CS 6375
Support Vector Machines

Rishabh Iyer

University of Texas at Dallas

Slides adapted from Nick Rouzzi and Andrew Zisserman

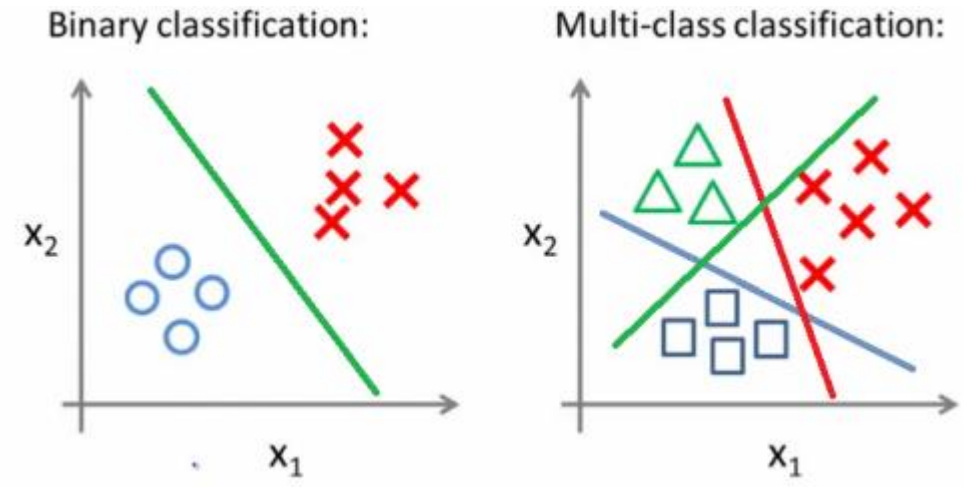
Announcements

- Assignment 1 will be out by EOW
  (Due in 2 weeks from the date it comes out)
- Either next Mon or Wed, we will have a discussion on project ideas
- Midterm exam will be mid to end October

Recap: Classification

Classification vs Regression

- Input: pairs of points $\left(x^{(1)}, y^{(1)}\right), \ldots, (x^{(M)}, y^{(M)})$ 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 $f: X \rightarrow Y$
- Goal: find the "best" element of the hypothesis space
  - How do we measure the quality of f?

$f_h(x, w, b) = \text{sign}(w^T x + b) \rightarrow$ classf^n

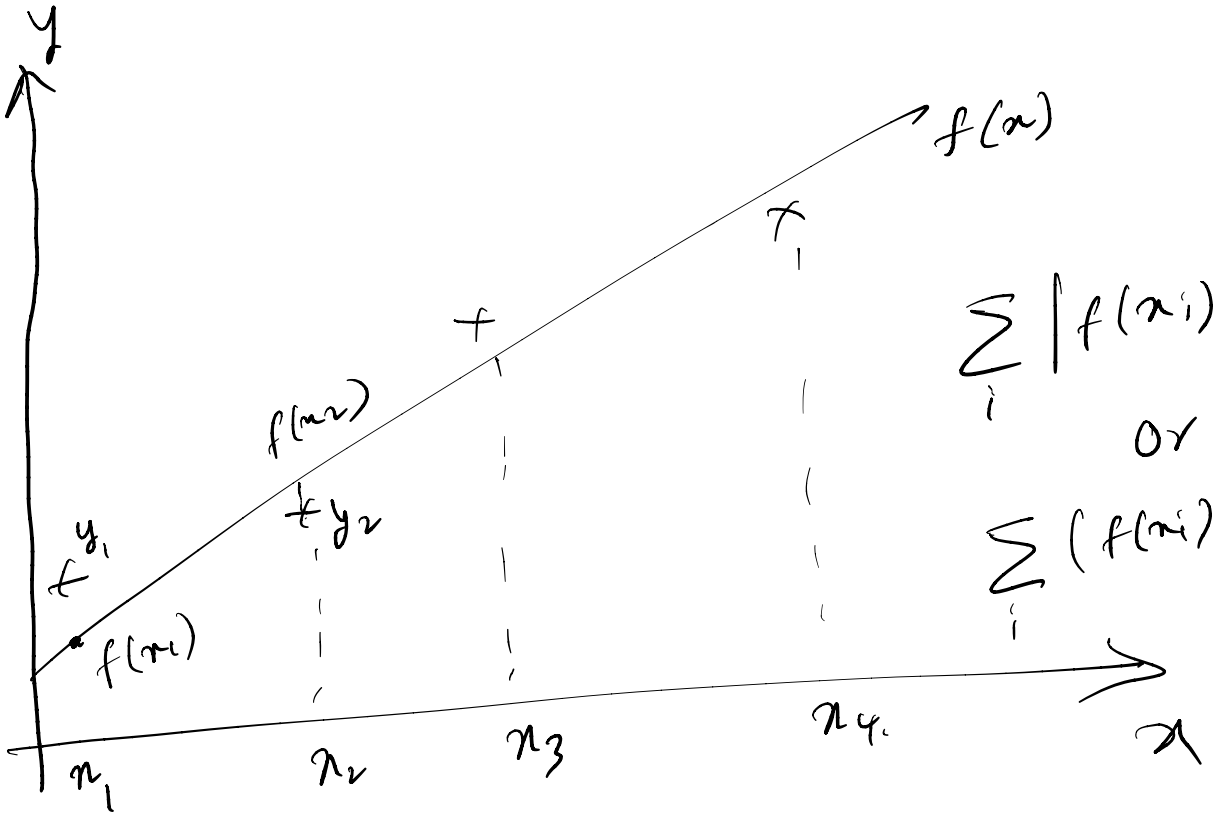
$w^T x + b \rightarrow$ Regression

Goal: $\forall i,\ f_h(x^{(i)}, w, b) \simeq y^{(i)}$

Loss: $L(w,b) = [y^{(i)} - f_h(x^{(i)}, w, b)]^2$, (regression)

$L(w,b) = \max_3(0, -y^{(i)}(w^T x + b))$ (Perceptron)

Why MSE / MAE and not sum?

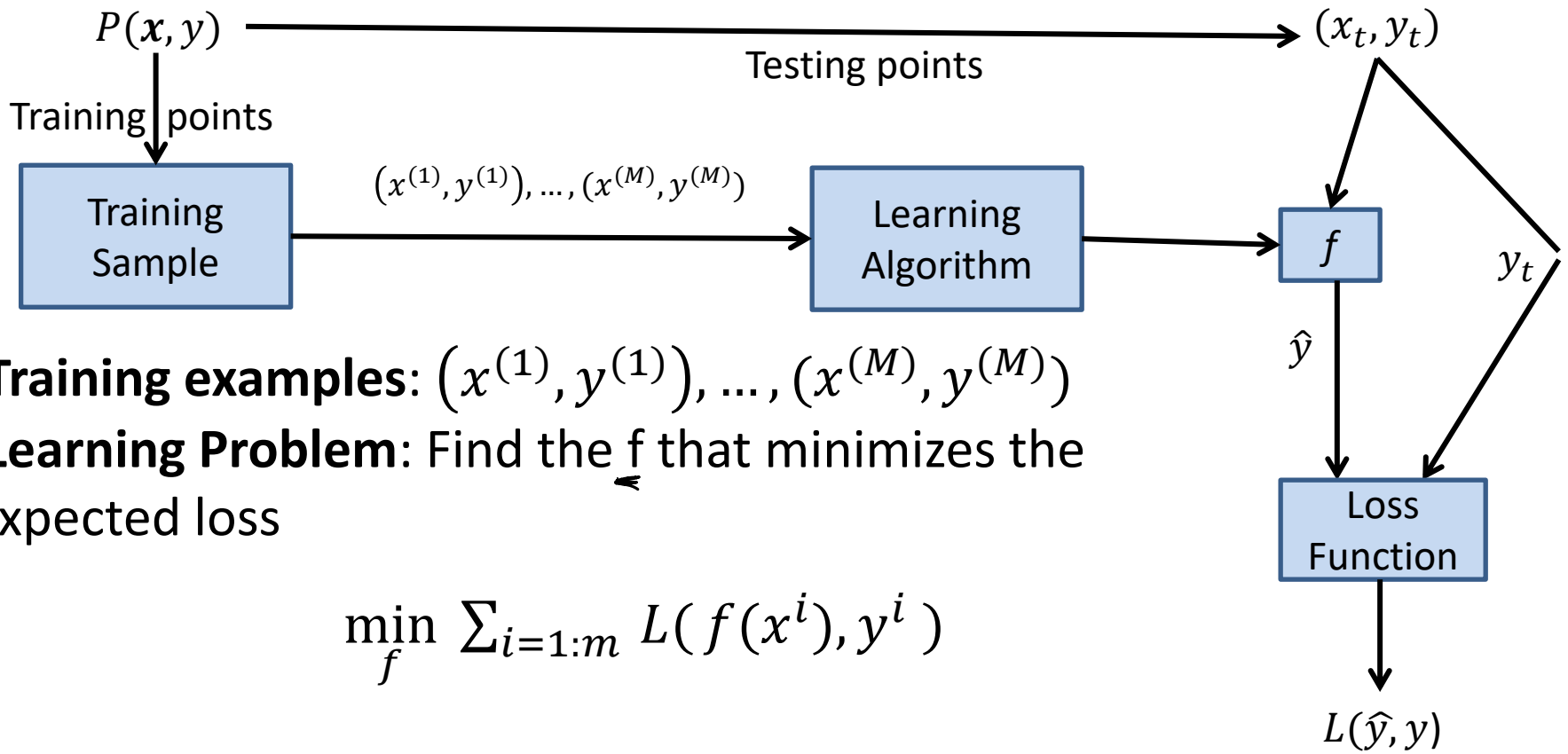


$$\sum_i |f(x_i) - y_i| \rightarrow \text{MAE}$$

or

$$\sum_i (f(x_i) - y_i)^2 \rightarrow \text{MSE}$$

Recap: Supervised Learning Workflow



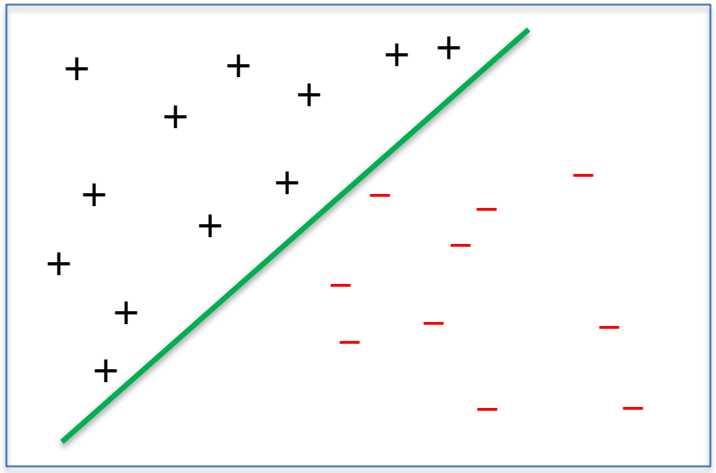
- **Training examples**: $(x^{(1)}, y^{(1)}), \ldots, (x^{(M)}, y^{(M)})$
- **Learning Problem**: Find the f that minimizes the expected loss

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

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

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

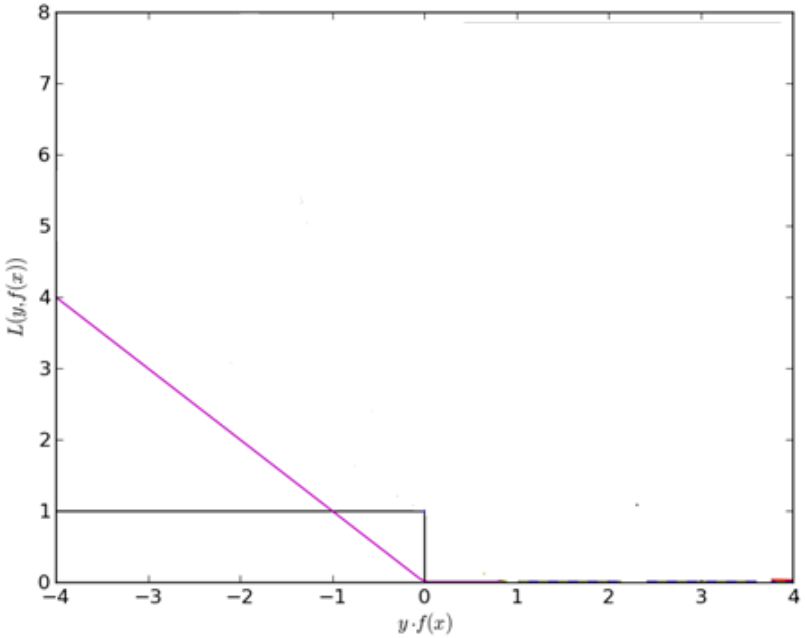
Not Cont

Grad techniques cant work

NP hard

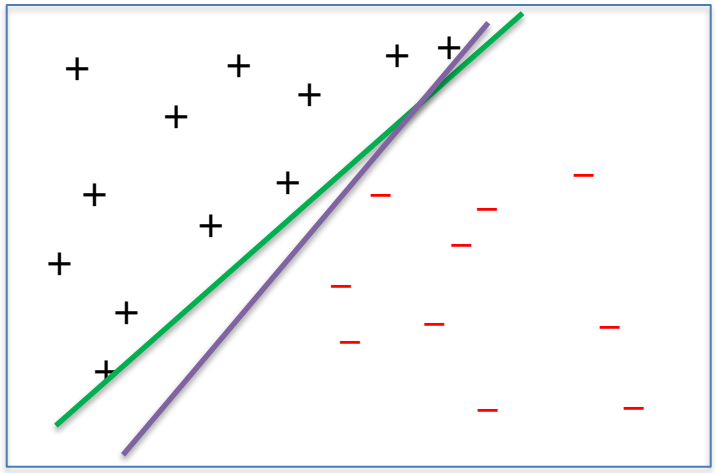
- Perceptron Loss:

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



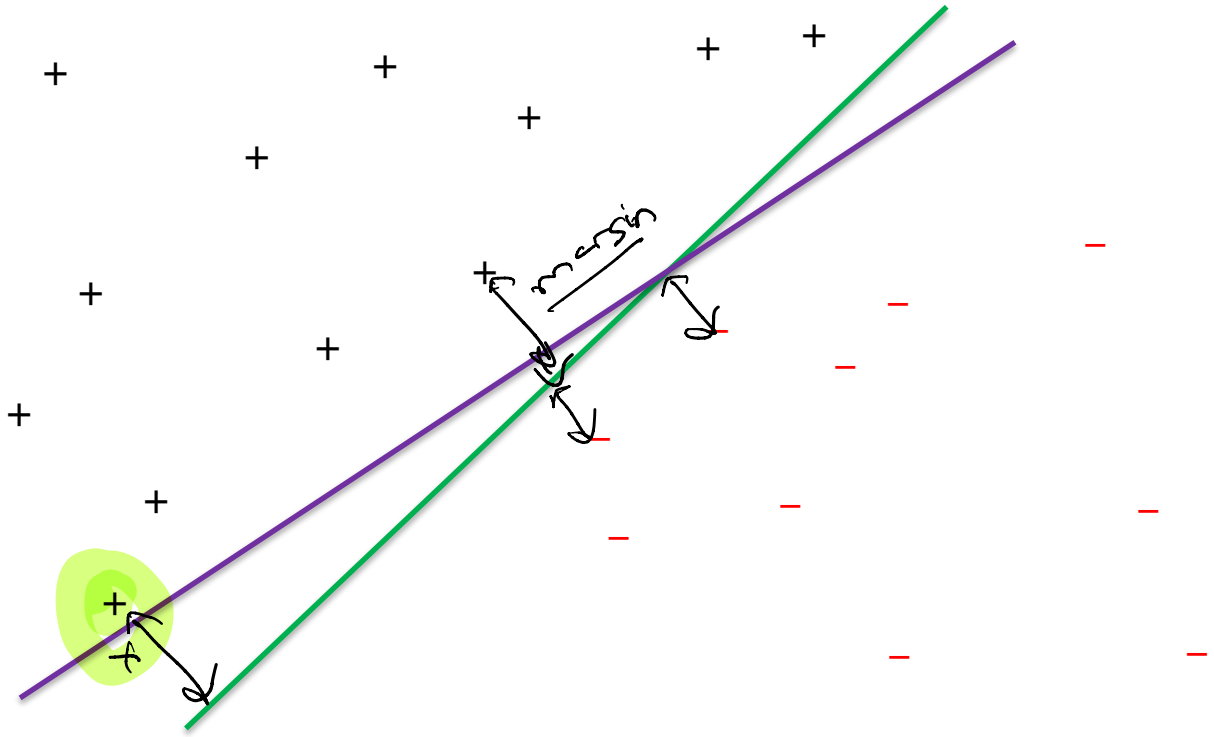
Perceptron Drawbacks

- 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



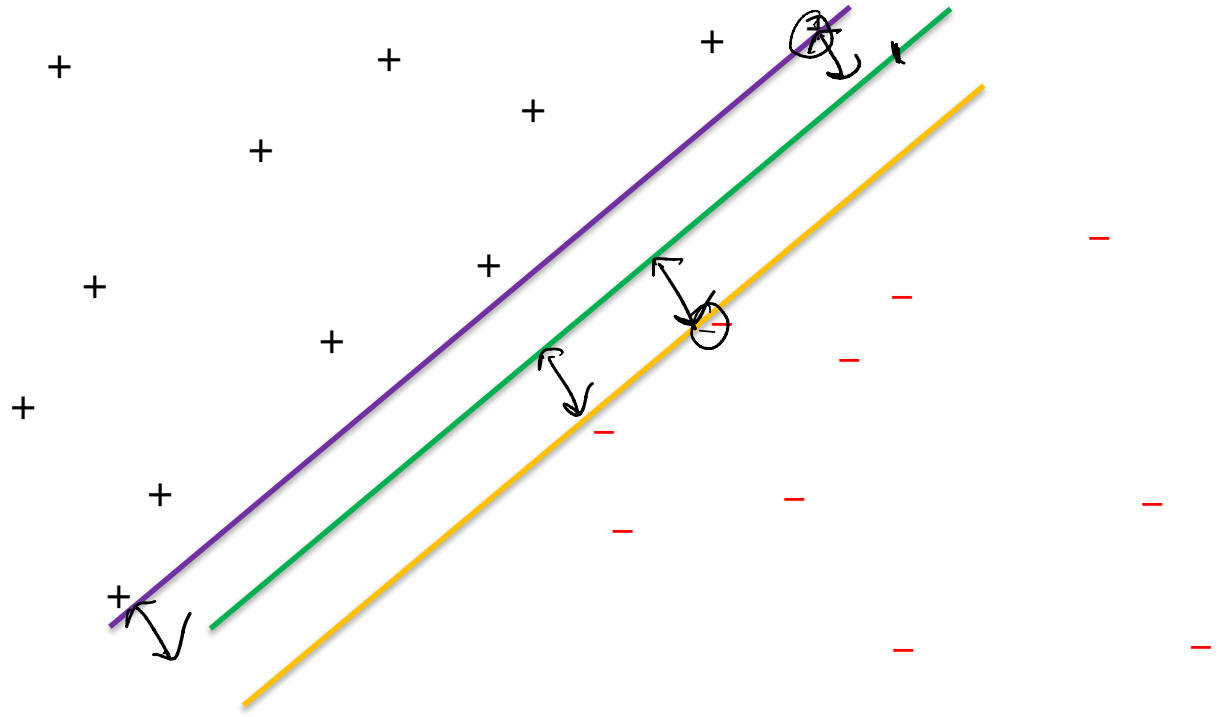
Support Vector Machines

- How can we decide between perfect classifiers?



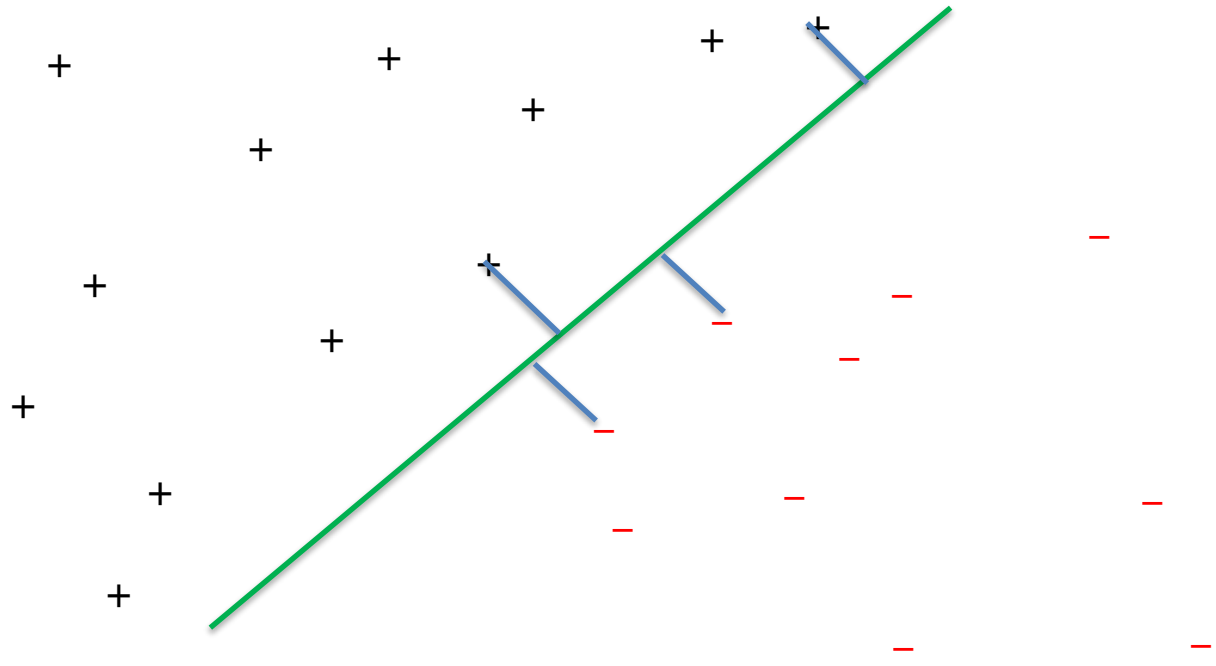
Support Vector Machines

- How can we decide between perfect classifiers?



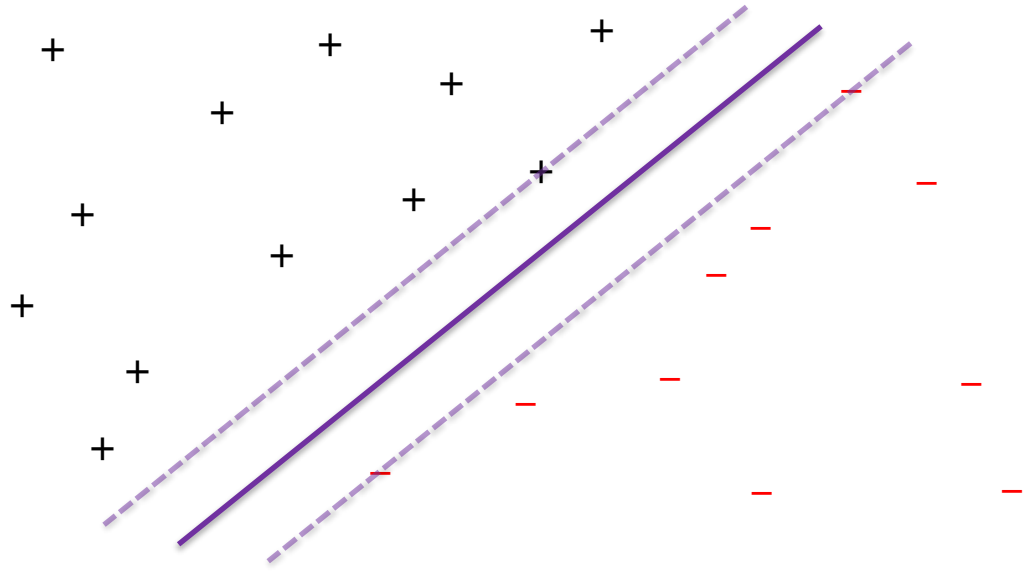
Support Vector Machines

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



Support Vector Machines

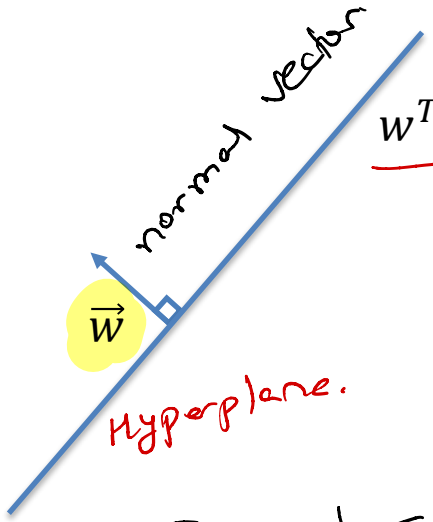
- Support vector machines (SVMs)



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

"Best Perf"
Generalization

Some Geometry

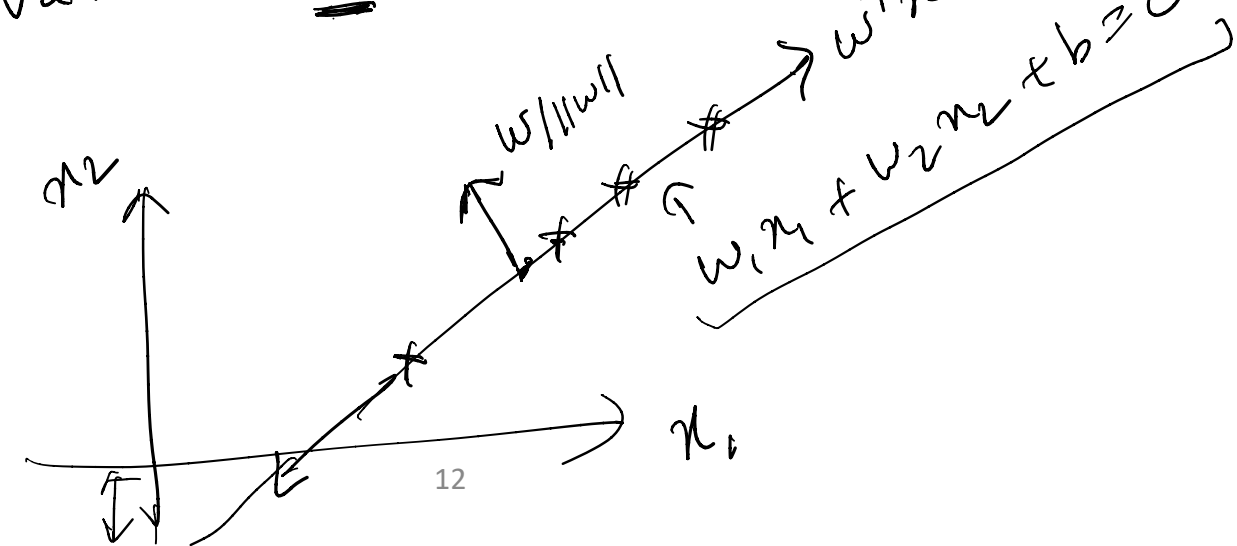


$w^T x + b = 0$

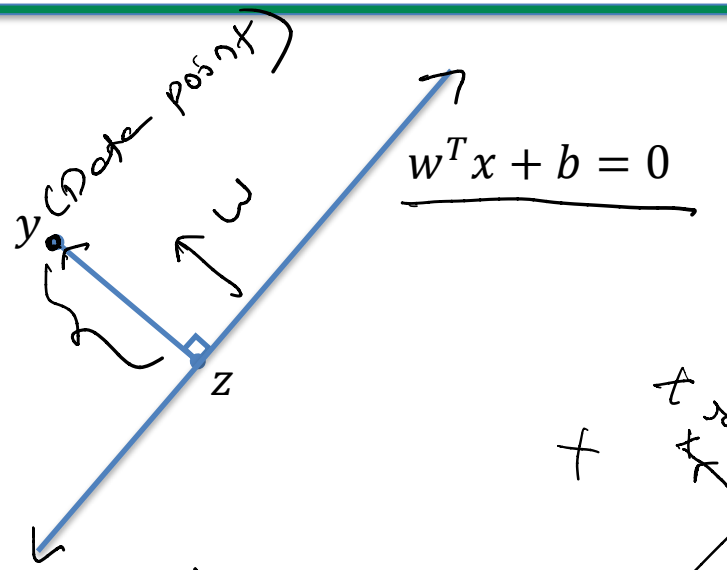
$2w^T x + 2b = 0$

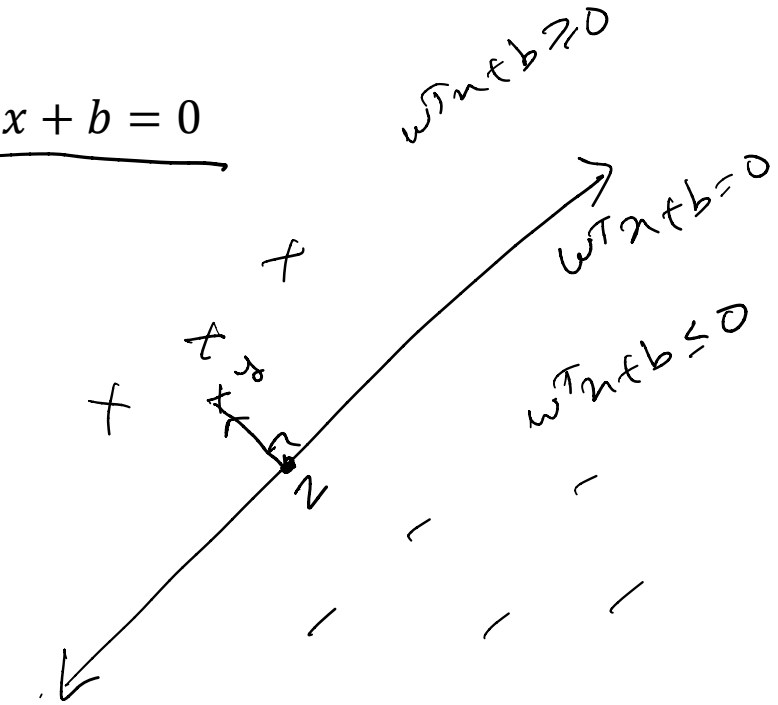
① Eq. of hyperplane : $w^T x + b = 0$

② Scale invariant $\leftarrow c(w^T x + b) = 0$



Some Geometry



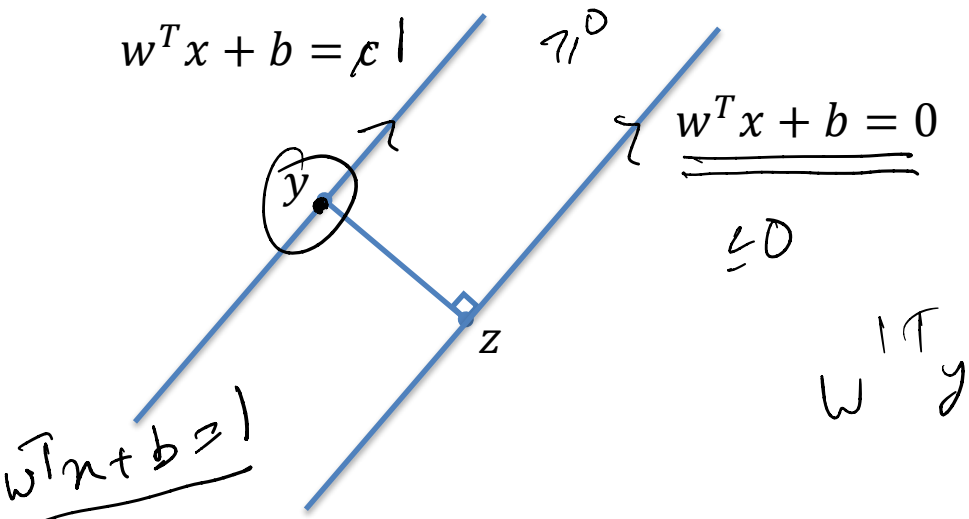


Claim: $y - z = \|y - z\| \frac{w}{\|w\|}$

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

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

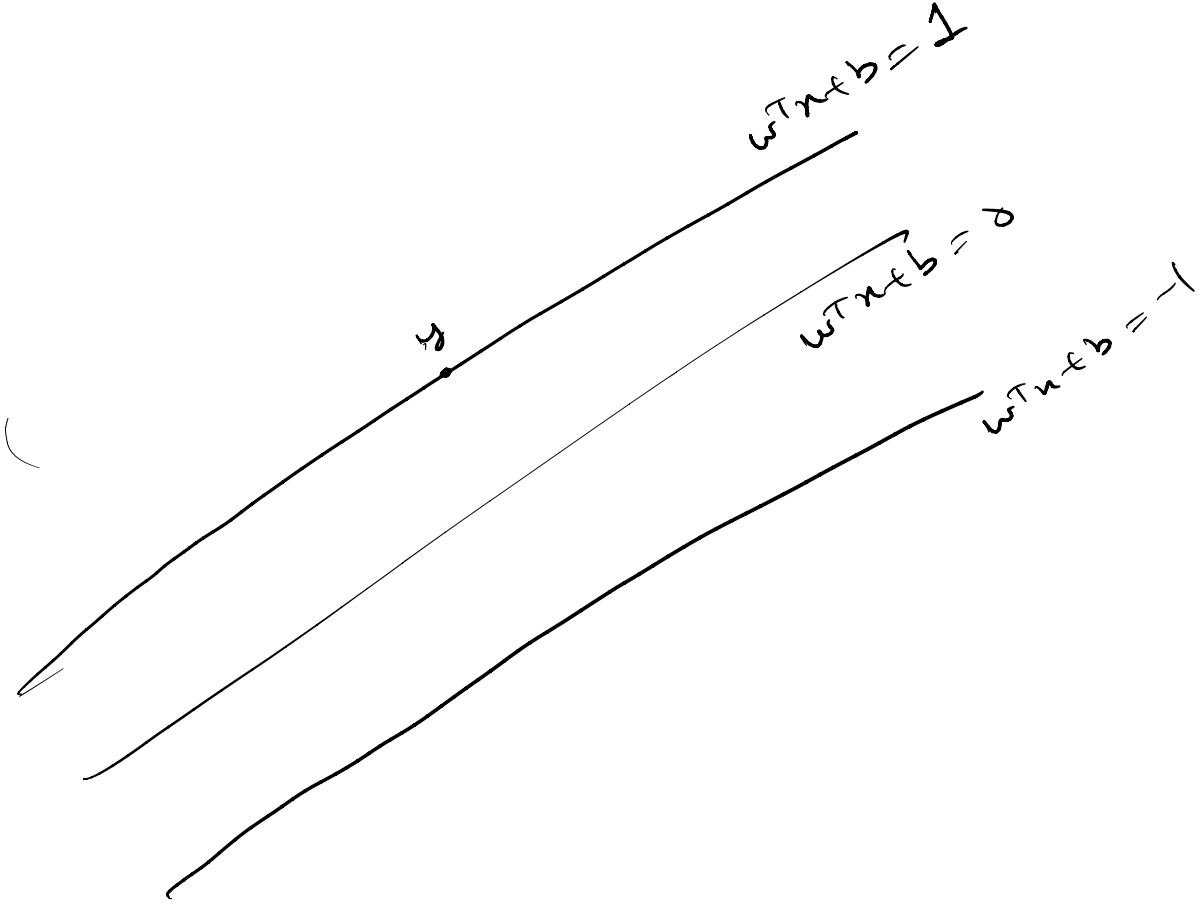
Scale Invariance



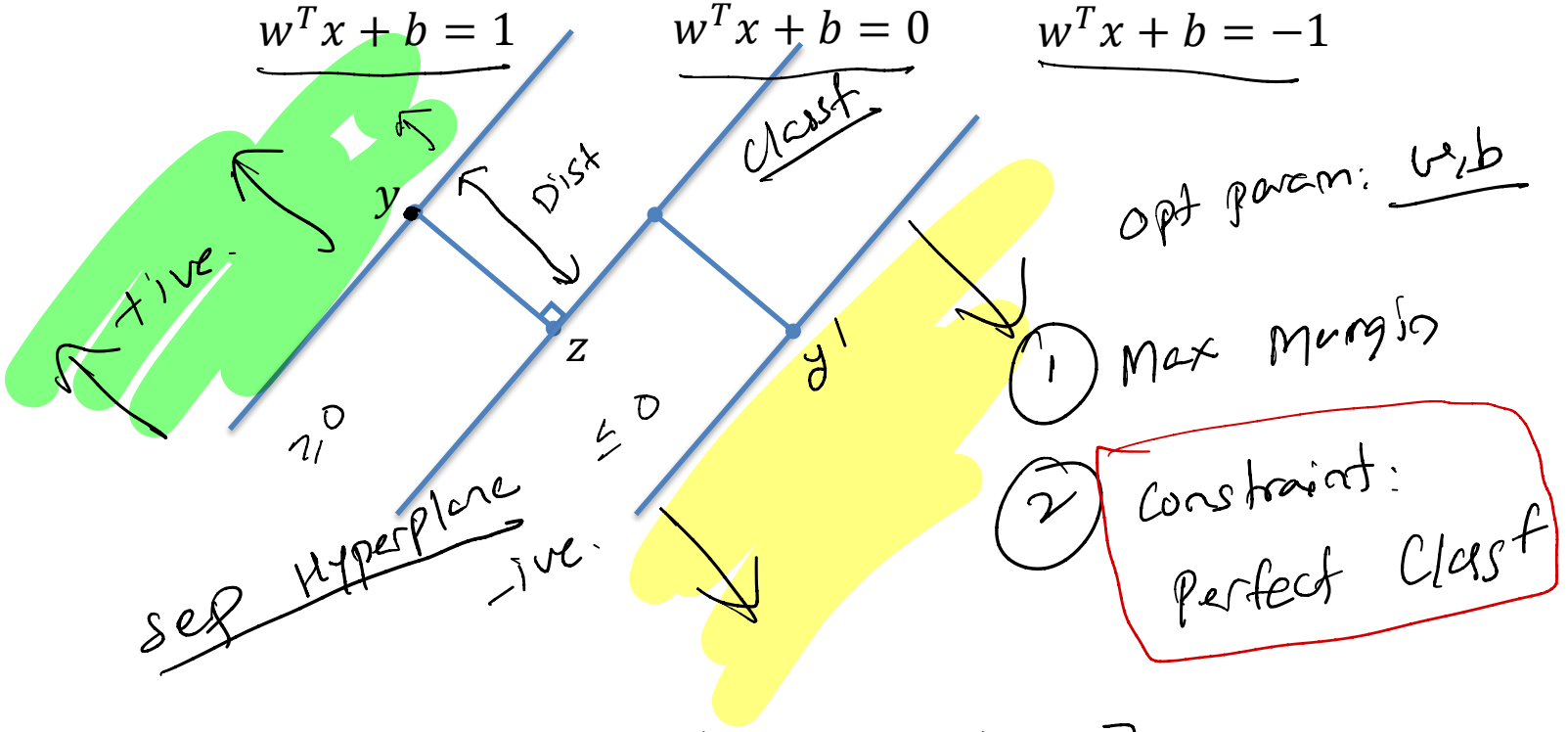
$w'^T y + b' = 1$

$w^T x + b = w^T y + b = c$

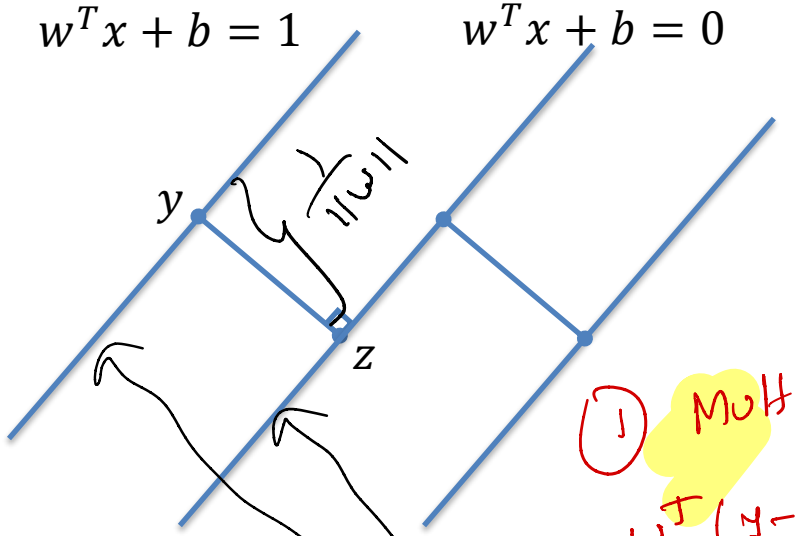
$w' = \frac{w}{c}$, $b' = \frac{b}{c}$


$w^T x + b = 1$
$w^T x + b = 0$
$w^T x + b = -1$
x

Constraints



What is the Margin?



$\frac{|w^T y + b|}{\|w\|} = \frac{1}{\|w\|}$

① $y - z = \|y - z\| \frac{w}{\|w\|}$

② $w^T y + b = 1$

$w^T z + b = 0$

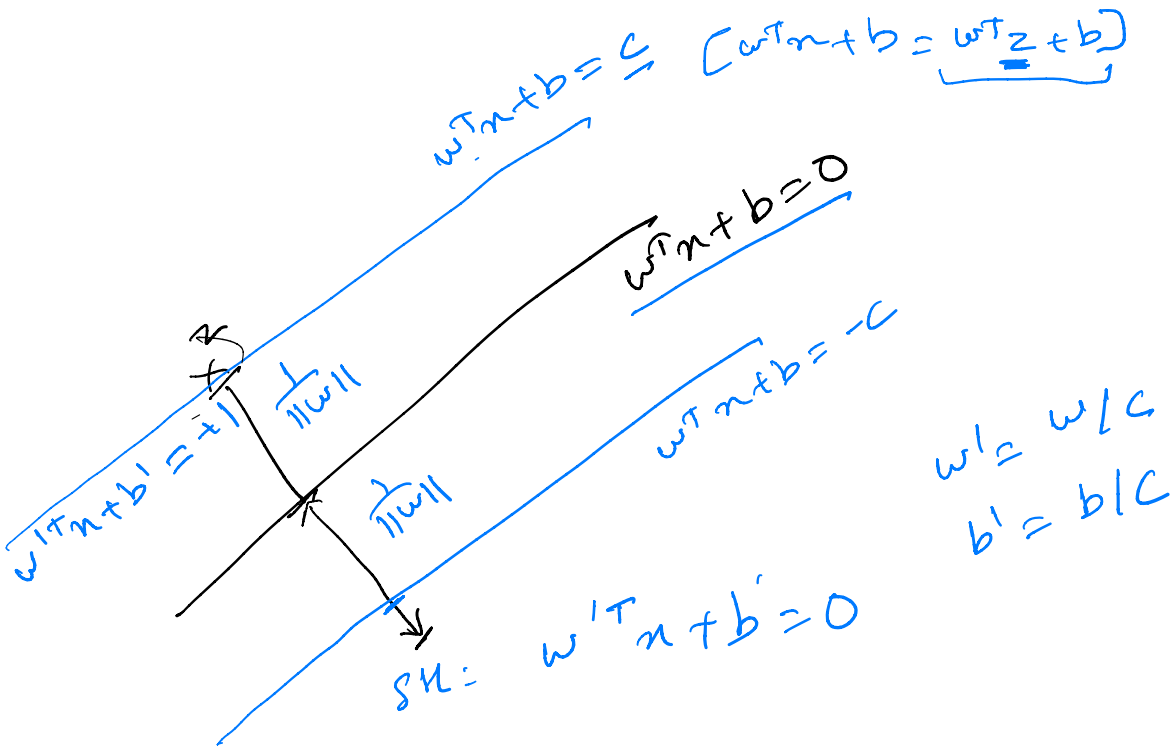
$\Rightarrow w^T (y - z) = 1$

① Mult w to both sides

$w^T (y - z) = \|y - z\| \frac{\|w\|^2}{\|w\|}$

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

$\|y - z\| = \frac{1}{\|w\|}$


$w^T x + b = c$ $[w^T x + b = w^T z + b]$
$w^T x + b = 0$
$w^T x + b = -c$
$w'^T x + b' = +1$
$\frac{1}{\|w\|}$
$\frac{1}{\|w\|}$
SH: $w'^T x + b' = 0$
$w' = w/c$
$b' = b/c$

SVMs

- This analysis yields the following optimization problem

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

such that

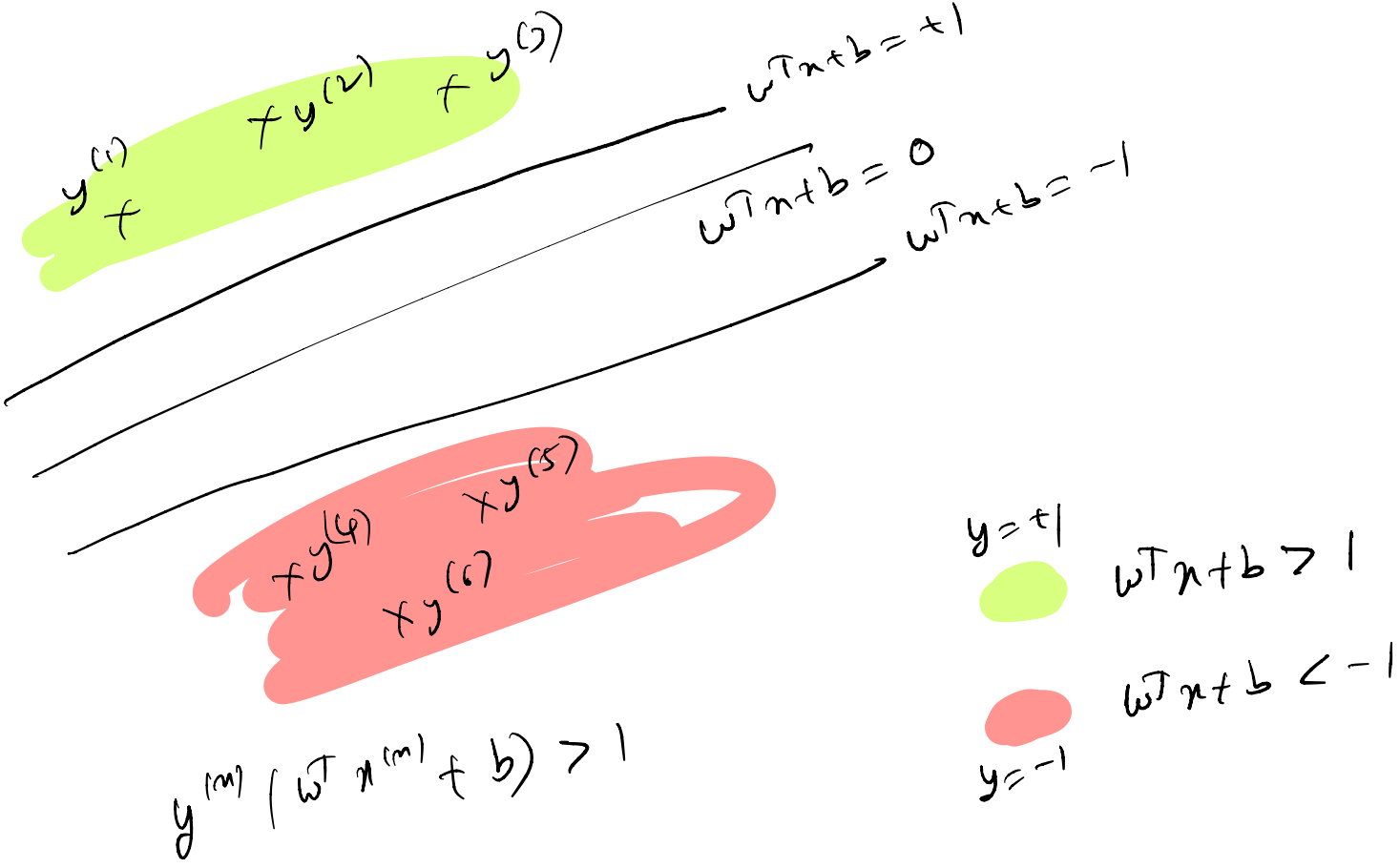
$$y^{(i)}\left(w^T x^{(i)} + b\right) \geq 1, \text{for all } i$$

- Or, equivalently,

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

such that

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


$y^{(1)}$
$y^{(2)}$
$y^{(3)}$
$w^T x + b = +1$
$w^T x + b = 0$
$w^T x + b = -1$
$y^{(4)}$
$y^{(5)}$
$y^{(6)}$
$y^{(7)}$
$y^{(m)}(w^T x^{(m)} + b) > 1$
$y = +1$
$w^T x + b \geq 1$
$w^T x + b < -1$
$y = -1$

First Principles of SVM

$$D = \{(x^{(1)}, y^{(1)}), \ldots (x^{(M)}, y^{(M)})\}$$

Model w, b.

$$\max_{w,b} \text{Margin}(w, b, D)$$

$$\text{s.t} \quad (w, b) \in \text{Perfect-Classifier}$$

$\Rightarrow$

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

$$\text{s.t} \quad y^{(i)}\left[w^T x^{(i)} + b\right) \geq 1$$

$$\forall \; i = 1 \in 1, \ldots M$$

SVMs

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

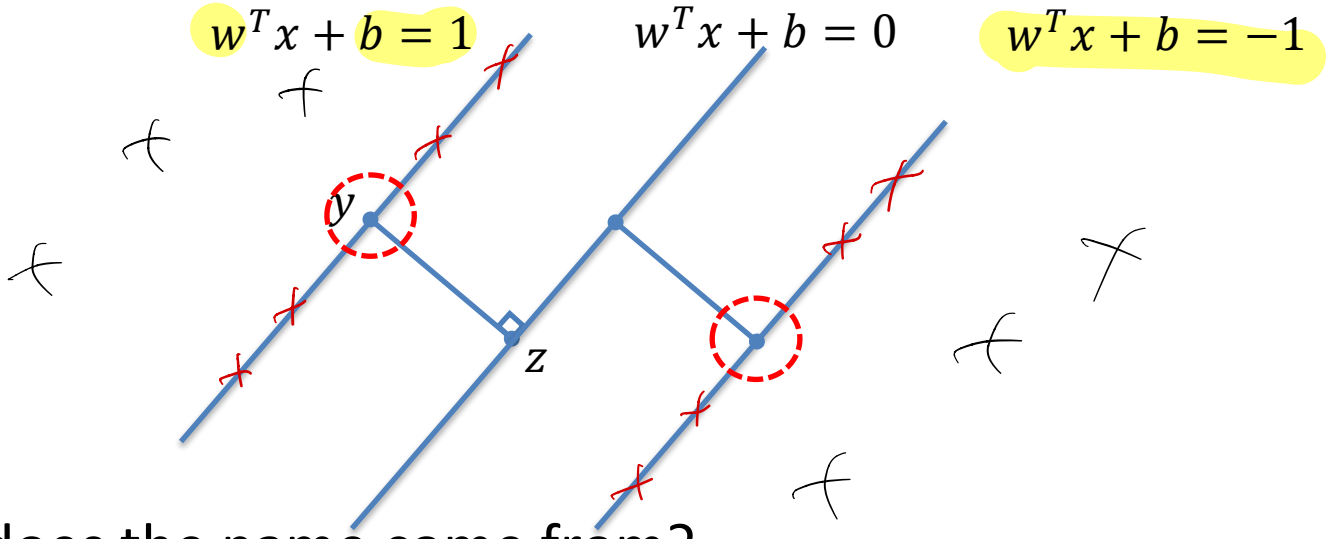
such that

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

- 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)

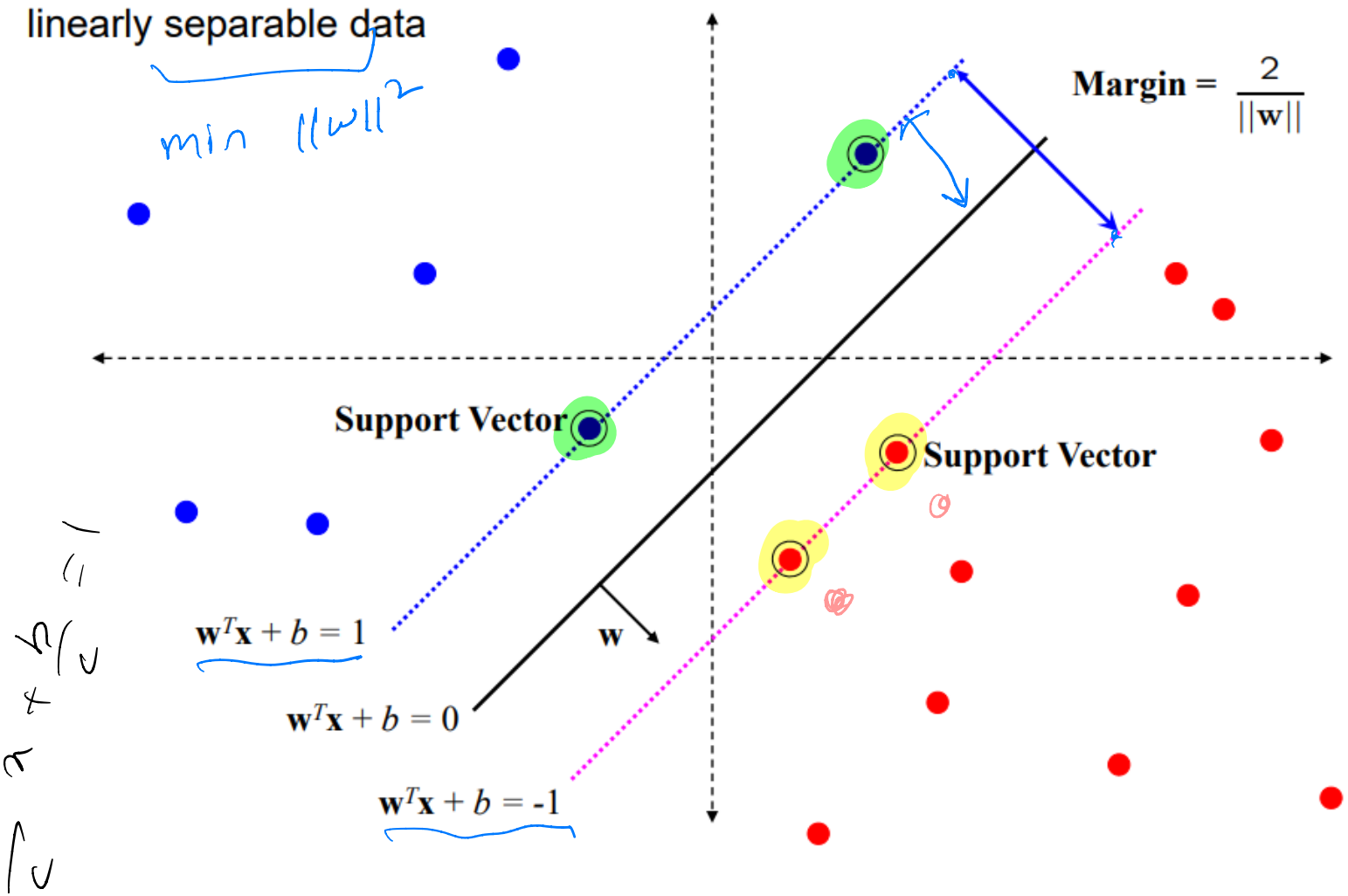
$$\sum_{i=1}^{M} L\left(f(x^i, w, b), y^i\right)$$

Support Vectors



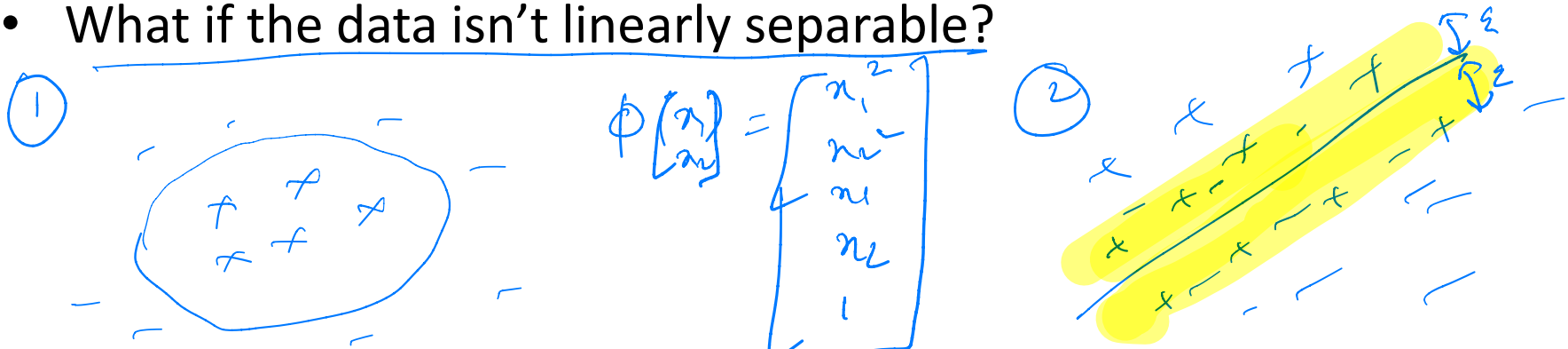
- Where does the name come from?
  - The set of all data points such that $y^{(i)}(w^T x^{(i)} + b) = 1$ are called support vectors
  - The SVM classifier is completely determined by the support vectors (you could delete the rest of the data and get the same answer)

Putting Everything Together



SVMs

- What if the data isn't linearly separable?



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

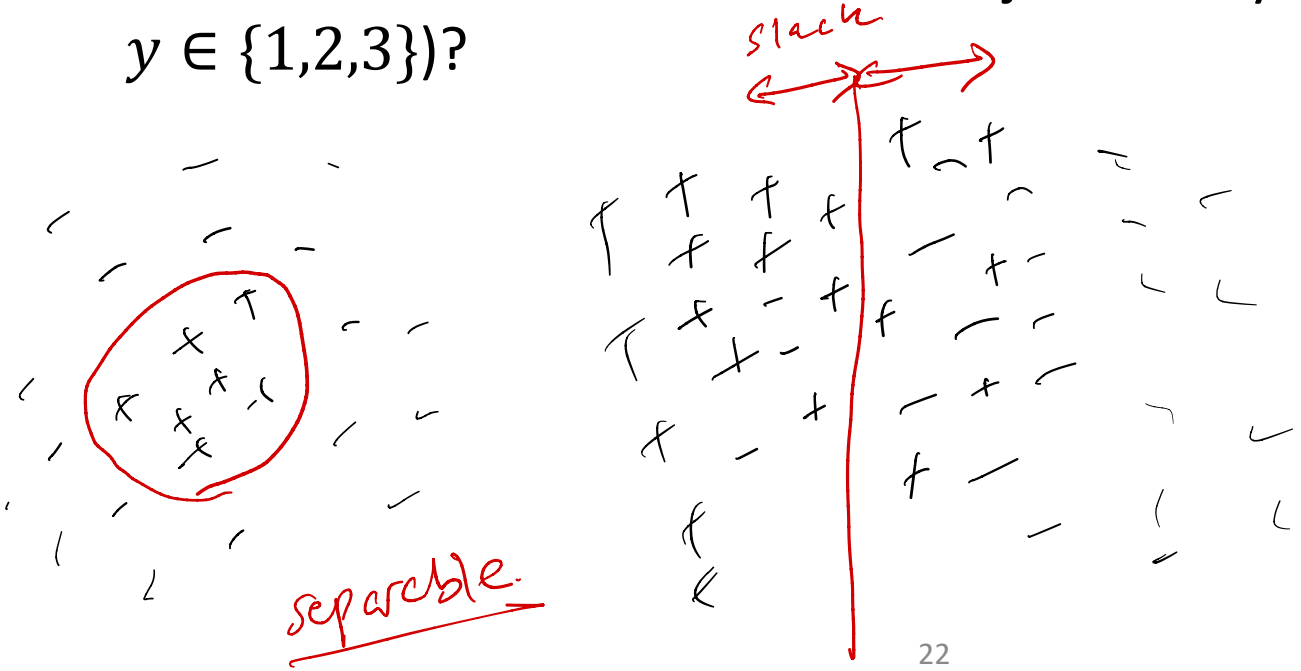
SVMs

- What if the data isn't linearly separable?
  - Higher order (polynomial features)
  - Relax the constraints (coming soon)

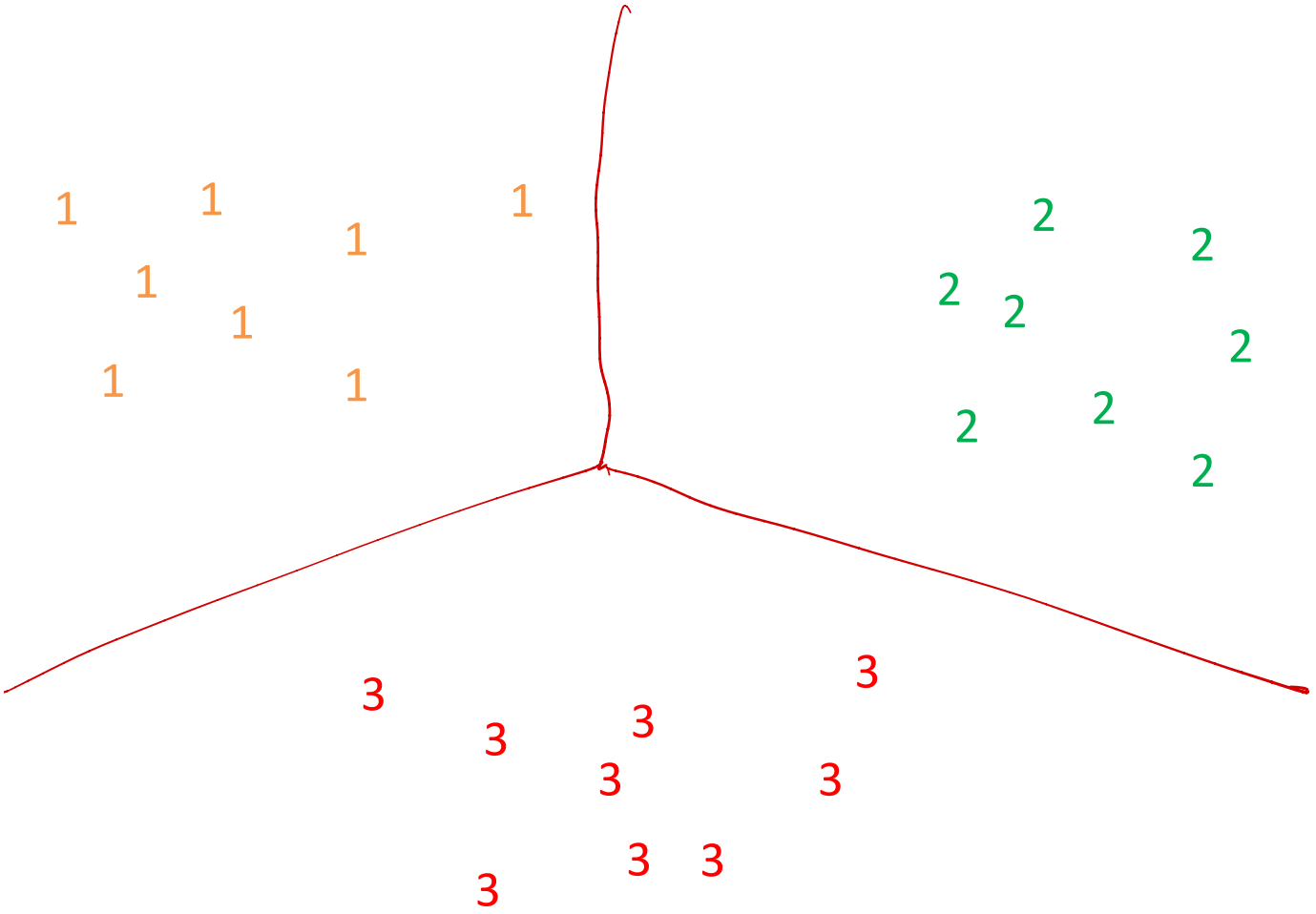
Linear Regression

Perceptron

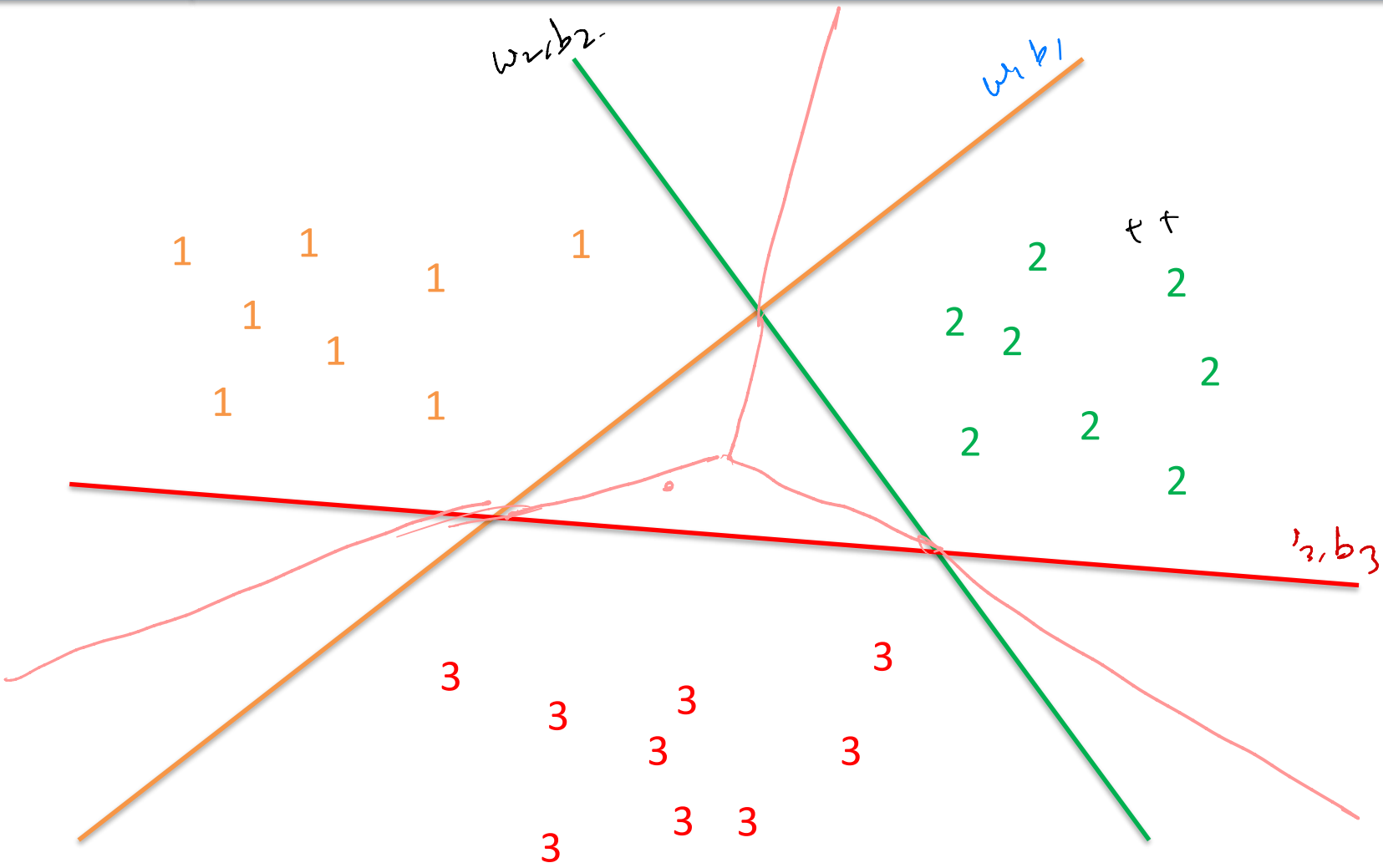
- What if we want to do more than just binary classification (i.e., if $y \in \{1,2,3\}$)?



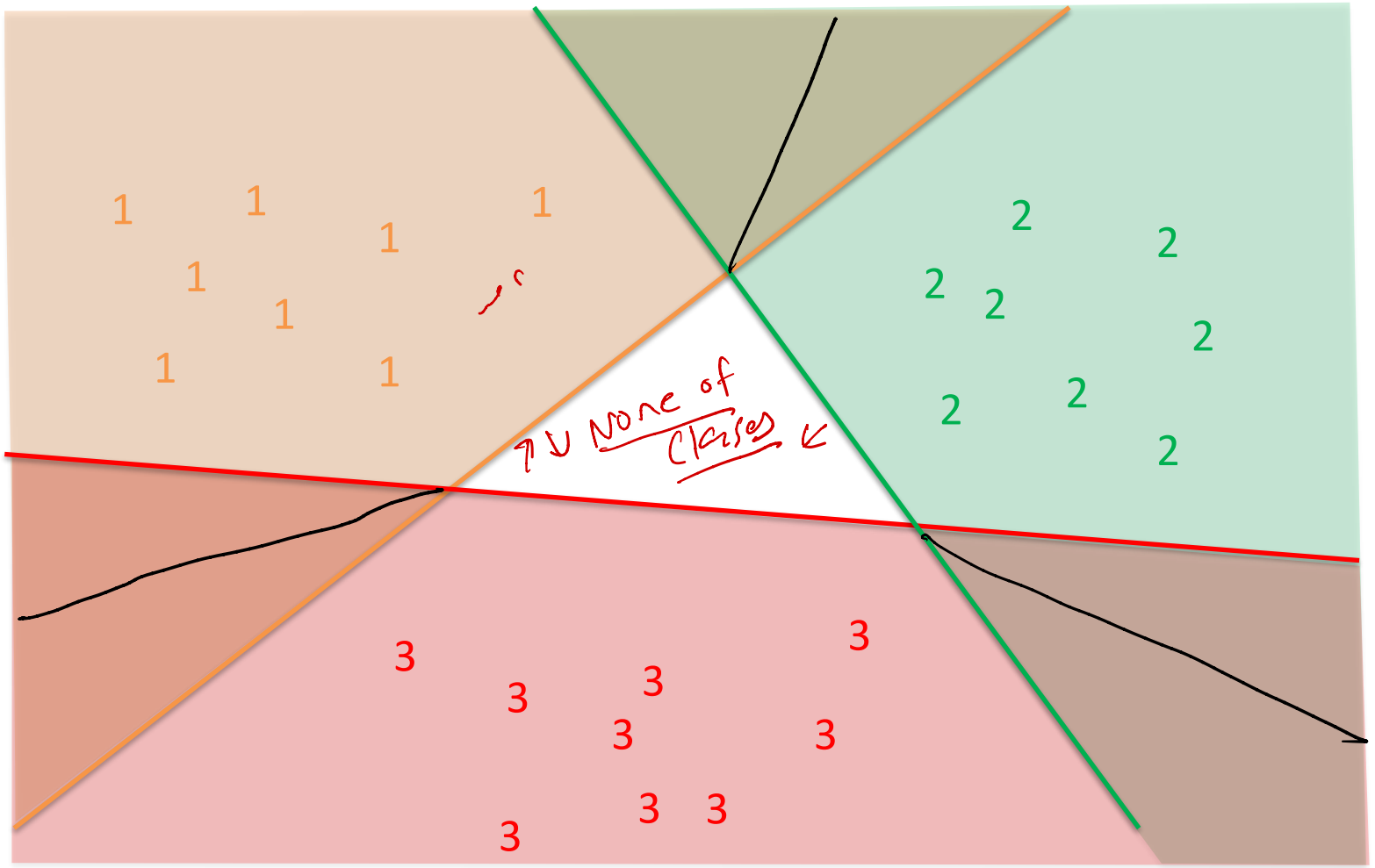
Multiclass Classification



One-Versus-All SVMs



One-Versus-All SVMs



Regions correctly classified by exactly one classifier

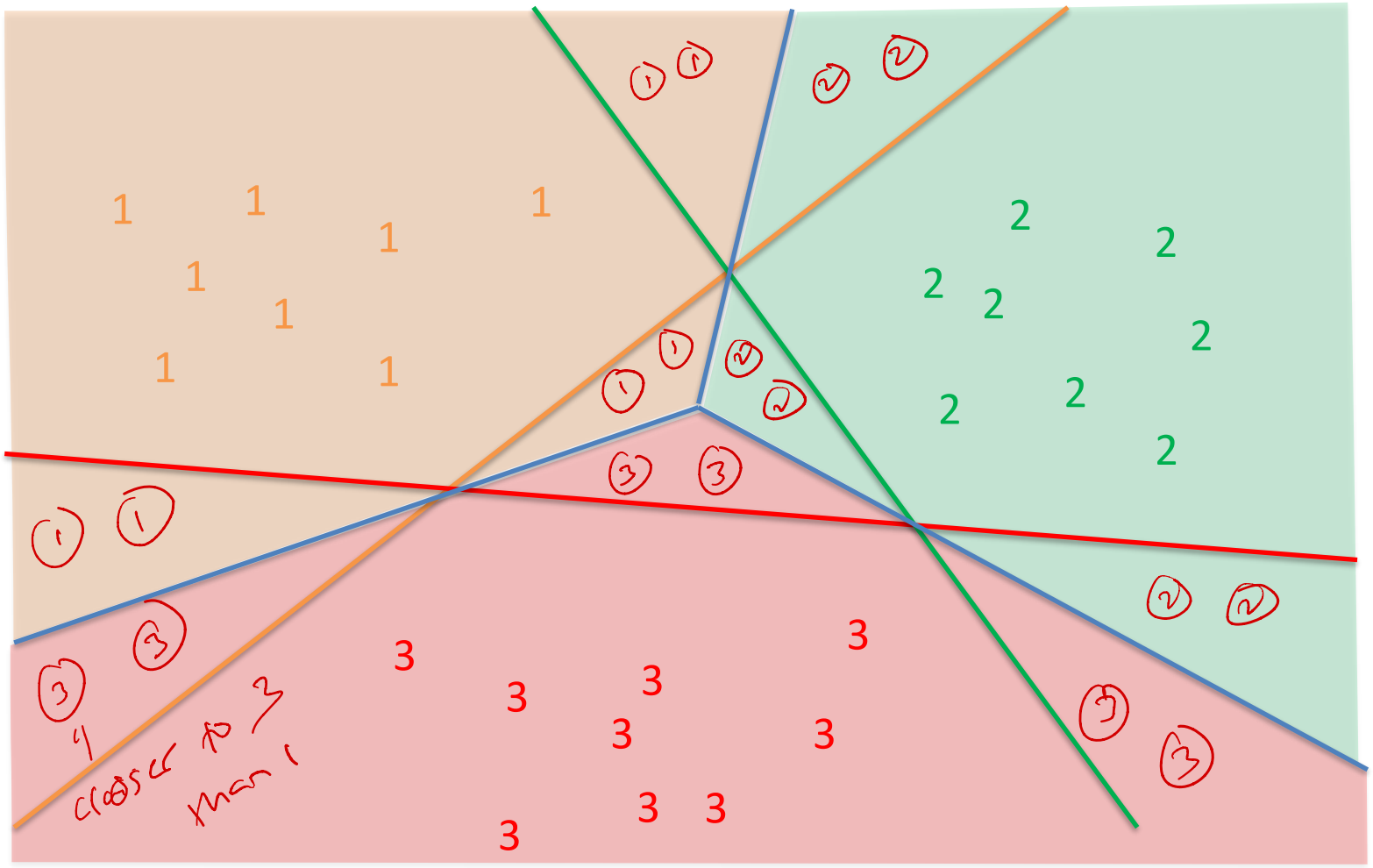
One-Versus-All SVMs

- 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

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

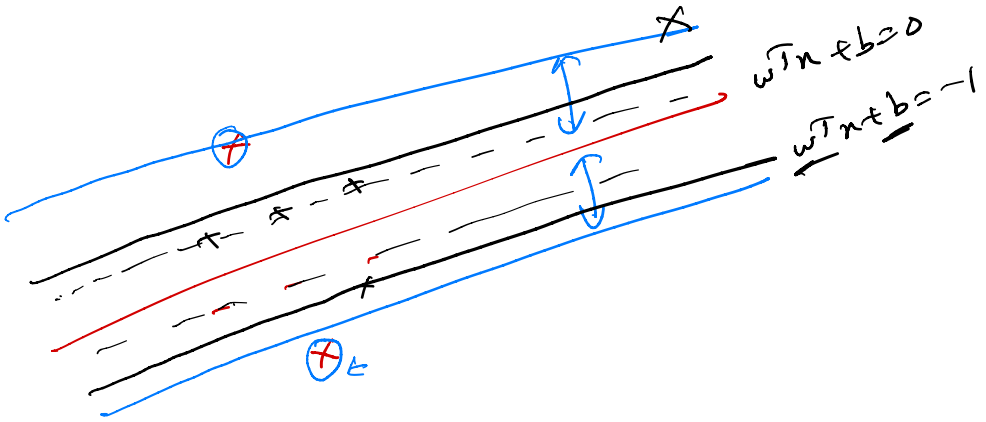
- Drawbacks:
  - If there are L possible labels, requires learning L classifiers over the entire data set

One-Versus-All SVMs



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

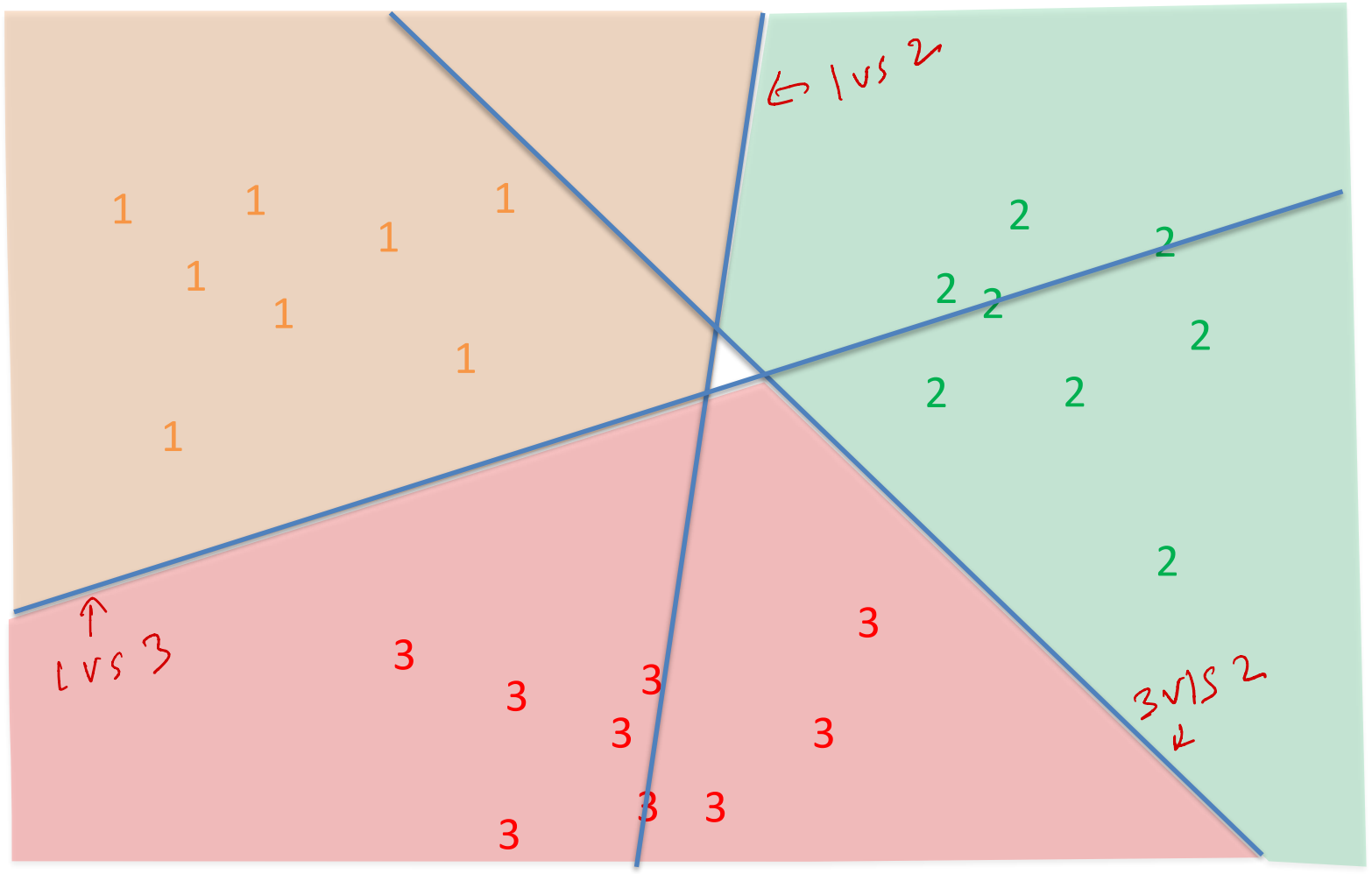
$$\underset{k}{\arg\max}\; w^{(k)T} x + b^{(k)}$$


$w^T x + b = 0$
$w^T x + b = -1$

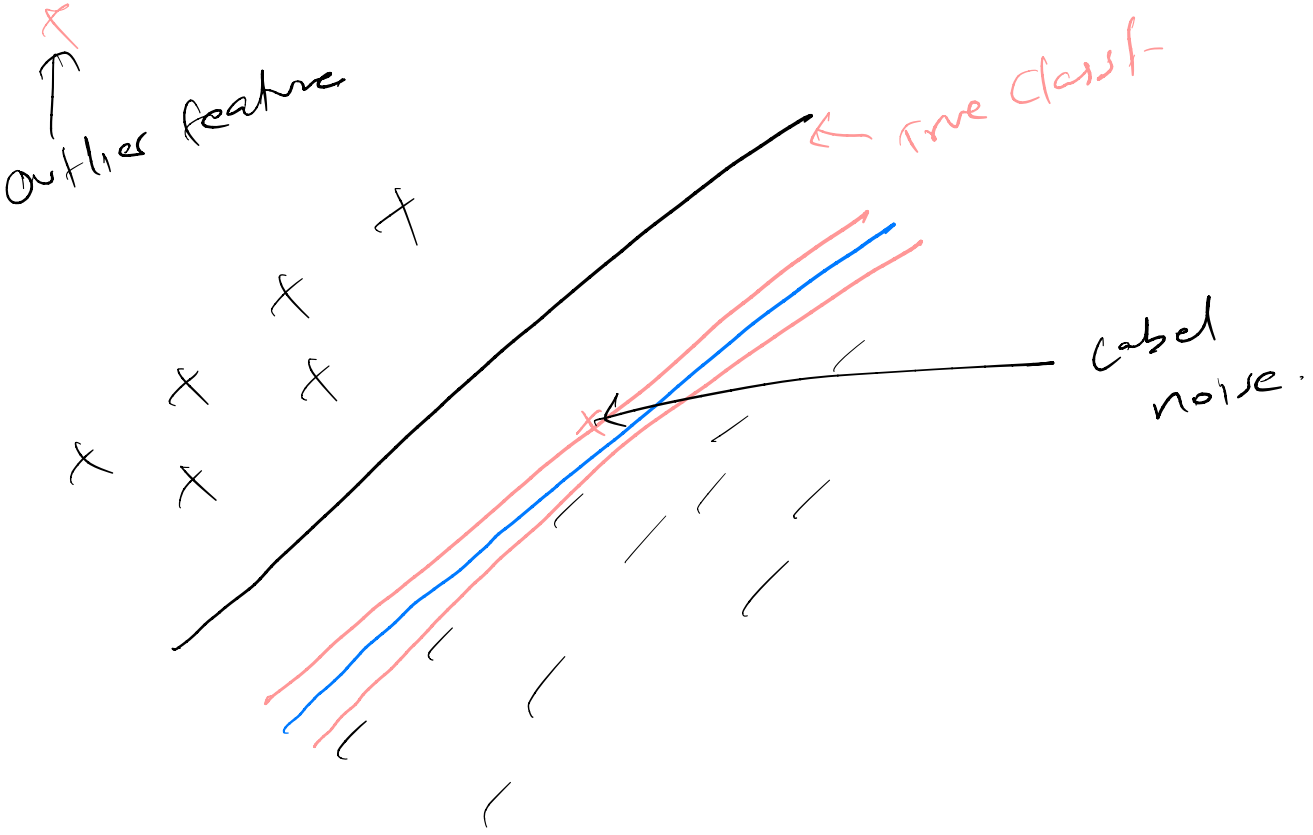
One-Versus-One SVMs (Pair)

- 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


outlier feature
true classif-
label
noise.