Using Zoom for Lectures

Sign in using:

your name

Please mute both:

your video cameras for the entire lecture

your audio/mics unless asking or answering a question

Asking/answering a question, option 1:

click on Participants

use the hand icon to raise your hand

I will call on you and ask you to unmute yourself

Asking/answering a question, option 2:

click on Chat

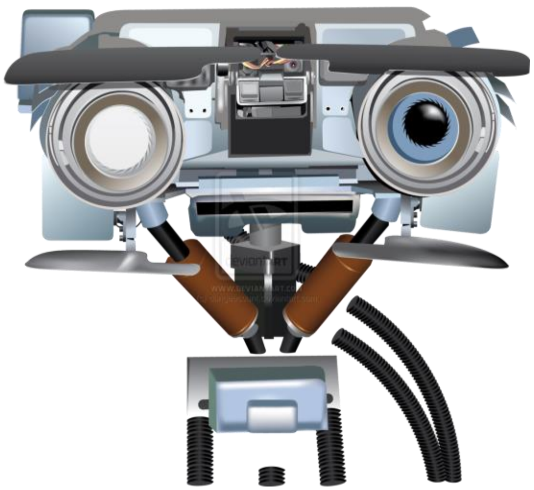
type your question, and I will answer it

Today: Outline

Maximum-Margin

Support Vector Machines

Reminder: PS3 Self Score due Mar 23



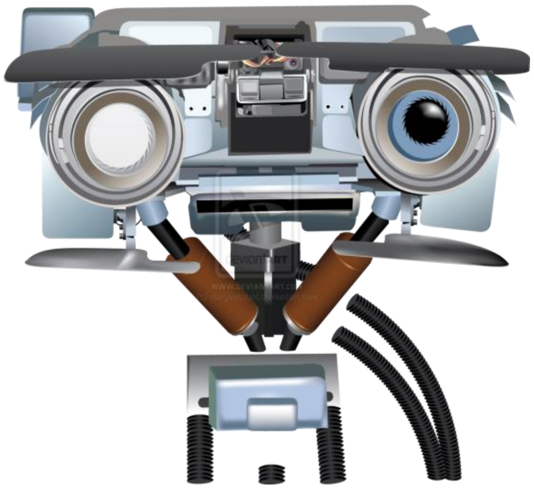
Support Vector Machines

CS542 Machine Learning

slides based on lecture by R. Urtasun
http://www.cs.toronto.edu/~urtasun/courses/CSC2515/CSC2515_Winter15.html

Support Vector Machine (SVM)

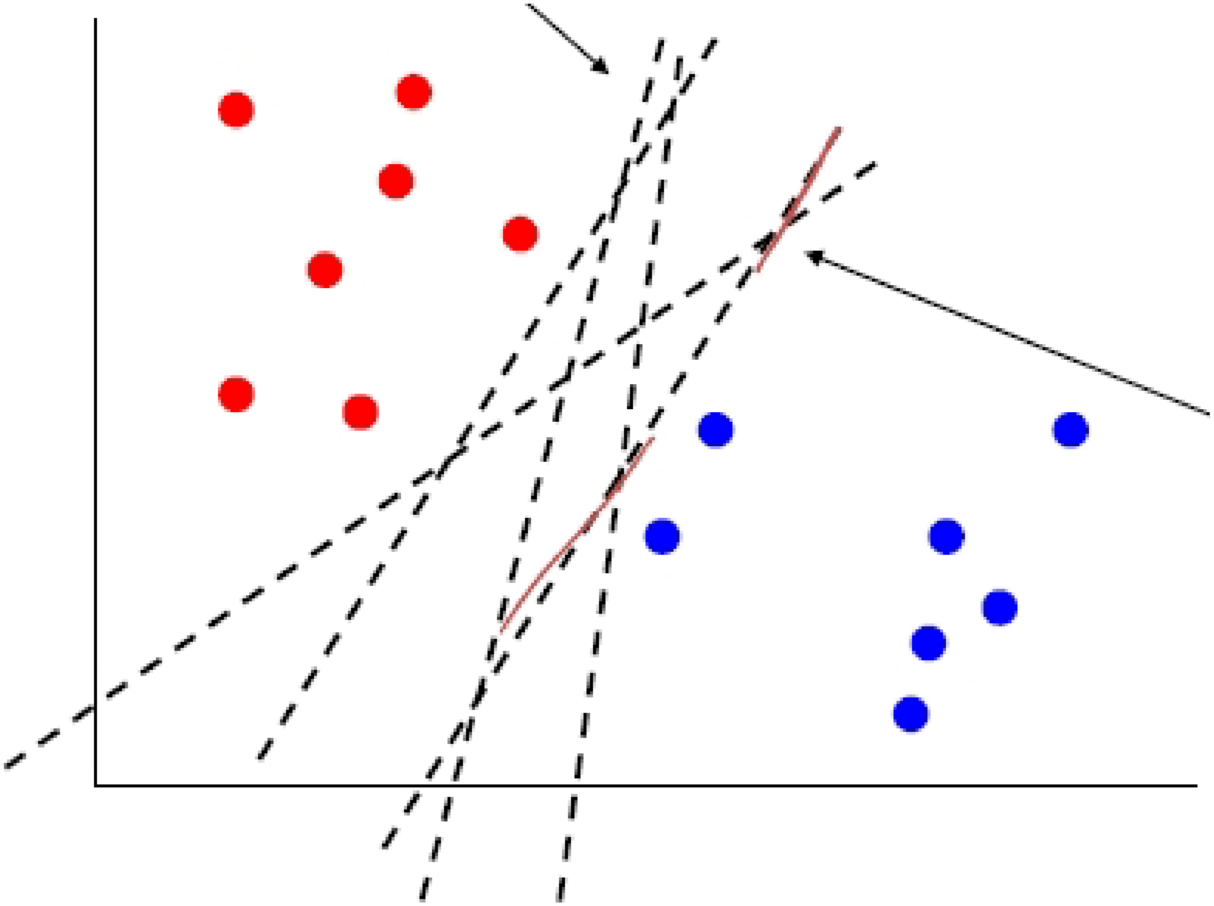
- A *maximum margin* method, can be used for classification or regression
- SVMs can efficiently perform a non-linear classification using what is called the kernel trick, implicitly mapping their inputs into high-dimensional feature spaces
- First, we will derive *linear, hard-margin SVM* for linearly separable data, later for non-separable (soft margin SVM), and for nonlinear boundaries (kernel SVM)



Maximum Margin

Recall: logistic regression

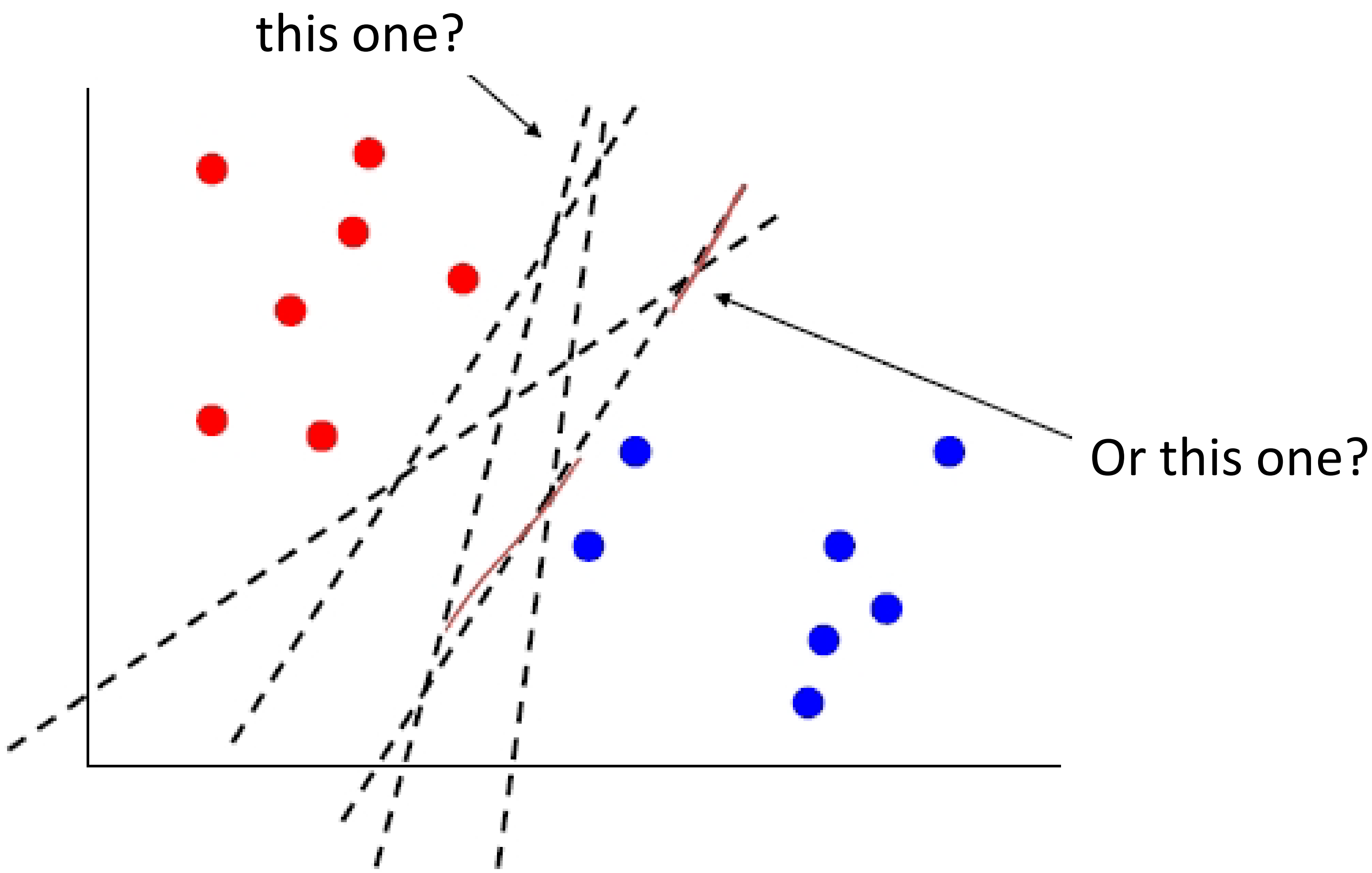
Decision boundary: $w^T x + b = 0$



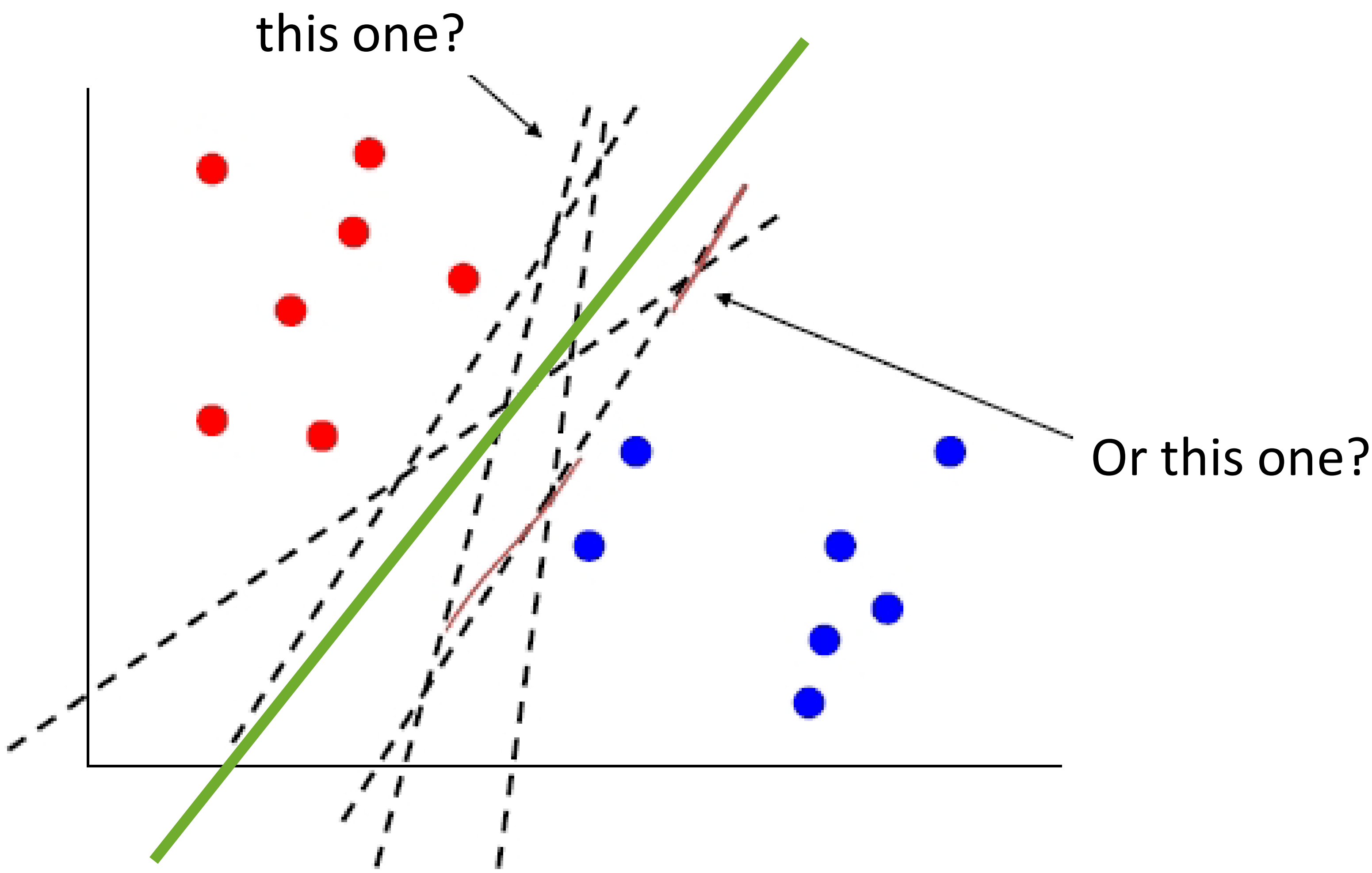
Many possible lines!

$$y = \begin{cases} +1 \text{ [red] if sign}(\mathbf{w}^T \mathbf{x} + b \geq 0) \\ -1 \text{[blue] if sign}(\mathbf{w}^T \mathbf{x} + b < 0) \end{cases}$$

Which classifier is best?

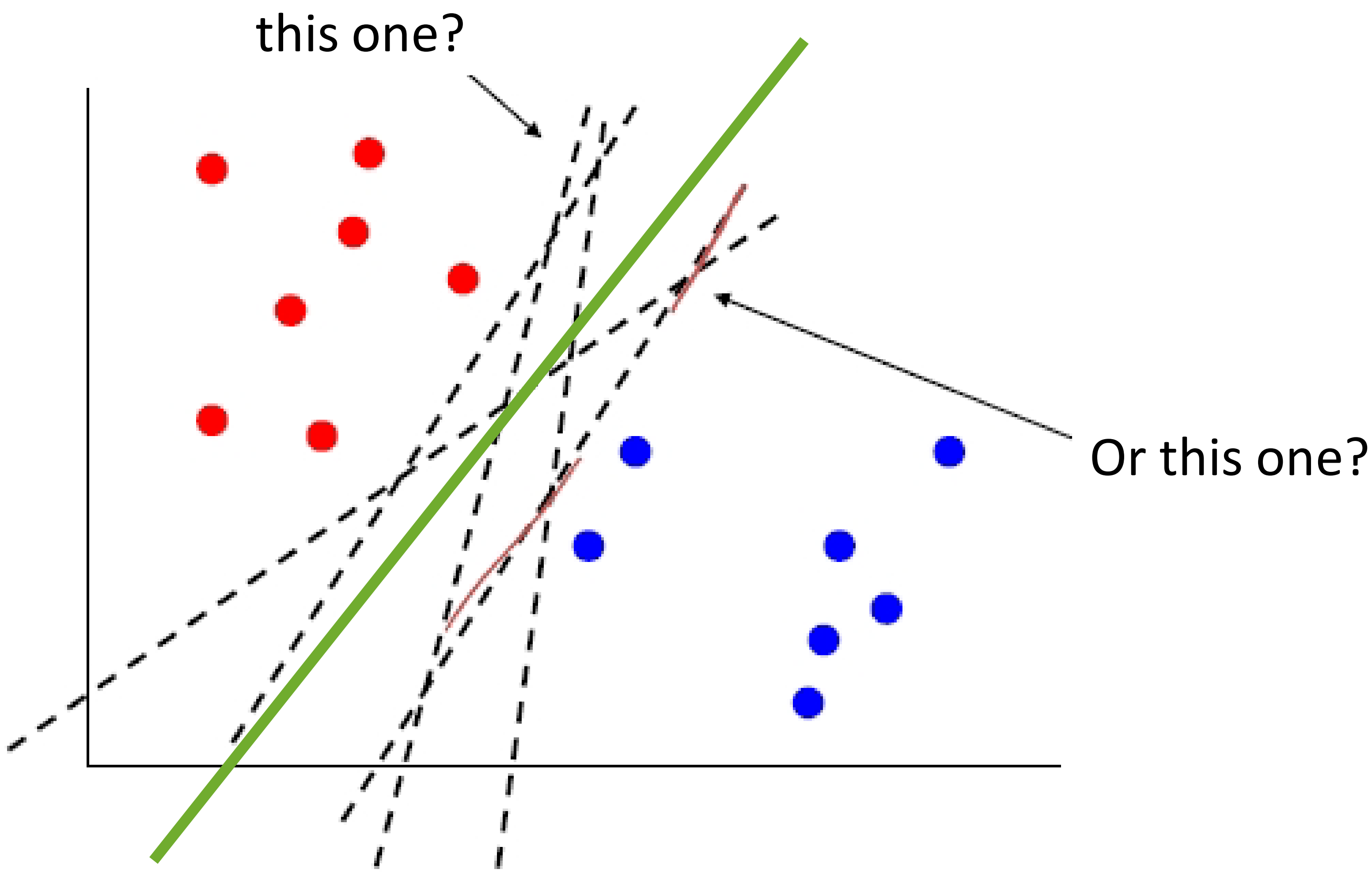


How about the one in the middle?



Intuitively, this classifier avoids misclassifying new test points generated from the same distribution as the training points

What is special about the green line?



It maximizes the margin between the two classes.

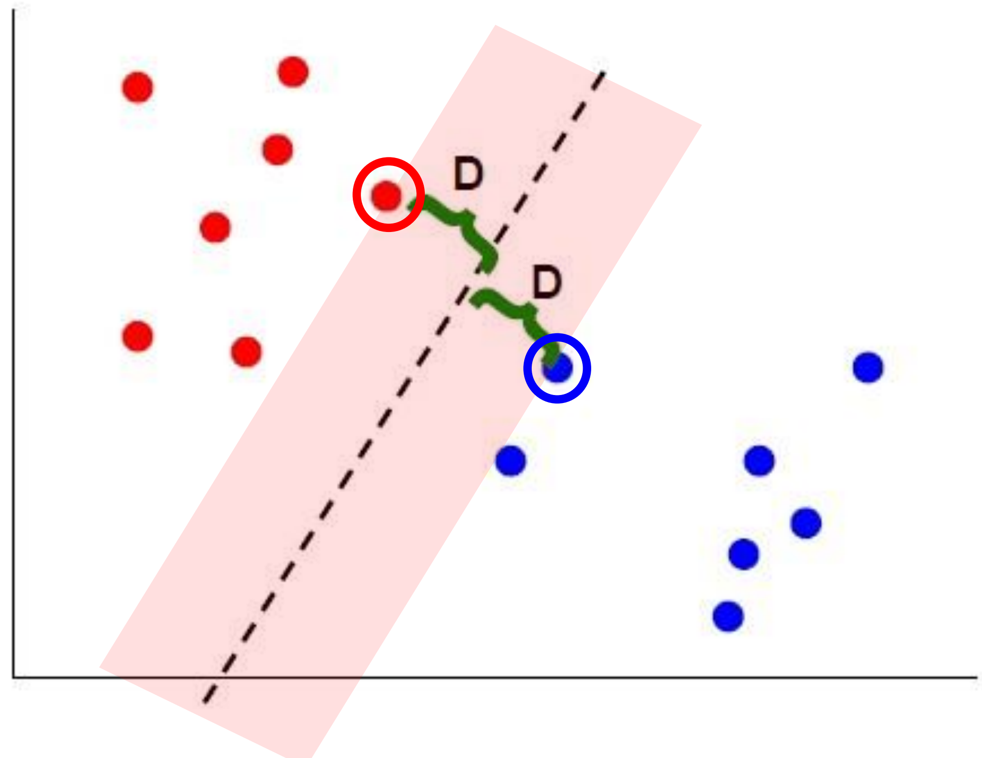
Max margin classification

Instead of fitting all the points, focus on boundary points

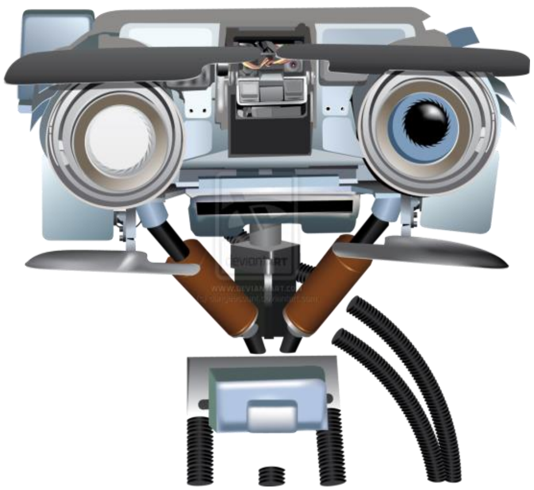
Aim: learn a boundary that leads to the largest margin (buffer) from points on both sides

Why: intuition; theoretical support: robust to small perturbations near the boundary

And works well in practice!



Subset of vectors that support (determine boundary) are called the support vectors (circled)



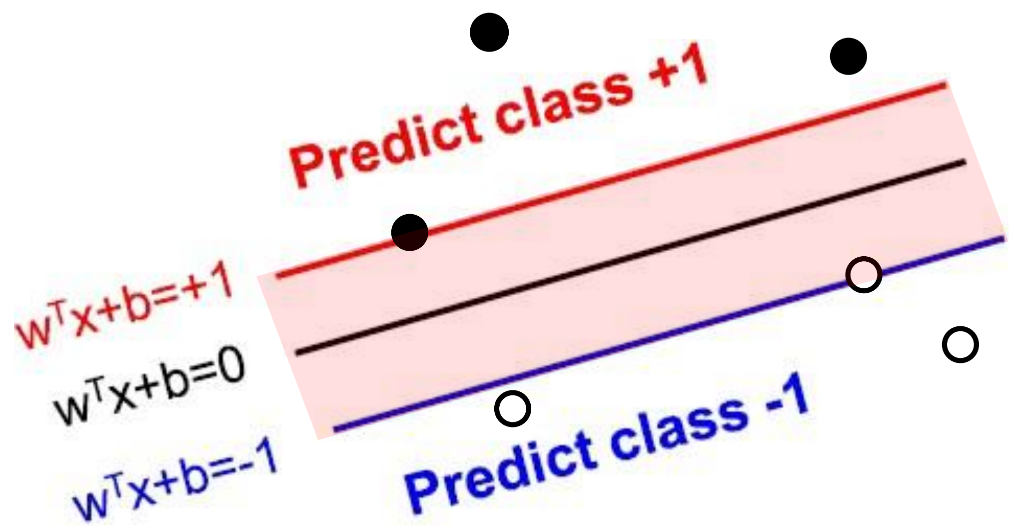
Max-Margin Classifier

Max Margin Classifier

"Expand" the decision boundary to include a margin (until we hit first point on either side)

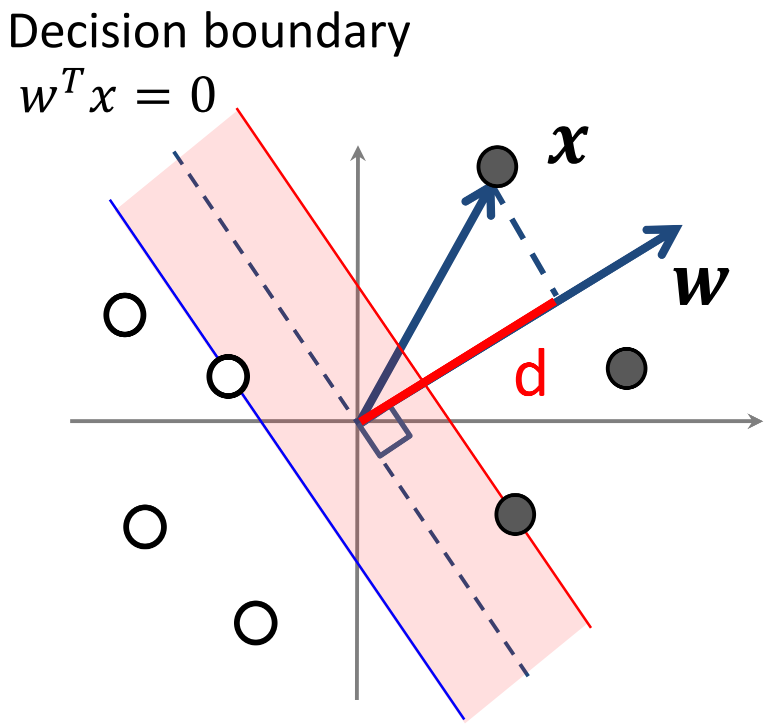
Use margin of 1

Inputs in the margins are of unknown class



Classify as +1	if	$w^Tx+b \geq 1$
Classify as -1	if	$w^Tx+b \leq -1$
Undefined	if	$-1 < w^Tx+b < 1$

Why is the margin = 1?

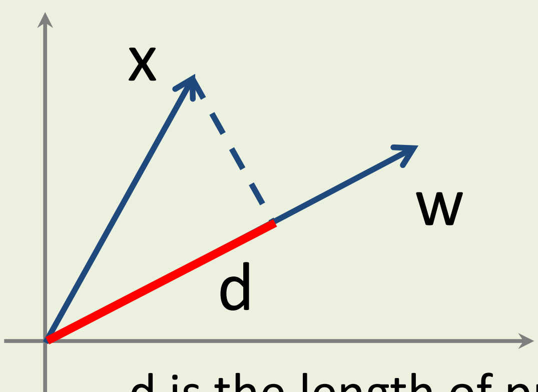


- Assume $b = 0$ for simplicity
- $\mathbf{w}$ is orthogonal to the decision plane
- Scaling margin and weight vector by the same constant c>0 does not change the inequality

$$\boldsymbol{w}^T \boldsymbol{x} \geq 1$$

$$c * \boldsymbol{w}^T \boldsymbol{x} \geq 1 * c$$

- So choose margin of 1 (arbitrary)



d is the length of projection

Aside: vector inner product

$$\boldsymbol{w}^T \boldsymbol{x} = \mathrm{w}_1 \mathrm{x}_1 + \mathrm{w}_2 \mathrm{x}_2$$

$$d = \frac{\boldsymbol{w}^T \boldsymbol{x}}{\|\boldsymbol{w}\|_2}$$

Computing the Margin

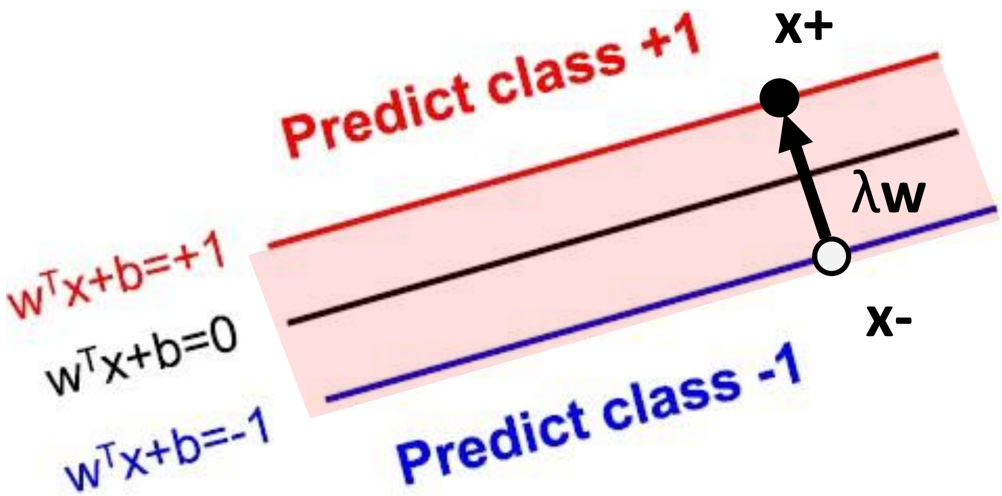
First note that the **w** vector is orthogonal to the +1 plane

If **u** and **v** are two points on that plane, then $\mathbf{w}^T(\mathbf{u}-\mathbf{v}) = 0$

Same is true for -1 plane

Also: for point $\mathbf{x}^+$ on +1 plane and $\mathbf{x}^-$ nearest point on -1 plane:

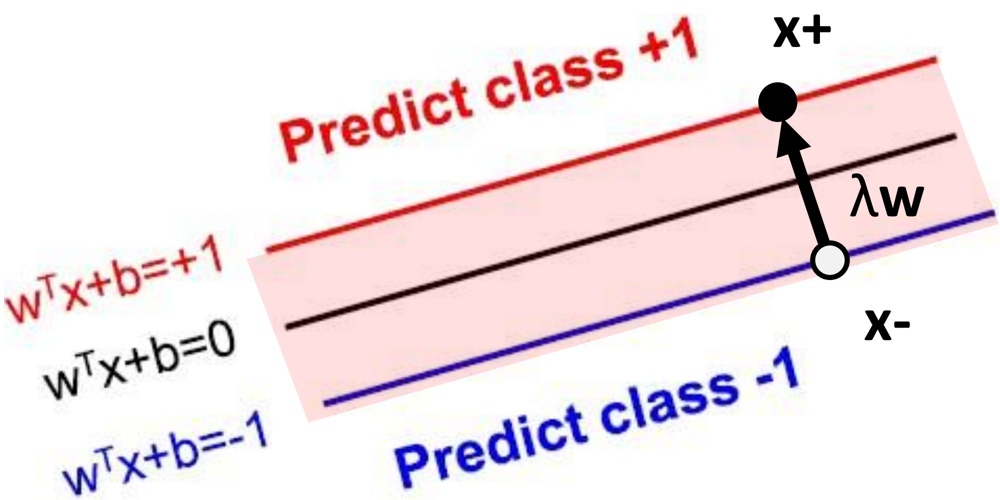
$$\mathbf{x}^+ = \lambda\mathbf{w} + \mathbf{x}^-$$



Computing the Margin

Also: for point **x+** on +1 plane and **x-**nearest point on -1 plane:

$$\mathbf{x}^{+} = \lambda \mathbf{w} + \mathbf{x}^{-}$$



$$\mathbf{w}^T \mathbf{x}^+ + b = 1$$

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

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

$$-1 + \lambda \mathbf{w}^T \mathbf{w} = 1$$

$$\lambda = 2 / \mathbf{w}^T \mathbf{w}$$

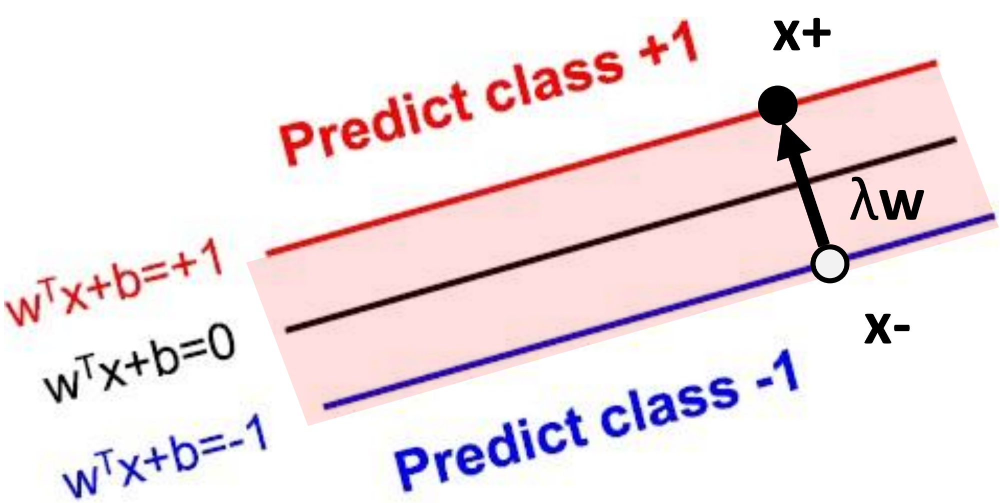
→ inversely proportional to $\mathbf{w}^{\mathrm{T}}\mathbf{w}$, the square of the length of $\mathbf{w}$

Computing the Margin

Define the margin M to be the distance between the +1 and -1 planes

We can now express this in terms of **w** →

to maximize the margin we minimize the length of **w**



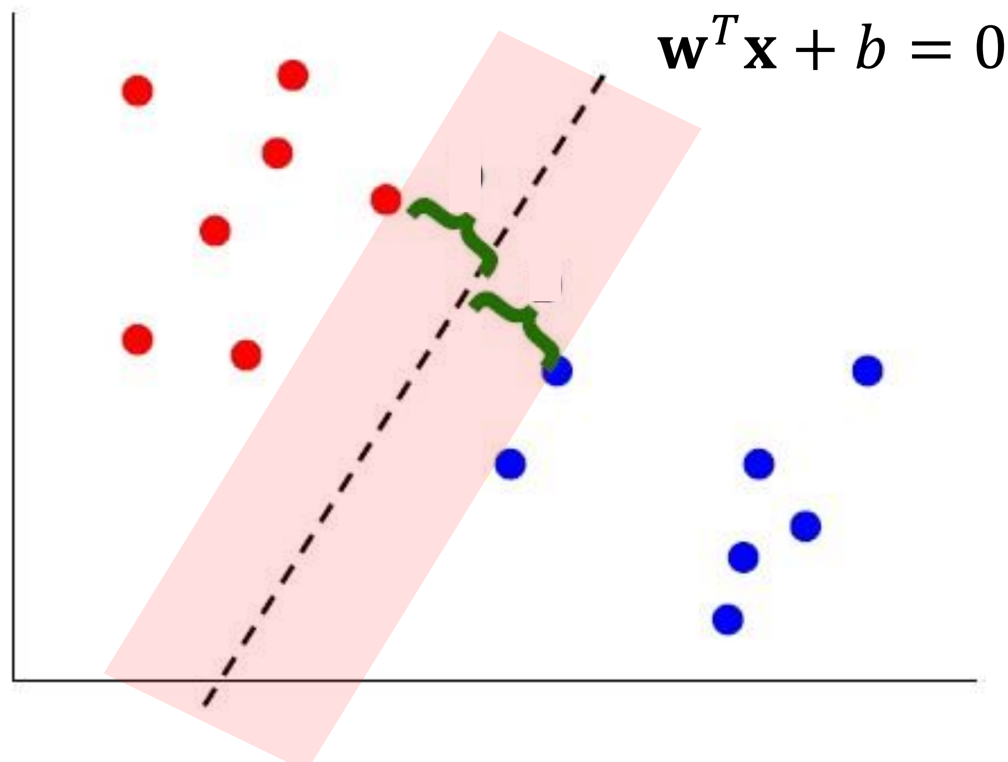
$$M = \left\| \mathbf{x}^{+} - \mathbf{x}^{-} \right\|$$

$$= \left\| \lambda \mathbf{w} \right\| = \lambda \sqrt{\mathbf{w}^T \mathbf{w}}$$

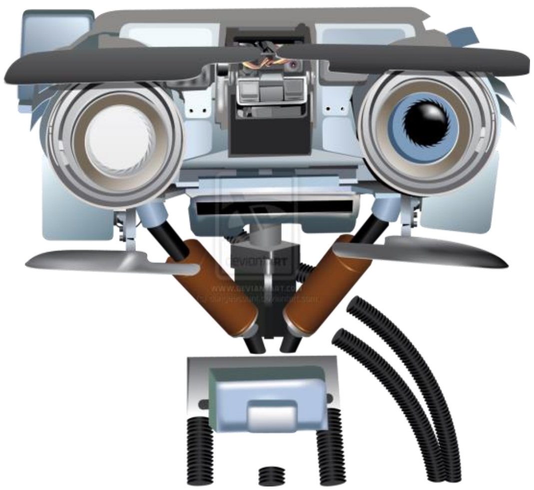
$$= 2 \frac{\sqrt{\mathbf{w}^T \mathbf{w}}}{\mathbf{w}^T \mathbf{w}} = \frac{2}{\sqrt{\mathbf{w}^T \mathbf{w}}}$$

Maximizing the margin is equivalent to regularization

To maximize the margin we minimize the length of $\mathbf{w}$, or $||\mathbf{w}||^2$



But not same as regularized logistic regression, the SVM loss is different! Only care about boundary points.

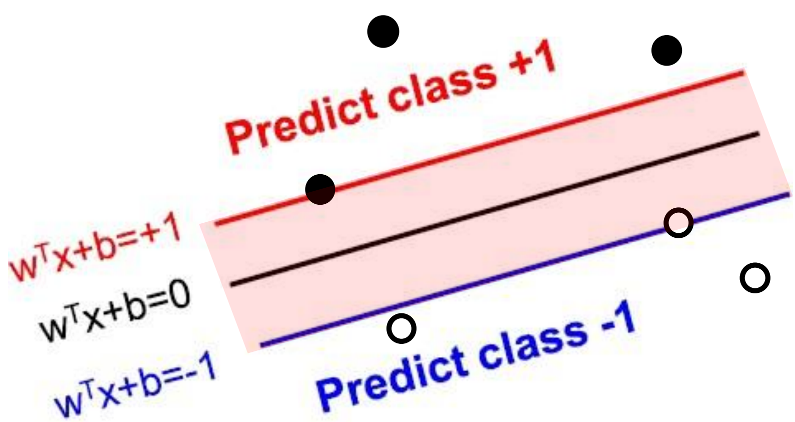


Linear SVM

Linear SVM Formulation

We can search for the optimal parameters (**w** and b) by finding a solution that:

1. Correctly classifies the training examples: $\{x_i,y_i\}$, i=1,..,n
2. Maximizes the margin (same as minimizing $\|\boldsymbol{w}\|^2$)



$$\min \ \frac{1}{2}\|\mathbf{w}\|^2$$

$$s.t. \ (\mathbf{w}^T\mathbf{x}_i + b)y_i \geq 1 \ \ \forall i$$

This is the primal formulation

Apply Lagrange multipliers: formulate equivalent problem

Lagrange Multipliers

Convert the primal constrained minimization to an unconstrained optimization problem: represent constraints as penalty terms:

$$\min_{\boldsymbol{w},b} \frac{1}{2}\|\boldsymbol{w}\|^2 + \boldsymbol{penalty\_term}$$

For data $\{(x_i,y_i)\}$ use the following penalty term:

$$\left\{\begin{matrix} 0 & \text{if } (\mathbf{w}^T\mathbf{x}_i + b)y_i \geq 1 \\ \infty & \text{otherwise} \end{matrix}\right\} = \max_{\alpha_i \geq 0} \alpha_i [1-(\mathbf{w}^T\mathbf{x}_i + b)y_i]$$

≤ 0 if constraint satisfied

Introduced Lagrange variables $\alpha_i \geq 0$; find ones that maximize term:

- If a constraint is satisfied, large α_i ensures smaller penalty
- If a constraint is violated, large α_i ensures larger penalty

Note, we are now minimizing with respect to **w** and b, and maximizing with respect to $\boldsymbol{\alpha}$ (additional parameters)

Lagrange Multipliers

Convert the primal constrained minimization to an unconstrained optimization problem: represent constraints as penalty terms:

$$\min_{\boldsymbol{w},b} \frac{1}{2}\|\boldsymbol{w}\|^2 + \boldsymbol{penalty\_term}$$

For data $\{(x_i,y_i)\}$ use the following penalty term:

$$\begin{Bmatrix} 0 & \text{if } (\mathbf{w}^T\mathbf{x}_i + b)y_i \geq 1 \\ \infty & \text{otherwise} \end{Bmatrix} = \max_{\alpha_i \geq 0} \alpha_i[1-(\mathbf{w}^T\mathbf{x}_i + b)y_i]$$

Rewrite the minimization problem:

$$\min_{\mathbf{w},b} \{\frac{1}{2}\|\mathbf{w}\|^2 + \sum_{i=1}^{n} \max_{\alpha_i \geq 0} \alpha_i[1-(\mathbf{w}^T\mathbf{x}_i + b)y_i]\}$$

Where $\{\alpha_i\}$ are the Lagrange multipliers

$$\min_{\mathbf{w},b} \max_{\alpha_i \geq 0} \{\frac{1}{2}\|\mathbf{w}\|^2 + \sum_{i=1}^{n} \alpha_i[1-(\mathbf{w}^T\mathbf{x}_i + b)y_i]\}$$

Solution to Linear SVM

Swap the 'max' and 'min':

$$\max_{\alpha_i \geq 0} \min_{\mathbf{w},b} \{\frac{1}{2}\|\mathbf{w}\|^2 + \sum_{i=1}^{n} \alpha_i [1 - (\mathbf{w}^T \mathbf{x}_i + b) y_i]\}$$

$$= \max_{\alpha_i \geq 0} \min_{\mathbf{w},b} J(\mathbf{w}, b; \alpha)$$

First minimize J() w.r.t. {**w**,b} for any fixed setting of the Lagrange multipliers:

$$\frac{\partial}{\partial \mathbf{w}} J(\mathbf{w}, b; \alpha) = \mathbf{w} - \sum_{i=1}^{n} \alpha_i \mathbf{x}_i y_i = 0$$

$$\frac{\partial}{\partial b} J(\mathbf{w}, b; \alpha) = -\sum_{i=1}^{n} \alpha_i y_i = 0$$

Then substitute back into J() and simplify to get final optimization:

$$L = \max_{\alpha_i \geq 0} \{\sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} y_i y_j \alpha_i \alpha_j (\mathbf{x}_i \cdot \mathbf{x}_j)\}$$

Dual Problem

Final optimization: maximize this loss over α_i's:

only dot products of data points needed

$$L = \max_{\alpha_i \geq 0} \{ \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} y_i y_j \alpha_i \alpha_j (\mathbf{x}_i \cdot \mathbf{x}_j) \}$$

subject to $\alpha_i \geq 0; \quad \sum_{i=1}^{n} \alpha_i y_i = 0$

Then use the obtained α_i's to solve for the weights and bias

$$\mathbf{w} = \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i \qquad\qquad b = y_i - \mathbf{w}^{\mathrm{T}} \mathbf{x}_{\mathrm{i}} \quad \forall i$$

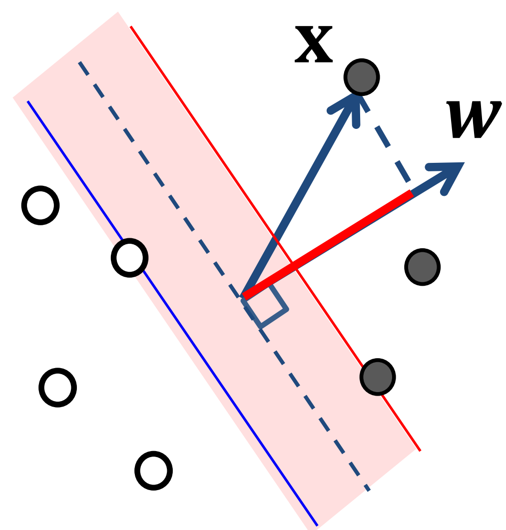
Prediction on Test Example

Now we have the solution for the weights and bias

$$\mathbf{w} = \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i \qquad\qquad b = y_i - \mathbf{w}^{\mathrm{T}}\mathbf{x}_{\mathrm{i}} \quad \forall i$$

Given a new input example $\mathbf{x}$, classify it as

$+1$ if $\mathbf{w}^{\mathrm{T}}\mathbf{x} + b \geq 1$, or
-1 if $\mathbf{w}^{\mathrm{T}}\mathbf{x} + b \leq -1$



In practice, predict $y = \mathrm{sign}[\mathbf{w}^{\mathrm{T}}\mathbf{x} + b]$

Dual vs Primal SVM

n is the number of training points, d is dimension of $\mathbf{x}, \mathbf{w}$

Primal problem: for $\mathbf{w} \in \mathbb{R}^{\boldsymbol{d}}$, hyperparameter C, the unconstrained version is

$$\min \ \frac{1}{2}\|\mathbf{w}\|^2 \quad s.t. \ (\mathbf{w}^T\mathbf{x}_i + b)y_i \geq 1$$

Dual problem: for $\boldsymbol{\alpha} \in \mathbb{R}^{\boldsymbol{n}}$

$$L = \max_{\alpha_i \geq 0}\{\sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{n} y_i y_j \alpha_i \alpha_j (\mathbf{x}_i \cdot \mathbf{x}_j)\} \quad s.t. \ \alpha_i \geq 0; \ \sum_{i=1}^{n}\alpha_i y_i = 0$$

- Efficiency: need to learn d parameters for primal, n for dual
- Dual form only involves data terms $\mathbf{x}_{\boldsymbol{i}}^{\mathbf{T}}\mathbf{x}_{\mathbf{j}}$

Dual vs Primal SVM

- Dual: quadratic programming problem in which we optimize a quadratic function of $\boldsymbol{\alpha}$ subject to a set of inequality constraints

- The solution to a quadratic programming problem in d variables in general has computational complexity that is $O(d^3)$

- If *d* is smaller than the number *n* of data points, the move to the dual problem appears disadvantageous

- However, it allows the model to be reformulated using kernels which allow *infinite* feature spaces (more on this later)

Bishop Ch. 7

Dual vs Primal SVM

- Most of the SVM literature and software solves the Lagrange dual problem formulation

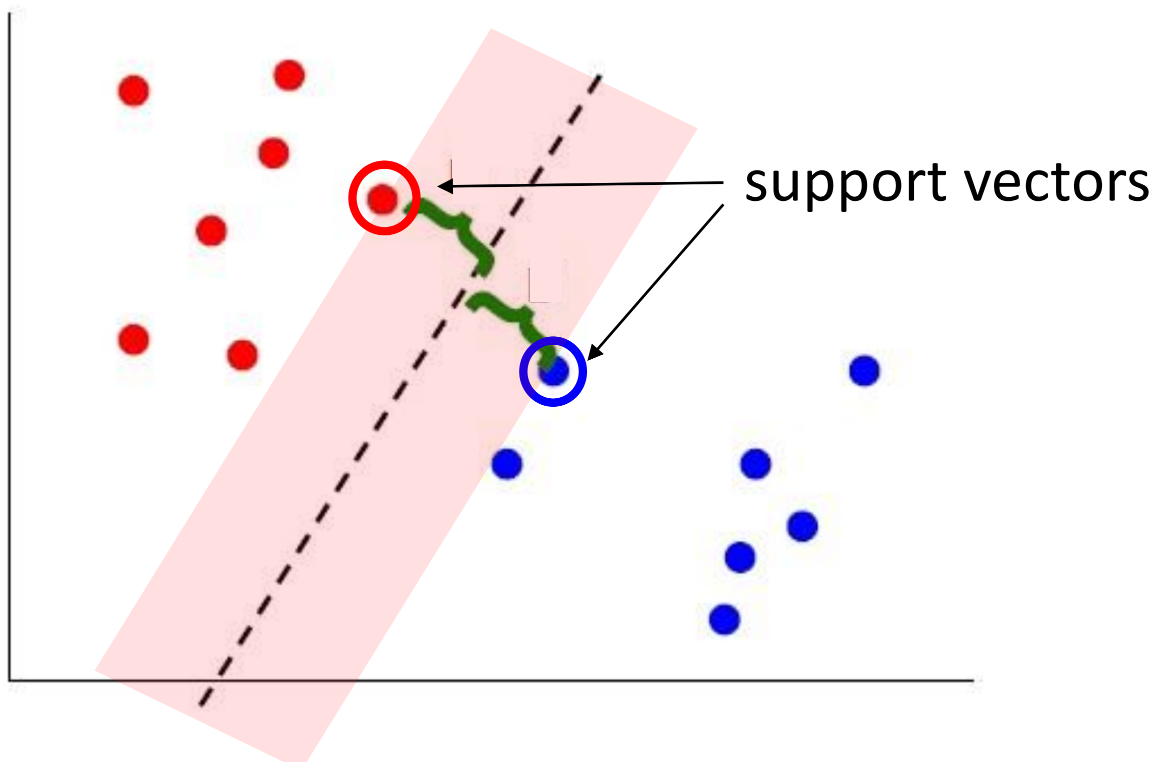
- Why prefer solving the dual problem over the primal?
  - expresses solution in terms of dot products of data points, allowing kernels
  - historical reasons

For an in-depth discussion refer to
http://olivier.chapelle.cc/pub/neco07.pdf (optional reading)

Support Vectors

Only a small subset of α_i's will be nonzero, and the corresponding x_i's are the support vectors **S**

$$y = \text{sign}[b + \mathbf{x} \cdot (\sum_{i=1}^{n} y_i \alpha_i \mathbf{x}_i)] = \text{sign}[b + \mathbf{x} \cdot (\sum_{i \in \mathbf{S}} y_i \alpha_i \mathbf{x}_i)]$$



Summary of Linear SVM

- Binary and linear separable classification
- Linear classifier with maximal margin
- Training SVM by maximizing

$$\sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{n} y_i y_j \alpha_i \alpha_j (\mathbf{x}_i \cdot \mathbf{x}_j)$$

$$s.t. \quad \alpha_i \geq 0; \sum_{i=1}^{n} \alpha_i y_i = 0$$

- Weights: $\mathbf{w} = \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i$
- Only a small subset of α_i's will be nonzero, and the corresponding x_i's are the support vectors **S**
- Prediction on a new example:

$$y = \text{sign}[b + \mathbf{x} \cdot (\sum_{i=1}^{n} y_i \alpha_i \mathbf{x}_i)] = \text{sign}[b + \mathbf{x} \cdot (\sum_{i \in \mathbf{S}} y_i \alpha_i \mathbf{x}_i)]$$


do you want to enforce social distancing?
yes, please
use SVM to maximize the margin

Software Needed to Access SCC

1. **Mac:**

Install xQuartz https://www.xquartz.org/

Very important: logoff your computer and login back again to finish the installation process.

2. **Windows:**

Install mobaXtern (https://mobaxterm.mobatek.net/download-home-edition.html)

Select "Installer Edition"

Very important: once the installer file is downloaded unzip it first and only then install the program from the unzipped folder.