CS 6375
Support Vector Machines

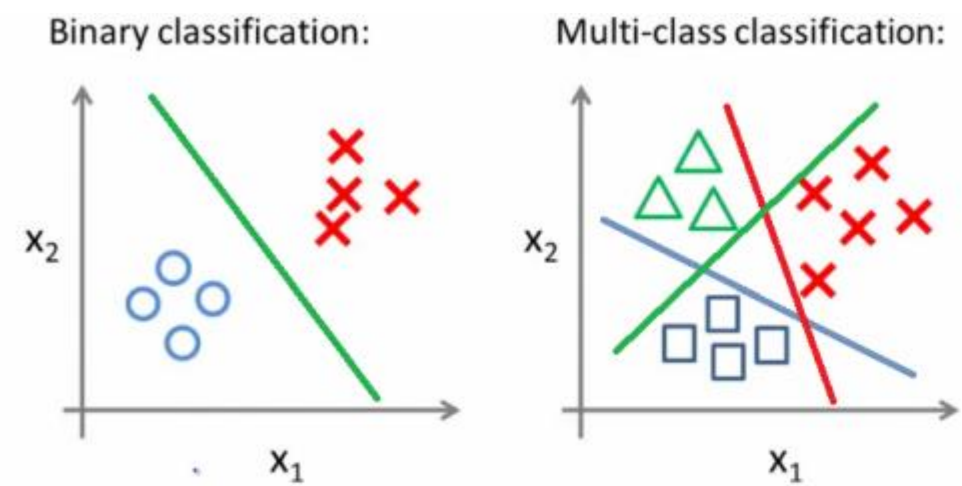
Rishabh Iyer
University of Texas at Dallas

Slides adapted from Nick Rouzzi and Andrew Zisserman

Recap: Classification

Classification vs Regression

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



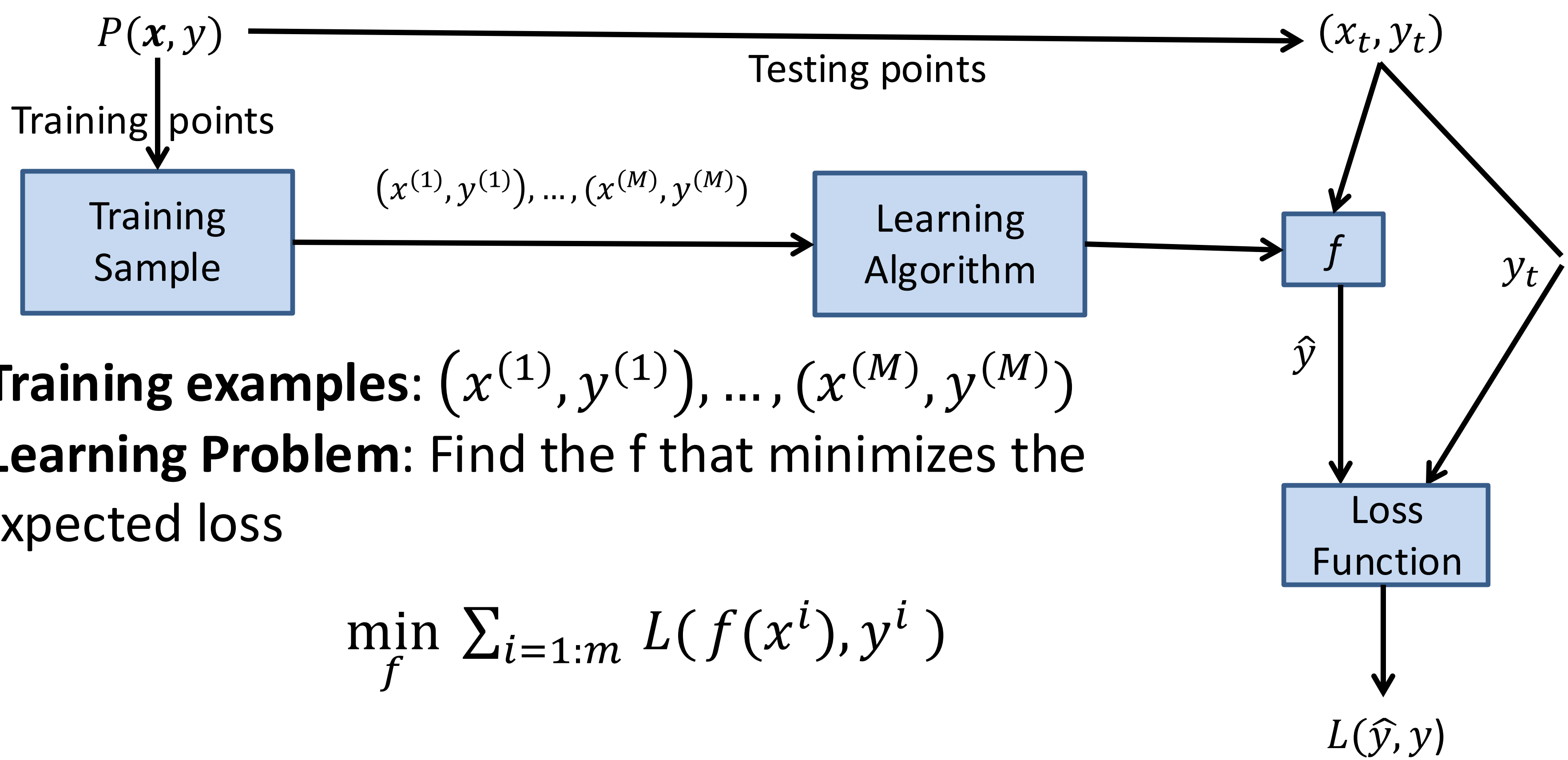
Recap: Hypothesis Space

- Hypothesis space: set of allowable functions $f: X \to Y$

- Goal: find the "best" element of the hypothesis space

  - How do we measure the quality of f?

Recap: Supervised Learning Workflow



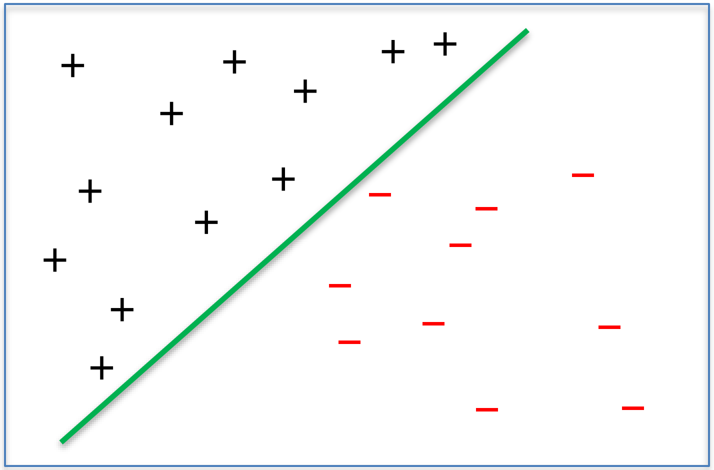
- **Training examples**: $(x^{(1)}, y^{(1)}), \ldots, (x^{(M)}, y^{(M)})$
- **Learning Problem**: Find the f that minimizes the expected loss

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

- **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 $(x^{(1)}, y^{(1)}), \ldots, (x^{(M)}, y^{(M)})$ 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

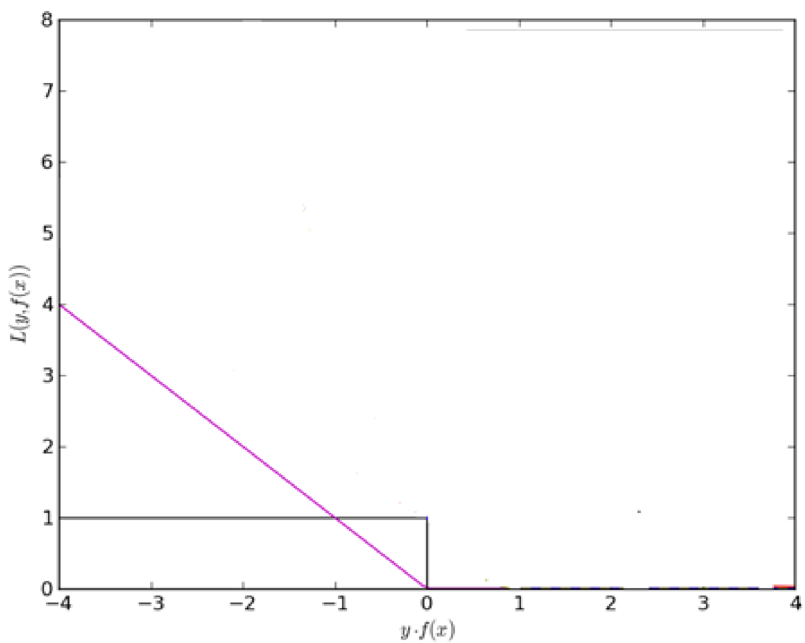
0/1 Loss Vs Perceptron Loss

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

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

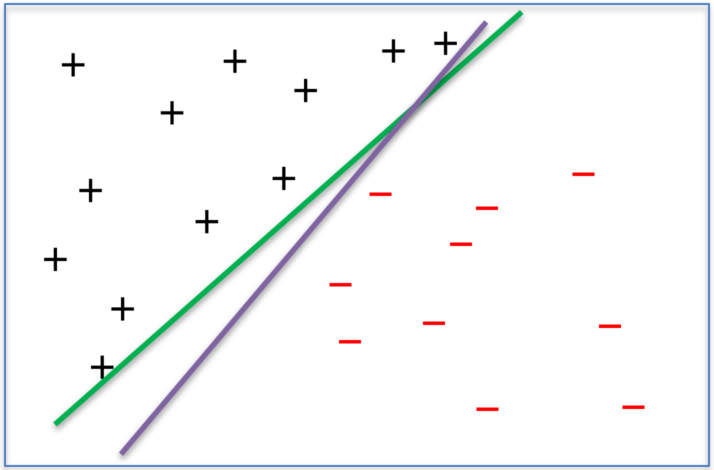
- Perceptron Loss:

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



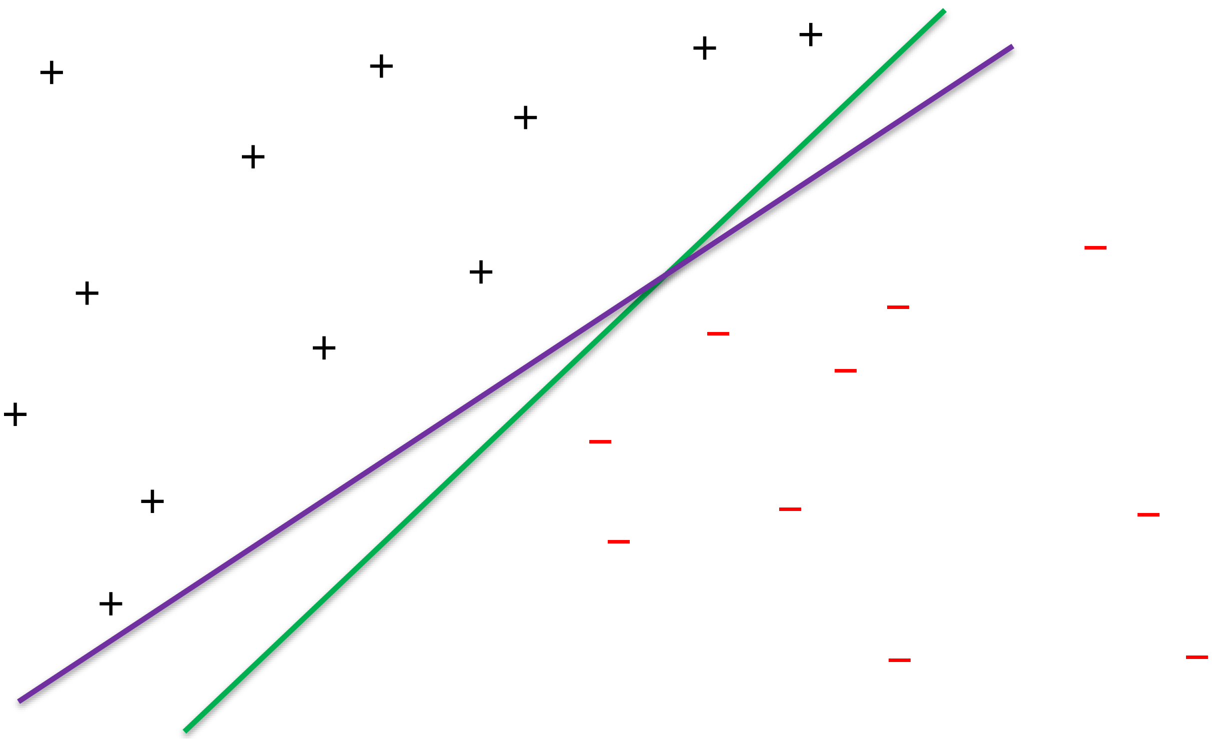
Perceptron Drawbacks

- 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



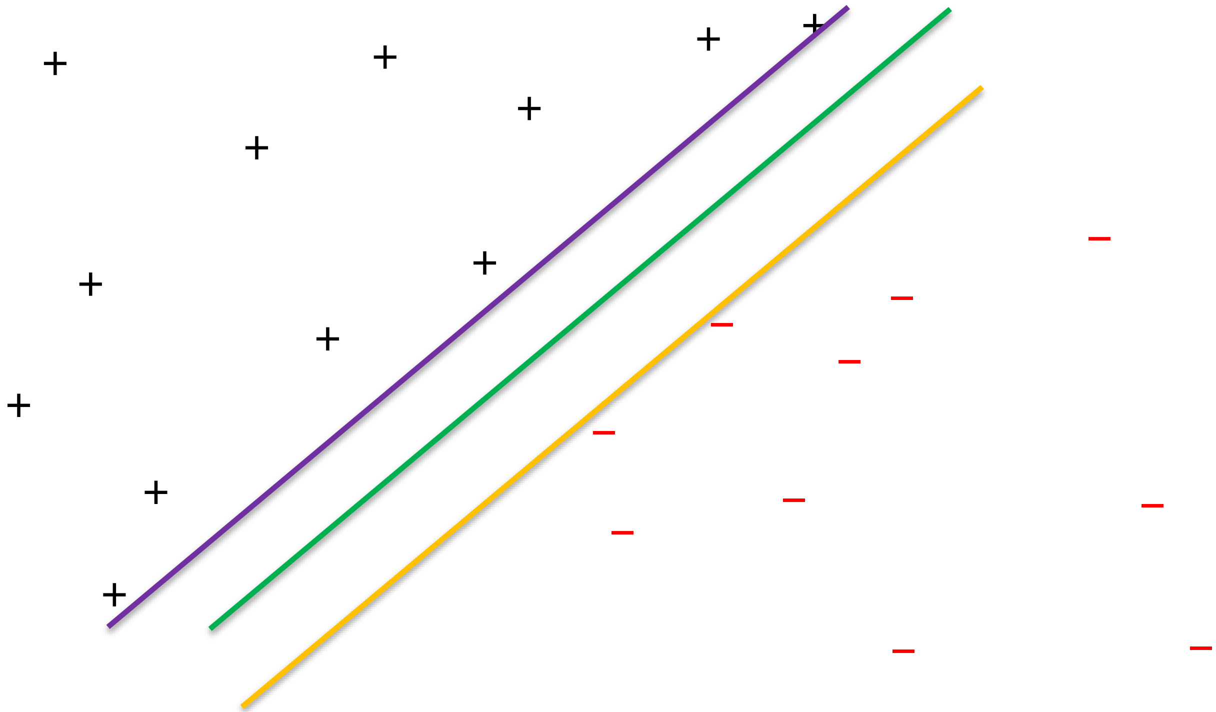
Support Vector Machines

- How can we decide between perfect classifiers?



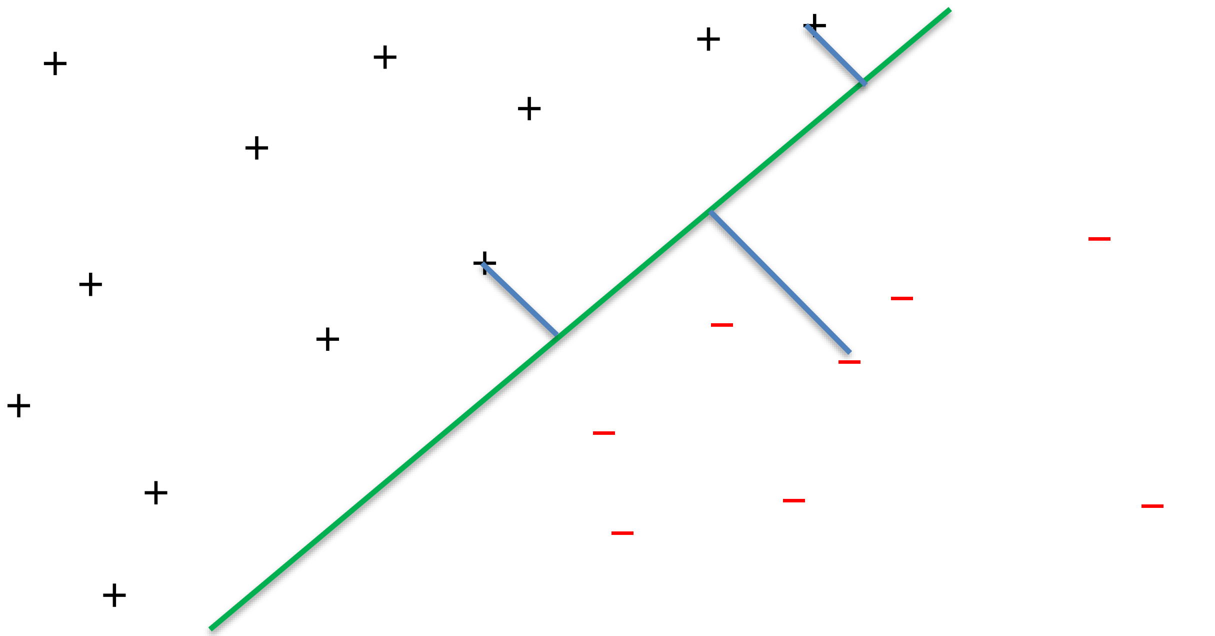
Support Vector Machines

- How can we decide between perfect classifiers?



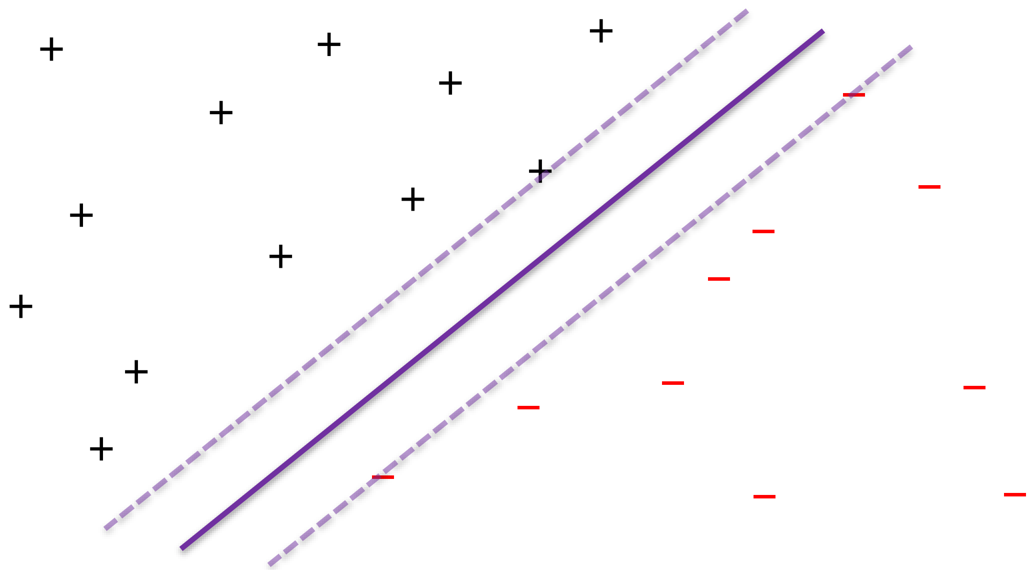
Support Vector Machines

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



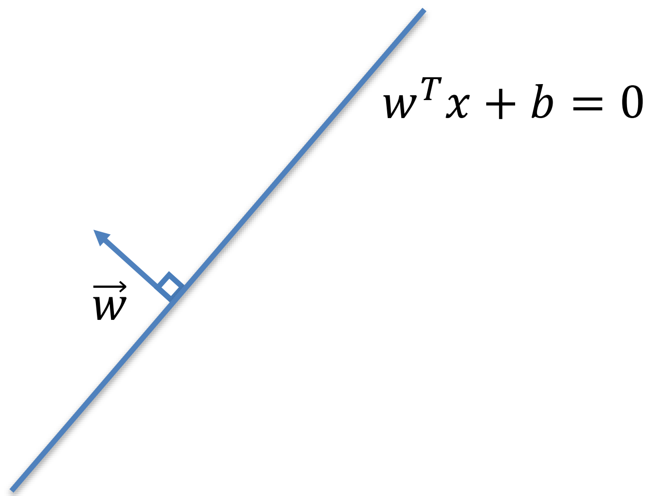
Support Vector Machines

- Support vector machines (SVMs)



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

Some Geometry



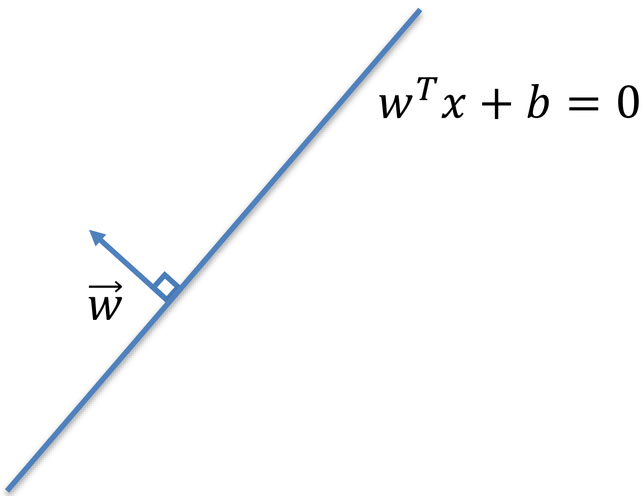
- In n dimensions, a hyperplane is a solution to the equation

$$w^T x + b = 0$$

with $w \in \mathbb{R}^n, b \in \mathbb{R}$

- The vector w is sometimes called the normal vector of the hyperplane

Some Geometry



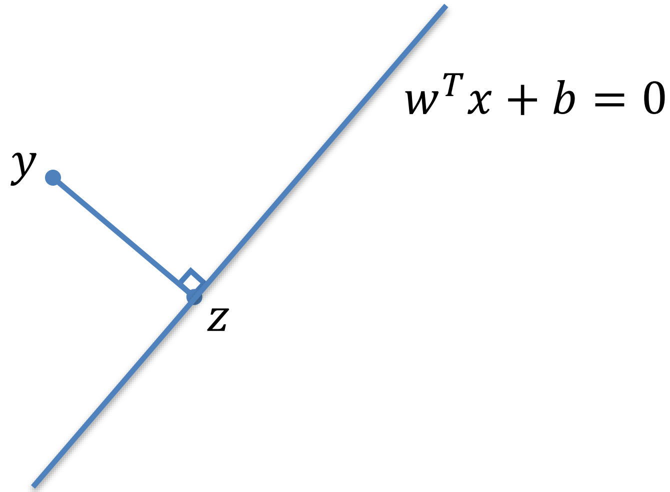
- In n dimensions, a hyperplane is a solution to the equation

$$w^T x + b = 0$$

- Note that this equation is scale invariant for any scalar c

$$c \cdot (w^T x + b) = 0$$

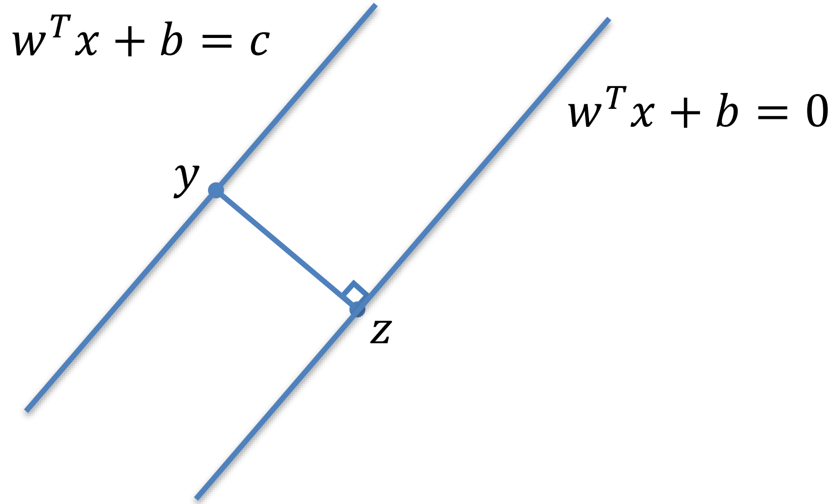
Some Geometry



- The distance between a point y and a hyperplane $w^T + b = 0$ is the length of the segment perpendicular to the line to the point y
- The vector from y to z is given by

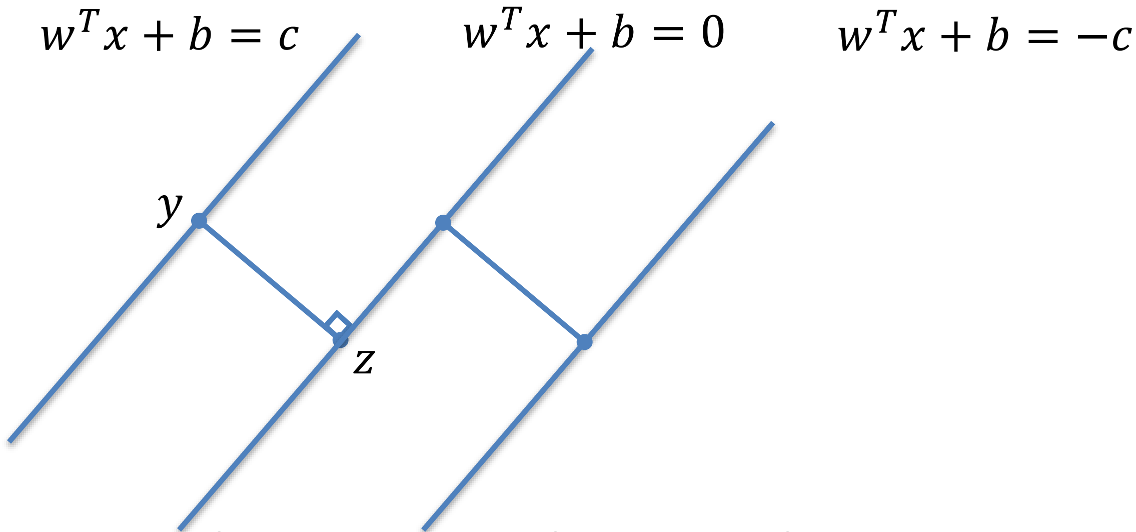
$$y - z = ||y - z|| \frac{w}{||w||}$$

Scale Invariance



- By scale invariance, we can assume that $c = 1$
- The maximum margin is always attained by choosing $w^T x + b = 0$ so that it is equidistant from the closest data point classified as +1 and the closest data point classified as -1

Scale Invariance

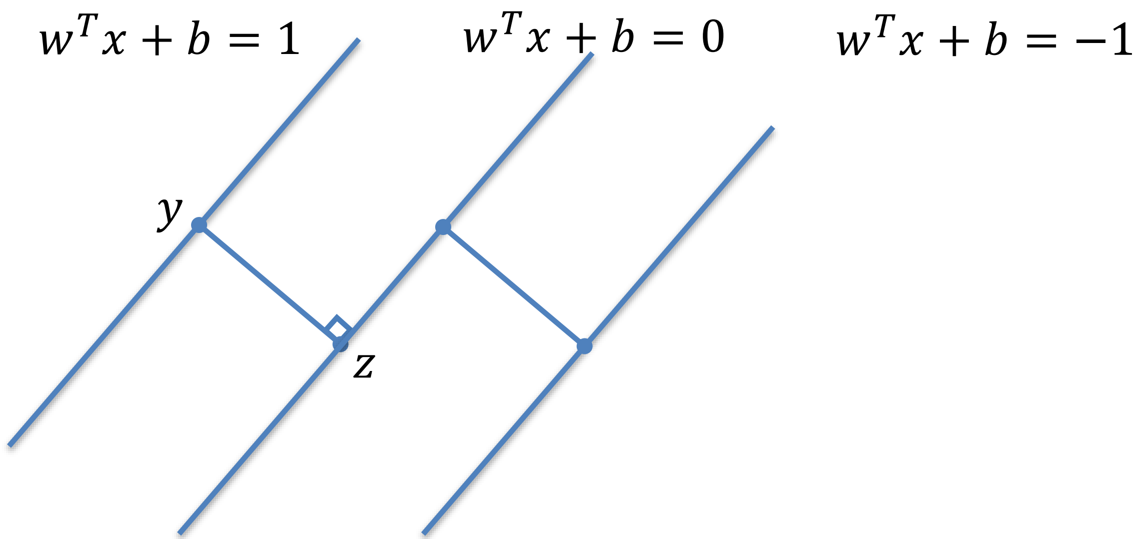


- We want to maximize the margin subject to the constraints that

$$y^{(i)}\left(w^T x^{(i)} + b\right) \geq 1$$

- But how do we compute the size of the margin?

Some Geometry



Putting it all together

$$y - z = \|y - z\| \frac{w}{\|w\|}$$

and

$$w^T y + b = 1$$
$$w^T z + b = 0$$

$$w^T (y - z) = 1$$

and

$$w^T (y - z) = \|y - z\| \|w\|$$

which gives

$$\|y - z\| = 1/\|w\|$$

SVMs

- This analysis yields the following optimization problem

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

such that

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

- Or, equivalently,

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

such that

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

SVMs

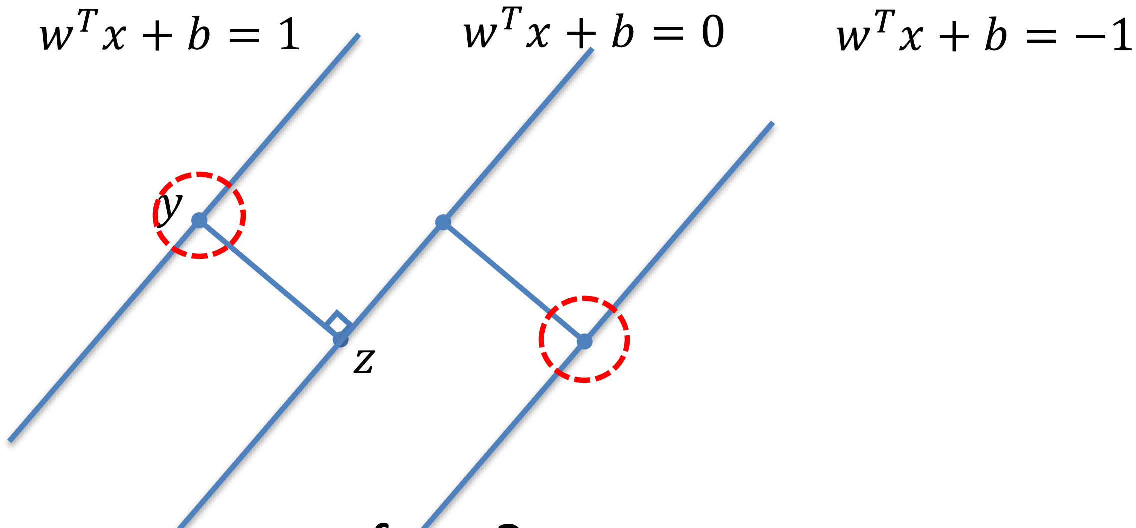
$$\min_{w,b} ||w||^2$$

such that

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

- 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)

SVMs



- 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)

SVMs

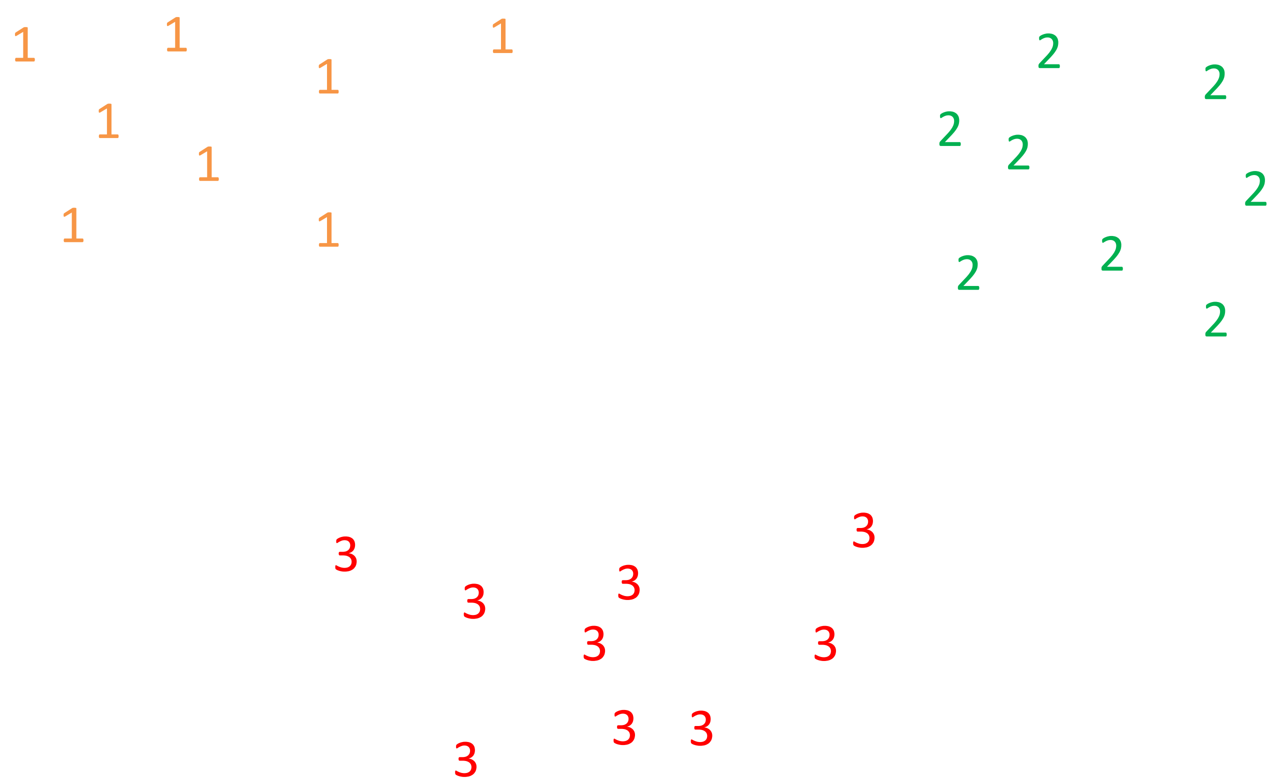
- What if the data isn't linearly separable?

- What if we want to do more than just binary classification (i.e., if $y \in \{1,2,3\}$)?

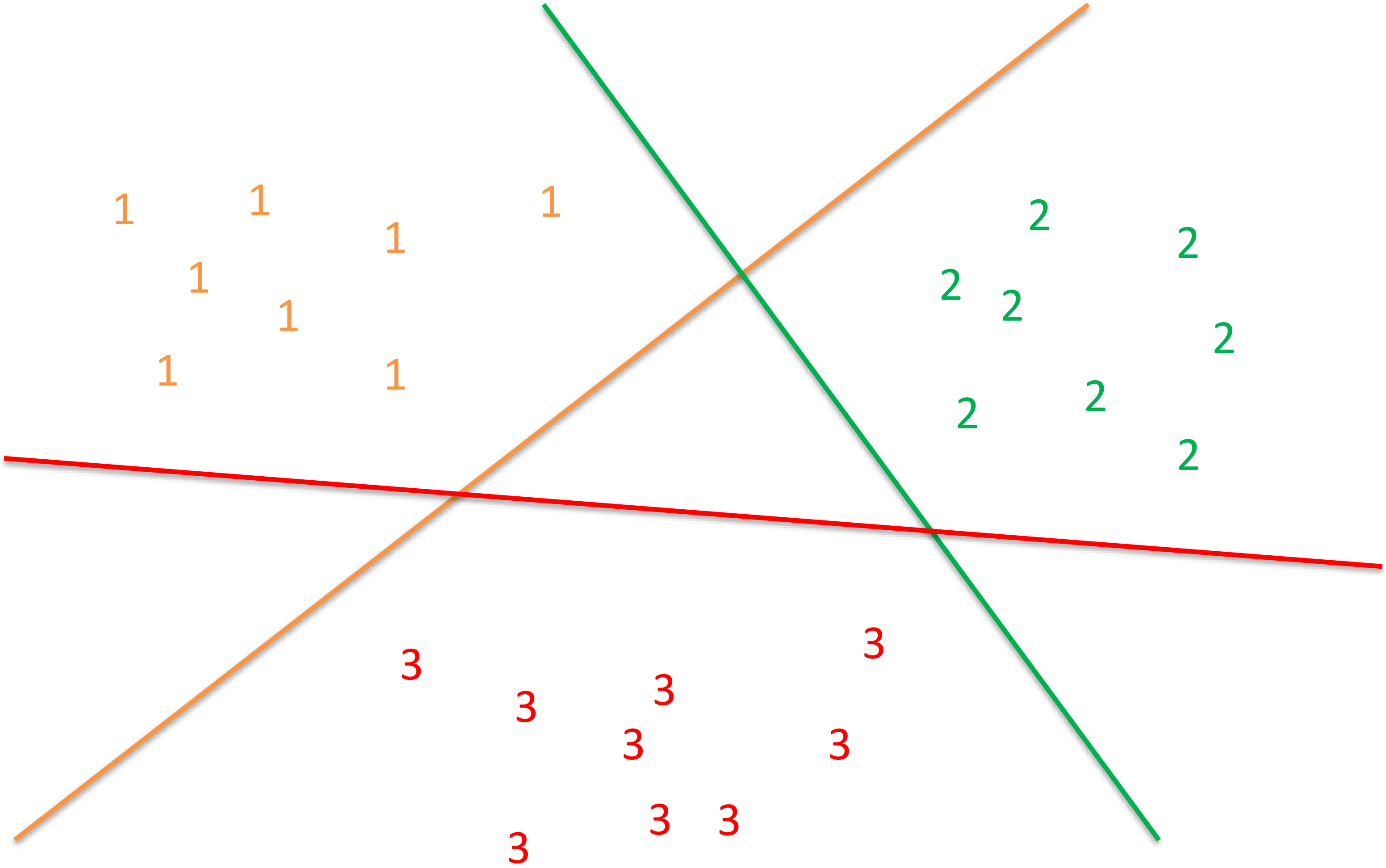
SVMs

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

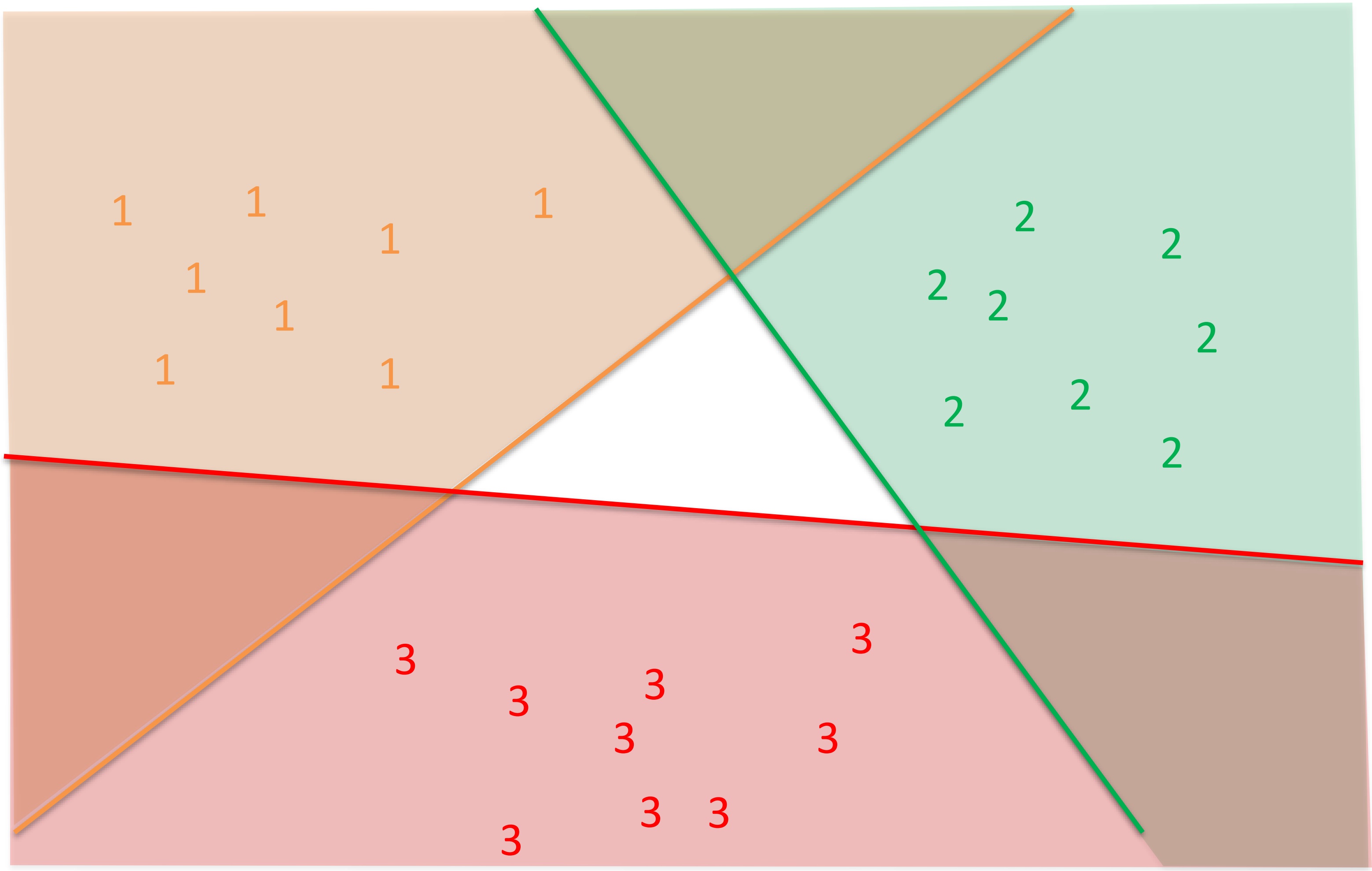
Multiclass Classification



One-Versus-All SVMs



One-Versus-All SVMs



Regions correctly classified by exactly one classifier

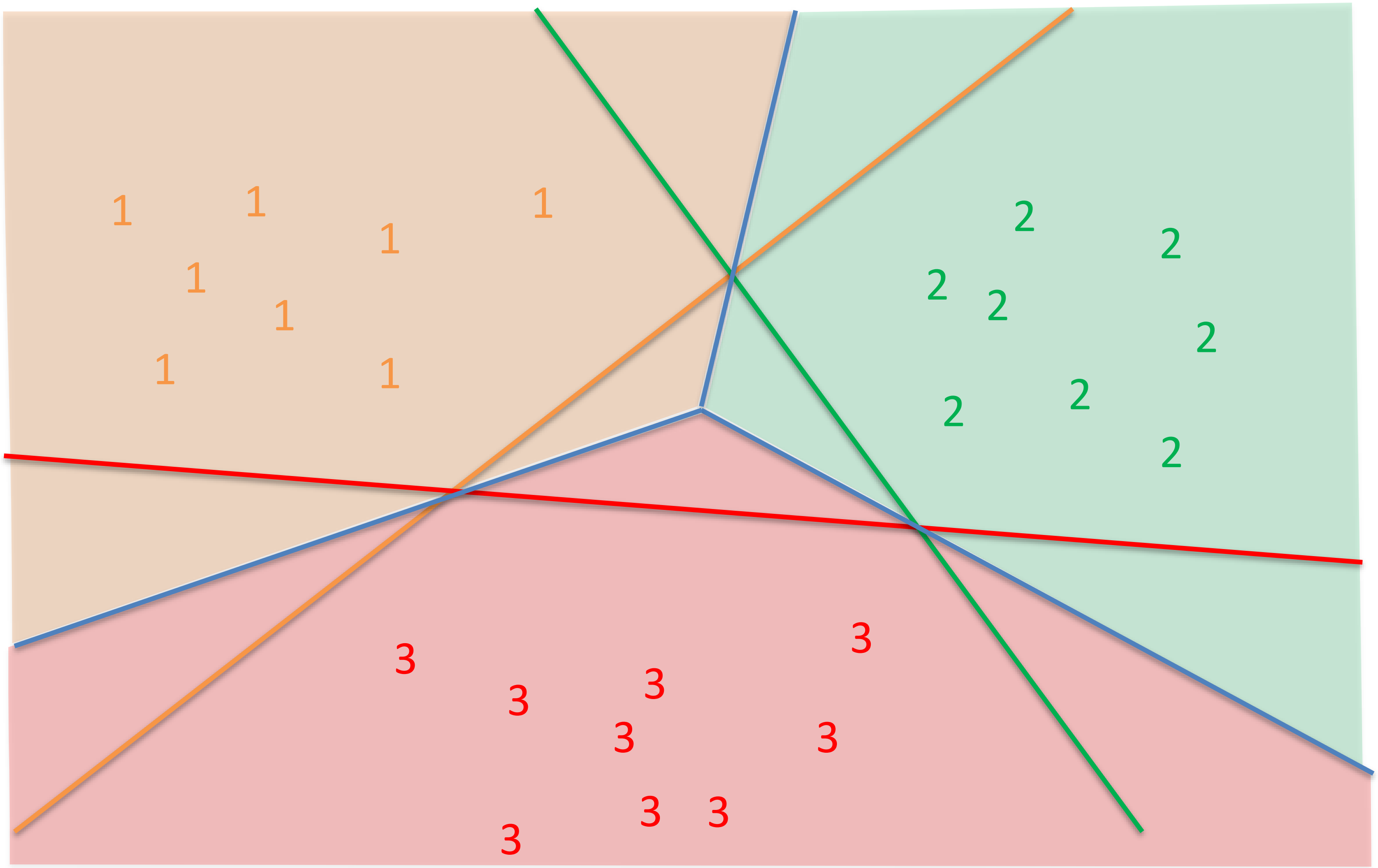
One-Versus-All SVMs

- Compute a classifier for each label versus the remaining labels (i.e., and 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

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

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

One-Versus-All SVMs

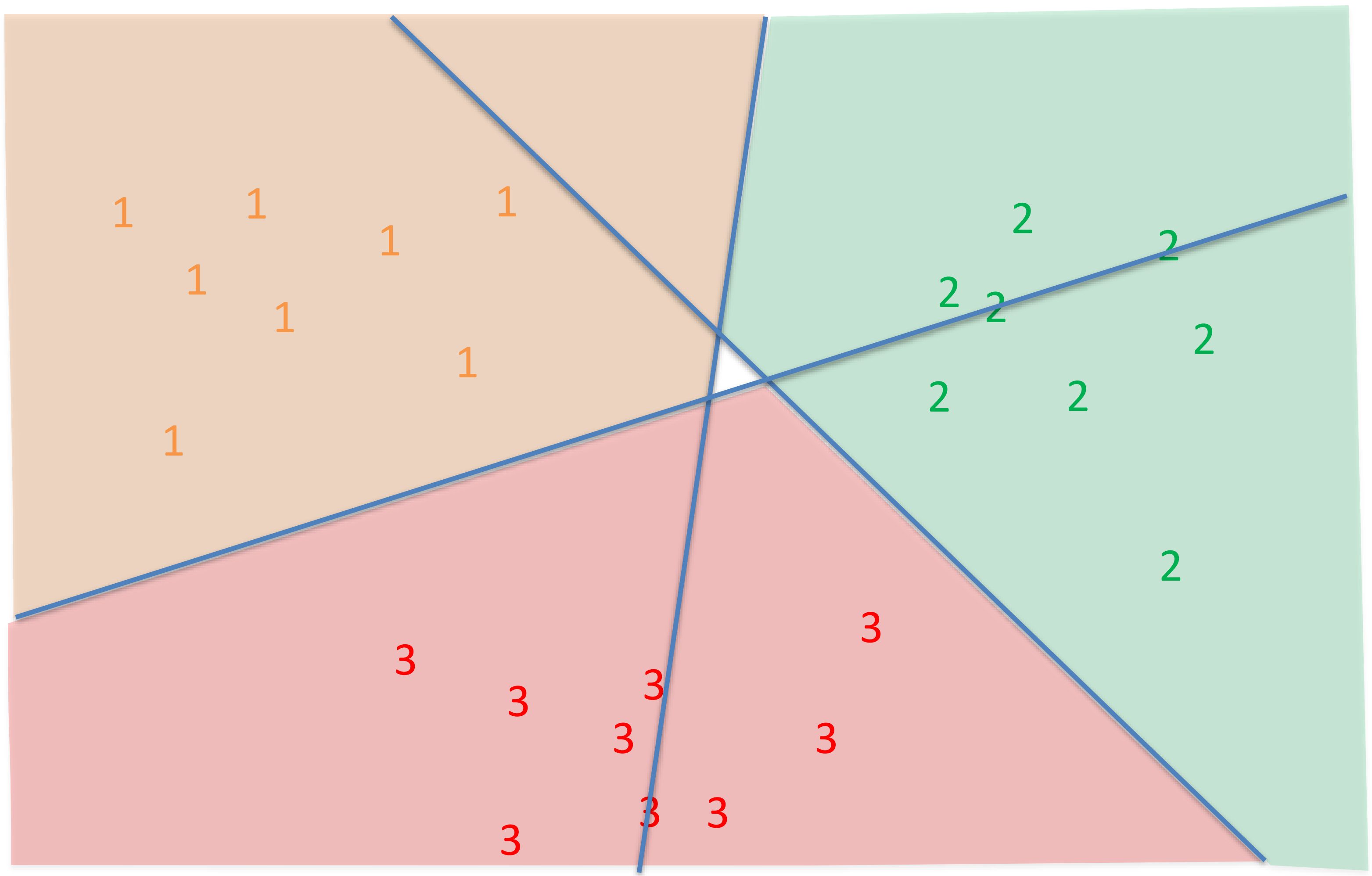


Regions in which points are classified by highest value of $w^T x + b$

One-Versus-One SVMs

- 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