UVA CS 4501: Machine Learning

Lecture 16: Support Vector Machine

Dr. Yanjun Qi

University of Virginia
Department of Computer Science

Where are we ? ➔
Five major sections of this course

- ~~Regression (supervised)~~
- Classification (supervised)
- Unsupervised models
- Learning theory
- Graphical models

Today

- Support Vector Machine (SVM)
  - ✓ History of SVM
  - ✓ Large Margin Linear Classifier
  - ✓ Define Margin (M) in terms of model parameter
  - ✓ Optimization to learn model parameters (w, b)
  - ✓ Linearly Non-separable case
  - ✓ Optimization with dual form
  - ✓ Nonlinear decision boundary
  - ✓ Multiclass SVM

History of SVM

- SVM is inspired from statistical learning theory [3]
- SVM was first introduced in 1992 [1]

- SVM becomes popular because of its success in handwritten digit recognition (1994)
  - 1.1% test error rate for SVM.
  - The same as the error rates of a carefully constructed neural network, LeNet 4.
    - Section 5.11 in [2] or the discussion in [3] for details
- Regarded as an important example of "kernel methods", arguably the hottest area in machine learning 20 years ago

[1] B.E. Boser *et al*. A Training Algorithm for Optimal Margin Classifiers. Proceedings of the Fifth Annual Workshop on Computational Learning Theory 5 144-152, Pittsburgh, 1992.
[2] L. Bottou *et al*. Comparison of classifier methods: a case study in handwritten digit recognition. Proceedings of the 12th IAPR International Conference on Pattern Recognition, vol. 2, pp. 77-82, 1994.
[3] V. Vapnik. The Nature of Statistical Learning Theory. 2nd edition, Springer, 1999.

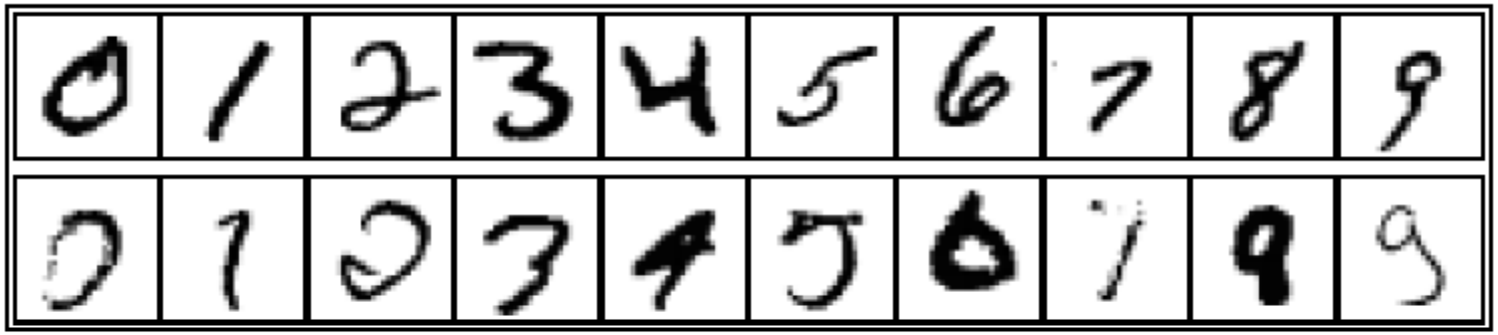
Applications of SVMs

- Computer Vision
- Text Categorization
- Ranking (e.g., Google searches)
- Handwritten Character Recognition
- Time series analysis
- Bioinformatics
- …….

→Lots of very successful applications!!!

Handwritten digit recognition

→ MNIST (SVM) 1994

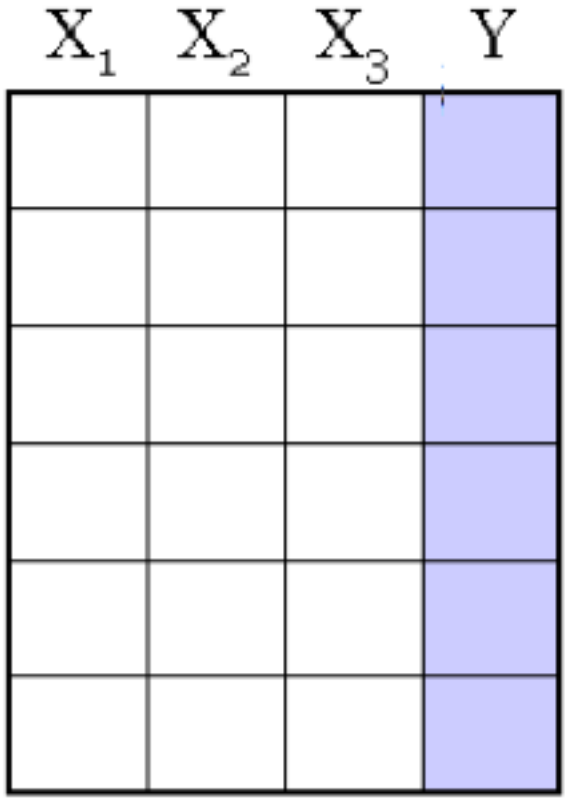


3-nearest-neighbor = 2.4% error

400–300–10 unit MLP = 1.6% error

LeNet: 768–192–30–10 unit MLP = 0.9% error

best (kernel machines, vision algorithms) ≈ 0.6% error



A Dataset for binary/ classification

$$f : X \longrightarrow Y$$

Output as Binary Class: only two possibilities

- **Data***/points/instances/examples/samples/records*: [rows]
- **Features***/attributes/dimensions/independent variables/covariates/predictors/regressors*: [columns, except the last]
- **Target***/outcome/response/label/dependent variable*: special column to be predicted [last column]

Affine Hyperplanes

- https://en.wikipedia.org/wiki/Hyperplane
- Any hyperplane can be given in coordinates as the solution of a single linear (algebraic) equation of degree 1.

$[a_1x_1 + a_2x_2 + a_3x_3 + \cdots + a_px_p = b]$, at least one $a_i \neq 0$

$\Rightarrow$ e.g. classification Boundary $w^Tx + b = 0$

$\begin{cases} x \in R^p \\ b \in R \end{cases}$

Review : Vector Product, Orthogonal, and Norm

For two vectors *x* and *y*,

$$x^T y$$

is called the (*inner) vector product.*

x and *y* are called *orthogonal* if

$$x^T y = 0$$

The square root of the product of a vector with itself,

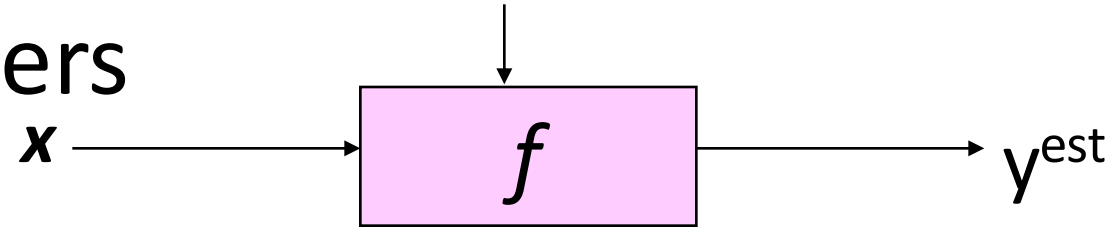
$$\sqrt{x^T x}$$

is called the *2-norm* ($|x|_2$), can also write as $|x|$

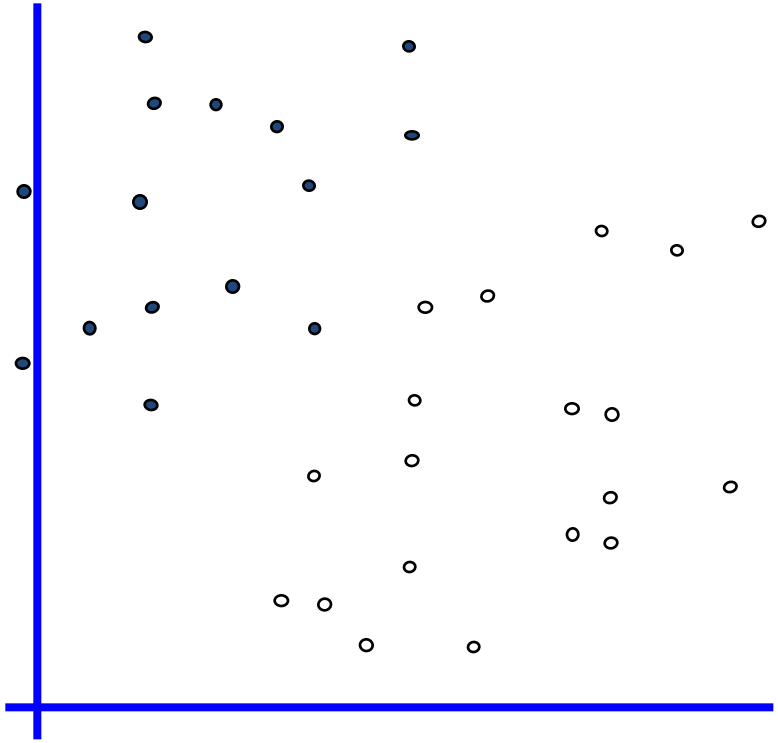
Today

- Support Vector Machine (SVM)
  - ✓ History of SVM
  - ✓ Large Margin Linear Classifier
  - ✓ Define Margin (M) in terms of model parameter
  - ✓ Optimization to learn model parameters (w, b)
  - ✓ Linearly Non-separable case
  - ✓ Optimization with dual form
  - ✓ Nonlinear decision boundary
  - ✓ Multiclass SVM

Linear Classifiers

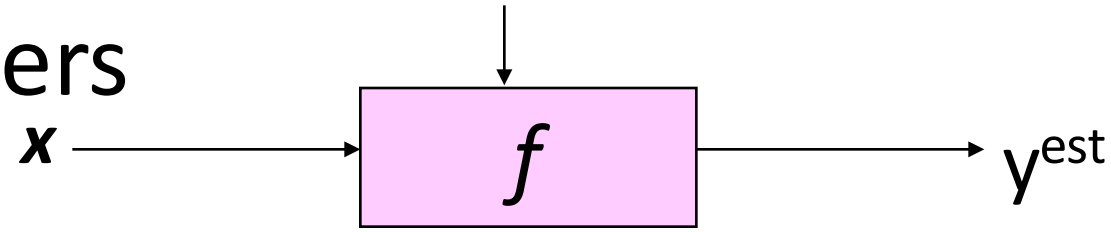


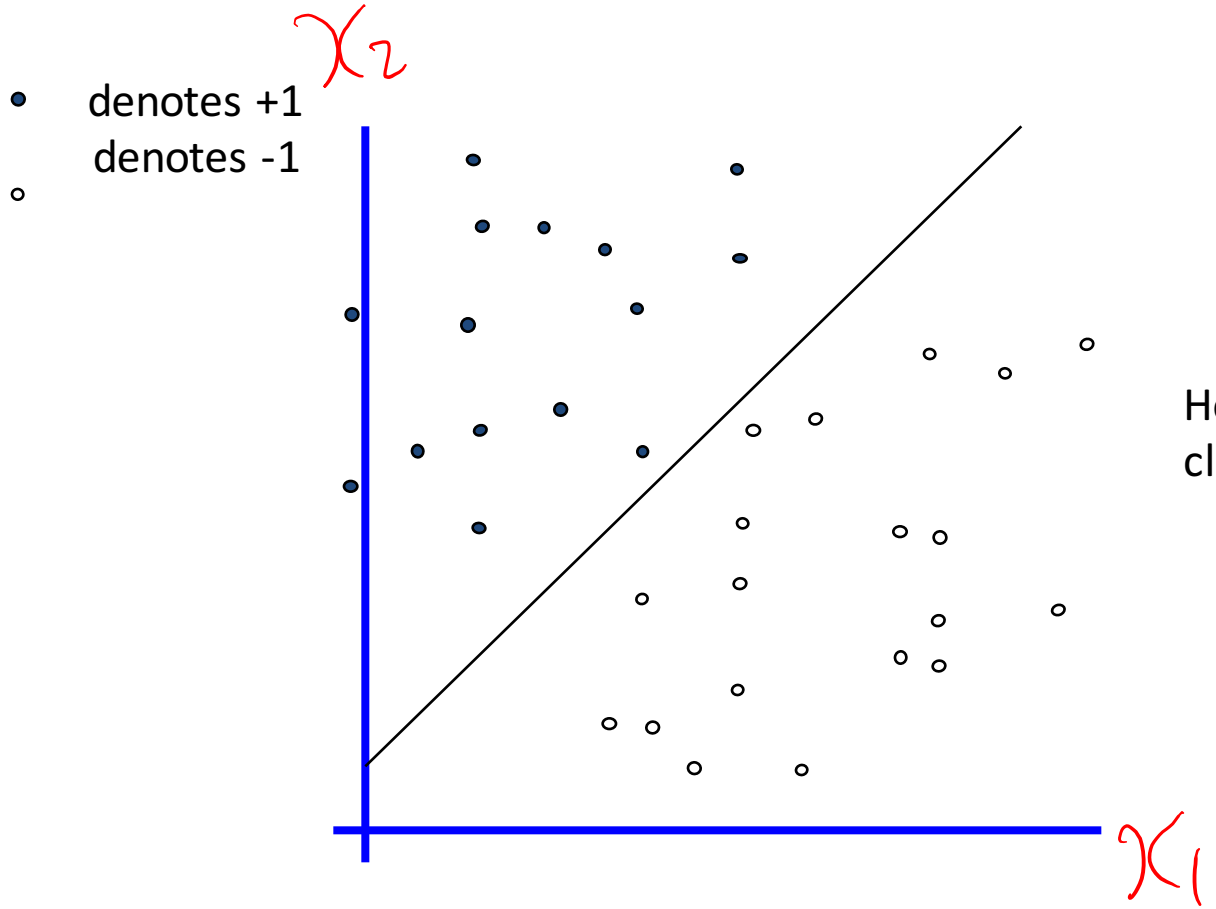
- denotes +1
- denotes -1



How would you classify this data?

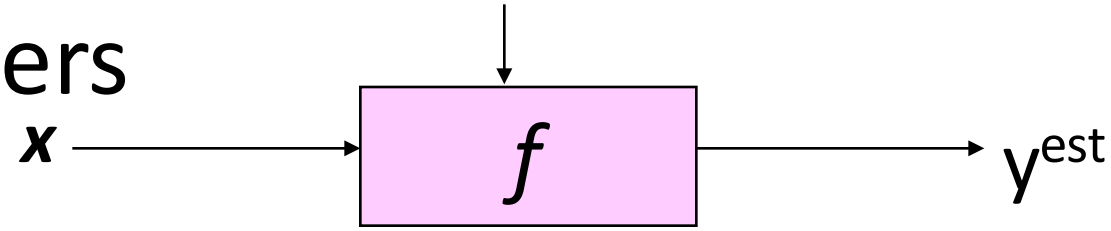
Linear Classifiers



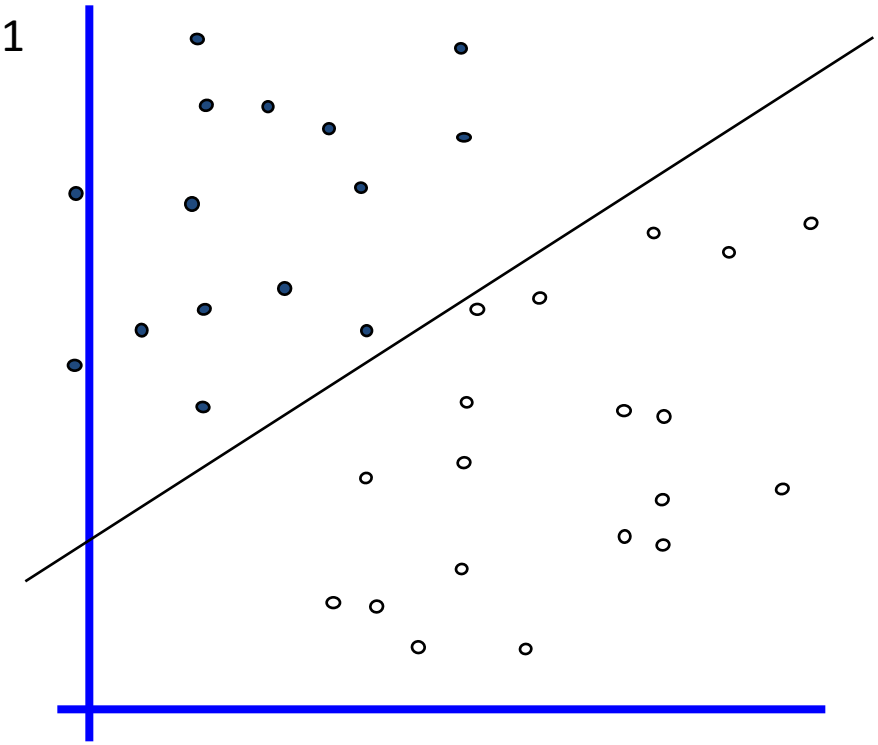


How would you classify this data?

Linear Classifiers

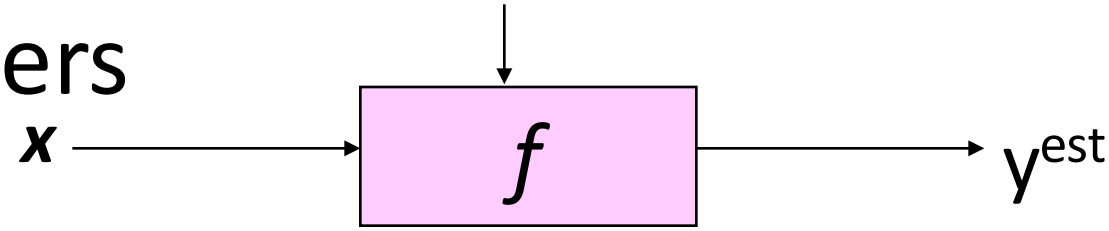


denotes +1
denotes -1

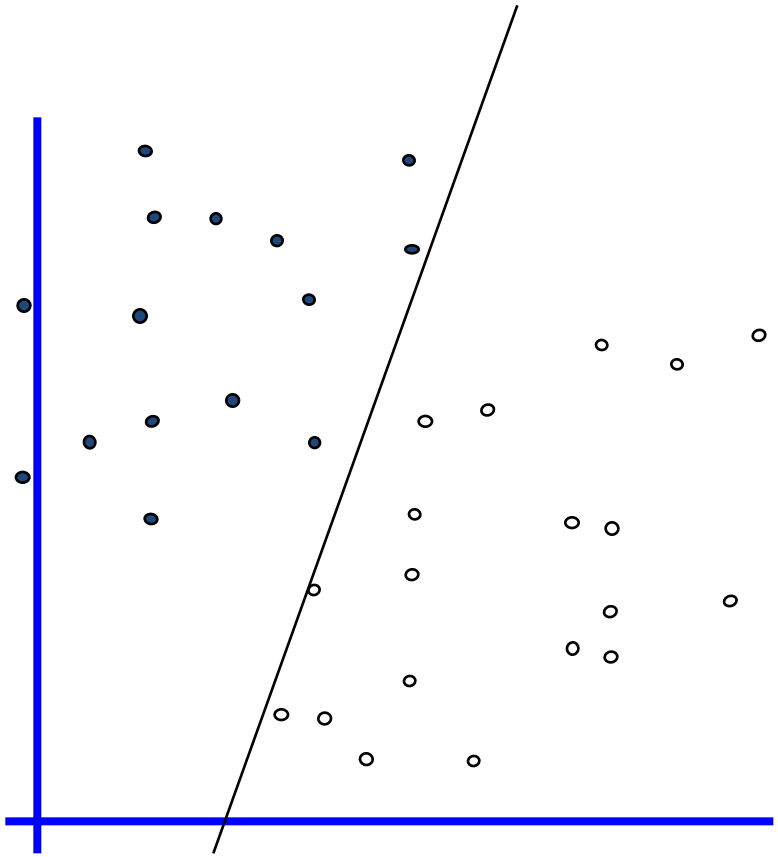


How would you classify this data?

Linear Classifiers

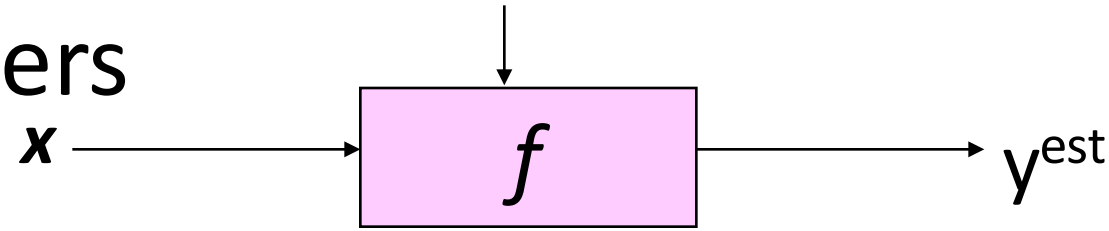


denotes +1
denotes -1

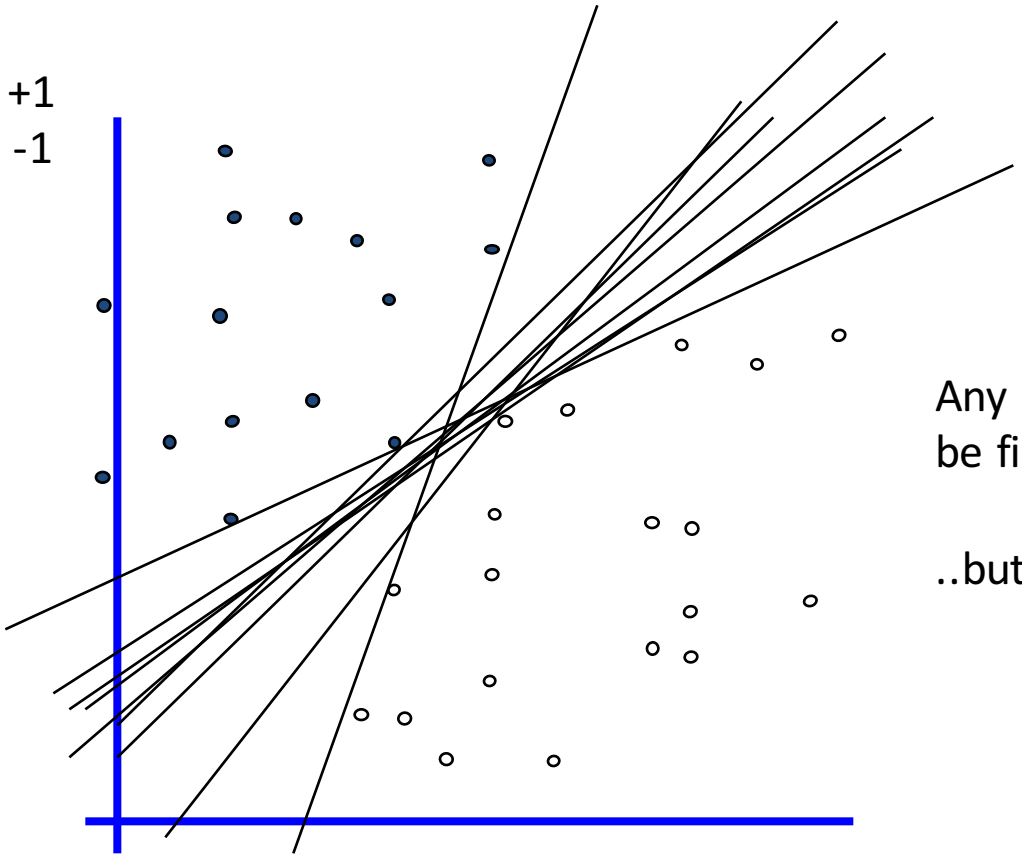


How would you classify this data?

Linear Classifiers



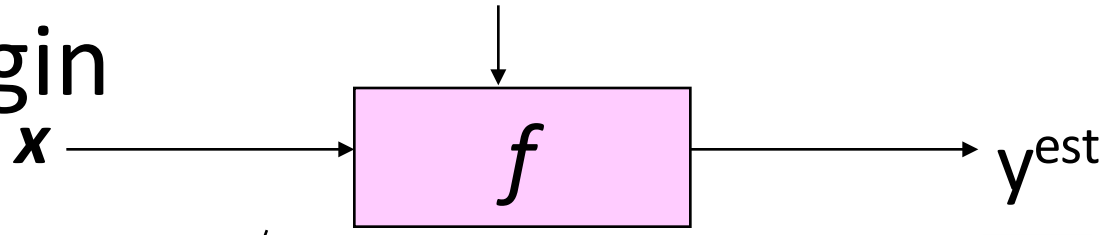
denotes +1
denotes -1

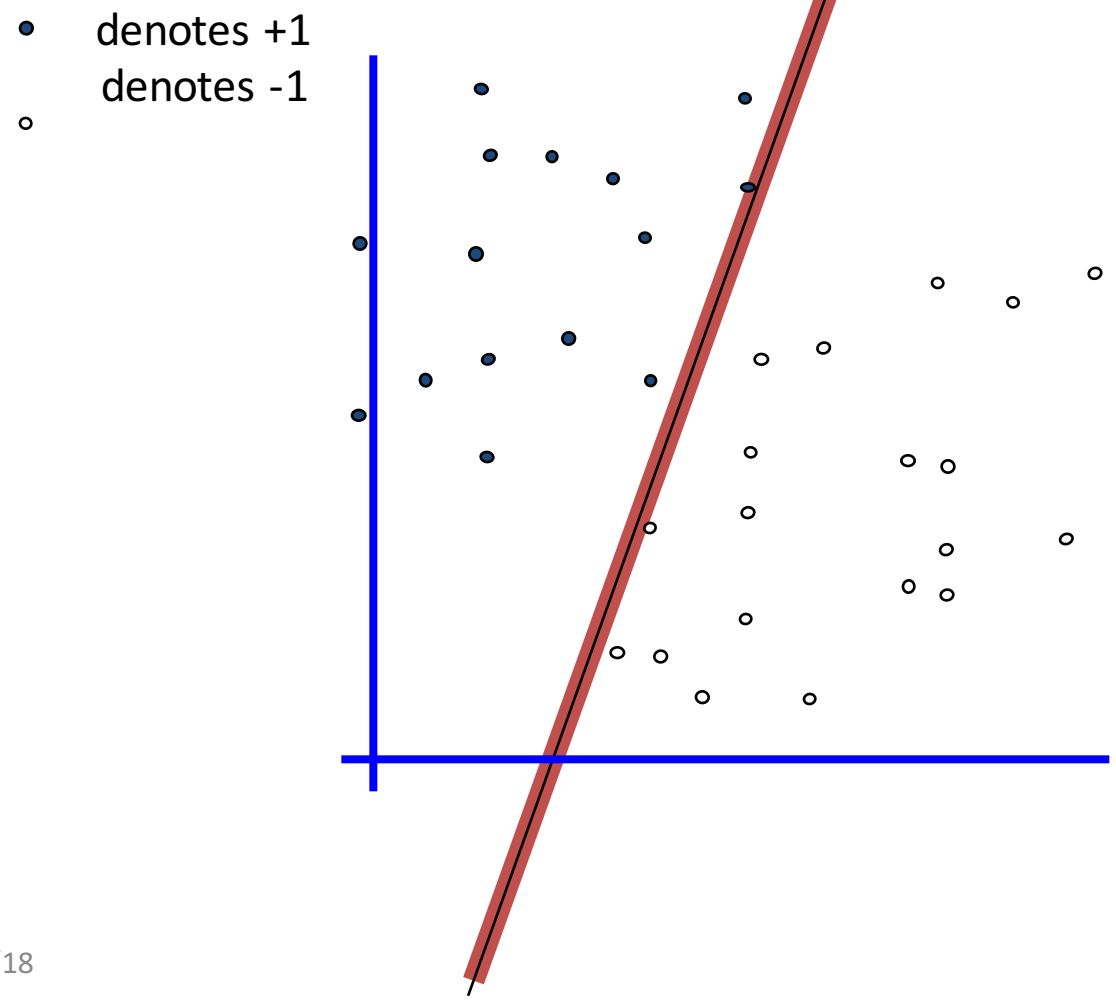


Any of these would be fine..

..but which is best?

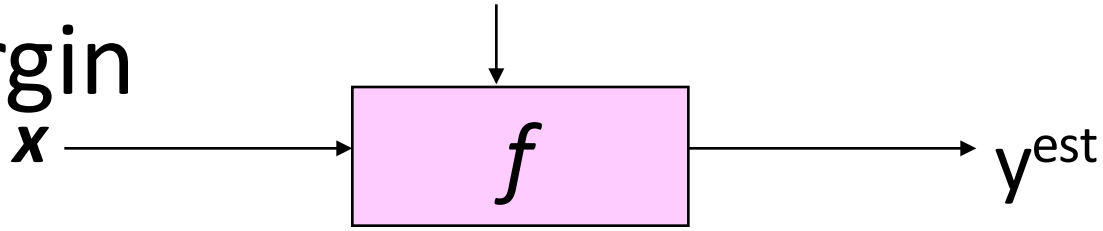
Classifier Margin

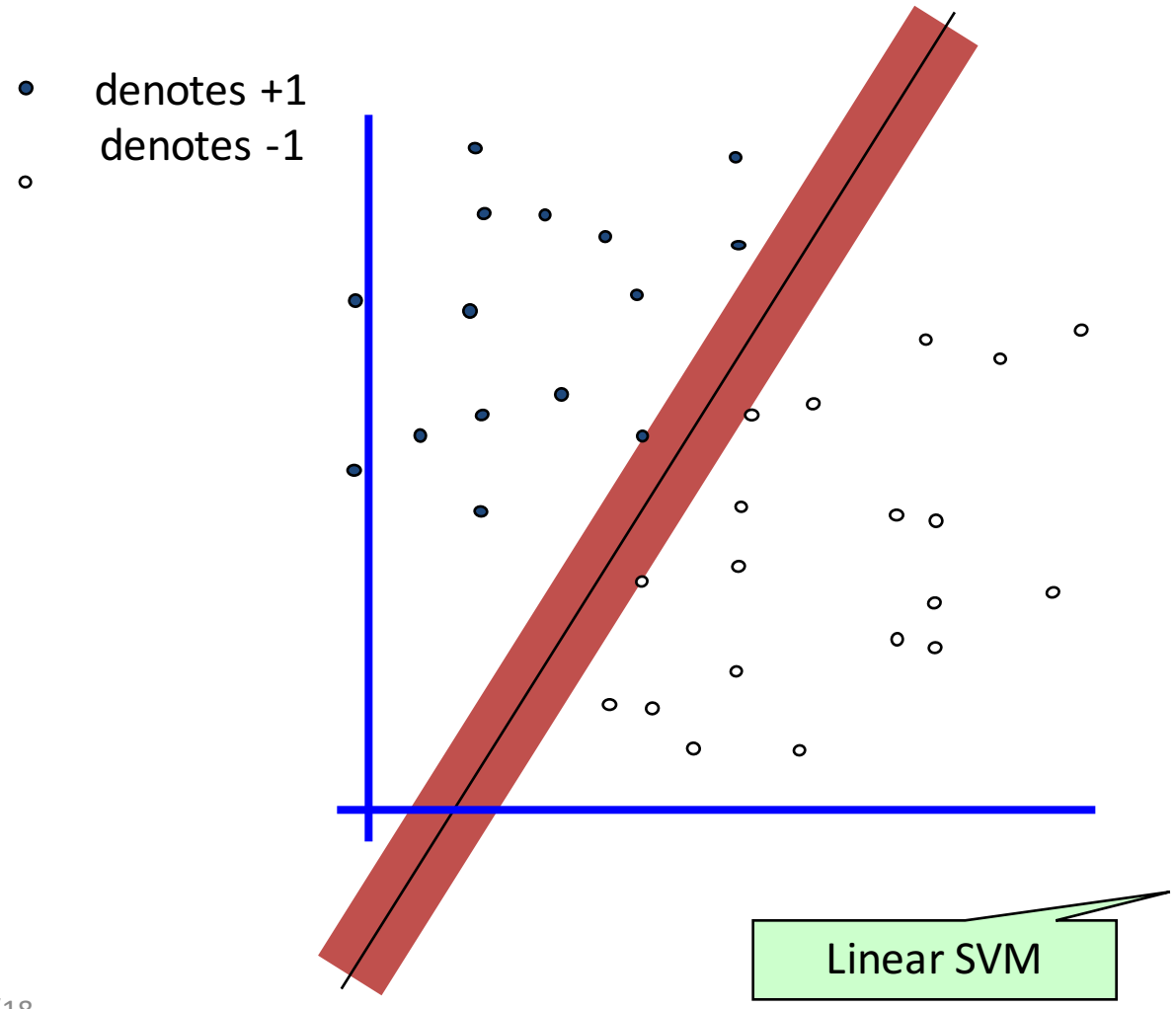




Define the margin of a linear classifier as the width that the boundary could be increased by before hitting a datapoint.

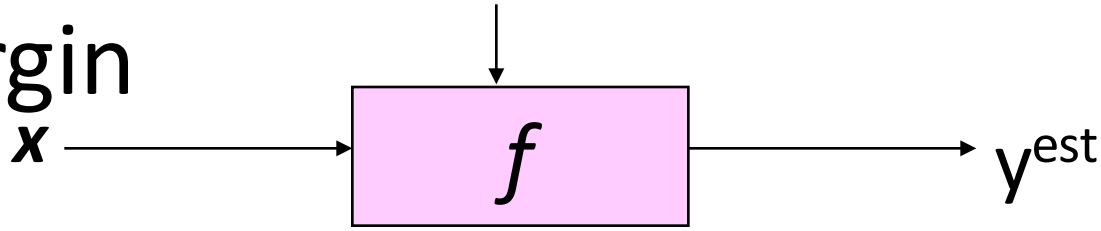
Maximum Margin





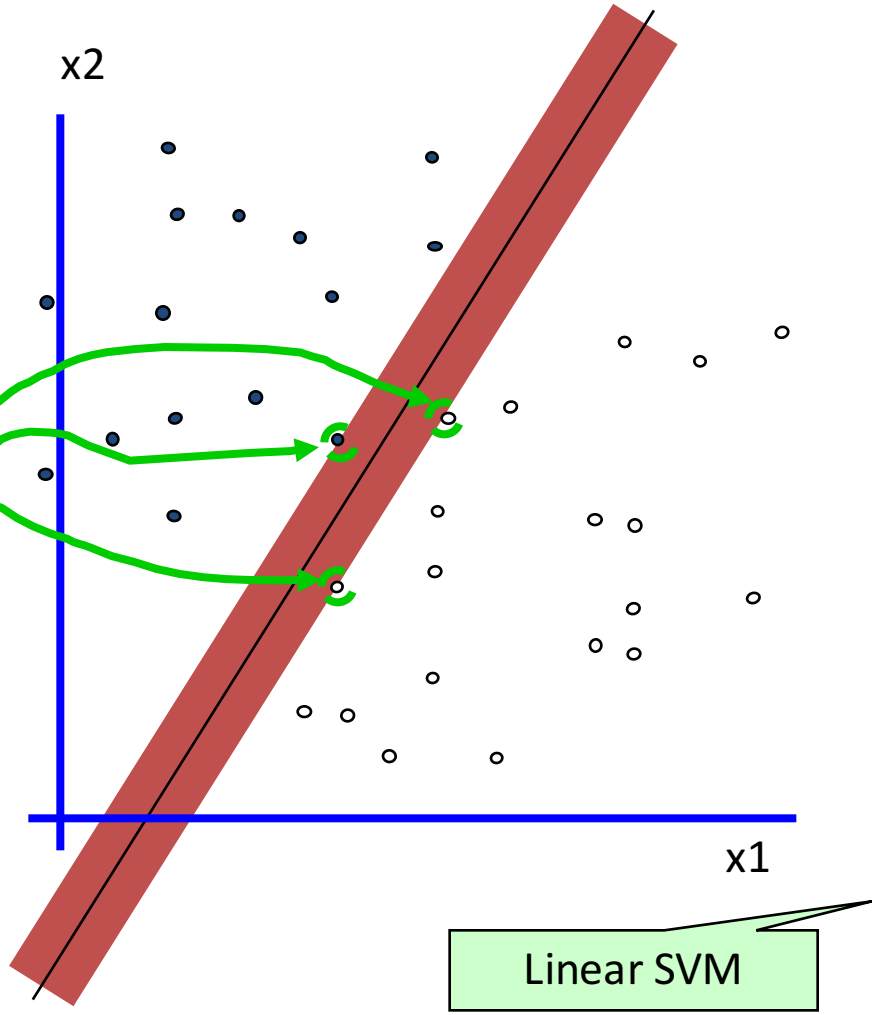
The maximum margin linear classifier is the linear classifier with the, maximum margin.
This is the simplest kind of SVM (Called an LSVM)

Maximum Margin



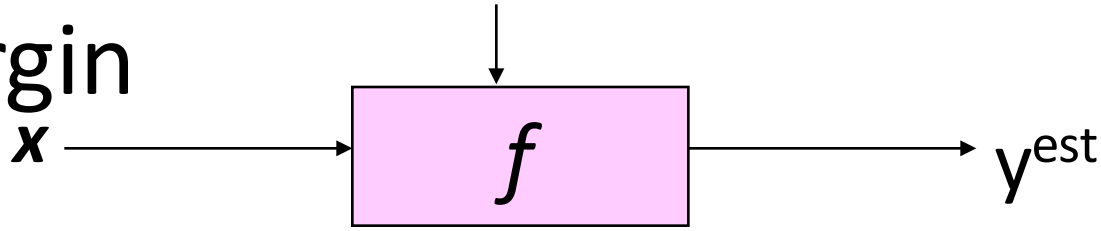
denotes +1
denotes -1

Support Vectors are those datapoints that the margin pushes up against

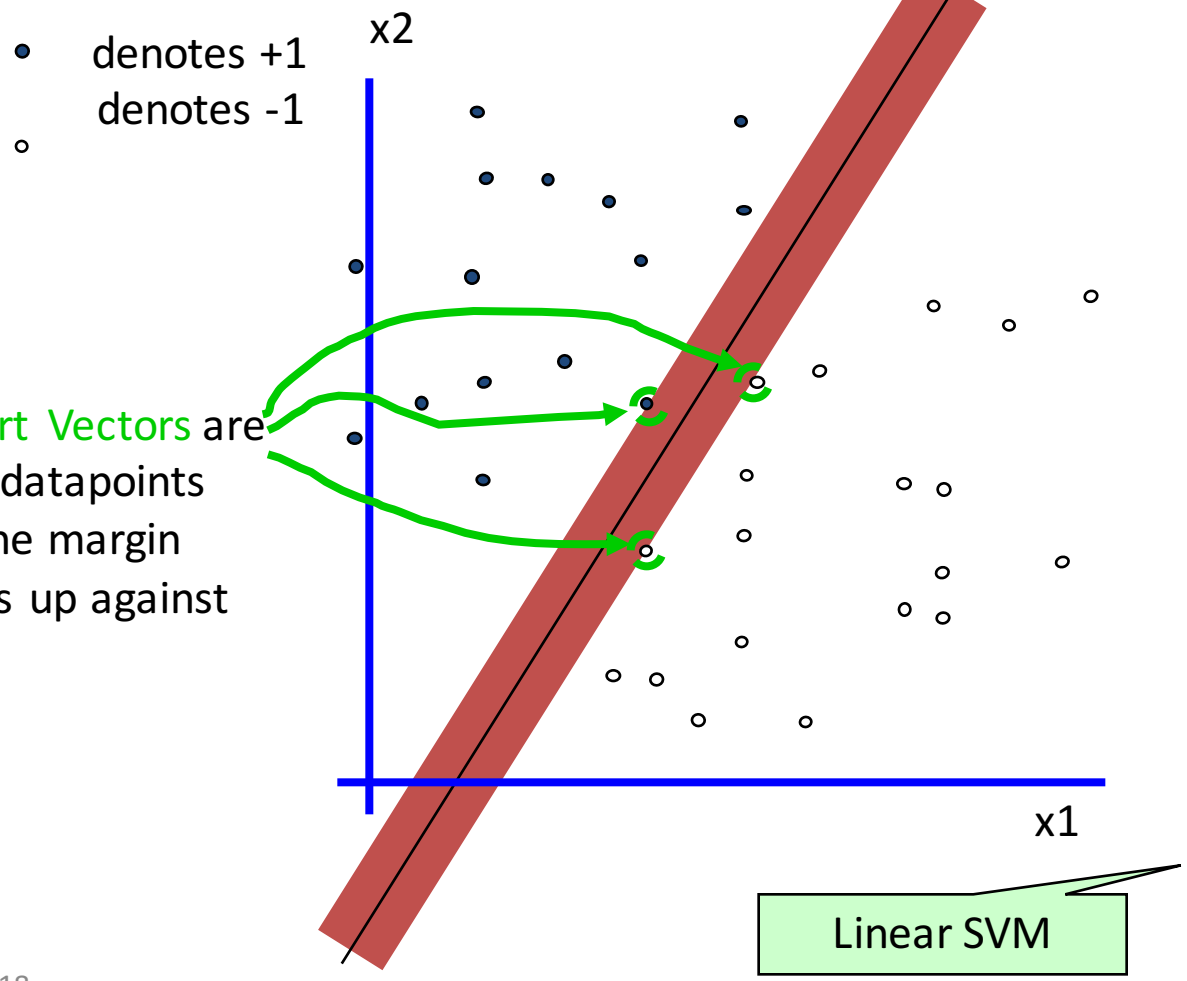


The maximum margin linear classifier is the linear classifier with the, maximum margin.
This is the simplest kind of SVM (Called an LSVM)

Maximum Margin



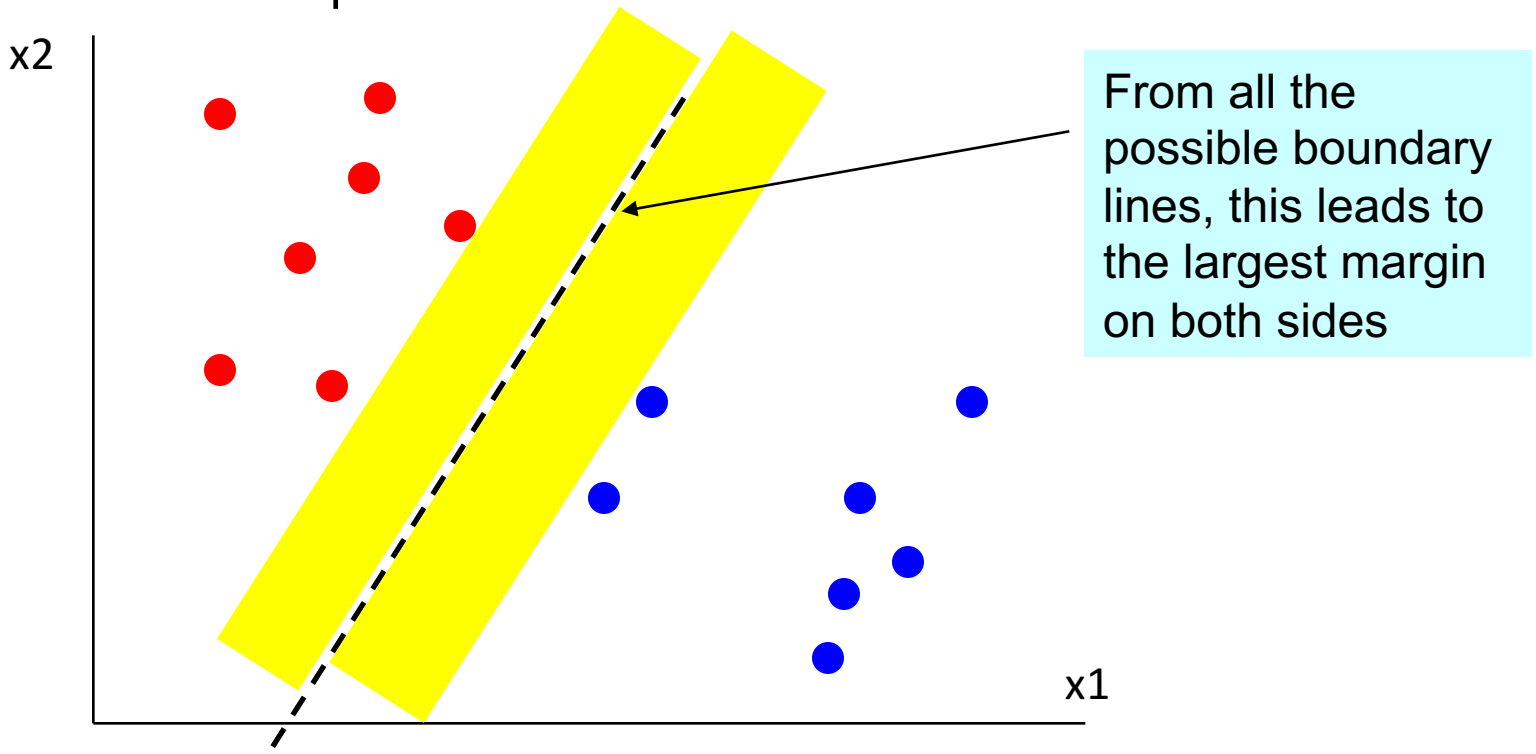
$f(\mathbf{x},\mathbf{w},b) = sign(\mathbf{w}^T\mathbf{x} + b)$



The maximum margin linear classifier is the linear classifier with the, maximum margin.
This is the simplest kind of SVM (Called an LSVM)

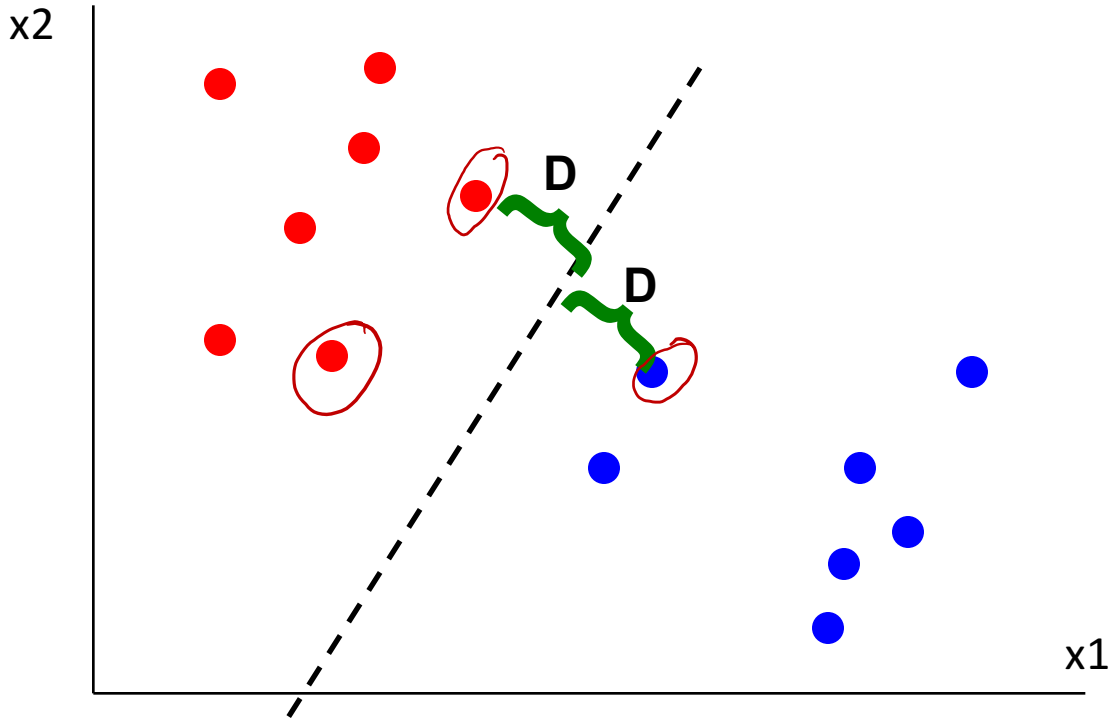
Max margin classifiers

- Instead of fitting all points, focus on boundary points
- Learn a boundary that leads to the largest margin from both sets of points



Max margin classifiers

- Instead of fitting all points, focus on boundary points
- Learn a boundary that leads to the largest margin from points on both sides

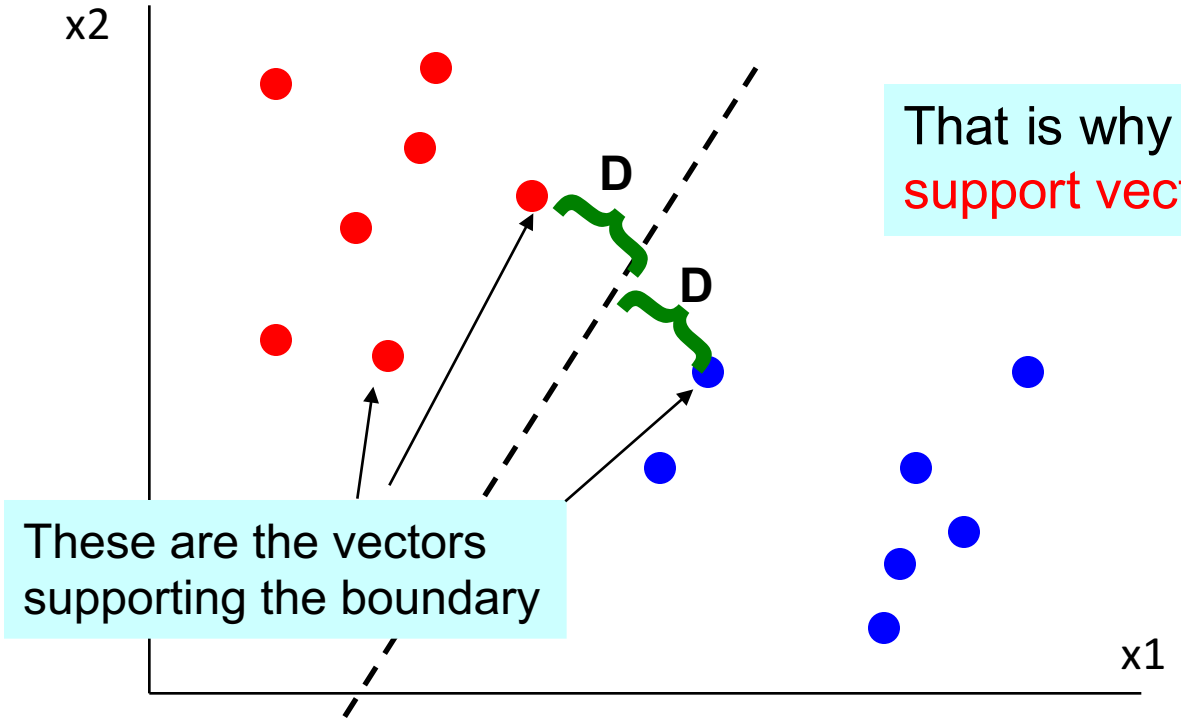


Why MAX margin?

- Intuitive, 'makes sense'
- Some theoretical support
- Works well in practice

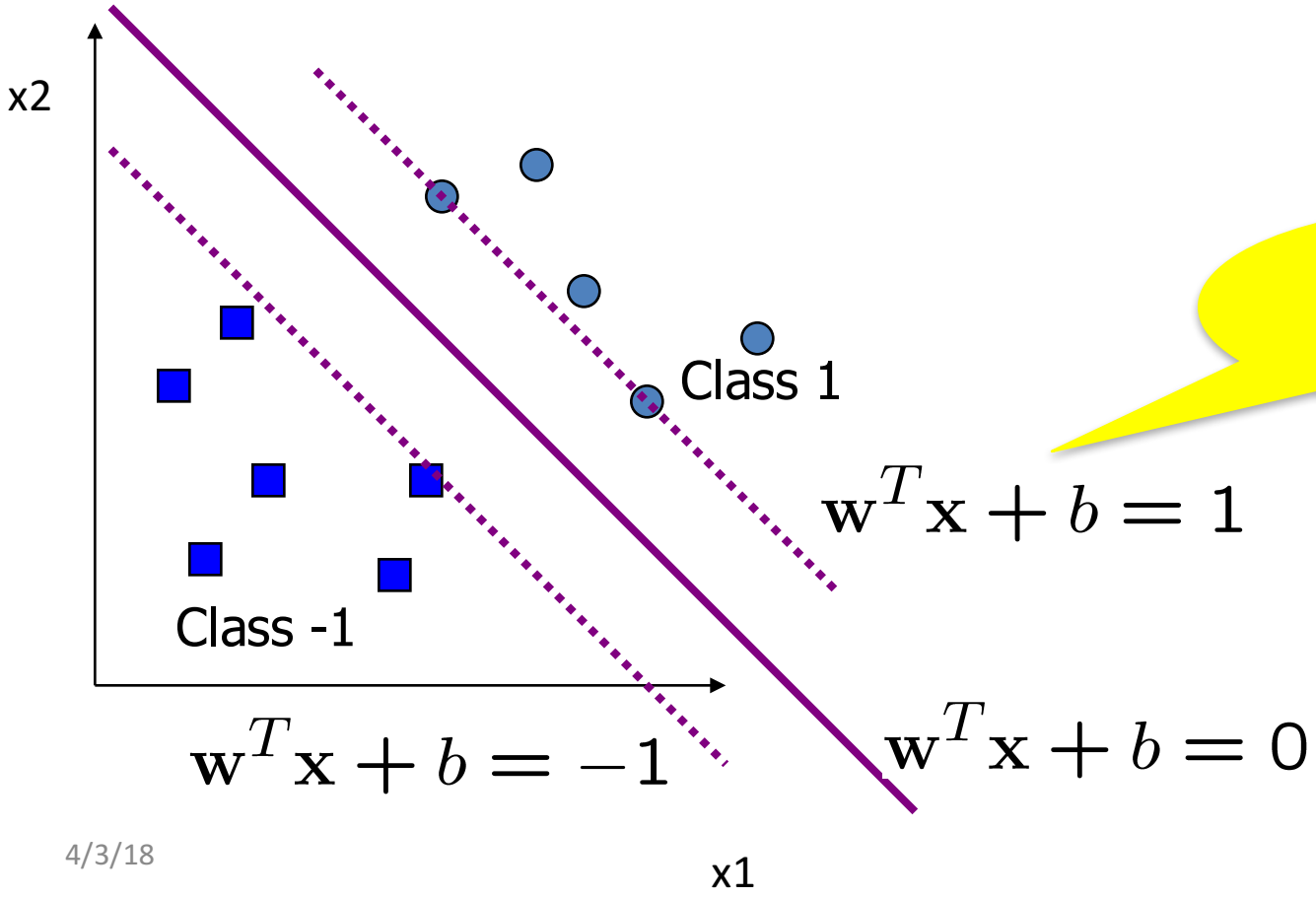
Max margin classifiers

- Instead of fitting all points, focus on boundary points
- Learn a boundary that leads to the largest margin from points on both sides



That is why Also known as linear support vector machines (SVMs)

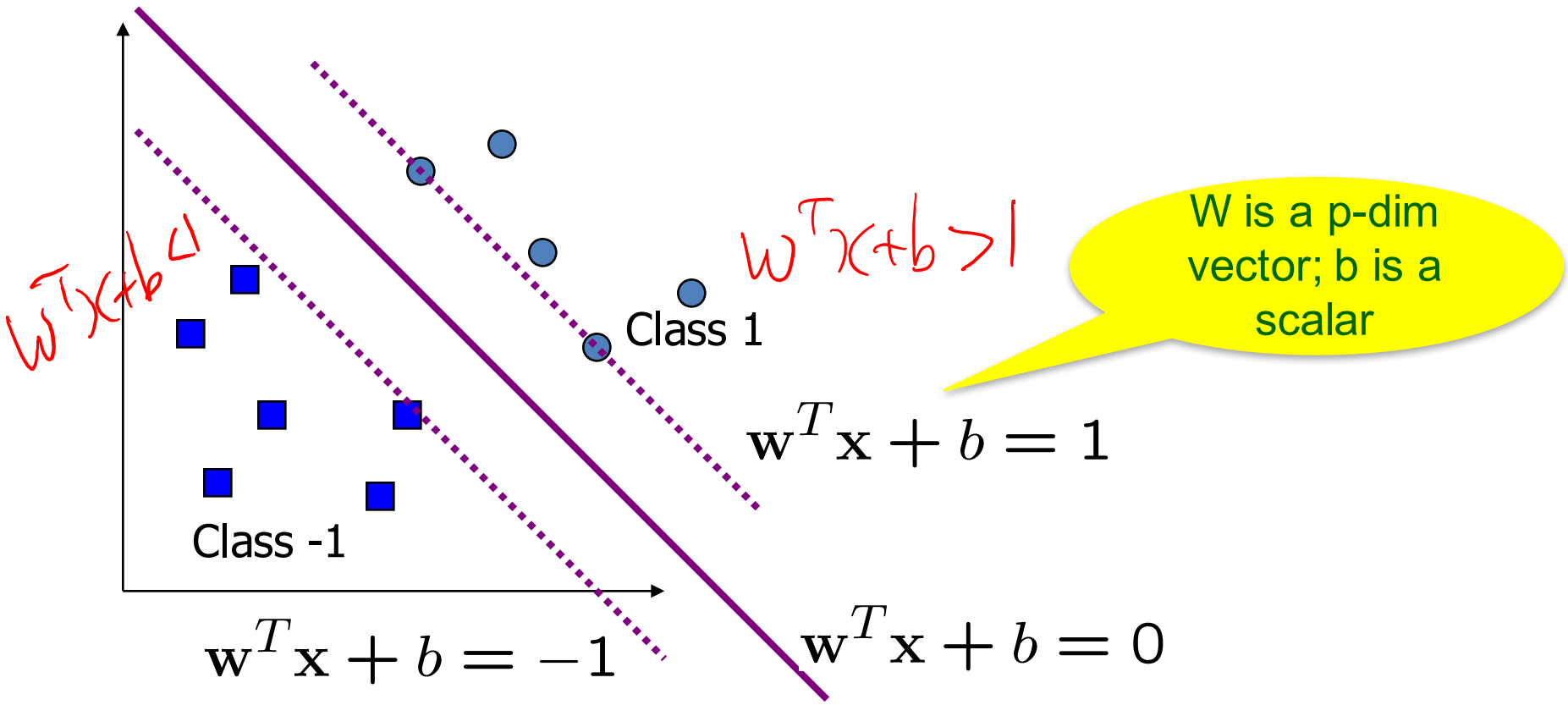
Max-margin & Decision Boundary



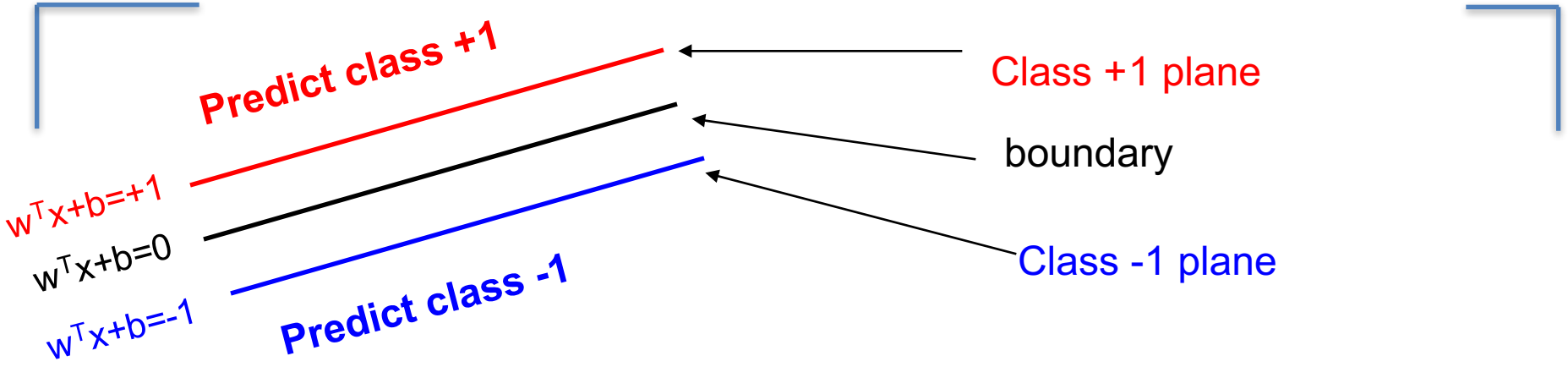
W is a p-dim vector; b is a scalar

Max-margin & Decision Boundary

- The decision boundary should be as far away from the data of both classes as possible



Specifying a max margin classifier

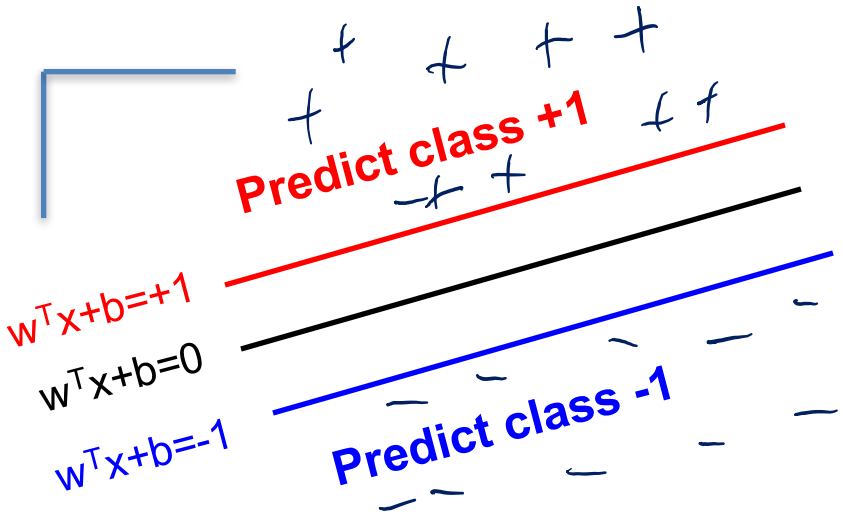


Classify as +1 if $w^T x+b >= 1$

Classify as -1 if $w^T x+b <= -1$

Undefined if $-1 < w^T x+b < 1$

Specifying a max margin classifier



Is the linear separation assumption realistic?

We will deal with this shortly, but lets assume it for now

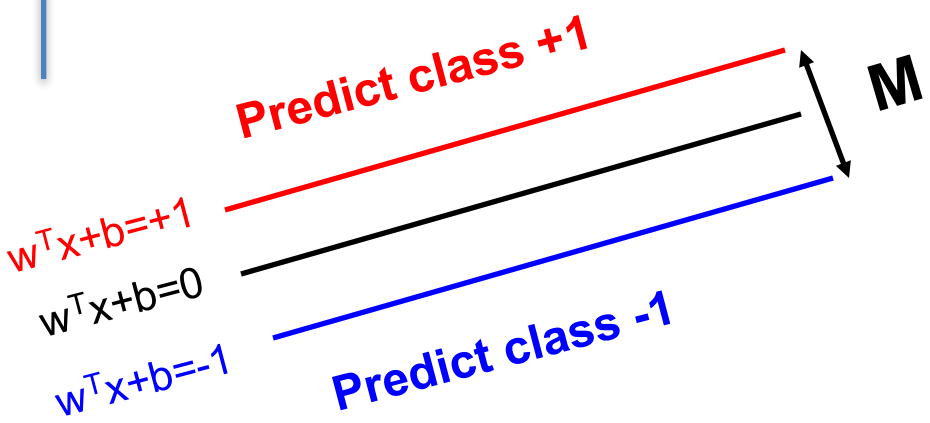
Classify as +1	if	$w^T x+b >= 1$
Classify as -1	if	$w^T x+b <= -1$
Undefined	if	$-1 < w^T x+b < 1$

Now Assuming such lines exist in our train

Today

- ❑ Supervised Classification
- ❑ Support Vector Machine (SVM)
  - ✓ History of SVM
  - ✓ Large Margin Linear Classifier
  - ✓ Define Margin (M) in terms of model parameter
  - ✓ Optimization to learn model parameters (w, b)
  - ✓ Linearly Non-separable case
  - ✓ Optimization with dual form
  - ✓ Nonlinear decision boundary
  - ✓ Multiclass SVM

Maximizing the margin



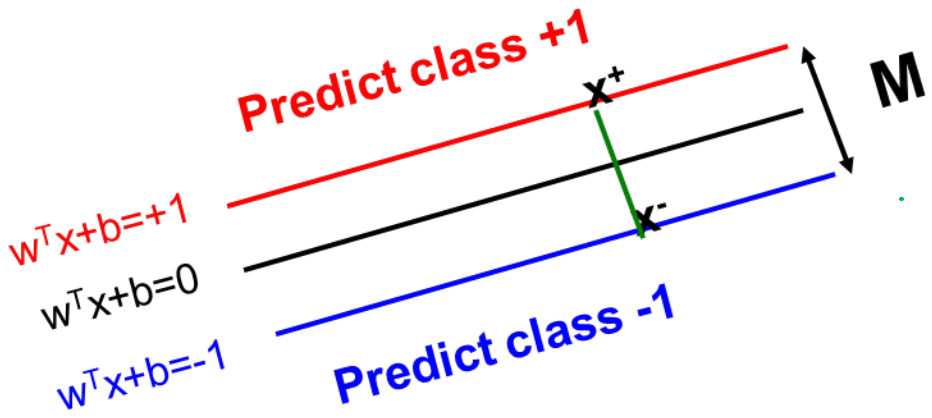
Classify as +1 if $w^T x+b >= 1$

Classify as -1 if $w^T x+b <= -1$

Undefined if $-1 < w^T x+b < 1$

- Lets define the width of the margin by M
- How can we encode our goal of maximizing M in terms of our parameters (w and b)?
- Lets start with a few obsevrations

Margin M



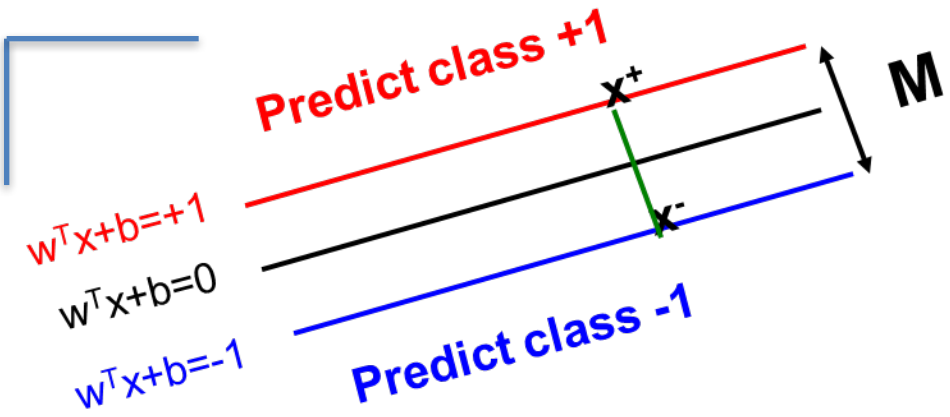
Classify as +1 if $w^Tx+b >= 1$

Classify as -1 if $w^Tx+b <= -1$

Undefined if $-1 < w^Tx+b < 1$

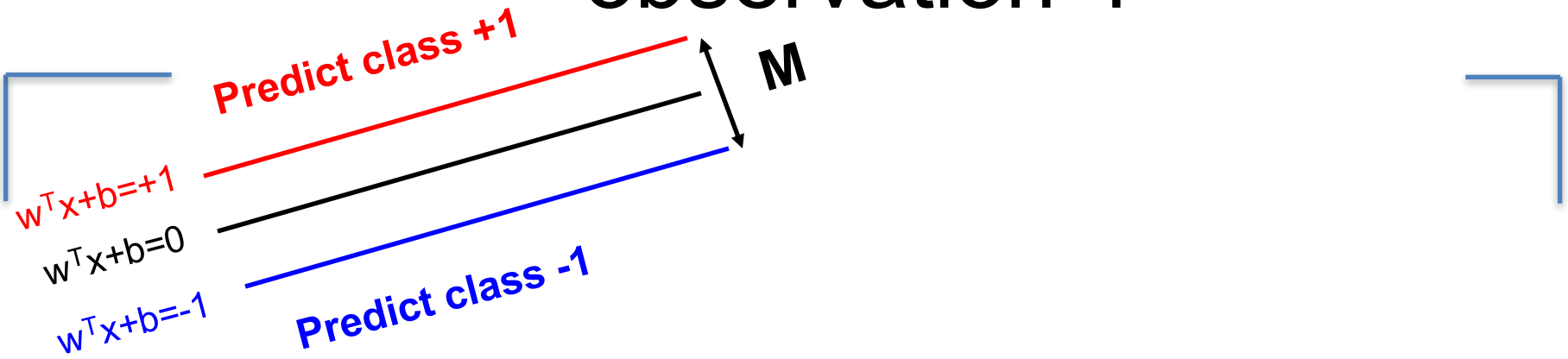
$M = |x^+ - x^-|$ length of vector $(x^+ - x^-)$

$\Rightarrow$ How to represent $(x^+ - x^-)$???



- $w^T x^+ + b = +1$
- $w^T x^- + b = -1$
- $M = | x^+ - x^- | = ?$

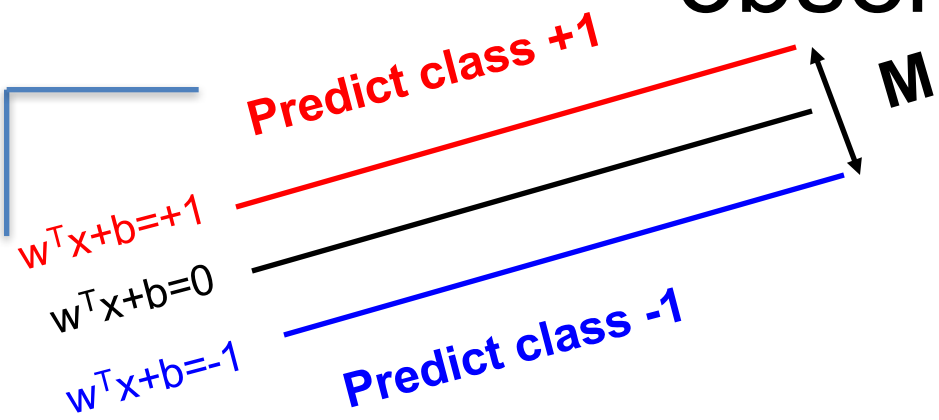
Maximizing the margin: observation-1



- **Observation 1: the vector w is orthogonal to the +1 plane**
- Why?

Corollary: the vector w is orthogonal to the -1 plane

Maximizing the margin: observation-1



Classify as +1 if $w^T x+b >= 1$

Classify as -1 if $w^T x+b <= -1$

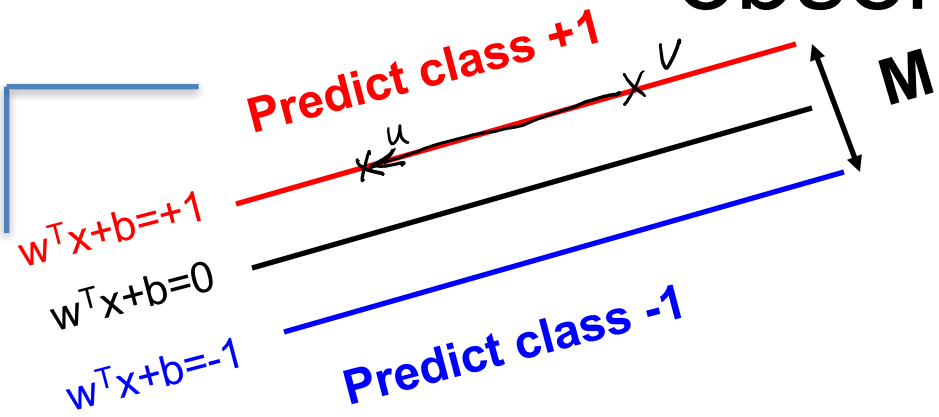
Undefined if $-1 < w^T x+b < 1$

- **Observation 1: the vector w is orthogonal to the +1 plane**
- Why?

Let u and v be two points on the +1 plane, then for the vector defined by u and v we have $w^T(u-v) = 0$

Corollary: the vector w is orthogonal to the -1 plane

Maximizing the margin: observation-1



Classify as +1 if $w^T x+b >= 1$

Classify as -1 if $w^T x+b <= -1$

Undefined if $-1 < w^T x+b < 1$

- **Observation 1: the vector w is orthogonal to the +1 plane**
- Why?

→ vector $(u-v)$ shown above

→ $w^T(u-v) = w^T u - w^T v = (1-b) - (1-b) = 0$

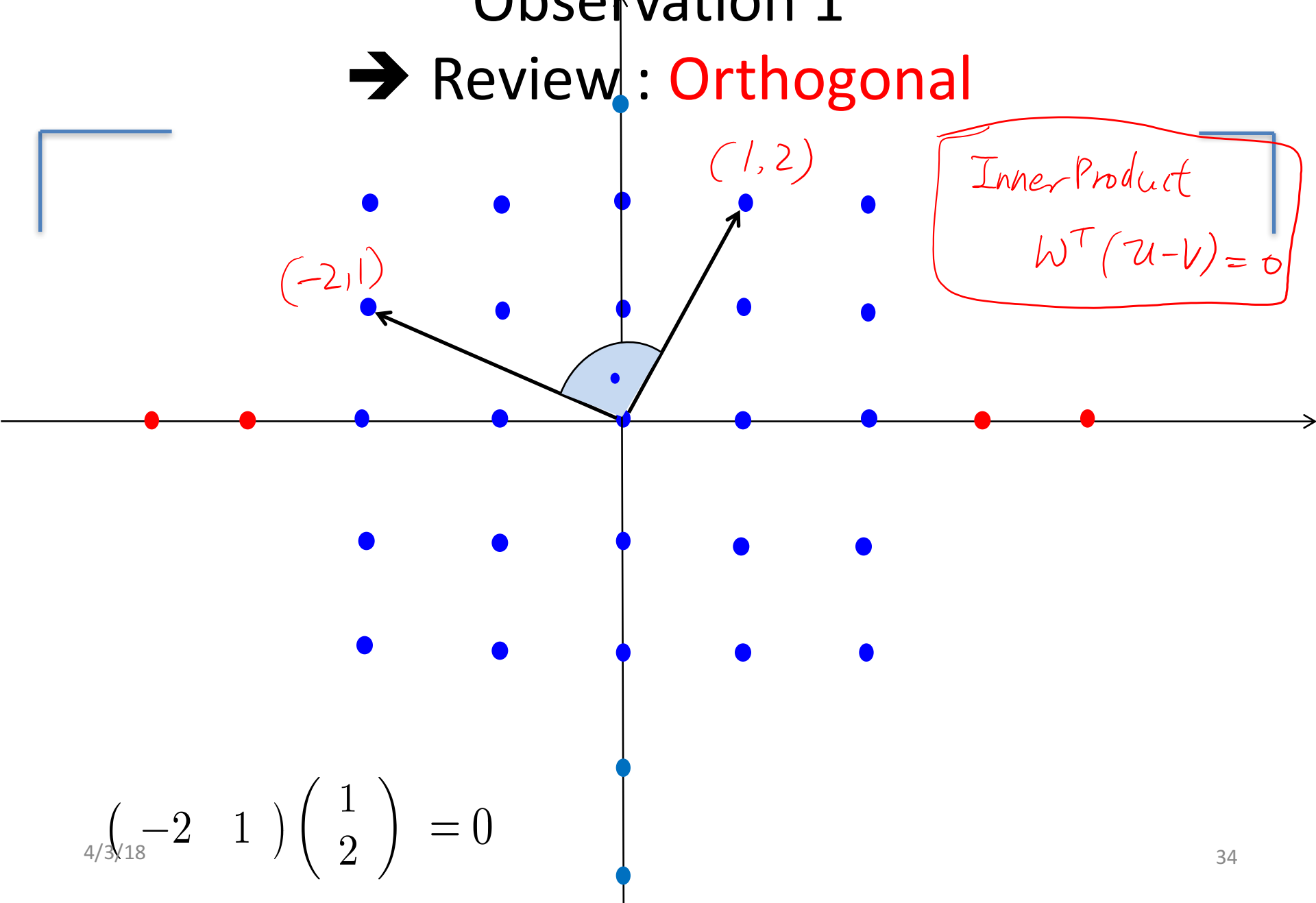
⇒ w orthogonal to $(u-v)$

Let u and v be two points on the +1 plane, then for the vector defined by u and v we have $w^T(u-v) = 0$

Corollary: the vector w is orthogonal to the -1 plane

Observation 1

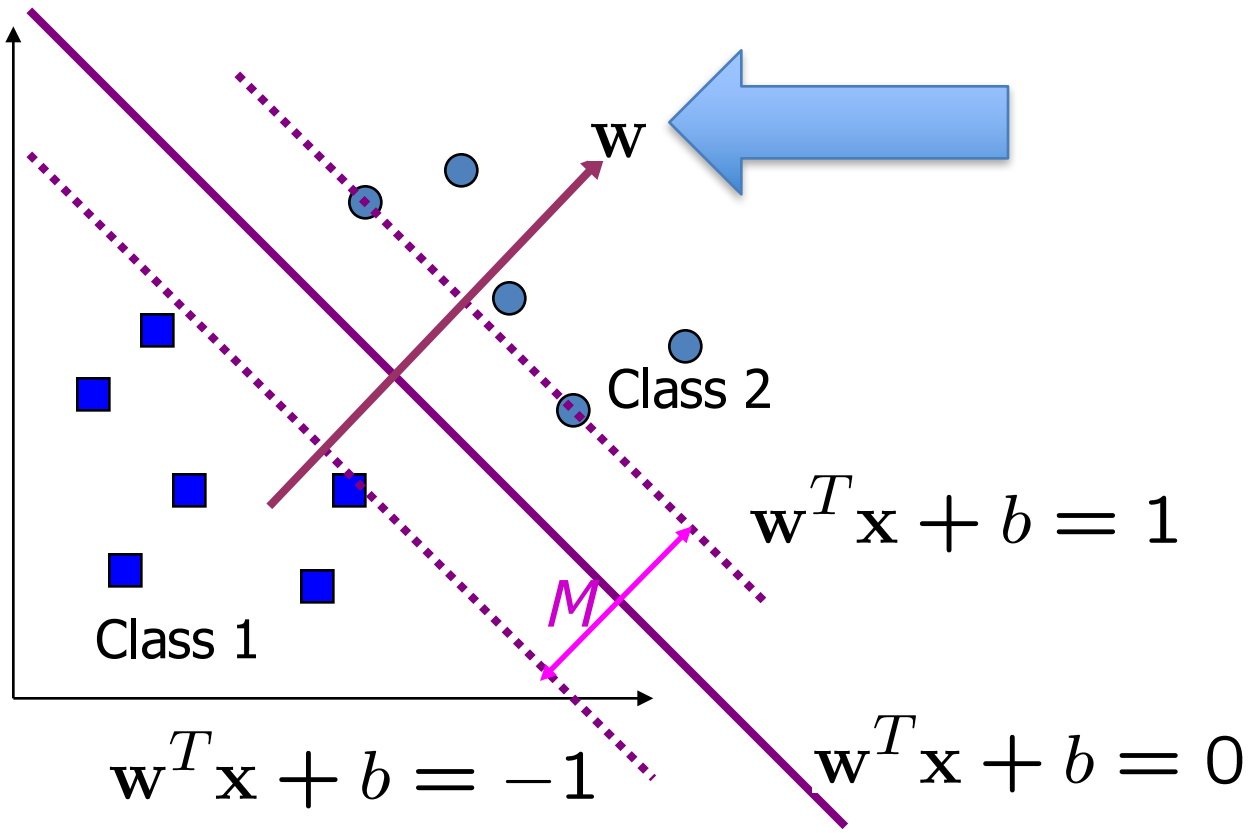
➔ Review : Orthogonal



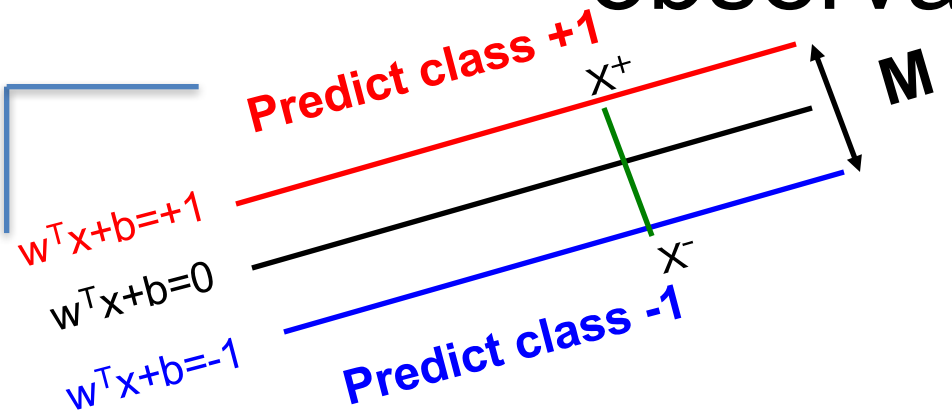
$$\begin{pmatrix} -2 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 2 \end{pmatrix} = 0$$

Maximizing the margin: observation-1

- **Observation 1: the vector w is orthogonal to the +1 plane**



Maximizing the margin: observation-2



Classify as +1 if $w^Tx+b >= 1$

Classify as -1 if $w^Tx+b <= -1$

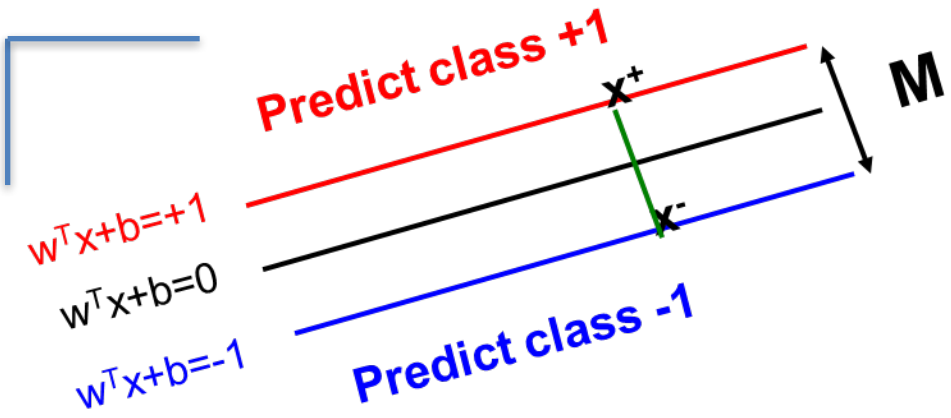
Undefined if $-1 < w^Tx+b < 1$

- Observation 1: the vector w is orthogonal to the +1 and -1 planes
- Observation 2: if x^+ is a point on the +1 plane and x^- is the closest point to x^+ on the -1 plane then

$$x^+ = \lambda w + x^-$$

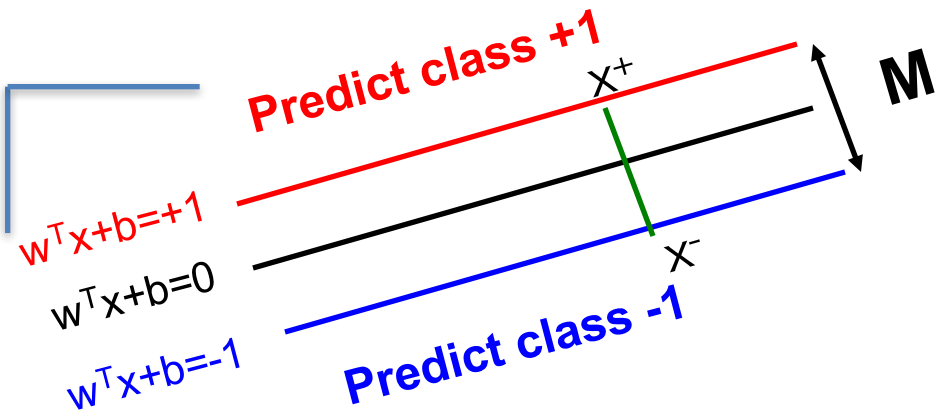
Since w is orthogonal to both planes we need to 'travel' some distance along w to get from x^+ to x^-

Putting it together



- $w^T x^+ + b = +1$
- $w^T x^- + b = -1$
- $M = | x^+ - x^- | = ?$
- $x^+ = \lambda w + x^-$

Putting it together



- $w^T x^+ + b = +1$
- $w^T x^- + b = -1$
- $x^+ = \lambda w + x^-$
- $| x^+ - x^- | = M$

We can now define M in terms of w and b

$$M = |x^+ - x^-|$$
$$= |\lambda w|$$
$$= \lambda |w|$$
$$= \lambda \sqrt{w^T w}$$
$$= \frac{2}{w^T w} \sqrt{w^T w}$$
$$\left[= \frac{2}{\sqrt{w^T w}} \right]$$

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

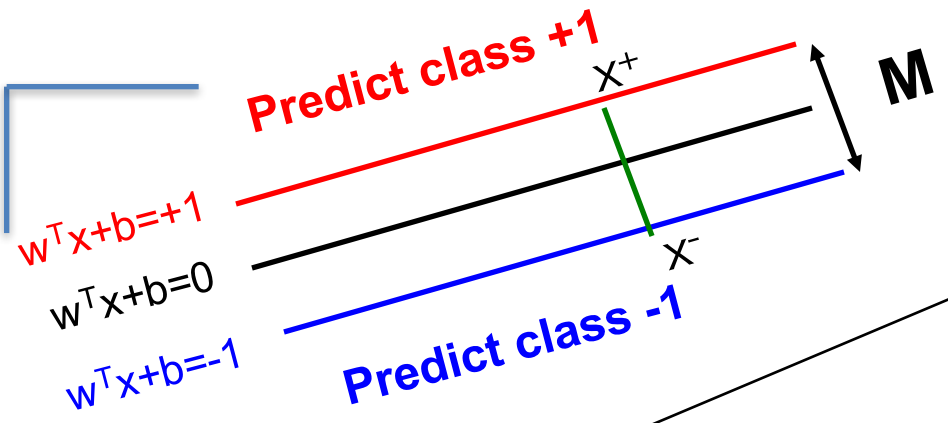
$$w^T (\lambda w + x^-) + b = +1$$

$$\lambda w^T w + \underbrace{w^T x^- + b}_{\Rightarrow -1} = 1$$

$$\lambda w^T w = 2$$

$$\Rightarrow \lambda = \frac{2}{w^T w}$$

Putting it together



- $w^T x^+ + b = +1$
- $w^T x^- + b = -1$
- $x^+ = \lambda w + x^-$
- $| x^+ - x^- | = M$

We can now define M in terms of w and b

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

=>

$w^T (\lambda w + x^-) + b = +1$

=>

$w^T x^- + b + \lambda w^T w = +1$

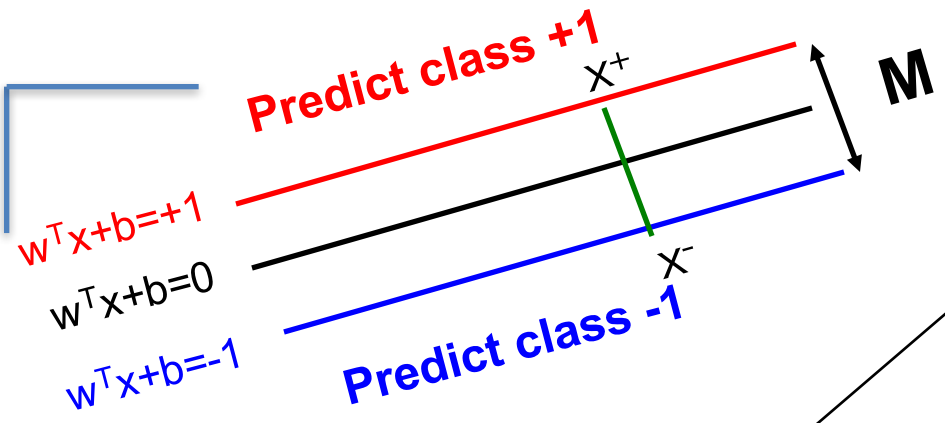
=>

$-1 + \lambda w^T w = +1$

=>

$\lambda = 2/w^T w$

Putting it together



- $w^T x^+ + b = +1$
- $w^T x^- + b = -1$
- $x^+ = \lambda w + x^-$
- $| x^+ - x^- | = M$
- $\lambda = 2/w^T w$

We can now define M in terms of w and b

$M = |x^+ - x^-|$

=>

$$M = |\lambda w| = \lambda |w| = \lambda \sqrt{w^T w}$$

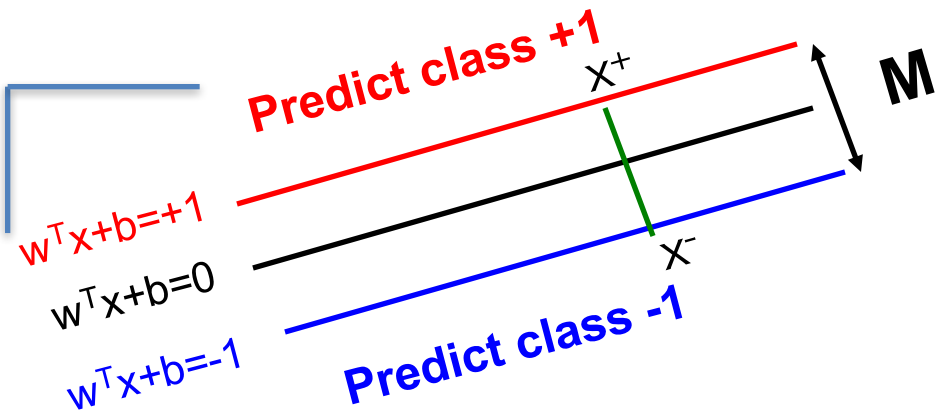
=>

$$M = 2\frac{\sqrt{w^T w}}{w^T w} = \frac{2}{\sqrt{w^T w}}$$

max (M)

$\Rightarrow$ min ($w^T w$)

Finding the optimal parameters



$$M = \frac{2}{\sqrt{w^T w}} = \frac{2}{||w||}$$

We can now search for the optimal parameters by finding a solution that:

1. Correctly classifies all points
2. Maximizes the margin (or equivalently minimizes $w^T w$)

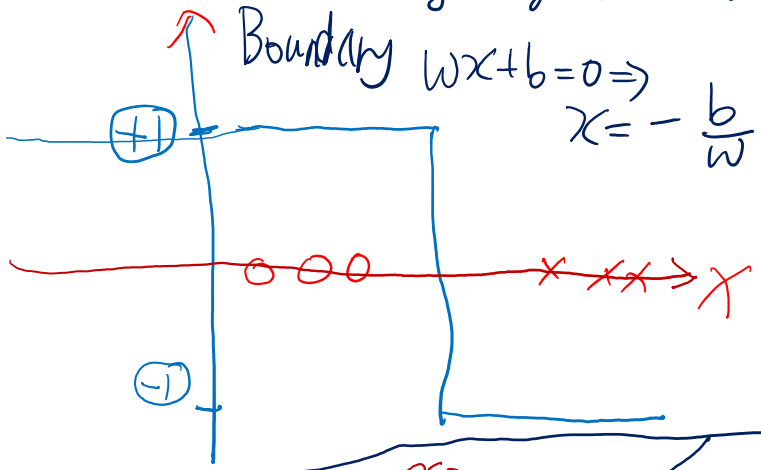
Several optimization methods can be used:
Gradient descent, OR SMO (see extra slides)

Binary Classification

$y \in \{-1, 1\}$

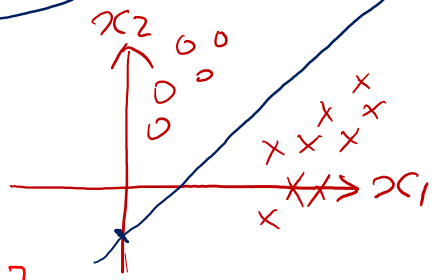
if $\Rightarrow$ 1D $[x \in R]$

$y = sign(wx+b)$



if $\Rightarrow$ 2D $[x \in R^2]$

Boundary $w^T x + b = 0$



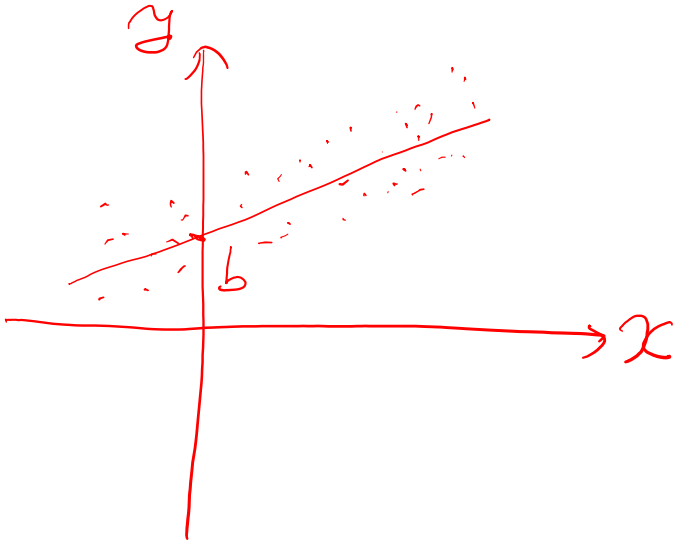
if $\Rightarrow$ P dim $[x \in R^P]$

Boundary : hyperplane

Regression

$y \in R$

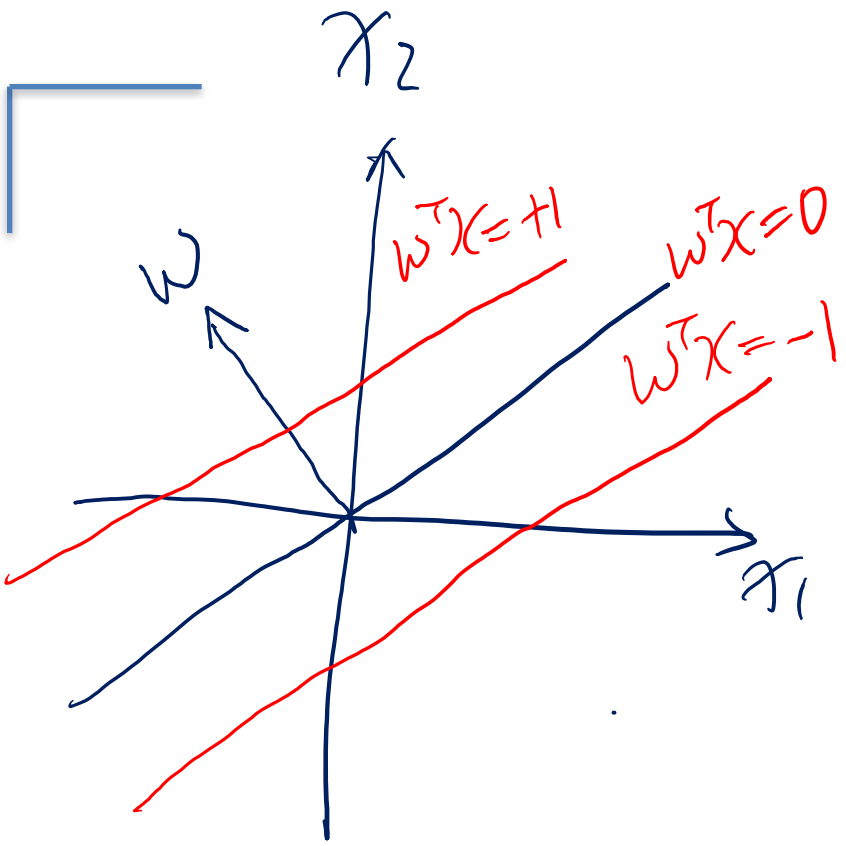
if $\Rightarrow$ 1D $x \in R$



$y = wx + b$

$y = w^T x + b$ if $x \in R^P$

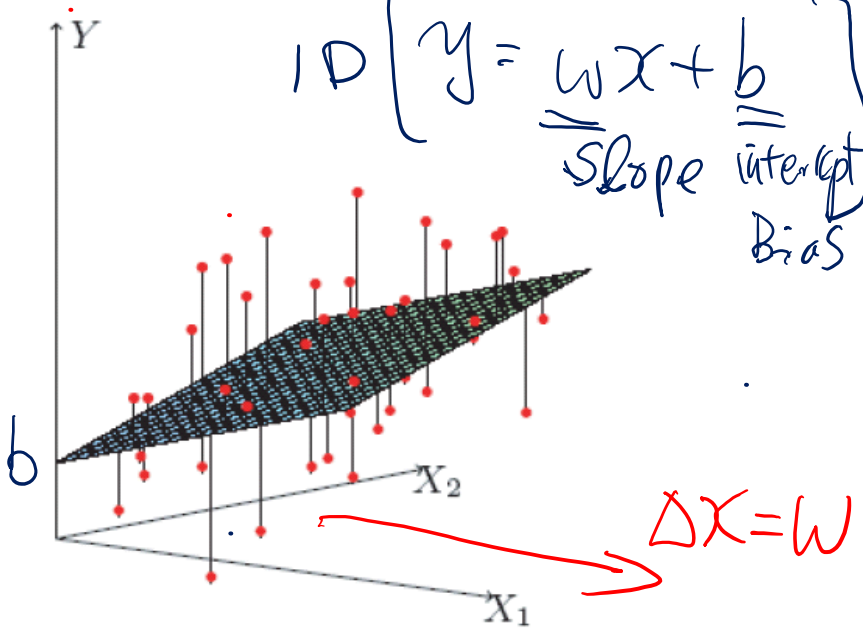
[Classification]



[Regression]

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

$$\Delta X = \frac{\partial y}{\partial x} = w$$

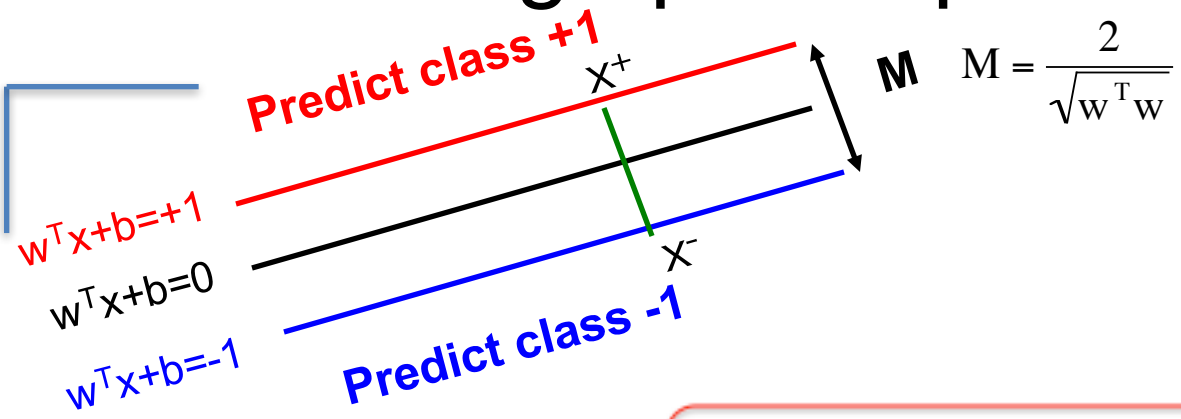


The gradient points in the direction of the greatest rate of increase of the function and its magnitude is the slope of the graph in that direction

Today

- Support Vector Machine (SVM)
  - ✓ History of SVM
  - ✓ Large Margin Linear Classifier
  - ✓ Define Margin (M) in terms of model parameter
  - ✓ Optimization to learn model parameters (w, b)
  - ✓ Linearly Non-separable case
  - ✓ Optimization with dual form
  - ✓ Nonlinear decision boundary
  - ✓ Practical Guide

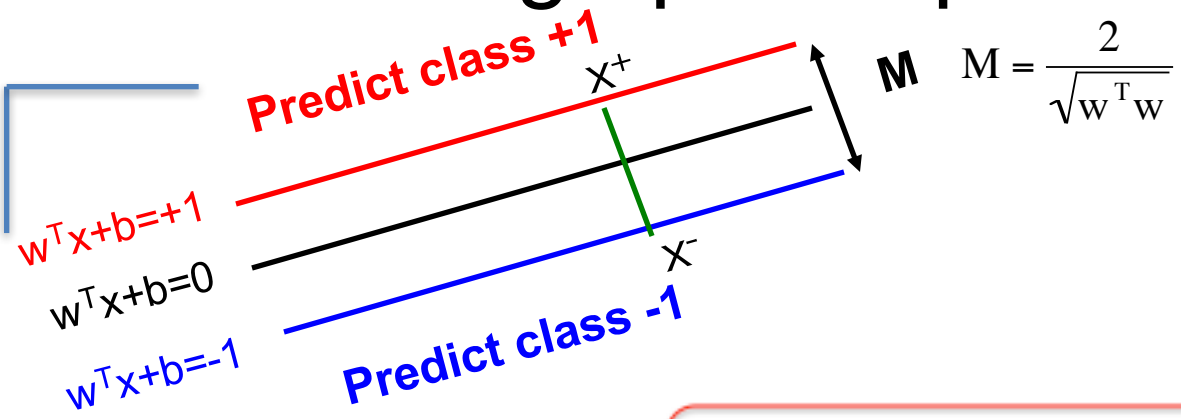
Optimization Step
i.e. learning optimal parameter for SVM



$$M = \frac{2}{\sqrt{w^T w}}$$

1. Correctly classifies all points
2. Maximizes the margin (or equivalently minimizes $w^T w$)

Optimization Step
i.e. learning optimal parameter for SVM

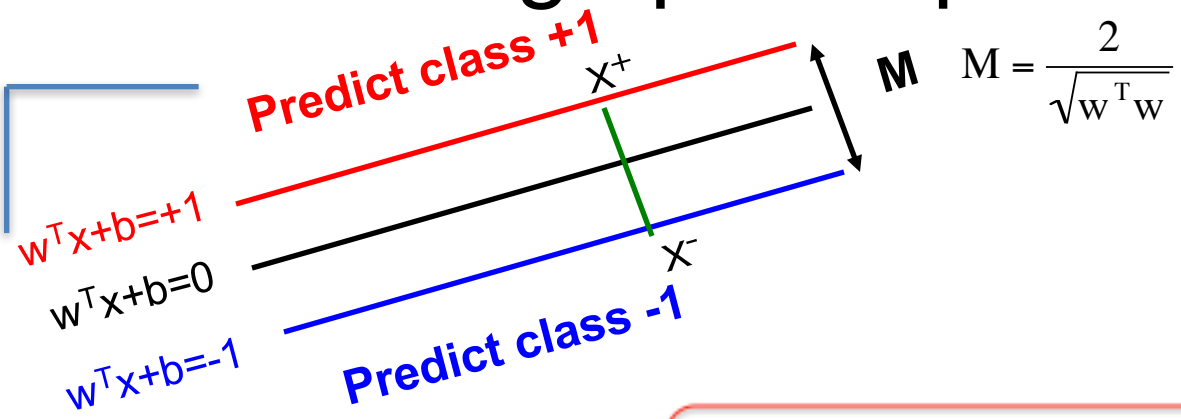


1. Correctly classifies all points
2. Maximizes the margin (or equivalently minimizes $w^T w$)

Min $(w^T w)/2$

subject to the following constraints:

Optimization Step
i.e. learning optimal parameter for SVM



1. Correctly classifies all points
2. Maximizes the margin (or equivalently minimizes $w^T w$)

Min $(w^T w)/2$

subject to the following constraints:

For all x in class + 1

$w^T x+b >= 1$

For all x in class - 1

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

A total of n constraints if we have n training samples

Optimization Reformulation

1. Correctly classifies all points
2. Maximizes the margin (or equivalently minimizes $w^T w$)

Min $(w^T w)/2$

subject to the following constraints:

For all x in class + 1

$w^T x+b >= 1$ $y_i = 1$

For all x in class - 1

$w^T x+b <= -1$ $y_i = -1$

A total of n constraints if we have n input samples

$\rightarrow$ pos $y_i = 1$, $w^T x_i + b \geq 1$

$y_i (w^T x_i + b) \geq 1$

$\rightarrow$ neg $y_i = -1$, $w^T x_i + b \leq -1$

$y_i (w^T x_i + b) \geq 1$

Optimization Reformulation

1. Correctly classifies all points
2. Maximizes the margin (or equivalently minimizes $w^T w$)

Min $(w^T w)/2$

subject to the following constraints:

For all x in class + 1

$w^T x+b >= 1$

For all x in class - 1

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

A total of n constraints if we have n input samples

$$\underset{\mathbf{w},b}{\operatorname{argmin}} \sum_{i=1}^{p} w_i^2$$

$$\text{subject to } \forall \mathbf{x}_i \in Dtrain : y_i\left(\mathbf{x}_i \cdot \mathbf{w} + b\right) \geq 1$$

Optimization Reformulation

1. Correctly classifies all points
2. Maximizes the margin (or equivalently minimizes $w^T w$)

Min $(w^T w)/2$

subject to the following constraints:

For all x in class + 1

$w^T x+b >= 1$

For all x in class - 1

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

A total of n constraints if we have n input samples

Quadratic Objective

$$\underset{\mathbf{w},b}{\operatorname{argmin}} \sum_{i=1}^{p} w_i^2$$

$$\text{subject to } \forall \mathbf{x}_i \in Dtrain : y_i\left(\mathbf{w}^T \mathbf{x}_i + b\right) \geq 1$$

Quadratic programming i.e.,

- Quadratic objective
- Linear constraints

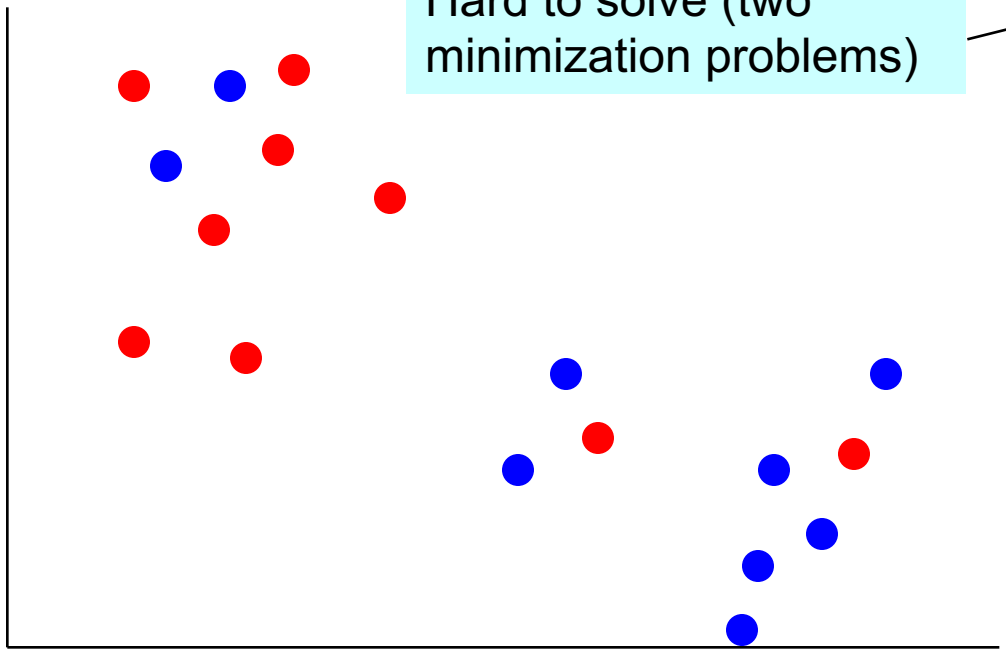
Today

- Support Vector Machine (SVM)
  - ✓ History of SVM
  - ✓ Large Margin Linear Classifier
  - ✓ Define Margin (M) in terms of model parameter
  - ✓ Optimization to learn model parameters (w, b)
  - ✓ Linearly Non-separable case (soft SVM)
  - ✓ Optimization with dual form
  - ✓ Nonlinear decision boundary
  - ✓ Practical Guide

Linearly Non separable case

- So far we assumed that a linear hyperplane can perfectly separate the points
- But this is not usually the case

  - noise, outliers



Hard to solve (two minimization problems)

How can we convert this to a QP problem?

- Minimize training errors?

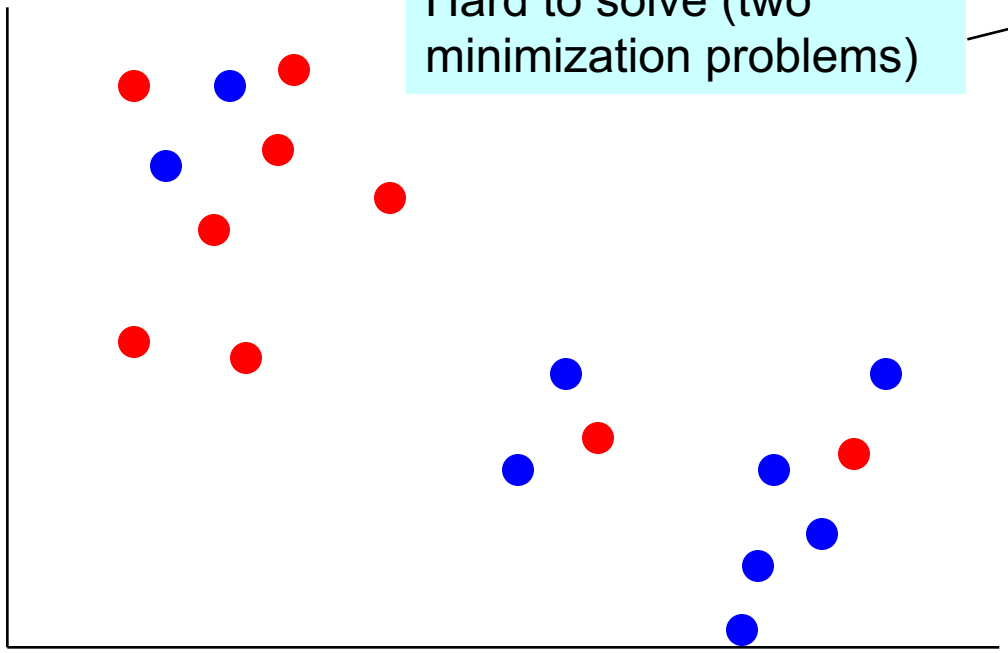
  min $w^T w$

  min #errors

Linearly Non separable case

- So far we assumed that a linear plane can perfectly separate the points
- But this is not usally the case

  - noise, outliers



Hard to solve (two minimization problems)

How can we convert this to a QP problem?

- Minimize training errors?

  min $w^T w$

  min #errors

- Penalize training errors:

  min $w^T w + C*(\#errors)$

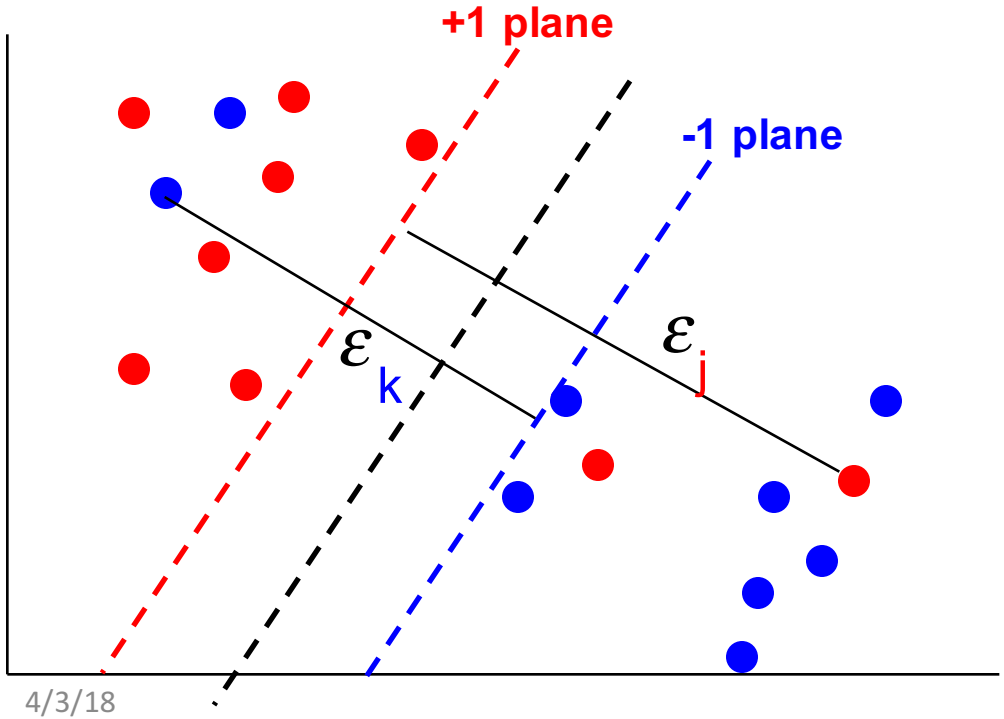
Hard to encode in a QP problem

Linearly Non separable case

- Instead of minimizing the number of misclassified points we can minimize the ***distance*** between these points and their correct plane

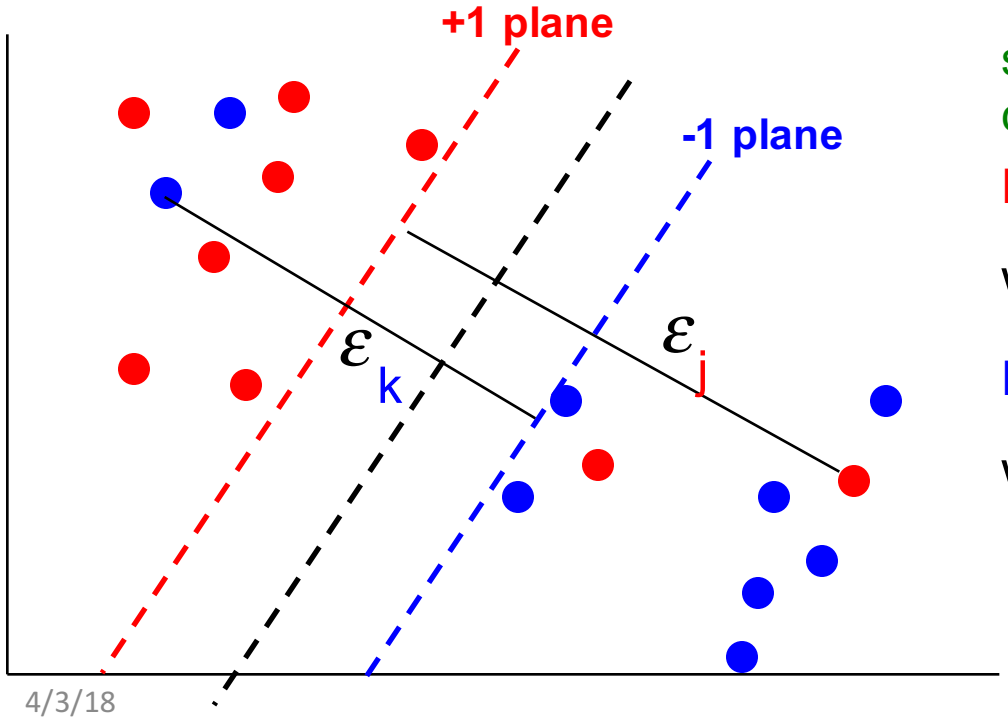
The new optimization problem is:

$$\min_w \frac{w^T w}{2} + \sum_{i=1}^{n} C\varepsilon_i$$



Linearly Non separable case

- Instead of minimizing the number of misclassified points we can minimize the ***distance*** between these points and their correct plane



The new optimization problem is:

$$\min_w \frac{w^T w}{2} + C\sum_{i=1}^{n} \varepsilon_i$$

subject to the following inequality constraints:

For all x_i in class + 1

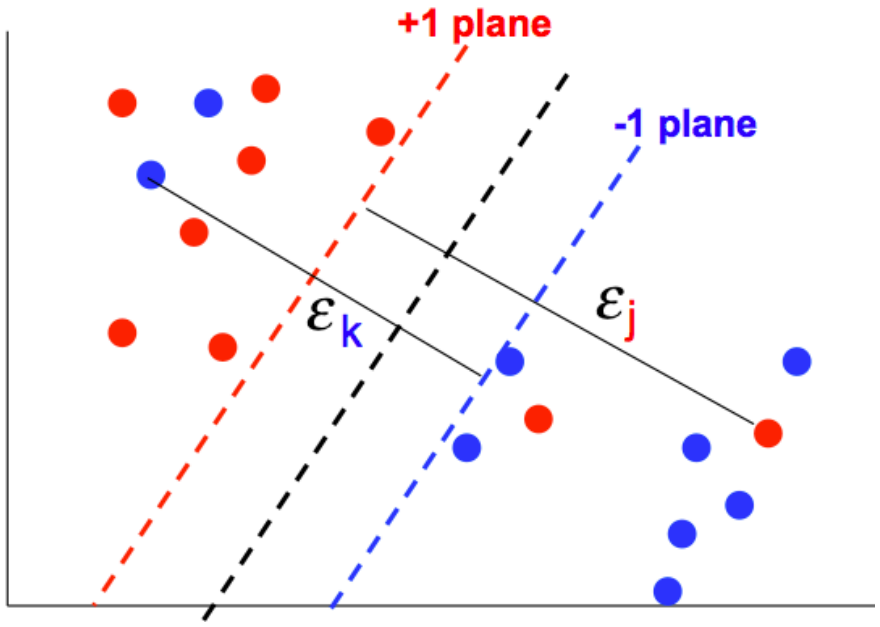
$$w^T x_i + b >= 1 - \varepsilon_i$$

For all x_i in class - 1

$$w^T x_i + b <= -1 + \varepsilon_i$$

Wait. Are we missing something?

Final optimization for linearly non-separable case



The new optimization problem is:

$$\min_w \frac{w^T w}{2} + C\sum_{i=1}^{n} \varepsilon_i$$

subject to the following inequality constraints:

For all x_i in class + 1

$w^T x_i + b >= 1 - \varepsilon_i$

For all x_i in class - 1

$w^T x_i + b <= -1 + \varepsilon_i$

} A total of n constraints

For all i

$$\varepsilon_i \geq 0$$

} Another n constraints

Two optimization problems:

For the separable and non separable cases

$$\min_w \frac{w^T w}{2}$$

For all x in class + 1

$w^T x+b >= 1$

For all x in class - 1

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

$$\min_w \frac{w^T w}{2}+C\sum_{i=1}^{n}\varepsilon_i$$

For all x_i in class + 1

$w^T x_i+b >= 1-\varepsilon_i$

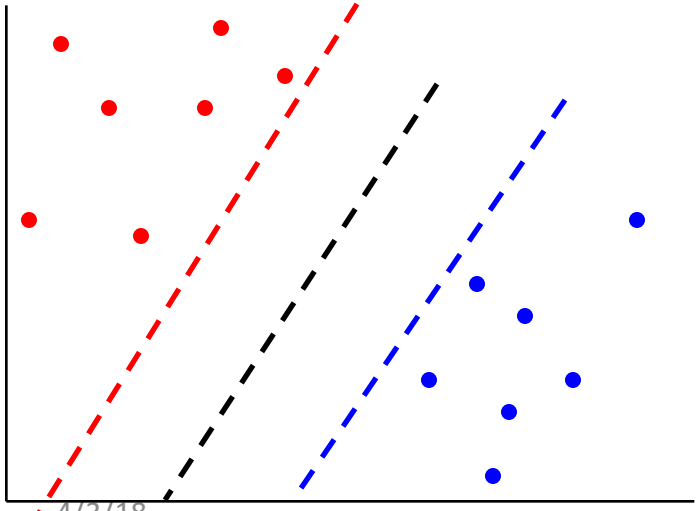
For all x_i in class - 1

$w^T x_i+b <= -1+\varepsilon_i$

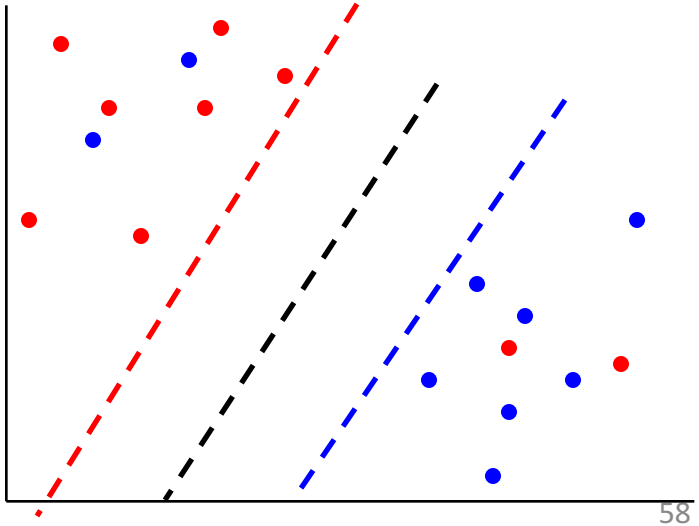
For all i

$$\varepsilon_i \geq 0$$

separable



non separable



Hinge Loss for Soft SVM

$$\min_w \frac{w^T w}{2} + C\sum_{i=1}^{n} \varepsilon_i$$

For all x_i in class + 1

$w^T x_i + b >= 1 - \varepsilon_i$

For all x_i in class - 1

$w^T x_i + b <= -1 + \varepsilon_i$

For all i

$$\varepsilon_i \geq 0$$

soft

$$\underset{\mathbf{w},b}{\text{argmin}} \frac{w^T w}{2} + C\sum_{i=1}^{n} \max(0, 1 - y_i(\mathbf{w}^T \mathbf{x}_i + b))$$

subject to:

$$y_i(\mathbf{w}^T \mathbf{x}_i + b) \geq 1 - \varepsilon_i$$

$$\varepsilon_i \geq 0$$

vs. Hard SVM

$$\underset{\mathbf{w},b}{\text{argmin}} \sum_{i=1}^{p} w_i^2$$

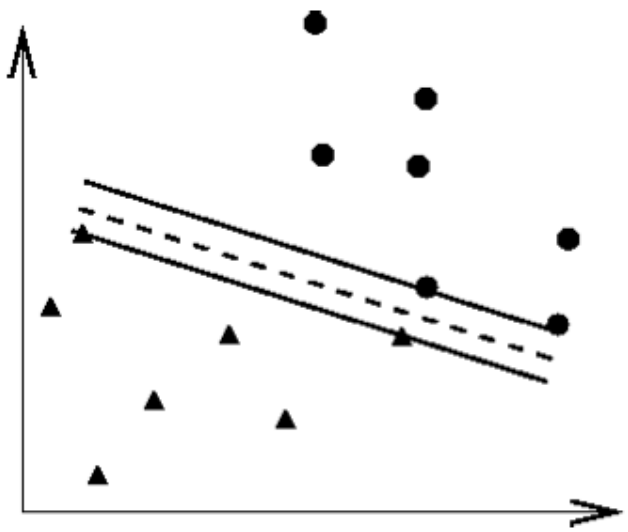
subject to $\forall \mathbf{x}_i \in Dtrain : y_i(\mathbf{w}^T \mathbf{x}_i + b) \geq 1$

Model Selection, find right C

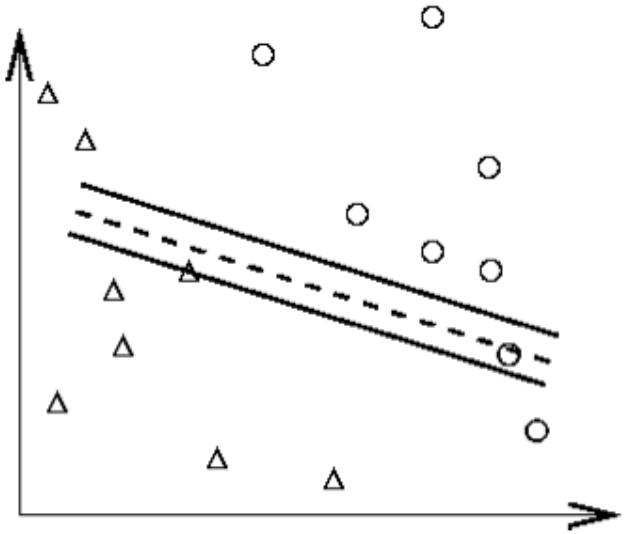
large C

Select the right penalty parameter C

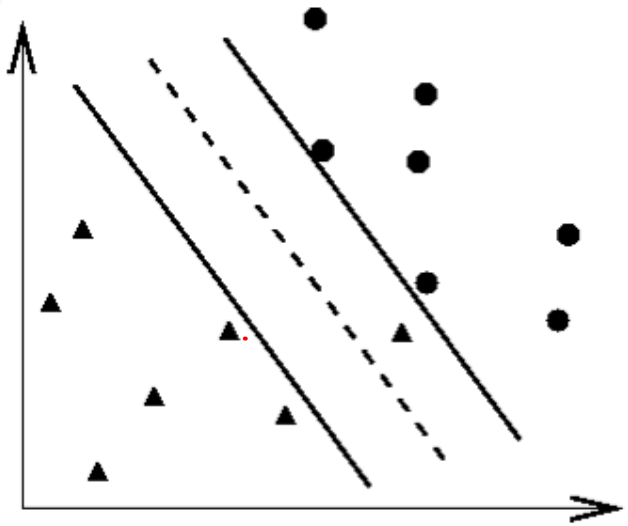
small C



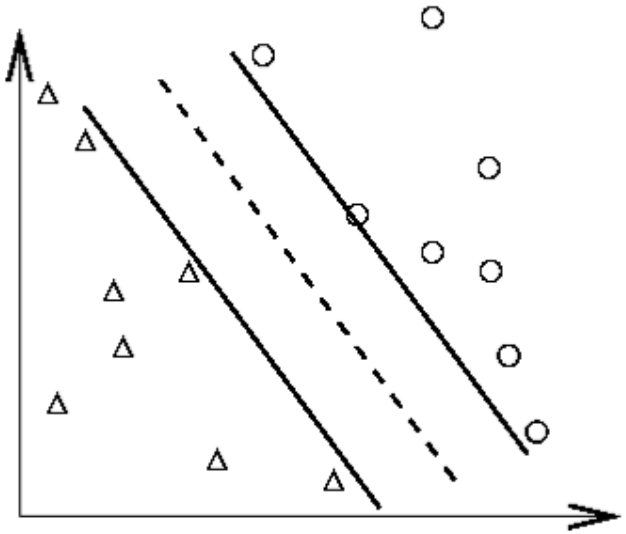
(a) Training data and an overfitting classifier



(b) Applying an overfitting classifier on testing data



(c) Training data and a better classifier



(d) Applying a better classifier on testing data

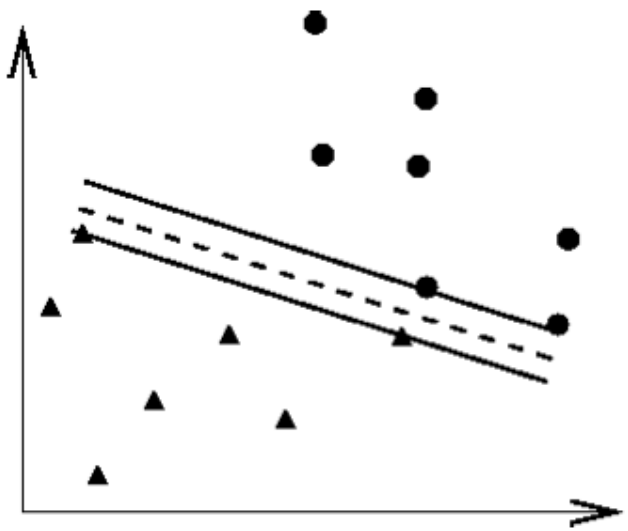
Model Selection, find right C

large C

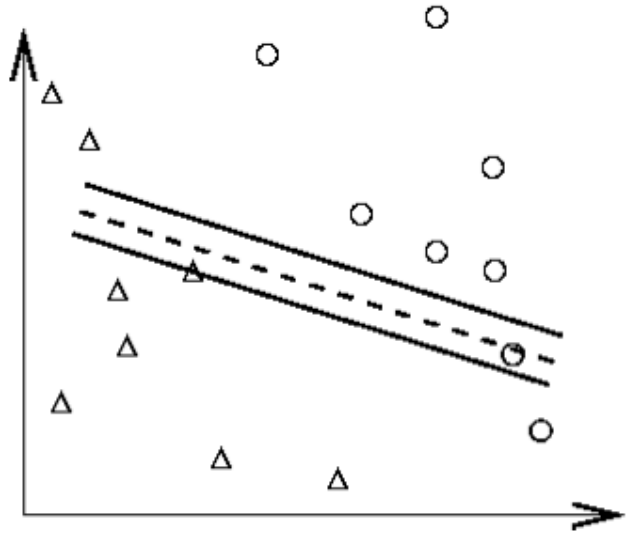
A large value of C means that misclassifications are bad - resulting in smaller margins and less training error (but more expected true error).

A small C results in more training error, hopefully better true error.

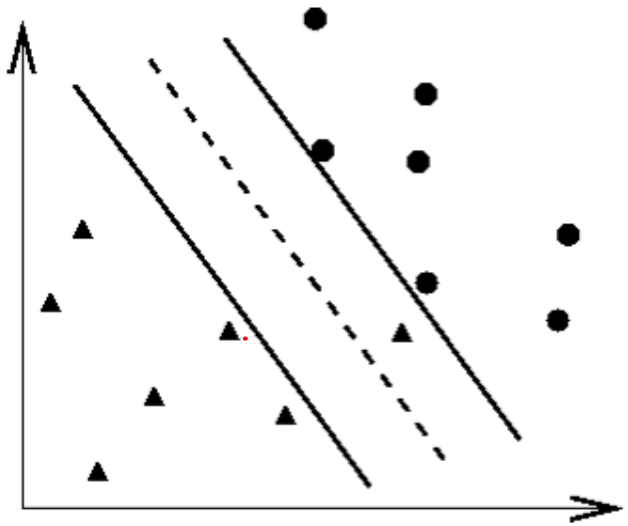
small C



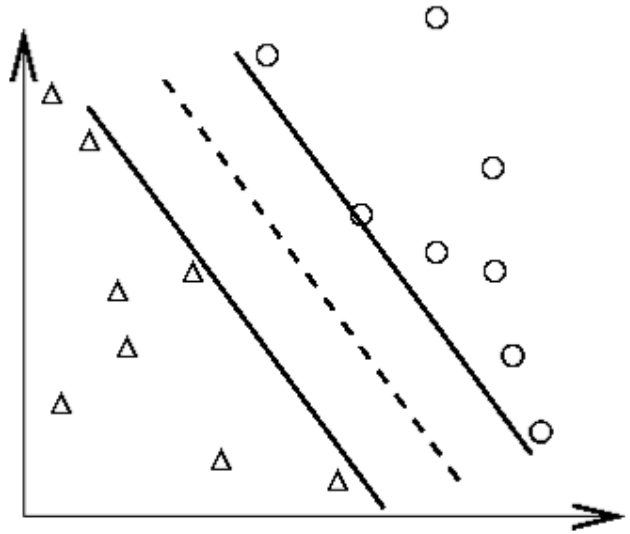
(a) Training data and an overfitting classifier



(b) Applying an overfitting classifier on testing data



(c) Training data and a better classifier



(d) Applying a better classifier on testing data

Today

- Support Vector Machine (SVM)
  - ✓ History of SVM
  - ✓ Large Margin Linear Classifier
  - ✓ Define Margin (M) in terms of model parameter
  - ✓ Optimization to learn model parameters (w, b)
  - ✓ Linearly non-separable case
  - ✓ Optimization with dual form
  - ✓ Nonlinear decision boundary
  - ✓ Practical Guide

Two optimization problems: For the separable and non separable cases

Min $(w^T w)/2$

For all x in class + 1

$w^T x+b >= 1$

For all x in class - 1

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

$$\min_w \frac{w^T w}{2}+C\sum_{i=1}^{n}\varepsilon_i$$

For all x_i in class + 1

$w^T x_i+b >= 1-\varepsilon_i$

For all x_i in class - 1

$w^T x_i+b <= -1+\varepsilon_i$

For all i

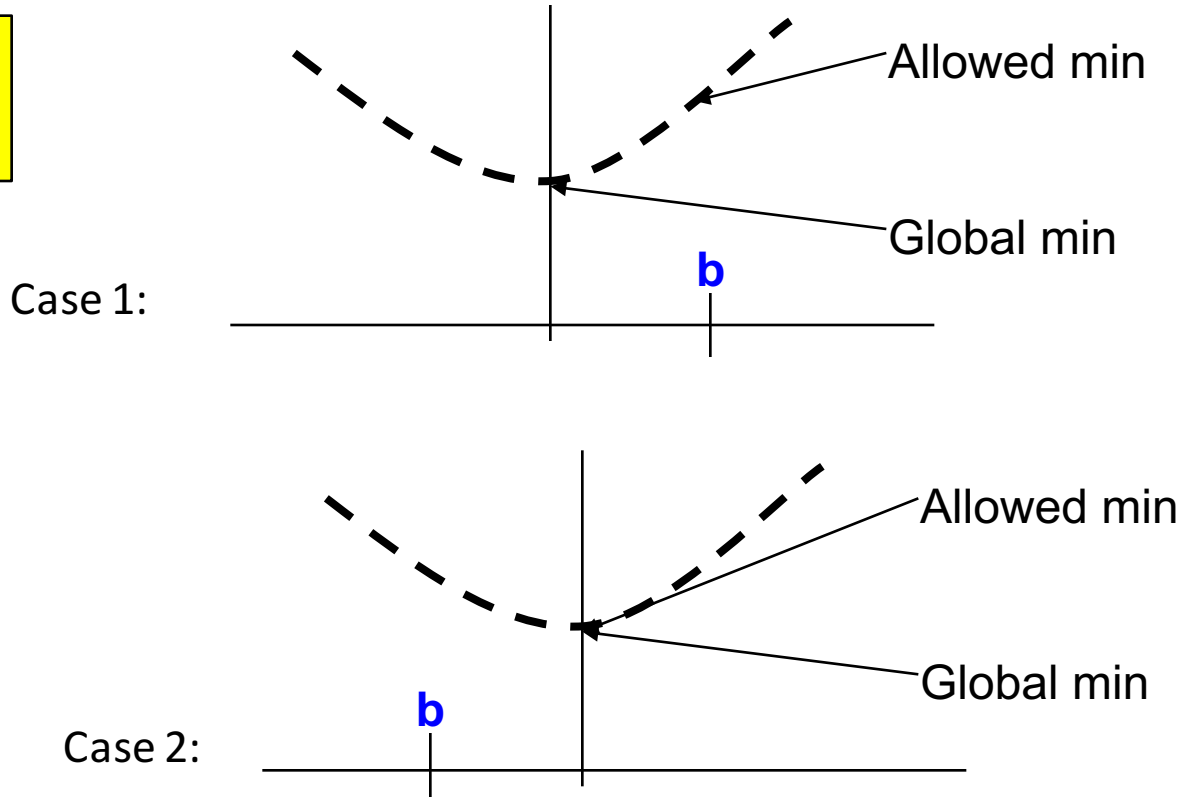
$$\varepsilon_i \geq 0$$

- Instead of solving these QPs directly we will solve a dual formulation of the SVM optimization problem
- The main reason for switching to this type of representation is that it would allow us to use a neat trick that will make our lives easier (and the run time faster)

Optimization Review: Constrained Optimization

$\min_u u^2$

s.t. $u >= b$

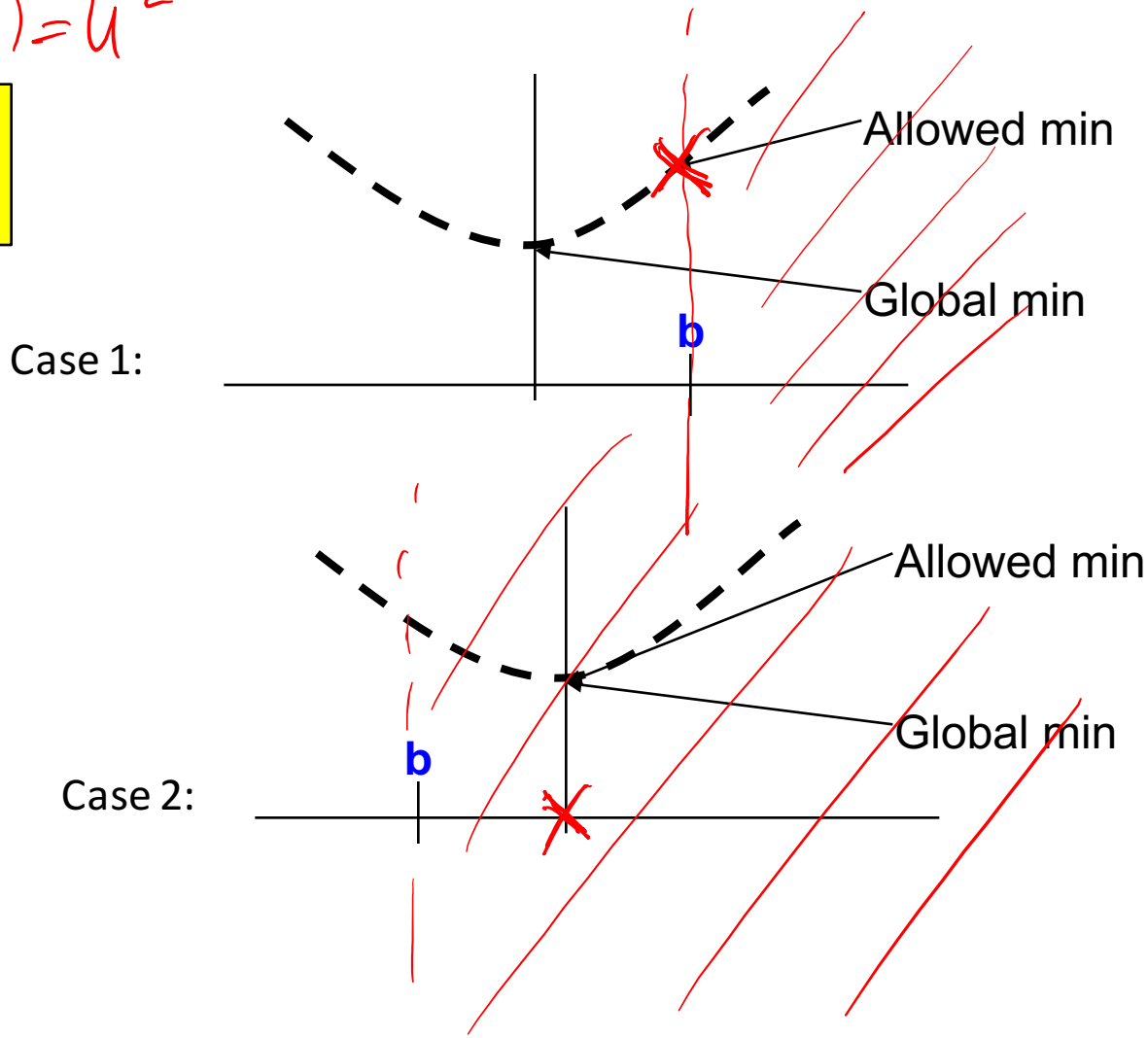


Optimization Review: Constrained Optimization

$f(u) = u^2$

$\min_u u^2$

s.t. $u >= b$



Optimization Review: Constrained Optimization

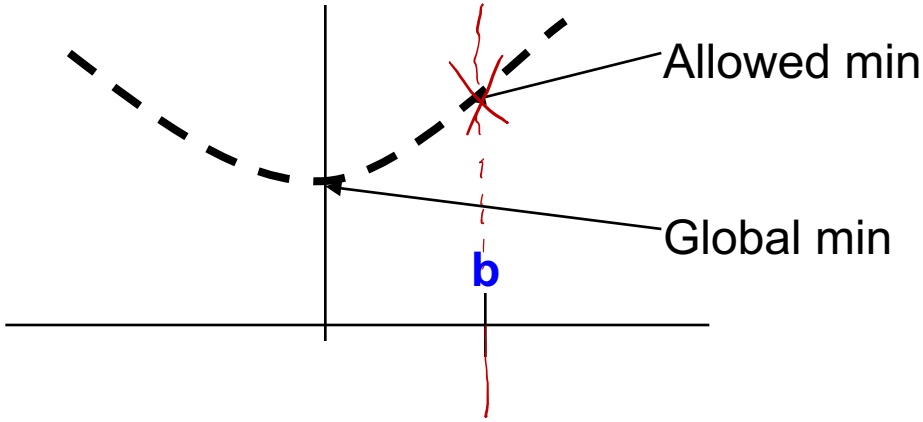
$f(u)$

$\min_u u^2$

s.t. $u >= b$

[Subject to]

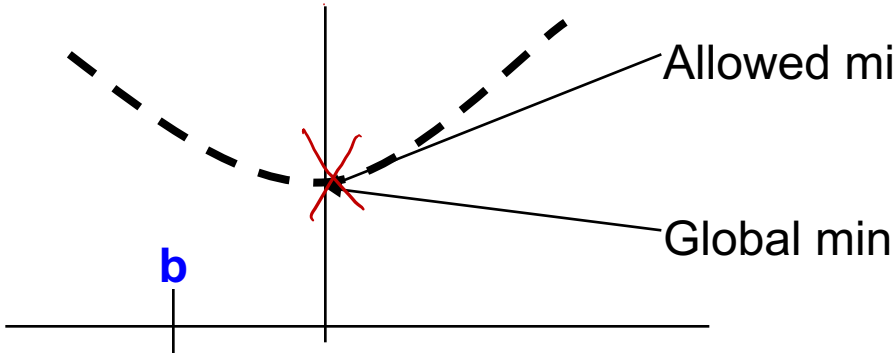
Case 1:



$b > 0$

$f(u) = b^2$

Case 2:



$b < 0$

$f(u) = 0$

Optimization Review: Ingredients

- Objective function
- Variables
- Constraints

Find values of the variables
that minimize or maximize the objective function
while satisfying the constraints

Optimization Review: Lagrangian Duality (Extra)

- The Primal Problem

Primal:

$$\min_w \quad f_0(w)$$
$$\text{s.t.} \quad f_i(w) \le 0, \quad i = 1,\ldots,k$$

The generalized Lagrangian:

"Method of Lagrange multipliers" convert to a higher-dimensional problem

$$L(w,\alpha) = f_0(w) + \sum_{i=1}^{k} \alpha_i f_i(w)$$

the α's ($\alpha_i \ge 0$) are called the Lagarangian multipliers

Lemma:

$$\max_{\alpha,\alpha_i \ge 0} L(w,\alpha) = \begin{cases} f_0(w) & \text{if } w \text{ satisfies primal constraints} \\ \infty & \text{o/w} \end{cases}$$

A re-written Primal:

$$\min_w \max_{\alpha,\alpha_i \ge 0} L(w,\alpha)$$

Optimization Review: Lagrangian Duality, cont. (Extra)

- Recall the Primal Problem:

$$\min_w \max_{\alpha, \alpha_i \geq 0} L(w, \alpha)$$

- The Dual Problem:

$$\max_{\alpha, \alpha_i \geq 0} \min_w L(w, \alpha)$$

- **Theorem (weak duality):**

$$d^* = \max_{\alpha, \alpha_i \geq 0} \min_w L(w, \alpha) \quad \leq \quad \min_w \max_{\alpha, \alpha_i \geq 0} L(w, \alpha) = p^*$$

- **Theorem (strong duality):**

Iff there exist a saddle point of $L(w, \alpha)$

we have $d^* = p^*$

An alternative representation of the SVM QP

- We will start with the linearly separable case
- Instead of encoding the correct classification rule and constraint we will use Lagrange multiplies to encode it as part of the our minimization problem

Min $(w^T w)/2$

s.t.

$(w^T x_i + b) y_i >= 1$

Recall that Lagrange multipliers can be applied to turn the following problem:

$$L_{primal}(w,b,\alpha) = \frac{1}{2}\left\|\mathbf{w}\right\|^2 - \sum_{i=1}^{N}\alpha_i\left(y_i(\mathbf{w}\cdot\mathbf{x}_i + b) - 1\right)$$

$\mathbf{w}^T\mathbf{w}$

The Dual Problem (Extra)

$$\max_{\alpha_i \geq 0} \min_{w,b} L(w,b,\alpha)$$

Dual formulation

- We minimize L with respect to w and b first:

$$\nabla_w L(w,b,\alpha) = w - \sum_{i=1}^{train} \alpha_i y_i x_i = 0, \quad (*)$$

$$\nabla_b L(w,b,\alpha) = \sum_{i=1}^{train} \alpha_i y_i = 0, \quad (**)$$

Note that **(*)** implies: $w = \sum_{i=1}^{train} \alpha_i y_i x_i$ (***)

- Plus (***) back to L , and using (**), we have:

$$L(w,b,\alpha) = \sum_{i=1} \alpha_i - \frac{1}{2} \sum_{i,j=1} \alpha_i \alpha_j y_i y_j (\mathbf{x}_i^T \mathbf{x}_j)$$

Summary: Dual for SVM (Extra)

Solving for **w** that gives maximum margin:

1. Combine objective function and constraints into new objective function, using Lagrange multipliers \alpha_i

$$L_{primal} = \frac{1}{2}\left\|\mathbf{w}\right\|^2 - \sum_{i=1}^{N} \alpha_i \left(y_i (\mathbf{w} \cdot \mathbf{x}_i + b) - 1 \right)$$

2. To minimize this Lagrangian, we take derivatives of **w** and *b* and set them to 0:

Summary: Dual for SVM (Extra)

3. Substituting and rearranging gives the dual of the Lagrangian:

$$L_{dual} = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_i \sum_j \alpha_i \alpha_j y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j$$

which we try to maximize (not minimize).

4. Once we have the *alpha*$_i$, we can substitute into previous equations to get **w** and *b*.
5. This defines **w** and *b* as linear combinations of the training data.

$$w = \sum_{i=1}^{train} \alpha_i y_i x_i$$

Summary: Dual SVM for linearly separable case

Dual formulation

$$\max_{\alpha} \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j$$

$$\sum_i \alpha_i y_i = 0$$

$$\alpha_i \geq 0 \qquad \forall i$$

Min $(w^T w)/2$

subject to the following inequality constraints:

For all x in class + 1

$w^T x+b >= 1$

For all x in class - 1

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

A total of n constraints if we have n input samples

Easier than original QP, more efficient algorithms exist to find α_i

Dual SVM for linearly separable case – Training

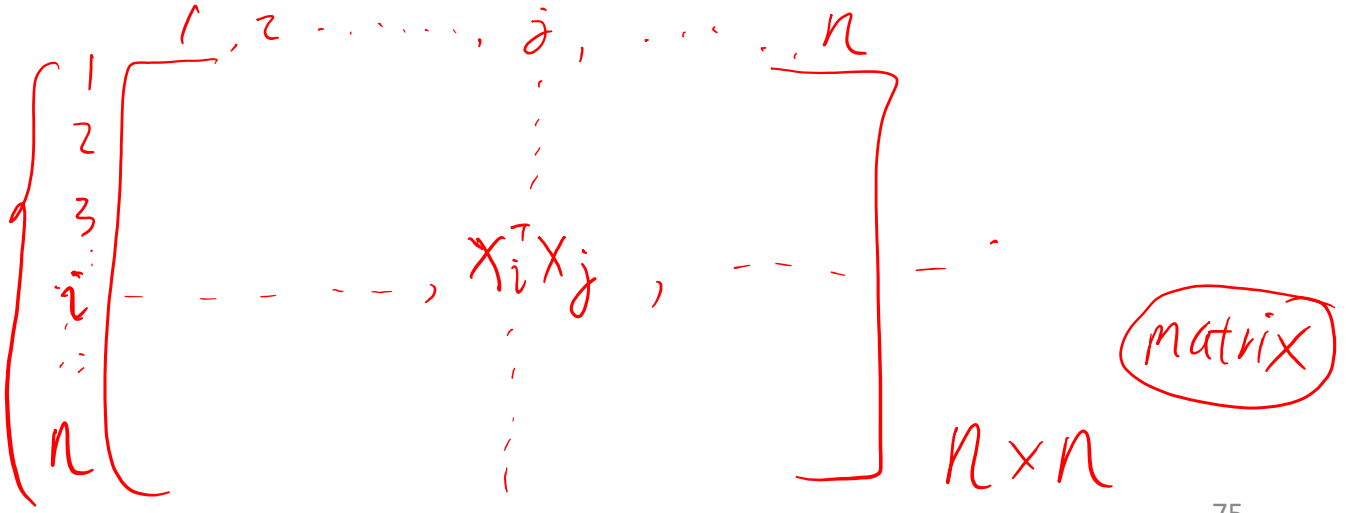
Our dual target function:

$$\max_\alpha \sum_i \alpha_i - \frac{1}{2}\sum_{i,j} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j$$

$$\sum_i \alpha_i y_i = 0$$

$$\alpha_i \geq 0 \qquad \forall i$$

Dot product for all training samples



Support vectors: non-zero α_i

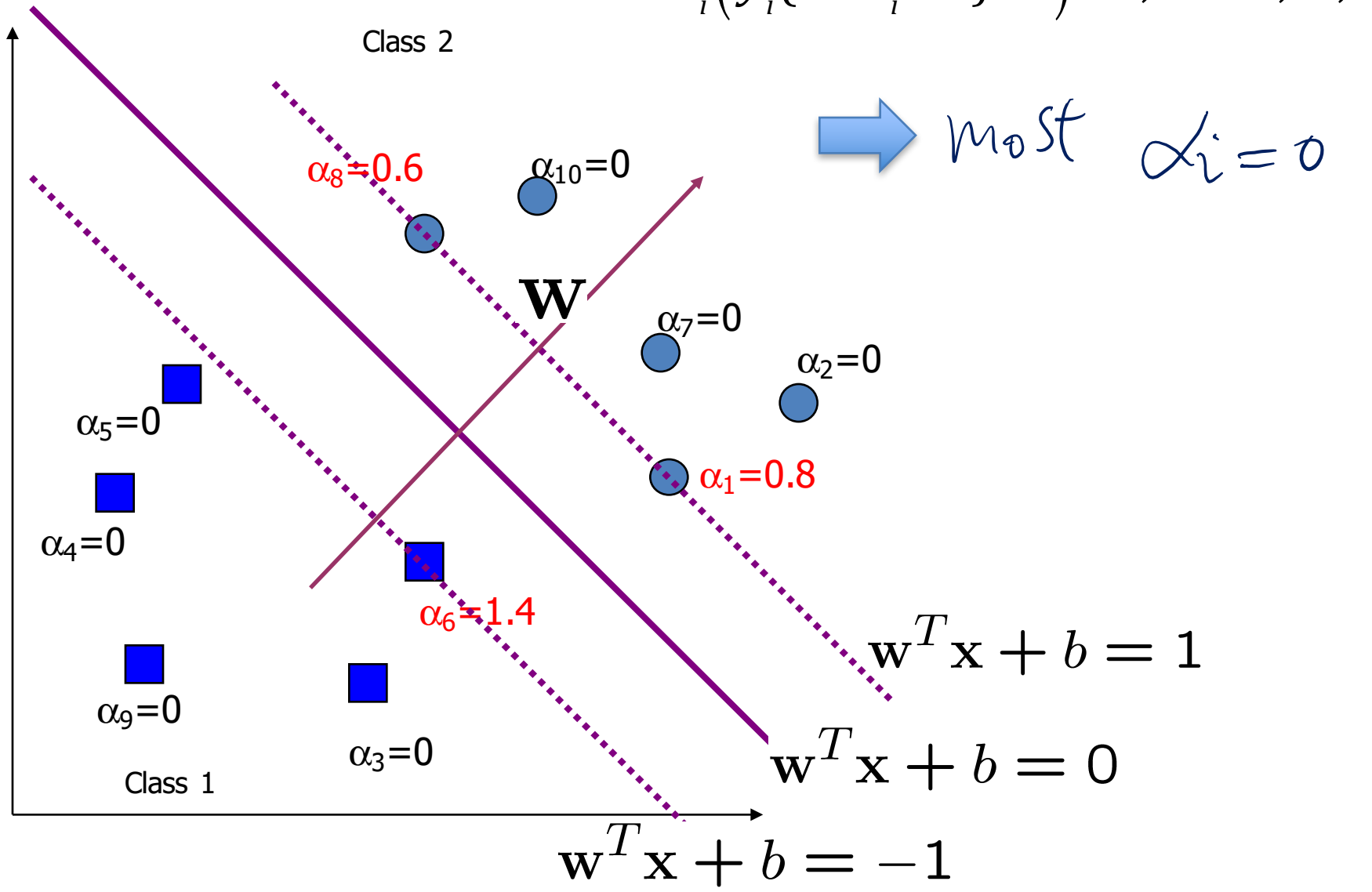
- only a few α_i's can be nonzero!!

$$\alpha_i \left(y_i (\mathbf{w} \cdot \mathbf{x}_i + b) - 1 \right) = 0, \quad i = 1, \ldots, n$$

for most $\alpha_i \Rightarrow \alpha_i = 0$

We call the training data points whose α_i's are nonzero the **support vectors** (SV)

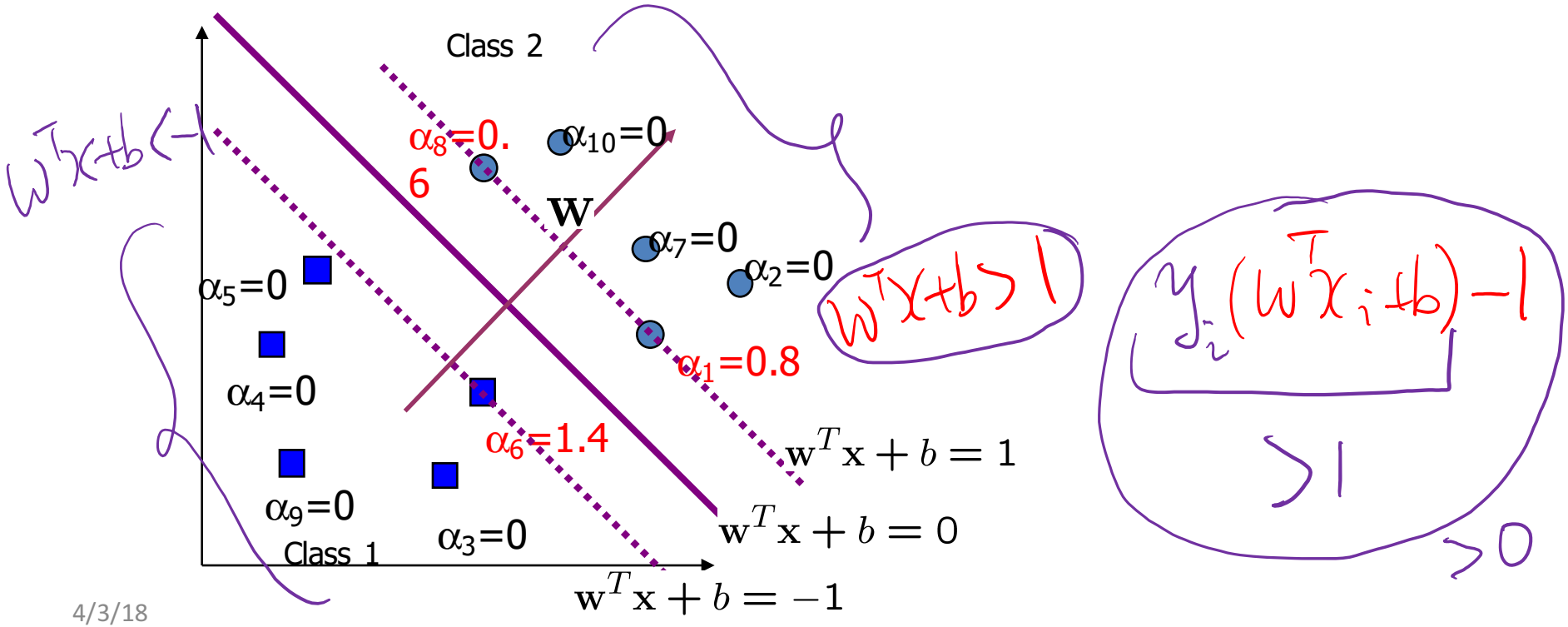
$$\alpha_i \left(y_i (\mathbf{w} \cdot \mathbf{x}_i + b) - 1 \right) = 0, \quad i = 1, \ldots, n$$



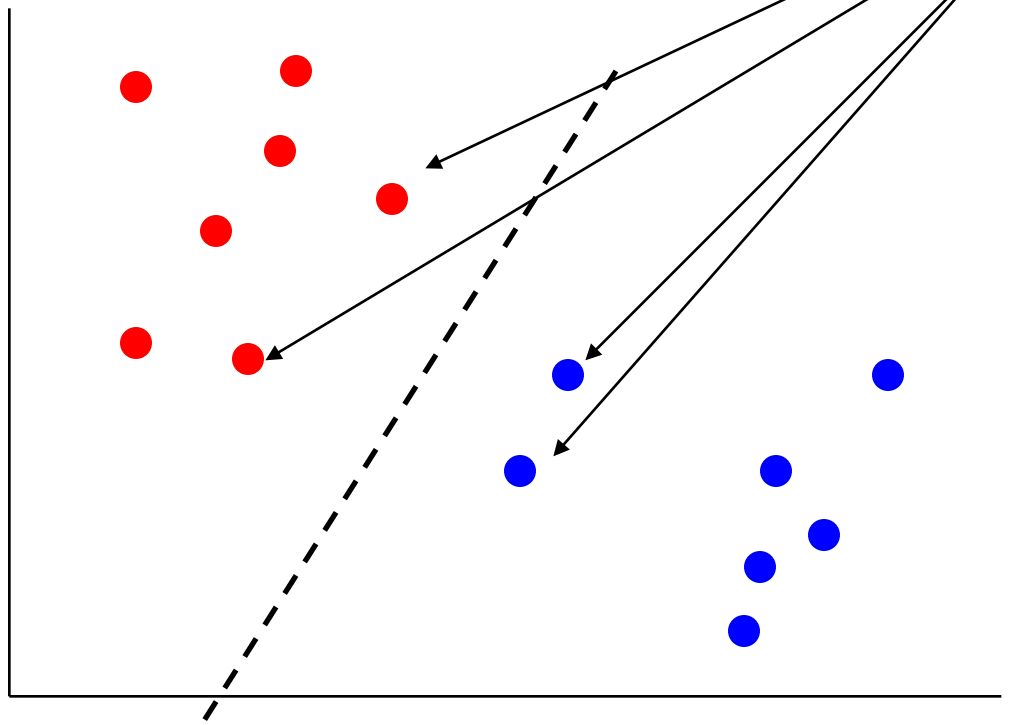
Most $\alpha_i = 0$

- only a few α_i can be nonzero!!

$$\alpha_i \left(y_i (\mathbf{w} \cdot \mathbf{x}_i + b) - 1 \right) = 0, \quad i = 1, \ldots, n$$



Dual SVM - interpretation



training points

$$w = \sum_i \alpha_i x_i y_i$$

$\alpha_i \geq 0$

For α_i that are 0, no influence

Dual SVM for linearly separable case – Testing

Dot product with ("all" ??) training samples

To evaluate a new sample x_{ts} we need to compute:

$$\widehat{y}_{ts} = \text{sign}(\mathrm{w}^{\mathrm{T}} x_{ts} + b) = \text{sign}(\sum_{\mathrm{i}=1..\mathrm{n}} \alpha_i \mathrm{y}_{\mathrm{i}} \mathbf{x}_{\mathbf{i}}^{T} \mathbf{x}_{ts} + b)$$

$$\Rightarrow \widehat{y}_{ts} = \text{sign}\left(\sum_{i \in SupportVectors} \alpha_i y_i \left(\mathbf{x}_i^T \mathbf{x}_{ts} \right) + b \right)$$

For \alpha$_i$ that are 0, no influence

Dual formulation for linearly non-separable case

Dual target function:

$$\max_{\alpha} \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j \mathbf{x_i}^T \mathbf{x_j}$$

$$\sum_i \alpha_i y_i = 0$$

$$C > \alpha_i \geq 0, \forall i$$

Hyperparameter C should be tuned through k-folds CV

The only difference is that the \alpha are now bounded

This is very similar to the optimization problem in the linear separable case, except that there is an upper bound C on α_i now

Once again, efficient algorithm exist to find α_i

Dual formulation for linearly non-separable case

Dual target function:

$$\max_{\alpha} \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j \mathbf{x_i}^T \mathbf{x_j}$$

$$\sum_i \alpha_i y_i = 0$$

$$C > \alpha_i \geq 0, \forall i$$

The only difference is that the \alpha are now bounded

Hyperparameter C should be tuned through k-folds CV

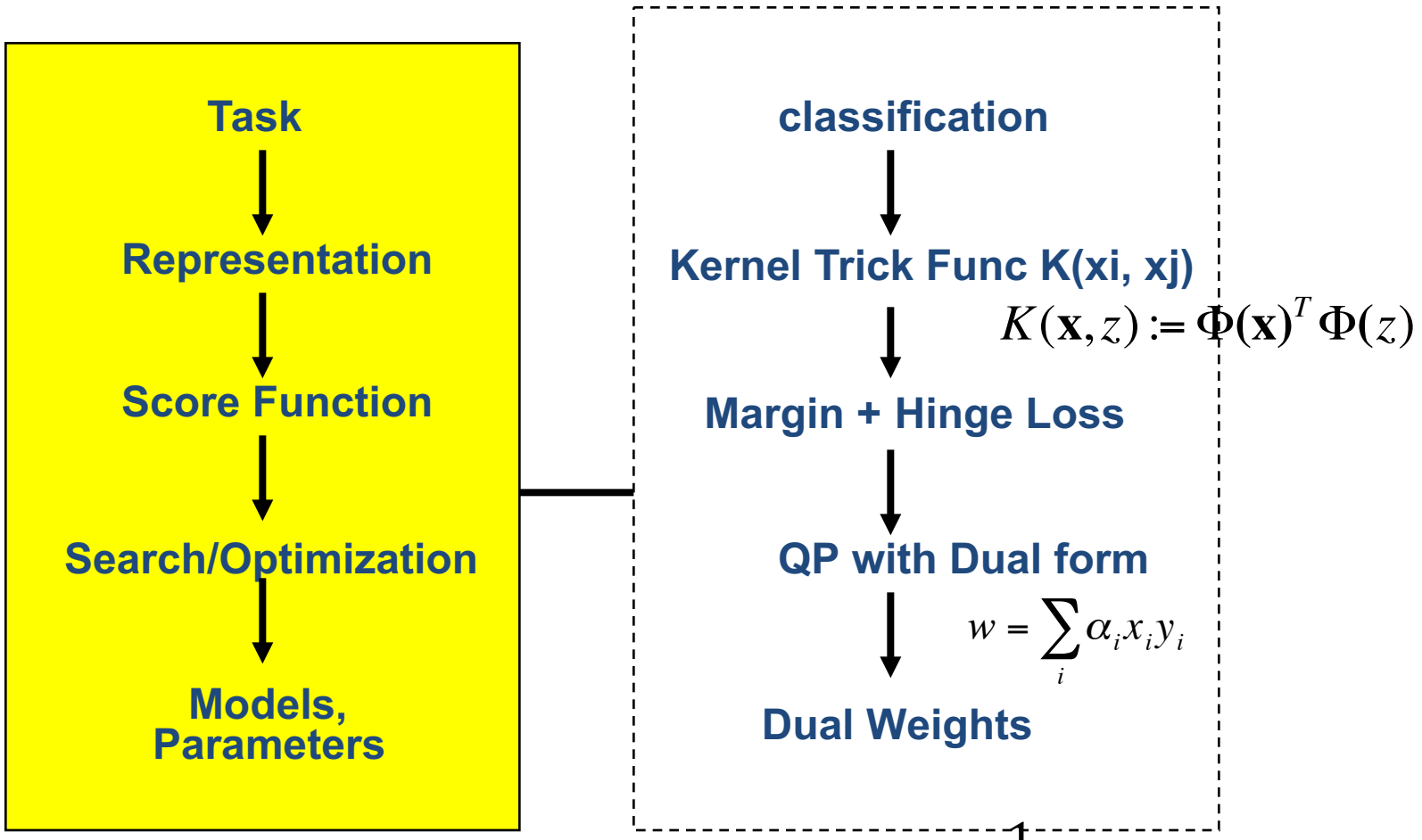
To evaluate a new sample x_{ts} we need to compute:

$$w^T x_{ts} + b = \sum_{i \in \text{supportV}} \alpha_i y_i \mathbf{x_i}^T \mathbf{x}_{ts} + b$$

This is very similar to the optimization problem in the linear separable case, except that there is an upper bound C on α_i now

Once again, efficient algorithm exist to find α_i

Support Vector Machine



$$\underset{\mathbf{w},b}{\operatorname{argmin}} \sum_{i=1}^{p} w_i^2 + \mathrm{C} \sum_{\mathrm{i}=1}^{\mathrm{n}} \varepsilon_{\mathrm{i}}$$

subject to $\forall \mathbf{x}_i \in Dtrain : y_i \left(\mathbf{x}_i \cdot \mathbf{w} + b \right) \geq 1 - \varepsilon_{\mathrm{i}}$

$$\max_{\alpha} \sum_i \alpha_i - \frac{1}{2} \sum_{\mathrm{i,j}} \alpha_i \alpha_j \mathrm{y}_{\mathrm{i}} y_j \mathbf{x_i}^T \mathbf{x_j}$$

$$\sum_i \alpha_i \mathrm{y}_{\mathrm{i}} = 0, \qquad \alpha_i \geq 0 \qquad \forall i$$

References

- Big thanks to Prof. Ziv Bar-Joseph and Prof. Eric Xing @ CMU for allowing me to reuse some of his slides
- Elements of Statistical Learning, by Hastie, Tibshirani and Friedman
- Prof. Andrew Moore @ CMU's slides
- Tutorial slides from Dr. Tie-Yan Liu, MSR Asia
- A Practical Guide to Support Vector Classification Chih-Wei Hsu, Chih-Chung Chang, and Chih-Jen Lin, 2003-2010
- Tutorial slides from Stanford "Convex Optimization I — Boyd & Vandenberghe