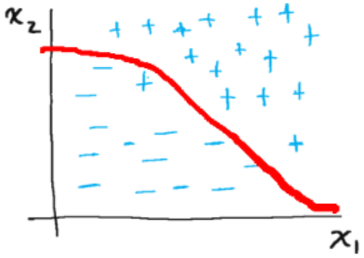
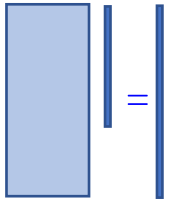
CS/ECE/ME 532
Activity 8 Complexity, Overfitting, and CV



$$\mathrm{sign}(\boldsymbol{x}_1^T \boldsymbol{w}) = -1$$
$$\mathrm{sign}(\boldsymbol{x}_2^T \boldsymbol{w}) = +1$$
$$\vdots$$

Drop ”sign”

$$\boldsymbol{X}\boldsymbol{w} \approx \boldsymbol{y}$$



Lots of training data

$$\boldsymbol{w}^* = \arg\min_{\boldsymbol{w}} ||\boldsymbol{X}\boldsymbol{w} - \boldsymbol{y}||_2^2$$

$$\boldsymbol{w}^* = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{y}$$

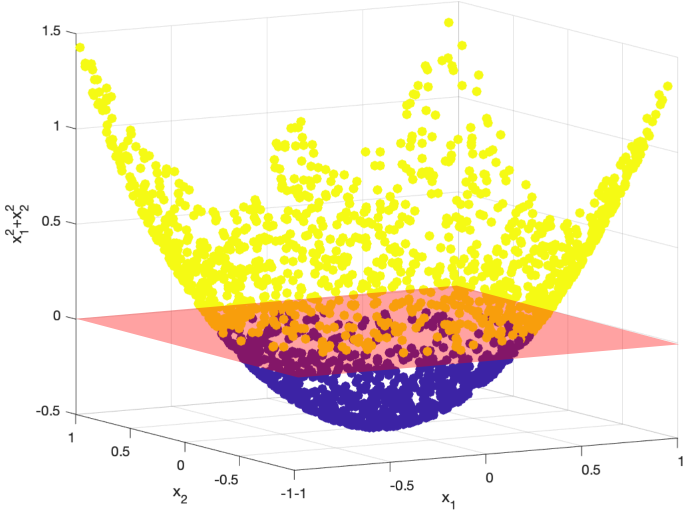
Squared error Loss function

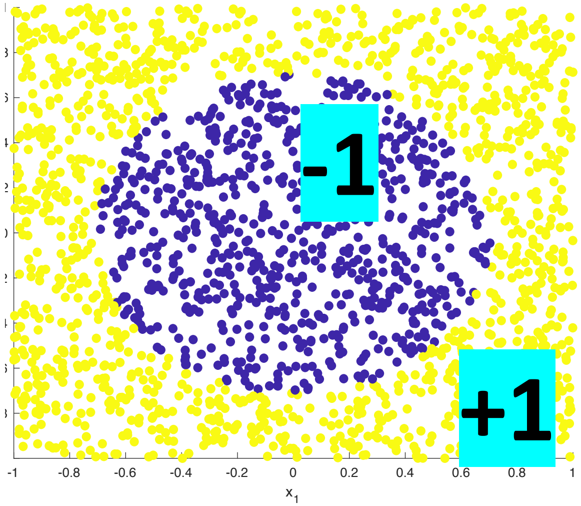
Idea 1: We can create curved boundaries by introducing non-linear features:

$$\text{new feature: } \boldsymbol{w} = (x_1, x_2, 1, x_1^2 + x_2^2)$$

$$\text{weight vector: } \boldsymbol{w} = (0, 0, -0.5, 1)$$

$$\Rightarrow \hat{y} = \mathrm{sign}(x_1^2 + x_2^2 - 0.5)$$



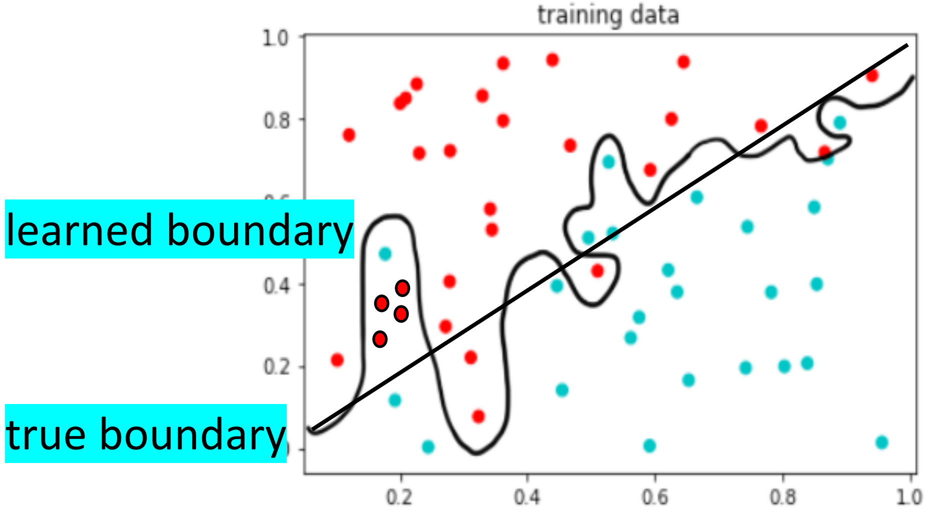


Idea 2: Overfitting.

Imagine that the true boundary is just linear.
But… there’s some random “noise flipping” around the boundary

Added too many powerful non-linear features. What happens?

Even though it looks working better on “train data”,
it may not perform well on “test data”. => cross-validation

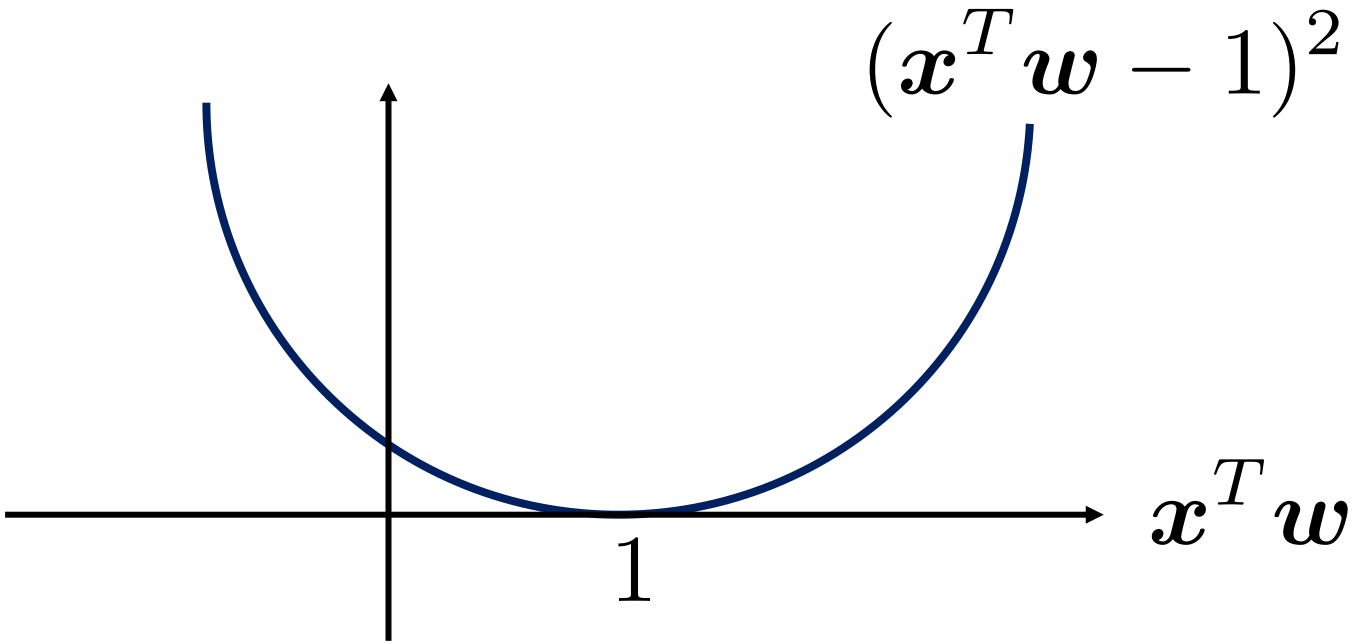


learned boundary

true boundary

Idea 3: Least squares can be a crummy loss function.

$$y = 1$$


$(\boldsymbol{x}^T\boldsymbol{w} - 1)^2$
$\boldsymbol{x}^T\boldsymbol{w}$
1