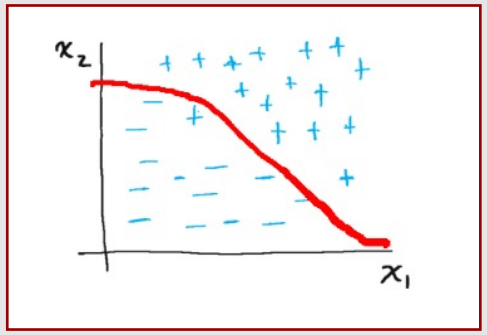
CS/ECE/ME 532
Matrix Methods in Machine Learning

Welcome!

Activity 8

Binary Linear Classifier

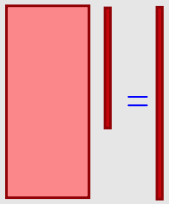


$$\text{sign}(\boldsymbol{x}_1^T \boldsymbol{w}) = -1$$
$$\text{sign}(\boldsymbol{x}_2^T \boldsymbol{w}) = +1$$
$$\vdots$$

Drop "sign"

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

Lots of training data



(no solution)

Any downsides to the squared error loss function?

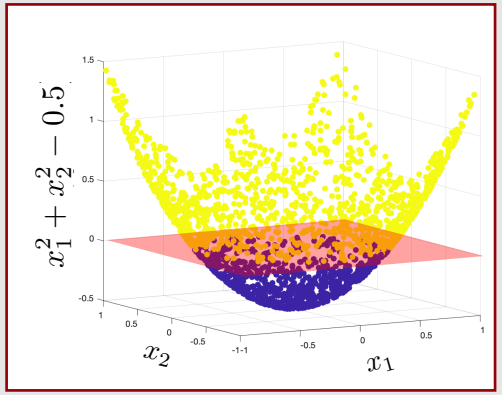
$$\boldsymbol{w}^* = \arg\min_{\boldsymbol{w}} ||\boldsymbol{X}\boldsymbol{w} - \boldsymbol{y}||_2^2$$

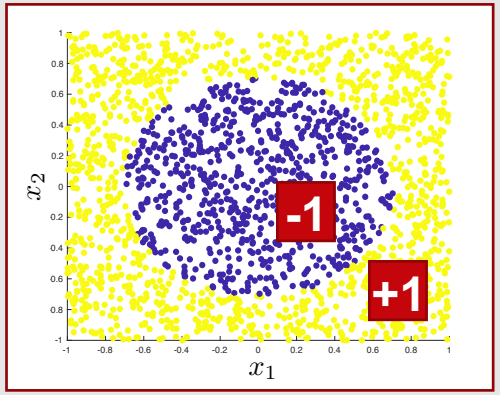
$$\boldsymbol{w}^* = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{y}$$

Curved boundaries

$$\mathbf{x}^T = \begin{bmatrix} x_1 & x_2 & 1 & x_1^2 + x_2^2 \end{bmatrix}$$

$$\boldsymbol{w} = (0, 0, -0.5, 1)$$
$$\Rightarrow \hat{y} = \text{sign}(x_1^2 + x_2^2 - 0.5)$$

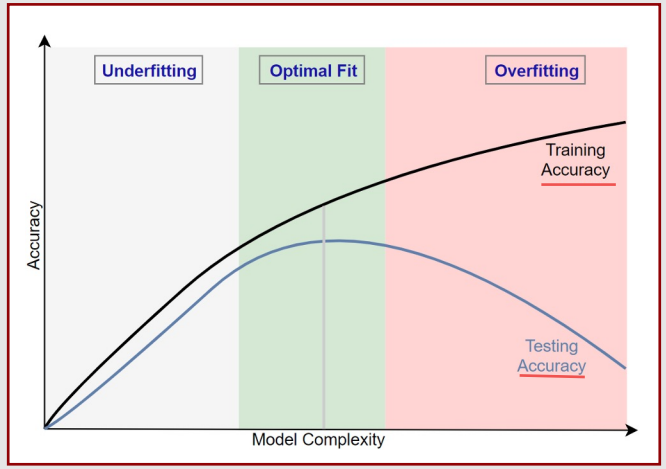




(Bad) idea: Keep adding many powerful non-linear features!

- Leads to **overfitting**
  - High training performance, poor testing performance

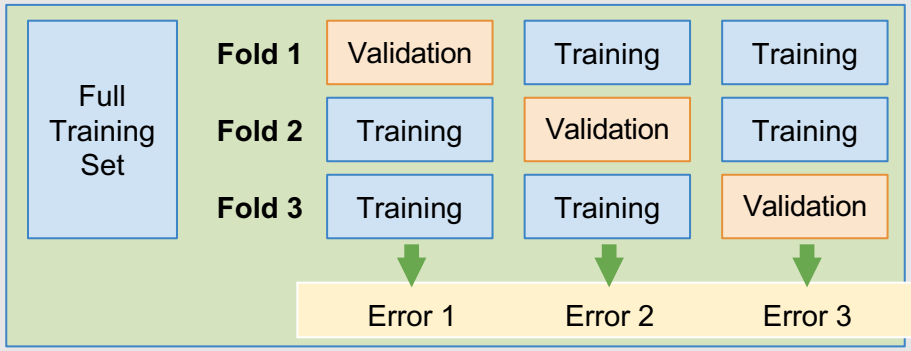
true boundary



Underfitting is also undesirable!

How can we find the "right" number of non-linear features?

Cross-validation:



Do this for each set of features

Choose the set with smallest average CV error