Data Analysis 3

Support Vector Machines

Jan Platoš

November 11, 2018

Department of Computer Science
Faculty of Electrical Engineering and Computer Science
VŠB - Technical University of Ostrava

Table of Contents

Support Vector Machines

Other application of the Kernel Methods

Support Vector Machines

Support Vector Machines

- Naturally defined binary classification of numeric data.
- Multi-class generalization possible using several different strategies.
- Categorical features may be binarized and used.
- The class labels are assumed to be from the set $\{-1, 1\}$.
- The separation hyperplanes are used as classification criterion as with all linear models.
- The hyperplane is determined using a notion of margin.

Support Vector Machines - Linearly separable case

- A hyperplane that clearly separate points that belongs to the two classes.
- An infinite number of possible ways of constructing a linear hyperplane between classes exists.
- A maximum margin between hyperplanes have to be set, e.g. the minimum perpendicular distance to data points have to be maximum.

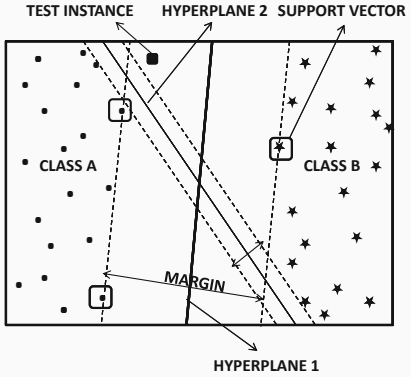


Figure 1: Linearly separable case

Support Vector Machines - Linearly separable case

- A hyperplane that cleanly separates two linearly separable classes exists.
- The margins of the hyperplanes is defined as he sum of its distances to the closest training points belonging to each of the two classes.
- The distance between the margin and the closest training points in either class is the same.
- A parallel hyperplanes may be constructed to the separating one that they touch the training points from either class and has no data points between them.
- The training points on these hyperplanes are referred to as the support vectors.
- The distance between the support vectors is the margin.
- The separating plane is precisely in the middle of these two hyperplanes in order to achieve the most accurate classification.

Support Vector Machines - Linearly separable case

Determination of the maximum margin hyperplane

- By setting up of the non-linear programming optimization formulation that maximizes the margin by expressing it as a functions of the coefficients of the hyperplane.
- The optimal coefficients can be determined by solving this optimization problem.

Support Vector Machines - Linearly separable case

Definition

- The n is the number of data points in the training set D
- The i-th data points is denoted as (X_i, y_i), where X_i is a d-dimensional row vector, and $y_i \in \{-1, +1\}$ is the binary class variable.

$$\overline{W} \cdot \overline{X} + b = 0$$

- $\overline{W} = (w_1, \ldots, w_d)$ is the d-dimensional row vector representing the direction of the normal of the hyperplane.
- b is a scalar, also know as bias.

Problem

Learning of the $(d + 1)$ *coefficients corresponding to the* $\overline{W}$ *and* b *from the training data that maximizes the margin.*

Support Vector Machines - Linearly separable case

- The points from either class have to lie on the opposite sides of the hyperplane.

$$\overline{W} \cdot \overline{X_i} + b \geq 0 \quad \forall i : y_i = +1$$
$$\overline{W} \cdot \overline{X_i} + b \leq 0 \quad \forall i : y_i = -1$$

- By introducing the margin parameter and its normalization and transformation we may get

$$y_i(\overline{W} \cdot \overline{X_i} + b) \geq +1 \quad \forall i \tag{1}$$

- The goal is to maximize the distance between two parallel hyperplanes.

$$\frac{2}{||\overline{W}||} = \frac{2}{\sqrt{\sum_{i=1}^{d} w_i^2}}$$

- Instead of the maximization of the above term we may minimize the following

$$\frac{||\overline{W}||^2}{2}$$

Support Vector Machines - Linearly separable case

- Minimization of the $\frac{\|\overline{W}\|^2}{2}$ is a complex quadratic programming problem because the parameter is minimized subject to a set of linear constraints, see eq. 1.
- Each data points leads to a constraint, therefore the SVM is computationally complex.
- One of the possible method that is able to solve such problem is a Lagrangian relaxations.
- It brings an set of non-negative multipliers $\overline{\lambda} = (\lambda_1, \ldots, \lambda_n)$ associated to each constraint.
- The constraints are then relaxed and the objective function is augmented by incorporating a Lagrangian penalty for constraints violation.

$$L_P = \frac{\|\overline{W}\|^2}{2} - \sum_{i=1}^{n} \lambda_i \left[y_i \left(\overline{W} \cdot \overline{X}_i + b \right) - 1 \right]$$

Support Vector Machines - Linearly separable case

- Conversion of the L_P into strictly pure maximization problem by eliminating the minimization part.
- The variables $\overline{W}$ and b are converted by gradient-based condition and set to zero.

$$\nabla L_P = \nabla \frac{\|\overline{W}\|^2}{2} - \nabla \sum_{i=1}^{n} \lambda_i \left[y_i \left(\overline{W} \cdot \overline{X}_i + b \right) - 1 \right] = 0$$

$$\overline{W} - \sum_{i=1}^{n} \lambda_i y_i \overline{X}_i = 0$$

- The expression of $\overline{W}$ is then derived directly

$$\overline{W} = \sum_{i=1}^{n} \lambda_i y_i \overline{X}_i$$

- The similar approach with variable b then generate

$$\sum_{i=1}^{n} \lambda_i y_i = 0$$

Support Vector Machines - Linearly separable case

- The final Lagrangian dual is as follows:

$$L_D = \sum_{i=1}^{n} \lambda_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \lambda_i \lambda_j y_i y_j \overline{X_i} \cdot \overline{X_j}$$

- The class label for the test instance $\overline{Z}$ defined by the decision boundary may be computed as

$$F(\overline{Z}) = sign\{\overline{W} \cdot \overline{Z} + b\} = sign\{(\sum_{i=1}^{n} \lambda_i y_i \overline{X_i} \cdot \overline{Z}) + b\}$$

- The solving of the L_D is done using gradient ascent according the parameter vector $\overline{\lambda}$.

Support Vector Machines - Soft margin for Linearly Non-separable Data

- The margin is defined as soft with penalization of the margin violation constraints.
- The definition of the Lagrangian is very similar to the Lagrangian for the separable case, with induction new constraints on the hyperplanes.

$$\begin{aligned} \overline{W} \cdot \overline{X_i} + b &\geq +1 - \xi_i \quad \forall i : y_i = +1 \\ \overline{W} \cdot \overline{X_i} + b &\leq -1 + \xi_i \quad \forall i : y_i = -1 \\ & \quad \forall i : \xi_i \geq 0 \end{aligned}$$

- Objective function O is then defined as

$$O = \frac{\|\overline{W}\|^2}{2} + C \sum_{i=0}^{n} \xi_i$$

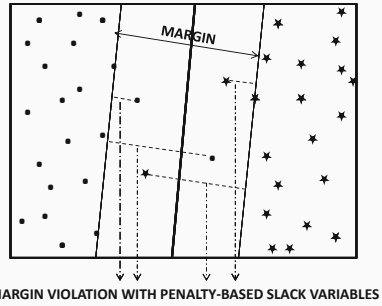


Figure 2: Soft margin for Non-separable Data

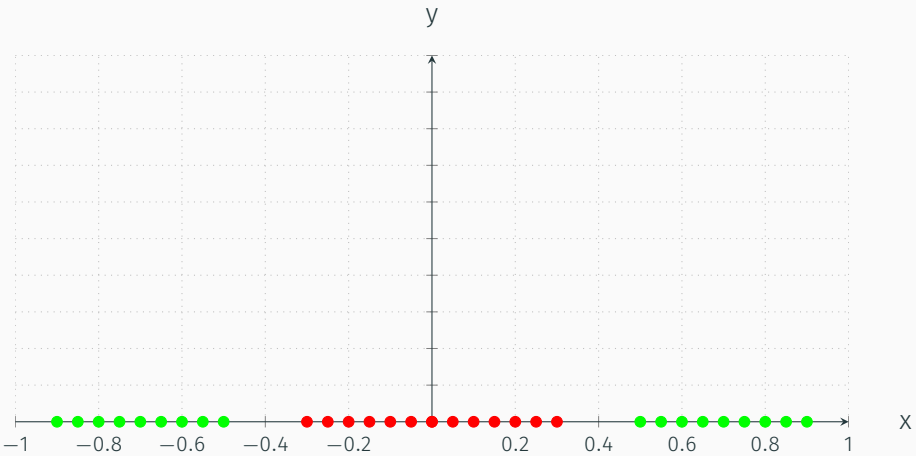
The C affects the allowed error during training (smaller C larger error)

Support Vector Machines

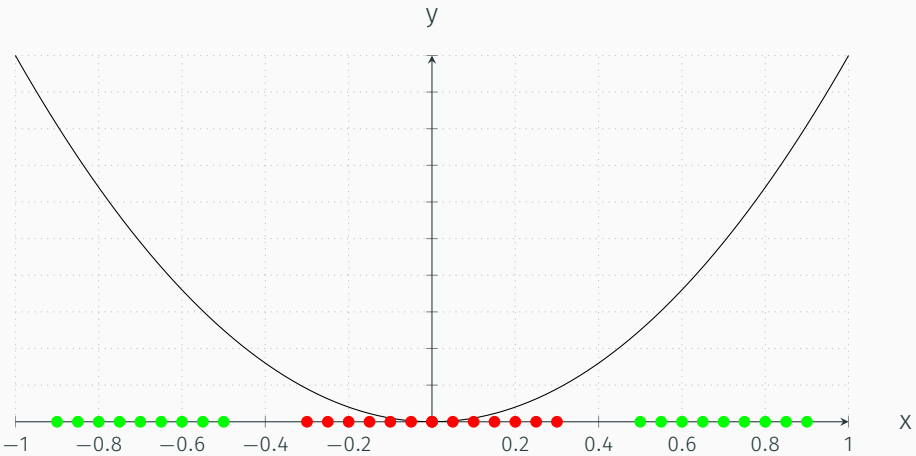
Non-linear decision boundary

- In real cases, the decision boundary is not linear.
- The points may be transformed into higher dimensions to enable linear decision boundary.

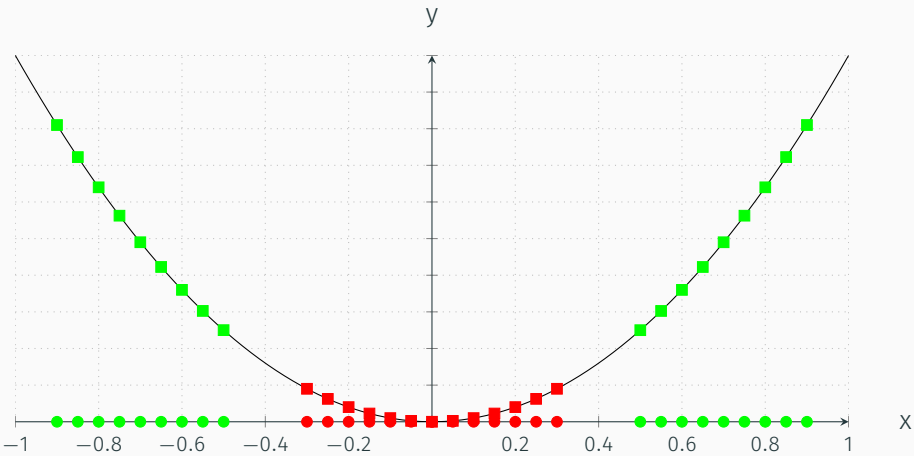
Support Vector Machines - Non-linear decision boundary



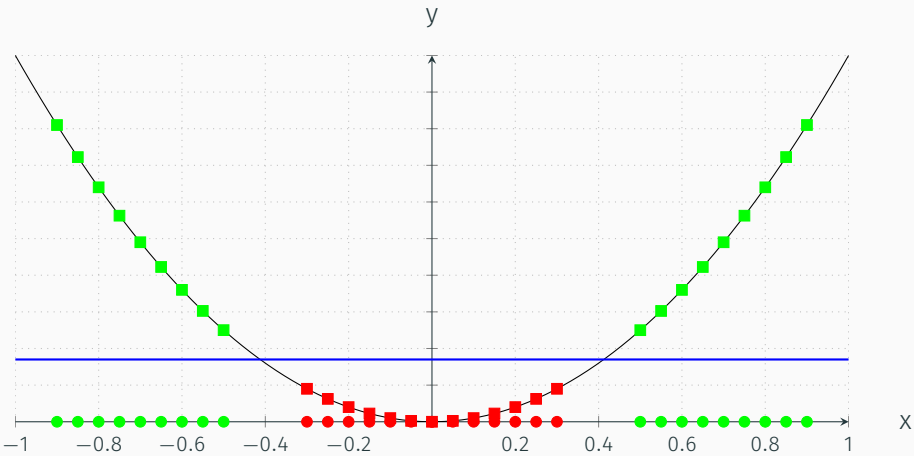
Support Vector Machines - Non-linear decision boundary



Support Vector Machines - Non-linear decision boundary



Support Vector Machines - Non-linear decision boundary



The Kernel Trick

- It leverages the important observation that the SVM formulation can be fully solved in the terms of dot products (or similarities) between pairs of data points.
- The feature values itself are not important or needed.
- The key is to define the pairwise dot products (similarity function) directly in the d'-dimensional transformed representation $\Phi(\overline{X})$ such as:

$$K(\overline{X_i}, \overline{X_j}) = \Phi(\overline{X_i}) \cdot \Phi(\overline{X_j})$$

- Only the dot product is required, therefore there is no need to compute transformed feature values $\Phi(X)$.

Support Vector Machines - The Kernel Trick

- The final Lagrangian dual with the substitution is defined as follows:

$$L_D = \sum_{i=1}^{n} \lambda_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \lambda_i \lambda_j y_i y_j K(\overline{X_i}, \overline{X_j})$$

- The class label for the test instance $\overline{Z}$ defined by the decision boundary may be computed as follows:

$$F(\overline{Z}) = sign\{\overline{W} \cdot \overline{Z} + b\} = sign\{(\sum_{i=1}^{n} \lambda_i y_i K(\overline{X_i}, \overline{Z})) + b\}$$

Support Vector Machines - The Kernel Trick

- All computations are performed in the original space.
- The actual transformation $\Phi(\cdot)$ does not to be known as long as the kernel similarity function $K(\cdot)$ is known.
- The kernel function have to be chosen carefully.
- The kernel function have to satisfy Mercer's theorem to be considered valid.
- This theorem ensures that the $n \times n$ kernel matrix (called Gramm matrix) $S = \left[K\left(\overline{X_i}, \overline{X_j}\right)\right]$ is symmetric, positive semidefinite.

Support Vector Machines - The Kernel Trick

Function	Form
Linear kernel	$K(\overline{X_i}, \overline{X_j}) = \overline{X_i} \cdot \overline{X_j} + c$
Polynomial kernel	$K(\overline{X_i}, \overline{X_j}) = (\alpha \overline{X_i} \cdot \overline{X_j} + c)^h$
Gaussian Radial Basis Function (RBF)	$K(\overline{X_i}, \overline{X_j}) = \exp\left(-\frac{\|\overline{X_i} - \overline{X_j}\|^2}{2\sigma^2}\right)$
Sigmoid kernel	$K(\overline{X_i}, \overline{X_j}) = \tanh(\kappa \overline{X_i} \cdot \overline{X_j} - \delta)$
Exponential kernel	$K(\overline{X_i}, \overline{X_j}) = \exp\left(-\frac{\|\overline{X_i} - \overline{X_j}\|}{2\sigma^2}\right)$
Laplacian kernel	$K(\overline{X_i}, \overline{X_j}) = \exp\left(-\frac{\|\overline{X_i} - \overline{X_j}\|}{\sigma}\right)$
Rational Quadratic Kernel	$K(\overline{X_i}, \overline{X_j}) = 1 - \frac{\|\overline{X_i} - \overline{X_j}\|^2}{\|\overline{X_i} - \overline{X_j}\|^2 + c}$

Other application of the Kernel Methods

Other application of the Kernel Methods

- Kernel K-means

$$\|\overline{X} - \overline{\mu}\|^2 = \|\overline{X} - \frac{\sum_{\overline{X_i} \in C} \overline{X_i}}{|C|}\|^2 = \overline{X} \cdot \overline{X} - 2\frac{\sum_{\overline{X_i} \in C} \overline{X} \cdot \overline{X_i}}{|C|} + \frac{\sum_{\overline{X_i}, \overline{X_j} \in C} \overline{X_i} \cdot \overline{X_j}}{|C|^2}$$

  - The μ is the centroid of cluster C.
  - the cluster is assigned to the data points according the minimal kernel-based distance.
- Kernel PCA
  - Replacement of the dot products in the mean-centered data matrix.
- Kernel fisher Discriminant
- Kernel Linear Discriminant Analysis.
- ...

Questions?