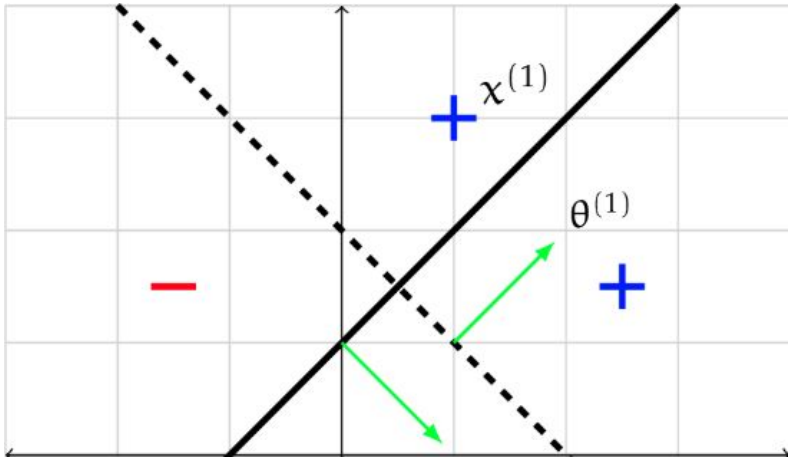
Spring 2023!

Introduction to Machine Learning



Lec-Rec 4: Classification & Logistic Regression

Review: Linear Regression

Given data $D = \{x^{(i)}, y^{(i)}\}_{i=1}^{n}, \quad x^{(i)} \in \mathbb{R}^d, y^{(i)} \in \mathbb{R}$

Define hypothesis class $h(x; \theta) = \theta^\top x + \theta_0$

Define cost function $J(\theta) = \frac{1}{n} \sum_{i=1}^{n} h(x^{(i)}, \theta) - y^{(i)})^2$

Find best hypothesis $\arg\min_{\theta} J(\theta)$

What about classification?

Given data $D = \{x^{(i)}, y^{(i)}\}_{i=1}^{n}, \quad x^{(i)} \in \mathbb{R}^d, y^{(i)} \in \mathbb{R}$

Define hypothesis class $h(x; \theta) = \theta^\top x + \theta_0$

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

Define cost function $J(\theta) = \frac{1}{n} \sum_{i=1}^{n} h(x^{(i)}, \theta) - y^{(i)})^2$

Find best hypothesis $\arg\min_{\theta} J(\theta)$

???

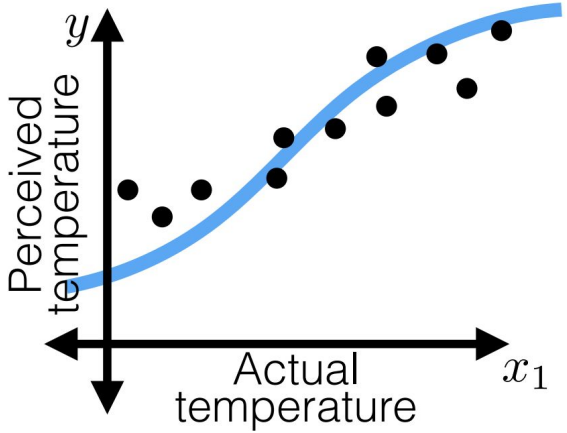
Example

Regression

- Datum i: feature vector

$x^{(i)} = (x_1^{(i)}, \ldots, x_d^{(i)})^\top \in \mathbb{R}^d$

  - Label $y^{(i)} \in \mathbb{R}$
- Hypothesis $h : \mathbb{R}^d \to \mathbb{R}$



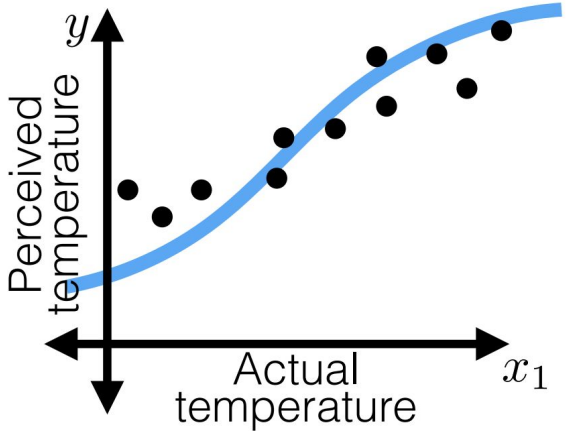
Example

Regression

- Datum i: feature vector

$x^{(i)} = (x_1^{(i)}, \ldots, x_d^{(i)})^\top \in \mathbb{R}^d$

  - Label $y^{(i)} \in \mathbb{R}$
- Hypothesis $h : \mathbb{R}^d \to \mathbb{R}$

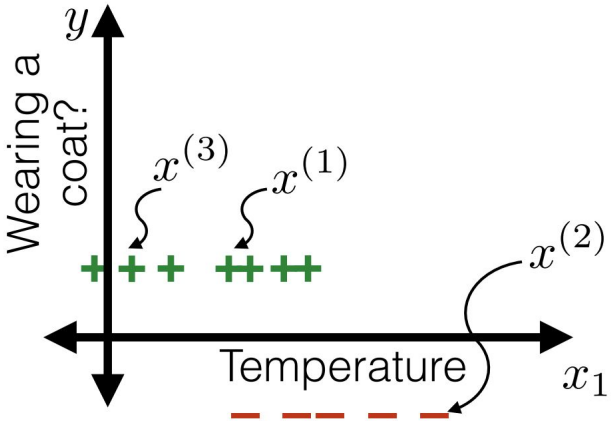


(Two-class) Classification

- Datum i : feature vector

$x^{(i)} = (x_1^{(i)}, \ldots, x_d^{(i)})^\top \in \mathbb{R}^d$

  - Label $y^{(i)} \in \{-1, +1\}$
- Hypothesis $h : \mathbb{R}^d \to \{-1, +1\}$



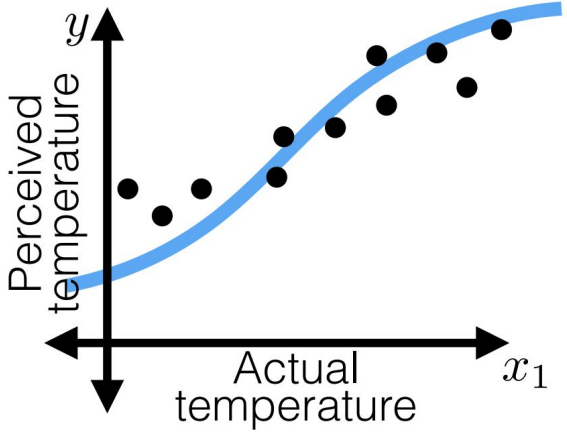
Example

Regression

- Datum i: feature vector

$x^{(i)} = (x_1^{(i)}, \ldots, x_d^{(i)})^\top \in \mathbb{R}^d$

  - Label $y^{(i)} \in \mathbb{R}$
- Hypothesis $h : \mathbb{R}^d \to \mathbb{R}$

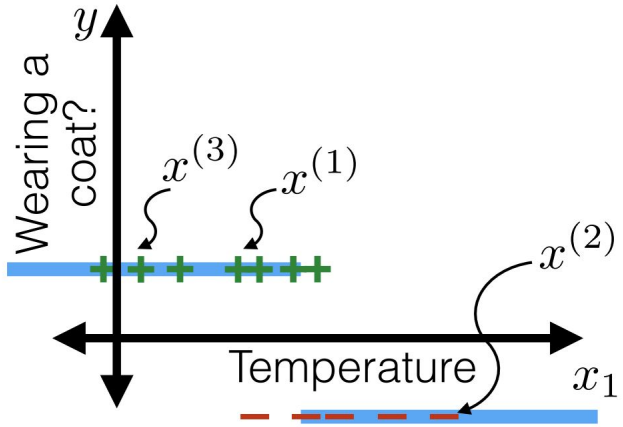


(Two-class) Classification

- Datum i : feature vector

$x^{(i)} = (x_1^{(i)}, \ldots, x_d^{(i)})^\top \in \mathbb{R}^d$

  - Label $y^{(i)} \in \{-1, +1\}$
- Hypothesis $h : \mathbb{R}^d \to \{-1, +1\}$



Linear Classifiers

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

- **Linear classifier**: a hypothesis class which labels +1 on one side of a **hyperplane** and -1 on the other

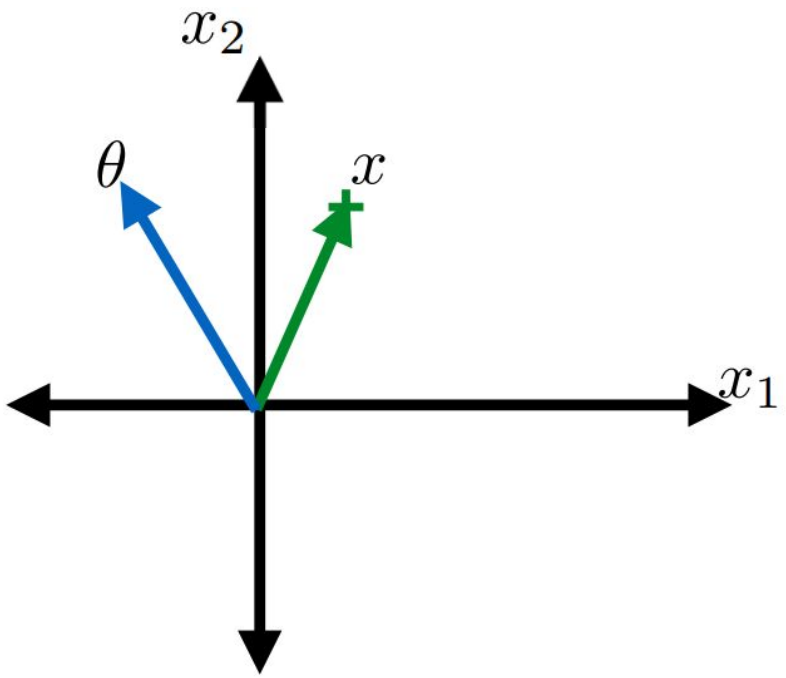
Linear Classifiers

- Hypothesis $h : \mathbb{R}^d \to \{-1, +1\}$
- **Linear classifier**: a hypothesis class which labels +1 on one side of a **hyperplane** and -1 on the other
- Hyperplane: high-dimensional line that can be characterized by:

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

Linear Classifiers

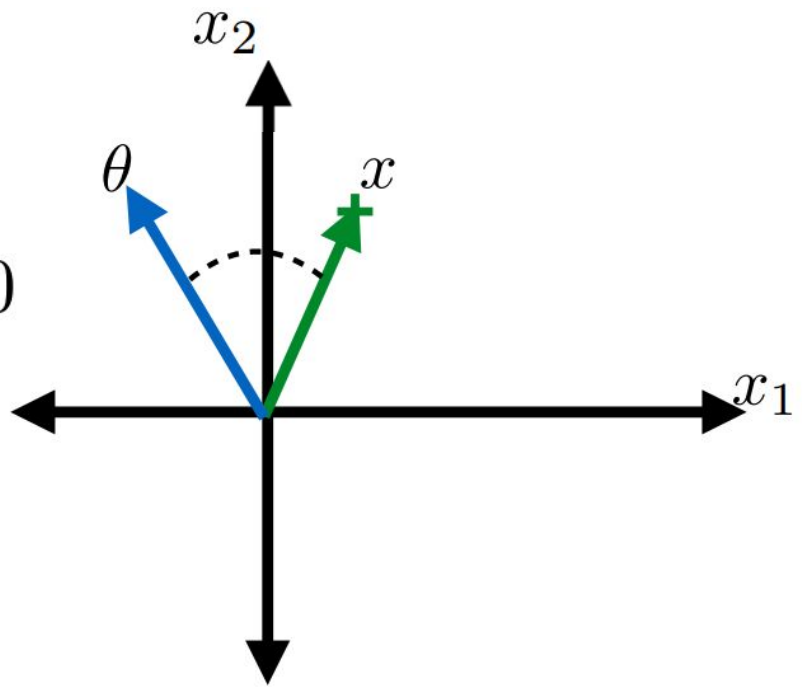
$$\theta^\top x = \|\theta\|\|x\| \cos \alpha$$



Linear Classifiers

$$\theta^\top x = \|\theta\|\|x\| \cos \alpha$$

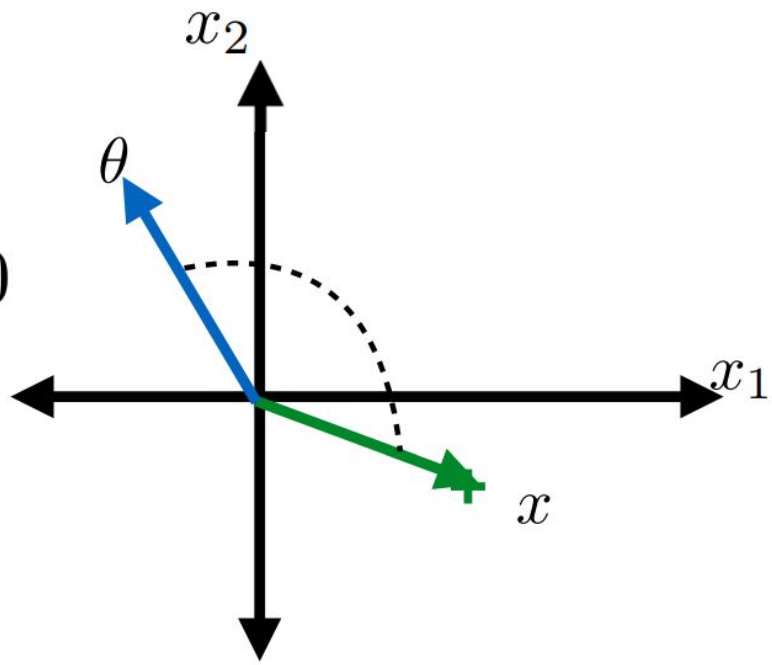
$$\theta^\top x > 0$$



Linear Classifiers

$$\theta^\top x = \|\theta\|\|x\| \cos \alpha$$

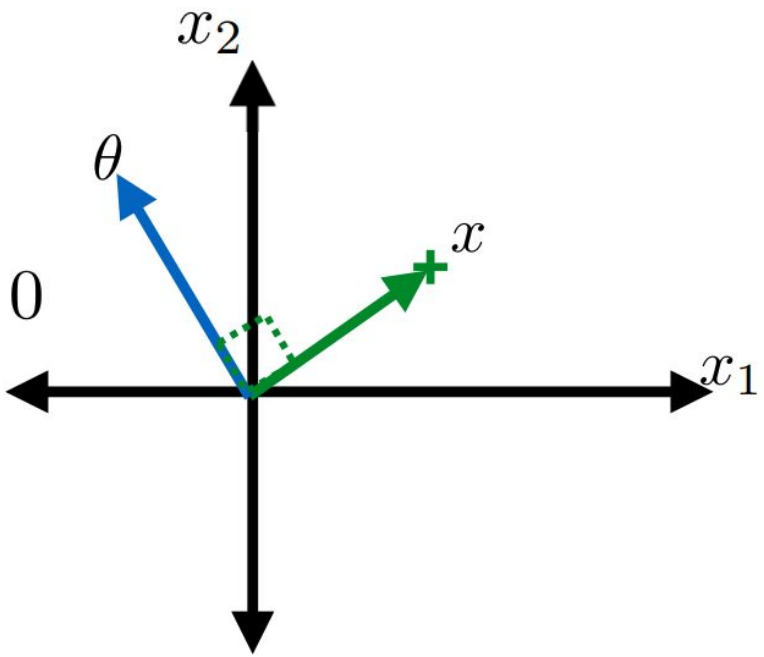
$$\theta^\top x < 0$$



Linear Classifiers

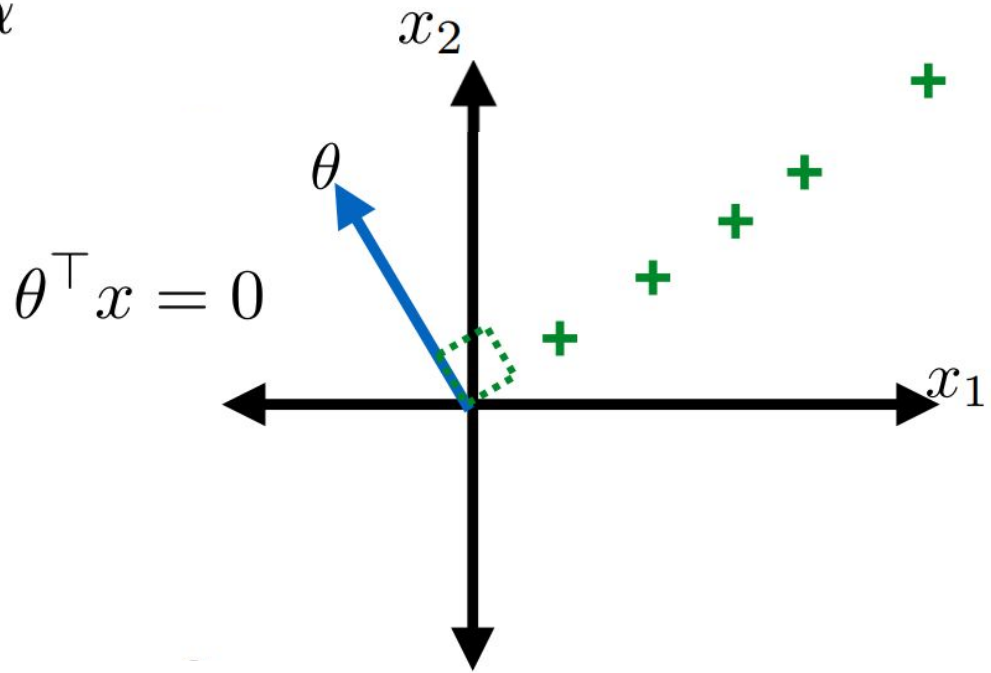
$$\theta^\top x = \|\theta\|\|x\| \cos \alpha$$

$\theta^\top x = 0$



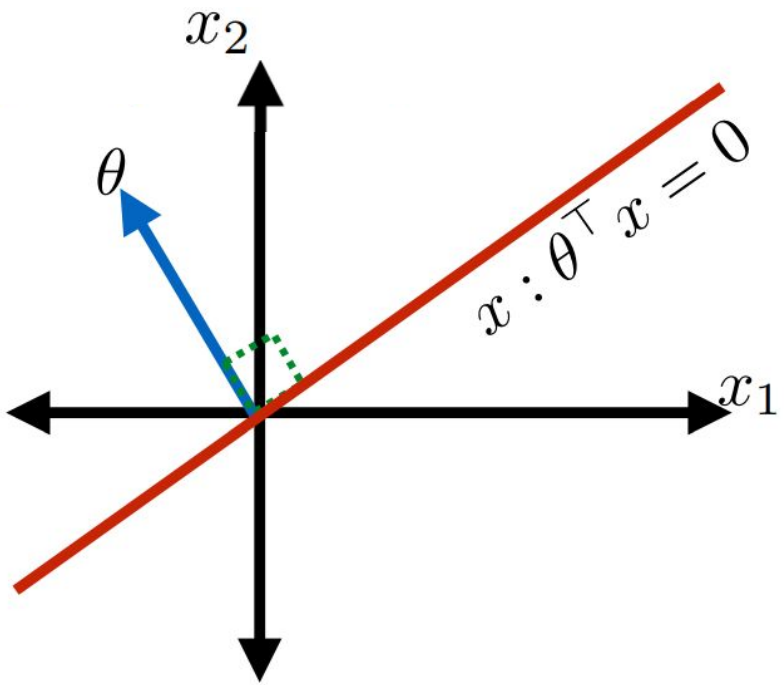
Linear Classifiers

$$\theta^\top x = \|\theta\|\|x\| \cos \alpha$$



Linear Classifiers

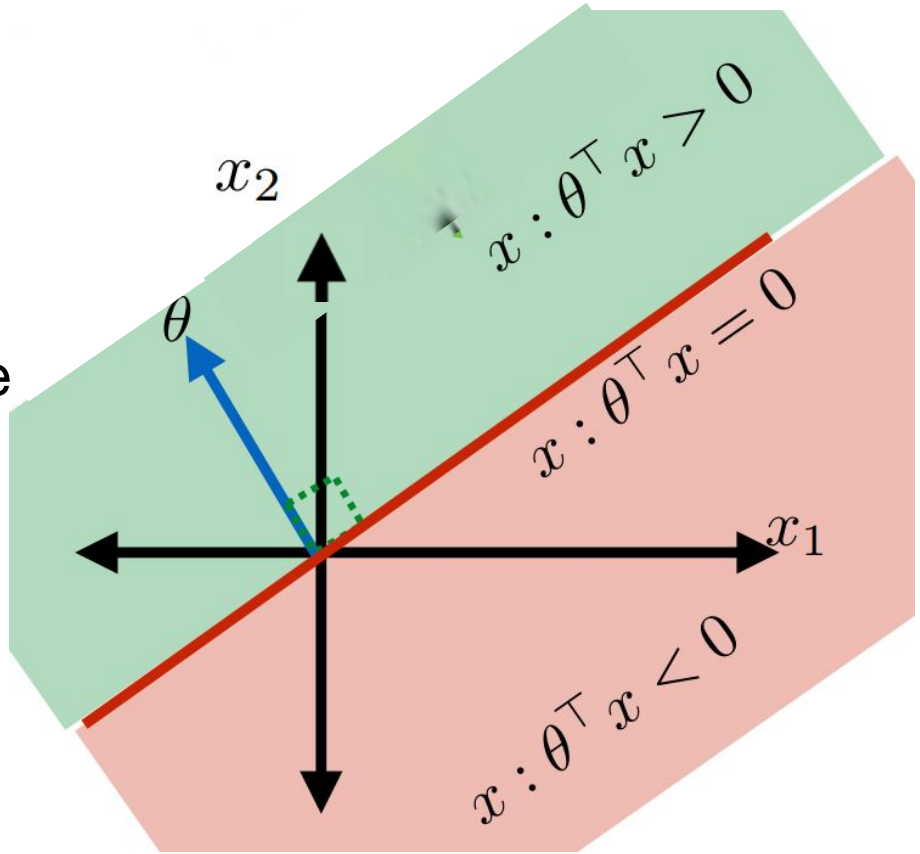
$$\theta^\top x = \|\theta\|\|x\| \cos \alpha$$



Linear Classifiers

$$\theta^\top x = 0$$

- θ is vector that is **normal** (or orthogonal) to the hyperplane

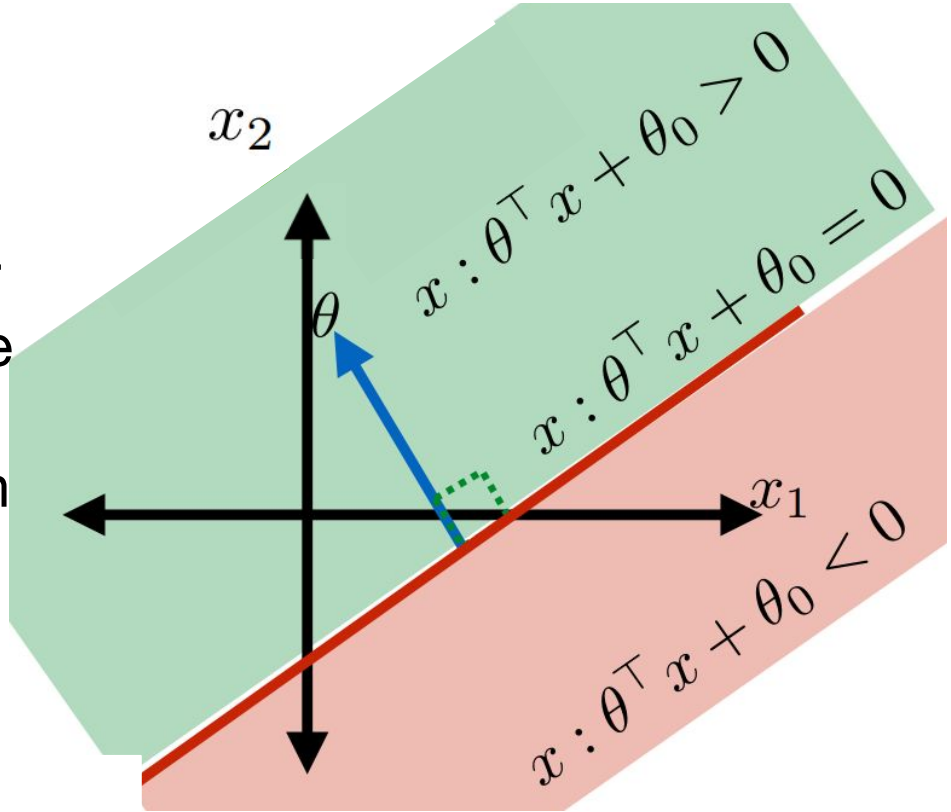


Linear Classifiers

$$\theta^\top x = 0$$

- θ is vector that is **normal** (or orthogonal) to the hyperplane
- θ_0 controls offset from origin

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



Linear Classifiers

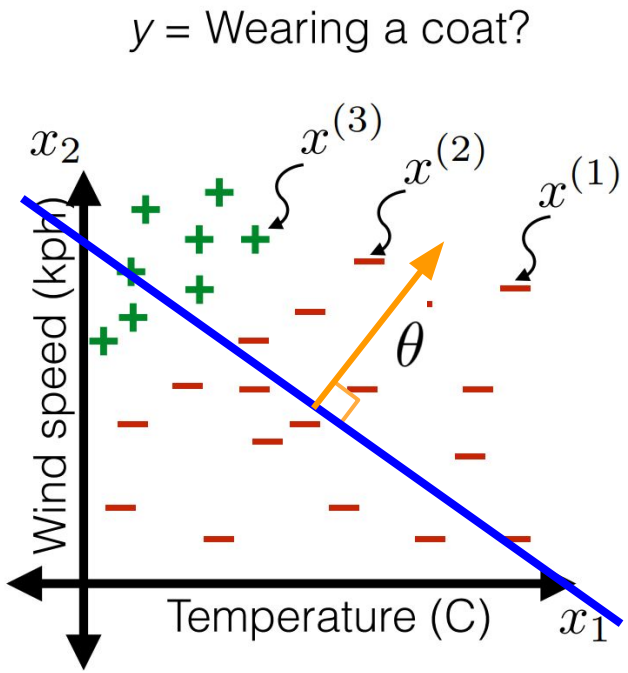
- Hyperplane:

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

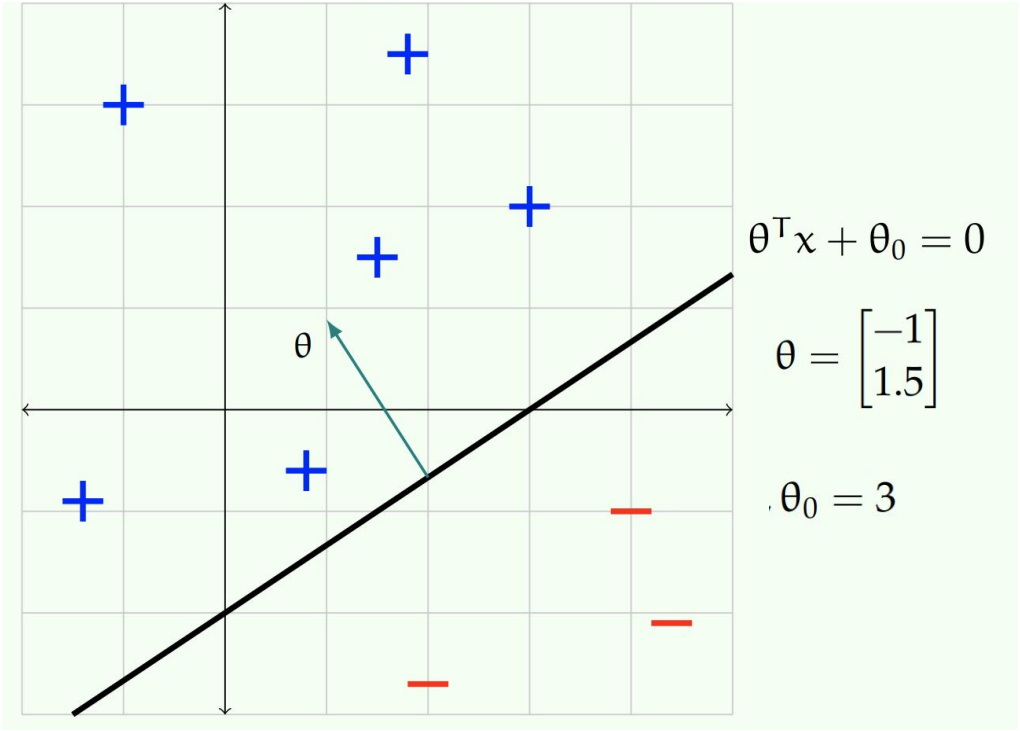
- Decision rule:

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

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



Linear Classifier Example

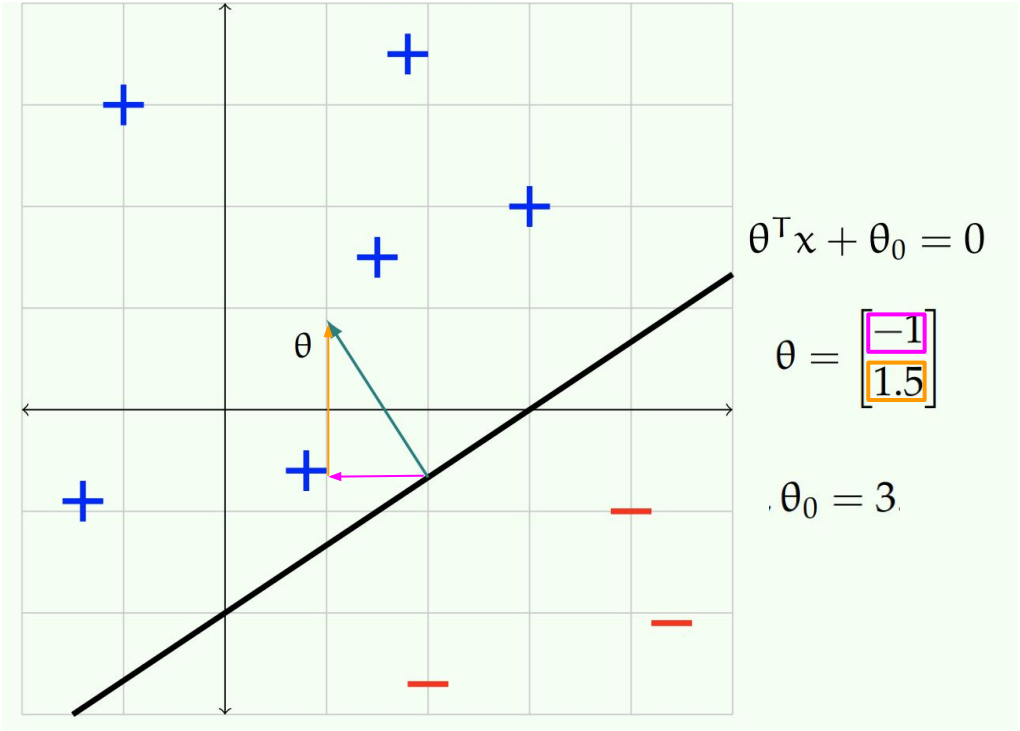


$$\theta^T x + \theta_0 = 0$$

$$\theta = \begin{bmatrix} -1 \\ 1.5 \end{bmatrix}$$

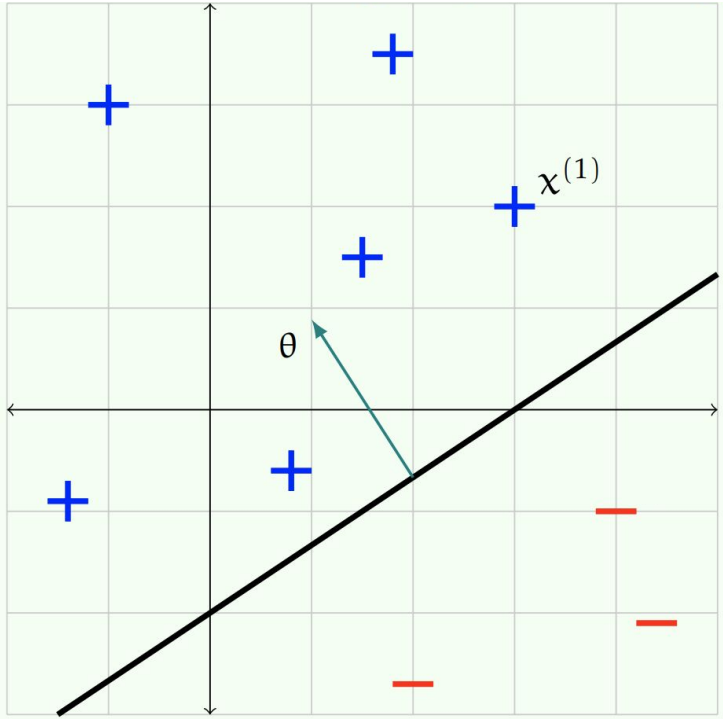
$$\theta_0 = 3$$

Linear Classifier Example



Direction along x1 axis
Direction along x2 axis

Linear Classifier Example



$$\theta^T x + \theta_0 = 0$$

$$\theta = \begin{bmatrix} -1 \\ 1.5 \end{bmatrix}$$

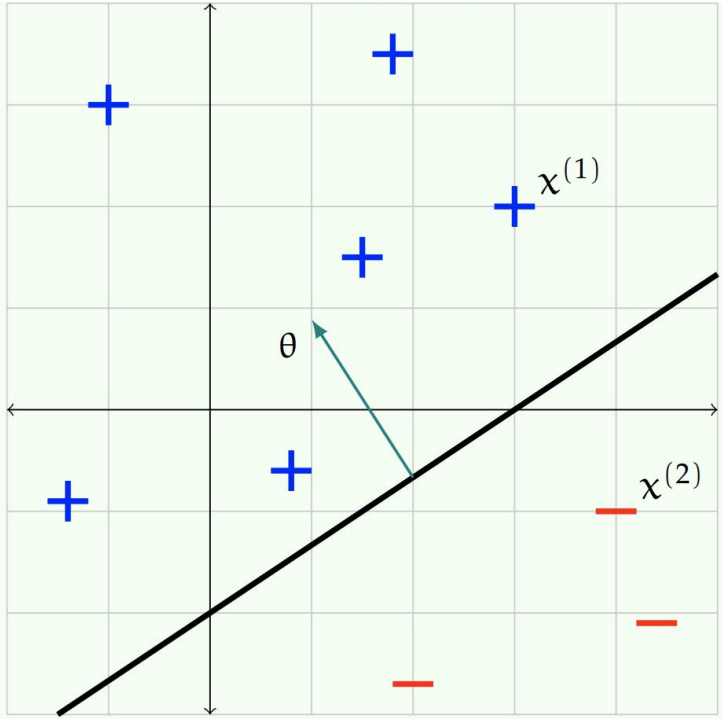
$$\theta_0 = 3$$

$$x^{(1)} = \begin{bmatrix} 3 \\ 2 \end{bmatrix}$$

$$h(x^{(1)}; \theta, \theta_0) = \text{sign}\left(\begin{bmatrix} -1 & 1.5 \end{bmatrix} \begin{bmatrix} 3 \\ 2 \end{bmatrix} + 3 \right)$$

$$= \text{sign}(3) = +1$$

Linear Classifier Example



$$\theta^T x + \theta_0 = 0$$

$$\theta = \begin{bmatrix} -1 \\ 1.5 \end{bmatrix}$$

$$\theta_0 = 3$$

$$x^{(1)} = \begin{bmatrix} 3 \\ 2 \end{bmatrix}$$

$$h(x^{(1)}; \theta, \theta_0) = \text{sign}\left(\begin{bmatrix} -1 & 1.5 \end{bmatrix} \begin{bmatrix} 3 \\ 2 \end{bmatrix} + 3 \right)$$

$$= \text{sign}(3) = +1$$

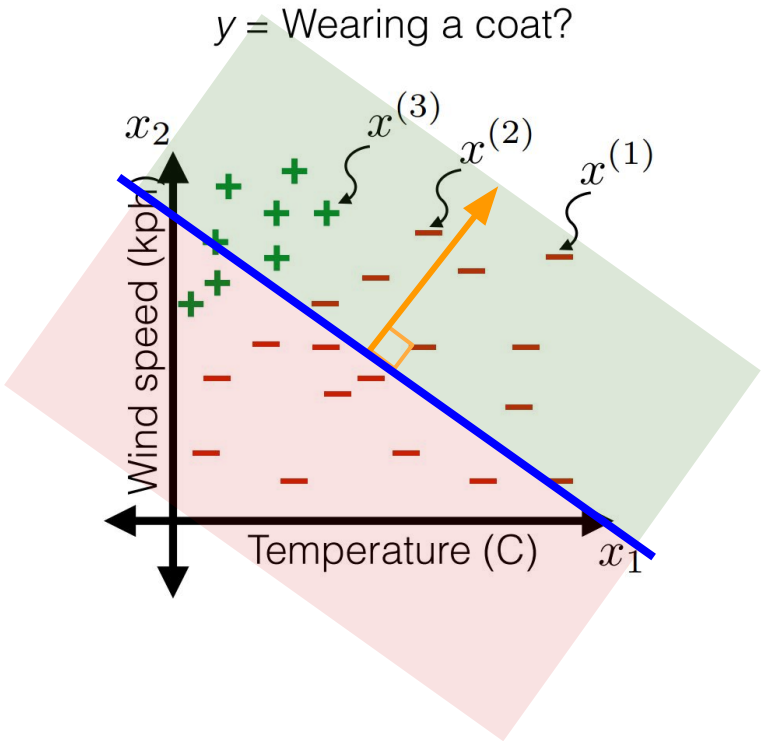
$$x^{(2)} = \begin{bmatrix} 4 \\ -1 \end{bmatrix}$$

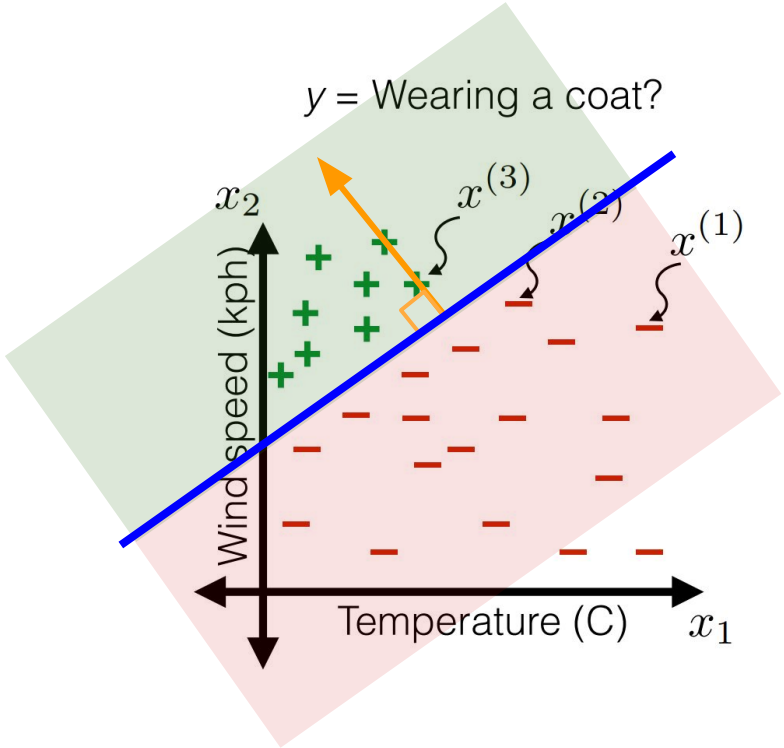
$$h(x^{(2)}; \theta, \theta_0) = \text{sign}\left(\begin{bmatrix} -1 & 1.5 \end{bmatrix} \begin{bmatrix} 4 \\ -1 \end{bmatrix} + 3 \right)$$

$$= \text{sign}(-2.5) = -1$$

Classifier Performance

- Which classifier is better?





Classifier Performance

- Which classifier is better?
- “Zero-one” loss:

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

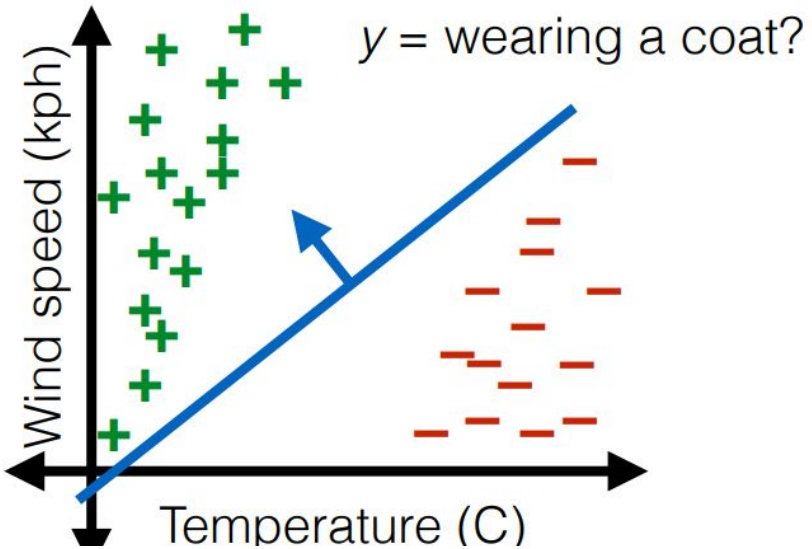
g: guess,
a: actual

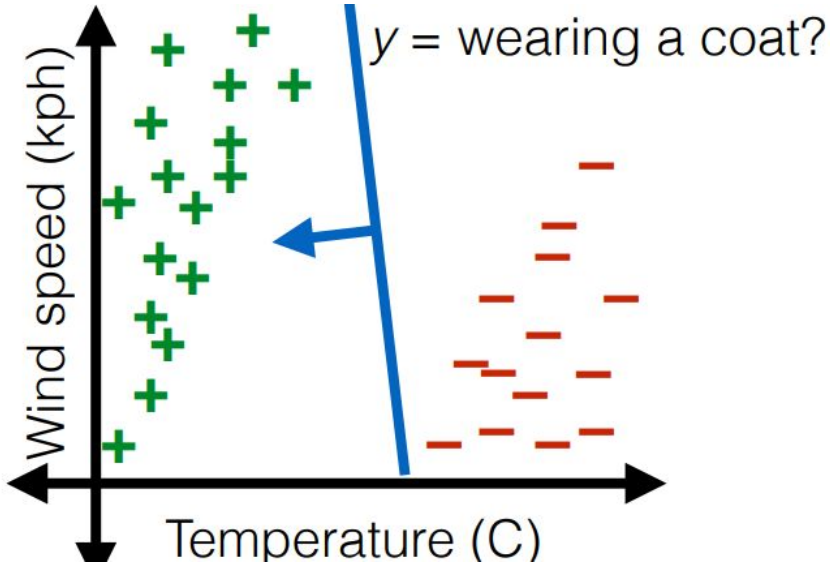
- Dataset level error: 1 - accuracy

$$\frac{1}{n} \sum_{i=1}^{n} L(g^{(i)}, y^{(i)})$$

Linear Classifier Issues

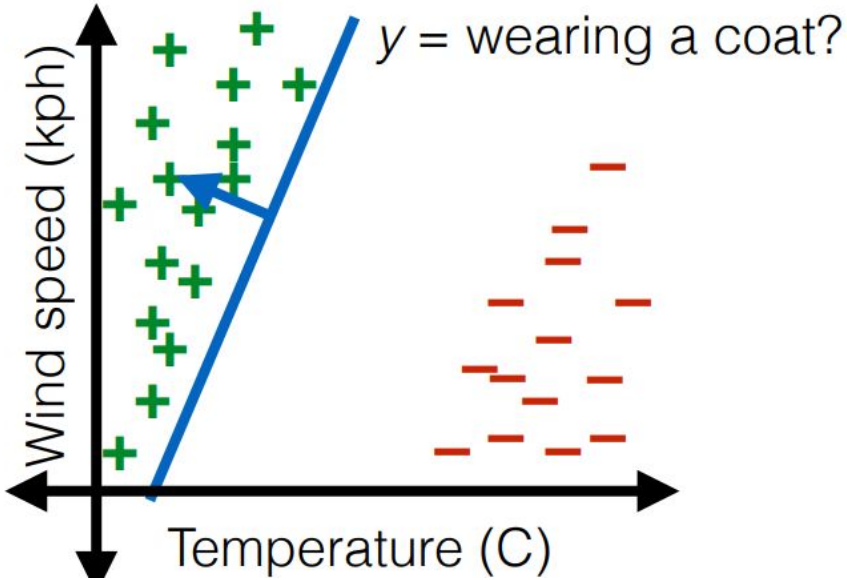
- Issue 1: No notion of uncertainty

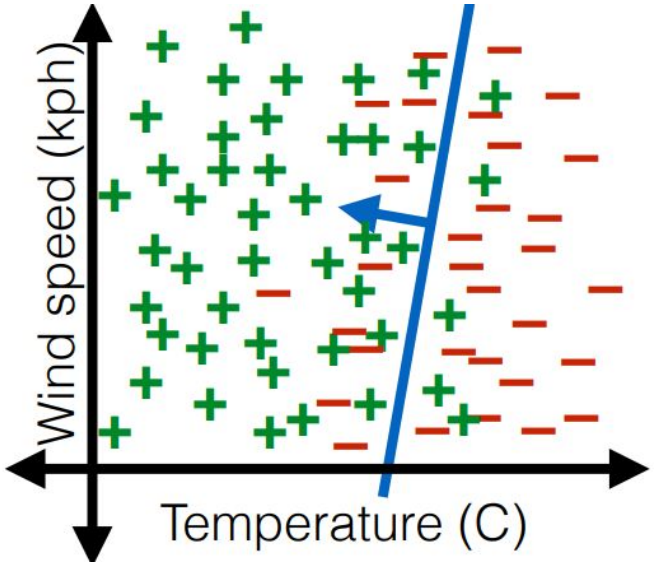




Linear Classifier Issues

- Issue 1: No notion of uncertainty





Linear Classifier Issues

- Issue 2: Gradient descent not possible

$$J(\theta) = \frac{1}{n}\sum_{i=1}^{n} L(g^{(i)}, y^{(i)})$$

$$= \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}\left\{y^{(i)} \neq \text{sign}(\theta^\top x^{(i)} + \theta_0)\right\}$$

What is the gradient of the above with respect the model parameters?

Linear Classifier Issues

- Issue 2: Gradient descent not possible

$$J(\theta) = \frac{1}{n}\sum_{i=1}^{n} L(g^{(i)}, y^{(i)})$$

$$= \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}\left\{y^{(i)} \neq \text{sign}(\theta^\top x^{(i)} + \theta_0)\right\}$$

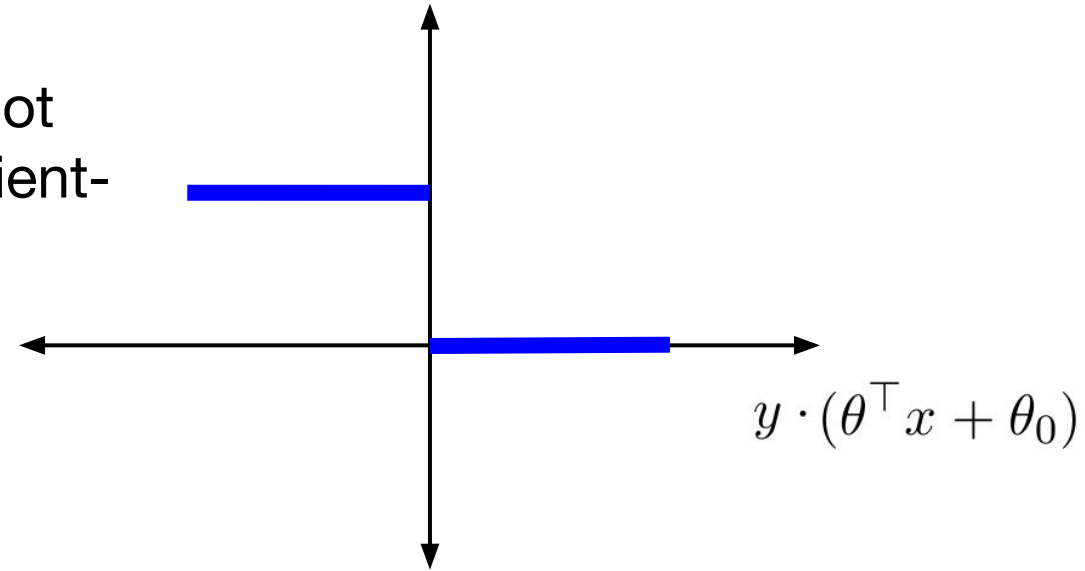
- Gradient is zero “almost everywhere”

$$\nabla_\theta J(\theta) = \frac{1}{n}\sum_{i=1}^{n} \nabla_\theta \mathbb{1}\left\{y^{(i)} \neq \text{sign}(\theta^\top x^{(i)} + \theta_0)\right\}$$

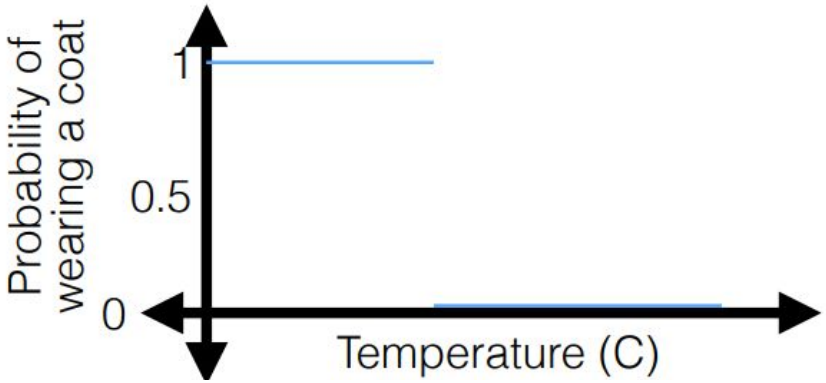
Linear Classifier Issues

$$\mathbb{1}\left\{y \neq \text{sign}(\theta^\top x + \theta_0)\right\}$$

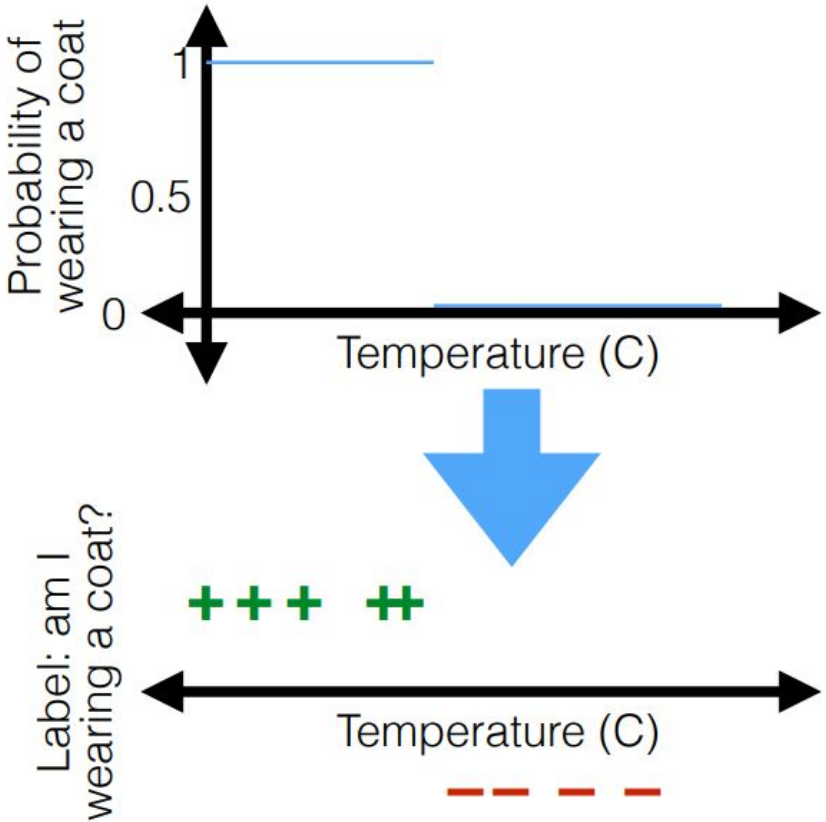
Zero-one loss is not amenable to gradient-based learning!



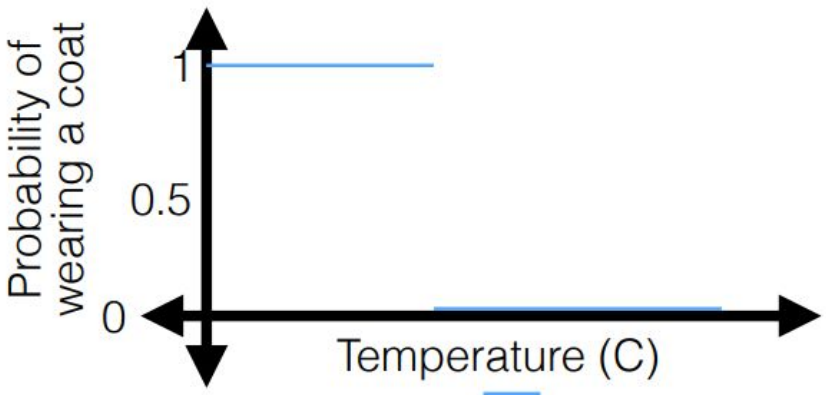
Capturing Uncertainty

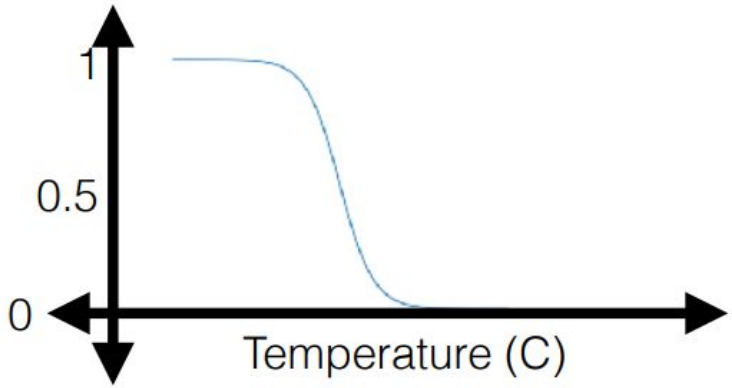


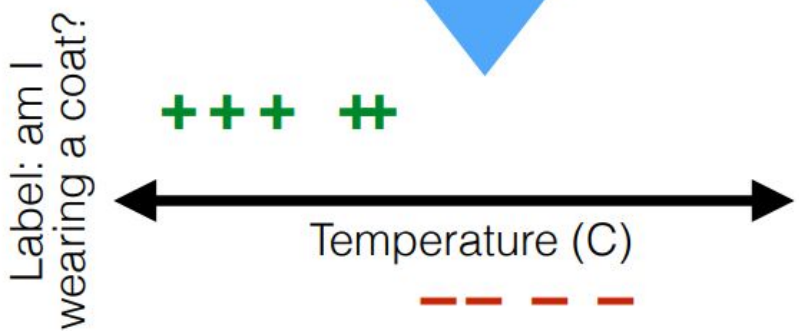
Capturing Uncertainty



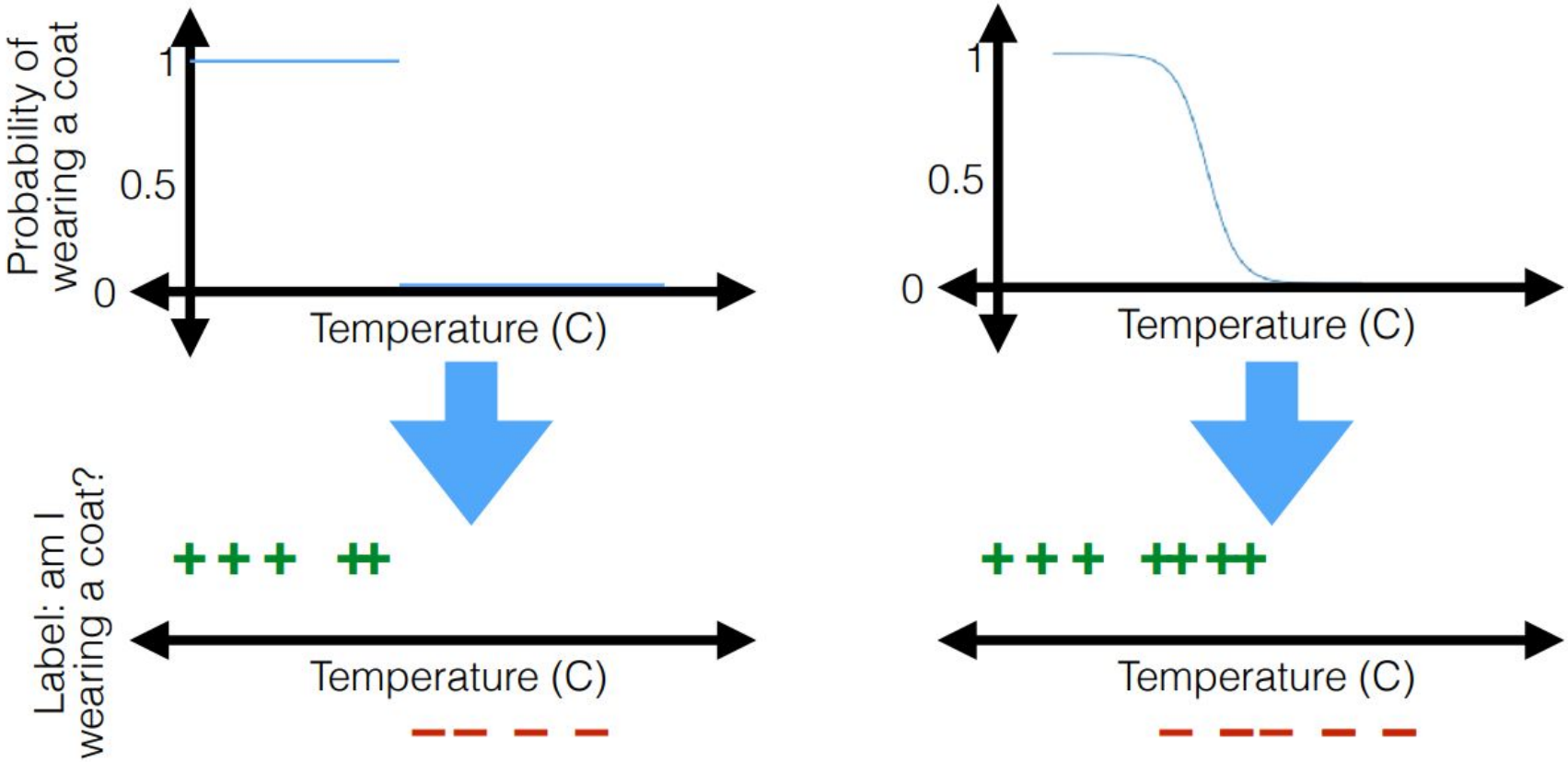
Capturing Uncertainty



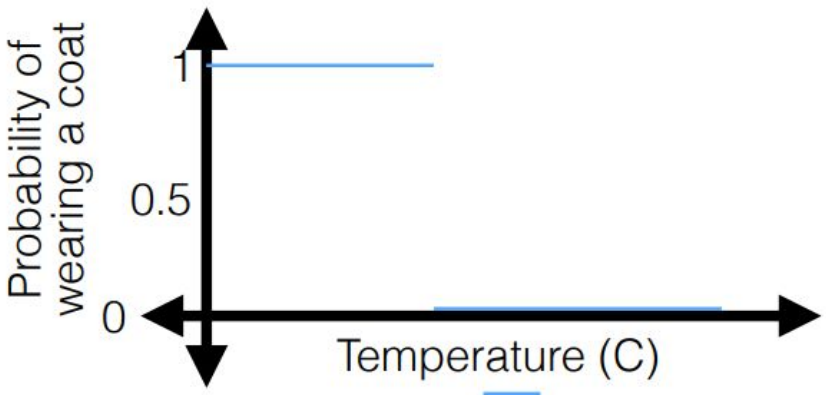


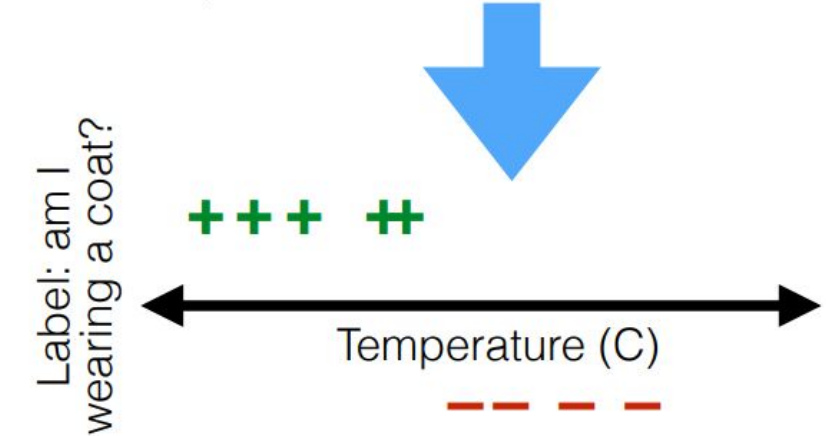


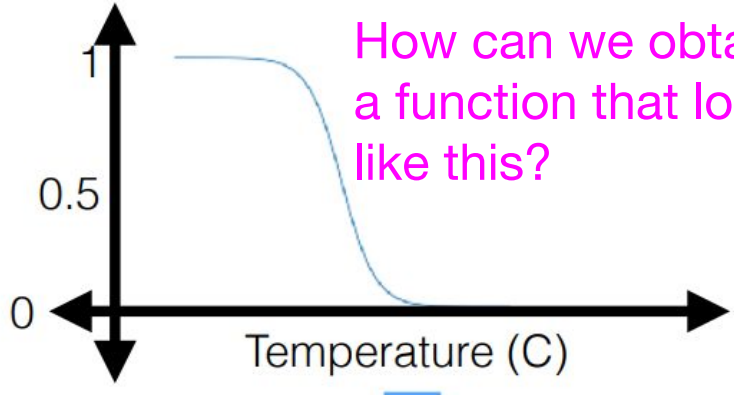
Capturing Uncertainty



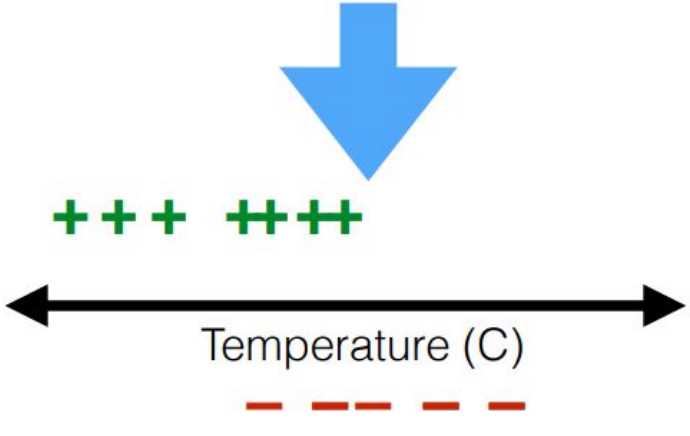
Capturing Uncertainty







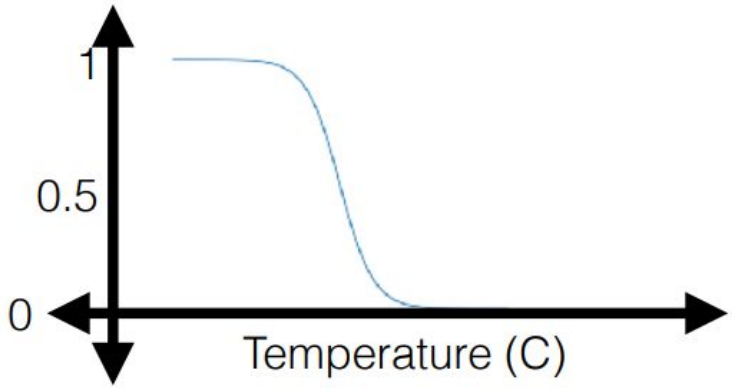
How can we obtain a function that looks like this?

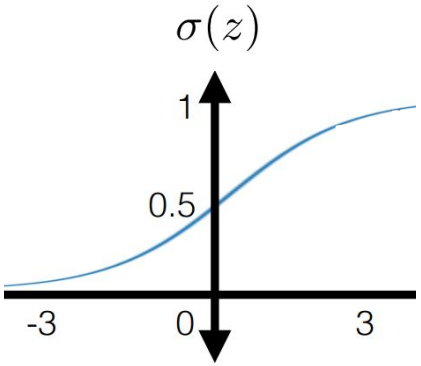


Capturing Uncertainty

Sigmoid Function $\sigma(z) : \mathbb{R} \rightarrow (0, 1)$

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

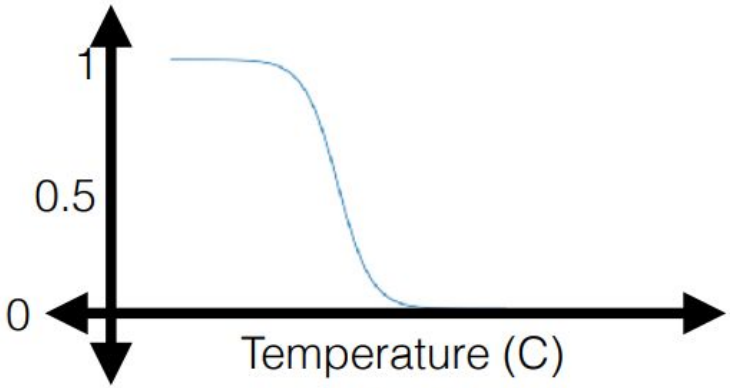


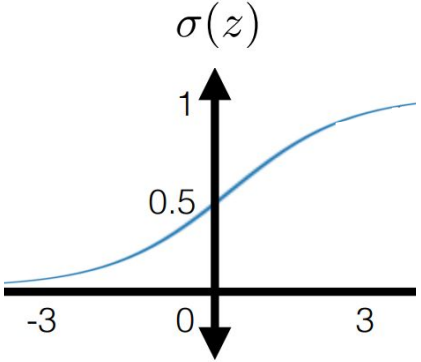


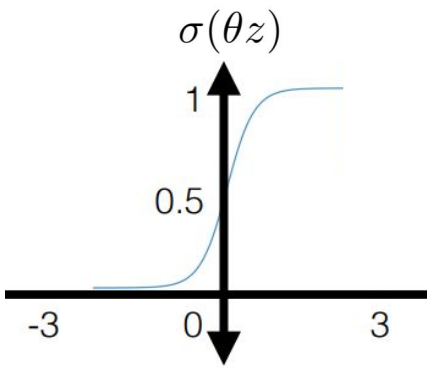
Capturing Uncertainty

Sigmoid Function $\sigma(z) : \mathbb{R} \to (0, 1)$

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



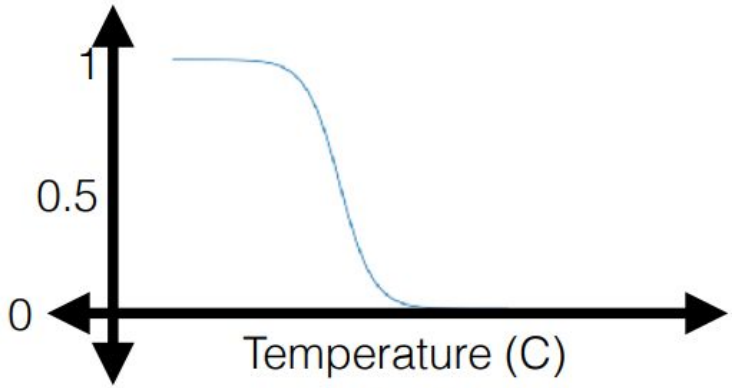


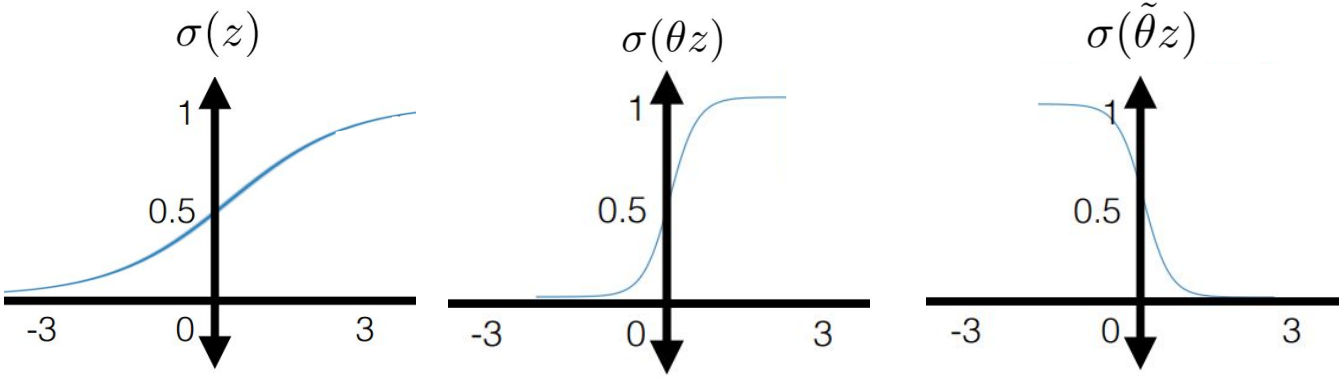


Capturing Uncertainty

Sigmoid Function $\sigma(z) : \mathbb{R} \rightarrow (0, 1)$

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

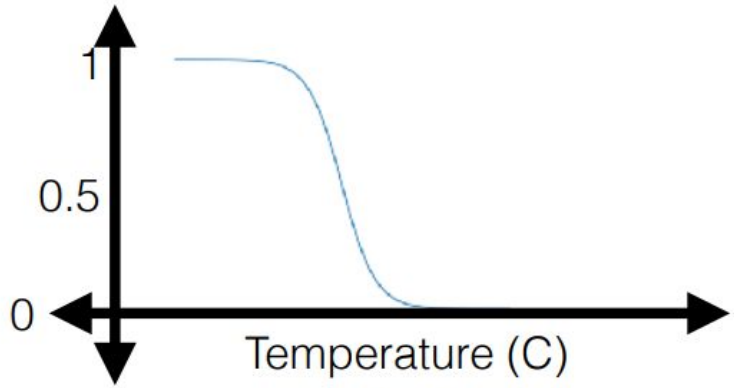


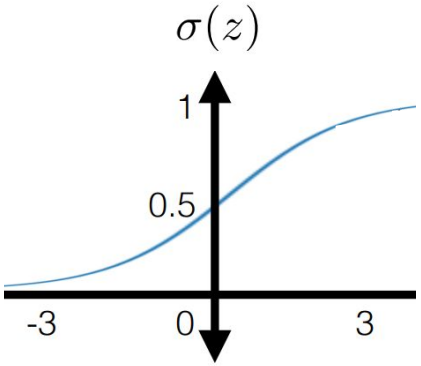


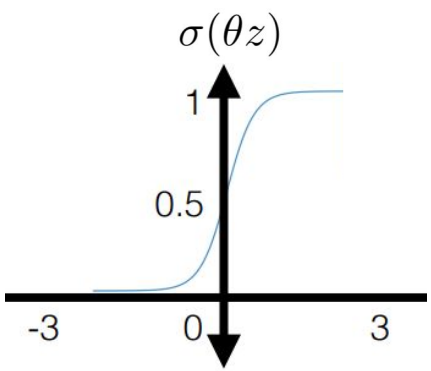
Capturing Uncertainty

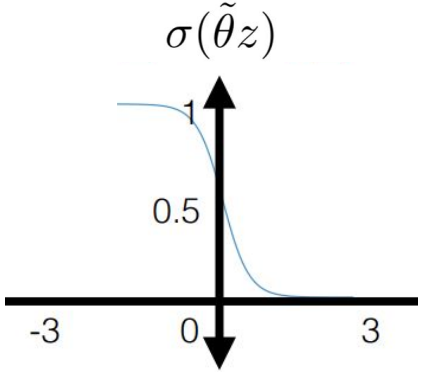
Sigmoid Function $\sigma(z) : \mathbb{R} \rightarrow (0, 1)$

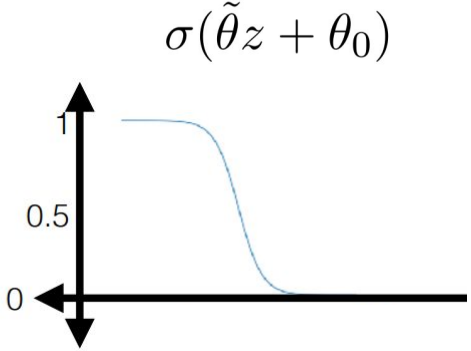
$$\sigma(z) = \frac{1}{1 + \exp(-z)}$$





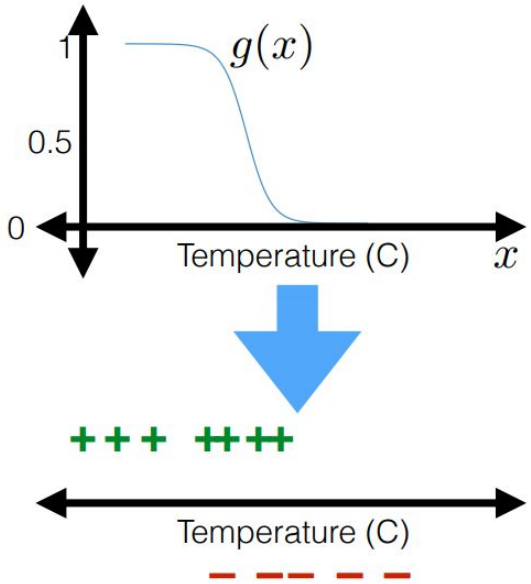


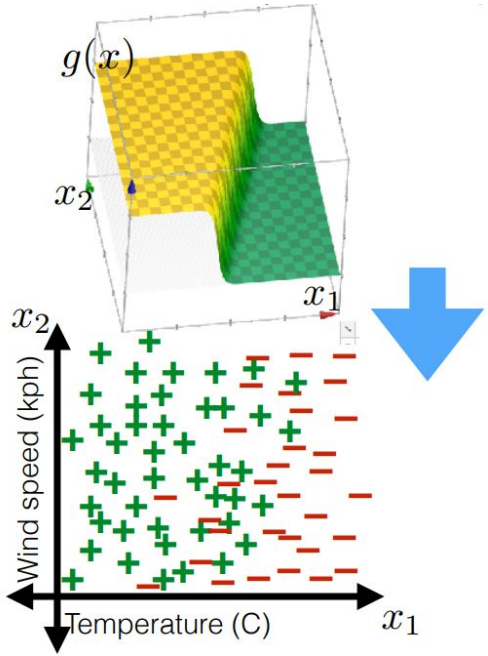




Capturing Uncertainty

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





Cost Function

- What (differentiable) cost function should we optimize?
- Mean squared error?

$$J(\theta) = \frac{1}{n} \sum_{i=1}^{n} \left(\sigma(\theta^\top x^{(i)} + \theta_0) - y^{(i)} \right)^2$$

$y \in \{0, 1\}$

- Will “work”, but loss is non-convex and somewhat unnatural.

Maximum Likelihood Estimation

- Idea: find parameters that would **maximize the probability of observed data.**

- Probability of a label

$$P(Y = +1 \mid X = x) = \sigma(\theta^\top x + \theta_0)$$

"Probability that the label is +1 given datum x"

Maximum Likelihood Estimation

- Idea: find parameters that would **maximize the probability of observed data.**

- Probability of a label

$$P(Y = +1 \mid X = x) = \sigma(\theta^\top x + \theta_0)$$

$$P(Y = -1 \mid X = x) = 1 - \sigma(\theta^\top x + \theta_0)$$

$$P(Y = y \mid X = x) = \sigma(\theta^\top x + \theta_0)^{\mathbb{1}\{y=+1\}} \cdot (1 - \sigma(\theta^\top x + \theta_0))^{\mathbb{1}\{y=-1\}}$$

Maximum Likelihood Estimation

- Idea: find parameters that would **maximize the probability of observed data.**

- Probability of a label

$$P(Y = +1 \mid X = x) = \sigma(\theta^\top x + \theta_0)$$

$$P(Y = -1 \mid X = x) = 1 - \sigma(\theta^\top x + \theta_0)$$

$$P(Y = y \mid X = x) = \sigma(\theta^\top x + \theta_0)^{\mathbb{1}\{y=+1\}} \cdot (1 - \sigma(\theta^\top x + \theta_0))^{\mathbb{1}\{y=-1\}}$$

- Probability of all data:

  "Likelihood" function

$$P(\text{Data}) = \prod_{i=1}^{n} P(Y = y^{(i)} \mid X = x^{(i)})$$

Maximum Likelihood Estimation

- Idea: find parameters that would **maximize the probability of observed data.**

- Maximum Likelihood Estimation

$$
\begin{aligned}
&\arg\max_{\theta} \; P(\text{Data}) \\
&= \arg\max_{\theta} \; \log P(\text{Data}) \\
&= \arg\min_{\theta} \; -\log P(\text{Data}) \\
&= \arg\min_{\theta} \; -\frac{1}{n} \log P(\text{Data})
\end{aligned}
$$

Average negative log likelihood

Maximum Likelihood Estimation

Cost function: average negative log likelihood

$$J(\theta) = -\frac{1}{n}\log P(\text{Data}) = -\frac{1}{n}\log \prod_{i-1}^{n} P(Y = y^{(i)} \mid X = x^{(i)})$$

$$= -\frac{1}{n}\sum_{i=1}^{n} \log P(Y = y^{(i)} \mid X = x^{(i)})$$

$$= -\frac{1}{n}\sum_{i=1}^{n} \log \left(\sigma(\theta^\top x^{(i)} + \theta_0)^{\mathbb{1}\{y^{(i)}=+1\}} \cdot (1 - \sigma(\theta^\top x^{(i)} + \theta_0))^{\mathbb{1}\{y^{(i)}=-1\}} \right.$$

$$= -\frac{1}{n}\sum_{i=1}^{n} \mathbb{1}\left\{y^{(i)} = +1\right\} \log \sigma(\theta^\top x^{(i)} + \theta_0) +$$

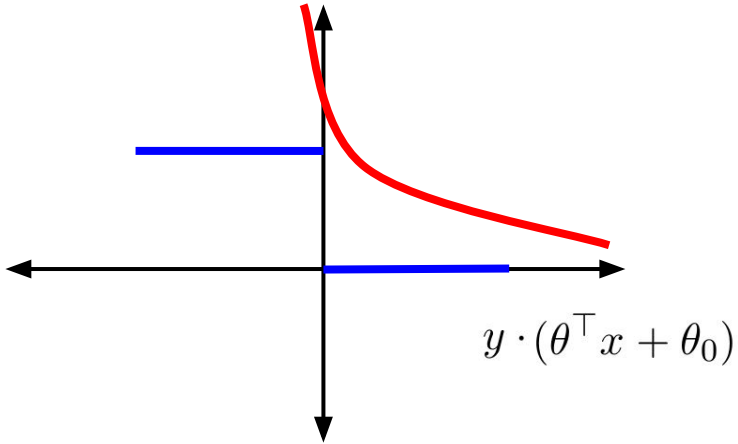
$$\mathbb{1}\left\{y^{(i)} = -1\right\} \log(1 - \sigma(\theta^\top x^{(i)} + \theta_0))$$

Differentiable (and convex) with respect to model parameters!

Logistic Regression

- Logistic Regression
  - Probability of label given by sigmoid function
  - Trained to maximize likelihood

Uncertainty-awareness + differentiable loss



Zero-one loss

Logistic regression loss

Logistic Regression Gradients

Cost function: average negative log likelihood

$y \in \{-1, +1\}$

$$J(\theta) = -\frac{1}{n}\sum_{i=1}^{n} \mathbb{1}\left\{y^{(i)} = +1\right\} \log \sigma(\theta^\top x^{(i)} + \theta_0) + \mathbb{1}\left\{y^{(i)} = -1\right\} \log(1 - \sigma(\theta^\top x^{(i)} + \theta_0))$$

$y \in \{0, 1\}$

$$J(\theta) = -\frac{1}{n}\sum_{i=1}^{n} y^{(i)} \log \sigma(\theta^\top x^{(i)} + \theta_0) + (1 - y^{(i)}) \log(1 - \sigma(\theta^\top x^{(i)} + \theta_0))$$

Use this formulation from now on

Logistic Regression Gradients

$$y^{(i)} \in \{0, 1\} \qquad g^{(i)} = \frac{1}{1 + \exp(-(\theta^\top x^{(i)} + \theta_0))}$$

$$J(\theta) = -\frac{1}{n} \sum_{i=1}^{n} y^{(i)} \log g^{(i)} + (1 - y^{(i)}) \log(1 - g^{(i)})$$

$$\nabla_\theta J(\theta) = -\frac{1}{n} \sum_{i=1}^{n} (y^{(i)} - g^{(i)}) x^{(i)}$$

Try to derive above and also find $\nabla_{\theta_0} J(\theta)$

Logistic vs. Linear Regression Gradients

Logistic Regression

$$J(\theta) = -\frac{1}{n}\sum_{i=1}^{n} y^{(i)} \log g^{(i)} + (1 - y^{(i)}) \log(1 - g^{(i)})$$

$$\nabla_\theta J(\theta) = -\frac{1}{n}\sum_{i=1}^{n}(y^{(i)} - g^{(i)})x^{(i)} = -\frac{1}{n}\sum_{i=1}^{n}\left(y^{(i)} - \sigma(\theta^\top x^{(i)} + \theta_0)\right)x^{(i)}$$

Linear Regression

$$J(\theta) = \frac{1}{n}\sum_{i=1}^{n}(y^{(i)} - g^{(i)})^2$$

$$\nabla_\theta J(\theta) = -\frac{2}{n}\sum_{i=1}^{n}(y^{(i)} - g^{(i)})x^{(i)} = -\frac{2}{n}\sum_{i=1}^{n}\left(y^{(i)} - (\theta^\top x^{(i)} + \theta_0)\right)x^{(i)}$$

Logistic vs. Linear Regression Gradients

Logistic Regression

$$J(\theta) = -\frac{1}{n}\sum_{i=1}^{n} y^{(i)} \log g^{(i)} + (1 - y^{(i)}) \log(1 - g^{(i)})$$

$$\nabla_\theta J(\theta) = -\frac{1}{n}\sum_{i=1}^{n}(y^{(i)} - g^{(i)})x^{(i)} = -\frac{1}{n}\sum_{i=1}^{n}\left(y^{(i)} - \sigma(\theta^\top x^{(i)} + \theta_0)\right) x^{(i)}$$

Linear Regression

$$J(\theta) = \frac{1}{n}\sum_{i=1}^{n}(y^{(i)} - g^{(i)})^2$$

$$\nabla_\theta J(\theta) = -\frac{2}{n}\sum_{i=1}^{n}(y^{(i)} - g^{(i)})x^{(i)} = -\frac{2}{n}\sum_{i=1}^{n}\left(y^{(i)} - (\theta^\top x^{(i)} + \theta_0)\right) x^{(i)}$$

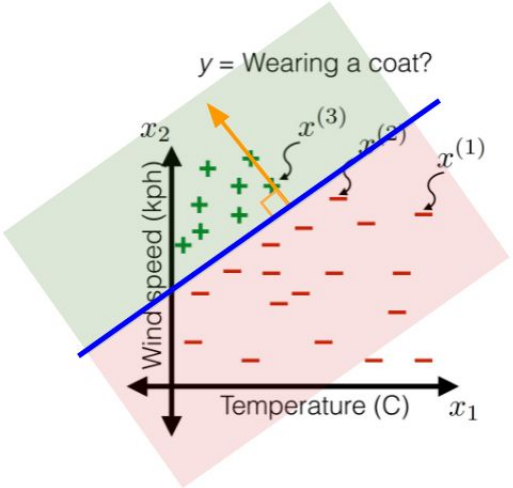
Intuition: Adjust model parameters to point in the same direction as datum, weighted by how "wrong" the current model is (residual)

Logistic Regression Evaluation

Classification rule:

$$h(x) = \mathbb{1}\{\sigma(\theta^\top x + \theta_0) > 0.5\} = \mathbb{1}\{\theta^\top x + \theta_0 > 0\}$$

⇒ Still a linear classifier! Logistic regression is just one way of learning a hyperplane



Evaluation:

- Accuracy on test set
- Negative log likelihood of test set