Support Vector Machine

Machine Learning 10-601B

Seyoung Kim

Many of these slides are derived fromTom Mitchell, Ziv Bar-Joseph. Thanks!

Types of classifiers

- We can divide the large variety of classification approaches into roughly three major types

1. Instance based classifiers
   - Use observation directly (no models)
   - e.g. K nearest neighbors

2. Classifiers based on generative models:
   - build a generative statistical model
   - e.g., Naïve Bayes classifier, classifiers derived from Bayesian networks

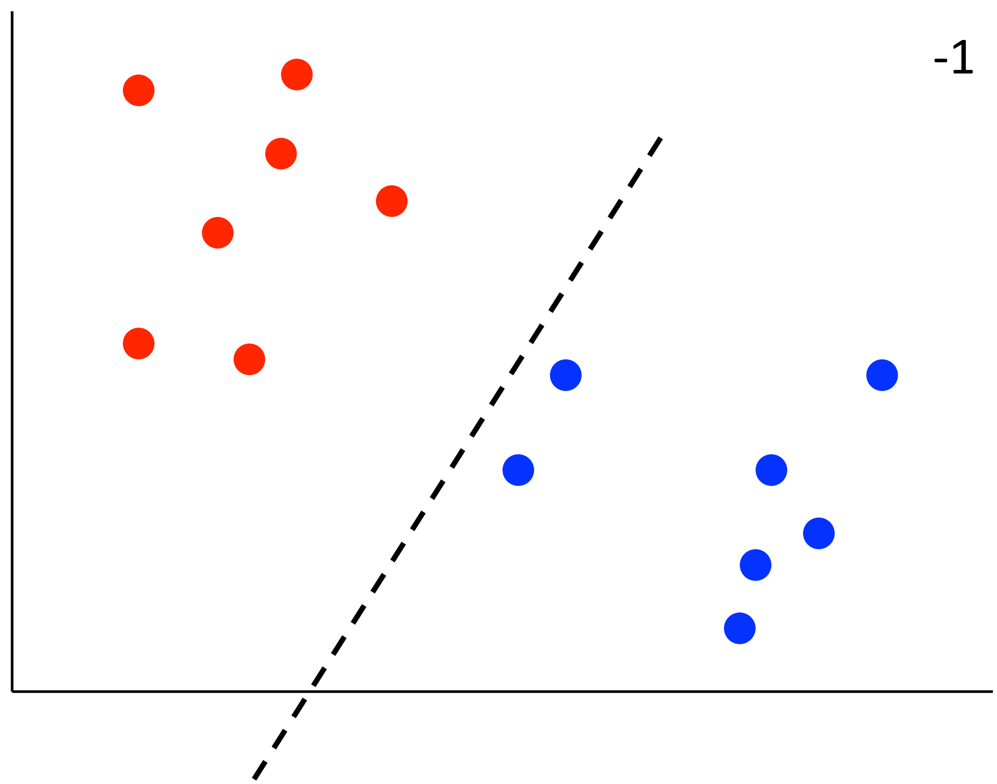
3. Classifiers based on discriminative models:
   - directly estimate a decision rule/boundary
   - e.g., decision tree, perceptron, logistic regression

Linear Classifiers

Recall logistic regression

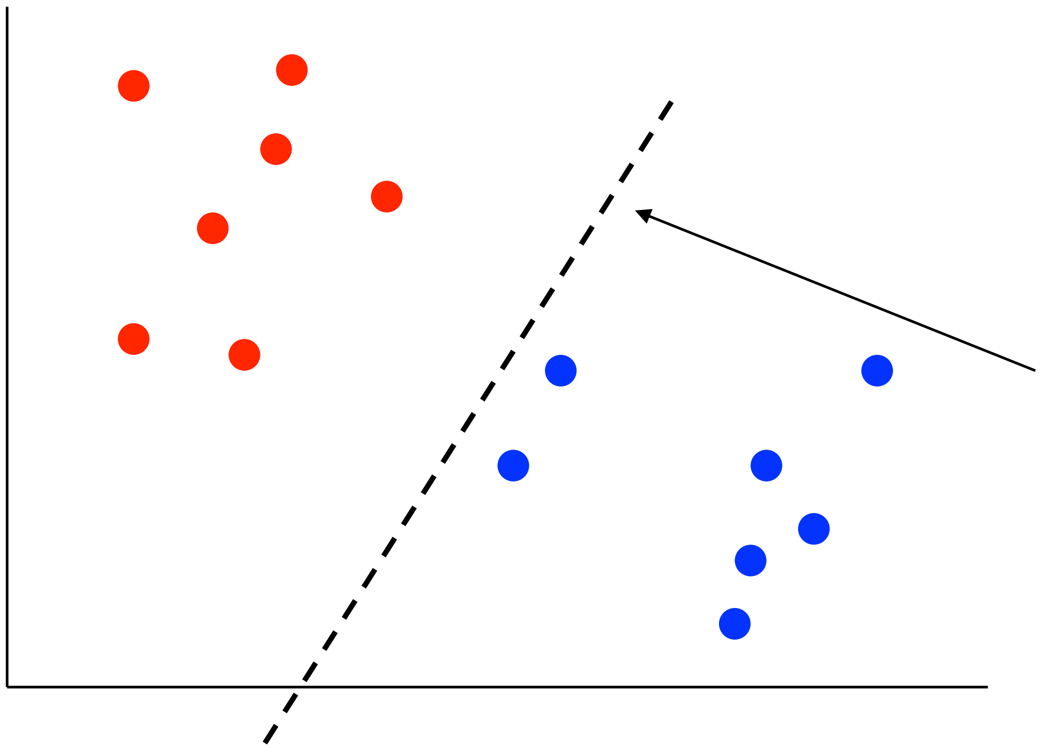
+1 if sign($w^T x+b$)≥0

-1 if sign($w^T x+b$)<0



Linear Classifiers

Recall logistic regression



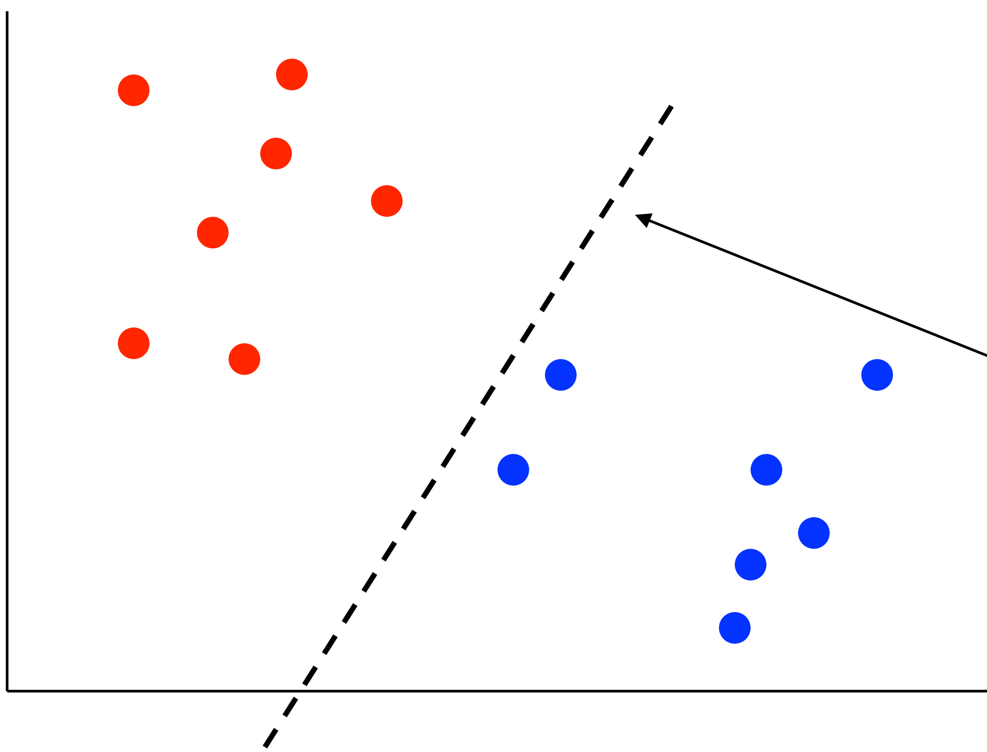
Line closer to the blue nodes since many of them are far away from the boundary

Linear Classifiers

Recall logistic regression

$$\min_w \sum_i Loss(y^i, w^T x^i)$$

Errors over all samples



Line closer to the blue nodes since many of them are far away from the boundary

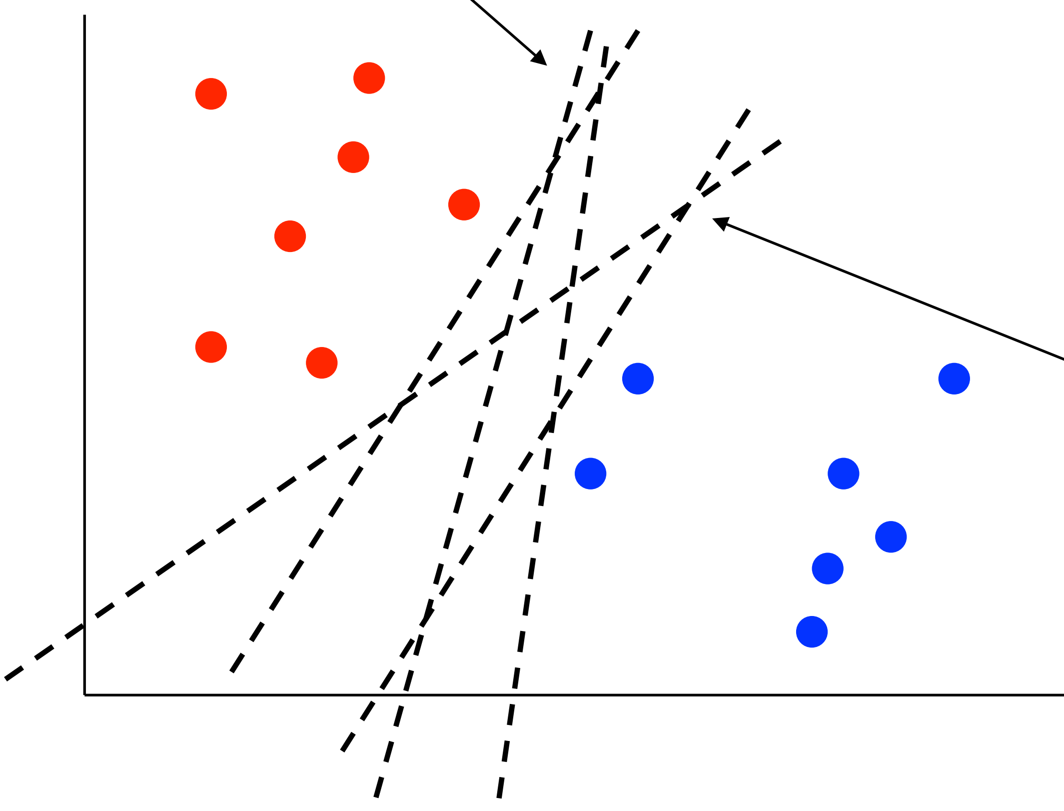
Linear Classifiers

Recall logistic regression

$$\min_w \sum_i Loss(y^i, w^T x^i)$$

Errors over all samples

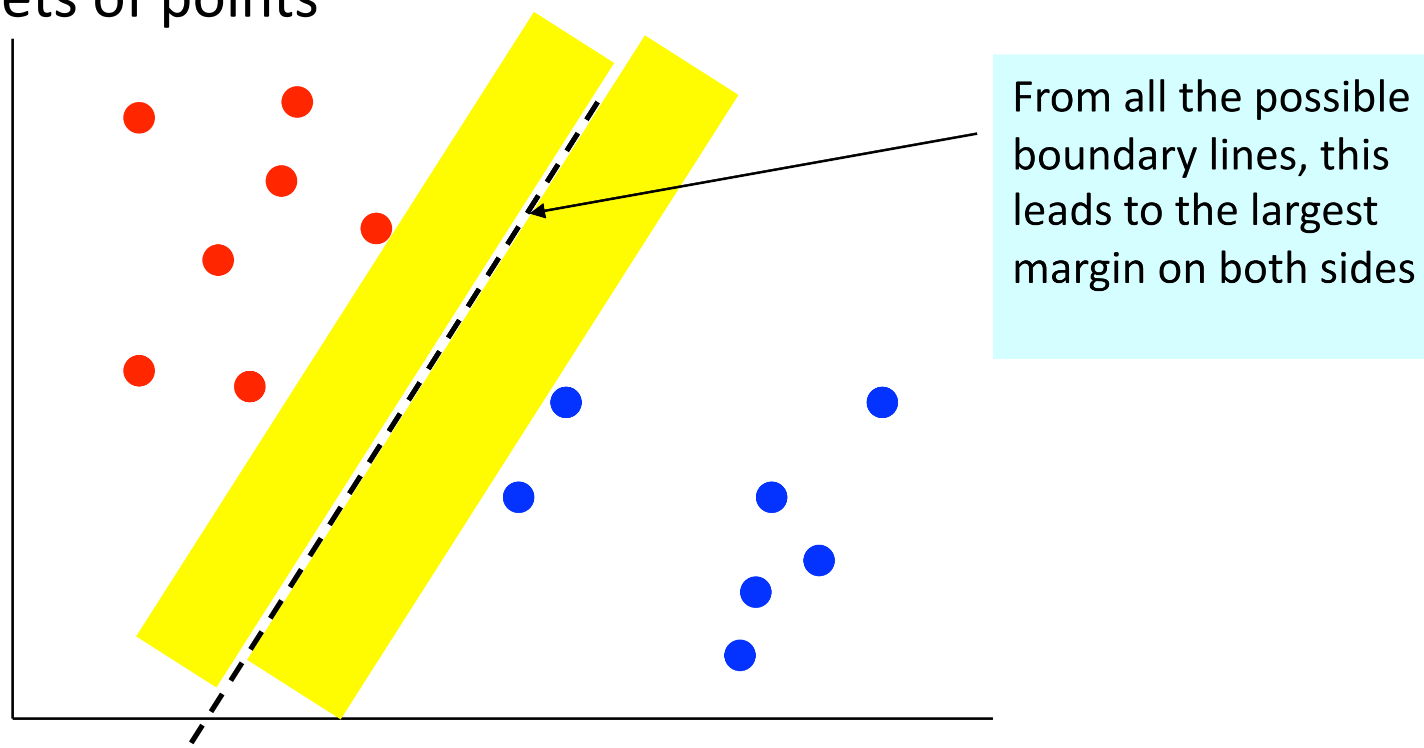
Many more possible classifiers



Line closer to the blue nodes since many of them are far away from the boundary

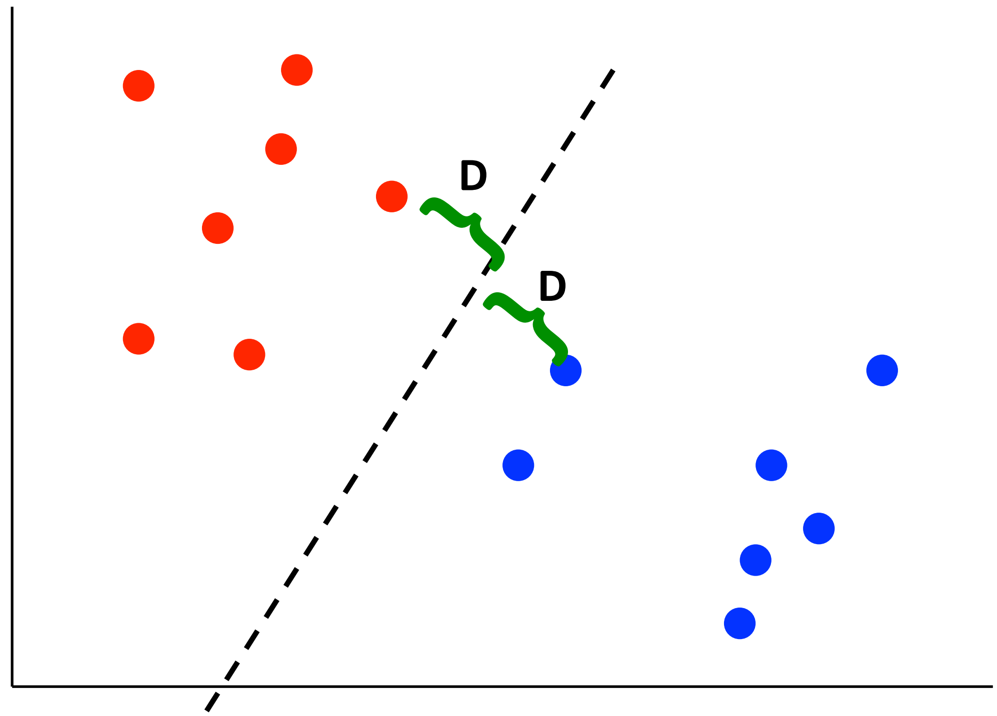
Max margin classifiers

- 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 both sets of points

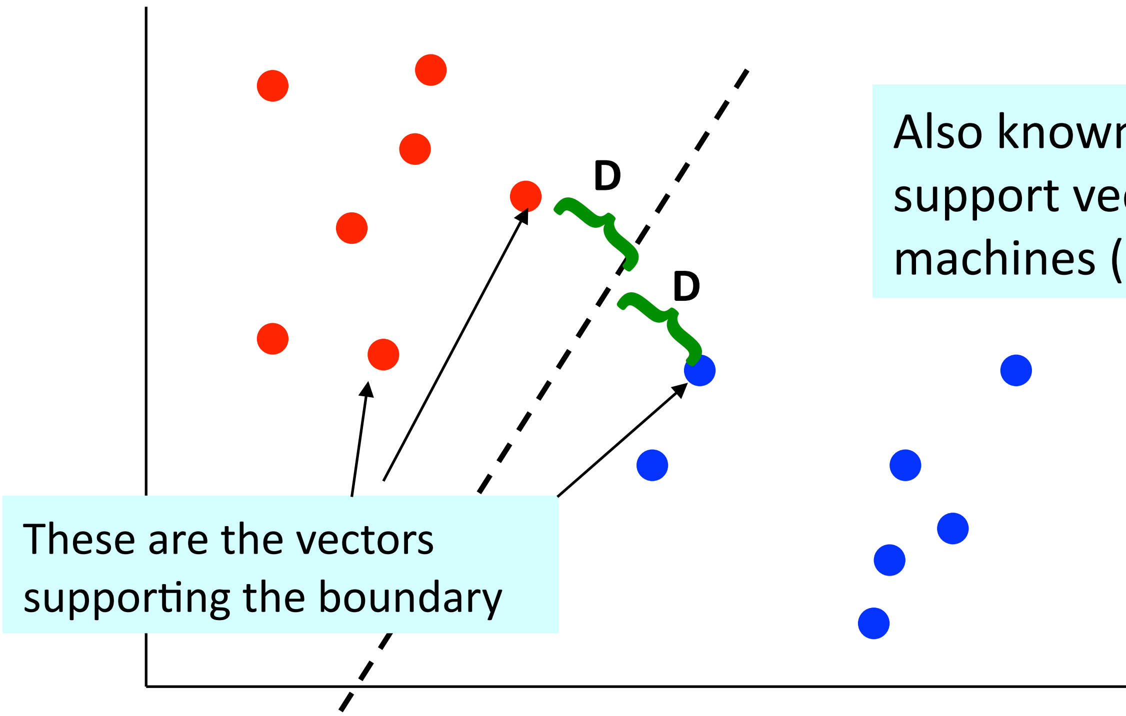


Why?

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

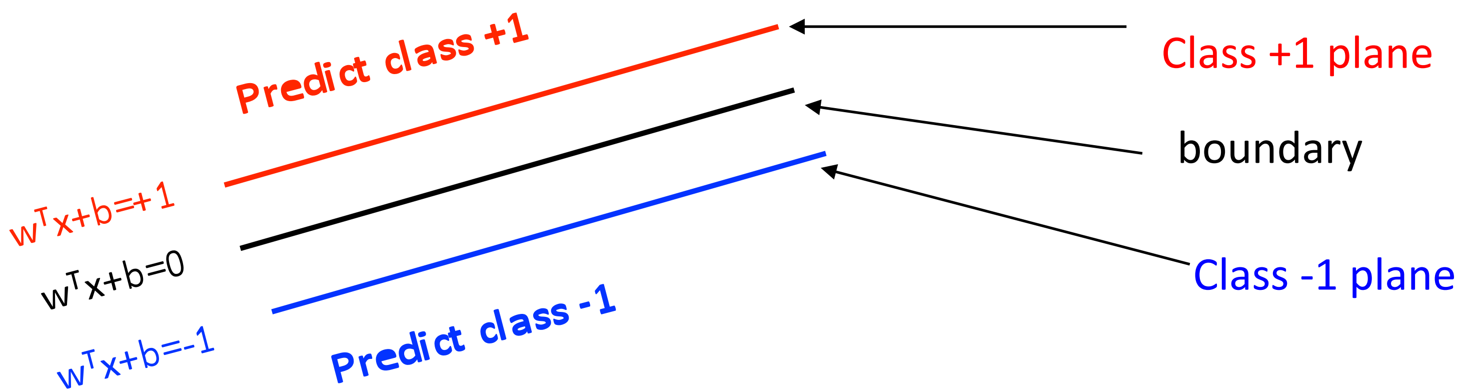
Max margin classifiers

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



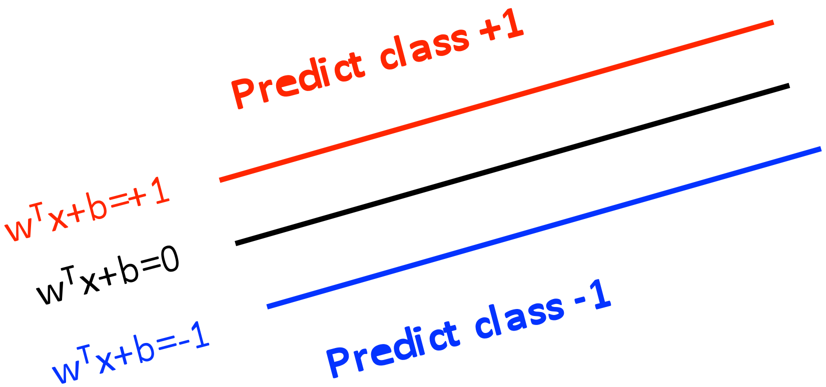
Also known as linear support vector machines (SVMs)

Specifying a max margin classifier



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

Specifying a max margin classifier

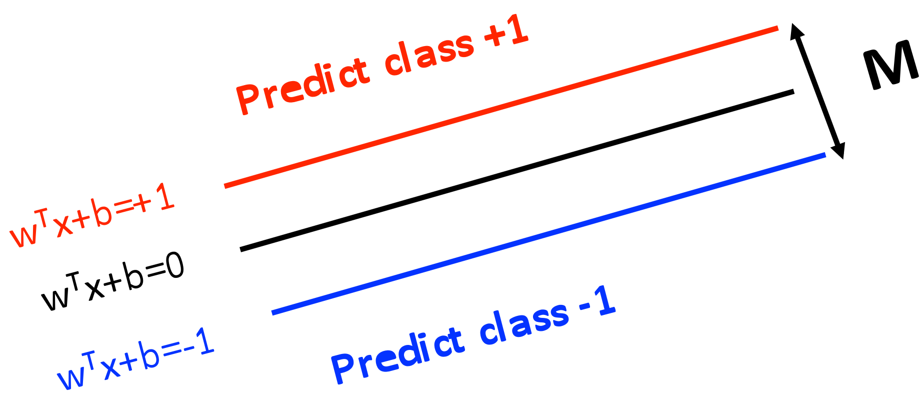


Is the linear separation assumption realistic?

We will deal with this shortly, but let's assume for now data are linearly separable

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

Maximizing the margin



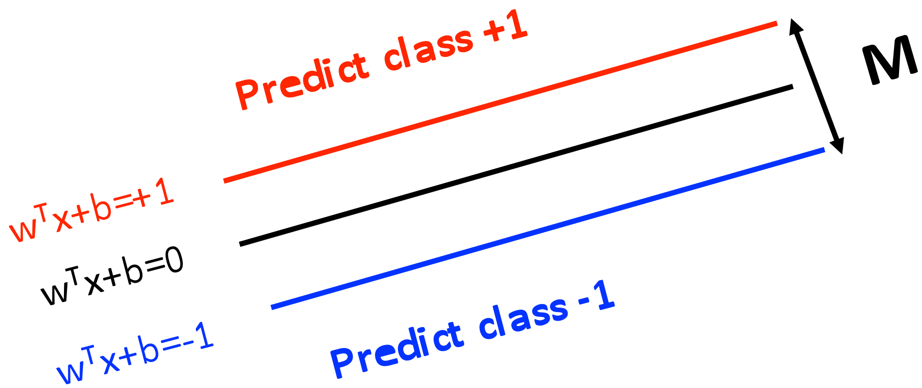
Classify as +1 if $w^Tx+b \geq 1$

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

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

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

Maximizing the margin



Classify as +1 if $w^Tx+b \geq 1$

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

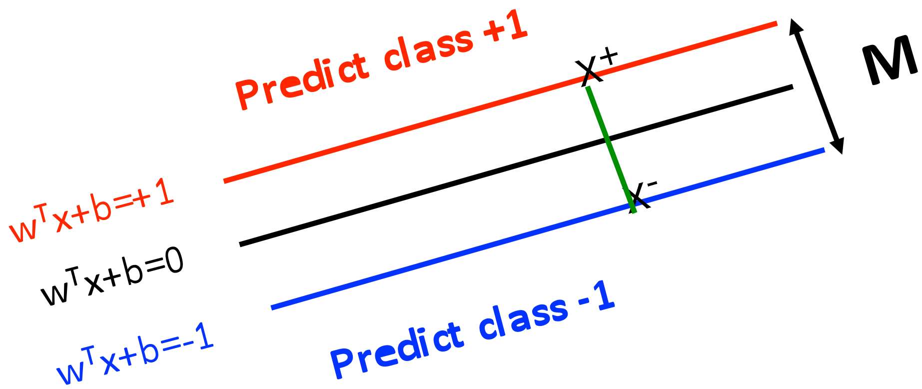
Undefined if $-1 < w^Tx+b < 1$

- 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



Classify as +1 if $w^Tx+b \geq 1$

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

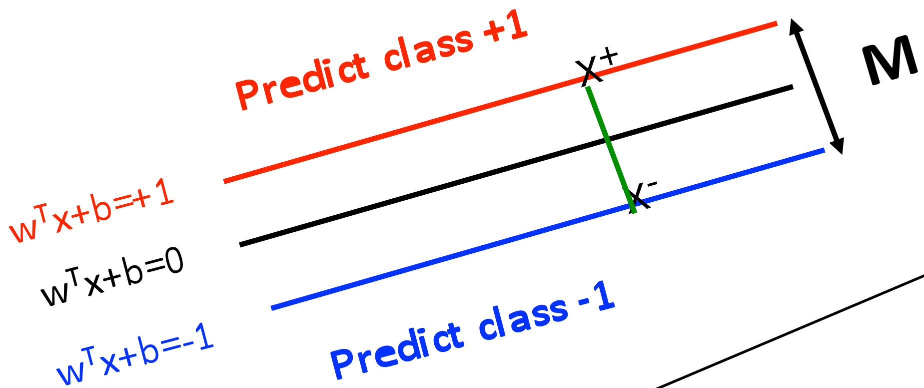
Undefined if $-1 < w^Tx+b < 1$

- 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



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

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

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

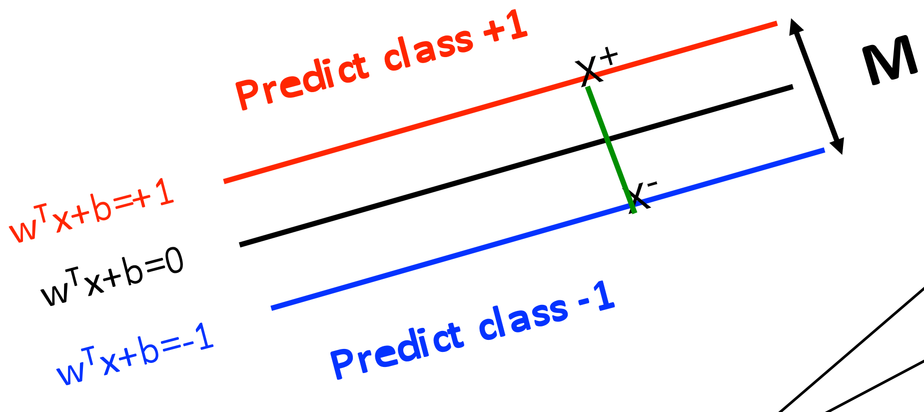
$\Rightarrow w^T x^- + b + \lambda w^T w = +1$

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

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

We can now define M in terms of w and b

Putting it together



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

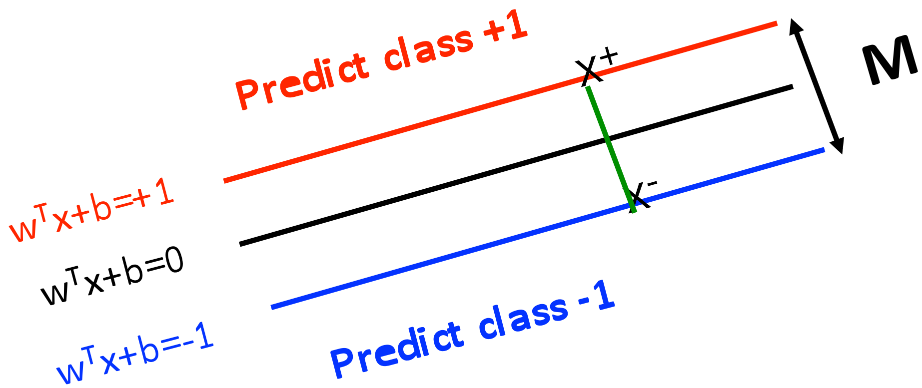
$$\Rightarrow M = |\lambda w| = \lambda |w| = \lambda \sqrt{w^T w}$$

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

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

We can now define M in terms of w and b

Finding the optimal parameters



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

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^Tw)

Quadratic programming (QP)

Quadratic programming solves optimization problems of the following form:

$$\min_U \frac{u^T R u}{2} + d^T u + c$$

Quadratic term

subject to n inequality constraints:

$$a_{11}u_1 + a_{12}u_2 + \ldots \leq b_1$$

$$\vdots \qquad \vdots \qquad \vdots$$

$$a_{n1}u_1 + a_{n2}u_2 + \ldots \leq b_n$$

and k equality constraints:

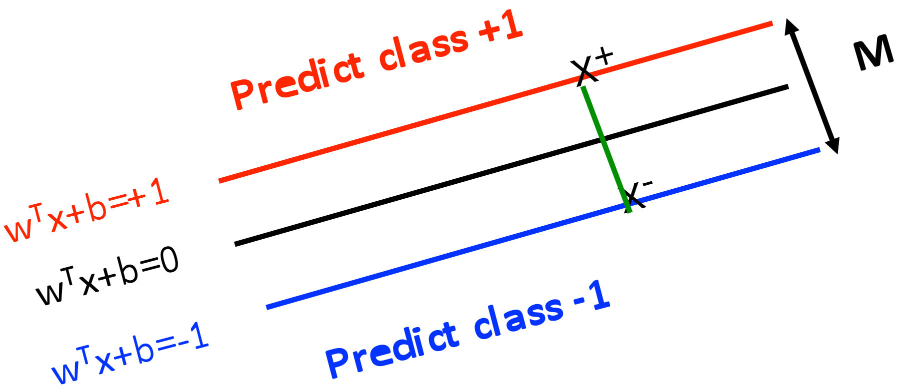
$$a_{n+1,1}u_1 + a_{n+1,2}u_2 + \ldots = b_{n+1}$$

$$\vdots \qquad \vdots \qquad \vdots$$

$$a_{n+k,1}u_1 + a_{n+k,2}u_2 + \ldots = b_{n+k}$$

When a problem can be specified as a QP problem we can use generic solvers that are better than gradient descent or simulated annealing

SVM as a QP problem



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

$$\min_U \frac{u^T R u}{2} + d^T u + c$$

subject to n inequality constraints:

$$a_{11}u_1 + a_{12}u_2 + \ldots \leq b_1$$
$$\vdots \qquad \vdots \qquad \vdots$$
$$a_{n1}u_1 + a_{n2}u_2 + \ldots \leq b_n$$

and k equivalency constraints:

$$a_{n+1,1}u_1 + a_{n+1,2}u_2 + \ldots = b_{n+1}$$
$$\vdots \qquad \vdots \qquad \vdots$$
$$a_{n+k,1}u_1 + a_{n+k,2}u_2 + \ldots = b_{n+k}$$

Min $(w^T w)/2$

subject to the following inequality constraints:

For all x in class + 1

$w^T x+b \geq 1$

For all x in class - 1

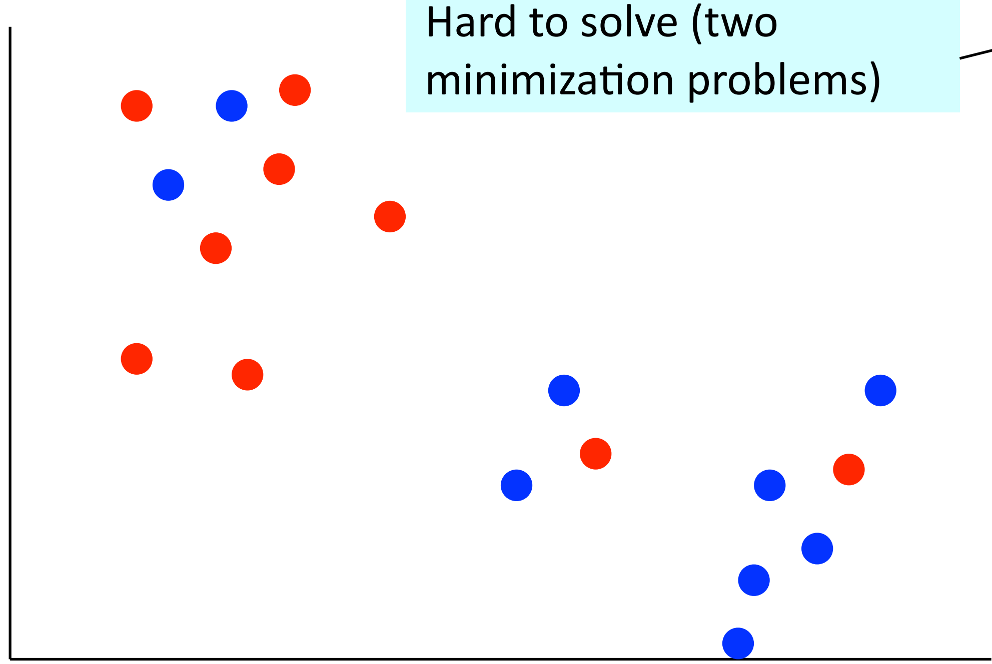
$w^T x+b \leq -1$

A total of n constraints if we have n input samples

Non linearly 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

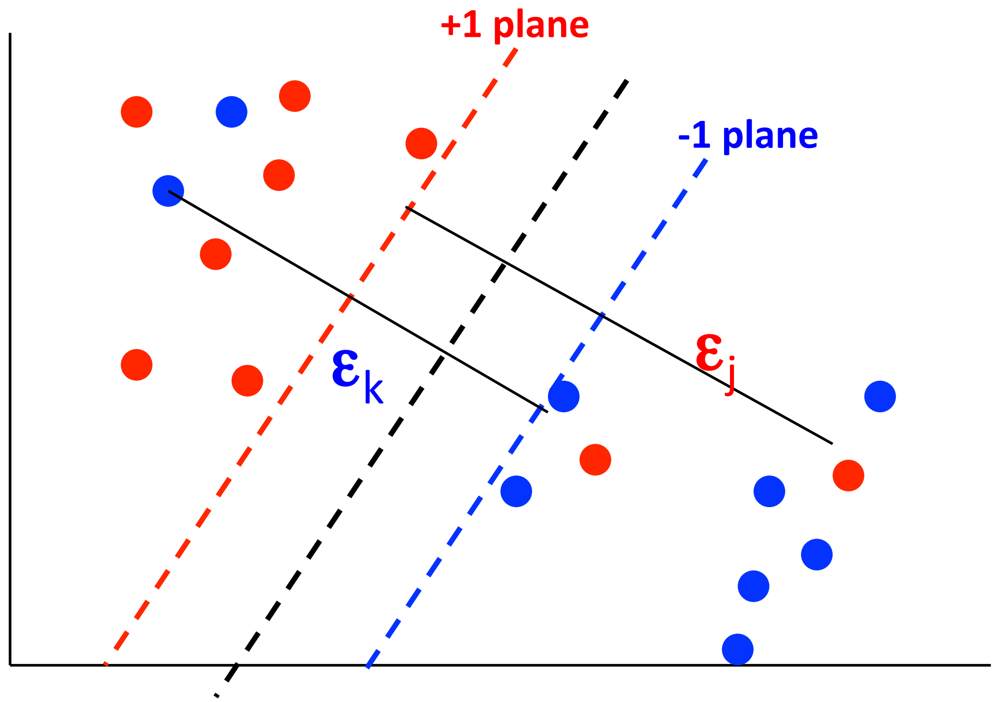
- Penalize training errors:

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

Hard to encode in a QP problem

Non linearly 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:

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

subject to the following inequality constraints:

For all x_i in class + 1

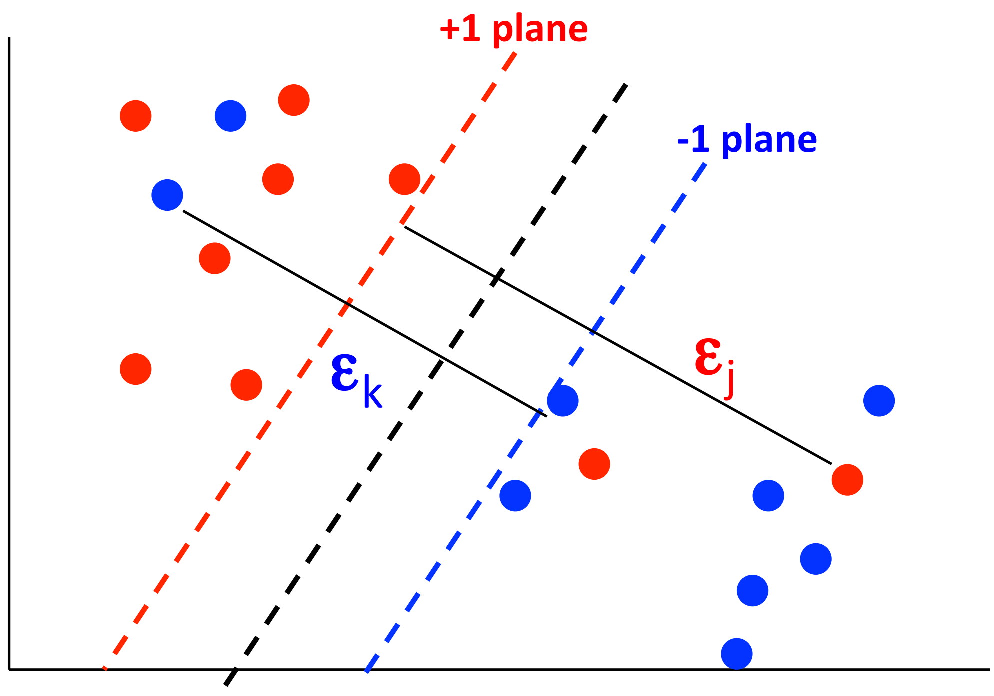
$$w^T x + b \geq 1 - \varepsilon_i$$

For all x_i in class - 1

$$w^T x + b \leq -1 + \varepsilon_i$$

Wait. Are we missing something?

Final optimization for non linearly separable case



The new optimization problem is:

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

subject to the following inequality constraints:

For all x_i in class + 1

$w^T x + b \geq 1 - \varepsilon_i$

For all x_i in class - 1

$w^T x + b \leq -1 + \varepsilon_i$

A total of n constraints

For all i

$\varepsilon_i \geq 0$

Another n constraints

Where we are

Two optimization problems: For the separable and non separable cases

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

For all x in class + 1

$w^T x + b \geq 1$

For all x in class - 1

$w^T x + b \leq -1$

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

For all x_i in class + 1

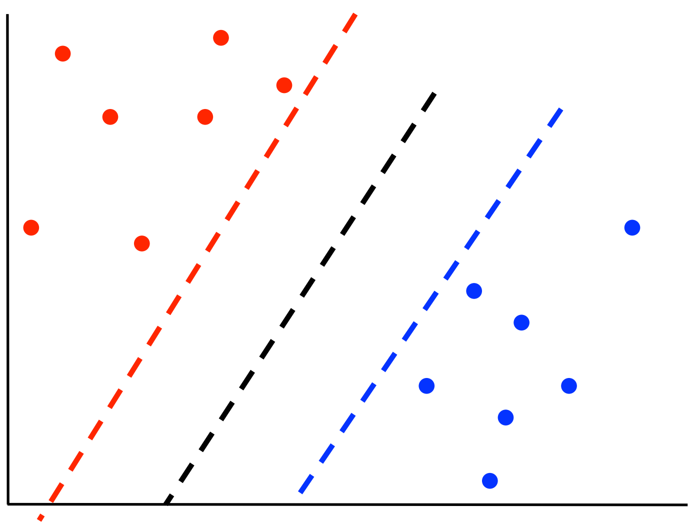
$w^T x + b \geq 1 - \varepsilon_i$

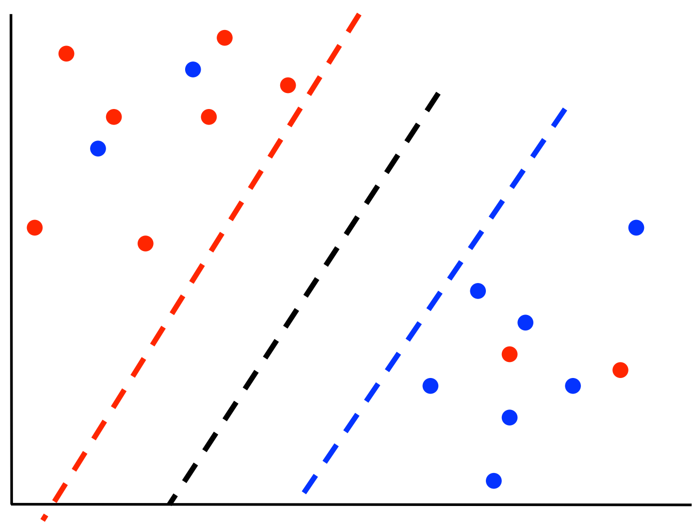
For all x_i in class - 1

$w^T x + b \leq -1 + \varepsilon_i$

For all i

$\varepsilon_i \geq 0$





An alternative (dual) 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 multipliers to encode it as part of our minimization problem

Min $(w^T w)/2$

For all x in class +1

$w^T x+b \geq 1$

For all x in class -1

$w^T x+b \leq -1$

Why? ⇓

Min $(w^T w)/2$

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

An alternative (dual) representation of the SVM QP

Min $(w^T w)/2$

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

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

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

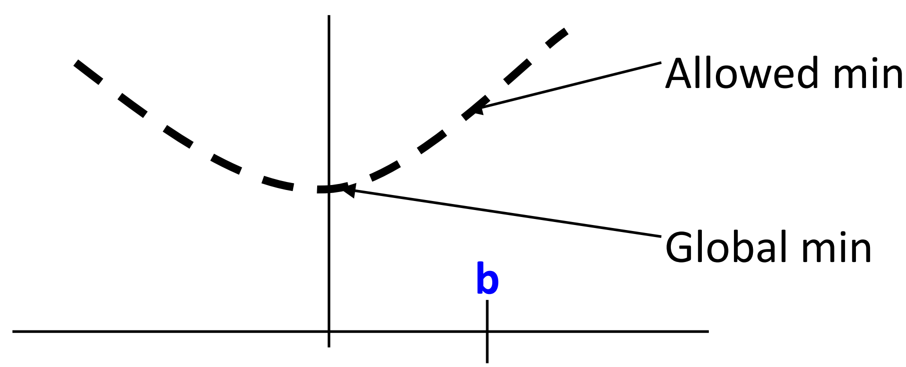
$\min_x x^2$

s.t. $x \geq b$

To

$\min_x \max_\alpha x^2 - \alpha(x-b)$

s.t. $\alpha \geq 0$



Lagrange multiplier for SVMs

Dual formulation

$$\min_{w,b} \max_{\alpha} \frac{\mathrm{w}^{\mathrm{T}}\mathrm{w}}{2} - \sum_i \alpha_i [(\mathrm{w}^{\mathrm{T}} x_i + b) y_i - 1]$$

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

Original formulation

Min $(w^T w)/2$

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

Using this new formulation we can derive w and b by taking the derivative w.r.t. w leading to:

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

$$\alpha_i \geq 0$$

Finally, taking the derivative w.r.t. b we get:

$$\sum_i \alpha_i y_i = 0$$

Dual SVM for linearly separable case

Substituting w into our target function and using the additional constraint we get:

$$\min_{w,b} \frac{\mathrm{w}^{\mathrm{T}}\mathrm{w}}{2} - \sum_i \alpha_i[(\mathrm{w}^{\mathrm{T}} x_i + b) y_i - 1]$$

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

Dual formulation

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

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

$$\alpha_i \geq 0$$

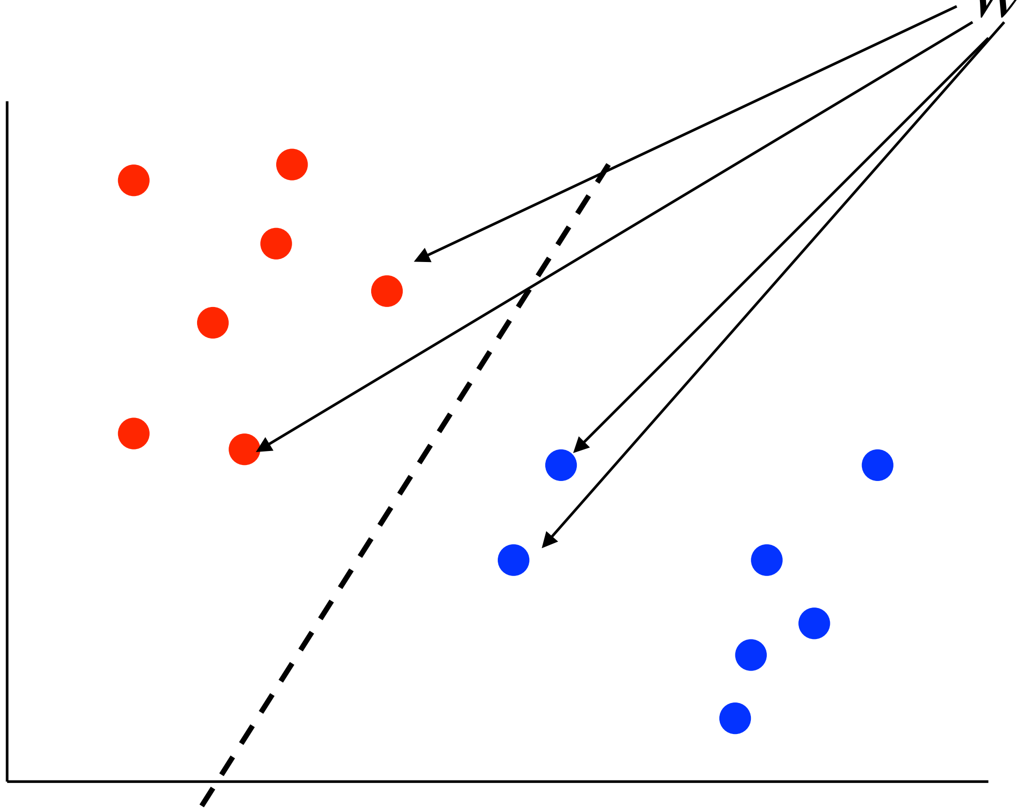
$$\sum_i \alpha_i \mathrm{y_i} = 0$$

$$\sum_i \alpha_i y_i = 0$$

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

Dual SVM - interpretation

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



For α's that are not 0

Dual SVM for linearly separable case

Our dual target function:

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

$$\sum_i \alpha_i \mathrm{y_i} = 0$$

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

Dot product for all training samples

Dot product with training samples

To evaluate a new sample x_j we need to compute:

$$\mathrm{w}^{\mathrm{T}} x_j + b = \sum_{\mathrm{i}} \alpha_i \mathrm{y_i} \mathbf{x_i}^{\mathbf{T}} \mathbf{x_j} + b$$

Is this too much computational work (for example when using transformation of the data)?

Important points

- Difference between regression classifiers and SVMs
- Maximum margin principle
- Target function for SVMs
- Linearly separable and non separable cases