# L14: 06.07.2022

## 4.5.4.4 Kernel trick

LDF → Linear decision boundary!

Feature mapping (4.5.2):

- $\underline{z} = \underline{\phi}(\underline{x}) \quad : \underline{x} \in \mathbb{R}^d \longrightarrow \underline{z} \in \mathbb{R}^{d'} \quad d' >> d$

- LDF $\underline{w}^T \underline{z} + w_0$, linear decision boundary in $\mathbb{R}^{d'}$

  $\hat{=} \; \underline{w}^T \underline{\phi}(\underline{x}) + w_0$, non-linear decision boundary in $\mathbb{R}^d$

But, often $d' >> d$:

- higher complexity $\xrightarrow{\text{kernel trick}}$ no problem
- overfitting $\xrightarrow{\text{VC theory}}$ no problem if max margin (see formula)

Observation:

Training of SVM: $\underline{Q} = [q_{mn}] \quad N \times N$

$$q_{mn} = y_m y_n \underline{x}_m^T \underline{x}_n$$

$$\rightarrow \underline{\alpha} \rightarrow \underline{w}, w_0$$

classification of SVM: $f(\underline{x}) = \underline{w}^T \underline{x} + w_0$

$$= \sum_{n \in SV} \alpha_n y_n \underline{x}_n^T \underline{x} + w_0$$

⇒ need only scalar products $\underline{x}_m^T \underline{x}_n$, $\underline{x}_n^T \underline{x}$

After feature mapping, $q_{mn} = y_m y_n \underline{\phi}(\underline{x}_m)^T \underline{\phi}(\underline{x}_n)$

$$f(\underline{x}) = \sum_{n \in SV} \alpha_n y_n \underline{\phi}(\underline{x}_n)^T \underline{\phi}(\underline{x})$$

## Kernel trick

choose $\underline{\phi}(x)$ such that $\underline{\phi}(\underline{x}_m)^T \underline{\phi}(\underline{x}_n) = K(\underline{x}_m, \underline{x}_n)$

i.e. the Kernel function $K(\underline{x}_m, \underline{x}_n)$ can be calculated directly with $\underline{x}_m$, $\underline{x}_n$ in $\mathbb{R}^d$.

$\Rightarrow$ No explicite calculation of $\underline{\phi}(\underline{x}_m)$, $\underline{\phi}(\underline{x}_n)$

E 4.17 : Kernel function

$$\underline{x} = [x_1, x_2]^T \in \mathbb{R}^2$$

Quadratic feature mapping

$$\underline{z} = \phi(\underline{x}) = [1, \sqrt{2}\, x_1, \sqrt{2}\, x_2, \sqrt{2}\, x_1 x_2, x_1^2, x_2^2] \in \mathbb{R}^6$$

$$\underline{\phi}(\underline{x})^T \underline{\phi}(\underline{y}) = 1 + 2x_1 y_1 + 2x_2 y_2 + 2x_1 x_2 y_1 y_2 + x_1^2 y_1^2 + x_2^2 y_2^2$$

$$= (1 + x_1 y_1 + x_2 y_2)^2 = \underbrace{(1 + \underline{x}^T \underline{y})^2}_{\mathbb{R}^2} = K(\underline{x}, \underline{y})$$

4-80

4-81

4-82

4-83

4-84

# 4.5.4.5 Soft margin SVM

Hard margin SVM: only for linearly separable datasets

If non-linearly separable, even after feature mapping:

$R_F = \phi$, no solution with training error rate $= 0$

Non-linearly separable due to outliers
↳ very untypical samples.

- overlapping classes
- large noise


## Soft margin SVM

- outliers allowed
- a reasonable solution for non-linearly separable data with training error rate $> 0$

Idea: max margin

s.t. $y_n(\underline{\omega}^T \underline{z}_n + w_0) \geq 1 - \xi_n \qquad \xi_n \geq 0 \quad \forall n$

$\xi_n$ : Slack variable $\triangleq$ degree of outliers

Hard margin SVM:

(a) $y_n(\underline{\omega}^T \underline{z}_n + w_0) > 1 - \xi_n \qquad \xi_n = 0$ : non-support vector

(b) $= 1 - \xi_n \qquad \xi = 0$ : support vector

Allowed outliers

(c) $\geq 1 - \xi_n \quad 0 < \xi_n < 1$ : between $\beta_1$, $\beta$

(d) $\geq 1 - \xi_n \qquad \xi_n = 1$ : on $\beta$

(e) $\geq 1 - \xi_n \qquad \xi_n > 1$ : wrong $\hat{y}$

Wishes:

- max margin or min $\frac{1}{2}\|\underline{w}\|^2$
- minimize $\sum_{n=1}^{N} \xi_n = \|\underline{\xi}\|_1 = \underline{1}^T\underline{\xi}$, $\underline{\xi} = [\xi_1, \dots \xi_N]^T \in \mathbb{R}^N$

$\Rightarrow$ multiobjective optimization $\Rightarrow$ compromise

Primal problem:

$$\min_{\underline{w}, w_0, \underline{\xi}} \frac{1}{2}\|\underline{w}\|^2 + C\|\underline{\xi}\|_1 \qquad C>0, \text{ hyperparameter}$$

If $C$ is large then emphasize more on reducing $\|\underline{\xi}\|$

" $C$ is small " " " " $\|\underline{w}\|^2$

s.t. • $y_n(\underline{w}^T \underline{x}_n + w_0) \geq 1 - \xi_n \quad \forall n$

• $\xi_n \geq 0 \quad \forall n$

$\left.\right\}$ $2N$ inequality constraints

Again, it is a convex QP.

20:33

4-85

⋮

4-88

⋮

4-91

Toolbox: libsvm for MATLAB

## 4.5.4.6 Multiclass SVM

Upto now: only $C = 2$ classes

multiclass SVM $(c > 2)$:

Combine results of many binary SVM classification

(A) One-against-one : Bundesliga

- $\frac{c(c-1)}{2}$ SVMs for all pairs of classes

  $C = 3$: $\omega_1$ vs $\omega_2$, $\omega_1$ vs $\omega_3$, $\omega_2$ vs $\omega_3$

- class with the most wins, wins

Disadv: High computational complexity $O(c^2)$

(B) One-vs-Rest:

- $C$ SVMs

  $\omega_1$ vs rest $(\omega_2 \cdots \omega_c)$, $\hat{\omega} = \omega_1$ if $f_1(\underline{x}) > 0$

  $\omega_2$ vs rest, $\hat{\omega} = \omega_2$ if $f_2(\underline{x}) > 0$

  $\vdots$

  $\omega_c$ vs rest, $\hat{\omega} = \omega_c$ if $f_c(\underline{x}) > 0$

- $\max_i f_i(\underline{x})$