03/30/15

SVM DUALITY + KERNELS

(1) Hard-margin SVM dual

General Lagrangian Duality			OP SVM (special case)	
$\min_{\vec{w}} f(\vec{w})$			$\min_{\vec{w},b} \frac{1}{2} w^T w$	
s.t $g(w) \leq 0$	$: \alpha$	α_i	$y_i(\vec{w}^T\vec{x}_i + b) \geq 1$	$\forall i$
$h(w) = 0$	$: \beta$	N/A		

Lagrangian Multiplier / Dual Variables

$$\rightarrow L(\vec{w}, \alpha, \beta) = f(\vec{w}) + \alpha g(\vec{w}) + \beta h(\vec{w}) = \frac{1}{2} w^T w + \sum_{i=1}^{N} \alpha_i \left[1 - y_i (w^T x_i + b)\right]$$

→ KKT Condition #1

$$\frac{\partial L}{\partial w}(w^*, \alpha^*, \beta^*) = 0 \qquad \frac{\partial L}{\partial \vec{w}} = w^T + \sum_{i=1}^{N} \alpha_i \left[-y_i \vec{x}_i^T\right] = 0$$

$$\Rightarrow \boxed{\vec{w}^* = \sum \alpha_i^* y_i \vec{x}_i}$$

w is a linear combination of data!

Also $\frac{\partial L}{\partial b} = \sum_{i=1}^{N} \alpha_i \left[-y_i\right] = 0 \Rightarrow \boxed{\sum_{i=1}^{N} \alpha_i^* y_i = 0}$

constraint on $\vec{\alpha}$

→ Dual Function

$$b_D(\alpha, \beta) = \min_{\vec{w}} f(w) + \alpha g(w) + \beta h(w) \qquad b_D(\alpha_1, \ldots, \alpha_N) = \frac{1}{2} \vec{w}^{*T} \vec{w}^* + \sum_{i=1}^{N} \alpha_i \left[1 - y_i (\vec{w}^{*T} \vec{x}_i + b^*)\right]$$

$$L_D(\vec{\alpha}) = \frac{1}{2}\left[\sum_{i=1}^{N} \alpha_i y_i x_i\right]^T \left[\sum_{j=1}^{N} \alpha_j y_j x_j\right] + \sum_{i=1}^{N} \alpha_i \left[1 - y_i \left\{\left(\sum_{j=1}^{N} \alpha_j y_j x_j\right)^T x_i + b^*\right\}\right]$$

$$= \frac{1}{2}\left[\sum_i \alpha_i y_i x_i\right]^T \left[\sum_j \alpha_j y_j x_j\right] + \sum_i \alpha_i - \left[\sum_i \alpha_i y_i x_i\right]^T \left[\sum_j \alpha_j y_j x_j\right] - \underbrace{\sum_i \alpha_i y_i}_{=0} b^*$$

$$= \sum_{i=1}^{N} \alpha_i - \frac{1}{2}\left[\sum_{i=1}^{N} \alpha_i y_i x_i\right]^T \left[\sum_{j=1}^{N} \alpha_j y_j x_j\right]$$

$$= \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j \vec{x}_i^T \vec{x}_j$$

So SVM Dual Program

$$\max_{\alpha_1, \ldots, \alpha_N} \quad \vec{\alpha}^T \vec{1} - \frac{1}{2} \vec{\alpha}^T H \vec{\alpha}$$

$$\text{s.t} \quad \sum_{i=1}^{N} \alpha_i y_i = 0$$

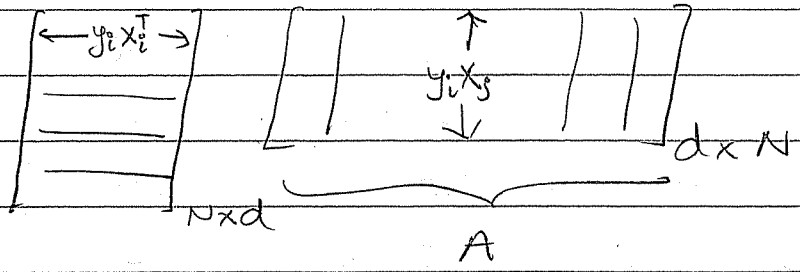
$$\alpha_i \geq 0$$

QP
→ N vars
→ N+1 constraints
(N easy positivity, 1 only non-trivial constraint)

where $H_{ij} = y_i y_j \vec{x}_i^T \vec{x}_j$

Is this a convex QP?
⇔ is H PSD? Yes!

Why?

$H =$ 

$$H = A^T A$$

$$\Rightarrow \quad \alpha^T H \alpha = \alpha^T A^T A \alpha = \|A\alpha\|_2^2 \geq 0$$

$$\Rightarrow \quad H \succeq 0$$

$\Rightarrow$ Dual QP objective is concave
$\Rightarrow$ QP is a convex OPT problem
$\Rightarrow$ can solve easily

But Primal QP was also convex.
So why do we care about the dual?

Answers:

① Exposes structure about the problem!

KKT #1 $\Rightarrow \quad \vec{w} = \sum \alpha_i y_i \vec{x}_i$

KKT #2 (complementary slackness) $\alpha_i^* g_i(w^*) = 0$

$$\Rightarrow \alpha_i^* \left[1 - y_i (\vec{w}^* x_i + b^*)\right] = 0$$

Thus, $\alpha_i^* > 0 \quad \Rightarrow \quad y_i(w^{*T} x_i + b) \underbrace{= 1}_{\text{Not just } \geq}$

AHA! So $\alpha_i^* > 0 \Rightarrow i$ lies on the Margin!

So $\vec{w} = \sum_{i=1}^{N} \alpha_i y_i \vec{x}_i$

$= \sum_{i \in \text{support vectors}} \alpha_i y_i \vec{x}_i$

Mathematically precise way of saying "other data points don't matter!"

Isn't that beautiful?

Answer (2)

- Dual OP has only 1 non-trivial constraint.

Unconstrained or "nearly unconstrained" QPs are sometimes easier to solve.

e.g Platt's SMO (Sequential Minimal OPt) algorithm solves SVM dual OP easily.

Answer (3) The "Kernel Trick"

Observation:
In the dual OP, data $\{\vec{x}_i\}$ only appears in pairs as $\vec{x}_i^T \vec{x}_j$

Idea: So if we increase dimensionality by feature augmentation

$\vec{x}_i \to \phi(\vec{x}_i)$

$\phi : \mathbb{R}^d \to \mathbb{R}^D \qquad D >> d$

all we need to do is replace $\vec{x}_i^T \vec{x}_j$ by $\phi(\vec{x}_i)^T \phi(\vec{x}_j)$

What is a Kernel?

Intuitively, a kernel $K: \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ is a function $K(\vec{x}_i, \vec{x}_j)$ that efficiently computes $\phi(\vec{x}_i)^T \phi(\vec{x}_j)$ for some ϕ

Example: Polynomial Kernel of degree 2

$\vec{u} \in \mathbb{R}^d \quad \vec{v} \in \mathbb{R}^d$

$$K(\vec{u}, \vec{v}) = (\vec{u}^T \vec{v})^2$$

$$= \left(\sum_{i=1}^{d} u_i v_i\right)^2$$

$$= (u_1 v_1 + u_2 v_2 + \cdots u_d v_d)^2$$

$$= (u_1 v_1)^2 + \cdots + (u_d v_d)^2 + (u_1 v_1)\cdot(u_2 v_2) + \cdots +$$

$$= \sum_i \sum_j u_i v_i \, u_j v_j$$

$$= \sum_i \sum_j (u_i u_j)(v_i v_j)$$

$$= \phi(\vec{u})^T \phi(\vec{v})$$

where $\phi(\vec{u}) =$

$$\begin{bmatrix} u_1 u_1 \\ \vdots \\ u_1 u_d \\ u_2 u_1 \\ \vdots \\ u_2 u_d \\ \vdots \\ u_d u_1 \\ \vdots \\ u_d u_d \end{bmatrix}$$

But $\phi(v)^T \phi(x)$ takes $O(d^2)$ time

$K(\vec{v}, \vec{x}) = (\vec{v}^T x)^2$ takes $O(d)$ time