

# Support Vector Machines (SVMs), Part 2

Yingming Li  
yingming@zju.edu.cn

Data Science & Engineering Research Center, ZJU

3rd April 2018

## Back to SVMs: Maximum margin solution is a fixed point of the Lagrangian function

- Recall, the maximum margin hyperplane is  $\operatorname{argmin}_{\mathbf{w}, b} \|\mathbf{w}\|^2$  subject to  $d_p(\mathbf{w}^T \mathbf{x}_p + b) \geq 1$ 
  - Minimization of a quadratic function subject to multiple linear inequality constraints

- Will use Lagrange multipliers,  $a_p$ , to write Lagrangian function

$$L(\mathbf{w}, b, \mathbf{a}) = \frac{1}{2} \|\mathbf{w}\|^2 - \sum_p a_p (d_p(\mathbf{w}^T \mathbf{x}_p + b) - 1)$$

- Note that  $\mathbf{x}_p$  and  $d_p$  are fixed for the optimization

# Dual form of Lagrangian eliminates $\mathbf{w}$ and $b$

- Set derivatives of  $L(\mathbf{w}, b, \mathbf{a})$  w.r.t.  $\mathbf{w}$  and  $b$  to 0

$$\frac{\partial}{\partial \mathbf{w}} L = 0 = \mathbf{w} - \sum_p a_p d_p \mathbf{x}_p$$

$$\Rightarrow \mathbf{w} = \sum_p a_p d_p \mathbf{x}_p$$

$$\frac{\partial}{\partial b} L = 0 = \sum_p a_p d_p$$

# Dual form of Lagrangian eliminates $\mathbf{w}$ and $b$

- Note that:

$$\mathbf{w}^T \mathbf{w} = \sum_p a_p d_p \mathbf{w}^T \mathbf{x}_p = \sum_p \sum_q a_p a_q d_p d_q \mathbf{x}_p^T \mathbf{x}_q$$

- “Primal” form of Lagrangian

$$\begin{aligned} L(\mathbf{w}, b, \mathbf{a}) &= \frac{1}{2} \mathbf{w}^T \mathbf{w} - \sum_p a_p (d_p (\mathbf{w}^T \mathbf{x}_p + b) - 1) \\ &= \frac{1}{2} \mathbf{w}^T \mathbf{w} - \sum_p a_p d_p \mathbf{w}^T \mathbf{x}_p - b \sum_p a_p d_p + \sum_p a_p \end{aligned}$$

## Dual form of Lagrangian eliminates $\mathbf{w}$ and $b$

$$\begin{aligned} L(\mathbf{w}, b, \mathbf{a}) &= \frac{1}{2} \mathbf{w}^T \mathbf{w} - \sum_p a_p d_p \mathbf{w}^T \mathbf{x}_p - b \sum_p a_p d_p + \sum_p a_p \\ &= \left( \frac{1}{2} - 1 \right) \sum_p \sum_a a_p a_q d_p d_q \mathbf{x}_p^T \mathbf{x}_q - b \cdot 0 + \sum_p a_p \end{aligned}$$

- So dual form of Lagrangian:

$$\tilde{L}(\mathbf{a}) = -\frac{1}{2} \sum_p \sum_q a_p a_q d_p d_q \mathbf{x}_p^T \mathbf{x}_q + \sum_p a_p$$

# Dual form of Lagrangian eliminates $\mathbf{w}$ and $b$

- Dual form of Lagrangian, maximize:

$$\tilde{L}(\mathbf{a}) = -\frac{1}{2} \sum_p \sum_q a_p a_q d_p d_q \mathbf{x}_p^T \mathbf{x}_q + \sum_p a_p$$

- Subject to the constraints

$$a_p \geq 0, \quad \forall p \quad \sum_p a_p d_p = 0$$

- Another quadratic programming problem subject to linear inequality and equality constraints

# Dual form allows use of Kernel function

- In dual form,  $\mathbf{x}_p$ ,  $\mathbf{x}_q$  only interact as inner products:

$$\tilde{L}(\mathbf{a}) = -\frac{1}{2} \sum_p \sum_q a_p a_q d_p d_q \mathbf{x}_p^T \mathbf{x}_q + \sum_p a_p$$

- Can replace  $\mathbf{x}_p^T \mathbf{x}_q$  with kernel function  $k(\mathbf{x}_p, \mathbf{x}_q)$
- Think of kernel function as inner product of feature vector of  $\mathbf{x}_p$ s in some high dimensional space

$$k(\mathbf{x}_p, \mathbf{x}_q) = \phi^T(\mathbf{x}_p) \phi(\mathbf{x}_q)$$

- But don't actually have to instantiate  $\phi(\mathbf{x}_p)$ 
  - More about kernels shortly

## Dual form is faster to solve when $D > N$

- Solving a quadratic program in  $M$  variables takes  $O(M^3)$  time in general
- Primal form involves  $D$  variables ( $\mathbf{w}$ )
  - Dimensionality of the data  $\mathbf{x}_p$ ,
  - Or dimensionality of features of the data  $\phi(\mathbf{x}_p)$
- Dual form involves  $N$  variables ( $\mathbf{a}$ )
  - Number of training points
- SVMs are generally most useful with kernels
  - So  $D > N$  and the dual is faster to solve



# Classify new points using $y(\mathbf{x})$

- Actual prediction function is still

$$y(\mathbf{x}) = \mathbf{w}^T \mathbf{x} + b$$

- Get  $\mathbf{w}$  from primal Lagrangian

$$\mathbf{w} = \sum_p a_p d_p \mathbf{x}_p$$

- Will discuss  $b$  shortly, so

$$y(\mathbf{x}) = \sum_p a_p d_p \mathbf{x}_p^T \mathbf{x} + b$$

# Classify new points using $y(\mathbf{x})$ , with kernel

- With a kernel,  $\mathbf{w}^T = \sum_p a_p d_p \phi(\mathbf{x}_p)$
- Actual prediction function is

$$\begin{aligned} y(\mathbf{x}) &= \mathbf{w}^T \phi(\mathbf{x}) + b \\ &= \sum_p a_p d_p \phi^T(\mathbf{x}_p) \phi(\mathbf{x}) + b \\ &= \sum_p a_p d_p k(\mathbf{x}_p, \mathbf{x}) + b \end{aligned}$$

- In practice, save all  $\mathbf{x}_p$  with  $a_p > 0$ 
  - And compute  $k(\mathbf{x}_p, \mathbf{x})$  at test time

# KKT Conditions

- In the case of SVMs, the KKT conditions are

$$a_p \geq 0$$

$$d_p y(\mathbf{x}_p) - 1 \geq 0$$

$$a_p (d_p y(\mathbf{x}_p) - 1) = 0$$

- So either  $a_p = 0$  or  $d_p y(\mathbf{x}_p) - 1 = 0$ 
  - Constraint from each point is either ignored or active
- When  $a_p = 0$ ,  $\mathbf{w}$  is independent of that point
- When  $d_p y(\mathbf{x}_p) = 1$ , that point is on the margin
  - It is a support vector
  - Thus only the support vectors contribute to  $\mathbf{w}$

# Compute $b$ from support vectors

- Get  $b$  from support vectors, which have margin 1
- In the linear case, for a support vector  $\mathbf{x}_q^s$

$$y(\mathbf{x}_q^s) = d_p = \mathbf{w}^T \mathbf{x}_q^s + b$$

$$b = d_a^s - \sum_p a_p d_p \mathbf{x}_p^T \mathbf{x}_a^s$$

- When using a kernel

$$b = d_a^s - \sum_p a_p d_p k(\mathbf{x}_p, \mathbf{x}_q^s)$$

- For numerical stability, average over all SVs

## Summary so far

- Finding the maximum margin hyperplane has been formulated as a constrained quadratic program
  - Convex problem, well studied, easy conceptually to solve
- Can be solved in the primal or dual formulation
  - Dual formulation permits the use of kernel functions
- Only some data points contribute to the solution
  - The support vectors
- So far, only applies to linearly separable data

# Thank you!