Statistical Machine Learning (BE4M33SSU)
Lecture 5: Support Vector Machines II

Czech Technical University in Prague
V.Franc

SVM as Quadratic Program

- Find linear classifier $h(x;\boldsymbol{w},b) = \mathrm{sign}(\langle \boldsymbol{\phi}(x), \boldsymbol{w}\rangle + b)$ by solving

$$(\boldsymbol{w}^*, b^*) = \operatorname*{argmin}_{\boldsymbol{w}\in\mathbb{R}^n, b\in\mathbb{R}} \left(\underbrace{\frac{1}{2}\|\boldsymbol{w}\|^2}_{\text{penalty term}} + \underbrace{C\sum_{i=1}^{m} \max\{0, 1 - y^i(\langle \boldsymbol{w}, \boldsymbol{\phi}(x^i)\rangle + b)\}}_{\text{empirical error}} \right)$$

where $C > 0$ is the regularization constant.

- It can be re-formulated as a convex *quadratic program*

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

subject to

$$\begin{array}{lll} \xi_i & \geq & 1 - y^i(\langle \boldsymbol{w}, \boldsymbol{\phi}(x^i)\rangle + b)\,, \quad i \in \{1,\ldots,m\} \\ \xi_i & \geq & 0\,, \qquad\qquad\qquad\qquad\qquad\quad\; i \in \{1,\ldots,m\} \end{array}$$

From Primal SVM to Dual SVM problem

- Lagrangian of the primal SVM problem:

$$L(\boldsymbol{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\mu}) = \underbrace{\frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{m}\xi_i}_{\text{original objective}}$$

$$\underbrace{+\sum_{i=1}^{m}\alpha_i\big(1 - y^i(\langle\boldsymbol{w}, \boldsymbol{\phi}(x^i)\rangle + b) - \xi_i\big) + \sum_{i=1}^{m}\mu_i\big(-\xi_i\big)}_{\text{constraint violation penalty}}$$

- Strong duality:

$$\underbrace{\min_{\substack{\boldsymbol{w}\in\mathbb{R}^n\\ b\in\mathbb{R}\\ \boldsymbol{\xi}\in\mathbb{R}^m}} \max_{\substack{\boldsymbol{\alpha}\in\mathbb{R}^m_+\\ \boldsymbol{\mu}\in\mathbb{R}^m_+}} L(\boldsymbol{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\mu})}_{\text{primal problem}} = \underbrace{\max_{\substack{\boldsymbol{\alpha}\in\mathbb{R}^m_+\\ \boldsymbol{\mu}\in\mathbb{R}^m_+}} \min_{\substack{\boldsymbol{w}\in\mathbb{R}^n\\ b\in\mathbb{R}\\ \boldsymbol{\xi}\in\mathbb{R}^m}} L(\boldsymbol{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\mu})}_{\text{dual problem}}$$

Dual SVM problem

- The dual SVM formulation is a convex quadratic program

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

- The primal variables $(\boldsymbol{w}, b)$ are obtained from the dual variables $\boldsymbol{\alpha}$ by

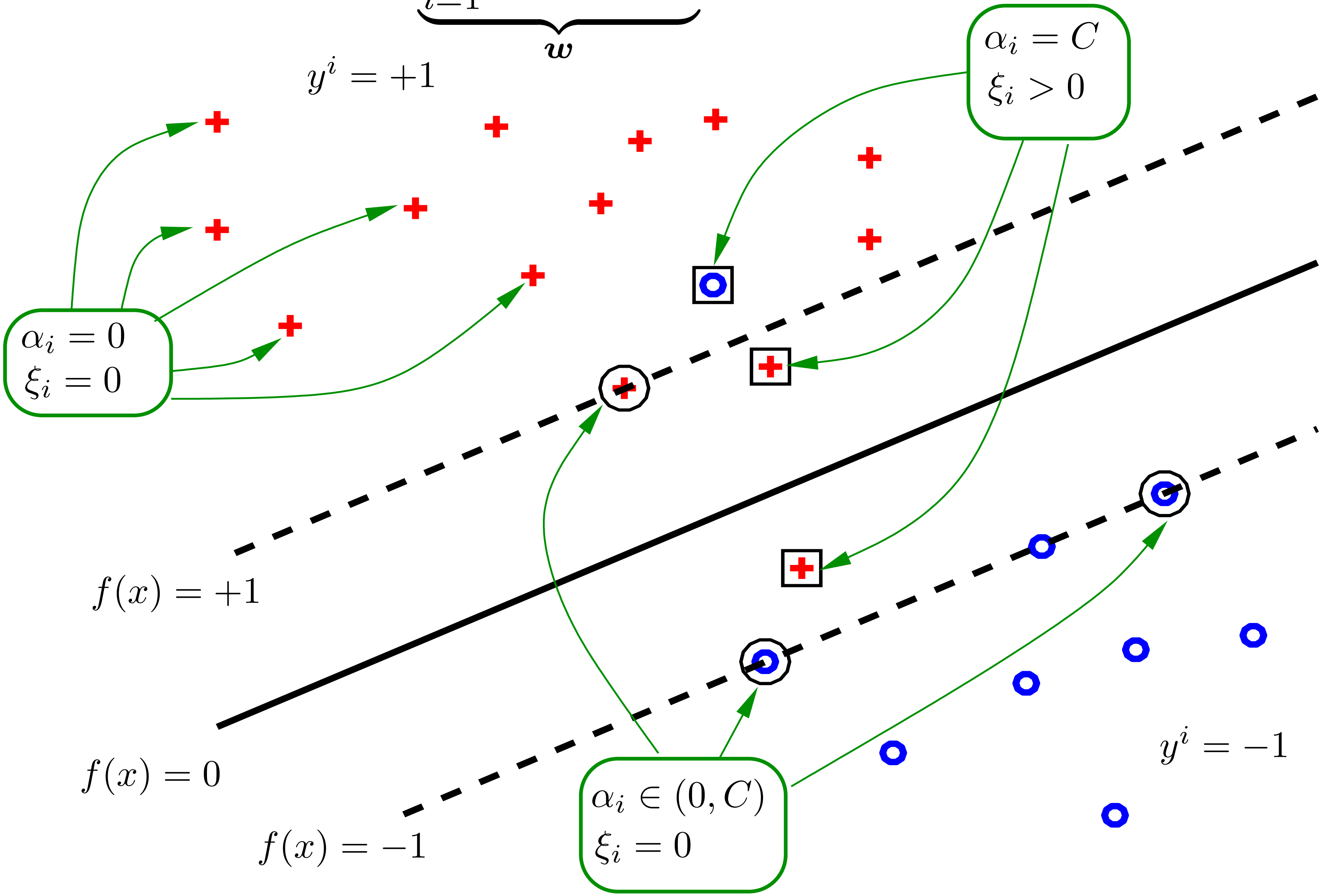
$$
\boldsymbol{w} = \sum_{i=1}^m y^i \, \boldsymbol{\phi}(x^i) \, \alpha_i = \sum_{i\in\mathcal{I}_{\rm SV}} y^i \, \boldsymbol{\phi}(x^i) \, \alpha_i
$$

where $\mathcal{I}_{\rm sv} = \{ j \mid \alpha_j > 0 \}$ are indices of **support vectors**, and

$$
b = y^i - \langle \boldsymbol{w}, \boldsymbol{\phi}(x^i) \rangle \,, \qquad \forall i \in \mathcal{I}^{\rm b}_{\rm sv} = \{ j \mid 0 < \alpha_j < C \}
$$

Example: SVM classifier

$$f(x) = \langle \boldsymbol{w}, \boldsymbol{\phi}(x) \rangle + b = \langle \underbrace{\sum_{i=1}^{m} y^i \alpha_i \boldsymbol{\phi}(x^i)}_{\boldsymbol{w}}, \boldsymbol{\phi}(x) \rangle + b$$



Kernel SVM

- The SVM algorithm requires observations in terms of **dot products** only:

Learning: Given $\mathcal{T}^m = \{(x^i, y^i) \in \mathcal{X} \times \{-1, +1\} \mid i = 1, \ldots, m\}$, solve

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

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

Prediction:

$$h(x; \boldsymbol{\alpha}, b) = \operatorname{sign}(\langle \boldsymbol{w}, \boldsymbol{\phi}(x) \rangle + b) = \operatorname{sign} \left(\sum_{i \in \mathcal{I}_{\mathrm{SV}}} y^i \alpha_i \;\; k(x^i, x) + b \right)$$

- Given a feature map $\boldsymbol{\phi} \colon \mathcal{X} \to \mathbb{R}^n$, define kernel function $k \colon \mathcal{X} \times \mathcal{X} \to \mathbb{R}$

$$k(x, x') = \langle \boldsymbol{\phi}(x), \boldsymbol{\phi}(x') \rangle$$

Homogeneous polynomial kernel of degree 2

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

$$f(\boldsymbol{x}) = x_1^2\, w_1 + \cdots + x_d^2\, w_d + \sqrt{2}\, x_1\, x_2\, w_{d+1} + \cdots + \sqrt{2}\, x_{d-1}\, x_d\, w_n + b$$
$$= \langle \boldsymbol{w}, \boldsymbol{\phi}(\boldsymbol{x}) \rangle + b$$

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

$$\boldsymbol{\phi}(\boldsymbol{x}) = (x_1^2, \ldots, x_d^2, \sqrt{2} x_1 x_2, \ldots, \sqrt{2} x_{d-1} x_d)$$

- The dot product of $\boldsymbol{\phi}(\boldsymbol{x})$ and $\boldsymbol{\phi}(\boldsymbol{x}')$ can be via kernel

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

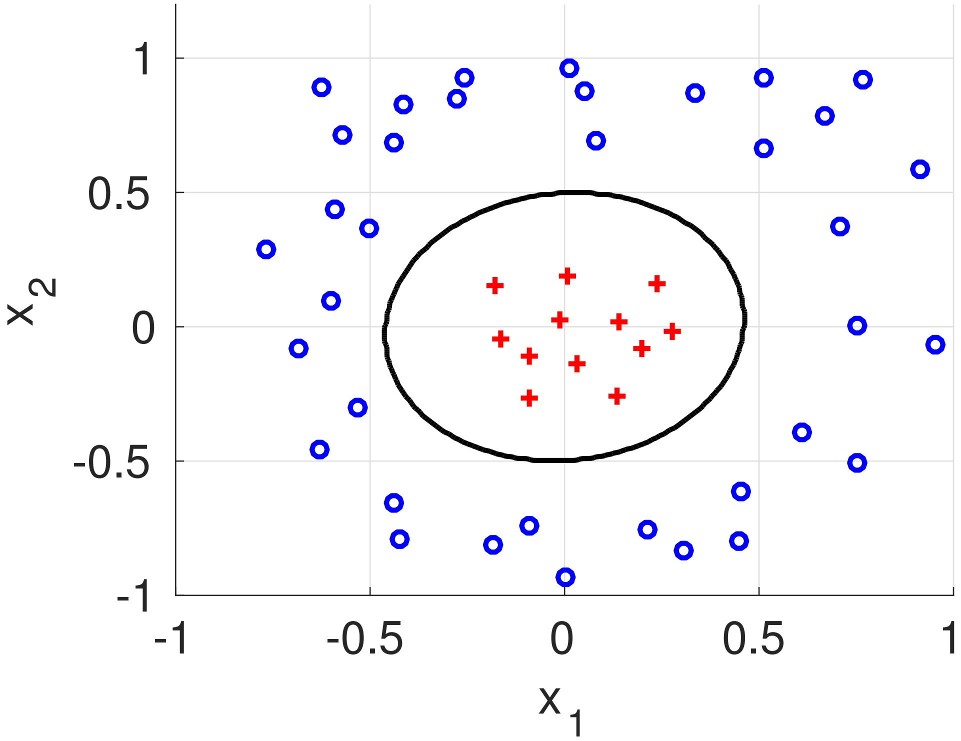
- In case of $d = 2$, we have

$$k(\boldsymbol{x}, \boldsymbol{x}') = (x_1\, x_1' + x_2\, x_2')^2 = x_1^2\, x_1'^2 + 2\, x_1\, x_2\, x_1'\, x_2' + x_2^2\, x_2'^2$$
$$= \Big\langle (x_1^2,\ \sqrt{2}\, x_1\, x_2\,, x_2^2), (x_1'^2,\ \sqrt{2}\, x_1'\, x_2'\,, x_2'^2) \Big\rangle$$

Example: Homogeneous polynomial kernel of degree 2

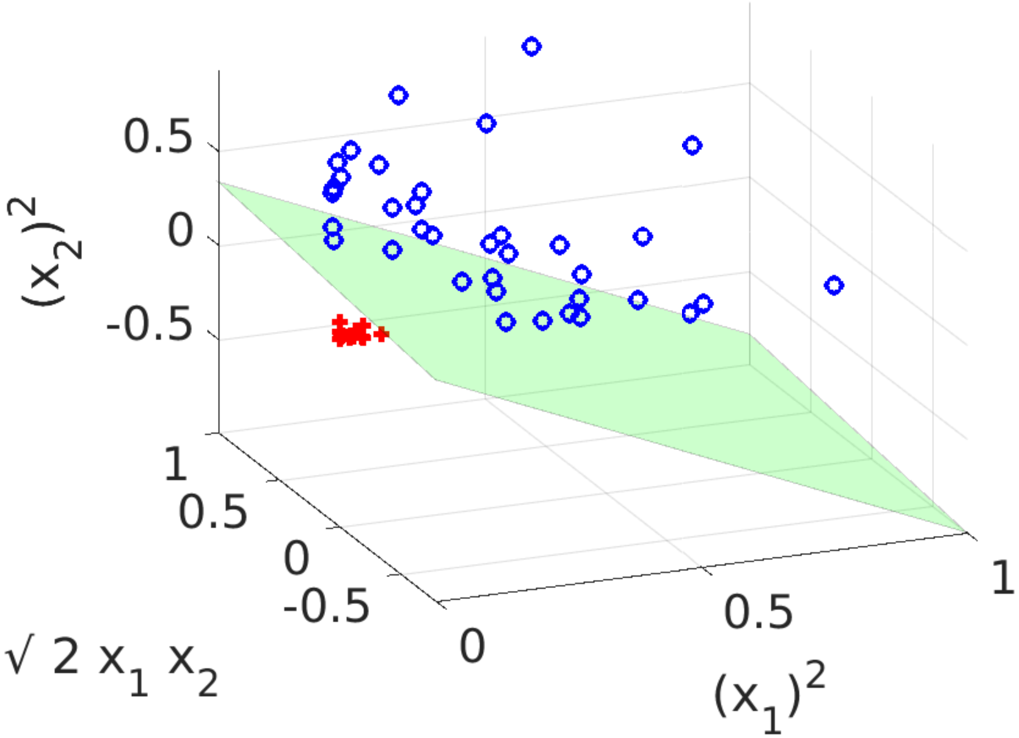
observations

$$\boldsymbol{x} = (x_1, x_2) \in \mathbb{R}^2$$



features

$$\phi(\boldsymbol{x}) = (x_1^2, \sqrt{2}x_1x_2, x_2^2) \in \mathbb{R}^3$$



$$f(\boldsymbol{x}) = w_1\phi_1(\boldsymbol{x}) + w_2\phi_2(\boldsymbol{x}) + w_3\phi_3(\boldsymbol{x}) + b = w_1x_1^2 + w_2\sqrt{2}x_1\,x_2 + w_3\,x_2^2 + b$$

Radial basis function kernel

- Assume the observations are real-valued features: $\mathcal{X} = \mathbb{R}^d$
- The RBF kernel
$$k(\boldsymbol{x}, \boldsymbol{x}') = \exp(-\gamma \|\boldsymbol{x} - \boldsymbol{x}'\|^2)$$
corresponds to dot product $\langle \boldsymbol{\phi}(\boldsymbol{x}), \boldsymbol{\phi}(\boldsymbol{x}') \rangle$ in infinite dimensional space.
- In one dimensional case, $\mathcal{X} = \mathbb{R}$ the RBF kernel reads

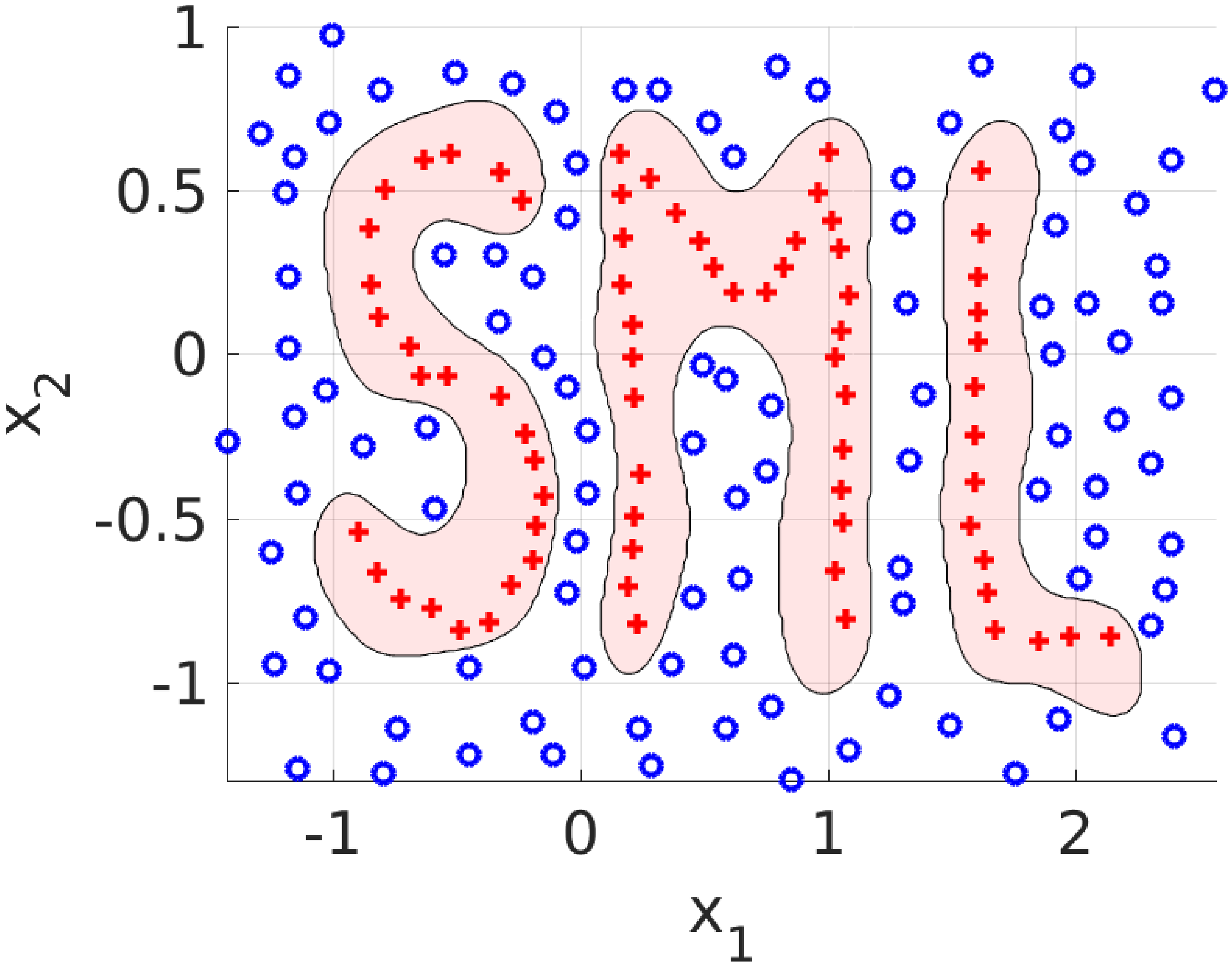
$$\begin{aligned}
k(x, x') &= \exp(-\gamma (x - x')^2) \\
&= \exp(-\gamma\, x^2) \exp(-\gamma\, x'^2) \exp(2\, \gamma\, x\, x') \\
&= \exp(-\gamma\, x^2) \exp(-\gamma\, x'^2) \sum_{n=0}^{\infty} \frac{(2\, x\, x'\, \gamma)^n}{n!} \\
&= \exp(-\gamma\, x^2)\ \exp(-\gamma\, x'^2) \sum_{n=0}^{\infty} \frac{(\sqrt{2\, \gamma}\, x)^n}{\sqrt{n!}} \frac{(\sqrt{2\, \gamma}\, x')^n}{\sqrt{n!}}
\end{aligned}$$

where we used the Taylor expansion $\exp(a) = \sum_{n=0}^{\infty} \frac{a^n}{n!}$

Example: SVM with RBF kernel

$$f(\boldsymbol{x}) = \sum_{i \in \mathcal{I}_{\mathrm{SV}}} y^i \, \alpha_i \, \exp(-\gamma \|\boldsymbol{x} - \boldsymbol{x}^i\|^2) + b$$

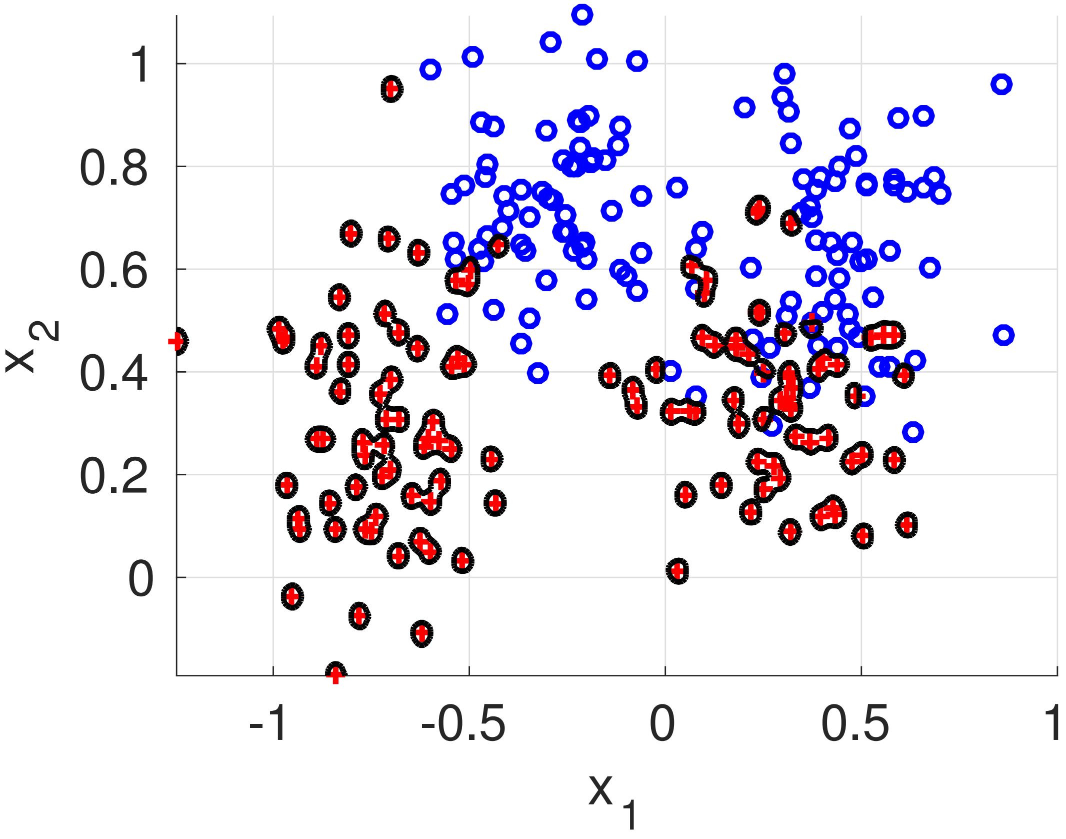
$$\gamma = 10, \, C = 1000$$

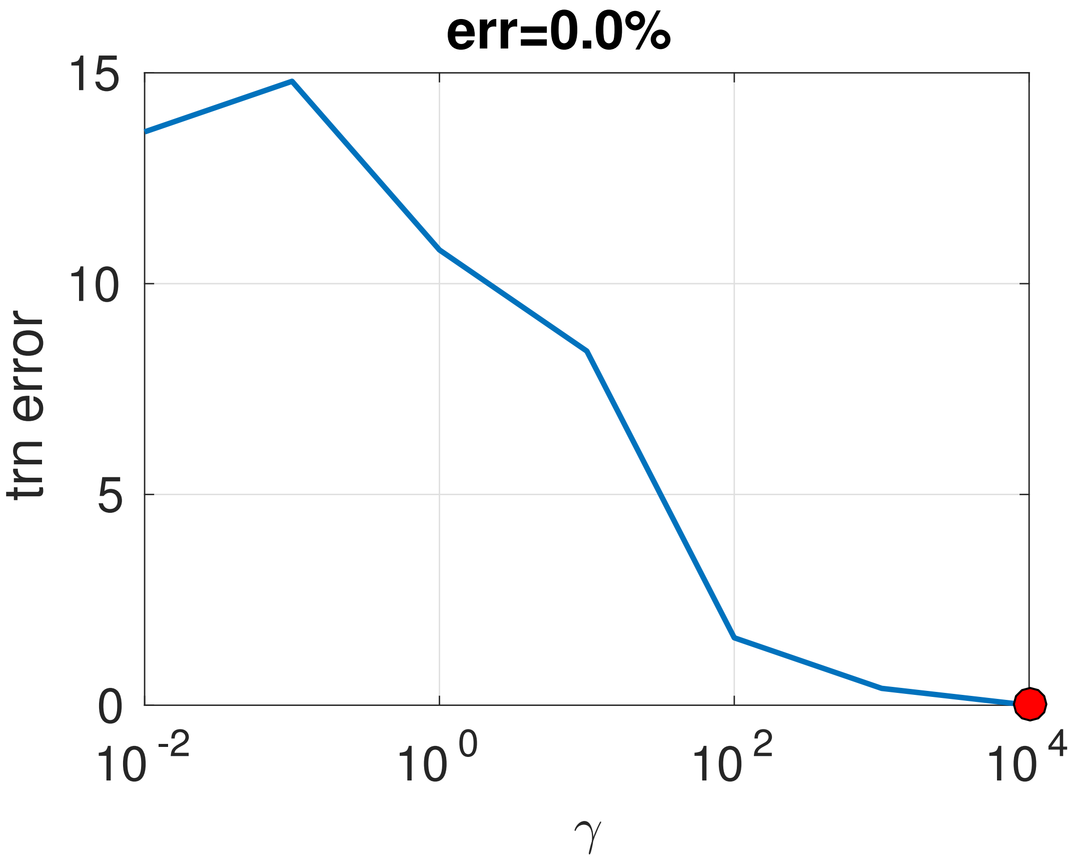


Example: SVM with RBF kernel training error vs. γ

$$f(\boldsymbol{x}) = \sum_{i \in \mathcal{I}_{\text{SV}}} y^i \, \alpha_i \, \exp(-\gamma \|\boldsymbol{x} - \boldsymbol{x}^i\|^2) + b$$

$$\gamma = 10000, \; C = 100$$

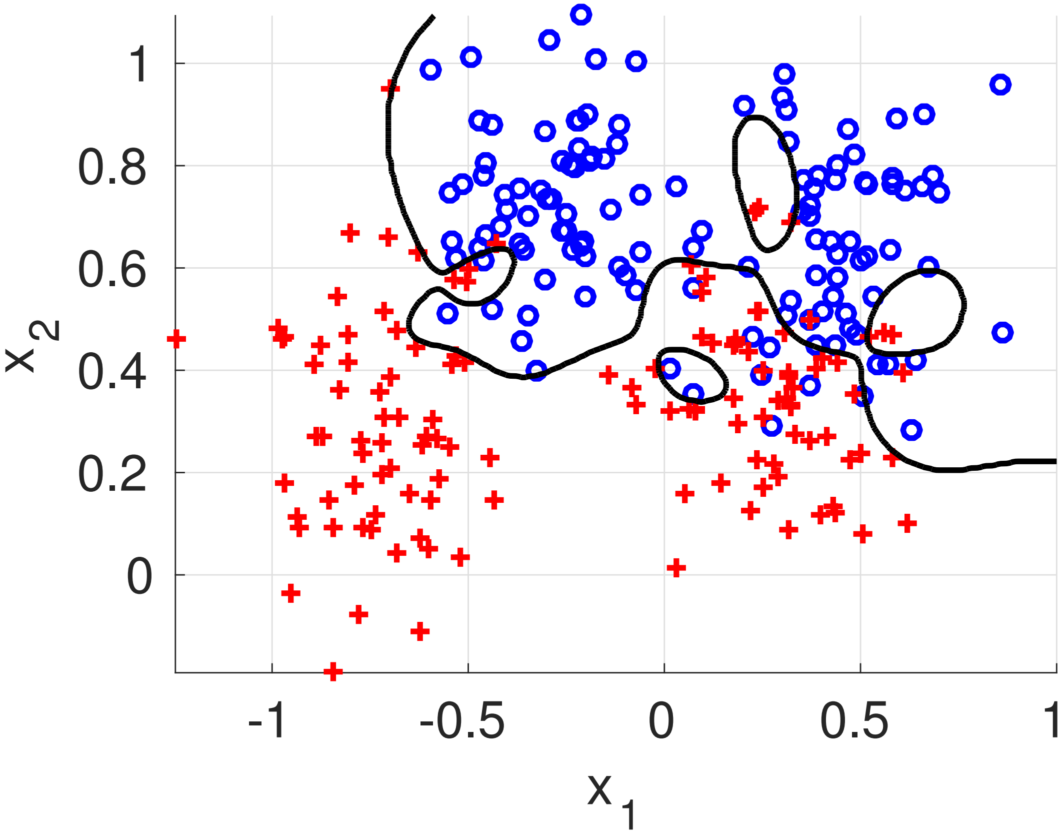


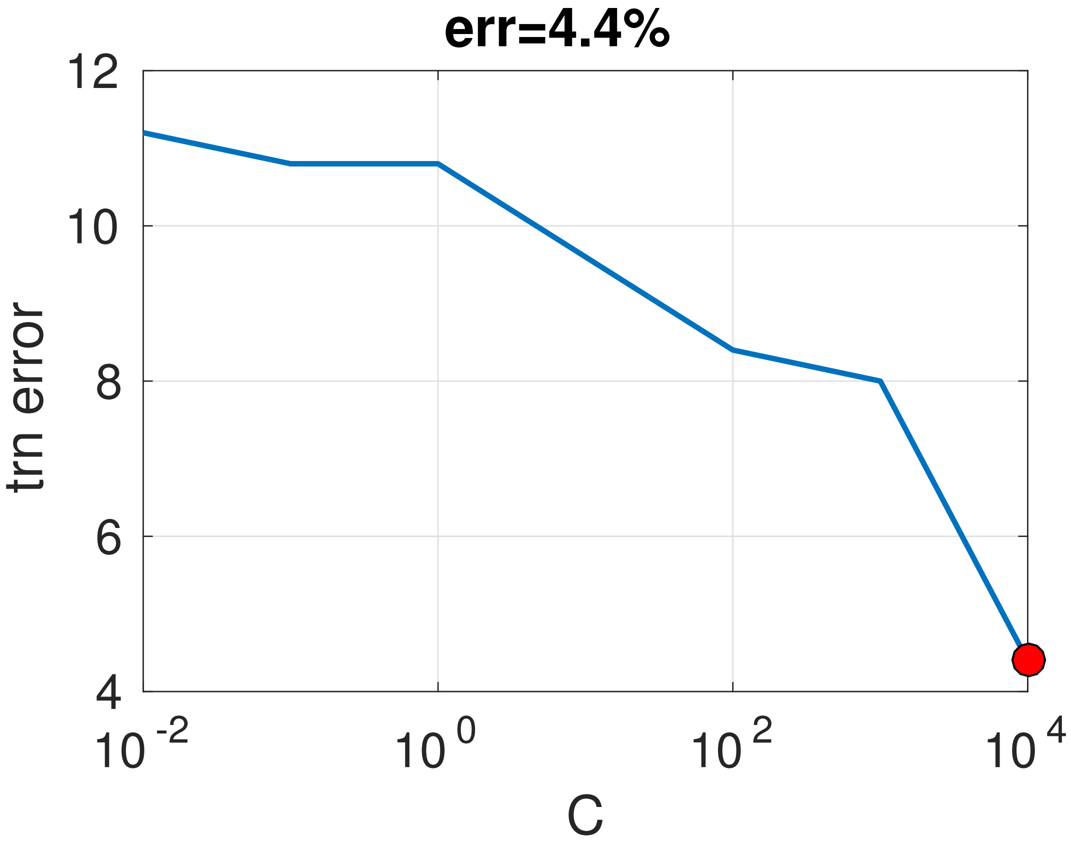


Example: SVM with RBF kernel training error vs. C

$$f(\boldsymbol{x}) = \sum_{i\in\mathcal{I}_{\mathrm{SV}}} y^i\,\alpha_i\,\exp(-\gamma\|\boldsymbol{x}-\boldsymbol{x}^i\|^2)+b$$

$$\gamma = 10,\; C = 10000$$





String Subsequence kernel

- Input space $\mathcal{X} = \cup_{d=1}^{\infty} \Sigma^d$ contains all strings on a finite alphabet Σ
- The features: occurrence+continuity of sub-sequences of length q:

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

  where $s[\boldsymbol{i}]$ denotes a sub-sequence of $\boldsymbol{s}$ and $\lambda \in (0,1]$ is a decay factor.

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

	ca	ct	at	ba	bt	cr	ar	br
$\boldsymbol{\phi}(''cat'')$	λ^2	λ^3	λ^2	0	0	0	0	0
$\boldsymbol{\phi}(''car'')$	λ^2	0	0	0	0	λ^3	λ^2	0
$\boldsymbol{\phi}(''bat'')$	0	0	λ^2	λ^2	λ^3	0	0	0
$\boldsymbol{\phi}(''bar'')$	0	0	0	λ^2	0	0	λ^2	λ^3

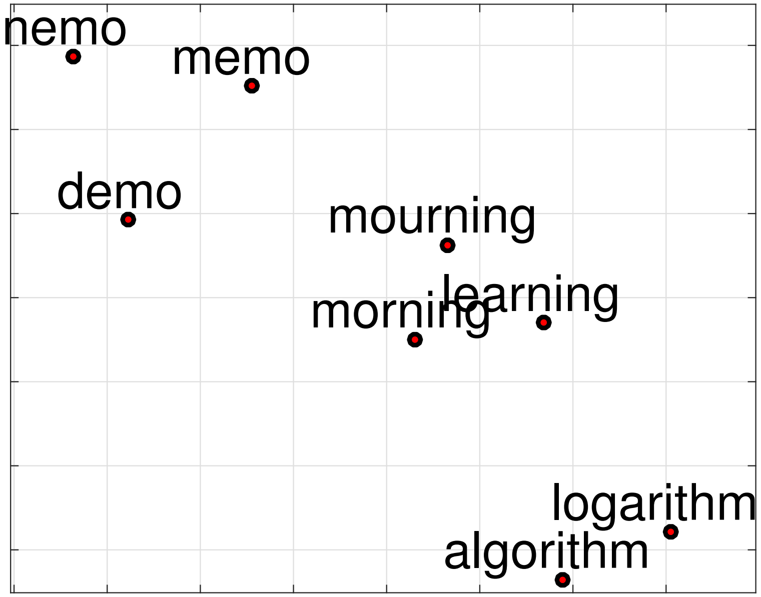
  $k(''cat'',''car'') = \lambda^4$, $k(''cat'',''bat'') = \lambda^4$, $k(''cat'',''bar'') = 0$, ...

- The kernel value $k(s,t)$ can be computed by dynamic programming in time $\mathcal{O}(q\,|s|\,|t|)$.

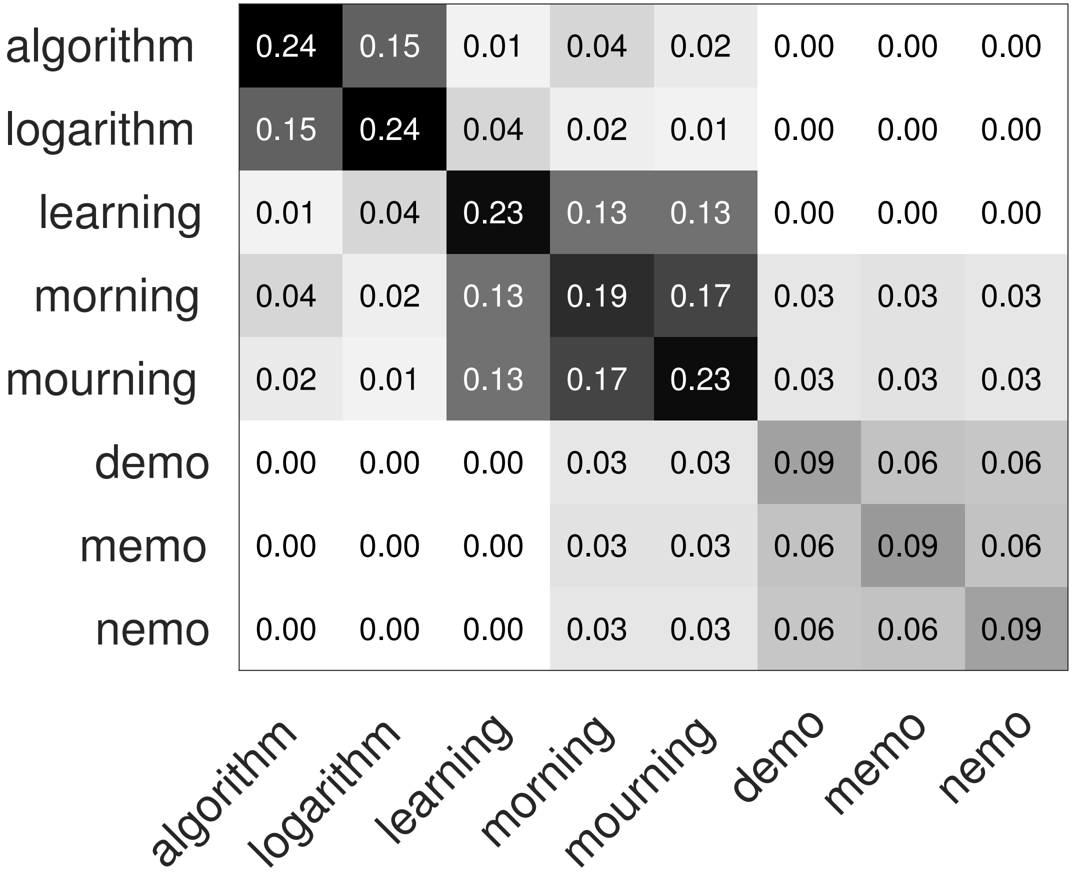
Example: String Subsequence kernel

kernel parameters: $q = 2,\ \lambda = 0.4$

Feature space
(2D embedding via tSNE)



Kernel matrix



	algorithm	logarithm	learning	morning	mourning	demo	memo	nemo
algorithm	0.24	0.15	0.01	0.04	0.02	0.00	0.00	0.00
logarithm	0.15	0.24	0.04	0.02	0.01	0.00	0.00	0.00
learning	0.01	0.04	0.23	0.13	0.13	0.00	0.00	0.00
morning	0.04	0.02	0.13	0.19	0.17	0.03	0.03	0.03
mourning	0.02	0.01	0.13	0.17	0.23	0.03	0.03	0.03
demo	0.00	0.00	0.00	0.03	0.03	0.09	0.06	0.06
memo	0.00	0.00	0.00	0.03	0.03	0.06	0.09	0.06
nemo	0.00	0.00	0.00	0.03	0.03	0.06	0.06	0.09

Primal vs. kernel SVM in terms of memory

ϕ: $\mathcal{X} \to \mathbb{R}^n$.. feature map; m .. number of training examples; $\mathcal{O}(d)$ .. size of the observation $x \in \mathcal{X}$ in bytes

Memory requirements: (assuming $x \in \mathcal{X}$ requires $\mathcal{O}(d)$ bytes)

	training set $\mathcal{T}^m$	prediction rule
primal formulation	$\mathcal{O}(m \cdot n)$	$\mathcal{O}(n)$
kernel formulation	$\mathcal{O}(m^2)$	$\mathcal{O}(d \cdot \lvert\mathcal{I}_{\mathrm{SV}}\rvert)$

Examples of kernel functions:

	Input space $\mathcal{X}$	$k(\boldsymbol{x}, \boldsymbol{x}')$	dim of $\lvert\boldsymbol{\Phi}(x)\rvert$
2nd polynomial	$\mathbb{R}^d$	$\langle \boldsymbol{x} \, , \boldsymbol{x}' \rangle^2$	$n = \frac{d(d+1)}{2}$
RBF	$\mathbb{R}^d$	$\exp\Big(-\gamma \, \lVert\boldsymbol{x} - \boldsymbol{x}'\rVert^2\Big)$	$n = \infty$
string sub-sequence	$\cup_{d=0}^{\infty} \Sigma^d$	dynamic programming	$n = \lvert\Sigma\rvert^q$

Positive definite kernel

Definition 1. *Let $\mathcal{X}$ be a non-empty set. The function $k \colon \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is a positive definite kernel if it is symmetric and for any finite set of inputs $x^1, \ldots, x^m$, 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 $\{x^1, \ldots, x^m\}$:

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

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

Every kernel defines a feature space

Theorem 1. *For every positive definite kernel $k\colon \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ there exists a Hilbert space $\mathcal{H}$ and a feature map $\phi\colon \mathcal{X} \to \mathcal{H}$ such that* $k(x,x') = \langle \boldsymbol{\phi}(x), \boldsymbol{\phi}(x') \rangle$.

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 $\mathbf{D} = \mathrm{diag}(\lambda_1, \ldots, \lambda_{|\mathcal{X}|})$ is diagonal matrix of non-negative eigenvalues.

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

Construction of valid kernels

Let $k_1 \colon \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ and $k_2 \colon \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ be p.d. kernels, $\lambda \in [0,1]$, $\alpha \geq 0$, $\boldsymbol{\phi} \colon \mathcal{X} \to \mathbb{R}^n$ and $k_3 \colon \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ a p.d. kernel, $\mathbf{K}$ a symmetric positive definite matrix. Then the following functions are also p.d. kernels:

$$\begin{aligned}
k(x,z) &= (1-\lambda)\, k_1(x,z) + \lambda\, k_2(x,z) \\
k(x,z) &= \alpha\, k_1(x,z) \\
k(x,z) &= k_1(x,z)\, k_2(x,z) \\
k(x,z) &= k_3(\boldsymbol{\phi}(x), \boldsymbol{\phi}(z)) \\
k(\boldsymbol{x},\boldsymbol{z}) &= \boldsymbol{x}^T \mathbf{K} \boldsymbol{z}
\end{aligned}$$