Support Vector Machines

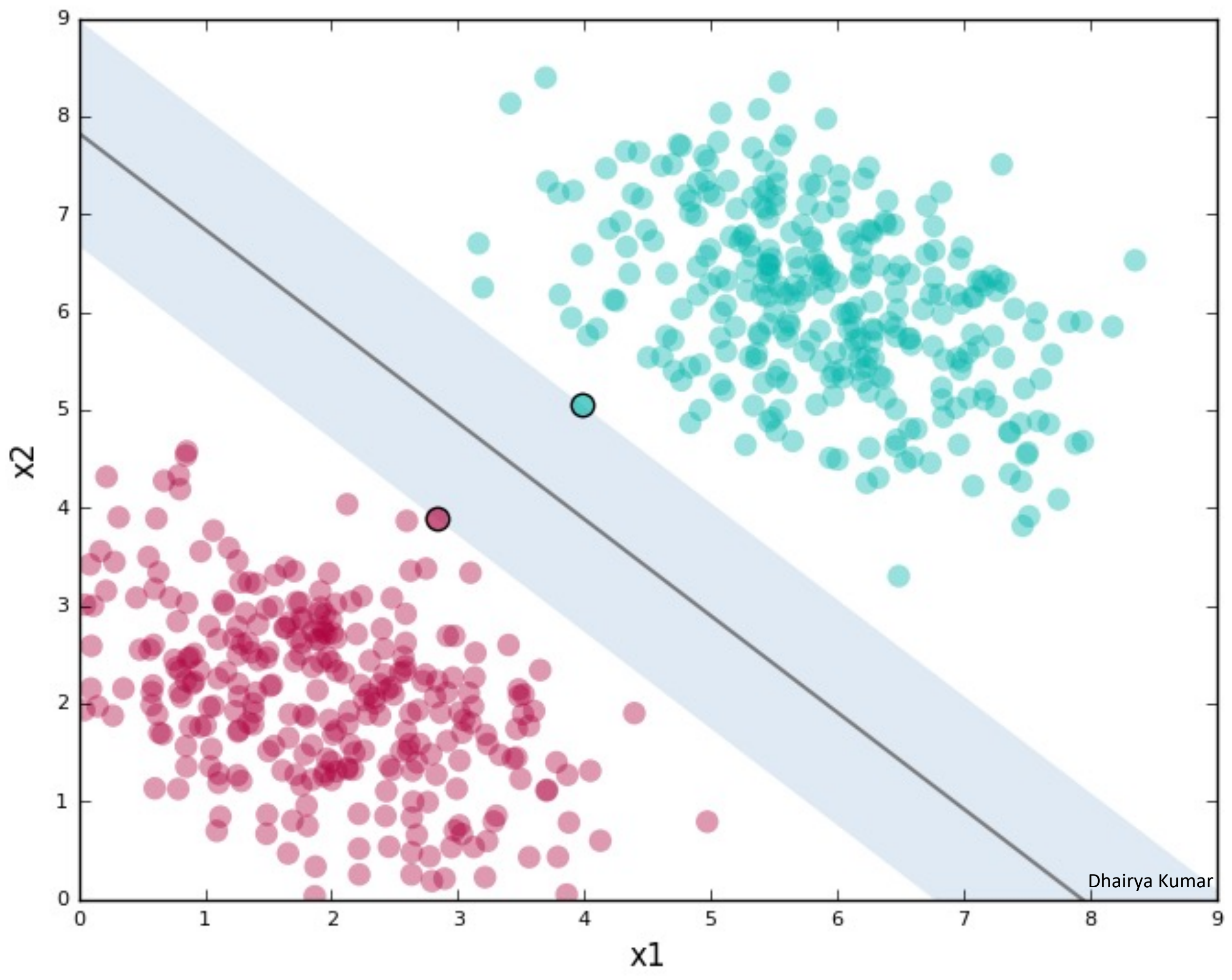
COSC 410: Applied Machine Learning

Spring 2022

Prof. Apthorpe

Outline

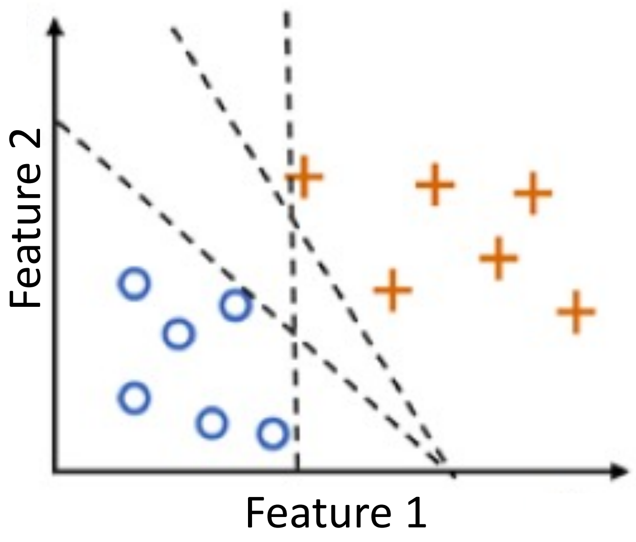
- Linear SVM Intuition
- SVM Model Notation
- Linear SVM
  - Model & Prediction
  - Training
- Non-Linear SVMs
  - Kernel Trick
- Multiclass SVM
- Practical Use of SVMs



Linear SVM Intuition

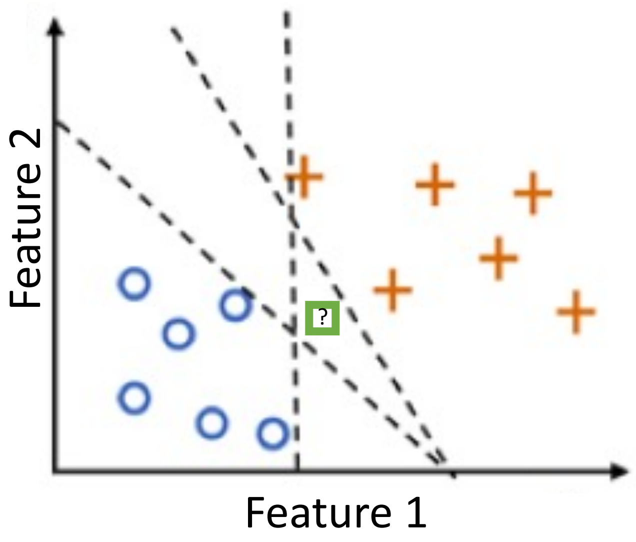
- Ideally, training data are **linearly separable**
  - An **N-dimensional line** can divide the data by class

Binary classification!
More on multiclass later…



Linear SVM Intuition

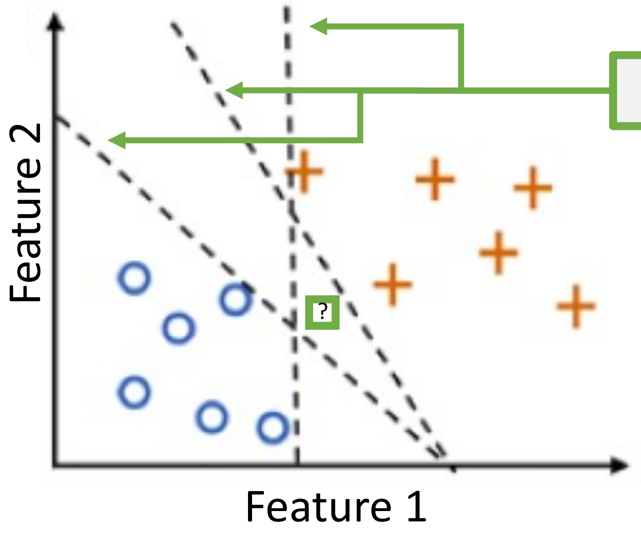
- Ideally, training data are **linearly separable**
  - An **N-dimensional line** can divide the data by class
  - Predict new examples based on which side of the line they are located



Linear SVM Intuition

- Ideally, training data are **linearly separable**
  - An **N-dimensional line** can divide the data by class
  - Predict new examples based on which side of the line they are located

Infinite possible lines separate this data.
Which is the best line to draw?

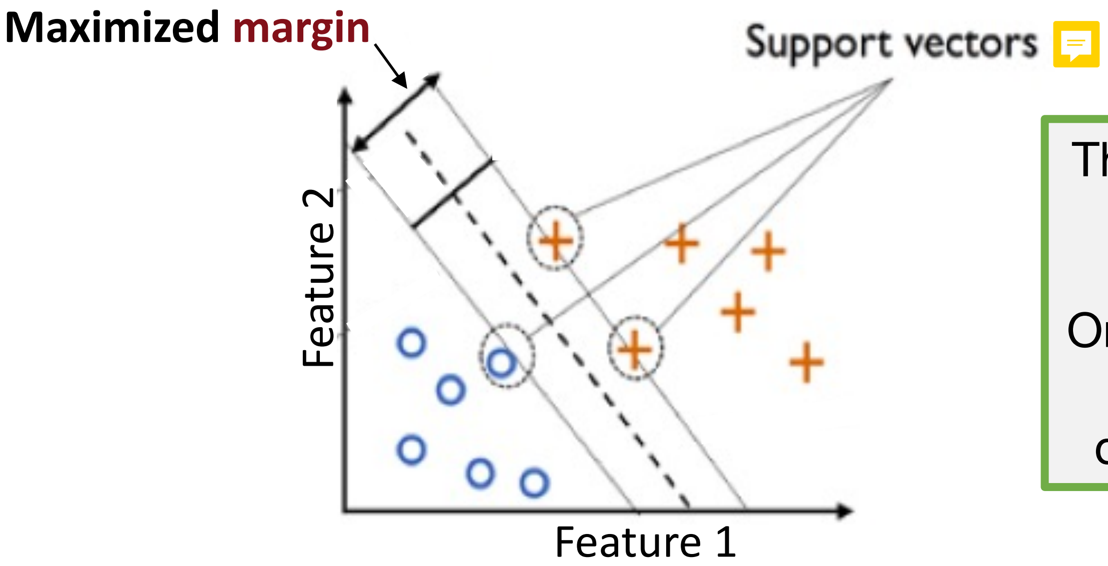


Why might none of these 3 lines be the best?

Generalizability!
The best line stays as far away from the training examples as possible

Linear SVM Intuition

- Ideally, training data are **linearly separable**
  - An **N-dimensional line** can divide the data by class
  - Predict new examples based on which side of the line they are located

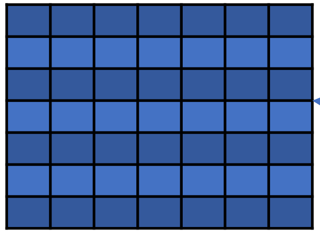


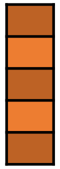
The best line maximizes the distance between it and the **nearest examples**

Only these **support vectors** matter to the line's position, adding more data points further away from the line doesn't change the model → very robust to overfitting

Training Models Review

Remember to **standardize**



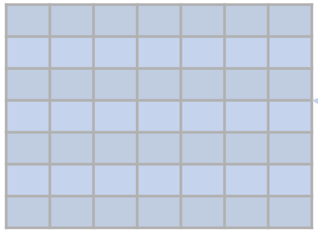
- Model training is a **minimization** process
  - **Training data** $\mathbf{X}$ of individual feature vectors $\mathbf{x}$
  - **Training labels** $\mathbf{y}$ 
  - Chosen **error function** $E()$
  - **Model** $h_{\boldsymbol{\theta}}()$ (features $\mathbf{X}$ → predicted labels $\hat{\mathbf{y}}$)
    - Parameters: $\boldsymbol{\theta} = [\theta_0, \theta_1, \theta_2]$
  - **GOAL:** Find model parameters that minimize $E(h_{\boldsymbol{\theta}}(\mathbf{X}), \mathbf{y})$

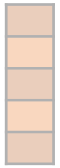
predicted labels

actual labels

Training Models: SVM Notation

Remember to **standardize**



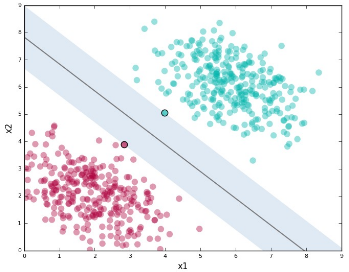
- Model training is a **minimization** process
  - **Training data** X of individual feature vectors $\mathbf{x}$
  - **Training labels** $\mathbf{y}$ 
  - Chosen **error function** $E()$
  - **Model** $h_{\mathbf{w},b}()$ (features $\mathbf{X}$ → predicted labels $\hat{\mathbf{y}}$)
    - Parameters: Weights vector $\mathbf{w} = [w_1, w_2, w_3]$ and bias scalar b

predicted labels

- **GOAL:** Find model parameters that minimize $E(h_{\mathbf{w},b}(\mathbf{X}, \mathbf{y})$

actual labels

Linear SVM Model



- Predicted labels are a **piecewise linear combination** of features
  - Model parameters: weights $\mathbf{w} = [w_1, w_2, w_3, \ldots]$ and bias b
  - For **each** example $\mathbf{x} = [x_1, x_2, x_3, \ldots]$

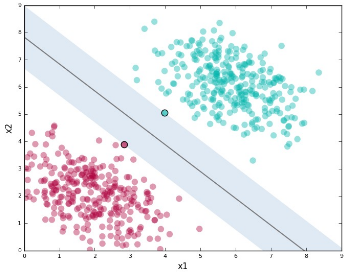
Predicted label

$$\hat{y} = \begin{cases} 0 & \text{if} \quad w_1 x_1 + w_2 x_2 + \cdots + b < 0 \\ 1 & \text{if} \quad w_1 x_1 + w_2 x_2 + \cdots + b \geq 0 \end{cases}$$

Example is “**above**” the decision line

Example is “**below**” the decision line

Linear SVM Model



- Predicted labels are a **piecewise linear combination** of features
  - Model parameters: weights $\mathbf{w} = [w_1, w_2, w_3, \ldots]$ and bias b
  - For **each** example $\mathbf{x} = [x_1, x_2, x_3, \ldots]$

Predicted label

$$\hat{y} = \begin{cases} 0 & \text{if} \quad w_1 x_1 + w_2 x_2 + \cdots + b < 0 \\ 1 & \text{if} \quad w_1 x_1 + w_2 x_2 + \cdots + b \geq 0 \end{cases}$$

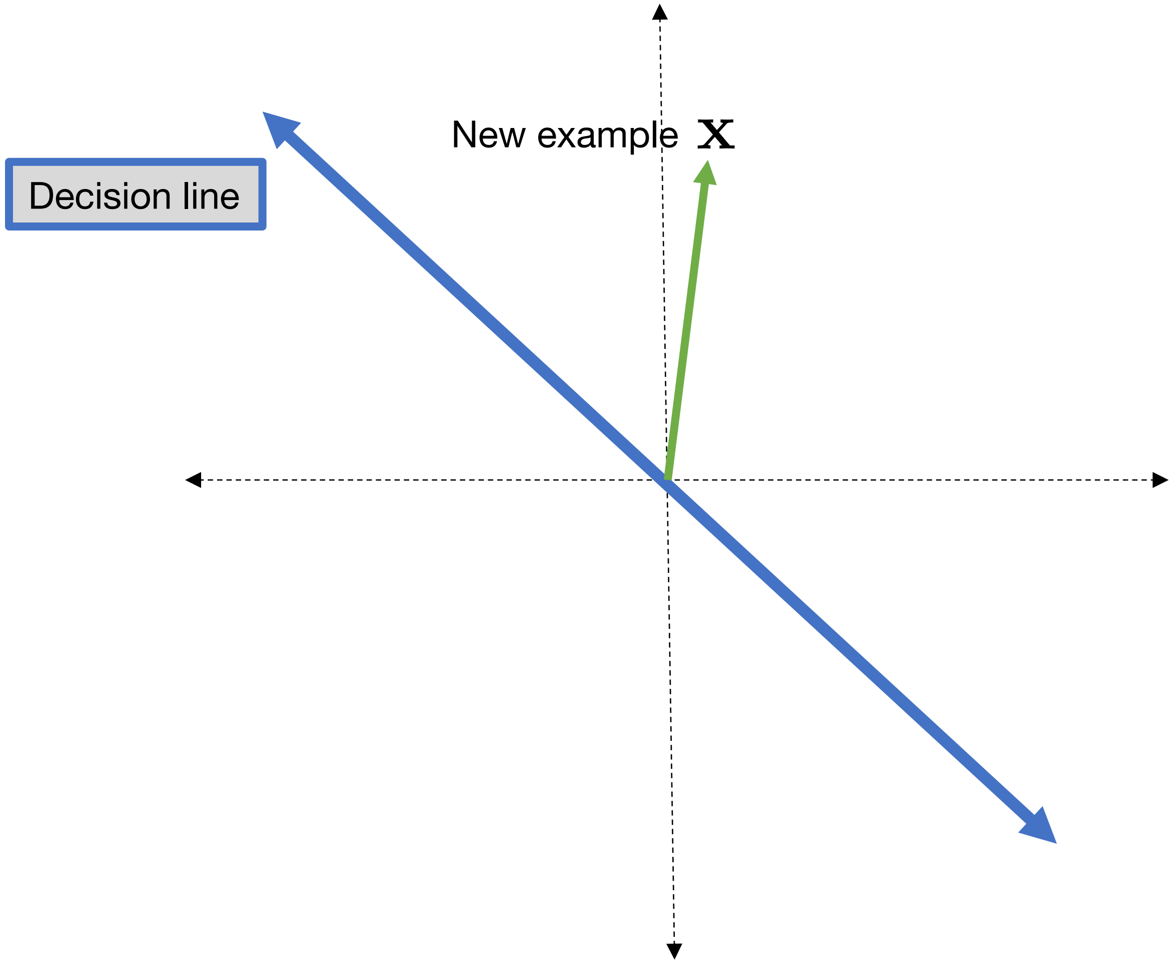
Example is “**below**” the decision line

Example is “**above**” the decision line

$$= \begin{cases} 0 & \text{if} \quad \mathbf{w} \cdot \mathbf{x} + b < 0 \\ 1 & \text{if} \quad \mathbf{w} \cdot \mathbf{x} + b \geq 0 \end{cases} \qquad = \begin{cases} 0 & \text{if} \quad \mathbf{w}^\intercal \mathbf{x} + b < 0 \\ 1 & \text{if} \quad \mathbf{w}^\intercal \mathbf{x} + b \geq 0 \end{cases}$$

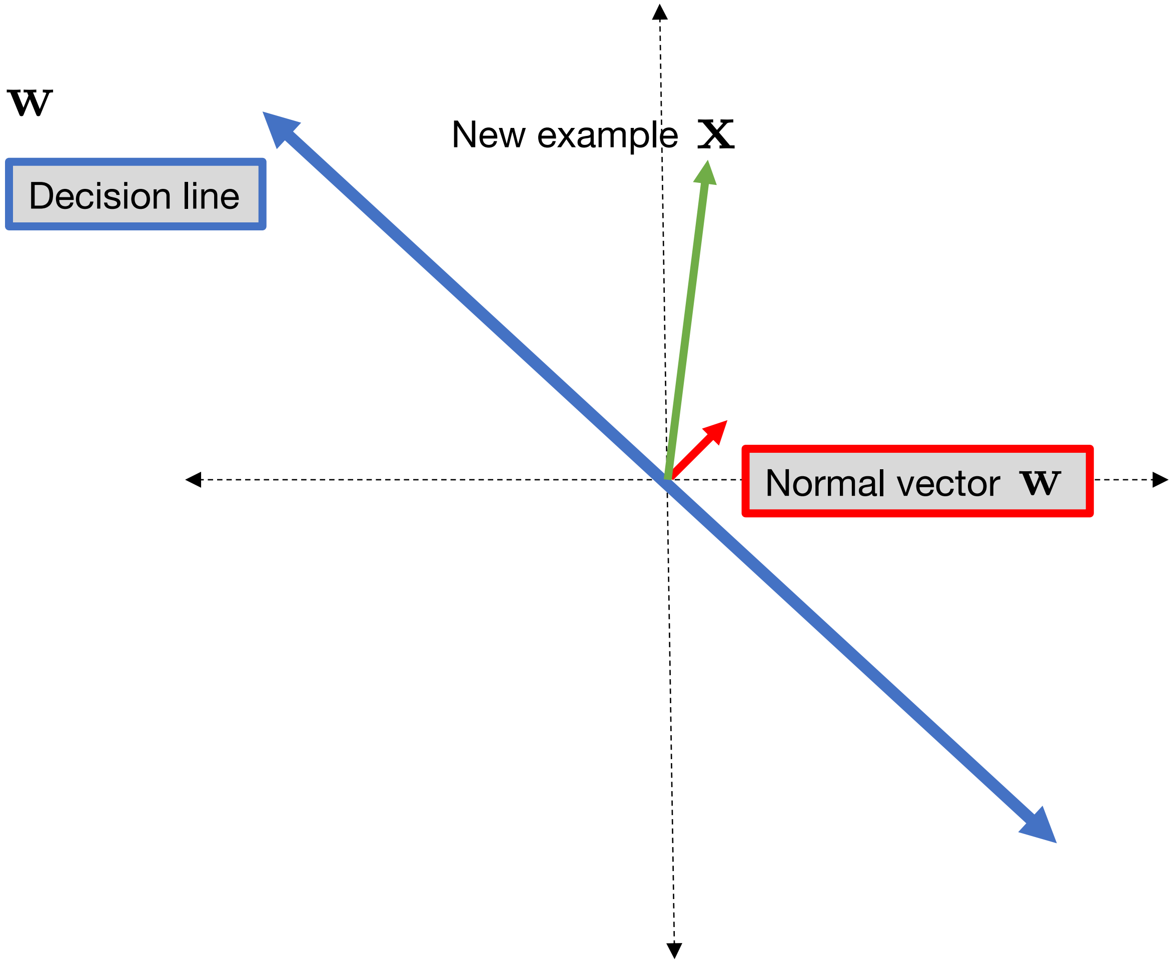
Linear SVM Model

- Is $\mathbf{x}$ "above" or "below" the decision line?



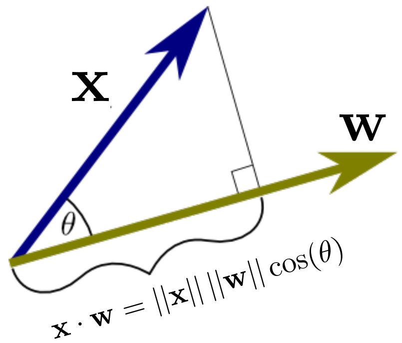
Linear SVM Model

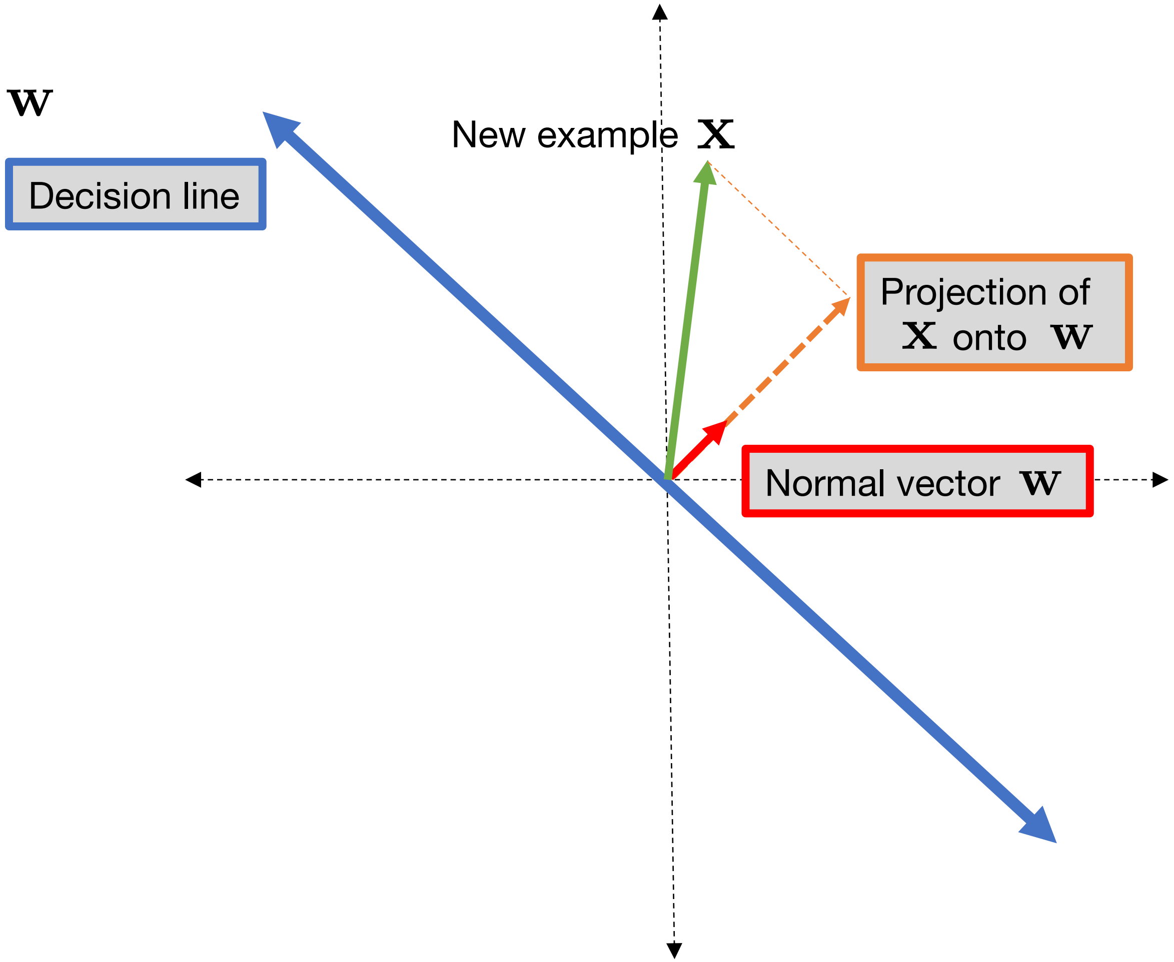
- Is $\mathbf{x}$ "above" or "below" the decision line?
- To find out, *project* $\mathbf{x}$ onto the line's normal vector $\mathbf{w}$



Linear SVM Model

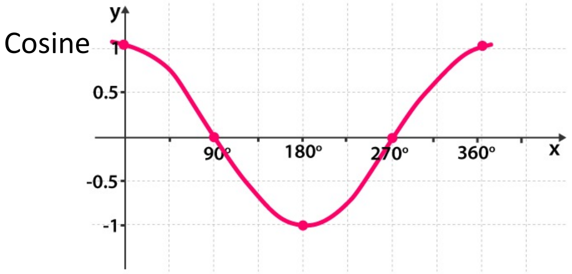
- Is $\mathbf{x}$ “above” or “below” the decision line?
- To find out, *project* $\mathbf{x}$ onto the line’s normal vector $\mathbf{w}$



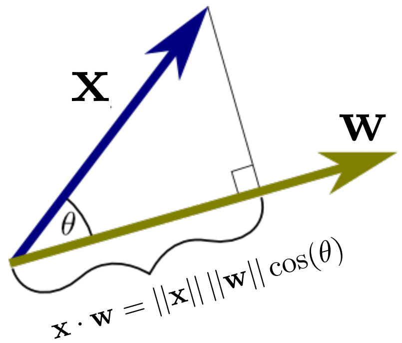


- This is the magnitude of $\mathbf{x}$ *in the direction of* $\mathbf{w}$

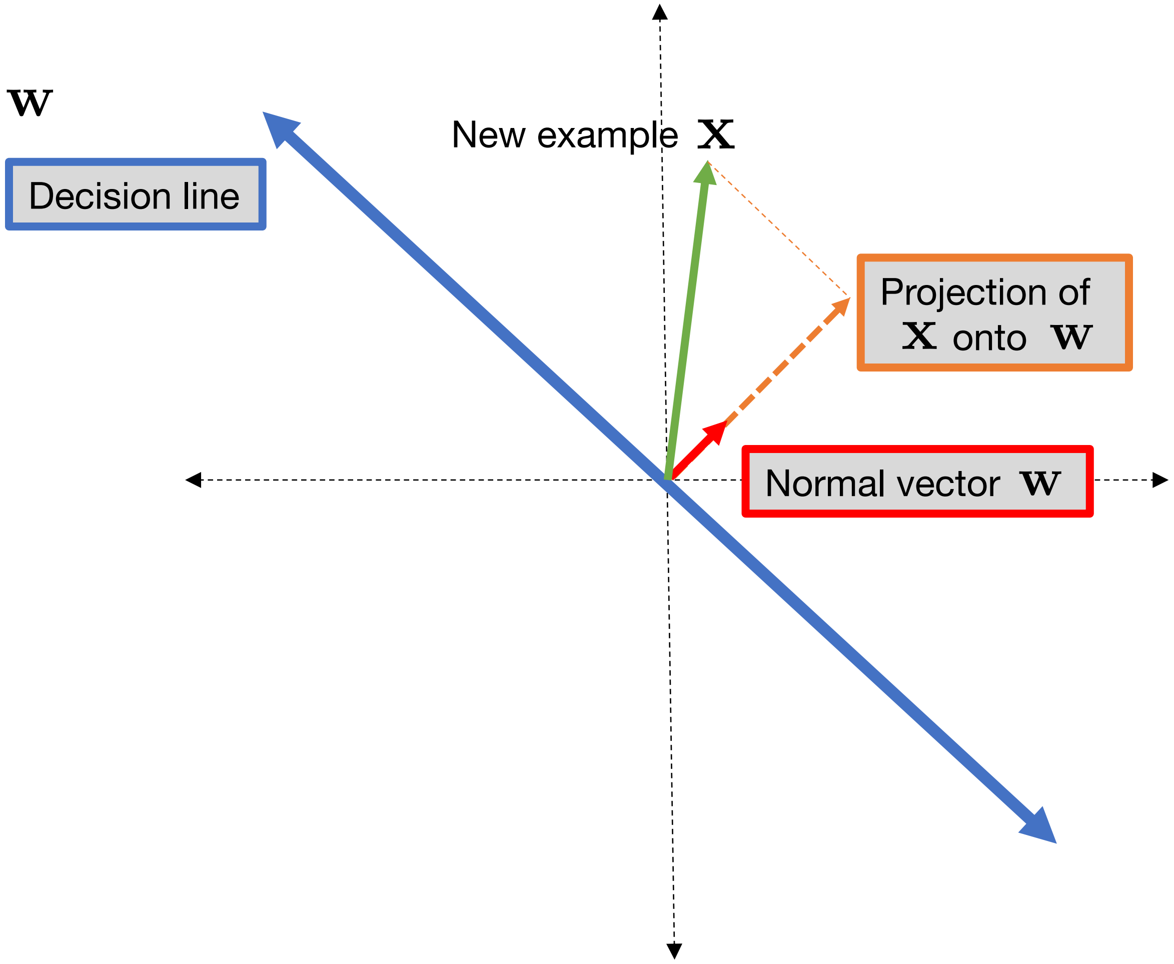
Linear SVM Model



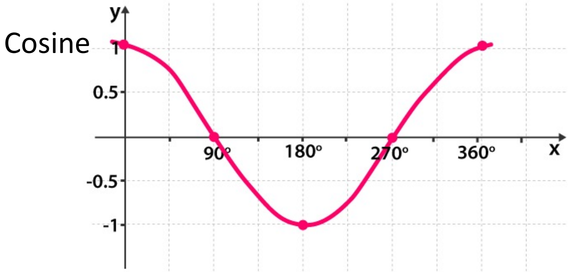
- Is $\mathbf{x}$ "above" or "below" the decision line?
- To find out, *project* $\mathbf{x}$ onto the line's normal vector $\mathbf{w}$



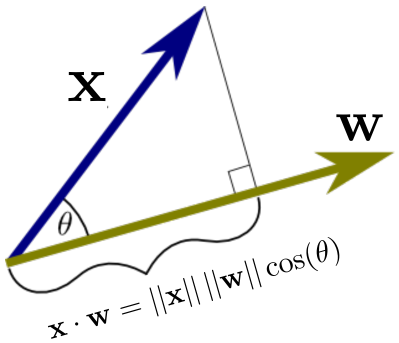
- This is the magnitude of $\mathbf{x}$ *in the direction of* $\mathbf{w}$
- If **positive,** $\mathbf{x}$ is **above** the decision line
- If **negative,** $\mathbf{x}$ is **below** the decision line



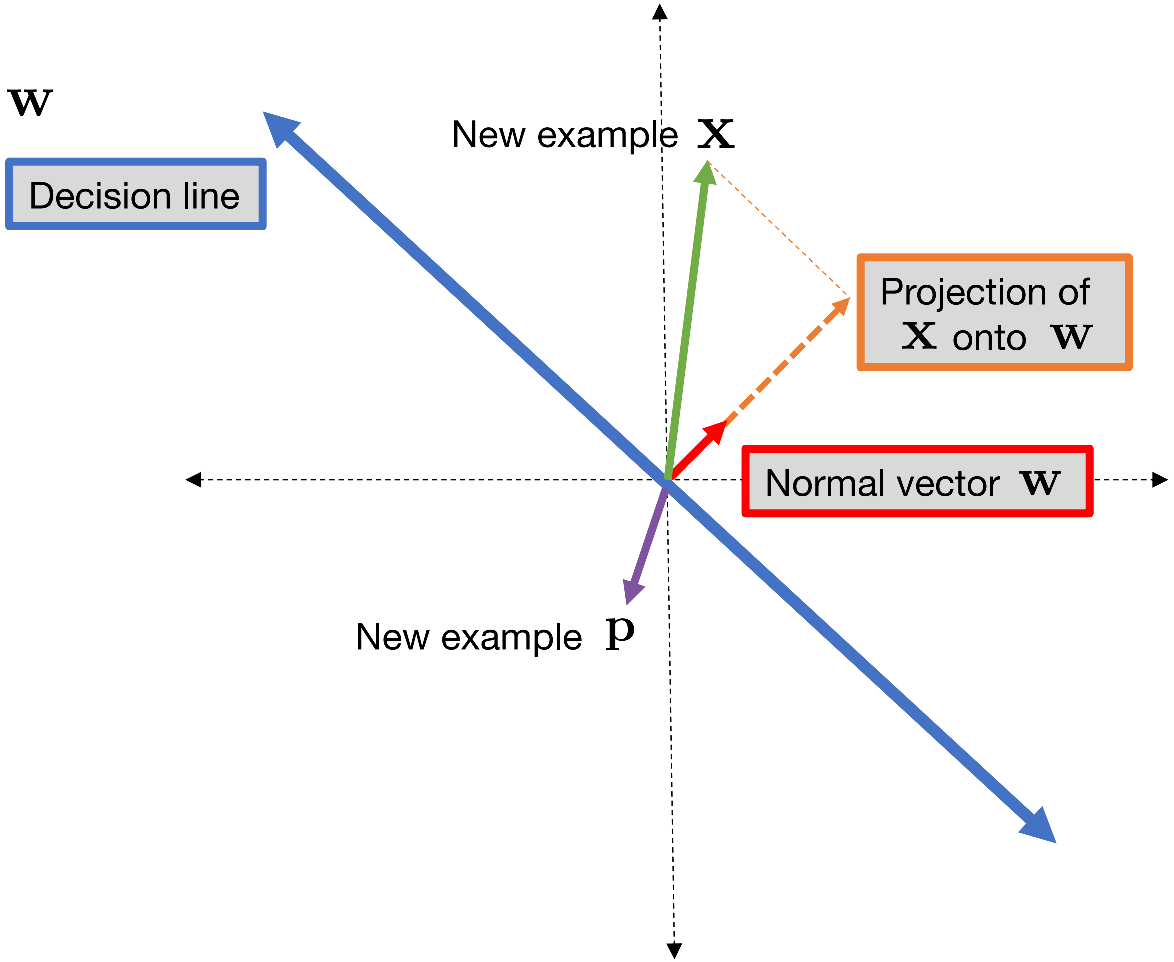
Linear SVM Model



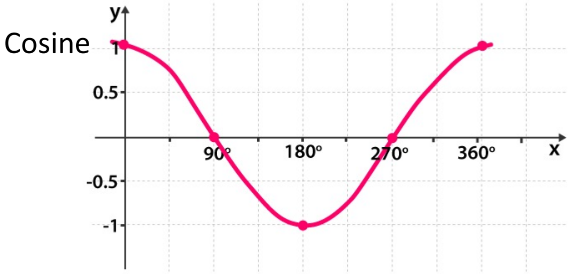
- Is $\mathbf{x}$ "above" or "below" the decision line?
- To find out, *project* $\mathbf{x}$ onto the line's normal vector $\mathbf{w}$



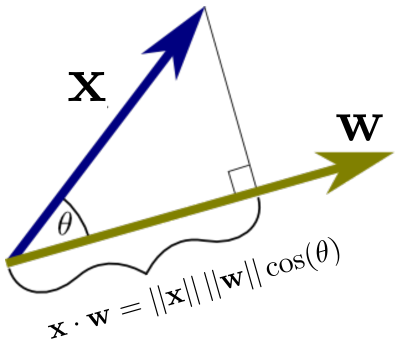
- This is the magnitude of $\mathbf{x}$ *in the direction of* $\mathbf{w}$
- If **positive,** $\mathbf{x}$ is **above** the decision line
- If **negative,** $\mathbf{x}$ is **below** the decision line



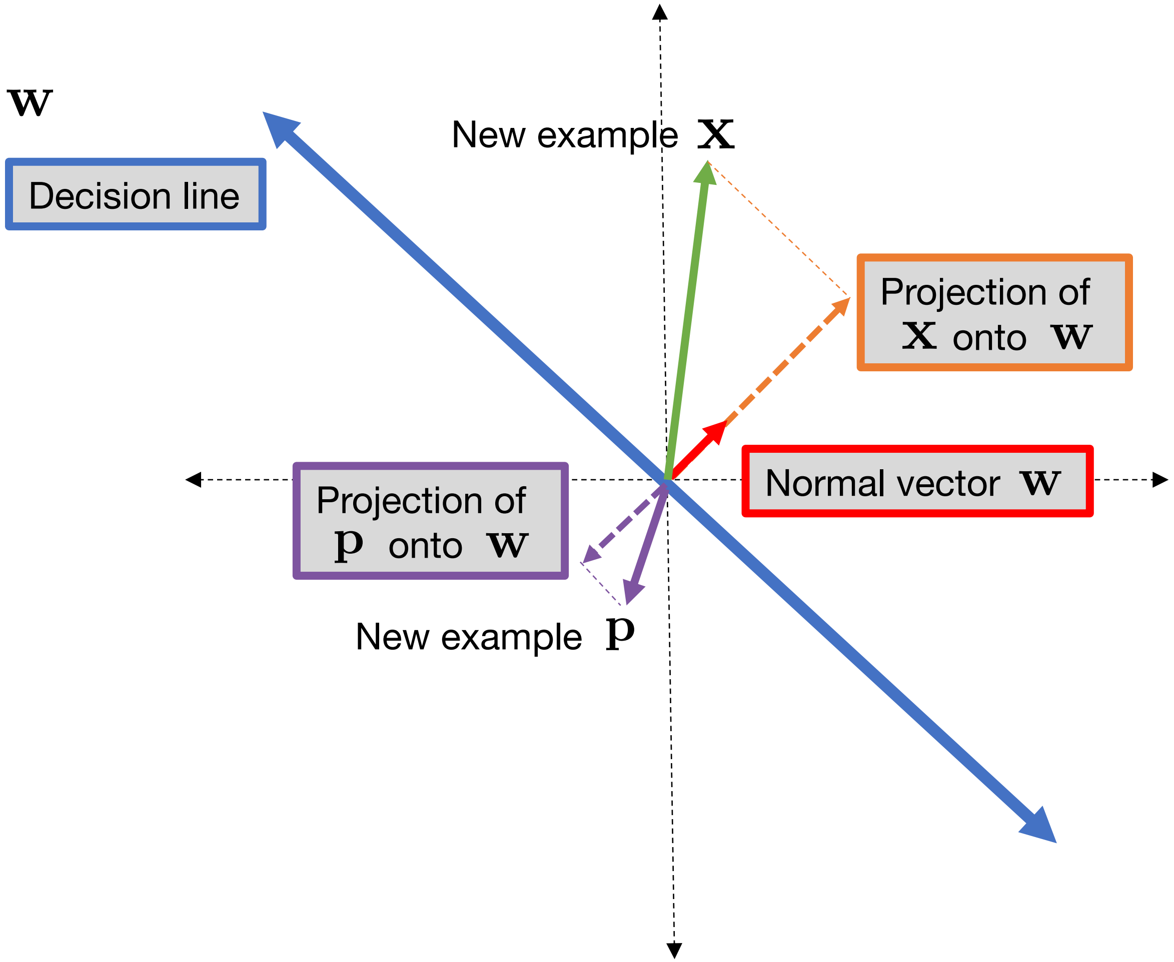
Linear SVM Model



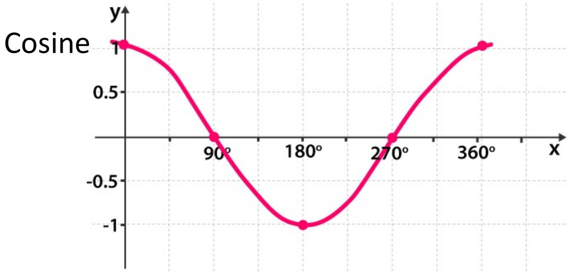
- Is $\mathbf{x}$ "above" or "below" the decision line?
- To find out, *project* $\mathbf{x}$ onto the line's normal vector $\mathbf{w}$



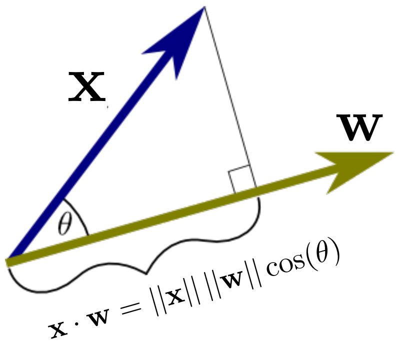
- This is the magnitude of $\mathbf{x}$ *in the direction of* $\mathbf{w}$
- If **positive,** $\mathbf{x}$ is **above** the decision line
- If **negative,** $\mathbf{x}$ is **below** the decision line

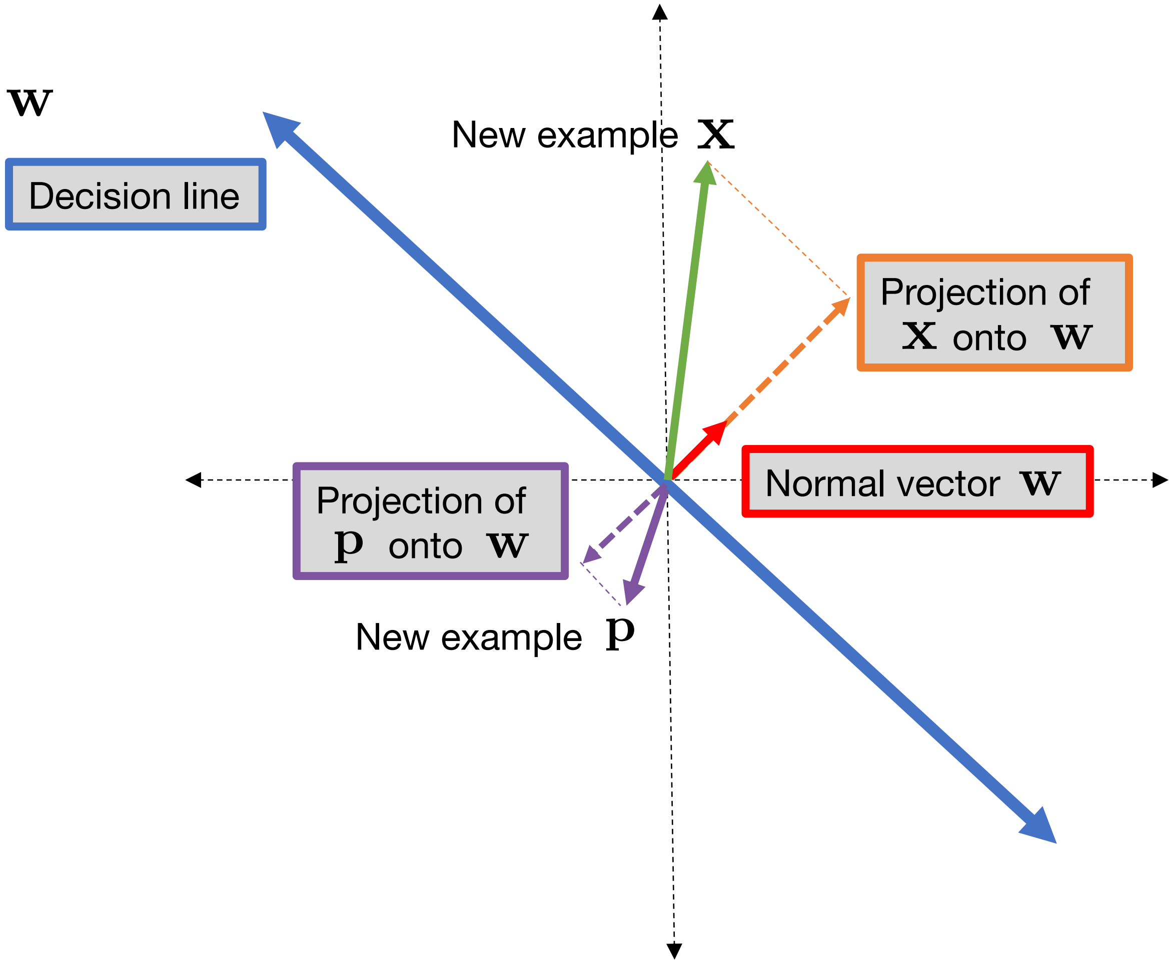


Linear SVM Model



- Is $\mathbf{x}$ “above” or “below” the decision line?
- To find out, *project* $\mathbf{x}$ onto the line’s normal vector $\mathbf{w}$





- This is the magnitude of $\mathbf{x}$ *in the direction of* $\mathbf{w}$
- If **positive,** $\mathbf{x}$ is **above** the decision line
- If **negative,** $\mathbf{x}$ is **below** the decision line
- If best decision line isn’t through origin, add bias b

Linear SVM Model

- Predicted labels

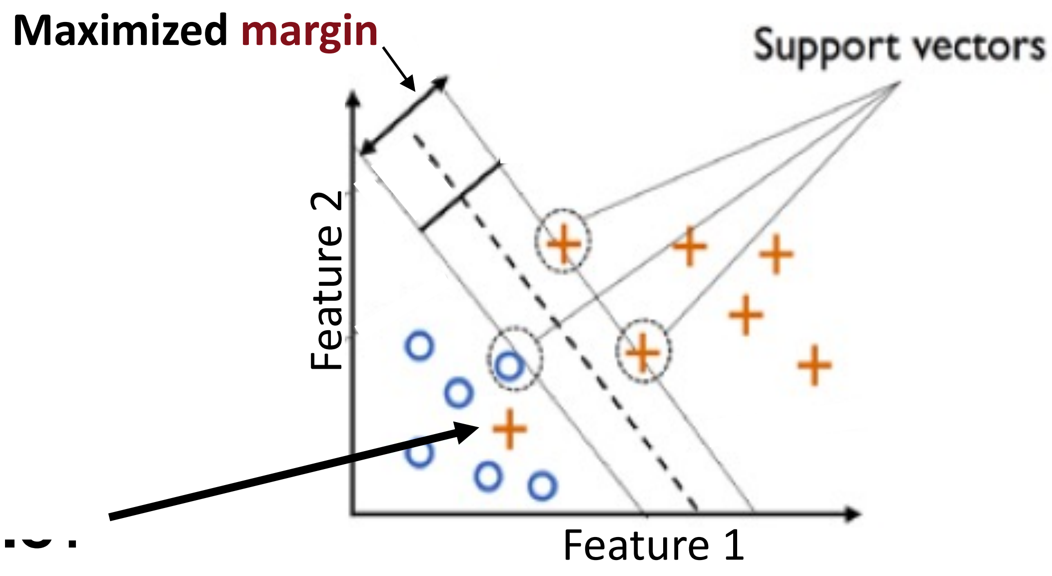
$$\hat{y} = \begin{cases} 0 & \text{if} \quad \mathbf{w}^\intercal \mathbf{x} + b < 0 \\ 1 & \text{if} \quad \mathbf{w}^\intercal \mathbf{x} + b \geq 0 \end{cases}$$

Example is "**below**" the decision line

Example is "**above**" the decision line

- What about the **margin?**
  - Important for **training**, not prediction

- What if data are **not linearly separable?**



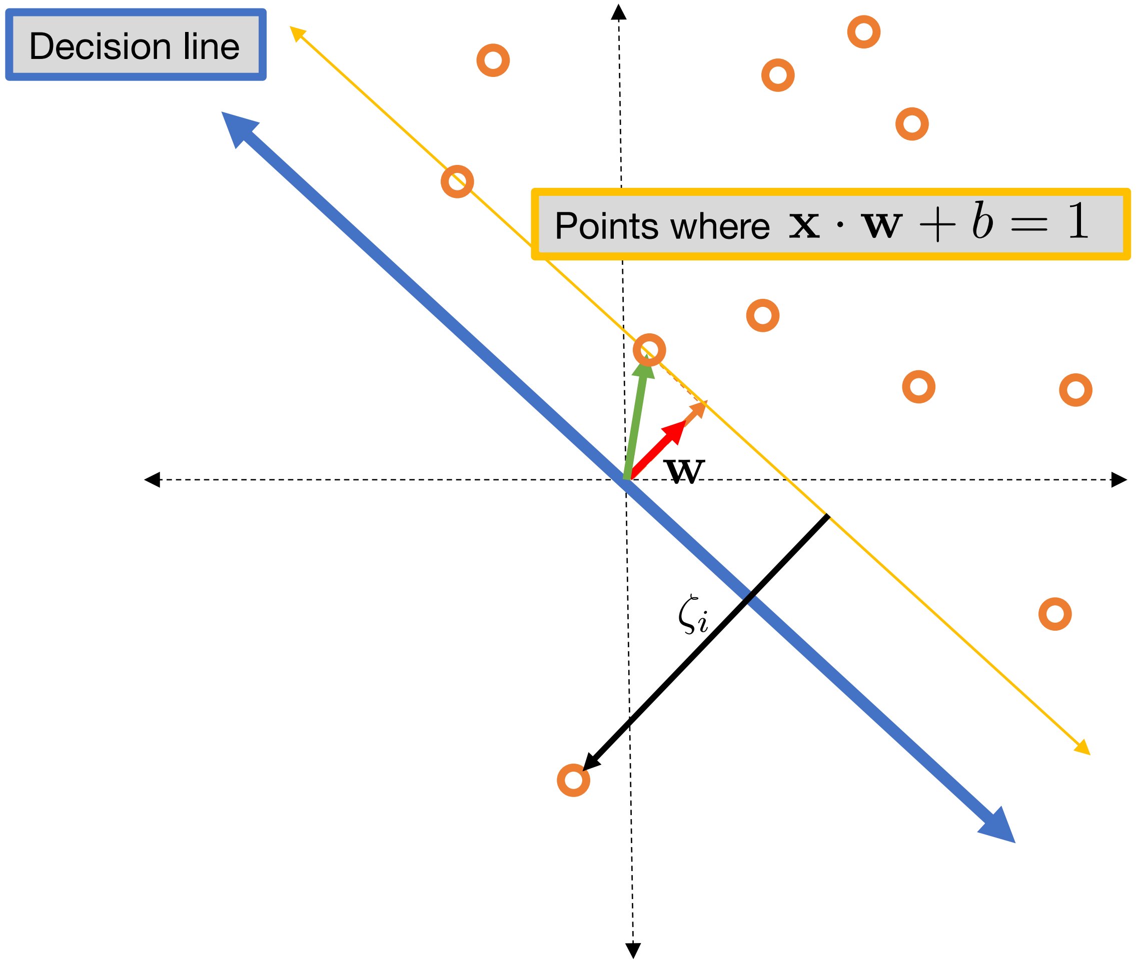
Linear SVM Training

Slack variable ζ accounts for non-separable data in a **soft-margin SVM**

- **Goal pt. 1:** Find $\mathbf{w}$ and b such that

$$\mathbf{w}^{\intercal}\mathbf{x} + b \geq 1 - \zeta$$

for all training examples in “1” class AND



Linear SVM Training

Slack variable ζ accounts for non-separable data in a **soft-margin SVM**

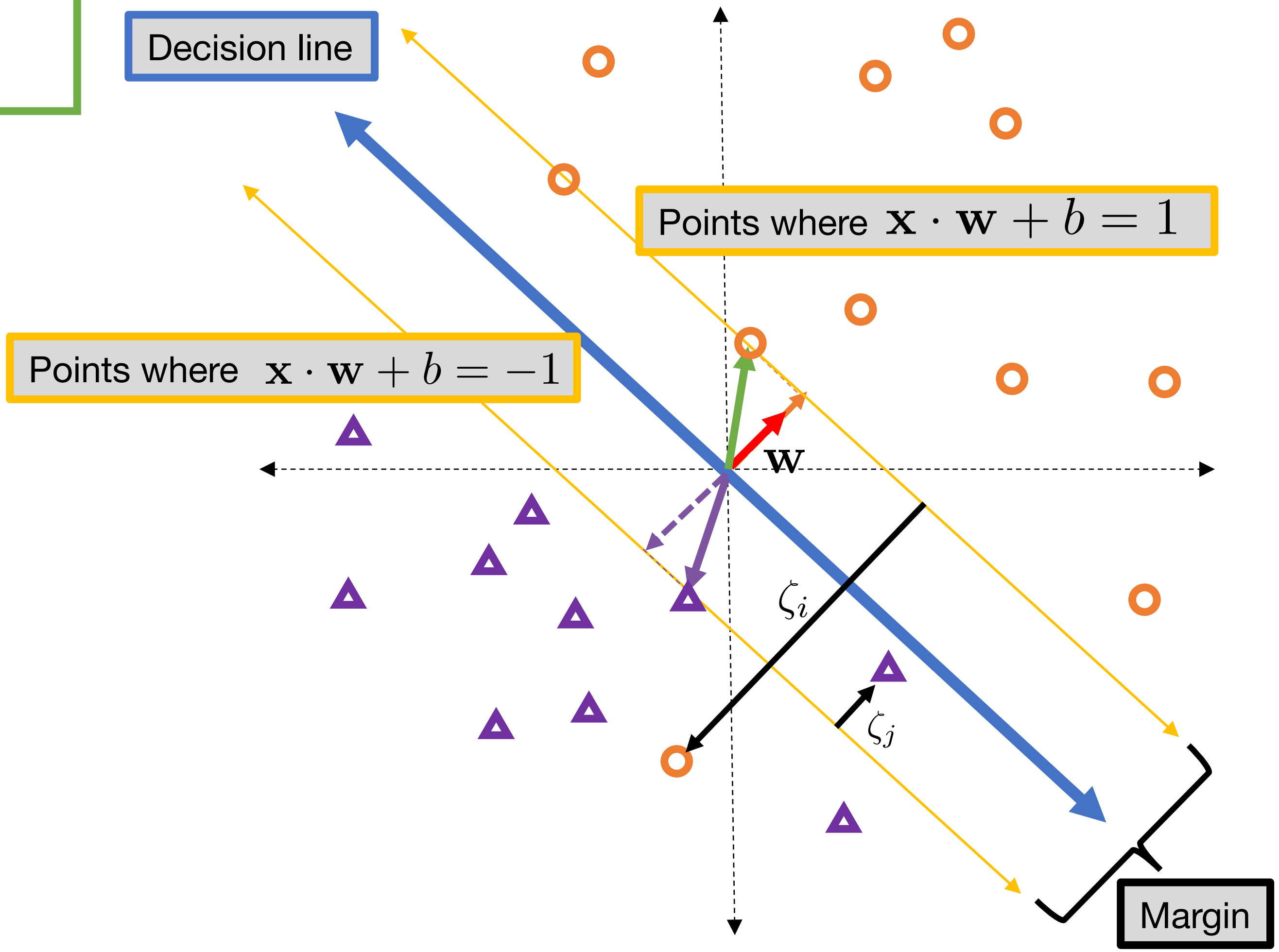
- **Goal pt. 1:** Find $\mathbf{w}$ and b such that

$$\mathbf{w}^\intercal \mathbf{x} + b \geq 1 - \zeta$$

for all training examples in “1” class AND

$$\mathbf{w}^\intercal \mathbf{x} + b \leq -1 + \zeta$$

for all training examples in “0” class



Linear SVM Training

Slack variable ζ accounts for non-separable data in a **soft-margin SVM**

- **Goal pt. 1:** Find $\mathbf{w}$ and b such that

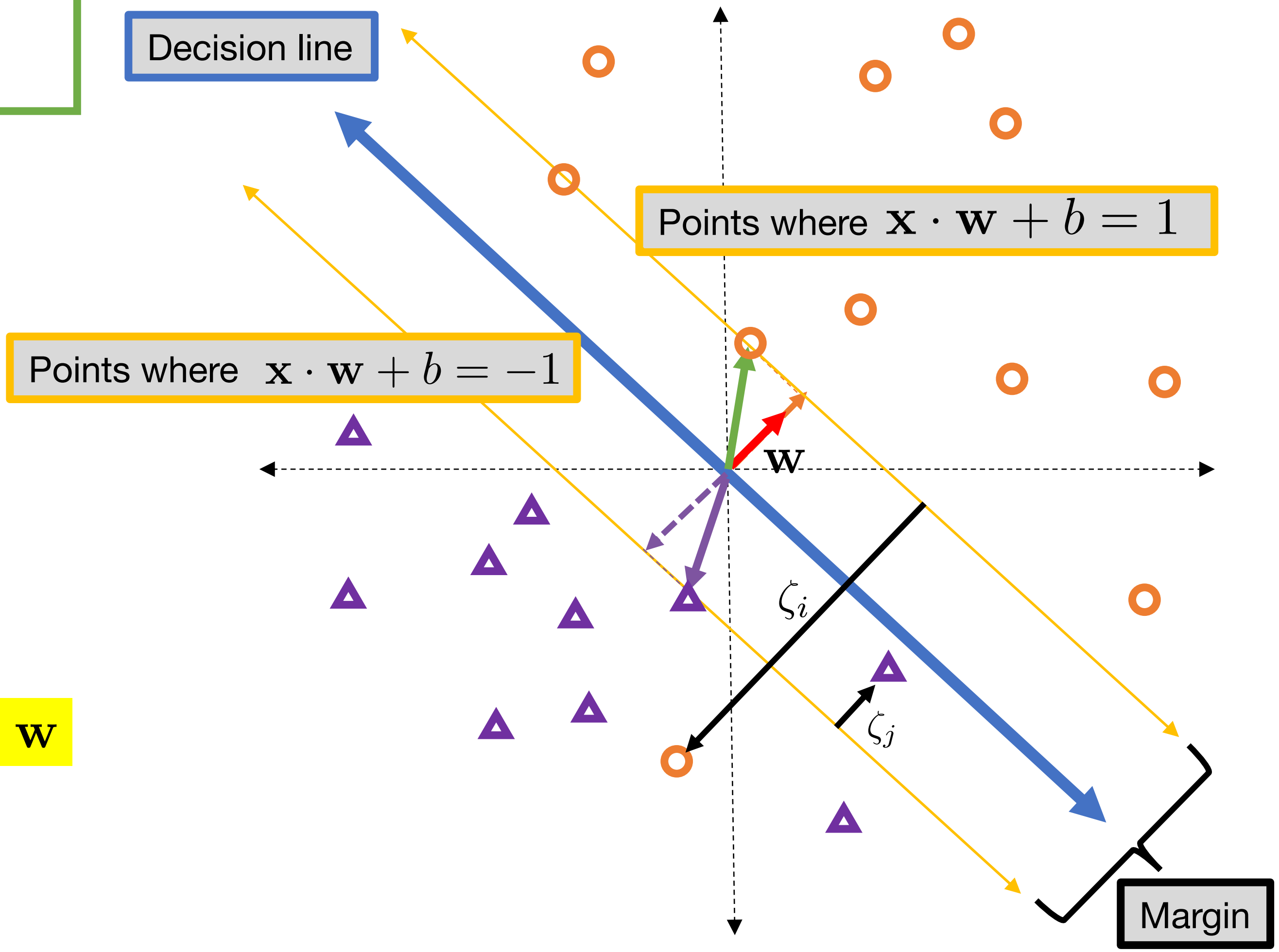
$$\mathbf{w}^{\intercal}\mathbf{x} + b \geq 1 - \zeta$$

for all training examples in “1” class AND

$$\mathbf{w}^{\intercal}\mathbf{x} + b \leq -1 + \zeta$$

for all training examples in “0” class

- **Goal pt. 2:** Maximize the width of the margin
  - Equivalent to minimizing the magnitude of $\mathbf{w}$



Linear SVM Training

Slack variable ζ accounts for non-separable data in a **soft-margin SVM**

- **Goal pt. 1:** Find $\mathbf{w}$ and b such that

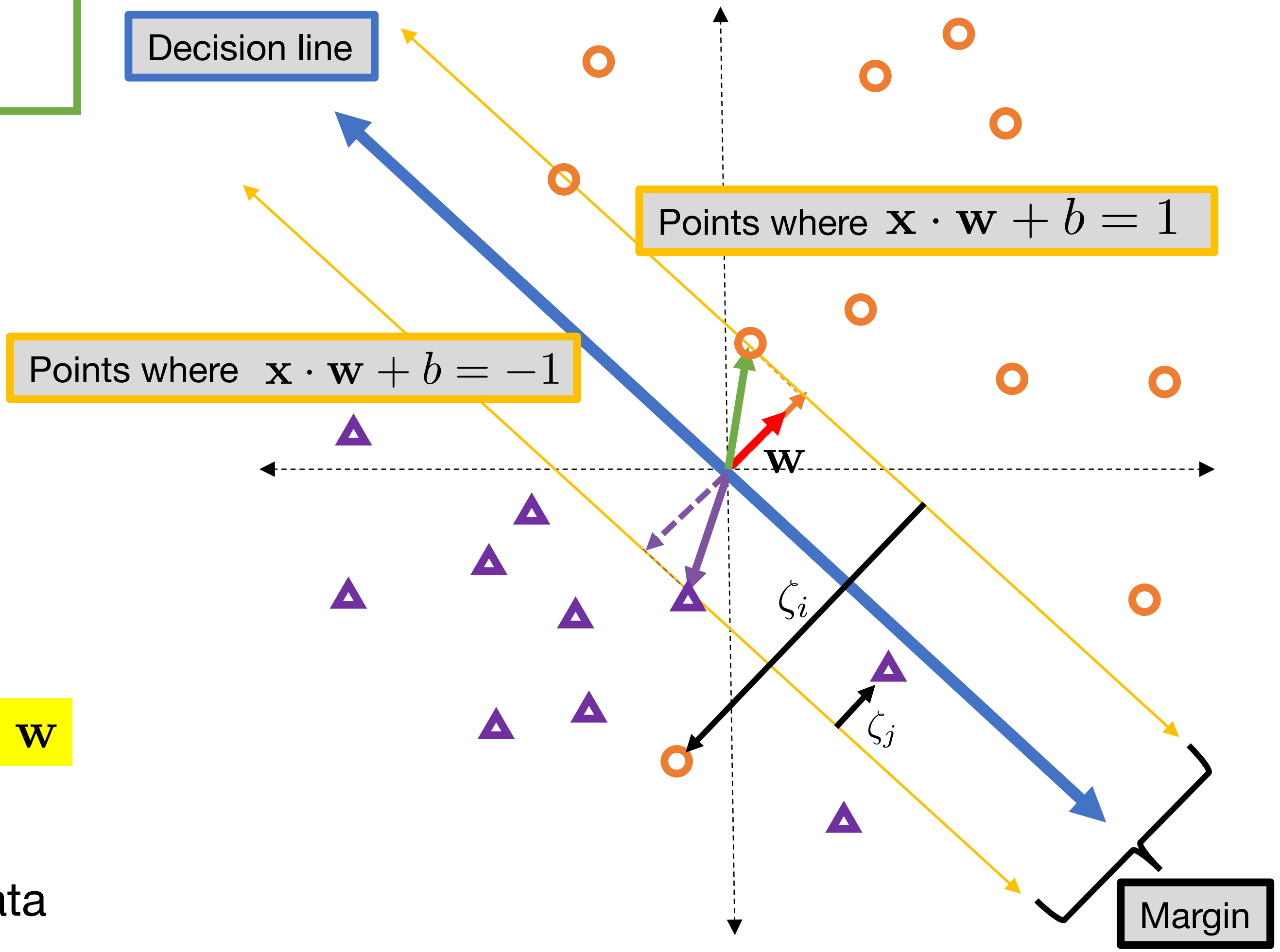
$$\mathbf{w}^\intercal \mathbf{x} + b \geq 1 - \zeta$$

for all training examples in "1" class AND

$$\mathbf{w}^\intercal \mathbf{x} + b \leq -1 + \zeta$$

for all training examples in "0" class

- **Goal pt. 2:** Maximize the width of the margin
  - Equivalent to minimizing the magnitude of $\mathbf{w}$
- **Goal pt. 3:** Minimize sum of slack variables ζ to minimize incorrect predictions on training data



Linear SVM Training

- **Goal pt. 1:** Find $\mathbf{w}$ and b such that

$$\mathbf{w}^\intercal \mathbf{x} + b \geq 1 - \zeta$$

for all training examples in "1" class AND

$$\mathbf{w}^\intercal \mathbf{x} + b \leq -1 + \zeta$$

for all training examples in "0" class

- **Goal pt. 2:** Maximize the width of the margin
  - Equivalent to minimizing the magnitude of $\mathbf{w}$
- **Goal pt. 3:** Minimize sum of slack variables ζ
to minimize incorrect predictions on training data

- **Option 1:** Use quadratic programming (QP) solver

Linear SVM Training

- **Goal pt. 1:** Find $\mathbf{w}$ and b such that

  $$\mathbf{w}^\intercal \mathbf{x} + b \geq 1 - \zeta$$

  for all training examples in “1” class AND

  $$\mathbf{w}^\intercal \mathbf{x} + b \leq -1 + \zeta$$

  for all training examples in “0” class

- **Goal pt. 2:** Maximize the width of the margin
  - Equivalent to minimizing the magnitude of $\mathbf{w}$

- **Goal pt. 3:** Minimize sum of slack variables ζ to minimize incorrect predictions on training data

- **Option 1:** Use quadratic programming (QP) solver

- **Option 2:** Minimize error function with gradient descent

Error function to minimize

Hyperparameter (weight placed on examples on “wrong side” of margin)

$$E(\mathbf{w}, b) = \frac{1}{2}\mathbf{w}^\intercal \mathbf{w} + C \sum_{i=1}^{m} hinge(\mathbf{w}^\intercal \mathbf{x_i} + b)$$

Maximize margin

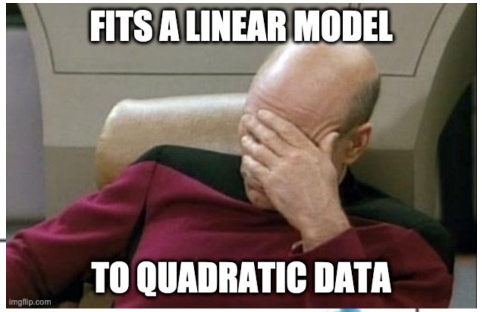
Penalty for examples on “wrong side” of margin

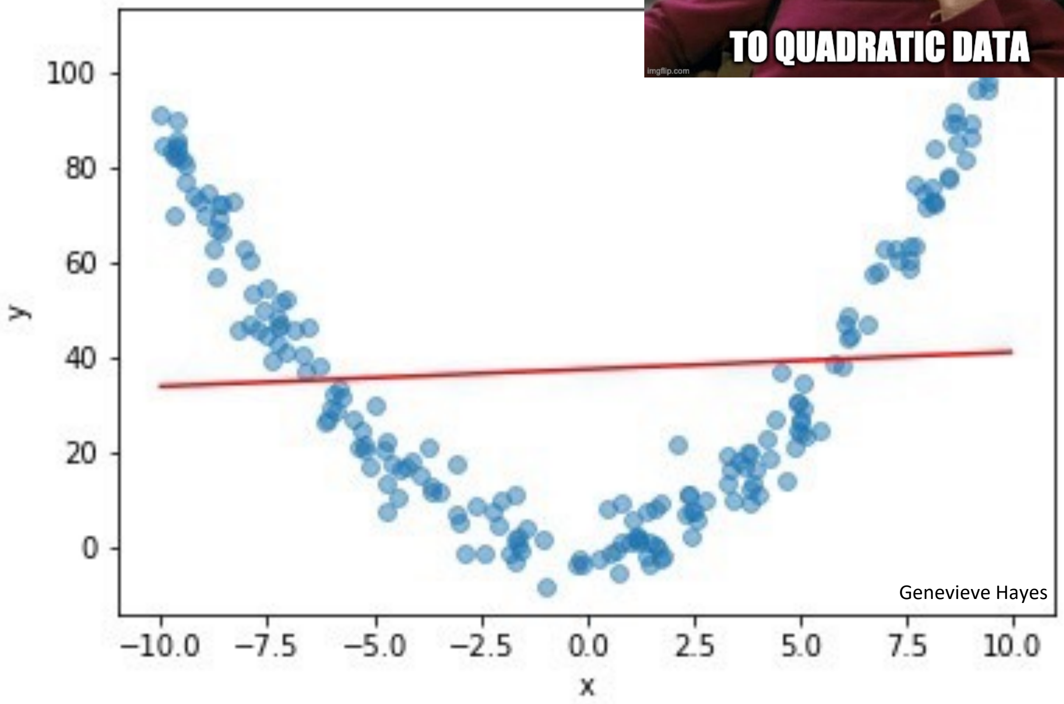
0 if on “right side” of margin, ζ_i otherwise

Non-Linear SVM

Non-Linear SVM

- Many datasets derive from nonlinear phenomena
  - Poor match for linear models

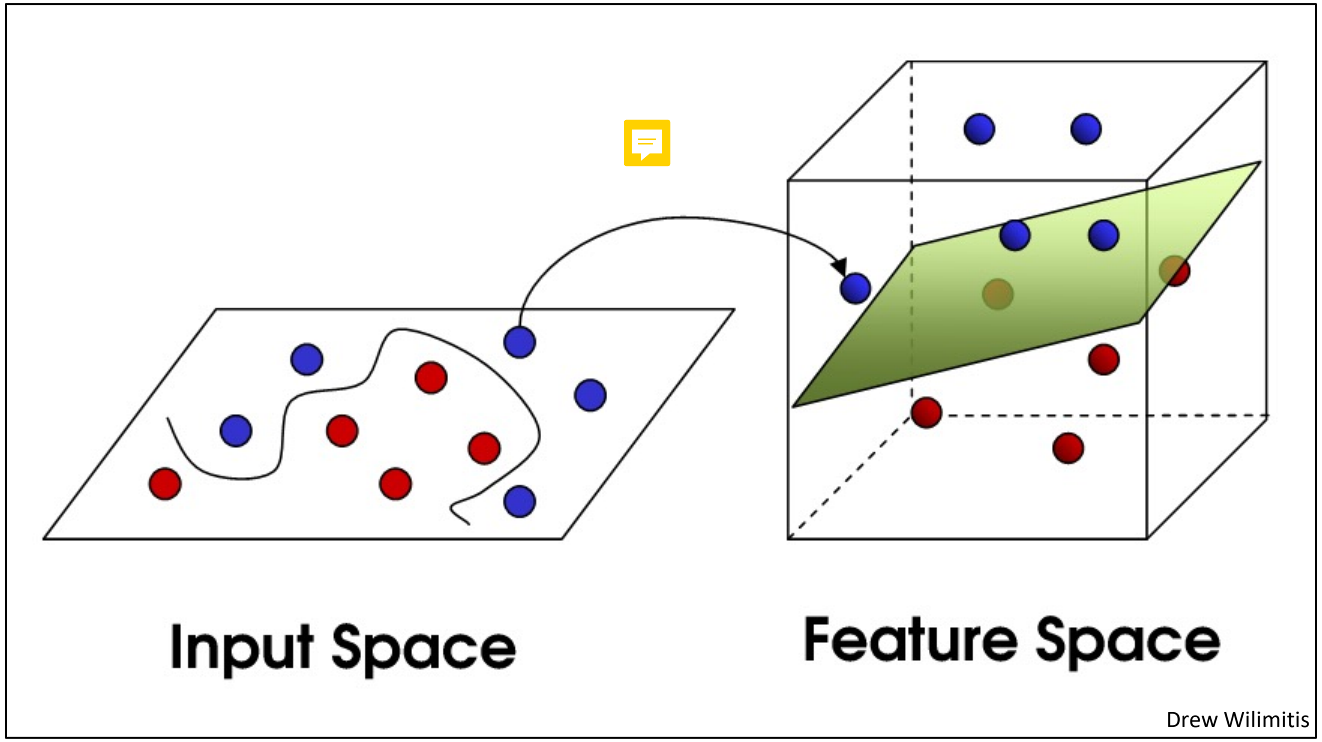




- Naïve option:
  - Add polynomial features to data
  - BUT produces too many features for high-degree polynomials

Non-Linear SVM: Kernel Trick

- Many non-linear datasets become linearly separable if projected to higher dimensional spaces



Non-Linear SVM: Kernel Trick

- Many non-linear datasets become linearly separable if projected to higher dimensional spaces

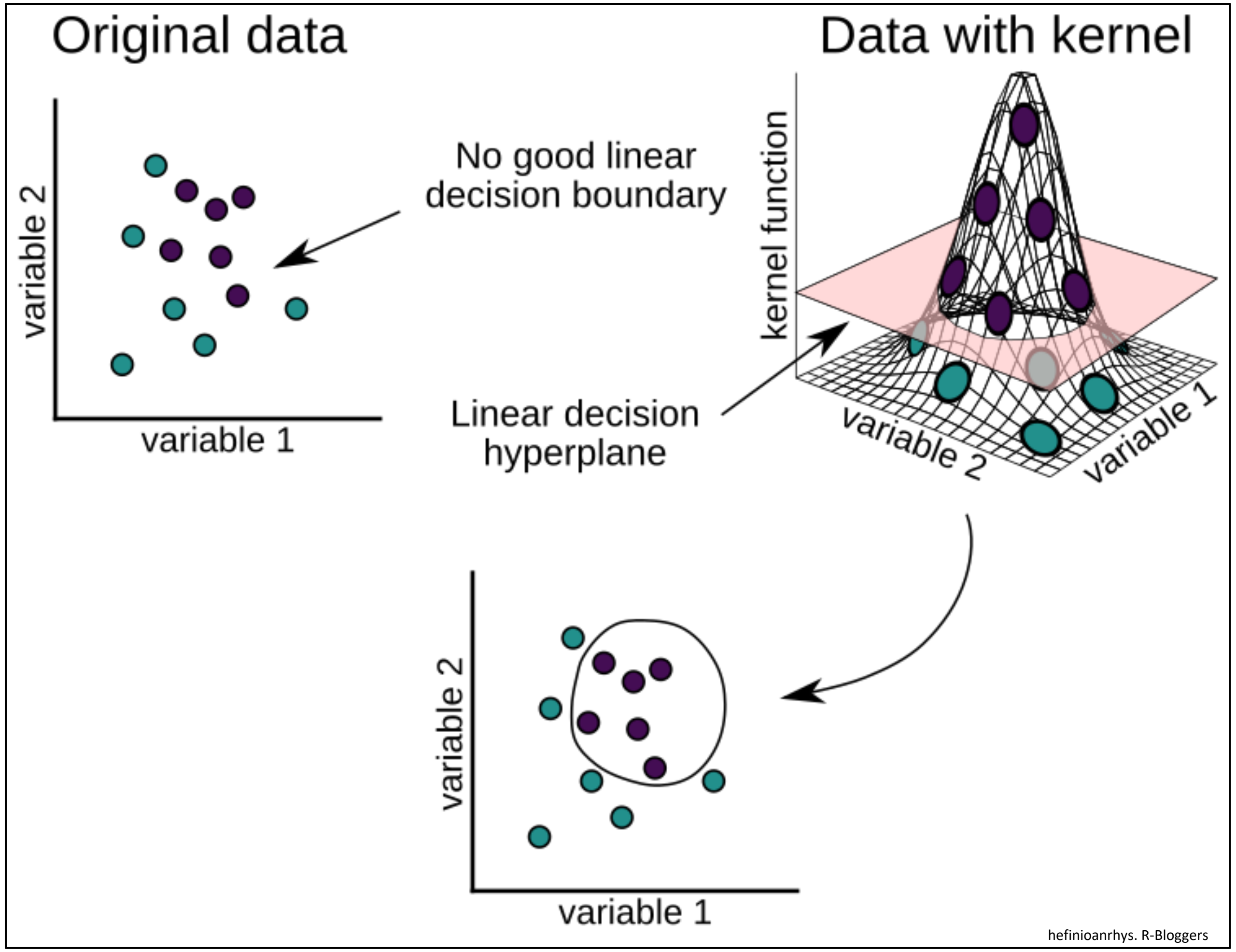
- BUT resulting projections often have too many features
    - Computationally infeasible training

Non-Linear SVM: **Kernel Trick**

- Re-derive SVM training minimization problem into **dual** format
  - See textbook for overview of math
  - ★ Only requires a **similarity metric** between **pairs of features,** not the features themselves

- A **kernel function** computes the similarity between vectors in a higher dimensional space…*without needing to project the vectors into that space!!!*
  - Plug-and-play kernel functions into linear SVM to train a non-linear model

Non-Linear SVM: **Kernel Trick**



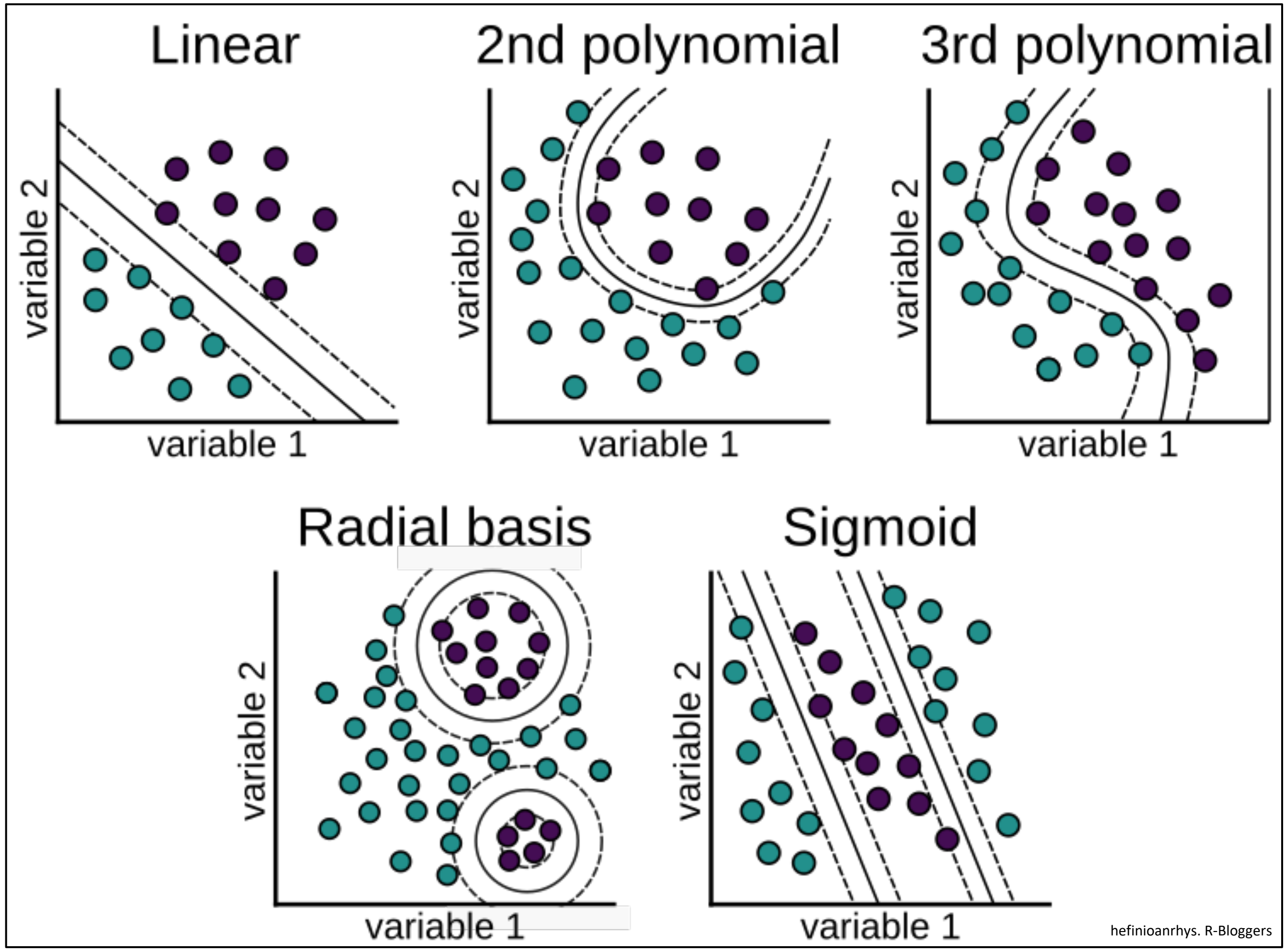
- d-degree polynomial kernel

$$K(\mathbf{x}_1, \mathbf{x}_2) = (\mathbf{x_1}^\intercal \mathbf{x_2})^d$$

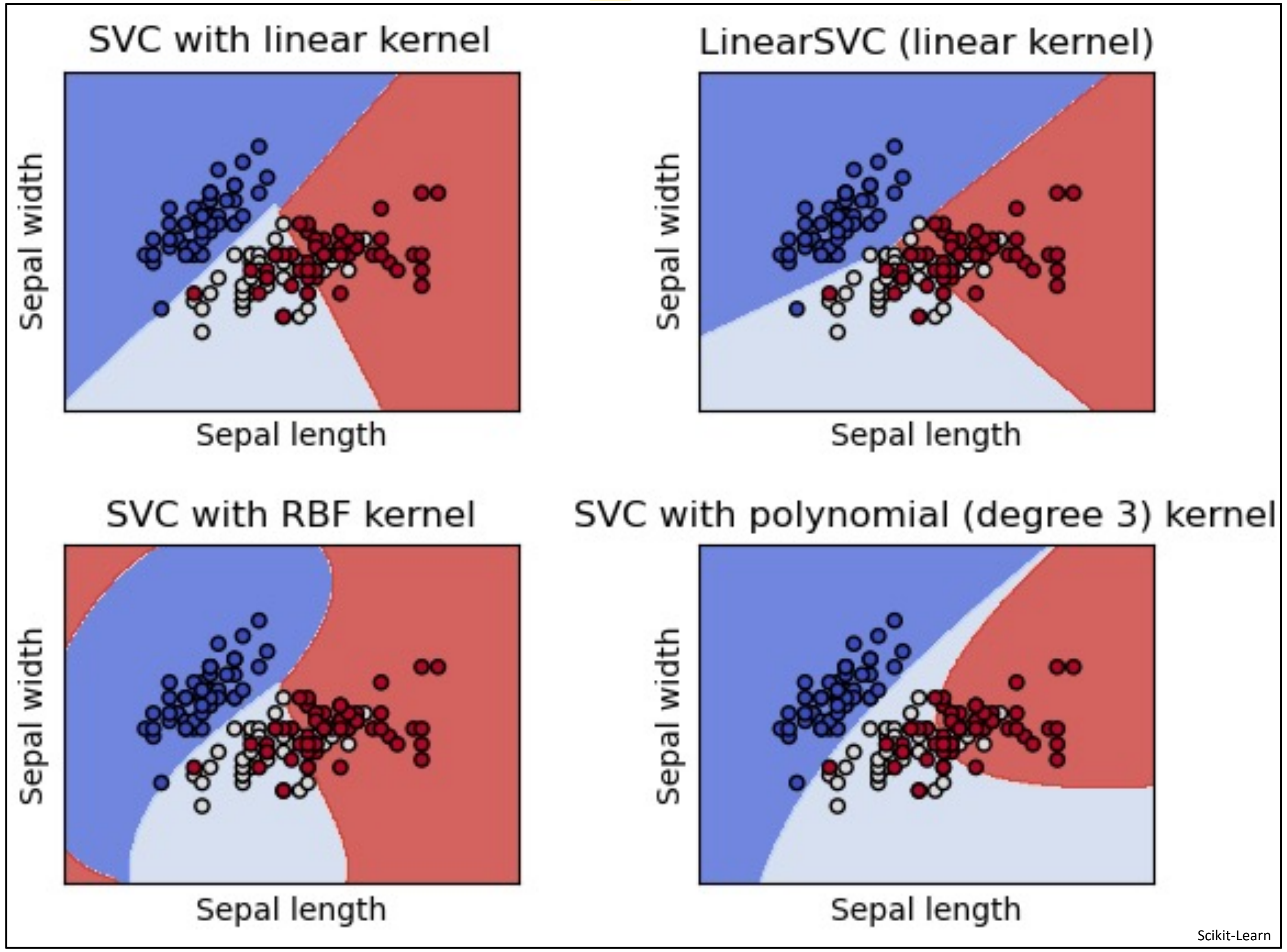
- Similarity metric between vectors with d-degree polynomial features

- No need to compute the features themselves!

Non-Linear SVM: Kernel Trick



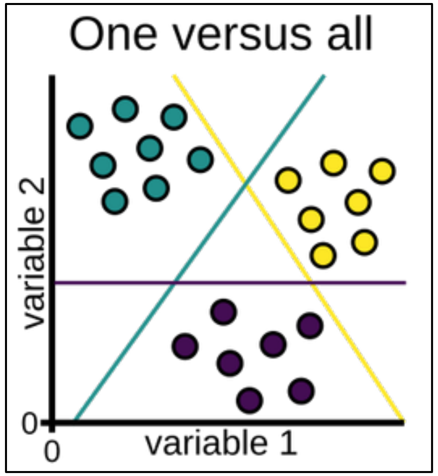
Non-Linear SVM: **Kernel Trick**

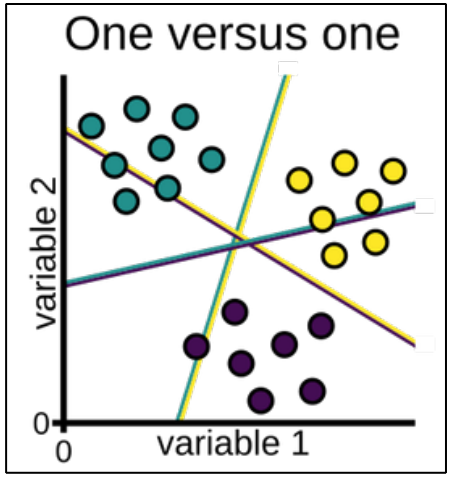


Multiclass SVM

Multiclass SVM

- One-versus-rest
  - For N classes, train N SVMs (each binary…one class versus all others)
- One-versus-one
  - For N classes, train $\binom{N}{2}$ classifiers (all pairwise combinations)
- Both
  - Predict new data using all models
  - Keep prediction that is *furthest from its decision line*





hefinioanrhys, R-Bloggers

Practical Use of SVMs

★ "Small" datasets

- Not enough data to train a neural network
- 1000s to 100,000s of examples
  - Very approximate limit → depends heavily on learning task

- Support vector **regression**
  - Fit data inside margin instead of outside margin
  - See textbook for more

Programming Practice

SVM.ipynb

Questions?