

# Statistical Machine Learning (BE4M33SSU)

## Lecture 4: Support Vector Machines II

Czech Technical University in Prague

- ◆ Lagrange duality
- ◆ Dual formulation of Support Vector Machines
- ◆ Positive definite kernel
- ◆ Examples of kernels

**BE4M33SSU – Statistical Machine Learning, Winter 2016**

# Constrained optimization problem and its Lagrange function

- ◆ For convex  $f: \mathbb{R}^n \rightarrow \mathbb{R}$ ,  $g_i \in \mathbb{R}^n \rightarrow \mathbb{R}$ ,  $i \in \{1, \dots, m\}$ , we want to solve:

$$\boldsymbol{\theta}^* = \underset{\boldsymbol{\theta} \in \mathbb{R}^n}{\operatorname{argmin}} f(\boldsymbol{\theta}) \quad \text{s.t.} \quad g_i(\boldsymbol{\theta}) \leq 0, i \in \{1, \dots, m\}$$

- ◆ Let us define a Lagrange function  $L: \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R} \cup \{\infty\}$  such that

$$L(\boldsymbol{\theta}, \boldsymbol{\alpha}) = f(\boldsymbol{\theta}) + \sum_{i=1}^m \alpha_i g_i(\boldsymbol{\theta})$$

- ◆ The original problem is equivalent to an unconstrained problem

$$\boldsymbol{\theta}^* = \underset{\boldsymbol{\theta} \in \mathbb{R}^n}{\operatorname{argmin}} P(\boldsymbol{\theta})$$

where

$$P(\boldsymbol{\theta}) = \max_{\boldsymbol{\alpha} \geq \mathbf{0}} L(\boldsymbol{\theta}, \boldsymbol{\alpha}) = \begin{cases} f(\boldsymbol{\theta}) & \text{if } \sum_{i=1}^m g_i(\boldsymbol{\theta}) \leq 0, i \in \{1, \dots, m\} \\ \infty & \text{otherwise} \end{cases}$$

## Lagrange dual problem

- ◆ The dual problem reads

$$\alpha^* = \operatorname{argmax}_{\alpha \geq 0} D(\alpha) \quad \text{where} \quad D(\alpha) = \min_{\theta \in \mathbb{R}^n} L(\theta, \alpha)$$

- ◆ **Strong duality:** if the problem is convex and there exists  $\theta \in \mathbb{R}^n$  such that  $g_i(\theta) < 0, i \in \{1, \dots, m\}$ , then the duality gap is zero:

$$P(\theta^*) = D(\alpha^*)$$

- ◆ **Karush-Kuhn-Tucker conditions:**  $(\theta^*, \alpha^*)$  is the solution of the primal and the dual problem with zero duality gap if and only if:

- 1)  $\nabla f(\theta^*) + \sum_{i=1}^m \alpha_i^* \nabla g_i(\theta^*) = \mathbf{0}$
- 2)  $g_i(\theta^*) \leq 0, \quad i \in \{1, \dots, m\}$
- 3)  $\sum_{i=1}^m \alpha_i^* g_i(\theta^*) = 0$

## Primal SVM problem

- ◆ The formulation of the linear SVM algorithm reads

$$(\mathbf{w}^*, b^*, \boldsymbol{\xi}^*) = \underset{\substack{(\mathbf{w}, b) \in \mathbb{R}^{n+1} \\ \boldsymbol{\xi} \in \mathbb{R}^m}}{\operatorname{argmin}} \left( \frac{\lambda}{2} \|\mathbf{w}\|^2 + \frac{1}{m} \sum_{i=1}^m \xi_i \right)$$

$$\begin{aligned} \text{s.t. } y^i (\langle \mathbf{w}, \phi(x^i) \rangle + b) &\geq 1 - \xi_i, & i \in \{1, \dots, m\} \\ \xi_i &\geq 0, & i \in \{1, \dots, m\} \end{aligned}$$

- ◆ The Lagrange function

$$L(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\mu}) = \frac{\lambda}{2} \|\mathbf{w}\|^2 + \sum_{i=1}^m \xi_i - \sum_{i=1}^m \alpha_i (y^i (\langle \mathbf{w}, \phi(x^i) \rangle + b) - 1 + \xi_i) - \sum_{i=1}^m \mu_i \xi_i$$

- ◆ The SVM learning expressed as an unconstrained problem

$$(\mathbf{w}^*, b^*, \boldsymbol{\xi}^*) \in \underset{\substack{(\mathbf{w}, b) \in \mathbb{R}^{n+1} \\ \boldsymbol{\xi} \in \mathbb{R}^m}}{\operatorname{argmin}} P(\mathbf{w}, b, \boldsymbol{\xi}) \quad \text{where} \quad P(\mathbf{w}, b, \boldsymbol{\xi}) = \max_{\substack{\boldsymbol{\alpha} \geq 0 \\ \boldsymbol{\mu} \geq 0}} L(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\mu})$$

## Dual SVM problem

- ◆ The dual SVM problem reads

$$(\boldsymbol{\alpha}^*, \boldsymbol{\mu}^*) = \underset{\substack{\boldsymbol{\alpha} \geq 0 \\ \boldsymbol{\mu} \geq 0}}{\operatorname{argmax}} D(\boldsymbol{\alpha}, \boldsymbol{\mu}) \quad \text{where} \quad D(\boldsymbol{\alpha}, \boldsymbol{\mu}) = \min_{\substack{\boldsymbol{w} \in \mathbb{R}^n \\ b \in \mathbb{R} \\ \boldsymbol{\xi} \in \mathbb{R}^m}} L(\boldsymbol{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\mu})$$

- ◆ Given  $(\boldsymbol{\alpha}, \boldsymbol{\mu})$ , the function  $L(\boldsymbol{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\mu})$  w.r.t.  $(\boldsymbol{w}, b, \boldsymbol{\xi})$  is convex and differentiable hence we find the optimum by solving:

$$\frac{\partial L(\boldsymbol{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\mu})}{\partial \boldsymbol{w}} = \mathbf{0}, \quad \frac{\partial L(\boldsymbol{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\mu})}{\partial b} = \mathbf{0}, \quad \frac{\partial L(\boldsymbol{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\mu})}{\partial \boldsymbol{\xi}} = \mathbf{0}$$

by which we get

$$D(\boldsymbol{\alpha}, \boldsymbol{\mu}^*(\boldsymbol{\alpha})) = \begin{cases} \sum_{i=1}^m \alpha_i - \frac{1}{2\lambda} \sum_{i=1}^m \sum_{j=1}^m \alpha_i \alpha_j y^i y^j \langle \boldsymbol{\phi}(x^i), \boldsymbol{\phi}(x^j) \rangle & \text{if } \sum_{i=1}^m \alpha_i y^i = 0 \\ -\infty & \text{otherwise} \end{cases}$$

$\boldsymbol{\alpha} \in [0, \frac{1}{m}]^m$

## Dual SVM problem

- ◆ The dual SVM formulation is a **convex quadratic program**

$$\begin{aligned}
 \boldsymbol{\alpha}^* &= \operatorname{argmax}_{\boldsymbol{\alpha} \in \mathbb{R}^m} \left( \sum_{i=1}^m \alpha_i - \frac{1}{2\lambda} \sum_{i=1}^m \sum_{j=1}^m \alpha_i \alpha_j y^i y^j \langle \phi(x^i), \phi(x^j) \rangle \right) \\
 \text{s.t.} \quad & \sum_{i=1}^m \alpha_i y^i = 0, \quad 0 \leq \alpha_i \leq \frac{1}{m} \quad i \in \{1, \dots, m\}
 \end{aligned}$$

- ◆ The primal solution  $(\boldsymbol{w}^*, b^*)$  is obtained from  $\boldsymbol{\alpha}^*$  using KKT conditions

$$\boldsymbol{w}^* = \sum_{i=1}^m y^i \phi(x^i) \alpha_i^* \quad \text{and} \quad b^* = y^i - \langle \boldsymbol{w}^*, \phi(x^i) \rangle, \quad \forall i \in \mathcal{I}_{\text{SV}}^<$$

where  $\mathcal{I}_{\text{SV}}^< = \{i \in \{1, \dots, m\} \mid 0 < \alpha_i < \frac{1}{m}\}$  are boundary SVs.

- ◆ To represent the classifier we need only the support vectors: training examples with indices  $\mathcal{I}_{\text{SV}} = \{i \in \{1, \dots, m\} \mid \alpha_i > 0\}$ .

## Kernel SVM

- ◆ We see that the values of **kernel function**  $k(x, x') = \langle \phi(x), \phi(x') \rangle$  are **sufficient to learn parameters**

$$\alpha^* = \operatorname{argmin}_{\alpha \in \mathbb{R}^m} \left( \sum_{i=1}^m \alpha_i - \frac{1}{2\lambda} \sum_{i=1}^m \sum_{j=1}^m \alpha_i \alpha_j y^i y^j k(x^i, x^j) \right)$$

$$\text{s.t.} \quad \sum_{i=1}^m \alpha_i y^i = 0, \quad 0 \leq \alpha_i \leq \frac{1}{m} \quad i \in \{1, \dots, m\}$$

and to evaluate the **SVM classifier**

$$h(x; \alpha^*, b^*) = \operatorname{sign}(\langle \mathbf{w}^*, \phi(x) \rangle + b^*) = \operatorname{sign} \left( \sum_{i=1}^m y^i \alpha_i^* k(x^i, x) + b^* \right)$$

- ◆ When is it useful to represent inputs by  $k(x, x')$  instead of  $\phi(x)$  ?
- ◆ What choices of  $k: \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$  are reasonable ?

# Linear SVM versus kernel SVM

Linear SVM

Kernel SVM

	Input representation	
comput. time	$\psi: \mathcal{X} \rightarrow \mathbb{R}^n$ $\mathcal{O}(n)$	$k: \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ typically $\mathcal{O}(d)$
	Classifier	
linear score memory/time	$\langle w, \psi(x) \rangle$ $\mathcal{O}(n)$	$\sum_{i=1}^m \alpha_i y_i k(x^i, x)$ $\mathcal{O}(m_{sv} + m_{sv} \cdot d)$
	Learning (QP solver)	
memory/minimal time	$\mathcal{O}(n \cdot m)$	$\mathcal{O}(m^2)$

$d$  is the size of a single input  $x \in \mathcal{X}$

$m$  is the number of training examples

$m_{sv} = |\{i \in \{1, \dots, m\} \mid \alpha_i > 0\}|$  number of support vectors



## Positive definite kernel

**Definition 1.** Let  $\mathcal{X}$  be a non-empty set. The function  $k: \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$  is a *positive definite kernel* if it is *symmetric* and for any finite set of inputs  $x^1, \dots, x^m$ , the *kernel matrix*  $\mathbf{K} \in \mathbb{R}^{m \times m}$  with elements  $K_{i,j} = k(x^i, x^j)$  is *positive semi-definite*.

- ◆ The kernel matrix  $\mathbf{K} \in \mathbb{R}^{m \times m}$  represents similarities between each pair of inputs  $\{x^1, \dots, x^m\}$ :

$$\mathbf{K} = \begin{pmatrix} k(x^1, x^1), & k(x^1, x^2), & \dots, & k(x^1, x^m) \\ k(x^2, x^1), & k(x^2, x^2), & \dots, & k(x^2, x^m) \\ \vdots & & & \\ k(x^m, x^1), & k(x^m, x^2), & \dots, & k(x^m, x^m) \end{pmatrix}$$

- ◆ A matrix  $\mathbf{K} \in \mathbb{R}^{m \times m}$  is PSD if for every  $\alpha \in \mathbb{R}^m$ ,  $\alpha^T \mathbf{K} \alpha \geq 0$ .

## Hilbert space

**Definition 2.** A Hilbert space  $\mathcal{H}$  is a complete vector space with a dot product  $\langle \cdot, \cdot \rangle: \mathcal{H} \times \mathcal{H} \rightarrow \mathbb{R}$  that satisfies the following properties:

- ◆ *Symmetry:*  $\langle f, g \rangle = \langle g, f \rangle, \forall f, g \in \mathcal{H}$
- ◆ *Linearity:*  $\langle \alpha_1 f_1 + \alpha_2 f_2, g \rangle = \alpha_1 \langle f_1, g \rangle + \alpha_2 \langle f_2, g \rangle, \forall f_1, f_2, g \in \mathcal{H}, \alpha_1, \alpha_2 \in \mathbb{R}$
- ◆ *Positive definiteness:*  $\langle f, f \rangle \geq 0$  with equality iff  $f = 0$

The dot product defines a norm  $\|f\|_{\mathcal{H}} = \sqrt{\langle f, f \rangle}$

## Every feature map defines PSD kernel

**Theorem 1.** *Let  $\phi: \mathcal{X} \rightarrow \mathcal{H}$  be a feature map representing inputs from  $\mathcal{X}$  in a Hilbert space  $\mathcal{H}$ . Then the function  $k(x, x') = \langle \phi(x), \phi(x') \rangle$  is a positive definite kernel.*

**Proof.** *Given  $\{x^i \in \mathcal{X} \mid i = 1, \dots, m\}$ , the kernel matrix  $\mathbf{K} \in \mathbb{R}^{m \times m}$ , with elements  $K_{i,j} = k(x^i, x^j)$  is PSD because*

$$\begin{aligned}
 \boldsymbol{\alpha}^T \mathbf{K} \boldsymbol{\alpha} &= \sum_{i=1}^m \sum_{j=1}^m \alpha_i \alpha_j \langle \phi(x^i), \phi(x^j) \rangle \\
 &= \left\langle \sum_{i=1}^m \alpha_i \phi(x^i), \sum_{j=1}^m \alpha_j \phi(x^j) \right\rangle \\
 &= \left\| \sum_{i=1}^m \alpha_i \phi(x^i) \right\|^2 \geq 0
 \end{aligned}$$

## Every kernel defines a feature space

**Theorem 2.** *For every positive definite kernel  $k: \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$  there exists a Hilbert space  $\mathcal{H}$  and a feature map  $\phi: \mathcal{X} \rightarrow \mathcal{H}$  such that  $k(x, x') = \langle \phi(x), \phi(x') \rangle$ .*

Proof for a kernel defined on a finite input space  $\mathcal{X}$ :

The kernel matrix  $\mathbf{K} \in \mathbb{R}^{|\mathcal{X}| \times |\mathcal{X}|}$  is symmetric and PSD hence the spectral decomposition exists  $\mathbf{K} = \mathbf{V}\mathbf{D}\mathbf{V}^T$  where  $\mathbf{V} \in \mathbb{R}^{m \times m}$  is orthogonal and  $\mathbf{D} = \text{diag}(\lambda_1, \dots, \lambda_{|\mathcal{X}|})$  is diagonal matrix of non-negative eigenvalues.

Therefore  $\mathbf{K} = \mathbf{\Phi}^T \mathbf{\Phi}$  where  $\mathbf{\Phi}^T = \mathbf{V}\mathbf{D}^{\frac{1}{2}}$ .

## String Subsequence kernel

- ◆ Input space  $\mathcal{X} = \cup_{d=0}^{\infty} \Sigma^d$  contains all strings on a finite alphabet  $\Sigma$
- ◆ The features measure the number of occurrences of subsequences of length  $q$  weighting them according to their length and decay factor  $\lambda \in (0, 1]$ .

$$\phi: \mathcal{X} \rightarrow \mathbb{R}^{|\Sigma|^q} \quad \text{and} \quad \phi_u(s) = \sum_{i: u=s[i]} \lambda^{l(i)}, \quad \forall u \in \Sigma^q$$

- ◆ Example for strings "cat", "car", "bat" and "bar" and  $q = 2$ :

	c-a	c-t	a-t	b-a	b-t	c-r	a-r	b-r
$\phi(\text{"cat"})$	$\lambda^2$	$\lambda^3$	$\lambda^2$	0	0	0	0	0
$\phi(\text{"car"})$	$\lambda^2$	0	0	0	0	$\lambda^3$	$\lambda^2$	0
$\phi(\text{"bat"})$	0	0	$\lambda^2$	$\lambda^2$	$\lambda^3$	0	0	0
$\phi(\text{"bar"})$	0	0	0	$\lambda^2$	0	0	$\lambda^2$	$\lambda^3$

$$k(\text{"cat"}, \text{"car"}) = \lambda^4, \quad k(\text{"cat"}, \text{"bar"}) = \lambda^4, \quad k(\text{"cat"}, \text{"bar"}) = 0, \dots$$

- ◆ The kernel value  $k(s, t)$  can be computed by dynamic programming in time  $\mathcal{O}(q |s| |t|)$  instead of  $\mathcal{O}(|\Sigma|^q)$  required by using explicit features.

## Polynomial kernel of degree 2

- ◆ Consider quadratic function of  $d$ -dimensional inputs  $\mathbf{x} \in \mathcal{X} = \mathbb{R}^d$

$$\begin{aligned}
 f(\mathbf{x}) &= w_0 + \sqrt{2}x_1w_1 + \cdots + \sqrt{2}x_dw_d + x_1^2w_{1,1} + \cdots + x_d^2w_{d,d} \\
 &\quad + \sqrt{2}x_1x_2w_{1,2} \cdots + \sqrt{2}x_1x_dw_{1,d} + \cdots + \sqrt{2}x_{d-1}x_dw_{d-1,d} \\
 &= \langle \mathbf{w}, \phi(\mathbf{x}) \rangle
 \end{aligned}$$

where  $\mathbf{w} \in \mathbb{R}^n$ ,  $n = \frac{(d+1)d}{2} + d + 1$ , and

$$\phi(\mathbf{x}) = (1, \sqrt{2}x_1, \dots, \sqrt{2}x_d, x_1^2, \dots, x_d^2, \sqrt{2}x_1x_2, \dots, \sqrt{2}x_{d-1}x_d)$$

- ◆ The dot product of two inputs mapped to the features can be computed via 2nd order polynomial kernel

$$k(\mathbf{x}, \mathbf{x}') = \langle \phi(\mathbf{x}), \phi(\mathbf{x}') \rangle = (\langle \mathbf{x}, \mathbf{x}' \rangle + 1)^2$$

- ◆ The kernel evaluation requires  $d$  multiplications/additions instead of  $n = \frac{(d+1)d}{2} + d + 1$  when computing the feature maps explicitly.

## Examples of kernels

List of frequently used kernels on vectorial inputs  $\mathcal{X} = \mathbb{R}^d$ :

- ◆ Homogeneous polynomial of degree  $p$ :  $k(\mathbf{x}, \mathbf{x}') = \langle \mathbf{x}, \mathbf{x}' \rangle^p$

The feature map composed of all monomials of degree  $p$ .

For example, if  $d = p = 2$ ,  $\phi(\mathbf{x}) = (x_1^2, \sqrt{2}x_1x_2, x_2^2)$
- ◆ Inhomogeneous polynomial of degree  $p$ :  $k(\mathbf{x}, \mathbf{x}') = (\langle \mathbf{x}, \mathbf{x}' \rangle + 1)^p$

The feature map composed of all monomials up to degree  $p$ .

For example, if  $d = p = 2$ ,  $\phi(\mathbf{x}) = (x_1^2, \sqrt{2}x_1x_2, x_2^2, \sqrt{2}x_1, \sqrt{2}x_2, 1)$
- ◆ Gaussian kernel:  $k(\mathbf{x}, \mathbf{x}') = \exp(-\sigma \|\mathbf{x} - \mathbf{x}'\|^2)$

The feature map represents inputs in infinite dimensional space.

# Summary

- ◆ Lagrange duality
- ◆ Dual formulation of Support Vector Machines
- ◆ Positive definite kernel
- ◆ Examples of kernels