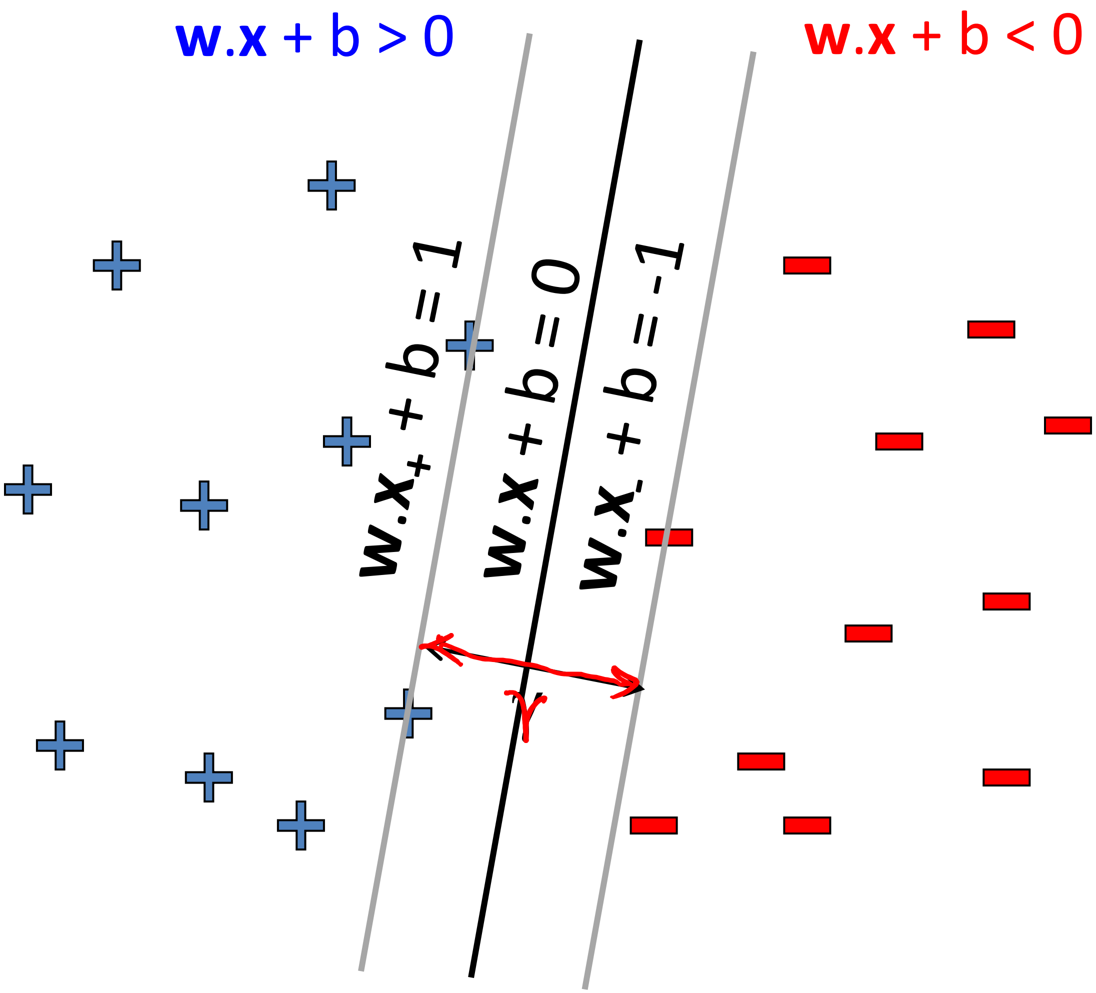
Support Vector Machines - Dual formulation

Aarti Singh

Machine Learning 10-701
Feb 6, 2023

Hard margin SVM

$\gamma \propto \frac{1}{\|w\|}$



$$\min_{\mathbf{w},b} \; \mathbf{w}.\mathbf{w} = \|w\|^2$$

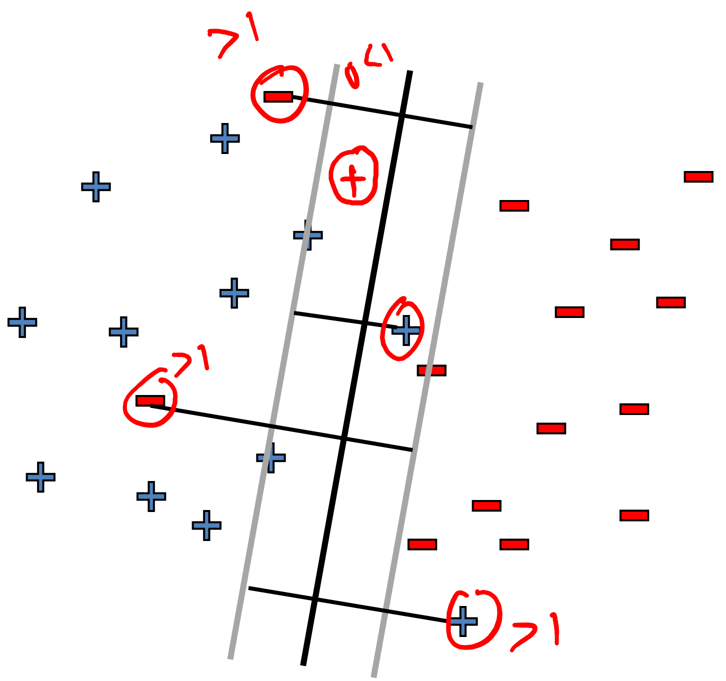
$$\text{s.t. } \underbrace{(\mathbf{w}.\mathbf{x}_j + b)\, y_j} \geq 1 \quad \forall j$$

Solve efficiently by quadratic programming (QP)

Assumes data is linearly separable

Soft margin SVM

Allow “error” in classification



$$\min_{\mathbf{w},b,\{\xi_j\}} \mathbf{w}.\mathbf{w} + C \sum_j \xi_j$$

$$\text{s.t. } (\mathbf{w}.\mathbf{x}_j + b)\, y_j \geq 1 - \xi_j \quad \forall j$$

$$\xi_j \geq 0 \quad \forall j$$

ξ_j - “slack” variables
= (>1 if x_j misclassifed)

pay linear penalty if mistake

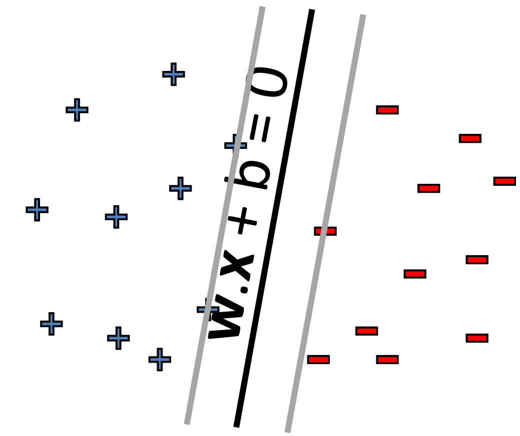
C - tradeoff parameter (C = ∞ recovers hard margin SVM)

Still QP ☺

SVM – linearly separable case

n training points $(\mathbf{x}_1, \ldots, \mathbf{x}_n)$

d features $\mathbf{x}_j$ is a d-dimensional vector



- Primal problem:

$$\text{minimize}_{\mathbf{w},b} \quad \frac{1}{2}\mathbf{w}.\mathbf{w}$$

$$\left(\mathbf{w}.\mathbf{x}_j + b\right) y_j \geq 1, \ \ \forall j$$

w - weights on features (d-dim problem)

- Convex quadratic program – quadratic objective, linear constraints
- But expensive to solve if d is very large
- Often solved in dual form (n-dim problem)

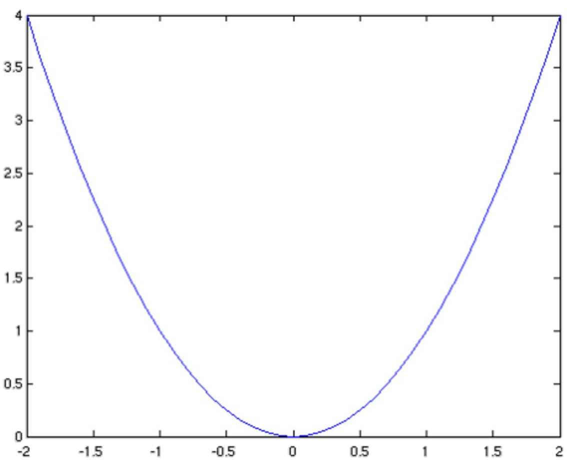
Detour - Constrained Optimization

$x \equiv (w, b)$

$$\min_x \; x^2 \quad \text{s.t.} \quad x \geq b$$

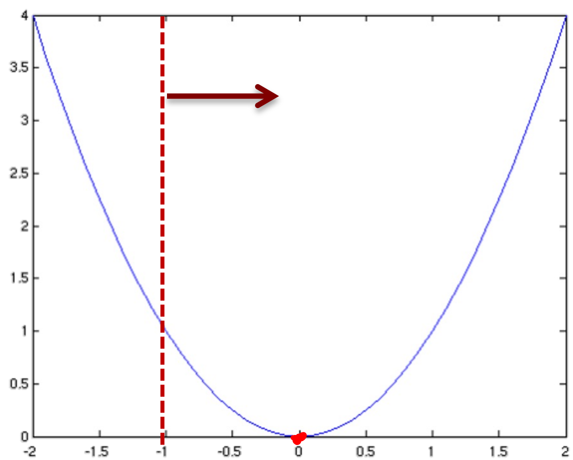
$$x^* = \max(b, 0)$$

$\min_x \; x^2$



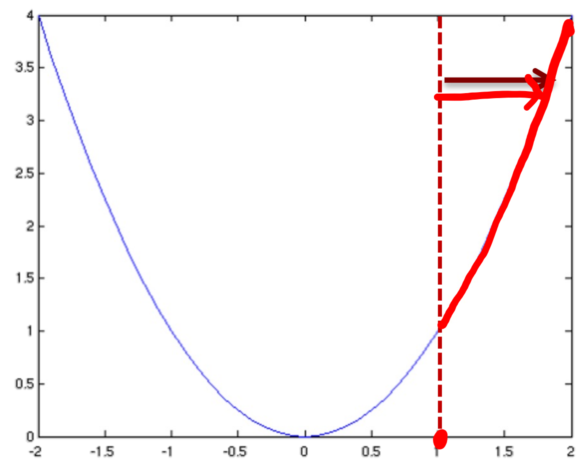
$x^* = 0$

$\min_x \; x^2$
s.t. $\; x \geq -1$



$x^* = 0$
Constraint inactive

$\min_x \; x^2$
s.t. $\; x \geq 1$

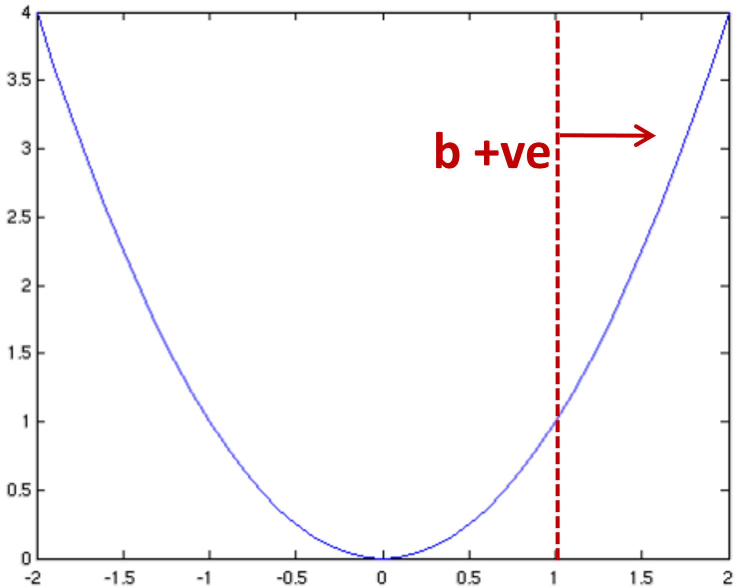


$x^* = 1$
Constraint active
(tight)

Constrained Optimization

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

$$\min_x \; x^2$$
$$\text{s.t.} \quad x \geq b$$



$$x^* = b$$

$I(A) = 1(x-b) = \begin{cases} \infty & x < b \\ 0 & x \geq b \end{cases}$

Equivalent unconstrained optimization:

$\min_x \; x^2 + I(x-b)$

Replace with lower bound ($\alpha >= 0$)

$x^2 + I(x-b) >= x^2 - \alpha(x-b)$ Lagrangian

Primal and Dual Problems

Notice that

Primal problem: p* = $\min_x \; x^2$ $\quad$ = $\displaystyle \min_x \max_{\alpha \geq 0} L(x, \alpha)$

s.t. $\; x \geq b$

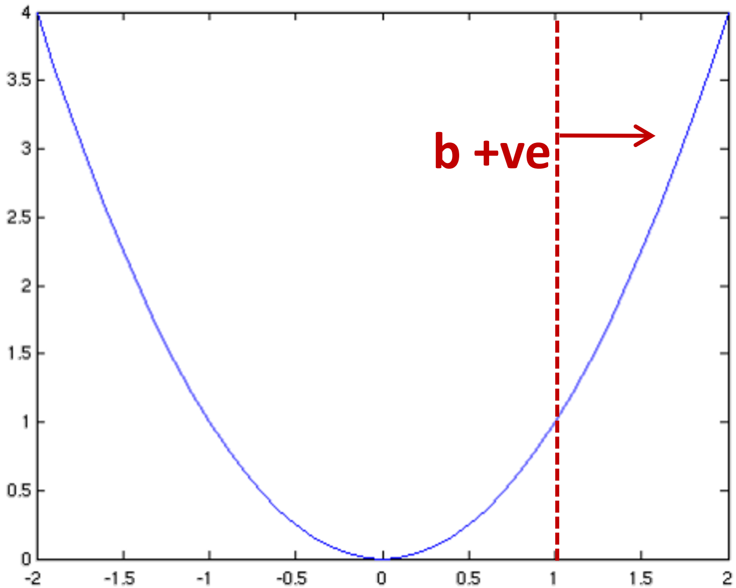
Why? $\quad L(x, \alpha) = x^2 - \alpha(x - b)$

$$\max_{\alpha \geq 0} L(x, \alpha) = x^2 - \min_{\alpha \geq 0} \alpha(x - b) = \begin{cases} +\infty & x < b \\ x^2 & x \geq b \end{cases}$$

Dual problem: d* = $\max_\alpha \; d(\alpha)$ $\quad$ = $\max_\alpha \underbrace{\min_x L(x, \alpha)}_{= d(\alpha)}$

s.t. $\; \alpha \geq 0$ $\quad\quad$ s.t. $\; \alpha \geq 0$

Recipe for deriving Dual Problem



$$x^* = b$$

Primal problem:

$$\min_x \; x^2$$
$$\text{s.t.} \quad x \geq b$$

$g(x) \geq 0$

$x - b \geq 0$

Moving the constraint to objective function

Lagrangian:

$$L(x, \alpha) = x^2 - \alpha(x - b)$$
$$\text{s.t.} \quad \alpha \geq 0$$

Dual problem:

$$\max_\alpha \; d(\alpha) \rightarrow \min_x L(x, \alpha)$$
$$\text{s.t.} \quad \alpha \geq 0$$

Why solve the Dual?

$d(\alpha) = \min_x L(x,\alpha)$

$f(x)$

Primal problem: p* = $\min_x \; x^2$
s.t. $x \geq b$

$= \min_x \max_{\alpha \geq 0} L(x, \alpha)$

Dual problem: d* = $\max_\alpha \; d(\alpha)$
s.t. $\alpha \geq 0$

$= \max_\alpha \min_x L(x, \alpha)$
s.t. $\alpha \geq 0$

- **Dual problem (maximization) is always concave even if primal is not convex**

  Why? Pointwise infimum of concave functions is concave.
  [Pointwise supremum of convex functions is convex.]

$$L(x, \alpha) = x^2 - \underbrace{\alpha(x - b)}_{f(x)}$$

- **As many dual variables α as constraints, helpful if fewer constraints than dimension of primal variable x**

Connection between Primal and Dual

Primal problem: p* = $\min_x \; x^2$
s.t. $x \geq b$

Dual problem: d* = $\max_\alpha \; d(\alpha)$
s.t. $\alpha \geq 0$

- **Weak duality:** The dual solution d* lower bounds the primal solution p* i.e. d* ≤ p*

To see this, recall $L(x, \alpha) = x^2 - \alpha(x - b)$

For every feasible x' (i.e. x' ≥ b) and feasible α' (i.e. α' ≥ 0) , notice that

$$d(\alpha) = \min_x L(x, \alpha) \leq x'^2 - \alpha'(x'-b) \leq x'^2$$

Since above holds true for every feasible x', we have $d(\alpha) \leq x^{*2} = p^*$

Connection between Primal and Dual

$x \equiv (w, b)$

Primal problem: p* = $\min_x \; x^2$ ✓
$\quad$ s.t. $\underbrace{x \geq b}_{g(x) \geq 0}$

x^*

Dual problem: d* = $\max_\alpha \; d(\alpha)$
$\quad$ s.t. $\alpha \geq 0$

α^*

- **Weak duality:** The dual solution d* lower bounds the primal solution p* i.e. d* ≤ p*

Duality gap = p*-d*

- **Strong duality:** d* = p* holds often for many problems of interest e.g. if the primal is a feasible convex objective with linear constraints (Slater's condition)

Connection between Primal and Dual

What does strong duality say about α^* (the α that achieved optimal value of dual) and x^* (the x that achieves optimal value of primal problem)?

KKT (Karush-Kuhn-Tucker conditions)

Whenever strong duality holds, the following conditions (known as KKT conditions) are true for α^* and x^*:

- 1. $\nabla L(x^*, \alpha^*) = 0$ i.e. Gradient of Lagrangian at x^* and α^* is zero. ✓
- 2. $x^* \geq b$ i.e. x^* is primal feasible ✓
- 3. $\alpha^* \geq 0$ i.e. α^* is dual feasible ✓
- 4. $\underbrace{\alpha^*(x^* - b) = 0}_{b=-1}$ (called as complementary slackness)

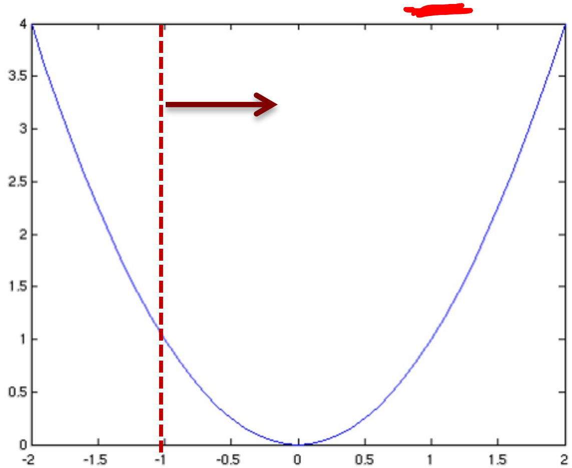
$\alpha^*(x^*+1) = 0 \Rightarrow$ either $\alpha^* = 0$ or $x^* = -1$

$\alpha^* (x'$

Constrained Optimization – Dual Problem

$\alpha^*(x^*-1) = 0$

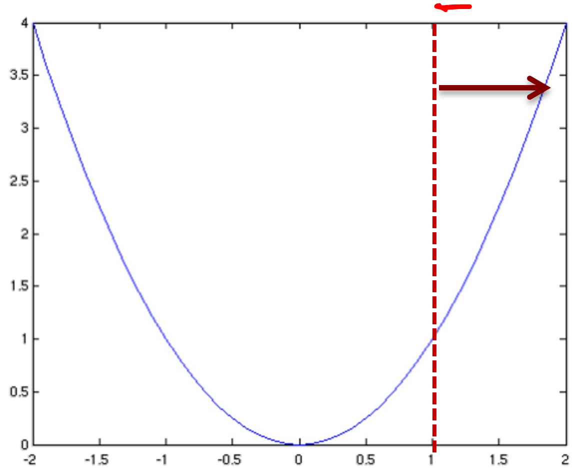
$$\min_x \; x^2$$
$$\text{s.t.} \quad x \geq -1$$



$$x^* = 0$$

α^* = 0
constraint is inactive
x^* > -1

$$\min_x \; x^2$$
$$\text{s.t.} \quad x \geq 1$$



$$x^* = 1$$

α^* > 0
constraint is active
x^* = 1

$$\min_x \; x^2$$
$$\text{s.t.} \quad x \geq -1$$
$g(x) \geq 0$
$$x \geq 1$$

$\alpha_1(x+1)$

$\alpha_2(x-1)$

- Poll:

Lagrangian

Dual variables

$x^2 - \alpha_1(x+1) - \alpha_2(x-1)$

$x^2 - \alpha g(x)$

Dual SVM – linearly separable case

n training points, d features $(\mathbf{x}_1, \ldots, \mathbf{x}_n)$ where x_i is a d-dimensional vector

- Primal problem:

$$\text{minimize}_{\mathbf{w},b} \quad \frac{1}{2}\mathbf{w}.\mathbf{w}$$
$$\left(\mathbf{w}.\mathbf{x}_j + b\right) y_j \geq 1, \quad \forall j$$

α_j

$\alpha_j \geq 0$

w - weights on features (d-dim problem)

- Dual problem (derivation):

$$L(\mathbf{w}, b, \alpha) = \frac{1}{2}\mathbf{w}.\mathbf{w} - \sum_j \alpha_j \left[\left(\mathbf{w}.\mathbf{x}_j + b\right) y_j - 1\right]$$
$$\alpha_j \geq 0, \quad \forall j$$

α - weights on training pts (n-dim problem)

Dual SVM – linearly separable case

$\alpha = (\alpha_1 .. \alpha_n)$

- Dual problem (derivation):

$$\max_\alpha \underbrace{\min_{\mathbf{w},b} L(\mathbf{w},b,\alpha)}_{d(\alpha)} = \frac{1}{2}\mathbf{w}.\mathbf{w} - \sum_j \alpha_j \left[\left(\mathbf{w}.\mathbf{x}_j + b\right) y_j - 1\right]$$

$\alpha_j \geq 0, \ \forall j$

$$\frac{\partial L}{\partial w} = w - \sum_j \alpha_j x_j y_j = 0$$

$$\frac{\partial L}{\partial \mathbf{w}} = 0 \qquad \Rightarrow \mathbf{w} = \sum_j \alpha_j y_j \mathbf{x}_j$$

$$\frac{\partial L}{\partial b} = 0 \qquad \Rightarrow \sum_j \alpha_j y_j = 0$$

$$\frac{\partial L}{\partial b} = -\sum_j \alpha_j y_j = 0.$$

Dual SVM – linearly separable case

- Dual problem:

$$\max_\alpha \min_{\mathbf{w},b} L(\mathbf{w},b,\alpha) = \frac{1}{2}\mathbf{w}.\mathbf{w} - \sum_j \alpha_j \left[\left(\mathbf{w}.\mathbf{x}_j + b\right) y_j - 1\right]$$

$\alpha_j \geq 0, \ \forall j$

$$\Rightarrow \underline{\mathbf{w} = \sum_j \alpha_j y_j \mathbf{x}_j} \qquad \Rightarrow \sum_j \underline{\alpha_j y_j = 0}$$

$$\min_{w,b} L(w,b,\alpha) = \frac{1}{2} \sum_j \alpha_j y_j x_j \cdot \sum_i \alpha_i y_i x_i - \sum_j \alpha_j \left[\left(\sum_i \alpha_i y_i x_i \cdot x_j + b\right) y_j - 1\right]$$

$$= \frac{1}{2} \sum_{ij} \alpha_i \alpha_j y_i y_j x_i \cdot x_j - \sum_j \sum_i \alpha_j \alpha_i y_i y_j x_i \cdot x_j + \sum_j \alpha_j$$

$$= -\frac{1}{2} \sum_{ij} \alpha_i \alpha_j y_i y_j x_i \cdot x_j + \sum_j \alpha_j = d(\alpha)$$

Dual SVM – linearly separable case

$d/d\alpha)$

$$\text{maximize}_\alpha \quad \sum_i \alpha_i - \frac{1}{2}\sum_{i,j} \alpha_i\alpha_j y_i y_j \mathbf{x}_i . \mathbf{x}_j$$

$$\sum_i \alpha_i y_i = 0$$

$$\alpha_i \geq 0$$

$sign(w \cdot \underline{x} + b)$

Dual problem is also QP

Solution gives α_js

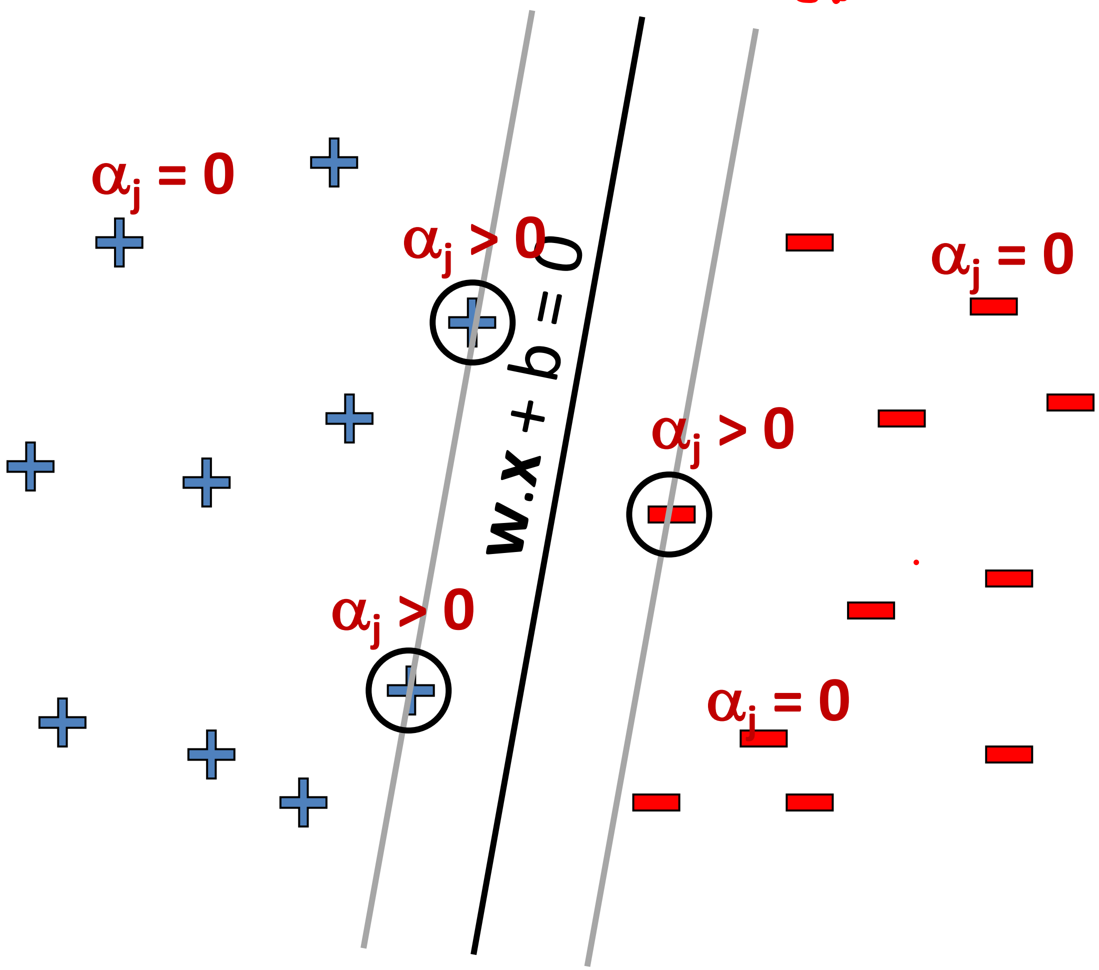
$$\mathbf{w} = \sum_i \alpha_i y_i \mathbf{x}_i$$

What about b?

Dual SVM: Sparsity of dual solution

complementary slackness

$\alpha_j \left((\mathbf{w} \cdot x_j + b) y_j - 1 \right) = 0$



$$\mathbf{w} = \sum_j \alpha_j y_j \mathbf{x}_j$$

Only few α_js can be non-zero : where constraint is active

$$(\mathbf{w}.\mathbf{x}_j + b)y_j = 1$$

Support vectors – training points j whose α_js are non-zero

Dual SVM – linearly separable case

$$\text{maximize}_\alpha \quad \sum_i \alpha_i - \frac{1}{2}\sum_{i,j} \alpha_i \alpha_j y_i y_j \mathbf{x}_i . \mathbf{x}_j$$

$$\sum_i \alpha_i y_i = 0$$

$$\alpha_i \geq 0$$

Dual problem is also QP

Solution gives α_js

$$\boxed{\begin{array}{l} \mathbf{w} = \sum_i \alpha_i y_i \mathbf{x}_i \\ b = y_k - \mathbf{w}.\mathbf{x}_k \\ \text{for any } k \text{ where } \alpha_k > 0 \end{array}}$$

Use any one of support vectors with α_k>0 to compute b since constraint is tight $(w.x_k + b)y_k = 1$

$\rightarrow \quad b = \frac{1}{y_k} - w.x_k$

Dual SVM – non-separable case

- Primal problem:

$$\text{minimize}_{\mathbf{w},b,\{\xi_j\}} \frac{1}{2}\mathbf{w}.\mathbf{w} + C\sum_j \xi_j$$

$$\left(\mathbf{w}.\mathbf{x}_j + b\right) y_j \geq 1 - \xi_j, \quad \forall j$$

$$\xi_j \geq 0, \quad \forall j$$

α_j

μ_j

$\boxed{\begin{matrix}\alpha_j \\ \mu_j\end{matrix}}$

Lagrange Multipliers

- Dual problem:

$$\max_{\alpha,\mu} \underbrace{\min_{\mathbf{w},b,\{\xi_j\}} L(\mathbf{w}, b, \xi, \alpha, \mu)}_{d(\alpha,\mu)}$$

$$s.t. \alpha_j \geq 0 \quad \forall j$$

$$\mu_j \geq 0 \quad \forall j$$

Dual SVM – non-separable case

maximize$_\alpha$ $\sum_i \alpha_i - \frac{1}{2}\sum_{i,j} \alpha_i\alpha_j y_i y_j \mathbf{x}_i.\mathbf{x}_j$

$\sum_i \alpha_i y_i = 0$

$\boxed{C \geq} \alpha_i \geq 0$

$C > \alpha_k > 0$

comes from $\frac{\partial L}{\partial \xi} = 0$

Intuition:

If C→∞, recover hard-margin SVM

Dual problem is also QP

Solution gives α_js

$$\mathbf{w} = \sum_i \alpha_i y_i \mathbf{x}_i$$

$$b = y_k - \mathbf{w}.\mathbf{x}_k$$

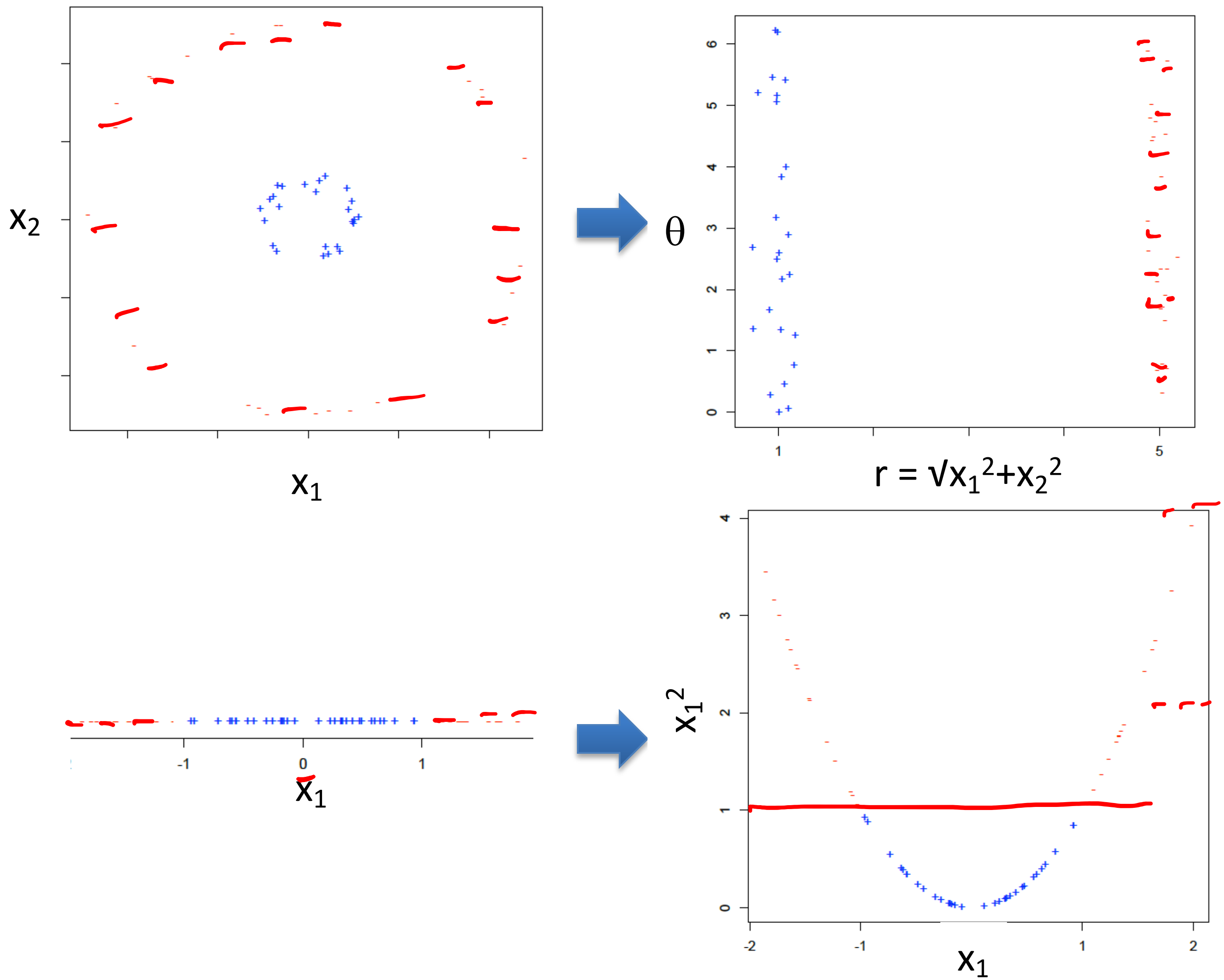
for any k where $C > \alpha_k > 0$

So why solve the dual SVM?

- There are some quadratic programming algorithms that can solve the dual faster than the primal, (specially in high dimensions d>>n)

- But, more importantly, the "**kernel trick**"!!!

Separable using higher-order features



Dual formulation only depends on dot-products, not on w!

$$\text{maximize}_\alpha \quad \sum_i \alpha_i - \frac{1}{2}\sum_{i,j}\alpha_i\alpha_j y_i y_j \underbrace{\mathbf{x}_i.\mathbf{x}_j}$$

$$\sum_i \alpha_i y_i = 0$$

$$C \geq \alpha_i \geq 0$$

$\overbrace{\phi(x_i)\cdot\phi(x_j)}$

$\phi(x_i) = \begin{bmatrix} x_i \\ x_i^2 \\ \log x_i \\ e^{x_i} \end{bmatrix}$

$$\text{maximize}_\alpha \quad \sum_i \alpha_i - \frac{1}{2}\sum_{i,j}\alpha_i\alpha_j y_i y_j \underbrace{K(\mathbf{x}_i, \mathbf{x}_j)}$$

$$K(\mathbf{x}_i, \mathbf{x}_j) = \Phi(\mathbf{x}_i)\cdot\Phi(\mathbf{x}_j)$$

$$\sum_i \alpha_i y_i = 0$$

$$C \geq \alpha_i \geq 0$$

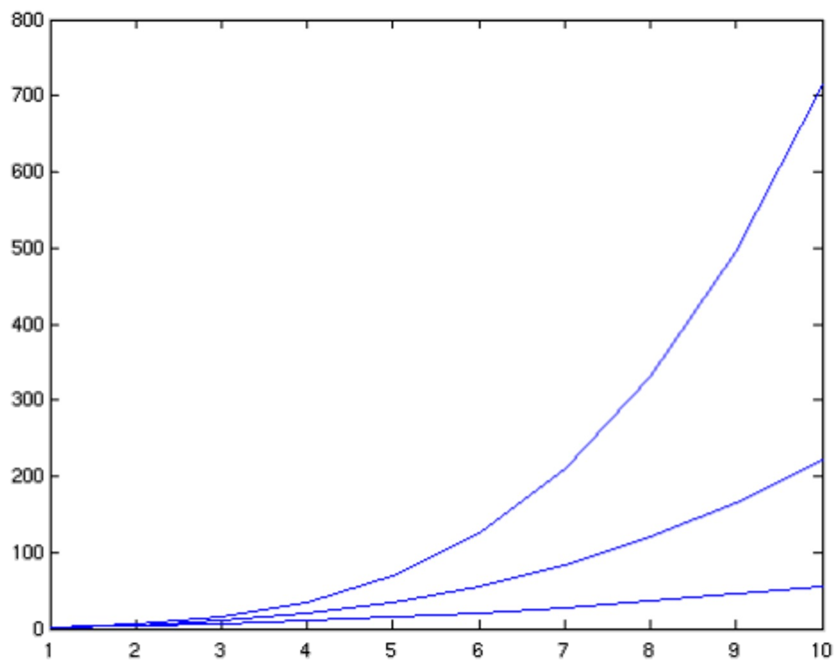
$\Phi(\mathbf{x})$ – High-dimensional feature space, but never need it explicitly as long as we can compute the dot product fast using some Kernel K

Polynomial features $\phi(x)$

$x \qquad \phi(x) = \begin{bmatrix} x \\ x^2 \\ x^3 \\ \vdots \\ x^d \end{bmatrix}$

m – input features   d – degree of polynomial

$$\text{num. terms} = \binom{d+m-1}{d} = \frac{(d+m-1)!}{d!(m-1)!} \sim m^d$$



grows fast!
d = 6, m = 100
about 1.6 billion terms

$x = \begin{bmatrix} x^{(1)} \\ x^{(2)} \\ \vdots \\ x^{(m)} \end{bmatrix} \qquad \phi(x) = \begin{bmatrix} x^{(1)3} \\ x^{(2)3} \\ \vdots \\ x^{(1)2} x^{(2)} \\ \vdots \\ x^{(1)} x^{(2)} x^{(3)} \\ \vdots \end{bmatrix} \quad d=3$

Dot Product of Polynomial features

$$\Phi(\mathbf{x}) = \text{polynomials of degree exactly d}$$

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \quad \mathbf{z} = \begin{bmatrix} z_1 \\ z_2 \end{bmatrix}$$

d=1 $\Phi(\mathbf{x}) \cdot \Phi(\mathbf{z}) = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \cdot \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = x_1 z_1 + x_2 z_2 = \mathbf{x} \cdot \mathbf{z}$

d=2 $\Phi(\mathbf{x}) \cdot \Phi(\mathbf{z}) = \begin{bmatrix} x_1^2 \\ \sqrt{2} x_1 x_2 \\ x_2^2 \end{bmatrix} \cdot \begin{bmatrix} z_1^2 \\ \sqrt{2} z_1 z_2 \\ z_2^2 \end{bmatrix}$

$$\begin{aligned} &= x_1^2 z_1^2 + x_2^2 z_2^2 + 2 x_1 x_2 z_1 z_2 \\ &= (x_1 z_1 + x_2 z_2)^2 \\ &= (\mathbf{x} \cdot \mathbf{z})^2 \end{aligned}$$

d=6 $(x \cdot z)^6$

d $\Phi(\mathbf{x}) \cdot \Phi(\mathbf{z}) = K(\mathbf{x}, \mathbf{z}) = (\mathbf{x} \cdot \mathbf{z})^d$