

Pattern Recognition XII

SVM

Michal Haindl

Faculty of Information Technology, KTI
Czech Technical University in Prague
Institute of Information Theory and Automation
Czech Academy of Sciences
Prague, Czechia



<http://ro.utia.cz/>



MI/MIE-ROZ 2023-2024

© Haindl

MI-ROZ - 12

1/17

Outline

- 1 Support Vector Machine
 - SVM Derivation
 - SVM Classification
 - Multiclass SVM
 - Nonlinear SVM
- 2 Classifier Combination



© Haindl

MI-ROZ - 12

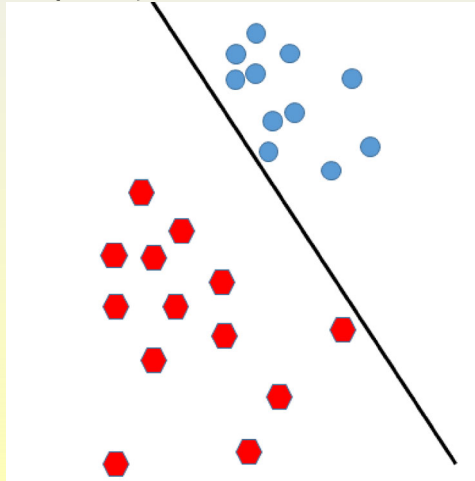
2/17

SVM MCS

SVMd SVMc mSVM nSVM

Support Vector Machine

V. Vapnik, A. Chervonenkis - (Generalised Portrait Method 1964,
Theory of Pattern Recognition, Nauka, Moscow, 1974)
widely accepted 1990s - kernel trick & successful text classification



$$g(X) = W^T X + b$$



© Haindl

MI-ROZ - 12

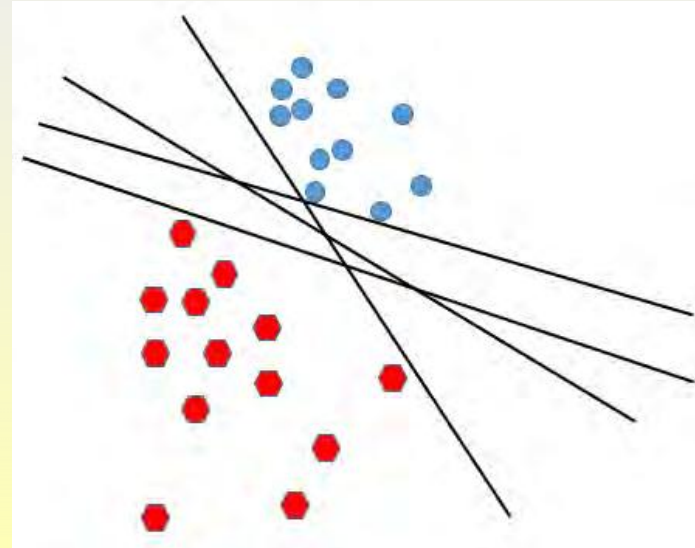
3/17

SVM MCS

SVMd SVMc mSVM nSVM

Support Vector Machine

V. Vapnik, A. Chervonenkis - (Generalised Portrait Method 1964,
Theory of Pattern Recognition, Nauka, Moscow, 1974)



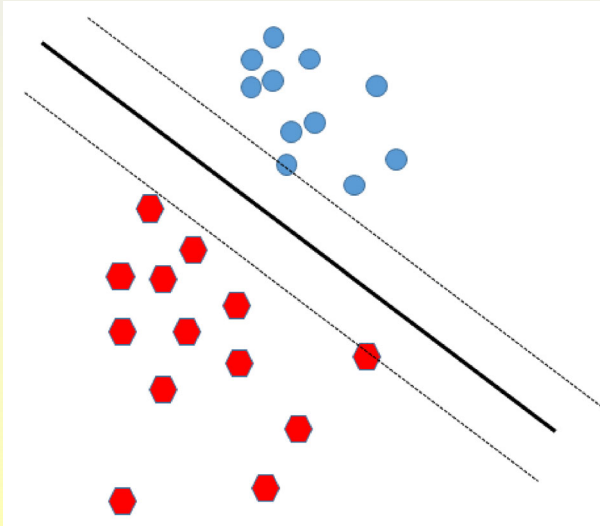
© Haindl

MI-ROZ - 12

3/17

Support Vector Machine

V. Vapnik, A. Chervonenkis - (Generalised Portrait Method 1964, Theory of Pattern Recognition, Nauka, Moscow, 1974)



Support Vector Machine

V. Vapnik, A. Chervonenkis - (Generalised Portrait Method 1964, Theory of Pattern Recognition, Nauka, Moscow, 1974)

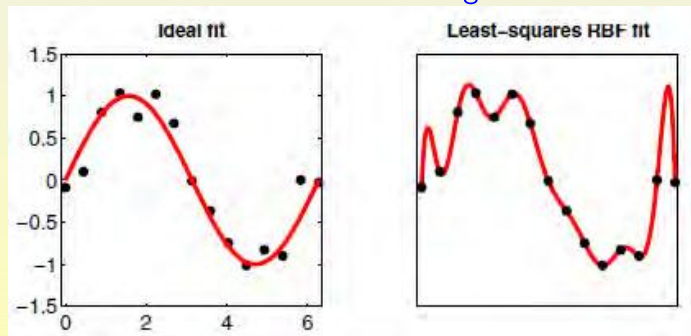
SVM idea (the bias variance tradeoff, capacity control, overfitting)

- optimal balance between accuracy on a given trainee set and the ability of a system to learn any trainee set without error
- feature vector mapping to a high dimensional space where the best separating hyperplane (maximal margin hyperplane) is constructing



SVM Properties

- class separation - **hyperplanes**
- hyperplanes placed to maximize distance to the class closest element
- aims to reduce the risk of **overfitting**



SVM Properties

- ⊕ SVMs aim to reduce the risk of overfitting
- ⊕ SVMs aim to learn class boundaries of the hidden underlying population rather than training set boundaries
- ⊕ SVMs generalize well on unseen data (no theoretical proof)
- ⊖ SVMs designed for 2 classes, direct extension to multiclass classification is not attractive due to the complexity of the associated optimization task
- ⊖ SVMs learning time proportional to the size of training set
- ⊖ large memory required to store the kernel matrix, directly proportional to the size of training set



SVM Properties

- ⊕ SVMs aim to reduce the risk of overfitting
- ⊕ SVMs aim to learn class boundaries of the hidden underlying population rather than training set boundaries
- ⊕ SVMs generalize well on unseen data (no theoretical proof)
- ⊖ SVMs designed for 2 classes, direct extension to multiclass classification is not attractive due to the complexity of the associated optimization task
- ⊖ SVMs learning time proportional to the size of training set
- ⊖ large memory required to store the kernel matrix, directly proportional to the size of training set



SVM Properties

- ⊕ SVMs aim to reduce the risk of overfitting
- ⊕ SVMs aim to learn class boundaries of the hidden underlying population rather than training set boundaries
- ⊕ SVMs generalize well on unseen data (no theoretical proof)
- ⊖ SVMs designed for 2 classes, direct extension to multiclass classification is not attractive due to the complexity of the associated optimization task
- ⊖ SVMs learning time proportional to the size of training set
- ⊖ large memory required to store the kernel matrix, directly proportional to the size of training set



SVM Properties

- ⊕ SVMs aim to reduce the risk of overfitting
- ⊕ SVMs aim to learn class boundaries of the hidden underlying population rather than training set boundaries
- ⊕ SVMs generalize well on unseen data (no theoretical proof)
- ⊖ SVMs designed for 2 classes, direct extension to multiclass classification is not attractive due to the complexity of the associated optimization task
- ⊖ SVMs learning time proportional to the size of training set
- ⊖ large memory required to store the kernel matrix, directly proportional to the size of training set



SVM Properties

- ⊕ SVMs aim to reduce the risk of overfitting
- ⊕ SVMs aim to learn class boundaries of the hidden underlying population rather than training set boundaries
- ⊕ SVMs generalize well on unseen data (no theoretical proof)
- ⊖ SVMs designed for 2 classes, direct extension to multiclass classification is not attractive due to the complexity of the associated optimization task
- ⊖ SVMs learning time proportional to the size of training set
- ⊖ large memory required to store the kernel matrix, directly proportional to the size of training set



SVM Properties

- ⊕ SVMs aim to reduce the risk of overfitting
- ⊕ SVMs aim to learn class boundaries of the hidden underlying population rather than training set boundaries
- ⊕ SVMs generalize well on unseen data (no theoretical proof)
- ⊖ SVMs designed for 2 classes, direct extension to multiclass classification is not attractive due to the complexity of the associated optimization task
- ⊖ SVMs learning time proportional to the size of training set
- ⊖ large memory required to store the kernel matrix, directly proportional to the size of training set



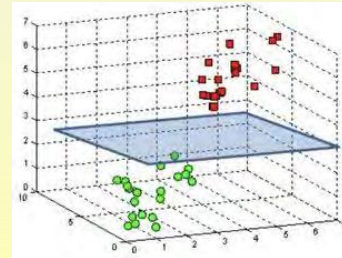
SVM Derivation

$$\omega_1 = 1 \quad \omega_2 = -1$$

$$\text{decision function: } g(X) = \langle \tilde{W}, X \rangle + \tilde{b}$$

assign X into ω_i iff:

$$g(X) \begin{cases} \geq 0 & X \rightarrow \omega_1 \\ < 0 & X \rightarrow \omega_2 \end{cases}$$



SVM Derivation

$$\omega_i (\tilde{W}^T X_i + \tilde{b}) > 0 \quad \forall i \quad \text{correctly classified features}$$

$$i = 1, \dots, n \quad n \text{ number of training patterns } X_i$$



SVM Derivation

$$\omega_i (\tilde{W}^T X_i + \tilde{b}) > 0 \quad \forall i \quad \text{correctly classified features}$$

margin c - points required at distance $> \frac{c}{|\tilde{W}|}$ from the hyperplane

$$\omega_i (\tilde{W}^T X_i + \tilde{b}) \geq c \quad \forall i \quad \rightarrow \quad \omega_i (W^T X_i + b) \geq 1 \quad \forall i$$

canonical hyperplanes

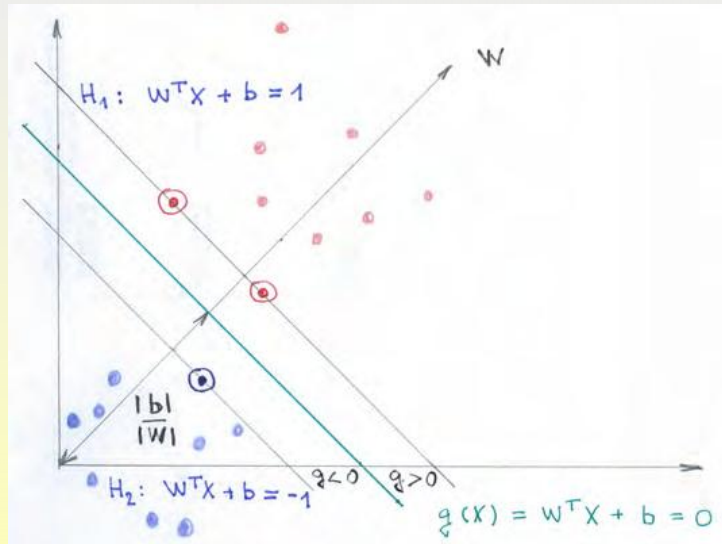
$$H_1 : W^T X_i + b = 1 \quad H_2 : W^T X_i + b = -1$$

points on H_i support vectors,

$$\frac{1}{|W|} \text{ margin (distance between } H_i \text{ and } g(X) = 0$$



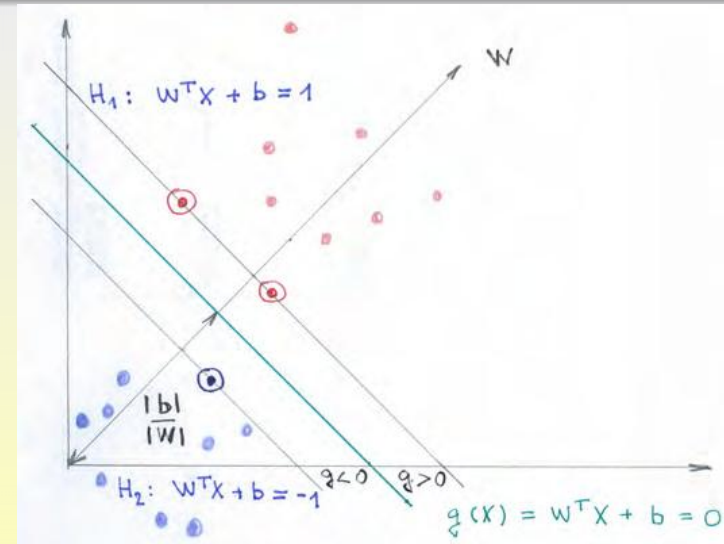
SVM Derivation



SV



SVM Derivation



SV

support vectors - the most informative points in the data set, other points do not effect the separating hyperplane solution



SVM Derivation

Maximize the margins (= min $|W|$ subject to the constraints C_i):

$$\min J(W) = \min \left\{ \frac{1}{2} W^T W \right\}$$

$$C_i: \quad \omega_i (W^T X_i + b) \geq 1 \quad i = 1, \dots, n$$

$$L_p = \frac{1}{2} W^T W - \sum_{i=1}^n \lambda_i (\omega_i (W^T X_i + b) - 1)$$

$$\nabla_{W, b, \lambda_1, \dots, \lambda_n} L_p = 0$$



SVM Derivation

Maximize the margins (= min $|W|$ subject to the constraints C_i):

$$\min J(W) = \min \left\{ \frac{1}{2} W^T W \right\}$$

$$C_i: \quad \omega_i (W^T X_i + b) \geq 1 \quad i = 1, \dots, n$$

$$L_p = \frac{1}{2} W^T W - \sum_{i=1}^n \lambda_i (\omega_i (W^T X_i + b) - 1)$$

$$\frac{\partial L_p}{\partial W} = W - \sum_{i=1}^n \lambda_i \omega_i X_i = 0$$

$$W = \sum_{i=1}^n \lambda_i \omega_i X_i$$

$$\frac{\partial L_p}{\partial b} = - \sum_{i=1}^n \lambda_i \omega_i = 0$$



SVM Derivation

Lagrangian (dual form):

$$L_D = \sum_{i=1}^n \lambda_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \lambda_i \lambda_j \omega_i \omega_j X_i^T X_j \quad \text{subject to}$$

$$\lambda_i \geq 0 \quad \sum_{i=1}^n \lambda_i \omega_i = 0$$

$$\lambda_i > 0 \quad \text{if } X_i \in \{SV\} \quad \lambda_i = 0 \quad \text{if } X_i \notin \{SV\}$$



SVM Derivation

1 Maximize L_D with respect to the λ_i subject to

$$\lambda_i > 0 \quad \sum_{i=1}^n \lambda_i \omega_i = 0$$

2 b from

- $\lambda_i (\omega_i (W^T X_i + b) - 1) = 0$ (Karush-Kuhn-Tucker complementarity condition)
- $n_{SV} b + W^T \sum_{i \in SV} X_i = \sum_{i \in SV} \omega_i$ (average over all support vectors SV)

3 $W = \sum_{i \in SV} \lambda_i \omega_i X_i$

since $\lambda_i = 0$ for non support vectors

Karush-Kuhn-Tucker conditions are necessary and sufficient for the optimal solution (non-linear programming & convex problem).



SVM Properties II

1 $J(W) = \frac{1}{2} W^T W$ strict convex (Hessian matrix PD)

2 inequality constraints linear

$\Rightarrow$ local minimum = global minimum & unique $\Rightarrow$ unique classifier



SVM Properties II

1 $J(W) = \frac{1}{2} W^T W$ strict convex (Hessian matrix PD)

2 inequality constraints linear

$\Rightarrow$ local minimum = global minimum & unique $\Rightarrow$ unique classifier



SVM Classification

assign X into ω_1 iff:

$$W^T X + b > 0$$



SVM Classification

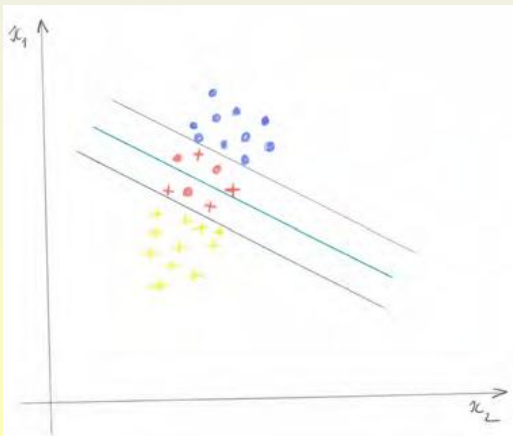
assign X into ω_1 iff:

$$\sum_{i \in SV} \lambda_i \omega_i X_i^T X + \frac{1}{n_{SV}} \sum_{i \in SV} \omega_i - \frac{1}{n_{SV}} \sum_{i \in SV} \sum_{j \in SV} \lambda_i \omega_i X_i^T X_j > 0$$

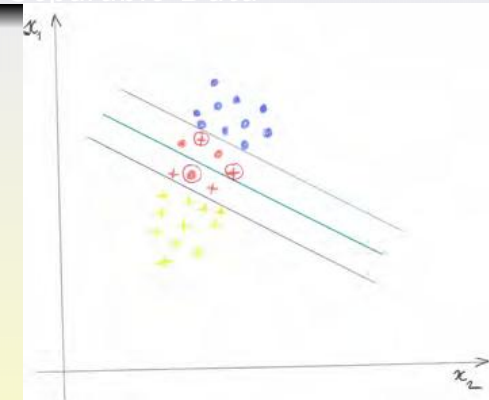


Linearly Non-Separable Data

transformation into high-dimensional feature space
 $\phi(X) \rightarrow$ linearly separable but over-fitting



Linearly Non-Separable Data



- + $\omega_i(W^T X_i + b) \geq 1$ vectors outside the band & correctly classified
- + $0 \leq \omega_i(W^T X_i + b) < 1$ vectors inside the band & correctly classified
- ⊕ $\omega_i(W^T X_i + b) < 0$ vectors misclassified



Linearly Non-Separable Data

constrains relaxing $\xi_i \geq 0$

$$\begin{aligned} W^T X_i + b &\geq +1 - \xi_i \\ W^T X_i + b &\leq -1 + \xi_i \end{aligned}$$

$$\min J(W, b, \xi) = \min \left\{ \frac{1}{2} W^T W + \tau \sum_i \xi_i \right\} \text{ subject to } \uparrow$$

- 1 margin as large as possible
- 2 min number of points with $\xi > 0$

- + $\xi = 0$
- + $0 < \xi \leq 1$
- ⊕ $\xi > 1$



Linearly Non-Separable Data

$$L_p = \frac{1}{2} W^T W + \tau \sum_i \xi_i - \sum_{i=1}^n \lambda_i \left(\omega_i (W^T X_i + b) - 1 + \xi_i \right) - \sum_i r_i \xi_i$$

λ_i, r_i Lagrange multipliers, r_i ensures ξ_i (slack variable) positivity,
 τ free regularization parameter

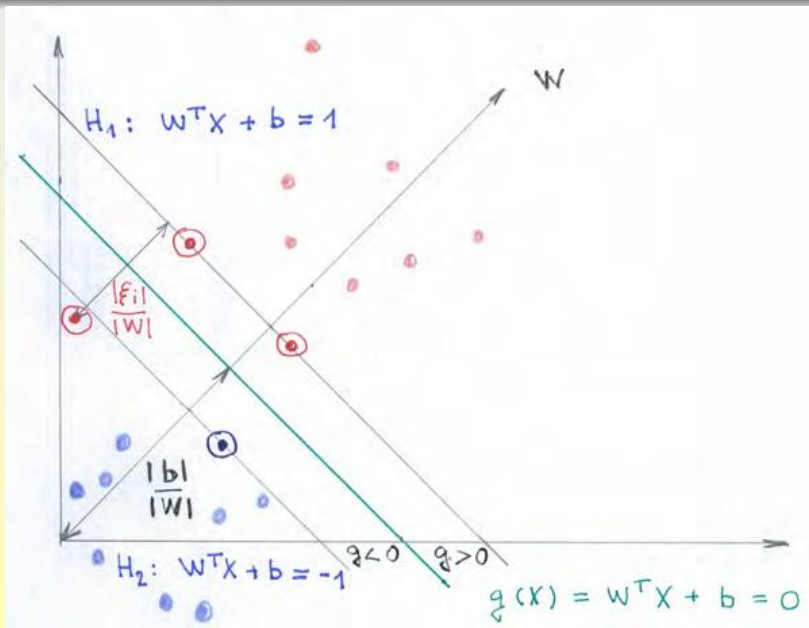
τ small allows constraints to be easily ignored $\rightarrow$ large margin

τ large constraints can be hardly ignored $\rightarrow$ narrow margin

$\tau = \infty$ enforces all constraints $\rightarrow$ hard margin



Linearly Non-Separable Data



Linearly Non-Separable Data

Karush-Kuhn-Tucker conditions:

$$\frac{\partial L_p}{\partial W} = W - \sum_{i=1}^n \lambda_i \omega_i X_i = 0$$

$$\frac{\partial L_p}{\partial b} = - \sum_{i=1}^n \lambda_i \omega_i = 0$$

$$\frac{\partial L_p}{\partial \xi_i} = \tau - \lambda_i - r_i = 0$$

$$0 = \lambda_i \left(\omega_i (W^T X_i + b) - 1 + \xi_i \right)$$

$$0 = r_i \xi_i \quad \lambda_i \geq 0 \quad r_i \geq 0 \quad \xi_i \geq 0 \quad \forall i$$

$$L_D = \sum_{i=1}^n \lambda_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \lambda_i \lambda_j \omega_i \omega_j X_i^T X_j$$



Linearly Non-Separable Data

- 1 max L_D with respect to the λ_i subject to

$$0 \leq \lambda_i \leq \tau \quad \sum_{i=1}^n \lambda_i \omega_i = 0$$

- 2 b from Karush-Kuhn-Tucker complementarity conditions:

$$\begin{aligned} \lambda_i \left(\omega_i (W^T X_i + b) - 1 + \xi_i \right) &= 0 \\ r_i \xi_i = (\tau - \lambda_i) \xi_i &= 0 \end{aligned}$$

X_i with $\lambda_i > 0 \rightarrow$ support vectors

X_i with $0 < \lambda_i < \tau \Rightarrow \xi_i = 0$ lies on a canonical hyperplane at $\frac{1}{|W|}$ (margin SV)



Linearly Non-Separable Data

- 1 max L_D with respect to the λ_i subject to

$$0 \leq \lambda_i \leq \tau \quad \sum_{i=1}^n \lambda_i \omega_i = 0$$

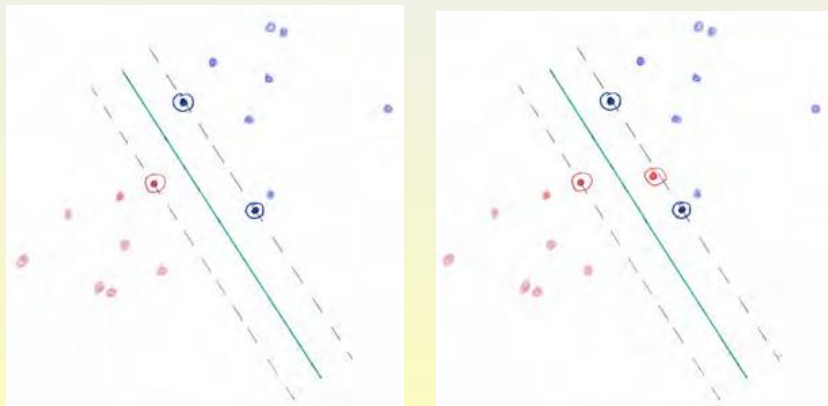
- 2 b from

$$b = \frac{1}{n_{SV}} \left\{ \sum_{i \in SV} \omega_i - \sum_{i \in SV, j \in SV} \lambda_i \omega_i X_i^T X_j \right\}$$

- 3 $W = \sum_{i \in SV} \lambda_i \omega_i X_i$



Linearly Non-Separable Data



linearly separable

linearly non-separable



Multiclass SVM

- dichotomic SVMs
 - K (one-against-all)
 - $\frac{K(K-1)}{2}$ (one-against-one) + majority voting
- K SVM discriminant functions



Multiclass SVM

assign X to ω_i iff:

$$g_i(X) = \max_j \{g_j(X)\}$$



Multiclass SVM

assign X to ω_i iff:

$$g_i(X) = \max_j \{g_j(X)\}$$

Solution: $\{({}^k W, {}^k b), k = 1, \dots, K\}$ such that

$$({}^k W^T X_i + {}^k b) - ({}^j W^T X_i + {}^j b) \geq 1 \quad \forall X \in \omega_k \wedge j \neq k$$

if possible (pairwise separable classes) $\min \sum_{k=1}^K {}^k W^T {}^k W$



Multiclass SVM

Solution: $\{({}^k W, {}^k b), k = 1, \dots, K\}$ such that

$$({}^k W^T X_i + {}^k b) - ({}^j W^T X_i + {}^j b) \geq 1 \quad \forall X \in \omega_k \wedge j \neq k$$

if possible (pairwise separable classes) $\min \sum_{k=1}^K {}^k W^T {}^k W$

no separable training data $\rightarrow$ + slack variables

minimize $L = \sum_{k=1}^K {}^k W^T {}^k W + K \sum_{i=1}^n \xi_i$ subject to

$$({}^k W^T X_i + {}^k b) - ({}^j W^T X_i + {}^j b) \geq 1 - \xi_i \quad \forall X_i \in \omega_k \wedge j \neq k$$



Nonlinear SVM

$$\omega_1 = 1 \quad \omega_2 = -1$$

decision function: $g(X) = W^T \phi(X) + b$

assign X into ω_i iff:

$$g(X) \begin{cases} \geq 0 & X \rightarrow \omega_1 \\ < 0 & X \rightarrow \omega_2 \end{cases}$$



Nonlinear SVM

$$\max L_D = \max \left\{ \sum_{i=1}^n \lambda_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \lambda_i \lambda_j \omega_i \omega_j \phi(X_i)^T \phi(X_j) \right\}$$

$$0 \leq \lambda_i \leq c \quad \sum_{i=1}^n \lambda_i \omega_i = 0$$

$$W = \sum_{i \in \mathcal{SV}} \lambda_i \omega_i \phi(X_i)$$



Nonlinear SVM

Classification:

$$g(X) = \sum_{i \in \mathcal{SV}} \lambda_i \omega_i \phi(X_i)^T \phi(X) + b$$

$$b = \frac{1}{n_{\mathcal{SV}}} \left\{ \sum_{i \in \mathcal{SV}} \omega_i - \sum_{i \in \mathcal{SV}, j \in \mathcal{SV}} \lambda_i \omega_i \phi(X_i)^T \phi(X_j) \right\}$$



SVM Kernels

$$K(X_i, X_j) = \phi(X_i)^T \phi(X_j) \longrightarrow L_D, g(X)$$

$$g(X) = \sum_{i \in \mathcal{SV}} \lambda_i \omega_i K(X_i, X) + b$$



SVM Kernels

$$K(X_i, X_j) = \phi(X_i)^T \phi(X_j) \longrightarrow L_D, g(X)$$

$$g(X) = \sum_{i \in \mathcal{SV}} \lambda_i \omega_i K(X_i, X) + b$$

Acceptable kernels must be expressible as an inner product in a feature space (Mercer's condition):

$$\int K(X_i, X) f(X_i) f(X) dX_i dX \geq 0$$

$$K(X_i, X) = K(X, X_i)$$

$\forall f$ such $\int f^2(X) dX < \infty$, i.e., $K(X_i, X)$ may be expanded as

$$K(X_i, X) = \sum_{j=1}^{\infty} \nu_j \hat{\phi}_j(X_i) \hat{\phi}_j(X)$$

ν_j eigenvalues, $\hat{\phi}_j$ eigenfunctions



SVM Kernels

linear $K(X_i, X_j) = X_i^T X_j$
 $\oplus$ for large training set

polynomial (homogeneous) $K(X_i, X_j) = (\gamma X_i^T X_j)^d$
 polynomial (inhomogeneous) $K(X_i, X_j) = (\gamma X_i^T X_j + 1)^d$
 $\ominus K(X_i, X_j) \rightarrow \infty$ or $K(X_i, X_j) \rightarrow 0$

Radial Basis Function $K(X_i, X_j) = \exp(-\gamma \|X_i^T X_j\|^2)$

Gaussian RBF $K(X_i, X_j) = \exp(-\frac{\|X_i^T X_j\|^2}{2\sigma^2})$
 $\oplus 0 < K(X_i, X_j) \leq 1$

sigmoid $K(X_i, X_j) = \tanh(\gamma X_i^T X_j + c)$
 $\ominus K(X_i, X_j)$ not valid for some parameters

$\gamma > 0, c < 0, d$ kernel parameters



SVM Kernels - Solution

- 1 $K(X_i, X_j)$ & its parameters selection
- 2 τ selection (a compromise between trainee set error minimalization and maximum tolerance range)
- 3 L_D optimization (quadratic programming - interior point / active set / simplex algorithm extension, etc.)
- 4 b from the SVs
- 5 new X_j classification



SVM Kernels

- $\ominus$ difficult good kernel selection, incorrect kernel $\uparrow$ error
- $\ominus$ high algorithm complexity and extensive memory requirements



SVM Learning Interpretation

empirical risk minimization
 to minimize the number of training patterns X_i, ω_i where
 $\omega_i f(X_i) < 0$ (wrong decision)



SVM Learning Interpretation

empirical risk minimization
to minimize the number of training patterns X_i, ω_i where
 $\omega_i f(X_i) < 0$ (wrong decision)

SVM $\rightarrow$ empirical risk minimization + confidence on a hyperplane
structural risk minimization



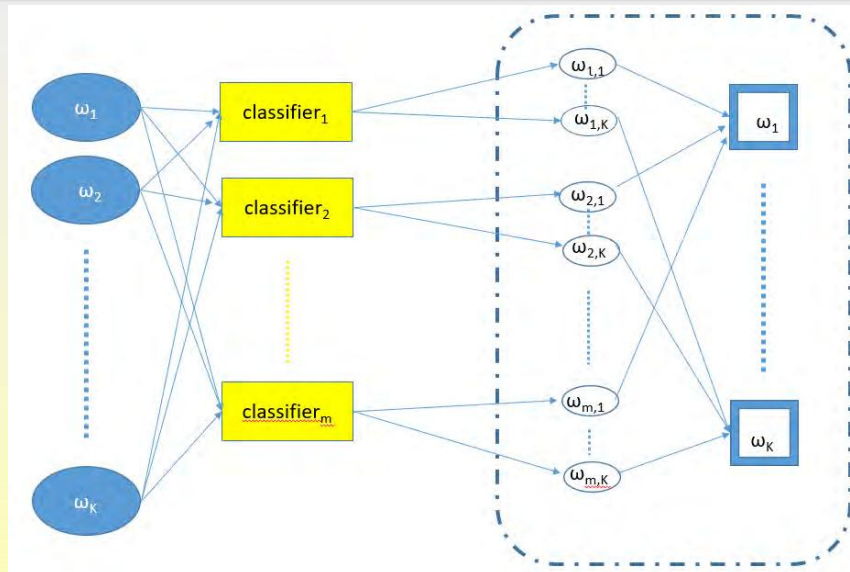
SVM Learning Interpretation

empirical risk minimization
to minimize the number of training patterns X_i, ω_i where
 $\omega_i f(X_i) < 0$ (wrong decision)

SVM $\rightarrow$ empirical risk minimization + confidence on a hyperplane
structural risk minimization
quantifies both **underfitting** (oversimplified model) and **overfitting** (excessively complex model) degree



Classifier Combination



Classifier Combination

classifier types

- type 1 output $\exists! \omega_i$
e.g. SVM classifier
- type 2 ranked list output $(\omega_{j,1}, \dots, \omega_{j,K})$ for each $g_{j,\bullet}(X)$
where $\omega_{j,1}$ has the highest ($\omega_{j,K}$ lowest) plausibility
e.g. multiclass SVM classifier
- type 3 output $\forall j^{th}$ classifier
 $g_{j,\bullet}(X) \rightarrow [p_j(\omega_1), \dots, p_j(\omega_k)]$
e.g. Bayesian classifier



Classifier Combination

classifiers $g_{1,\bullet}, \dots, g_{m,\bullet}$ combination

- voting
- Borda count
- Bayes combination



Voting

$$g_{j,i}(X) = \max_k \{g_{j,k}(X)\} \quad \forall j = 1, \dots, m$$

majority voting:

assign X into ω_i iff

$$\omega_i = \text{majority}\{\omega_{1,\cdot}, \dots, \omega_{m,\cdot}\}$$



Borda Count

$r_j(\omega_i)$ rank function ($\searrow$ position of the class ω_i for the j th classifier)

$$R(\omega_i) = \sum_{j=1}^m r_j(\omega_i) \quad \in \langle m; mK \rangle$$

Ramon Llull Blanquerna (novel - election of an abbes), 1283.

Nicholas of Cusa De Concordantia Catholica, 1433.

Jean-Charles de Borda Memoire sur les elections au scrutin. Histoire de l'Academie Royale des Sciences pour 1781 (Paris, 1784).



Borda Count

$r_j(\omega_i)$ rank function ($\searrow$ position of the class ω_i for the j th classifier)

$$R(\omega_i) = \sum_{j=1}^m r_j(\omega_i) \quad \in \langle m; mK \rangle$$

assign X into ω_i iff

$$\omega_i = \min\{R(\omega_1), \dots, R(\omega_K)\}$$



Borda Count

assign X into ω_i iff

$$\omega_i = \min\{R(\omega_1), \dots, R(\omega_K)\}$$

$$K = 5, \quad m = 3$$

$$r_1(\cdot) = [1, 5, 4, 2, 3]$$

$$r_2(\cdot) = [5, 1, 3, 2, 4]$$

$$r_3(\cdot) = [5, 3, 1, 4, 2]$$

$$R(\cdot) = [6, 13, 10, 12, 4] \Rightarrow \omega_5$$



Bayes Combination

assign X into ω_i iff

$$\tilde{P}_i = \max\{\tilde{P}_1, \dots, \tilde{P}_K\} \text{ where } \tilde{P}_i \text{ is a combined plausibility value}$$

combination strategies:

$$\tilde{P}_i = \max\{p_1(\omega_i), \dots, p_m(\omega_i)\}$$

$$\tilde{P}_i = \min\{p_1(\omega_i), \dots, p_m(\omega_i)\}$$

$$\tilde{P}_i = \frac{1}{m} \sum_{j=1}^m p_j(\omega_i)$$

$$\tilde{P}_i = \prod_{j=1}^m p_j(\omega_i)$$

