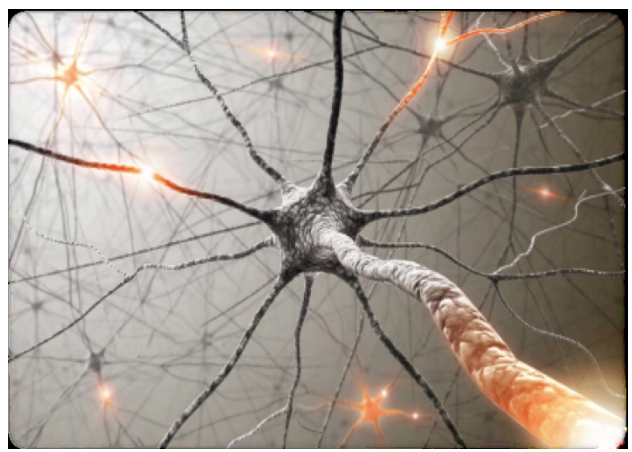
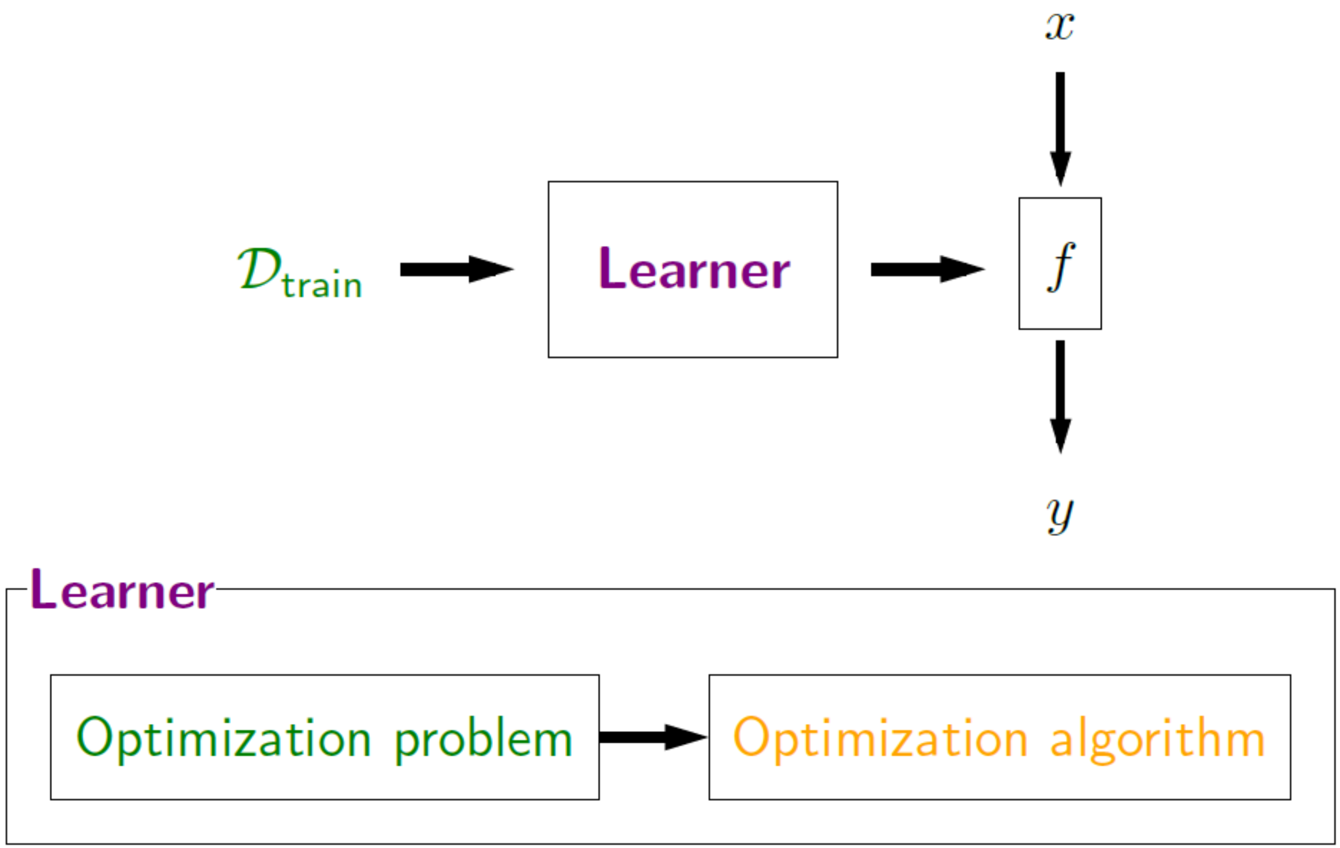
02 Linear Predictor



Question

- Can we obtain decision boundaries which are circles by using linear classifiers?
  - Yes
  - No

Framework



x
$\mathcal{D}_{\text{train}}$
Learner
f
y
Learner
Optimization problem
Optimization algorithm

Review: optimization problem

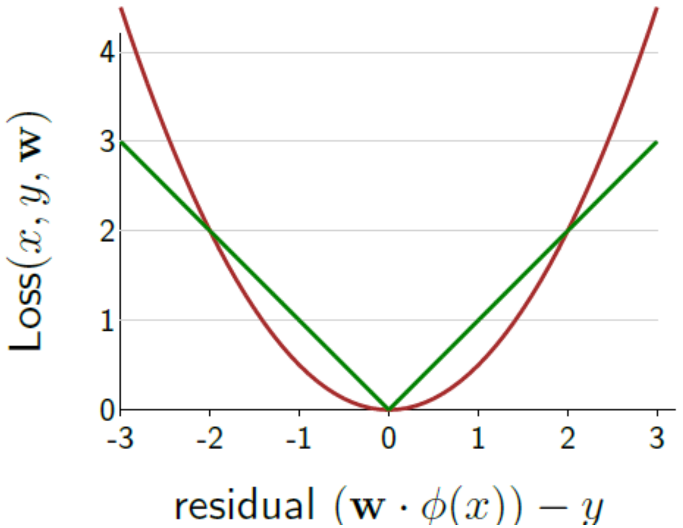
Key idea: minimize training loss

$$\text{TrainLoss}(\mathbf{w}) = \frac{1}{|\mathcal{D}_{\text{train}}|} \sum_{(x,y)\in\mathcal{D}_{\text{train}}} \text{Loss}(x, y, \mathbf{w})$$

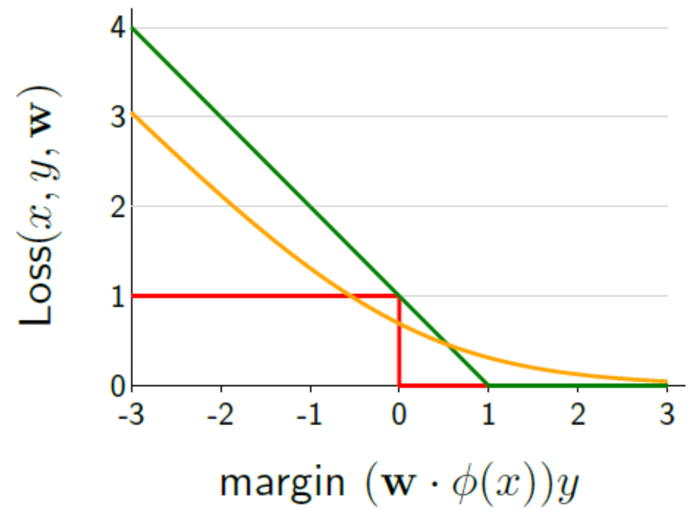
$$\min_{\mathbf{w}\in\mathbb{R}^d} \text{TrainLoss}(\mathbf{w})$$

Review: loss functions

Regression

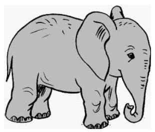


Binary classification



Captures properties of the desired predictor

Review: optimization algorithms

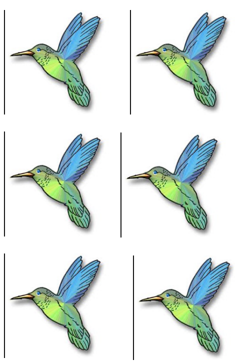


Algorithm: gradient descent

Initialize $\mathbf{w} = [0, \ldots, 0]$

For $t = 1, \ldots, T$:

$$\mathbf{w} \leftarrow \mathbf{w} - \underbrace{\eta}_{\text{step size}} \underbrace{\nabla_{\mathbf{w}} \text{TrainLoss}(\mathbf{w})}_{\text{gradient}}$$



Algorithm: stochastic gradient descent

Initialize $\mathbf{w} = [0, \ldots, 0]$

For $t = 1, \ldots, T$:

For $(x, y) \in \mathcal{D}_{\text{train}}$:

$$\mathbf{w} \leftarrow \mathbf{w} - \eta_t \nabla_{\mathbf{w}} \text{Loss}(x, y, \mathbf{w})$$

Two components

Score (drives prediction):

$$\mathbf{w} \cdot \phi(x)$$

- Previous: learning sets w via optimization
- Next: feature extraction species $\phi(x)$ based on domain knowledge

Organization of features

Task: predict whether a string is an email address

abc@gmail.com

feature extractor

arbitrary!

Length > 10	:1
fracOfAlpha	:0.85
contains_@	: 1
endsWith_com	: 1
endsWith_org	: 0

Which features to include? Need an organizational principle...

Feature templates

Definition: feature template (informal)

A feature template is a group of features all computed in a similar way.

Input: abc@gmail.com

Some feature templates:

- Length greater than ___
- Last three character equals___
- Contains character___
- Pixel intensity of position___, ___

Feature templates

Feature template: last three characters equals___

abc@gmail.com

endsWith_aaa: 0
endsWith_aab: 0
endsWith_aac: 0
…
endsWith_com: 1
…
endsWith_ZZZ: 0

Sparsity in feature vectors

Feature template: last character equals___

abc@gmail.com →

endsWith a : 0
endsWith b : 0
endsWith c : 0
endsWith d : 0
endsWith e : 0
endsWith f : 0
endsWith g : 0
endsWith h : 0
endsWith i : 0
endsWith j : 0
endsWith k : 0
endsWith l : 0
endsWith m: 1
endsWith n : 0
endsWith o : 0
endsWith p : 0
endsWith q : 0
endsWith r : 0
endsWith s : 0
endsWith t : 0
endsWith u : 0
endsWith v : 0
endsWith w : 0
endsWith x : 0
endsWith y : 0
endsWith z : 0

Feature vector representations

fracOfAlpha :0.85
contains_a : 0
…
endsWith_@ : 1
…

Array representation (good for dense features):

[0.85, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0]

Map representation (good for sparse features):

{"fracOfAlpha": 0.85, "contains @": 1}

Hypothesis class

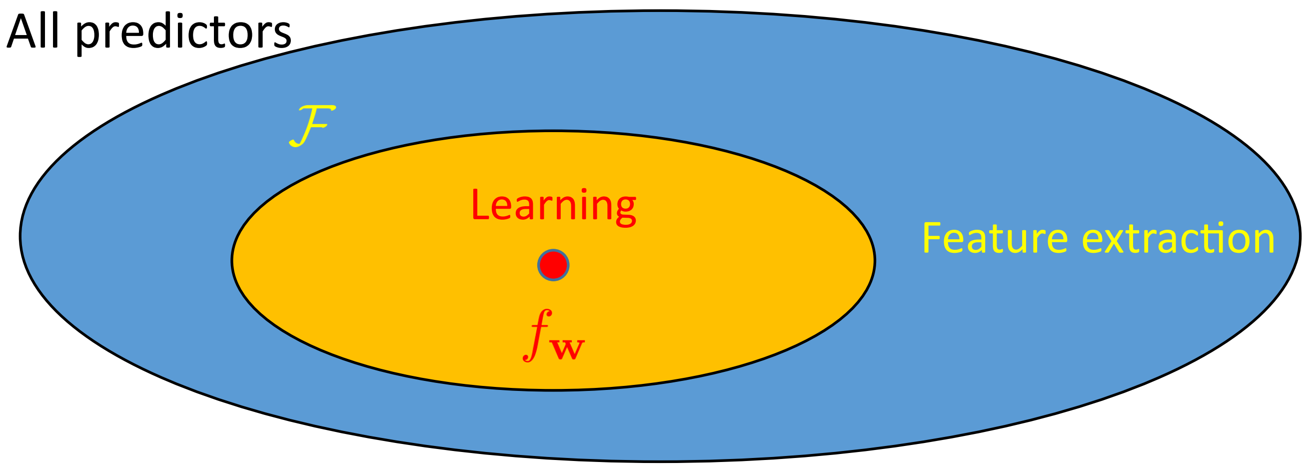
Predictor: $f_{\mathbf{w}}(x) = \mathbf{w} \cdot \phi(x)$ or $\text{sign}(\mathbf{w} \cdot \phi(x))$

Definition: hypothesis class

A **hypothesis class** is the set of possible predictors with a fixed $\phi(x)$ and varying w:

$$\mathcal{F} = \{f_{\mathbf{w}} : \mathbf{w} \in \mathbb{R}^d\}$$

Feature extraction + learning



- Feature extraction: set $\mathcal{F}$ based on domain knowledge
- Learning: set $f_{\mathbf{w}} \in \mathcal{F}$ based on data

Linear regression

training data

x	y
1	1
2	3
4	3

learning algorithm $\longrightarrow$ f **predictor**

$3 \rightarrow f \rightarrow 2.71$

Which predictors are possible?

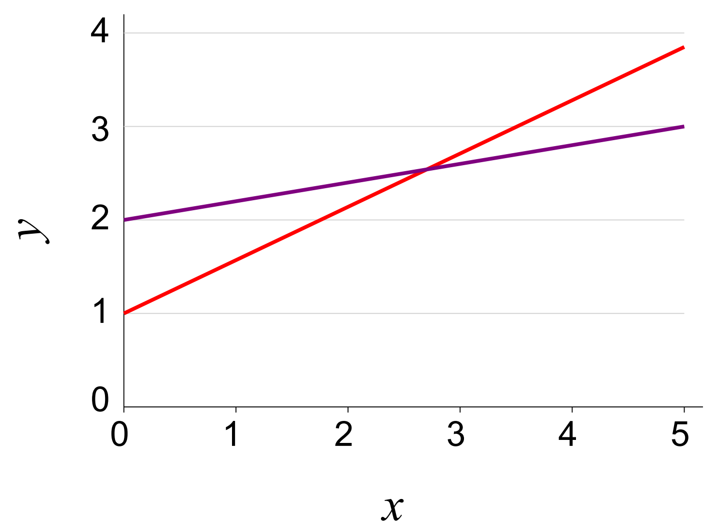
Hypothesis class

$$F = \{f_{\mathbf{w}}(x) = \mathbf{w} \cdot \varphi(x) : \mathbf{w} \in \mathbb{R}^d\}$$

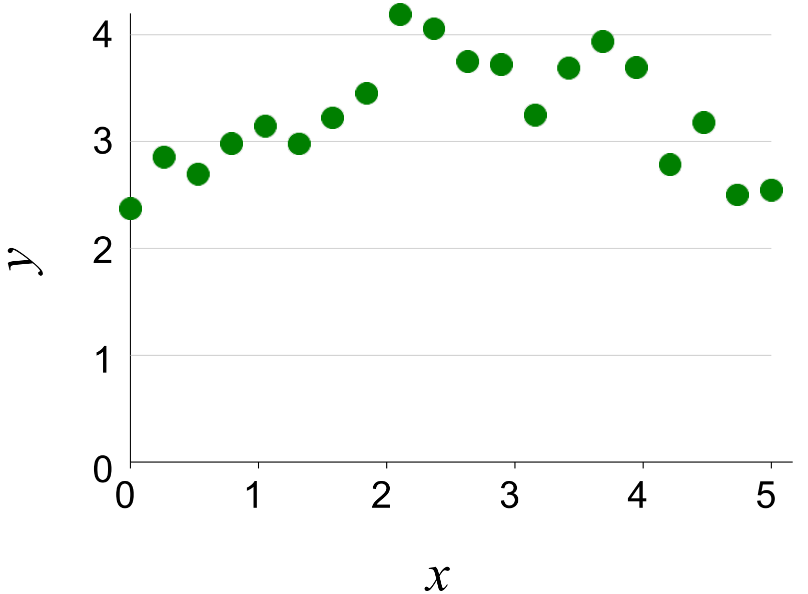
$\varphi(x) = [1, x]$

$f(x) = [1, 0.57] \cdot \varphi(x)$

$f(x) = [2, 0.2] \cdot \varphi(x)$



More complex data



How do we fit a non-linear predictor?

Example: beyond linear functions

Regression: $x \in \mathbb{R}, y \in \mathbb{R}$

Linear functions: $\phi(x) = x$

$$\mathcal{F} = \{x \mapsto w_1 x \quad : w_1 \in \mathbb{R}, \quad \}$$

Quadratic functions:

$$\phi(x) = [x, x^2]$$

$$\mathcal{F} = \{x \mapsto w_1 x + w_2 x^2 : w_1 \in \mathbb{R}, w_2 \in \mathbb{R}\}$$

[blackboard]

Quadratic predictors

$$\varphi(x) = [1, x, x^2]$$

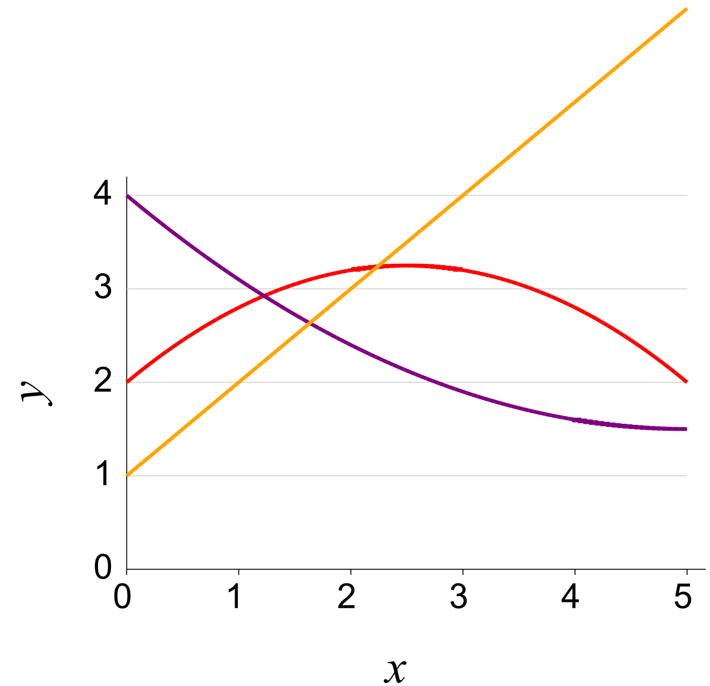
Example: $\varphi(3) = [1, 3, 9]$

$f(x) = [2, 1, -0.2] \cdot \varphi(x)$

$f(x) = [4, -1, 0.1] \cdot \varphi(x)$

$f(x) = [1, 1, 0] \cdot \varphi(x)$

$F = \{f_{\mathbf{w}}(x) = \mathbf{w} \cdot \varphi(x) : \mathbf{w} \in \mathbb{R}^3\}$



Non-linear predictors just by changing φ

Piecewise constant predictors

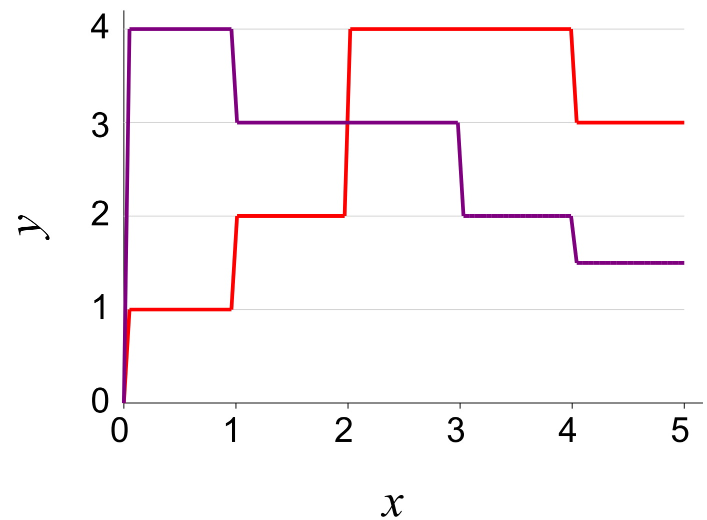
$$\varphi(x) = [\mathbf{1}[0 < x \leq 1], \mathbf{1}[1 < x \leq 2], \mathbf{1}[2 < x \leq 3], \mathbf{1}[3 < x \leq 4], \mathbf{1}[4 < x \leq 5]]$$

Example: $\varphi(2.3) = [0, 0, 1, 0, 0]$

$$f(x) = [1, 2, 4, 4, 3] \cdot \varphi(x)$$

$$f(x) = [4, 3, 3, 2, 1.5] \cdot \varphi(x)$$

$$F = \{f_{\mathbf{w}}(x) = \mathbf{w} \cdot \varphi(x) : \mathbf{w} \in \mathbb{R}^5\}$$



Expressive non-linear predictors by partitioning the input space

Predictors with periodicity structure

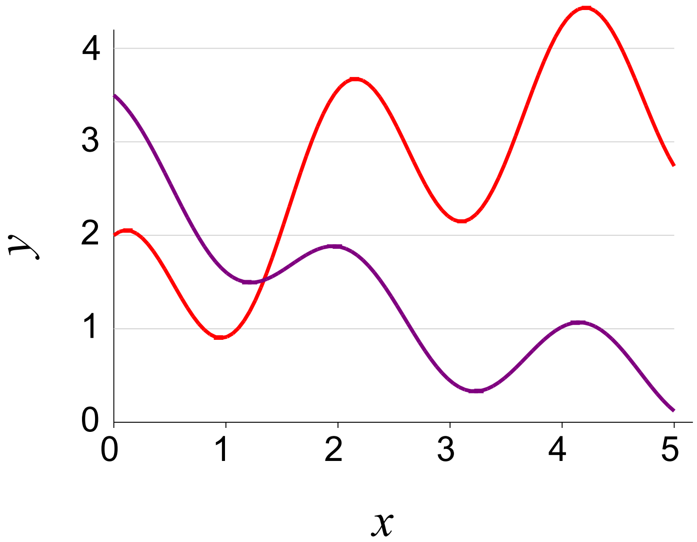
$$\varphi(x) = [1, x, x^2, \cos(3x)]$$

Example: $\varphi(2) = [1, 2, 4, 0.96]$

$$f(x) = [1, 1, -0.1, 1] \cdot \varphi(x)$$

$$f(x) = [3, -1, 0.1, 0.5] \cdot \varphi(x)$$

$$F = \{f_{\mathbf{w}}(x) = \mathbf{w} \cdot \varphi(x) : \mathbf{w} \in \mathbb{R}^4\}$$



Just throw in any features you want

Linear in what?

Prediction driven by score:

$$\mathbf{w} \cdot \phi(x)$$

Linear in w?	Yes
Linear in $\phi(x)$	Yes
Linear in x?	No! (x not necessarily even a vector)

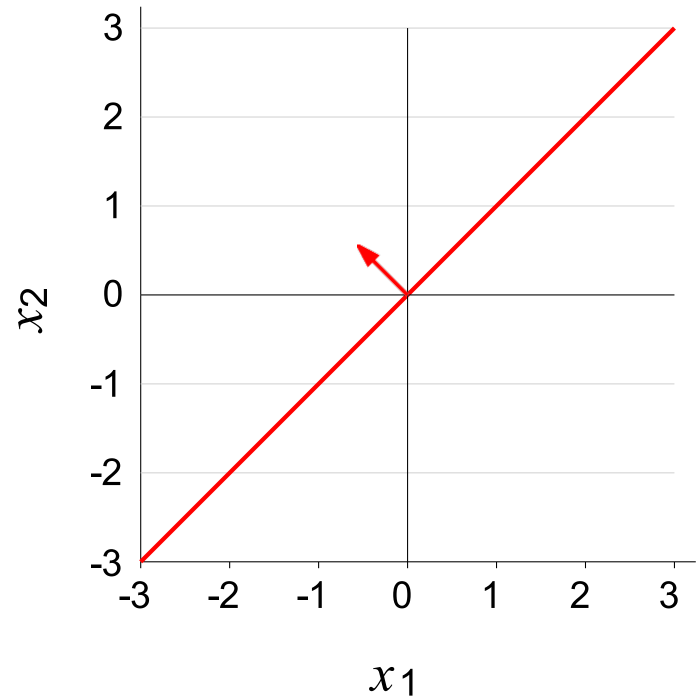
Key idea: minimize training loss

- Predictors $f_{\mathbf{w}}(x)$ can be expressive non-linear functions and decision boundaries of x.
- Score $\mathbf{w} \cdot \phi(x)$ is linear function of w, which permits efficient learning.

Linear classification

$\varphi(x) = [x_1, x_2]$

$f(x) = \text{sign}([-0.6, 0.6] \cdot \varphi(x))$



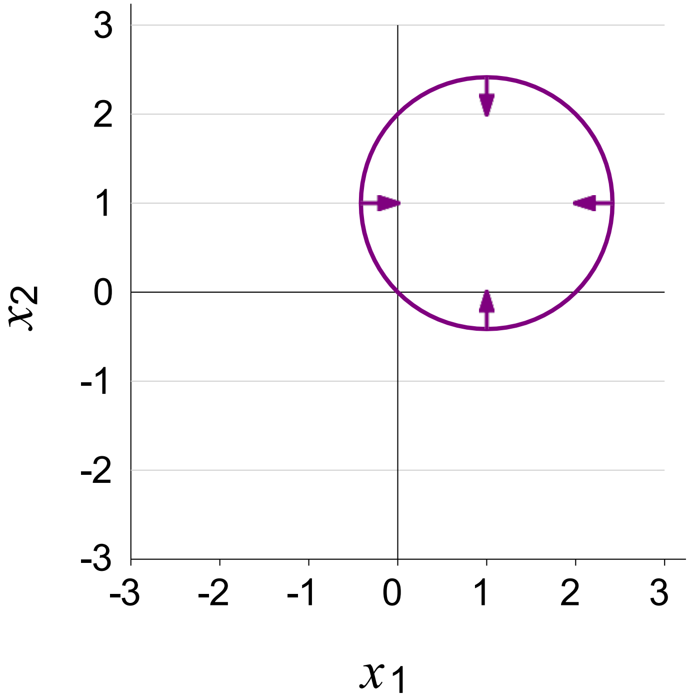
Decision boundary is a line

Quadratic classifiers

$$\phi(x) = [x_1, x_2, x_1^2 + x_2^2]$$
$$f(x) = \text{sign}([2, 2, -1] \cdot \phi(x))$$

Equivalently:

$$f(x) = \begin{cases} 1 & \text{if } \{(x\_1 - 1)^2 + (x\_2 - 1)^2 \le 2\} \\ -1 & \text{otherwise} \end{cases}$$

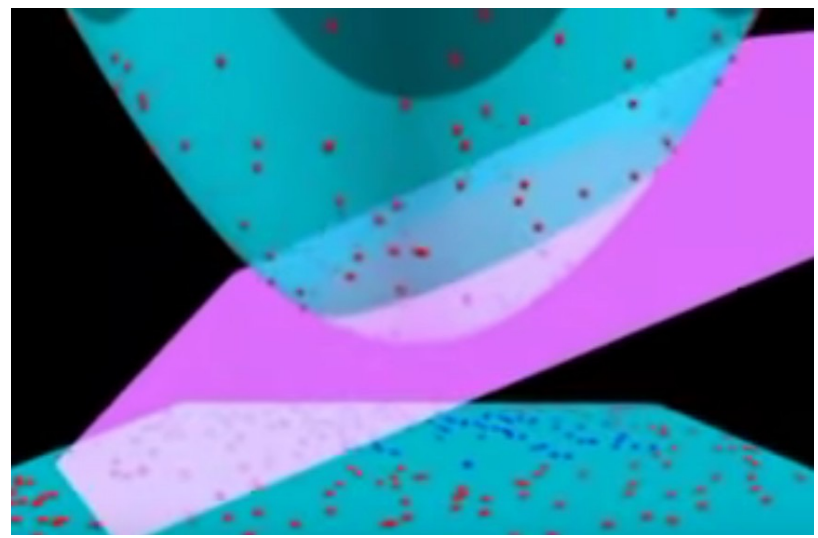


Decision boundary is a circle

Visualization in feature space

Input space: $x = [x_1, x_2]$, decision boundary is a circle

Feature space: $\phi(x) = [x_1, x_2, x_1^2 + x_2^2]$, decision boundary is a hyperplane

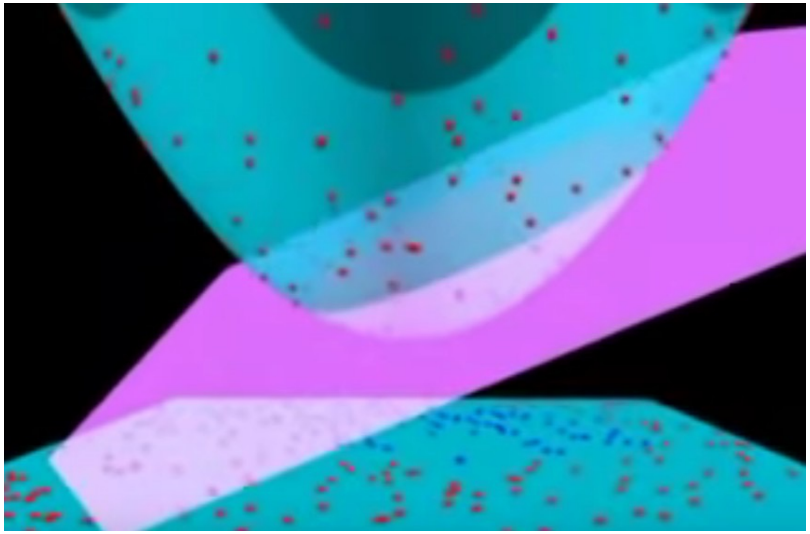


Summary

$$f_{\mathbf{w}}(x) = \mathbf{w} \cdot \varphi(x)$$

linear in $\mathbf{w}$, $\varphi(x)$

non-linear in x



- Regression: non-linear predictor, classification: non-linear decision boundary
- Types of non-linear features: quadratic, piecewise constant, etc.

Non-linear predictors with linear machinery

Summary so far

- Feature templates: organize related (sparse) features
- Hypothesis class: defined by features (what is possible)
- Linear classifiers: can produce non-linear decision boundaries