Support Vector Machines - Dual formulation and Kernel Trick

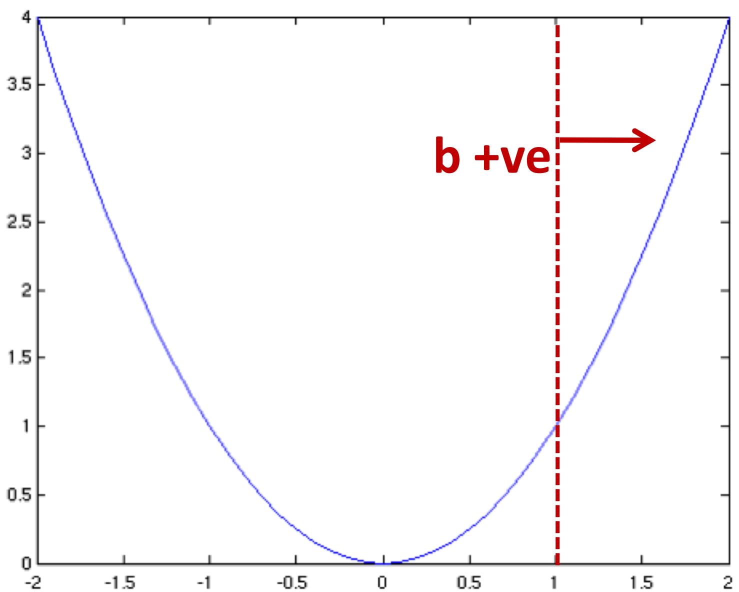
Aarti Singh

Machine Learning 10-315
Oct 28, 2020

MACHINE LEARNING DEPARTMENT
Carnegie Mellon.
School of Computer Science

Constrained Optimization – Dual Problem

$w \in \mathbb{R}^d \longrightarrow \mathbb{R}^n \ni \vec{\alpha}$



$$x^* = b$$

Primal problem:

$\min_w \|w\|^2$

s.t. $(w \cdot x_i + b) y_i \geq 1$

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

$\alpha_i \geq 0$

$x - b \geq 0$

Moving the constraint to objective function

Lagrangian:

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

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

$\geq L(x, \alpha)$

$\min_x \max_\alpha L(x, \alpha)$

Dual problem:

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

Connection between Primal and Dual

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

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

- **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)

$(w^*, b^*) \equiv x^*$

α^*

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)?

Karush-Kuhn-Tucker

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

- 1. $\bigtriangledown L(x^*, \alpha^*) = 0$ 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. $\alpha^*(x^* - b) = 0$ (called as complementary slackness)

$\alpha^* = 0$ or $x^* = b$

$x^* > b \Downarrow \alpha^* = 0$

$\alpha^* > 0 \Rightarrow x^* = b$

α_i $(w \cdot x_i + b) y_i \geq 1$

We use the first one to relate x^* and α^*. We use the last one (complimentary slackness) to argue that $\alpha^* = 0$ if constraint is inactive and $\alpha^* > 0$ if constraint is active and tight.

Solving the dual

Solving:

Dual:

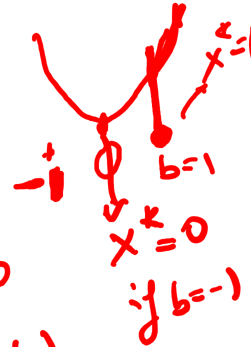
$$\max_\alpha \left(\min_x \underbrace{x^2 - \alpha(x-b)}_{L(x,\alpha)} \right)$$

s.t. $\alpha \geq 0$

$\min_x x^2$

s.t. $\{x \geq b \leftarrow \alpha \geq 0$

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

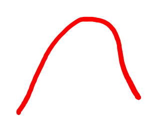


Dual derivation:

$$\frac{\partial L(x,\alpha)}{\partial x} = 2x - \alpha = 0 \Rightarrow x = \alpha/2$$

$$\min_x L(x,\alpha) = \left(\frac{\alpha}{2}\right)^2 - \alpha\left(\frac{\alpha}{2} - b\right) = \frac{\alpha^2}{4} - \frac{\alpha^2}{2} + \alpha b = -\frac{\alpha^2}{4} + \alpha b$$

Dual: $\max_{\alpha \geq 0} \underline{-\frac{\alpha^2}{4} + \alpha b}$



$$\frac{\partial}{\partial \alpha} = -\frac{\alpha}{2} + b = 0$$

$\Rightarrow \alpha^* = \cancel{2b}$ $\max(0, 2b)$

$\alpha^* = 0$ if $b = -1$

$= 2$ if $b = 1$

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

$\alpha^* \neq 0 \Rightarrow x^* = b$

$\alpha^* = 0$

$\alpha^* = 2 \neq 0 \Rightarrow x^* = 1$

Solving the dual

Solving:

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

<u>Find the dual:</u> Optimization over x is unconstrained.

$$\frac{\partial L}{\partial x} = 2x - \alpha = 0 \Rightarrow x^* = \frac{\alpha}{2}$$

$$L(x^*, \alpha) = \frac{\alpha^2}{4} - \alpha\left(\frac{\alpha}{2} - b\right) = -\frac{\alpha^2}{4} + b\alpha$$

<u>Solve</u>: Now need to maximize $L(x^*,\alpha)$ over $\alpha \geq 0$

Solve unconstrained problem to get α' and then take $\max(\alpha',0)$

$$\frac{\partial}{\partial \alpha} L(x^*, \alpha) = -\frac{\alpha}{2} + b \quad \Rightarrow \alpha' = 2b$$

$$\Rightarrow \alpha^* = \max(2b, 0) \qquad \Rightarrow x^* = \frac{\alpha^*}{2} = \max(b, 0)$$

α = 0 constraint is inactive, <u>α > 0</u> <u>constraint is active (tight)</u>

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:

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

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

$(w.x_j+b)y_j - 1 \geq 0$ $\qquad \alpha_j \geq 0$

w – weights on features (d-dim problem)

- Dual problem (derivation):

$\frac{1}{2} w.w - \sum_j \alpha_j ((w.x_j+b)y_j - 1) = L(w,b,\alpha)$

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

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

Dual SVM – linearly separable case

- Dual problem (derivation):

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

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

$w - \sum \alpha_j x_j y_j = 0$

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

$- \sum \alpha_j y_j = 0$

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

If we can solve for αs (dual problem), then we have a solution for **w**,b (primal problem)

Dual SVM – linearly separable case

$\vec{\alpha} = \begin{pmatrix} \alpha_1 \\ \vdots \\ \alpha_d \end{pmatrix}$

- Dual problem:

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

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

$$\Rightarrow \underset{\in \mathbb{R}^d_{d\times 1}}{\mathbf{w}} = \sum_j \alpha_j y_j \underset{= d\times 1}{\mathbf{x}_j} \in \mathbb{R}^d \qquad \Rightarrow \sum_j \alpha_j y_j = 0$$

$$d(\alpha) = \frac{1}{2} \underbrace{\sum_j \alpha_j y_j \vec{x}_j}_{} \cdot \underbrace{\sum_i \alpha_i y_i \vec{x}_i}_{} - \sum_j \alpha_j \underbrace{\left(\sum_i \alpha_i y_i \vec{x}_i\right)}_{} \cdot \vec{x}_j y_j - b \cancel{\sum_j \alpha_j y_j}^{0} + \sum_j \alpha_j$$

$$= -\frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j \underbrace{\vec{x}_i \cdot \vec{x}_j}_{} + \sum_j \alpha_j$$

Dual SVM – linearly separable case

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

$$\sum_i \alpha_i y_i = 0$$

$$\alpha_i \geq 0$$

$\{\vec{x}_i, y_i\}_{i=1}^{n}$

Dual problem is also QP

Solution gives α_js

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

What about b?

$b^* = \frac{1 - w^* y_i x_i}{y_i} = \begin{cases} 1 - w^* x_i & y_i = 1 \\ -1 - w^* x_i & y_i = -1 \end{cases}$

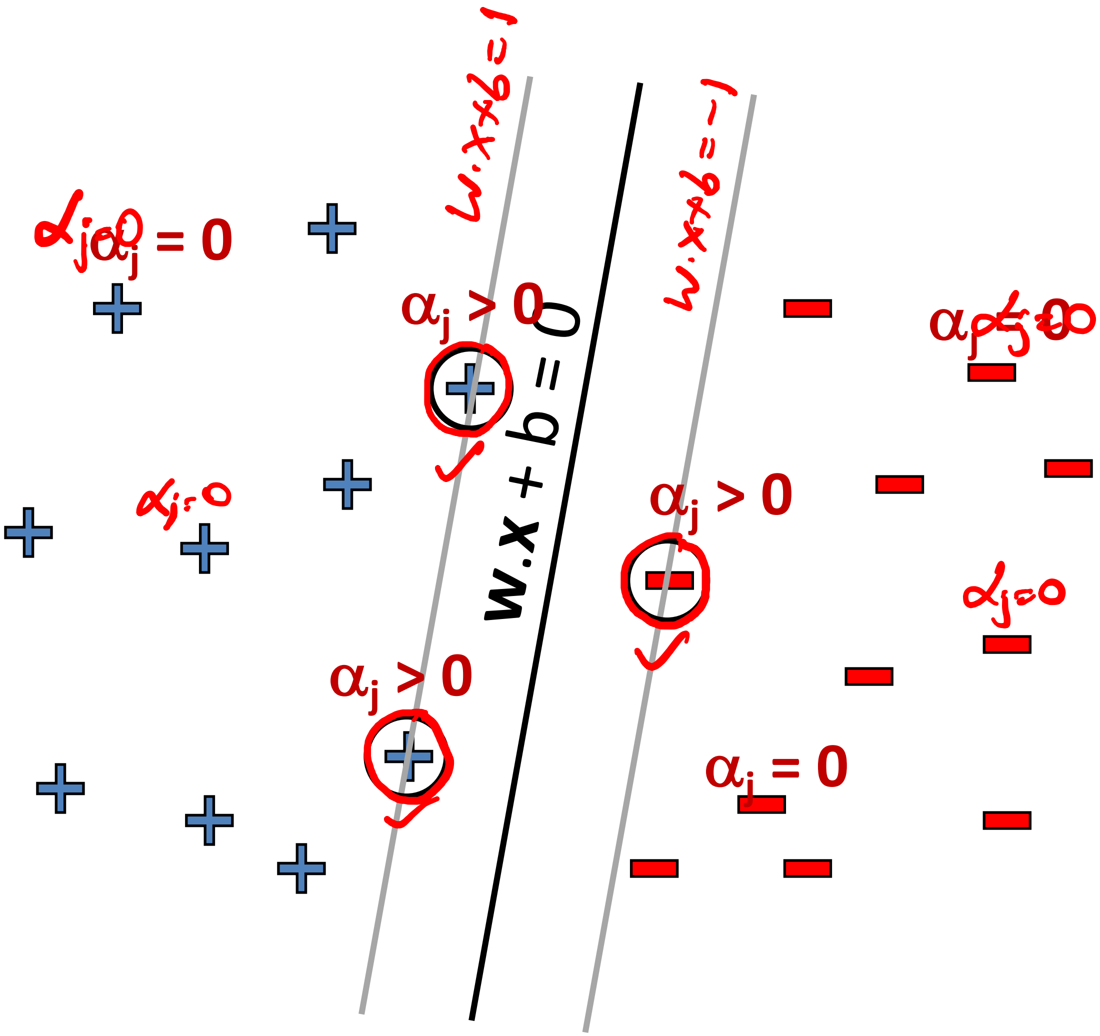
$\alpha_i^* \left(\underbrace{(w^* \cdot x_i + b^*) y_i - 1}_{} \right) = 0$ Comp slackness

Toy eg: $\alpha^* (x^* - b) = 0 \quad x \geq b$

$w^* x_i y_i + b^* y_i = 1$

$\frac{1}{y_i} - w^* \cdot x_i$

Dual SVM: Sparsity of dual solution



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

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

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

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

Dual SVM – linearly separable case

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 $\longrightarrow$

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

Use any one of support vectors with α_k>0 to compute b since constraint is tight $(\mathbf{w}.\mathbf{x}_k + b)y_k = 1$

Dual SVM – non-separable case

- Primal problem:

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

$$\left(\mathbf{w}.\mathbf{x}_j + b\right) y_j \geq 1 - \xi_j, \quad \forall j \qquad n \quad \boxed{\alpha_j} \geq 0$$

$$\xi_j \geq 0, \quad \forall j \qquad n \quad \boxed{\mu_j} \geq 0$$

w - d×1
b - 1×1
{ξ_j - 1×1}_n

Lagrange Multipliers

- Dual problem:

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

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

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

HW3!

Dual SVM – non-separable case

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

$\sum_i \alpha_i y_i = 0$ —

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

$\alpha_i \geq 0$

$-\alpha_i \geq -C$

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

Intuition:

If C→∞, recover hard-margin SVM

Dual problem is also QP

Solution gives α_js

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

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

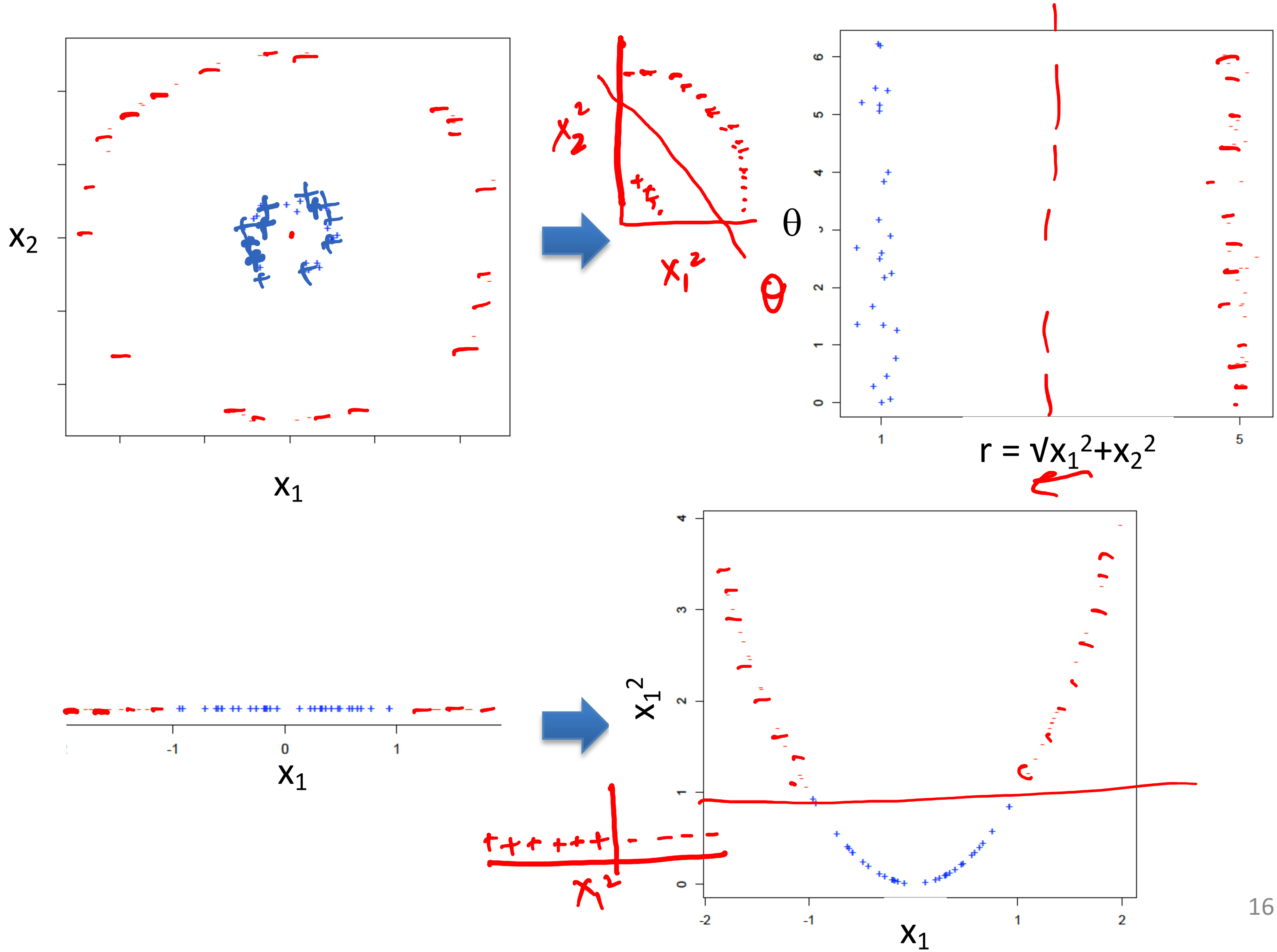
for any k where $C > \alpha_k > 0$

So why solve the dual SVM?

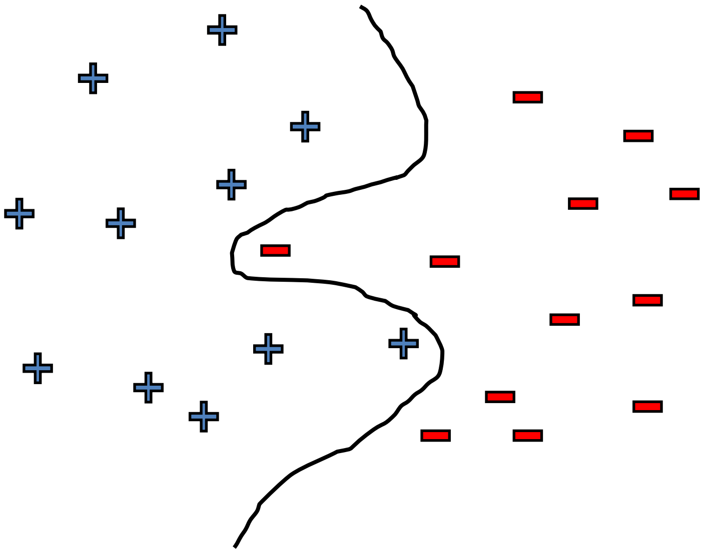
- There are some quadratic programming algorithms that can solve the dual faster than the primal, (specially in high dimensions d>>n)

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

Separable using higher-order features



What if data is not linearly separable?



Use features of features of features of features....

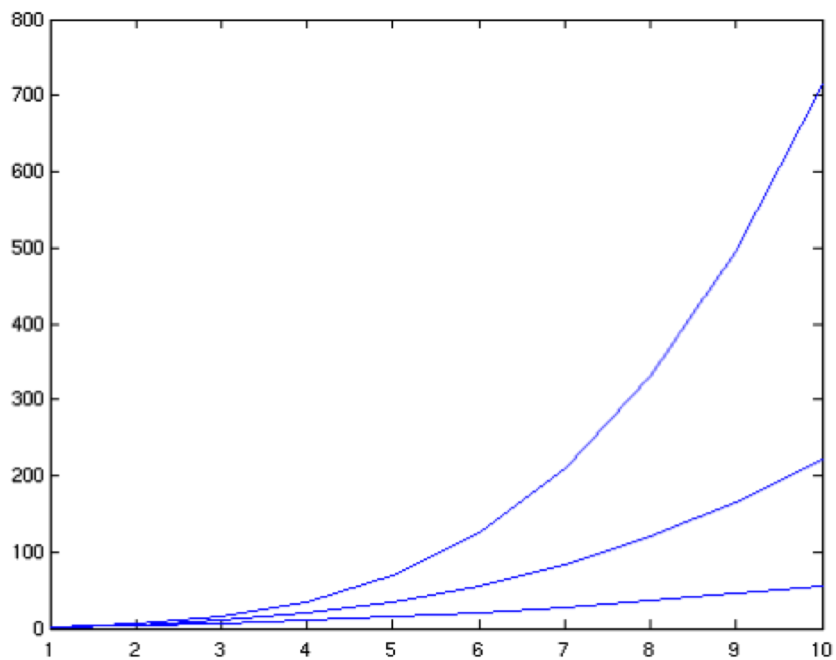
$$\Phi(\mathbf{x}) = (x_1^2, x_2^2, x_1 x_2, ..., \exp(x_1))$$

Feature space becomes really large very quickly!

Higher Order Polynomials

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

$m=3$ $d=4$

$x_1^{\alpha_1}$ $x_2^{\alpha_2}$ $x_3^{\alpha_3}$ $\alpha_1+\alpha_2+\alpha_3=4$

$x_1^4 + x_2^4 + x_1 x_2^3 + x_1 x_2 x_3^2$

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

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

$$\sum_i \alpha_i y_i = 0$$

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

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

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

$\phi(x_i) = D \times 1$

$D \gg d$

$$\sum_i \alpha_i y_i = 0$$

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

Φ(**x**) – High-dimensional feature space, but never need it explicitly as long as we can compute the dot product fast using some Kernel K

Dot Product of Polynomials

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

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

$$\phi(x) = \begin{pmatrix} x_1^d \\ x_1^{d-1} x_2 \\ x_1^{d-2} x_2^2 \\ \vdots \end{pmatrix}$$

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

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

4 × 3 $\qquad$ 4 × 3

$$= (x_1 z_1 + x_2 z_2)^2$$

$K(x,z) \quad = (\mathbf{x} \cdot \mathbf{z})^2 \leftarrow$ 4 operations

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

Finally: The Kernel Trick!

$(x_i \cdot x_j)^d \leftarrow \text{deg}$

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

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

$$\sum_i \alpha_i y_i = 0$$

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

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

- Never represent features explicitly
  - Compute dot products in closed form
- Constant-time high-dimensional dot-products for many classes of features

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

$w \cdot \phi(x) = \sum_i \alpha_i y_i \underline{\phi(x_i) \cdot \phi(x)} \rightarrow K(x_i, x)$

Common Kernels

- Polynomials of degree d

$$K(\mathbf{u},\mathbf{v}) = (\mathbf{u}\cdot\mathbf{v})^d$$

- Polynomials of degree up to d

$d=2$

$x_1 \quad x_2 \quad x_1^2 \quad x_2^2$

$x_1 x_2$

$$K(\mathbf{u},\mathbf{v}) = (\mathbf{u}\cdot\mathbf{v}+1)^d$$

- Gaussian/Radial kernels (polynomials of all orders – recall series expansion of exp)

$$K(\mathbf{u},\mathbf{v}) = \exp\left(-\frac{||\mathbf{u}-\mathbf{v}||^2}{2\sigma^2}\right)$$

- Sigmoid

$$K(\mathbf{u},\mathbf{v}) = \tanh(\eta\mathbf{u}\cdot\mathbf{v}+\nu)$$