



# CS 6375

## Support Vector Machines

Rishabh Iyer  
University of Texas at Dallas

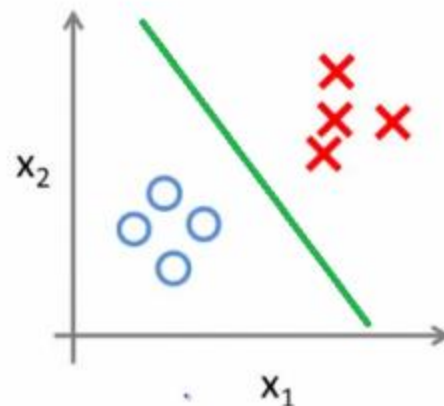
# Recap: Classification



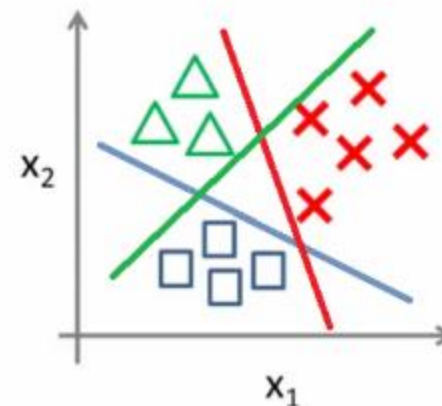
## Classification vs Regression

- Input: pairs of points  $(x^{(1)}, y^{(1)}), \dots, (x^{(M)}, y^{(M)})$  with  $x^{(m)} \in \mathbb{R}^n$
- $y^{(m)} \in [0, k - 1]$
- If  $k = 2$ , we get Binary classification

Binary classification:



Multi-class classification:

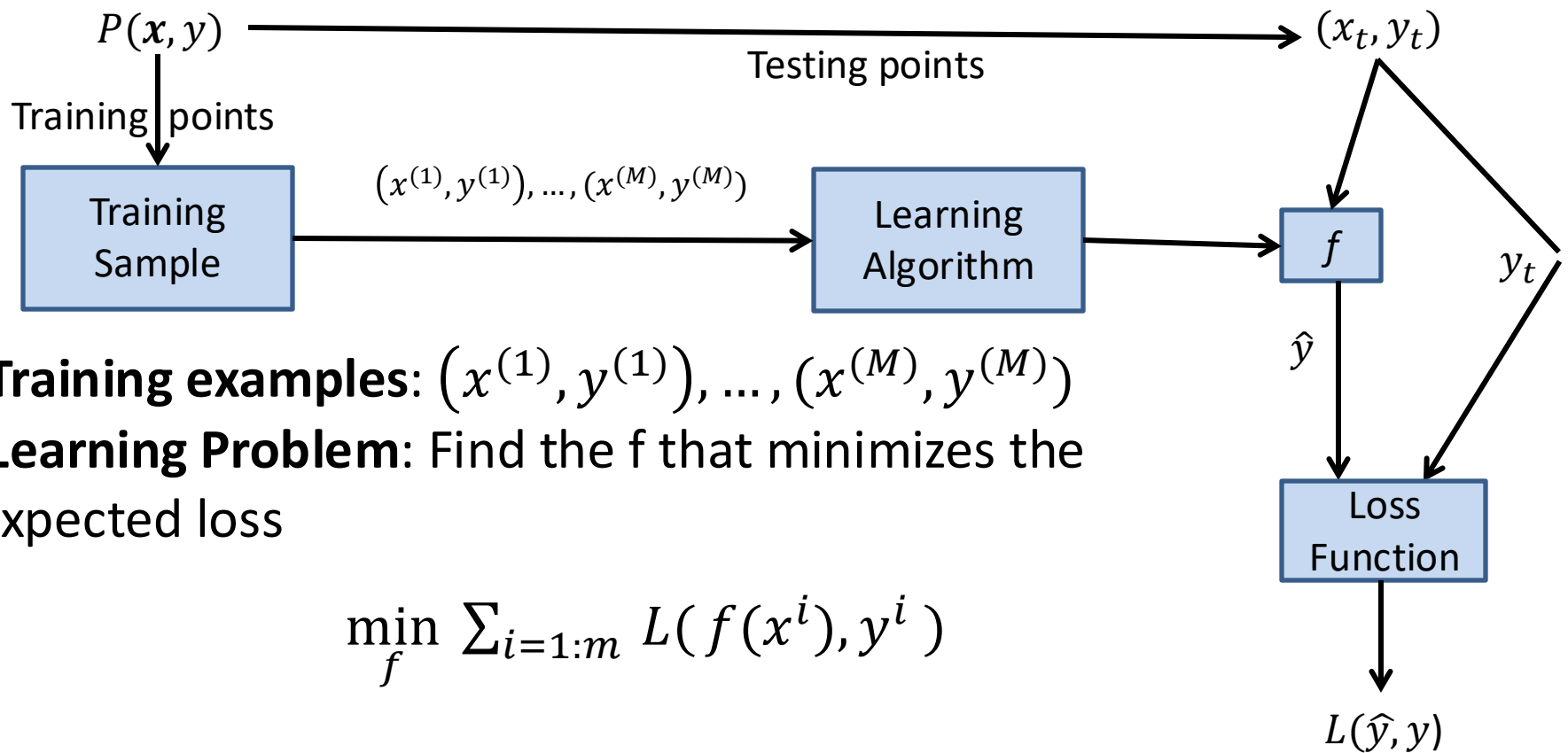


# Recap: Hypothesis Space



- **Hypothesis space**: set of allowable functions  $f: X \rightarrow Y$
- Goal: find the “best” element of the hypothesis space
  - How do we measure the quality of  $f$ ?

# Recap: Supervised Learning Workflow



- **Training examples:**  $(x^{(1)}, y^{(1)}), \dots, (x^{(M)}, y^{(M)})$
- **Learning Problem:** Find the  $f$  that minimizes the expected loss

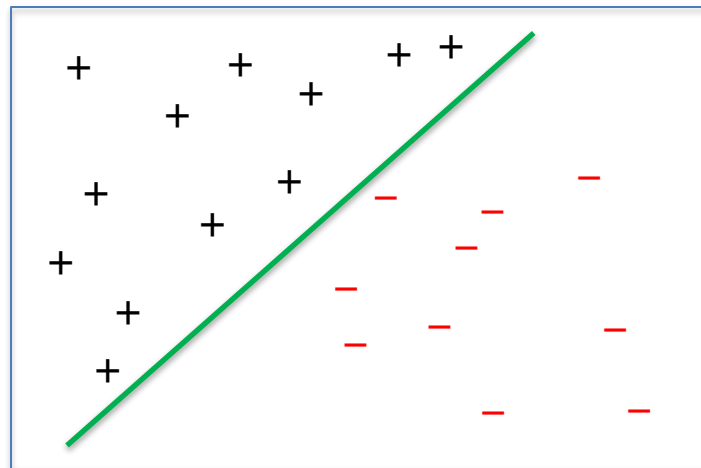
$$\min_f \sum_{i=1:m} L(f(x^i), y^i)$$

- **Testing:** Given a new point  $(x_t, y_t)$  drawn from  $P$ , the classifier is given  $x$  and predicts  $\hat{y}_t = f(x_t)$
- **Evaluation:** Measure the error  $Err(\hat{y}_t, y_t)$  – often same as  $L$

# Recap: Binary Classification



- Input  $(x^{(1)}, y^{(1)}), \dots, (x^{(M)}, y^{(M)})$  with  $x^{(m)} \in \mathbb{R}^n$  and  $y^{(m)} \in \{-1, +1\}$
- We can think of the observations as points in  $\mathbb{R}^n$  with an associated sign (either +/- corresponding to 0/1)
- An example with  $n = 2$



In this case, we say  
that the  
observations are  
**linearly separable**

# 0/1 Loss Vs Perceptron Loss

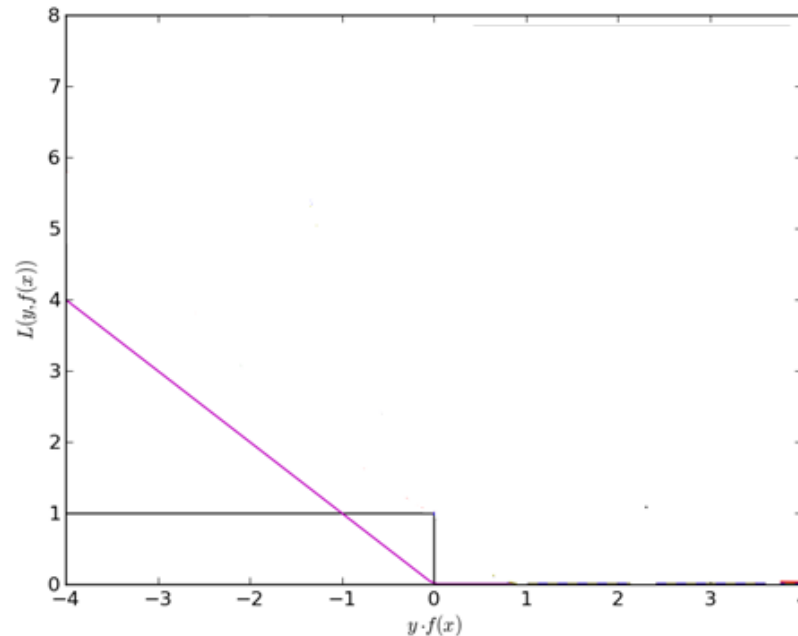


- Zero/One Loss which counts the number of mis-classifications:

$$\text{zero/one loss} = \frac{1}{2} \sum_m |y^{(m)} - \text{sign}(w^T x^{(m)} + b)|$$

- Perceptron Loss:

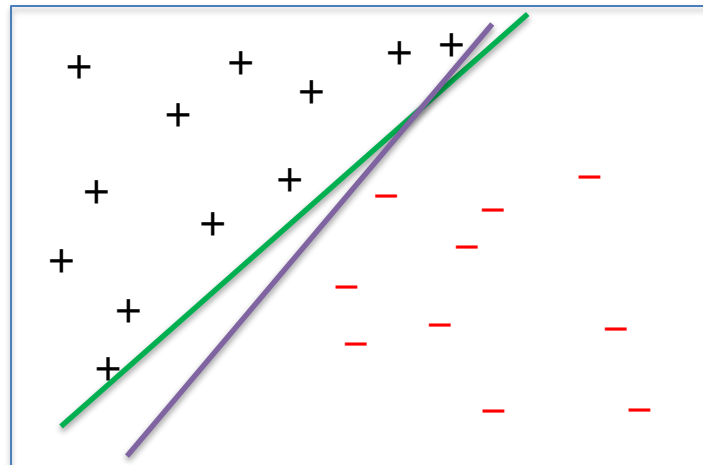
$$\text{perceptron loss} = \sum_m \max\{0, -y^{(m)}(w^T x^{(m)} + b)\}$$



# Perceptron Drawbacks



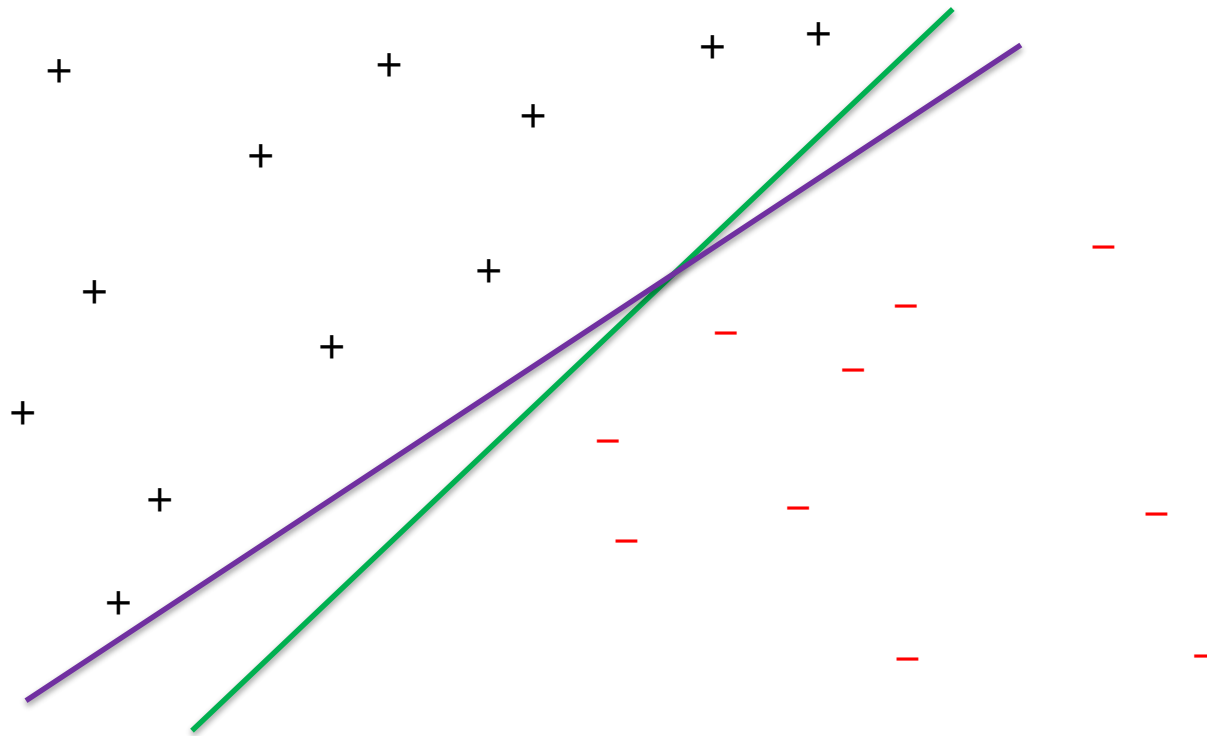
- No convergence guarantees if the observations are not linearly separable
- Can overfit
  - There can be a number of perfect classifiers, but the perceptron algorithm doesn't have any mechanism for choosing between them



# Support Vector Machines



- How can we decide between perfect classifiers?

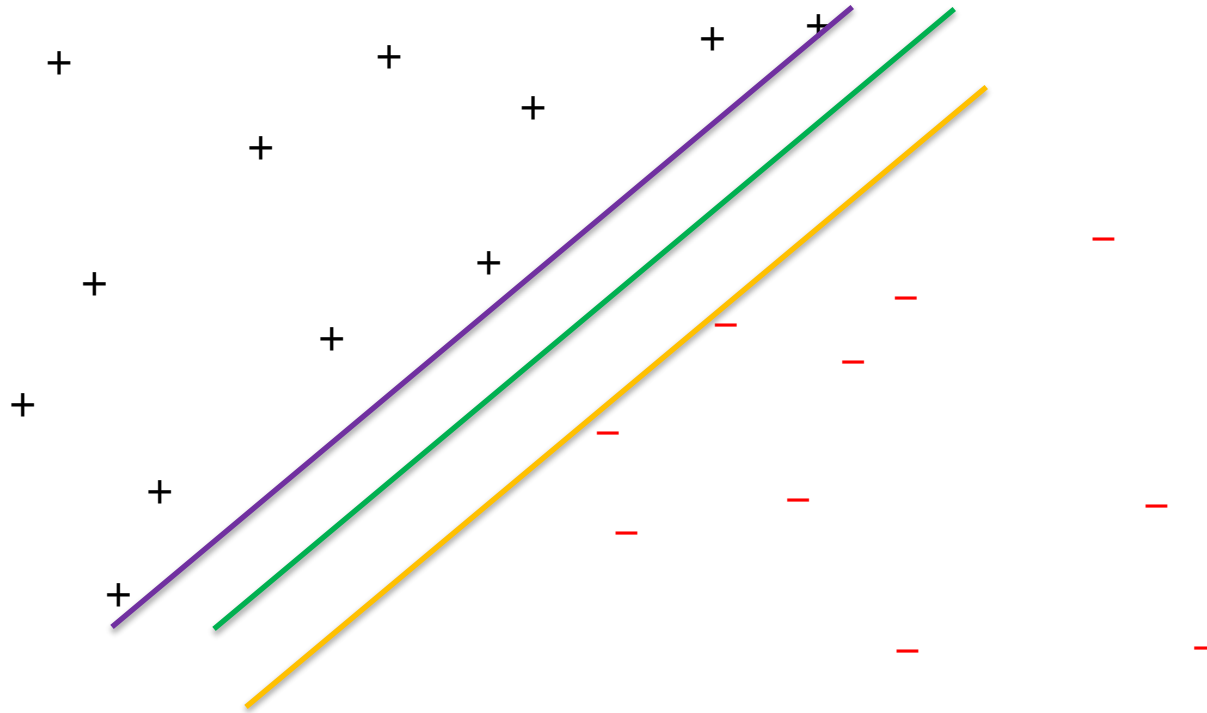




# Support Vector Machines



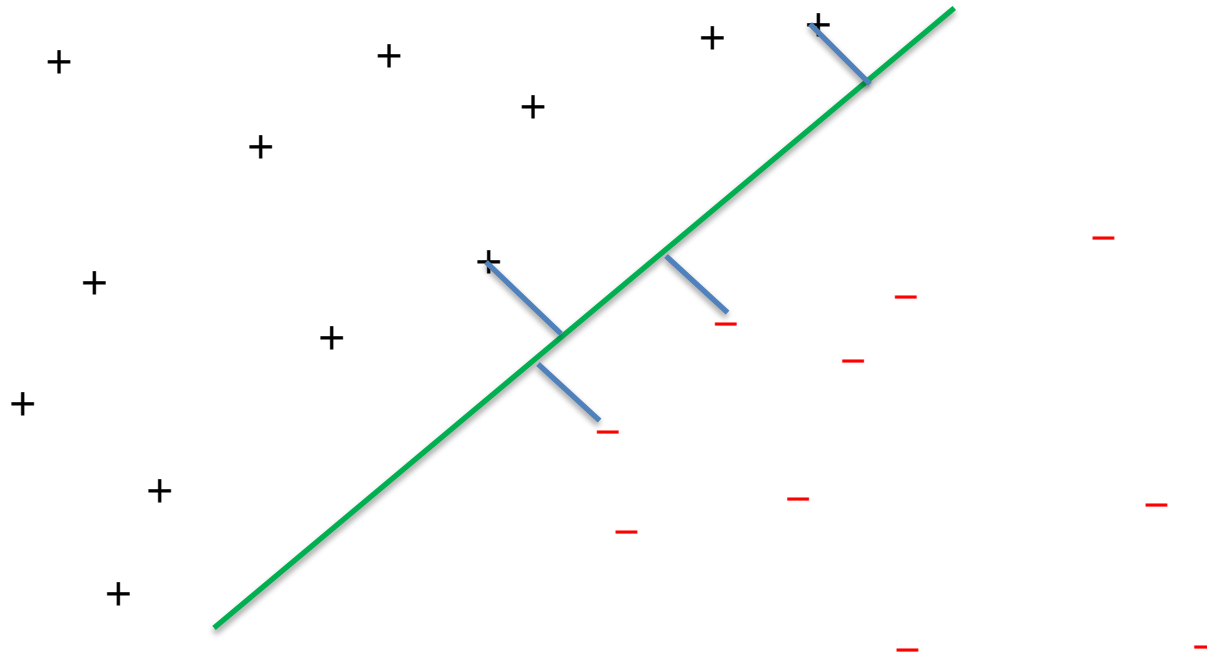
- How can we decide between perfect classifiers?



# Support Vector Machines



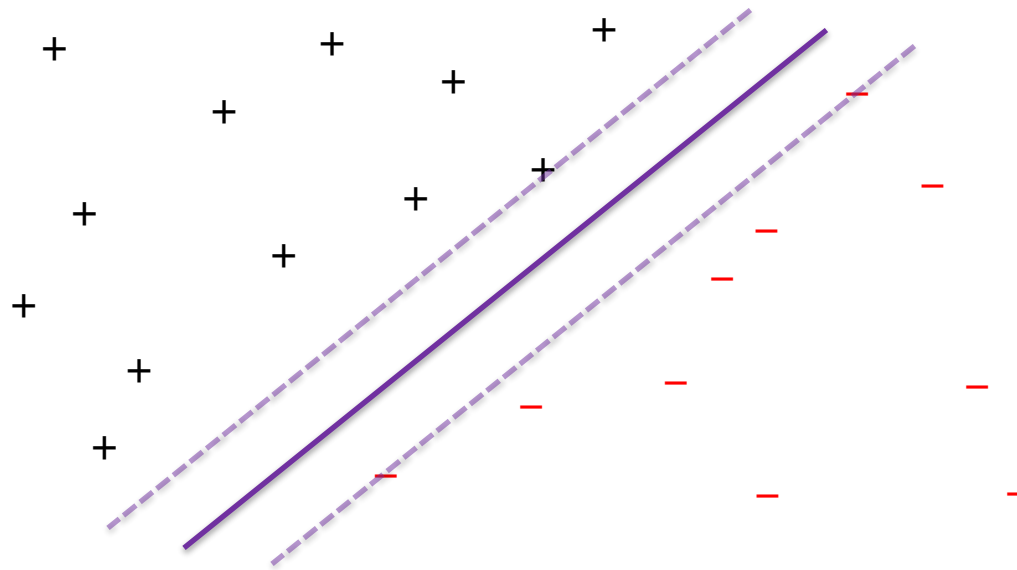
- Define the **margin** to be the distance of the closest data point to the classifier



# Support Vector Machines

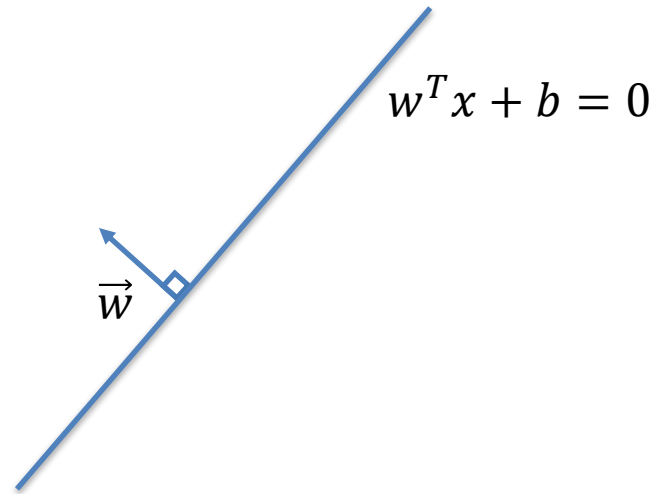


- Support vector machines (SVMs)

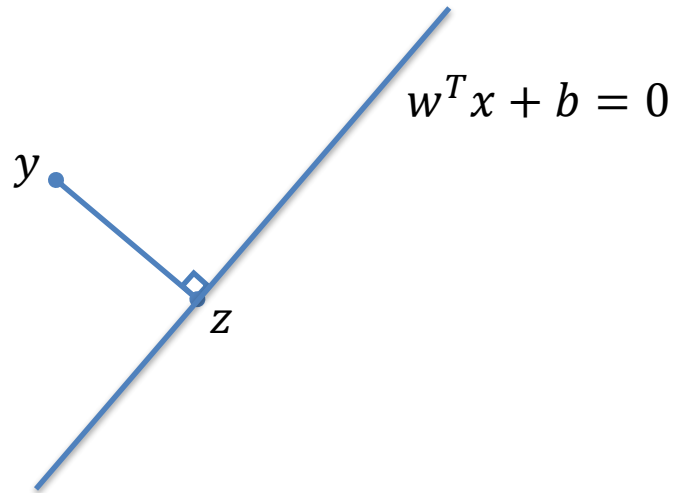


- Choose the classifier with the largest margin
  - Has good practical and theoretical performance

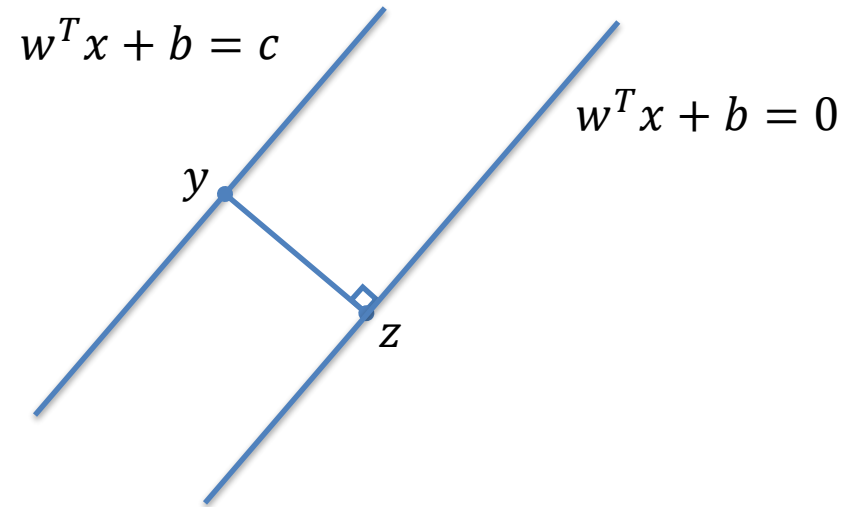
# Some Geometry



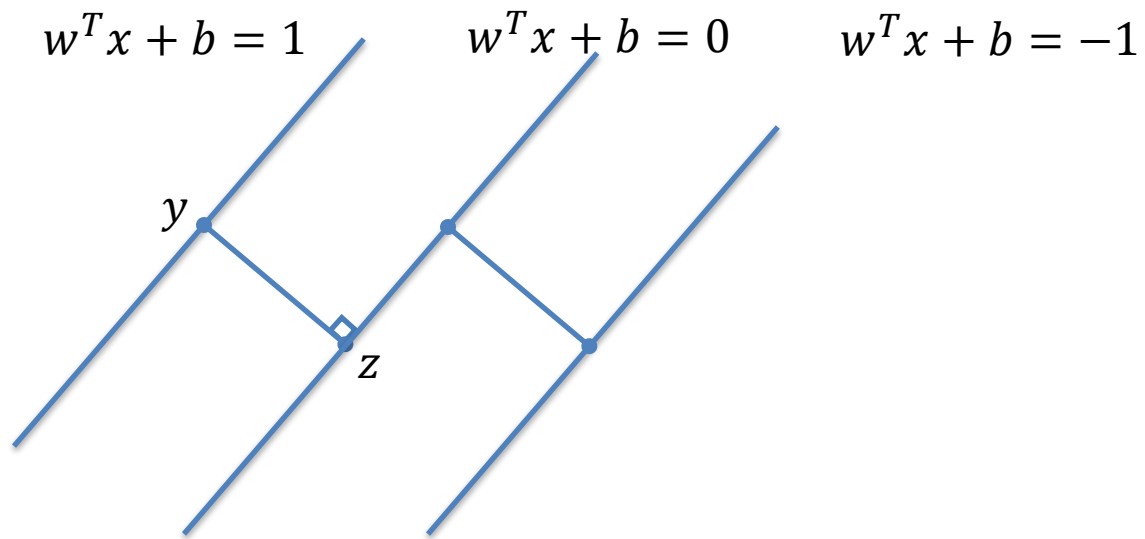
# Some Geometry



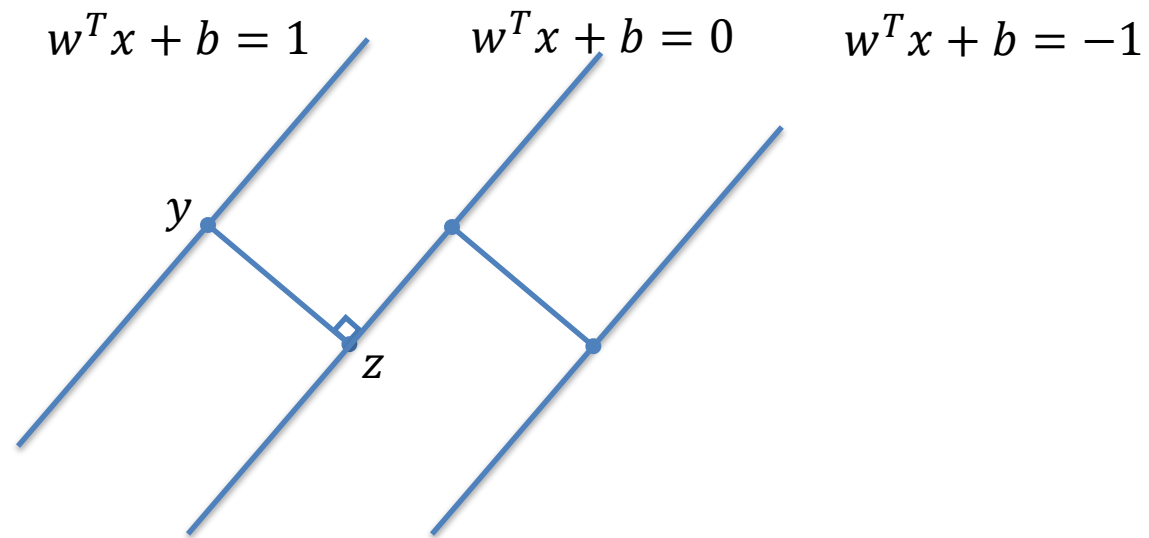
# Scale Invariance



# Constraints



# What is the Margin?





- This analysis yields the following optimization problem

$$\max_{w,b} \frac{1}{\|w\|}$$

such that

$$y^{(i)}(w^T x^{(i)} + b) \geq 1, \text{ for all } i$$

- Or, equivalently,

$$\min_{w,b} \|w\|^2$$

such that

$$y^{(i)}(w^T x^{(i)} + b) \geq 1, \text{ for all } i$$

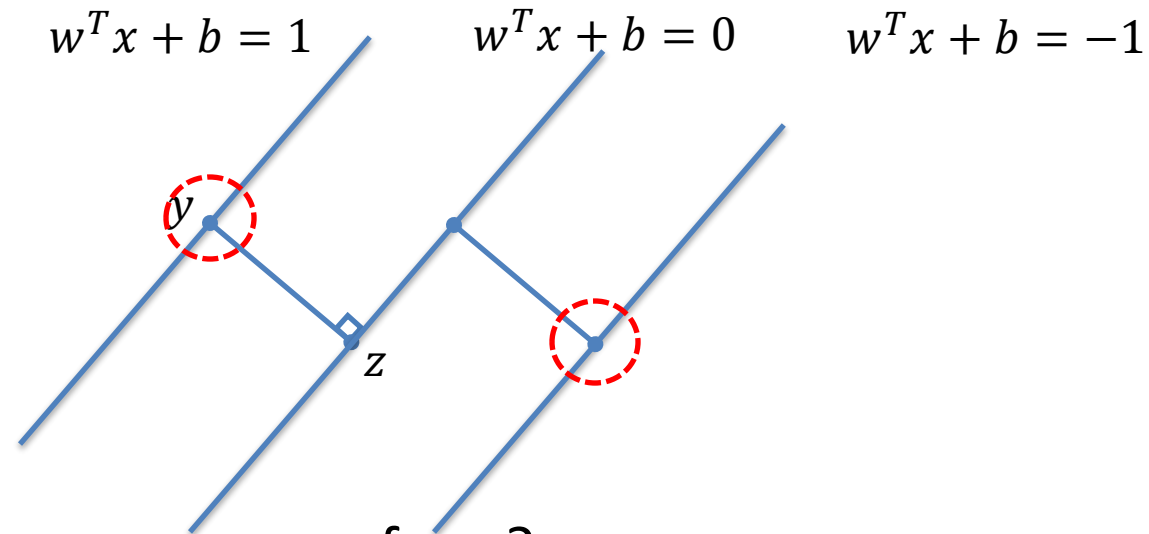
$$\min_{w,b} \|w\|^2$$

such that

$$y^{(i)}(w^T x^{(i)} + b) \geq 1, \text{ for all } i$$

- This is a standard quadratic programming problem
  - Falls into the class of **convex optimization problems**
  - Can be solved with many specialized optimization tools (e.g., `quadprog()` in MATLAB)

# Support Vectors

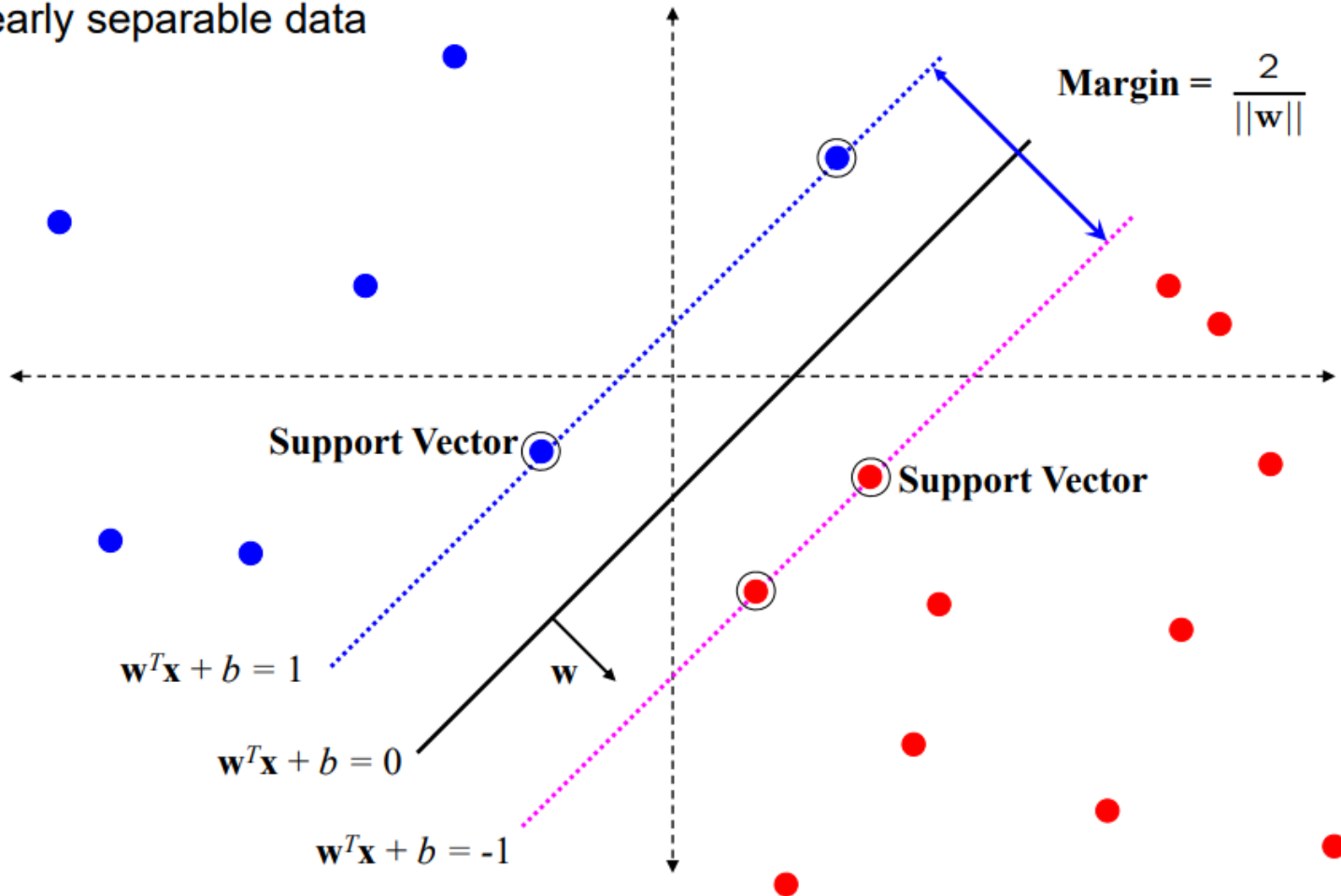


- Where does the name come from?
  - The set of all data points such that  $y^{(i)}(w^T x^{(i)} + b) = 1$  are called **support vectors**
  - The SVM classifier is completely determined by the support vectors (you could delete the rest of the data and get the same answer)

# Putting Everything Together



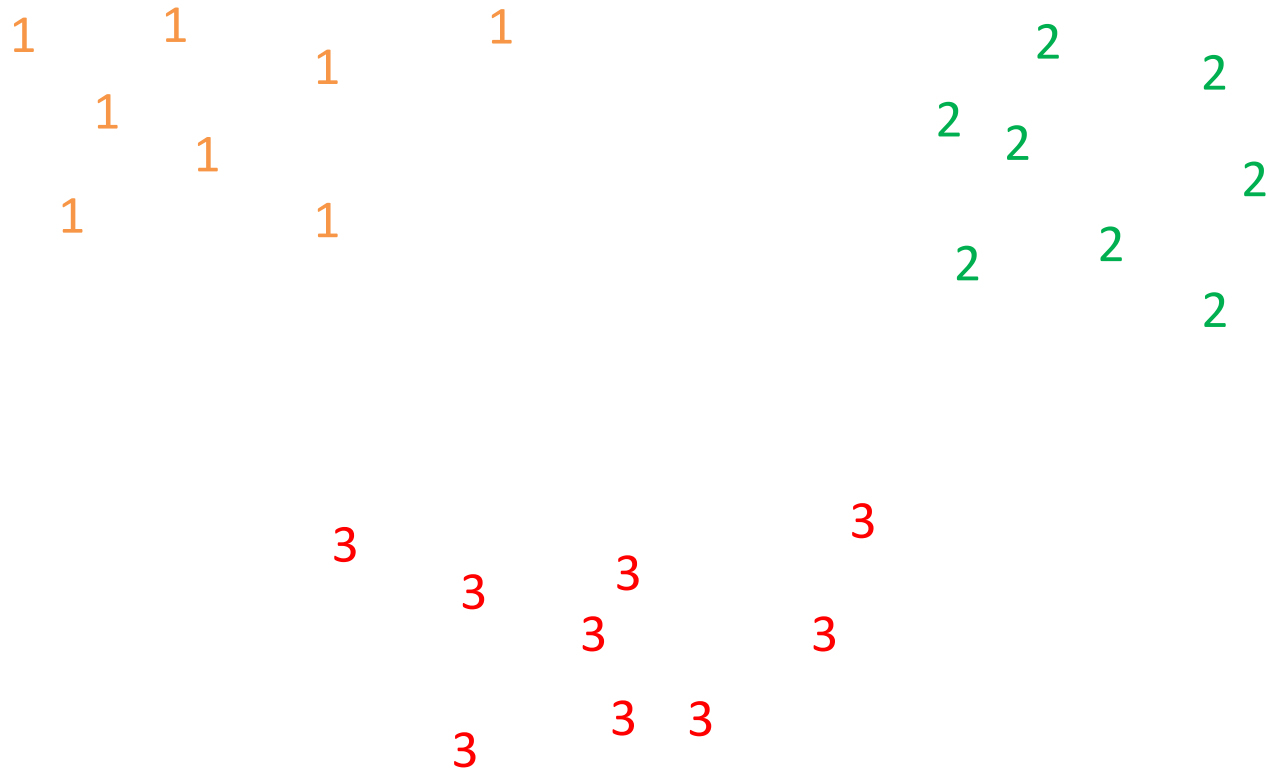
linearly separable data



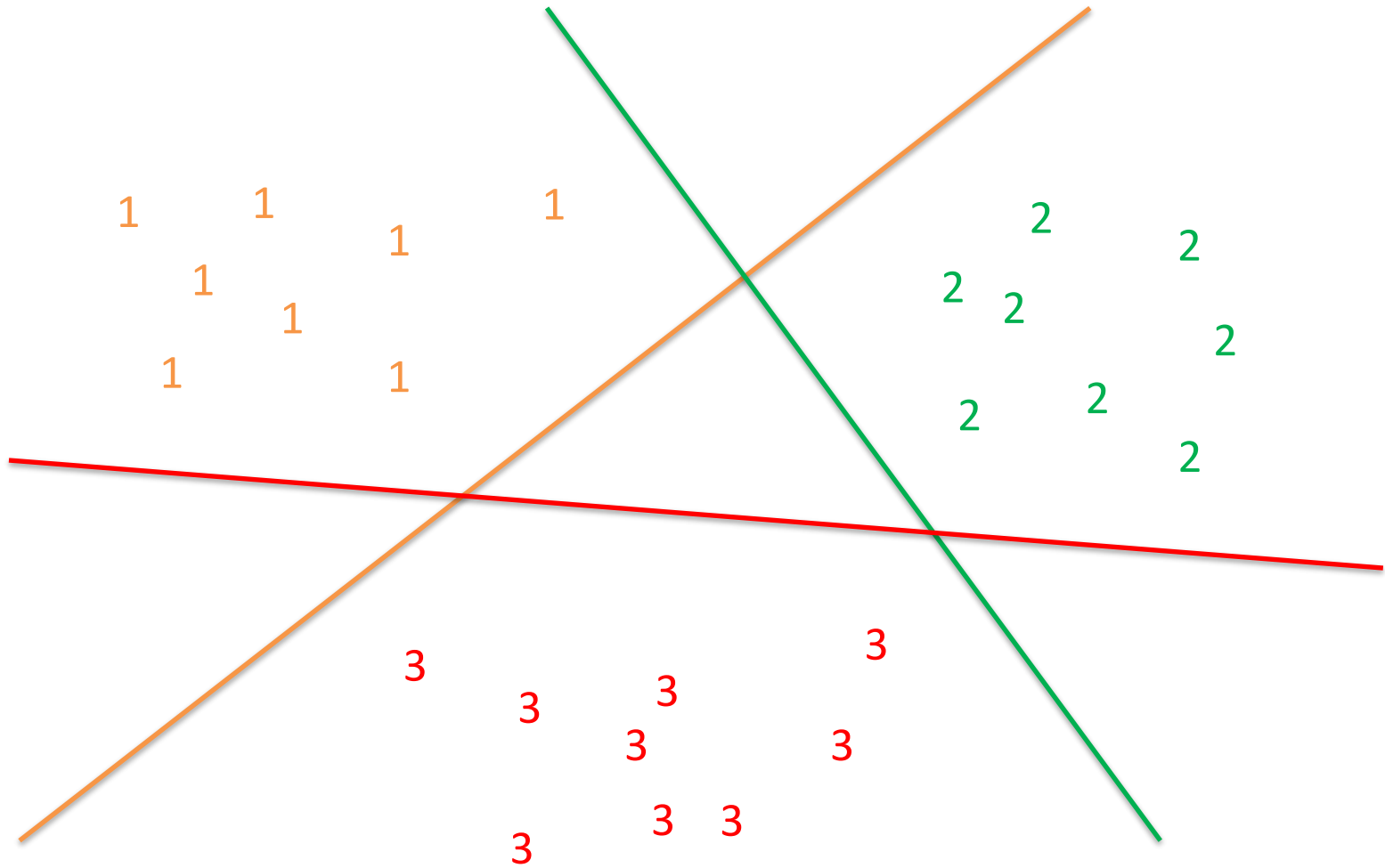
- What if the data isn't linearly separable?
- What if we want to do more than just binary classification (i.e., if  $y \in \{1,2,3\}$ )?

- What if the data isn't linearly separable?
  - Higher order (polynomial features)
  - Relax the constraints (coming soon)
- What if we want to do more than just binary classification (i.e., if  $y \in \{1,2,3\}$ )?

# Multiclass Classification

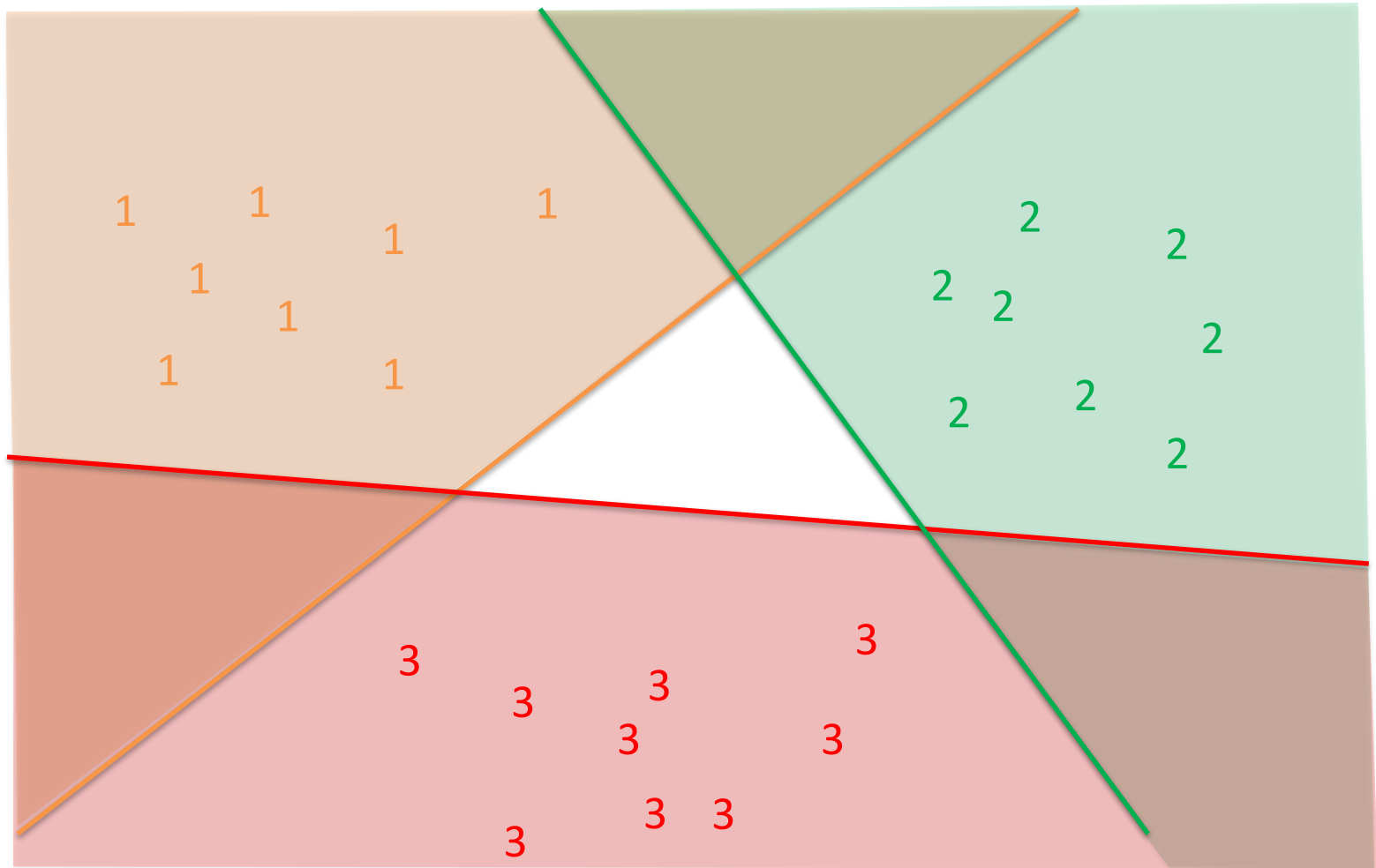


# One-Versus-All SVMs





# One-Versus-All SVMs



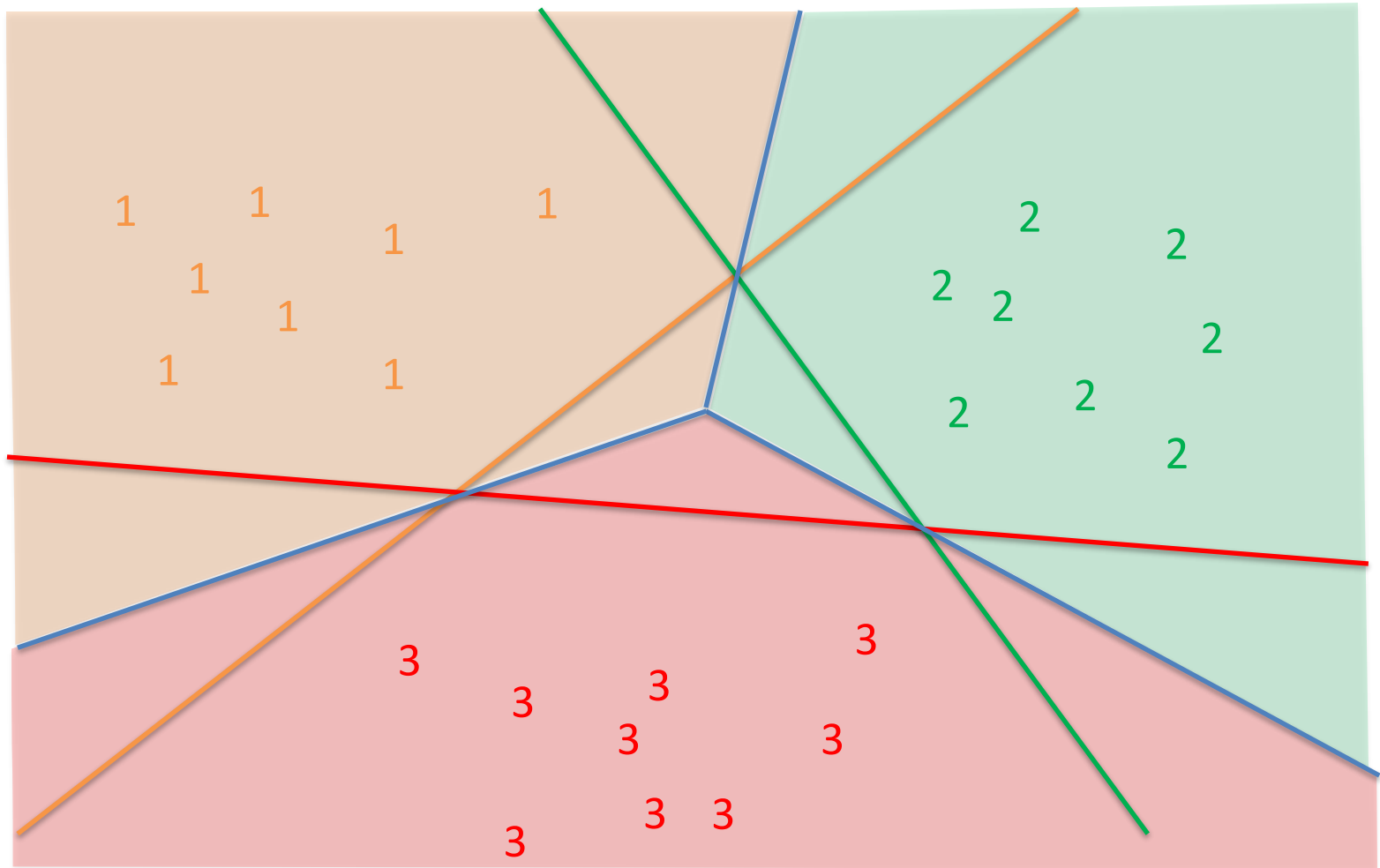
Regions correctly classified by exactly one classifier

- Compute a classifier for each label versus the remaining labels (i.e., an SVM with the selected label as plus and the remaining labels changed to minuses)
- Let  $f^k(x) = w^{(k)T}x + b^{(k)}$  be the classifier for the  $k^{th}$  label
- For a new datapoint  $x$ , classify it as

$$k' \in \operatorname{argmax}_k f^k(x)$$

- Drawbacks:
  - If there are  $L$  possible labels, requires learning  $L$  classifiers over the entire data set

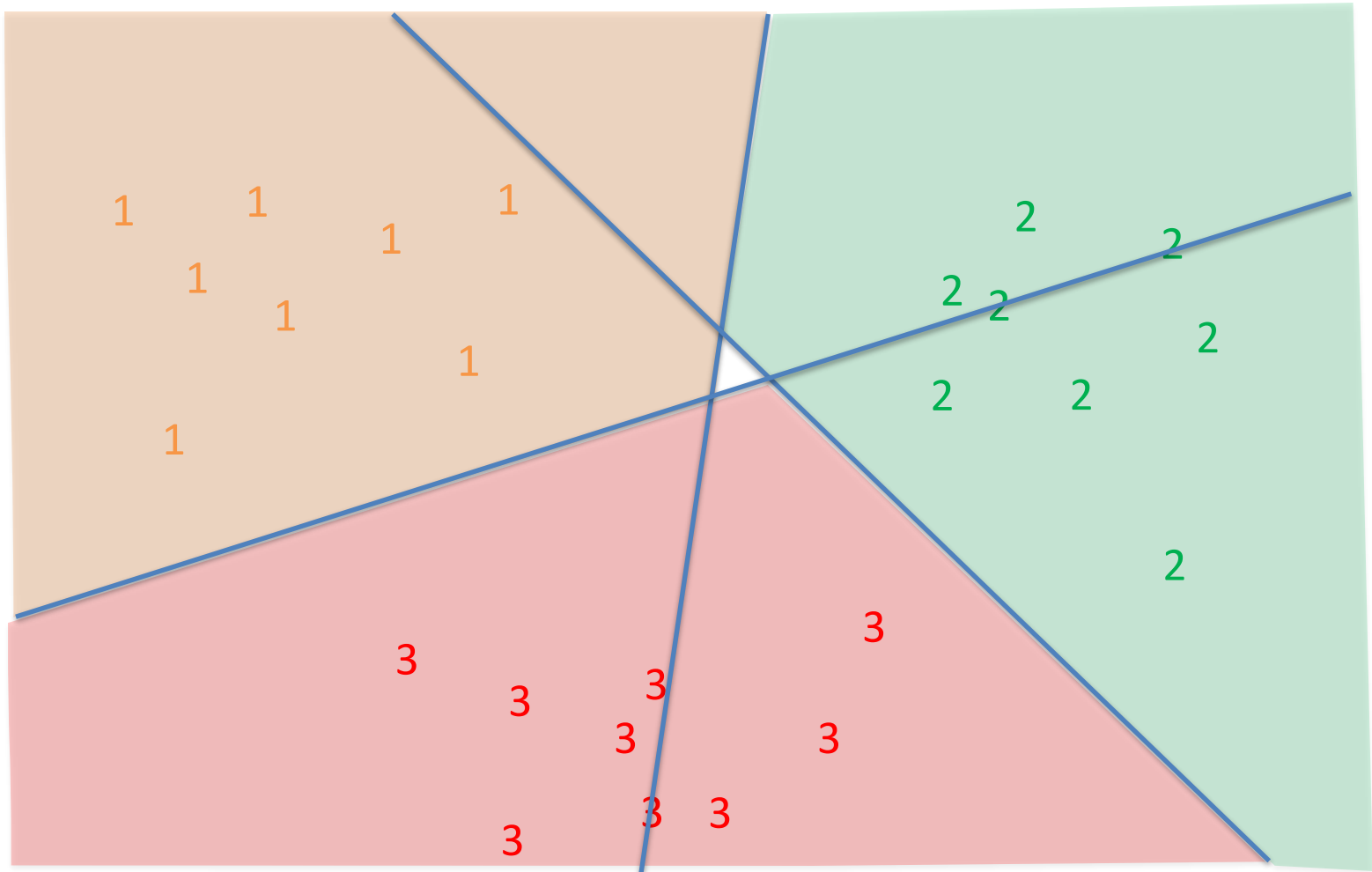
# One-Versus-All SVMs



Regions in which points are classified by highest value of  $w^T x + b$

- Alternative strategy is to construct a classifier for all possible pairs of labels
- Given a new data point, can classify it by majority vote (i.e., find the most common label among all of the possible classifiers)
- If there are  $L$  labels, requires computing  $\binom{L}{2}$  different classifiers each of which uses only a fraction of the data
- Drawbacks: Can overfit if some pairs of labels do not have a significant amount of data (plus it can be computationally expensive)

# One-Versus-One SVMs



Regions determined by majority vote over the classifiers