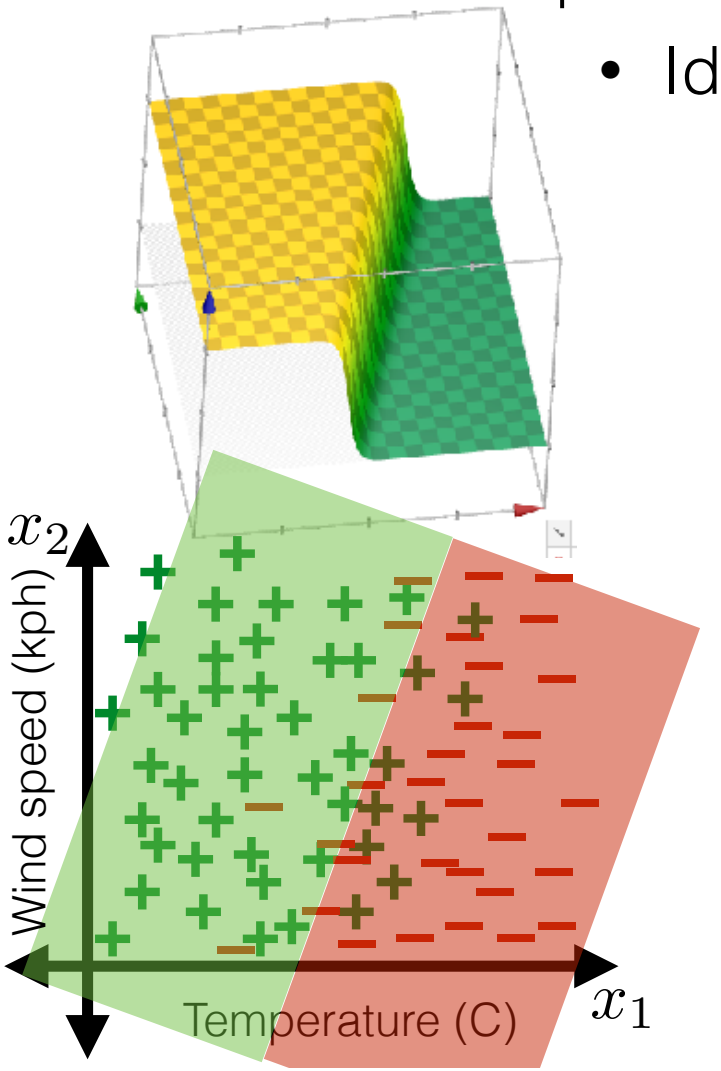
- How do we learn a classifier (i.e. learn θ, θ_0)?
- How do we make predictions?



Linear logistic classification

aka logistic reg...sion

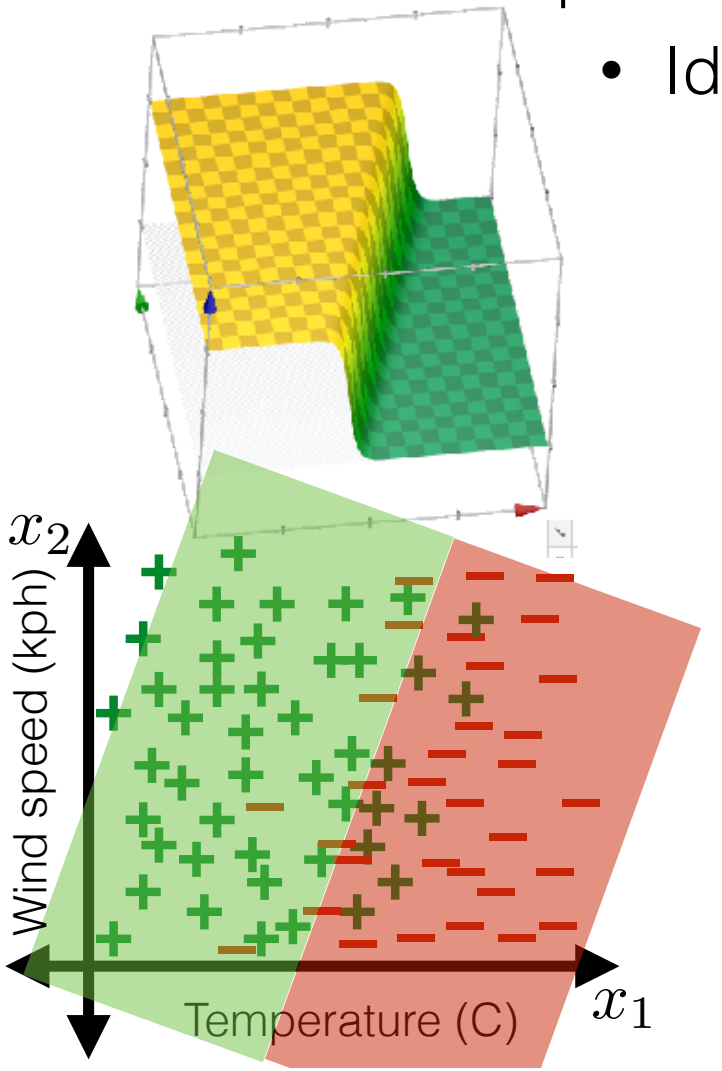
- How do we learn a classifier (i.e. learn θ, θ_0)?
- How do we make predictions?
  - Idea: predict +1 if:



Linear logistic classification

aka logistic reg...sion

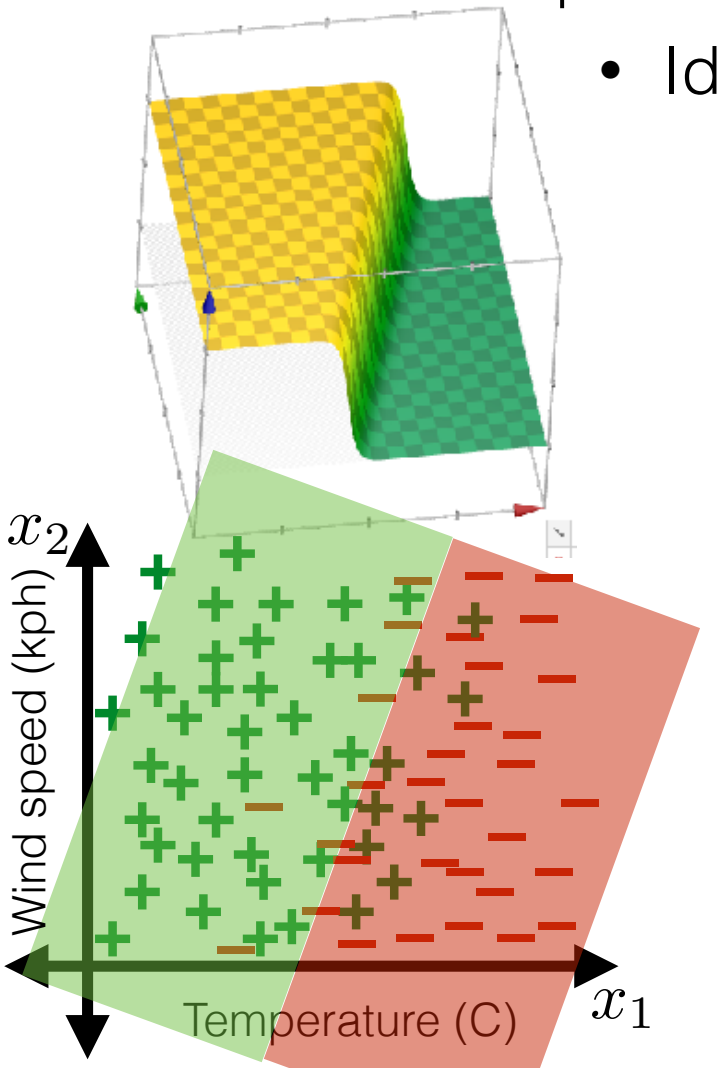
- How do we learn a classifier (i.e. learn θ, θ_0)?
- How do we make predictions?
  - Idea: predict +1 if: probability > 0.5



Linear logistic classification

aka logistic regression

- How do we learn a classifier (i.e. learn θ, θ_0)?
- How do we make predictions?



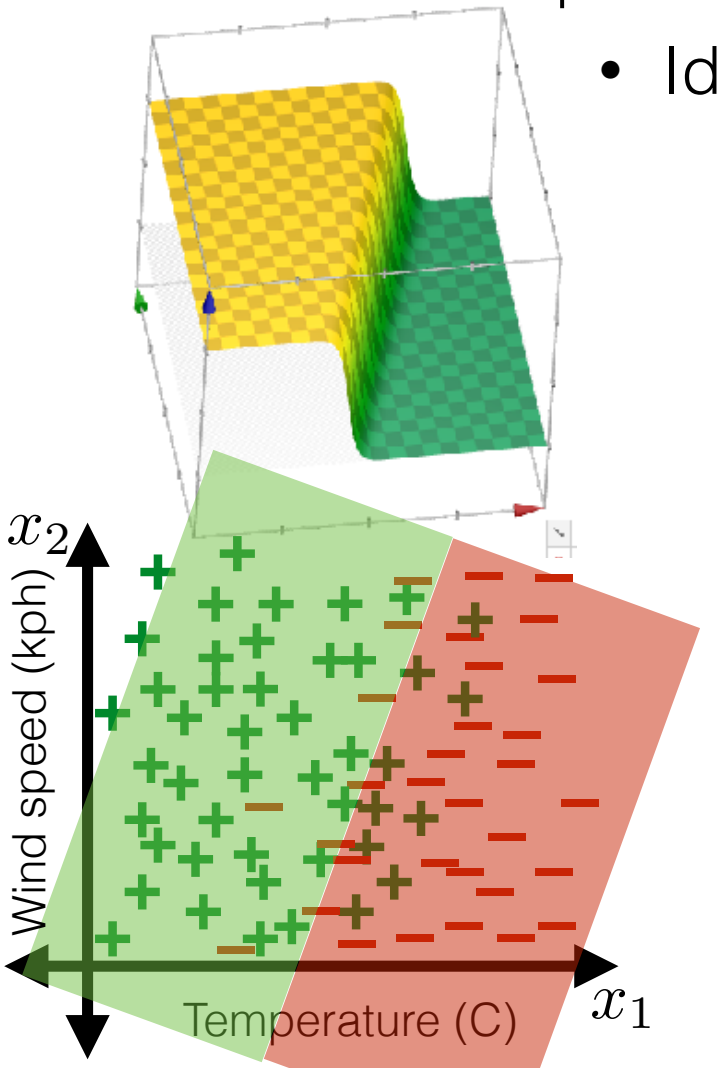
- Idea: predict +1 if: probability > 0.5

$$\sigma(\theta^\top x + \theta_0) > 0.5$$

Linear logistic classification

aka logistic reg...sion

- How do we learn a classifier (i.e. learn θ, θ_0)?
- How do we make predictions?



- Idea: predict +1 if: probability > 0.5

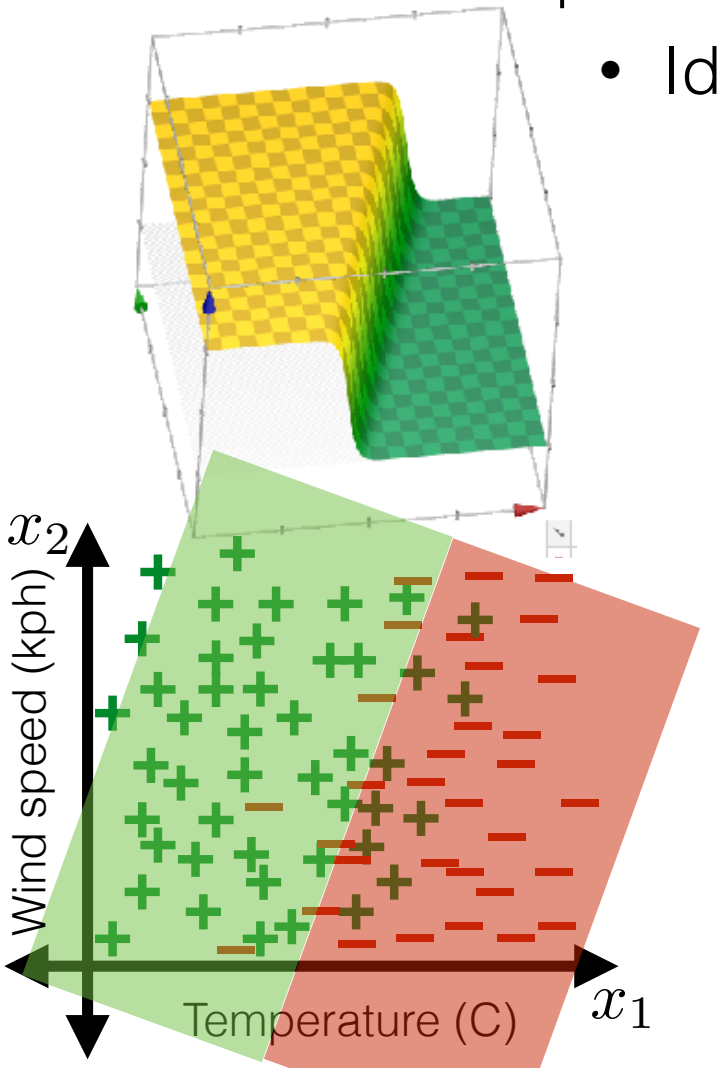
$$\sigma(\theta^\top x + \theta_0) > 0.5$$

$$\frac{1}{1 + \exp\{-(\theta^\top x + \theta_0)\}} > 0.5$$

Linear logistic classification

aka logistic regression

- How do we learn a classifier (i.e. learn θ, θ_0)?
- How do we make predictions?



- Idea: predict +1 if: probability > 0.5

$$\sigma(\theta^\top x + \theta_0) > 0.5$$

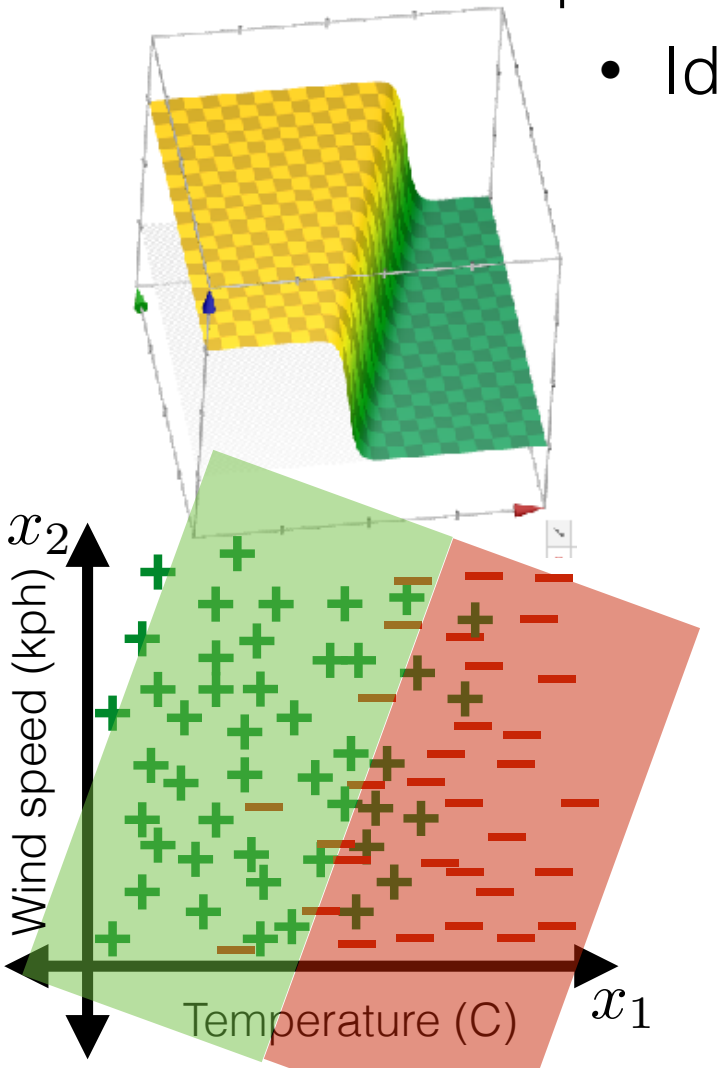
$$\frac{1}{1 + \exp\{-(\theta^\top x + \theta_0)\}} > 0.5$$

$$\exp\{-(\theta^\top x + \theta_0)\} < 1$$

Linear logistic classification

aka logistic regression

- How do we learn a classifier (i.e. learn θ, θ_0)?
- How do we make predictions?



- Idea: predict +1 if: probability > 0.5

$$\sigma(\theta^\top x + \theta_0) > 0.5$$

$$\frac{1}{1 + \exp\{-(\theta^\top x + \theta_0)\}} > 0.5$$

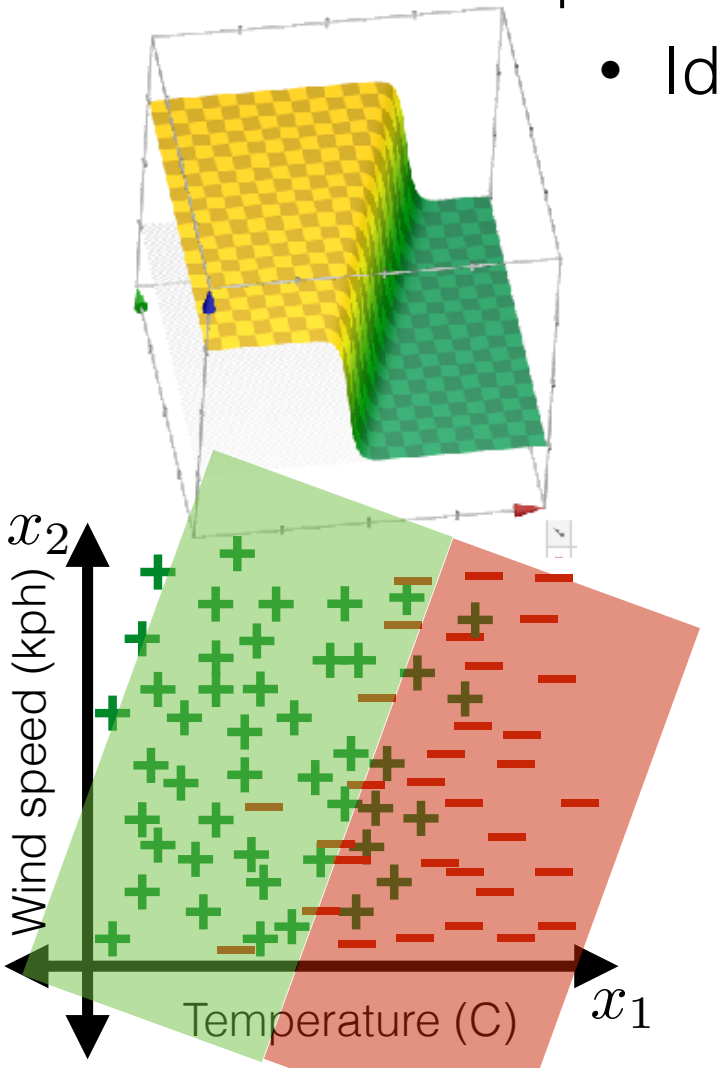
$$\exp\{-(\theta^\top x + \theta_0)\} < 1$$

$$\theta^\top x + \theta_0 > 0$$

Linear logistic classification

aka logistic reg[illegible]ion

- How do we learn a classifier (i.e. learn θ, θ_0)?
- How do we make predictions?



- Idea: predict +1 if: probability > 0.5

$$\sigma(\theta^\top x + \theta_0) > 0.5$$

$$\frac{1}{1 + \exp\{-(\theta^\top x + \theta_0)\}} > 0.5$$

$$\exp\{-(\theta^\top x + \theta_0)\} < 1$$

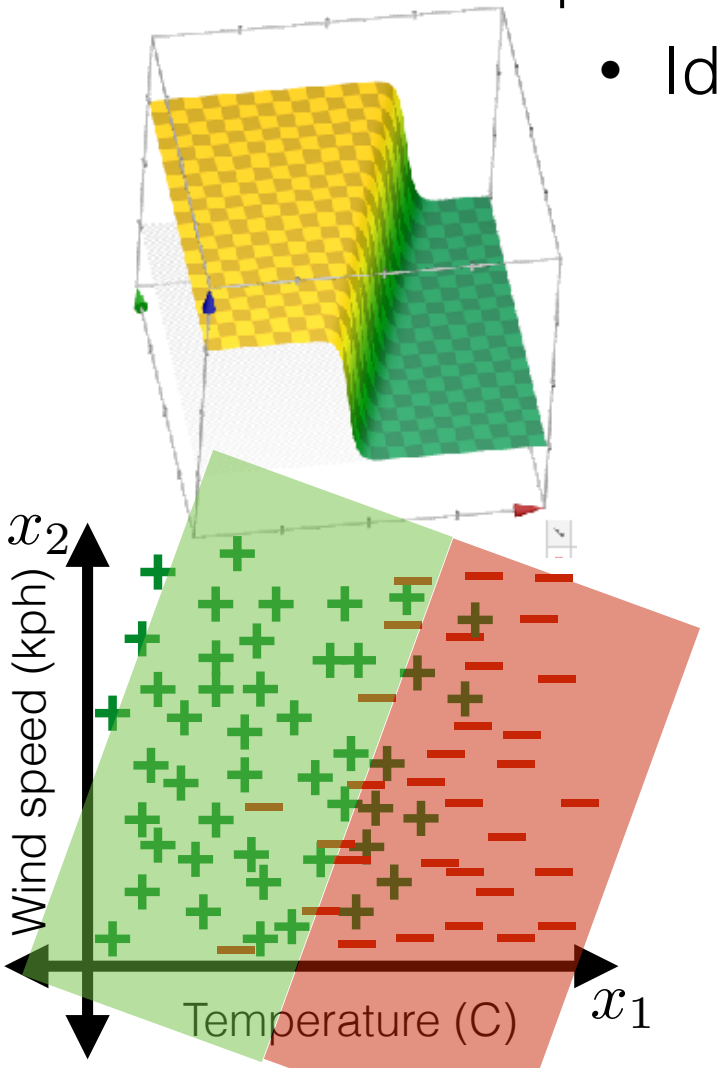
$$\theta^\top x + \theta_0 > 0$$

- Same hypothesis class as before!

Linear logistic classification

aka logistic reg[illegible]ion

- How do we learn a classifier (i.e. learn θ, θ_0)?
- How do we make predictions?



- Idea: predict +1 if: probability > 0.5

$$\sigma(\theta^\top x + \theta_0) > 0.5$$

$$\frac{1}{1 + \exp\{-(\theta^\top x + \theta_0)\}} > 0.5$$

$$\exp\{-(\theta^\top x + \theta_0)\} < 1$$

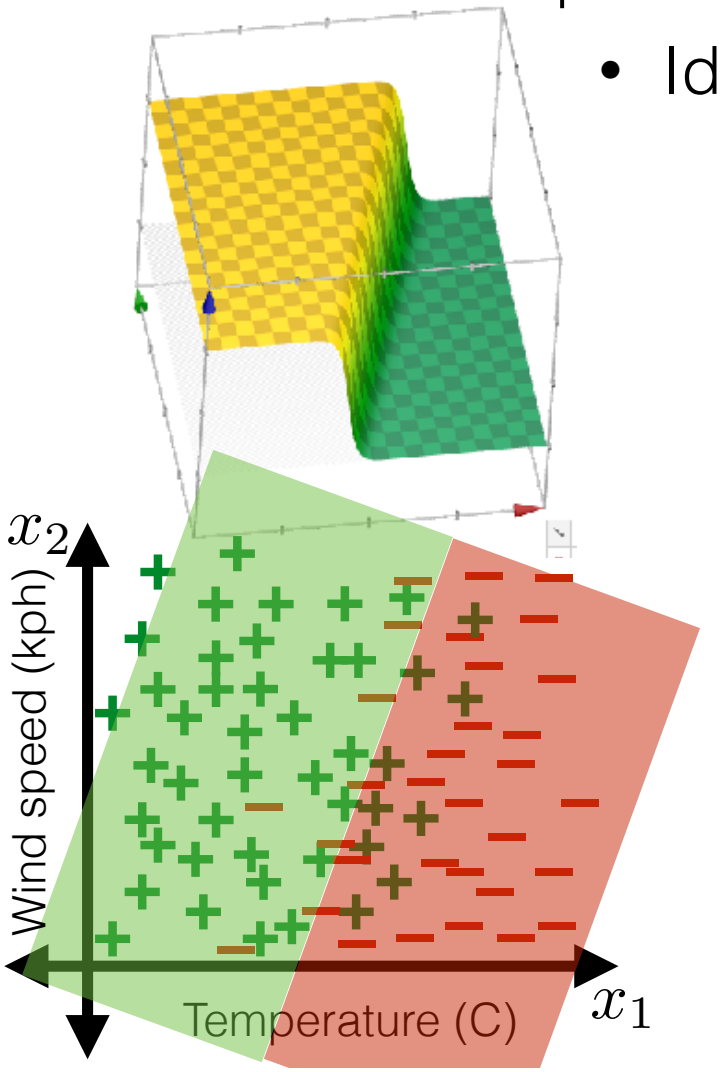
$$\theta^\top x + \theta_0 > 0$$

- Same hypothesis class as before! But we will get:

Linear logistic classification

aka logistic regression

- How do we learn a classifier (i.e. learn θ, θ_0)?
- How do we make predictions?



- Idea: predict +1 if: probability > 0.5

$$\sigma(\theta^\top x + \theta_0) > 0.5$$

$$\frac{1}{1 + \exp\{-(\theta^\top x + \theta_0)\}} > 0.5$$

$$\exp\{-(\theta^\top x + \theta_0)\} < 1$$

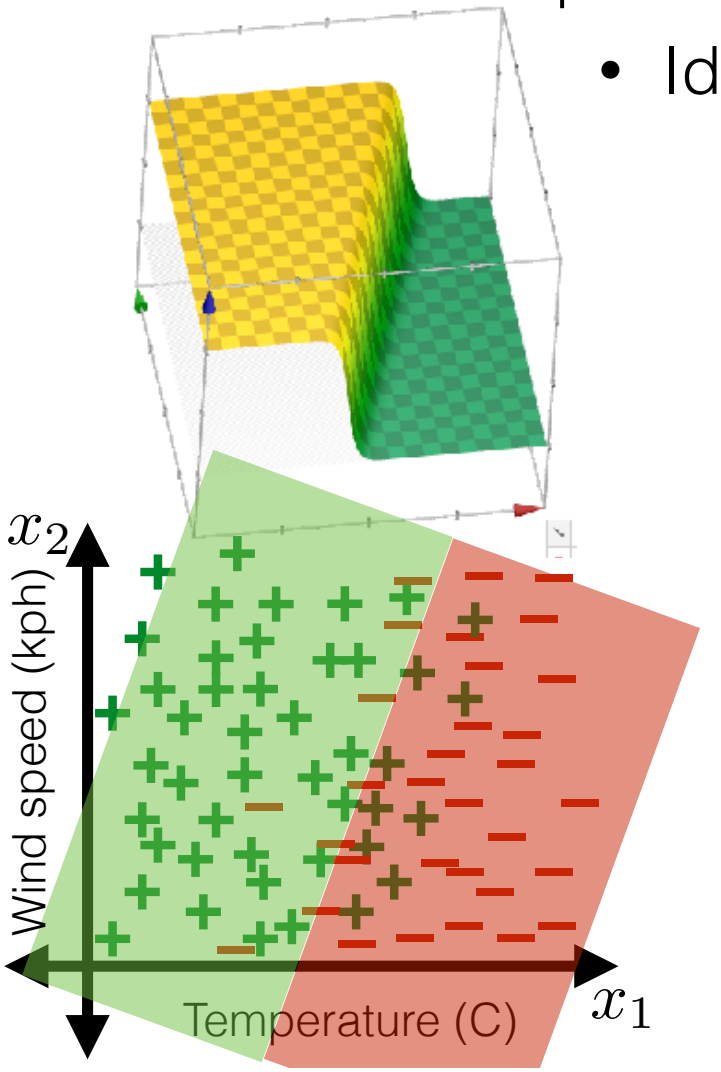
$$\theta^\top x + \theta_0 > 0$$

- Same hypothesis class as before! But we will get:
  - Uncertainties

Linear logistic classification

aka logistic reg...sion

- How do we learn a classifier (i.e. learn θ, θ_0)?
- How do we make predictions?



- Idea: predict +1 if: probability > 0.5

$$\sigma(\theta^\top x + \theta_0) > 0.5$$

$$\frac{1}{1 + \exp\{-(\theta^\top x + \theta_0)\}} > 0.5$$

$$\exp\{-(\theta^\top x + \theta_0)\} < 1$$

$$\theta^\top x + \theta_0 > 0$$

- Same hypothesis class as before! But we will get:
  - Uncertainties
  - Quality guarantees when data not linearly separable

Hvala na pažnji.

Sada idemo na prikaz komandi.

Da li imate pitanja?