Data Analytics
EEE 4774 & 6777

Module 4 - Classification

Support Vector Machine (SVM)

Spring 2022

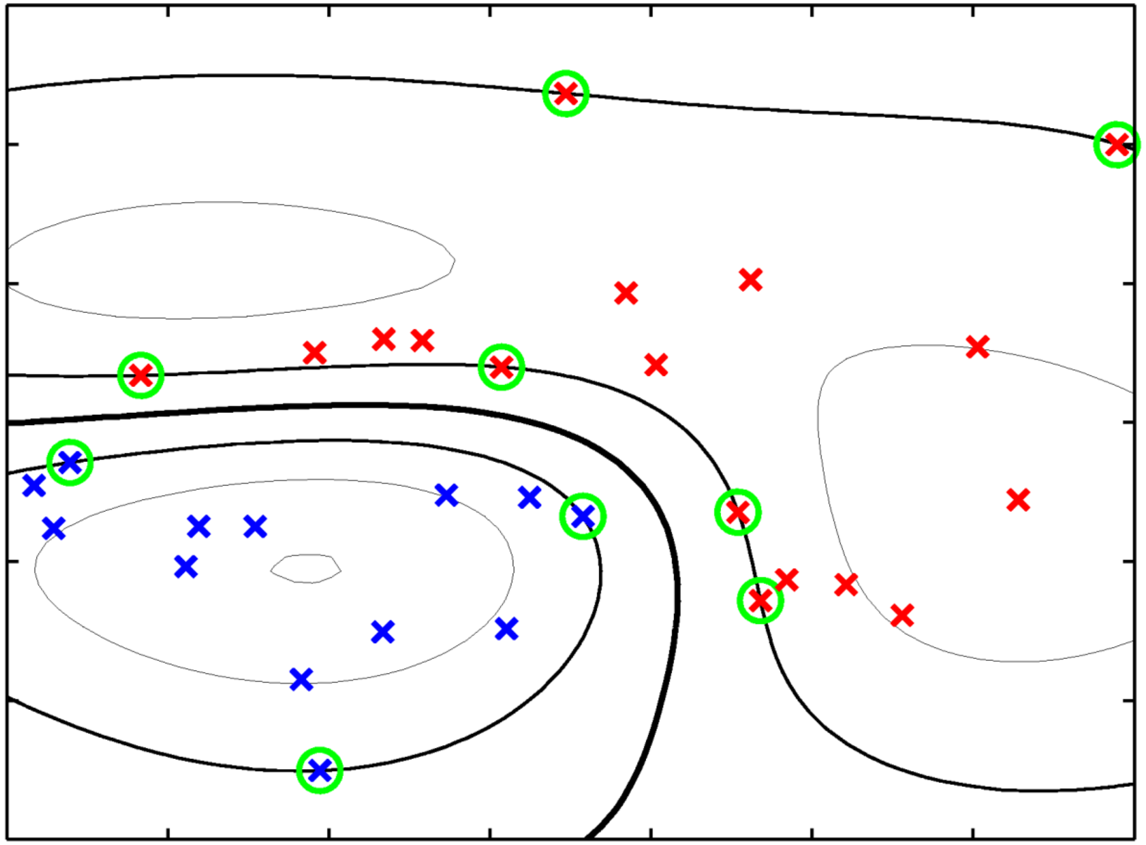
Support Vector Machine (SVM)

- Aims at maximizing the margin between decision boundary and data points

- Used for both classification and regression

- Determination of model parameters corresponds to a convex optimization problem, so any local solution is also a global optimum

- SVM makes extensive use of the Lagrange Multipliers concept from the Optimization Theory

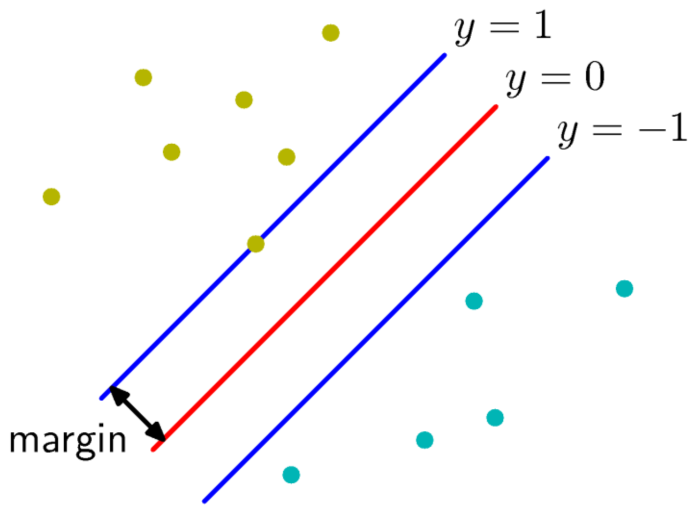


- SVM is a decision machine, so does not provide posterior probabilities (unfortunately).

- Relevance Vector Machine (RVM) is based on Bayesian formulation, and provides posterior probabilities.

Support Vector Machine (SVM)

- $y(\boldsymbol{x}) = \boldsymbol{w}^T \phi(\boldsymbol{x}) + b$
- Nonlinear fixed feature space mapping $\phi(\boldsymbol{x})$
- 2-class model, $t_n \in \{-1, +1\}$
- $\hat{t}_n = \begin{cases} +1, & y(\boldsymbol{x}) \geq 0 \\ -1, & y(\boldsymbol{x}) < 0 \end{cases}$
- If linearly separable, many solutions exist
- SVM: Maximum margin classifier



Support Vector Machine (SVM)

- Correctly classified points: $t_n y(\boldsymbol{x}_n) > 0$

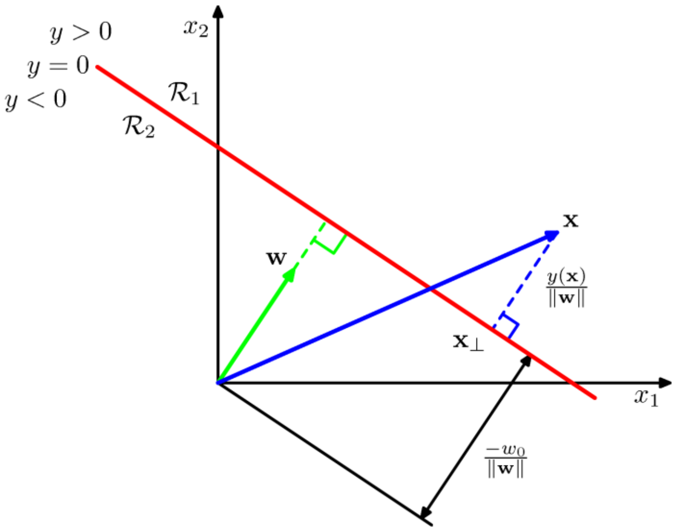
- $\max \frac{t_n y(\boldsymbol{x}_n)}{\|\boldsymbol{w}\|} = \max \frac{t_n(\boldsymbol{w}^T \phi(\boldsymbol{x}_n)+b\ )}{\|\boldsymbol{w}\|}$

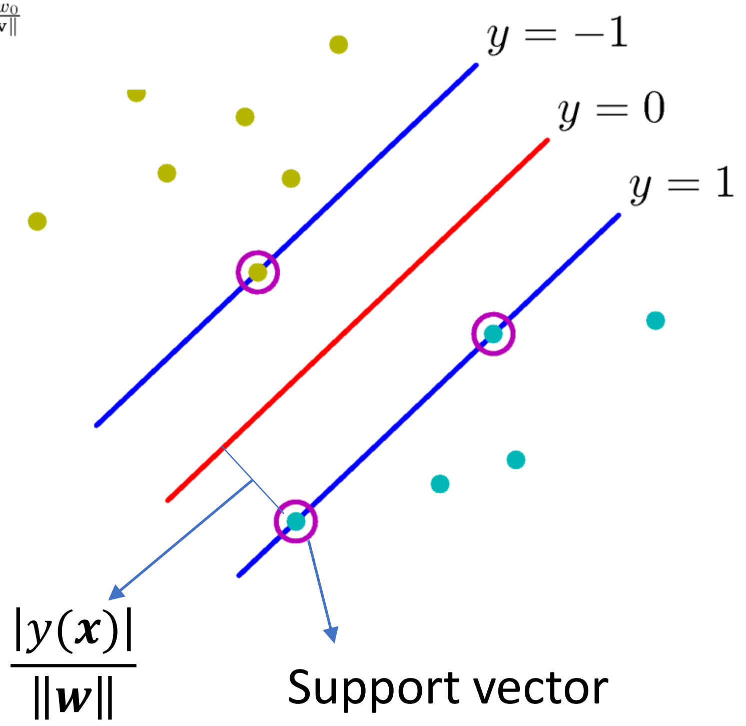
- $\arg\max_{\boldsymbol{w},b} \left\{ \frac{1}{\|\boldsymbol{w}\|} \min_n [t_n(\boldsymbol{w}^T \phi(\boldsymbol{x}_n) + b\ )] \right\}$

- $\frac{t_n y(\boldsymbol{x}_n)}{\|\boldsymbol{w}\|}$ does not change when $\boldsymbol{w} \rightarrow \kappa \boldsymbol{w}$ and b $\rightarrow \kappa b$

- Choose κ such that $t_n\left(\boldsymbol{w}^T \phi(\boldsymbol{x}_n) + b\ \right) = 1$

- $\arg\min_{\boldsymbol{w},b} \|\boldsymbol{w}\|^2$ s.t. $t_n\left(\boldsymbol{w}^T \phi(\boldsymbol{x}_n) + b\ \right) \geq 1, \ \ n = 1, \ldots, N$





Support Vector Machine (SVM)

- Using Lagrange multipliers $a_n \geq 0$ we obtain $\boldsymbol{w} = \sum_{n=1}^{N} a_n t_n \, \phi(\boldsymbol{x}_n)$

- Hence, $y(\boldsymbol{x}) = \sum_{n=1}^{N} a_n t_n k(\boldsymbol{x}, \boldsymbol{x}_n) + b$

- Kernel function: $k(\boldsymbol{x}_1, \boldsymbol{x}_2) = \phi(\boldsymbol{x}_1)^T \phi(\boldsymbol{x}_2)$

- Stationary kernels: $k(\boldsymbol{x}_1, \boldsymbol{x}_2) = k(\boldsymbol{x}_1 - \boldsymbol{x}_2)$

- Homogeneous kernels (Radial basis functions): $k(\boldsymbol{x}_1, \boldsymbol{x}_2) = k(\|\boldsymbol{x}_1 - \boldsymbol{x}_2\|)$

Support Vector Machine (SVM)

Kernel Trick: Compute the similarity score $k(\boldsymbol{x}, \boldsymbol{x}_n)$ directly without defining $\phi(\boldsymbol{x})$

- Linear Kernel:

$$k(\boldsymbol{x}_1, \boldsymbol{x}_2) = \boldsymbol{x}_1^T \boldsymbol{x}_2 = x_{11}x_{21} + x_{12}x_{22}$$

- Polynomial kernel

$$k(\boldsymbol{x}_1, \boldsymbol{x}_2) = \left(\boldsymbol{x}_1^T \boldsymbol{x}_2 + r\right)^d$$

e.g., (r=0, d=2)

$$k(\boldsymbol{x}_1, \boldsymbol{x}_2) = x_{11}^2 x_{21}^2 + x_{12}^2 x_{22}^2 + 2x_{11}x_{12}\, x_{21}x_{22}$$

(Hidden) $\phi(\boldsymbol{x}_1) = [x_{11}^2, x_{12}^2, \sqrt{2}\, x_{11}x_{12}]^T$

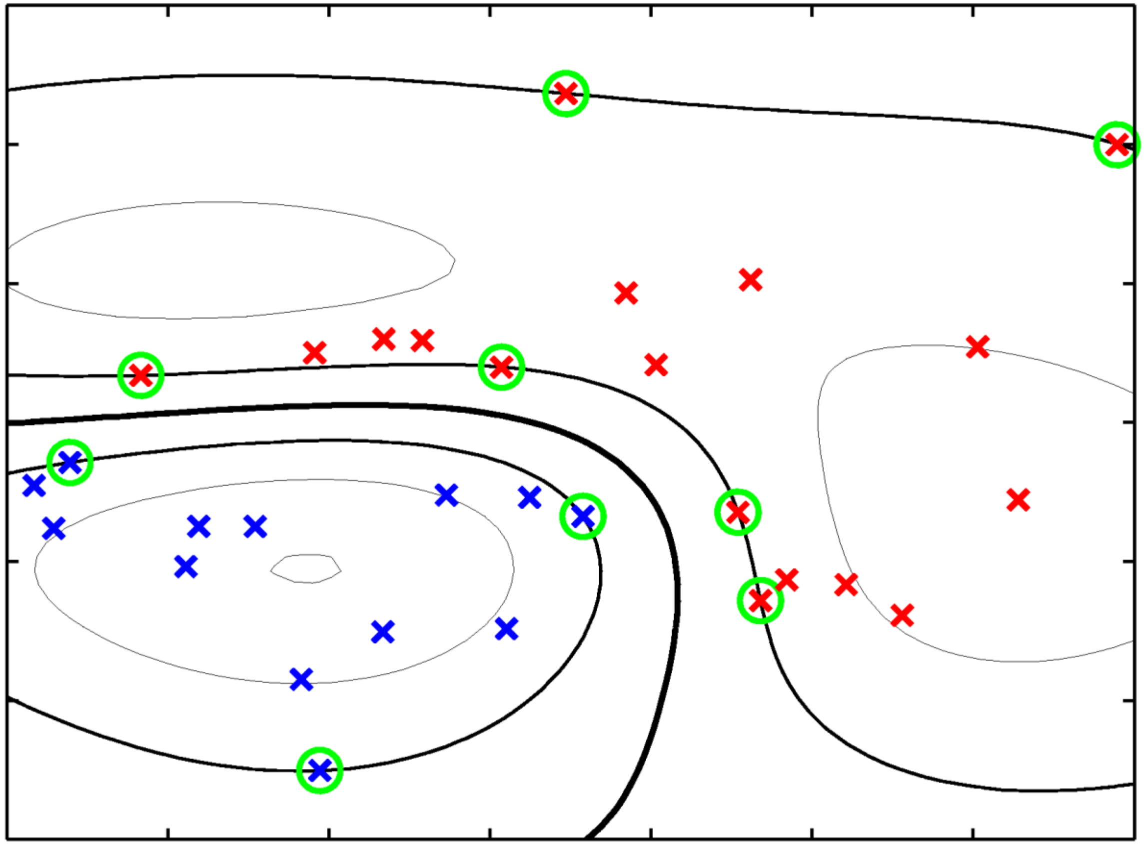
$$k(\boldsymbol{x}_1, \boldsymbol{x}_2) = \phi(\boldsymbol{x}_1)^T \phi(\boldsymbol{x}_2)$$

- Rbf

$$k(\boldsymbol{x}_1, \boldsymbol{x}_2) = e^{-\gamma \|\boldsymbol{x}_1 - \boldsymbol{x}_2\|^2} = e^{-\gamma (\boldsymbol{x}_1 - \boldsymbol{x}_2)^T (\boldsymbol{x}_1 - \boldsymbol{x}_2)}$$

- Sigmoid

$$k(\boldsymbol{x}_1, \boldsymbol{x}_2) = \tanh(\boldsymbol{x}_1^T \boldsymbol{x}_2 + r)$$



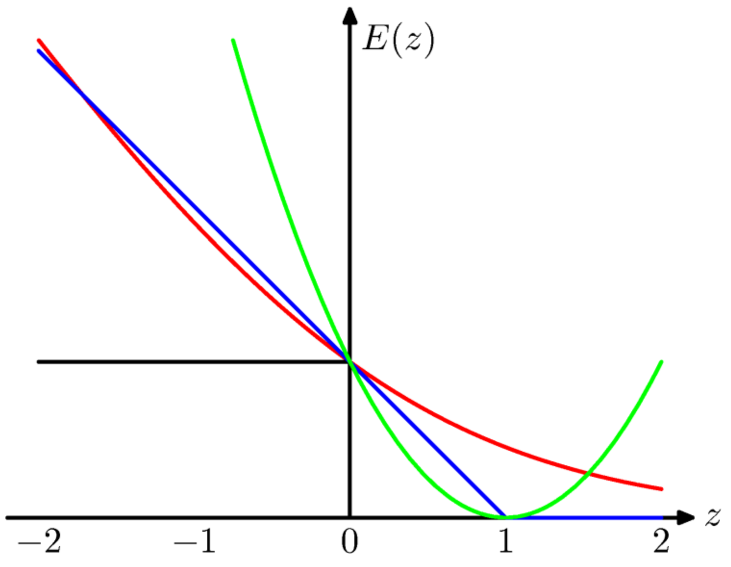
Custom Kernels: You can define your own kernel function. Must be a valid kernel function!

Support Vector Machine (SVM)

- When written in terms of minimization of a regularized error function, SVM has similarities with Logistic Regression and Perceptron:

  in the figure,
  blue for SVM (also for Perceptron by a shift of 1)
  red for Logistic Regression
  black for Misclassification error
  green for Quadratic error

  Hinge Loss: $\mathrm{E}(y_n t_n) = \max\{1 - y_n t_n, 0\}$



- Weighted voting (compare to kNN) with weights coming from the similarity metric $k(\boldsymbol{x}, \boldsymbol{x}_n)$

- Extends to multiclass problems

- Used also for anomaly detection (One-Class SVM) and regression (SVR)