

SVM

Support Vector Machines

COMPUTATIONAL INTELLIGENCE

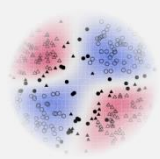


AGH University of Science and Technology

Adrian Horzyk

horzyk@agh.edu.pl

Google: [Horzyk](#)

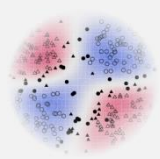


Support Vector Machines



Prof. V. Vapnik in 1998 created a new approach to shaping the neural network structure and defining the learning problem. He tried to eliminate the well-known **disadvantages of MLP and RBF neural networks** that minimize the non-linear error functions:

- The minimized function is usually multimodal with respect to the optimized parameters and has many **local minima** in which the learning process often **stuck** depending on the starting point that is typically defined by random weights.
- Learning algorithms are usually unable to effectively control the complexity of the neural network structures, which has a significant impact on the **generalizability** of the constructed solutions based on neural networks.

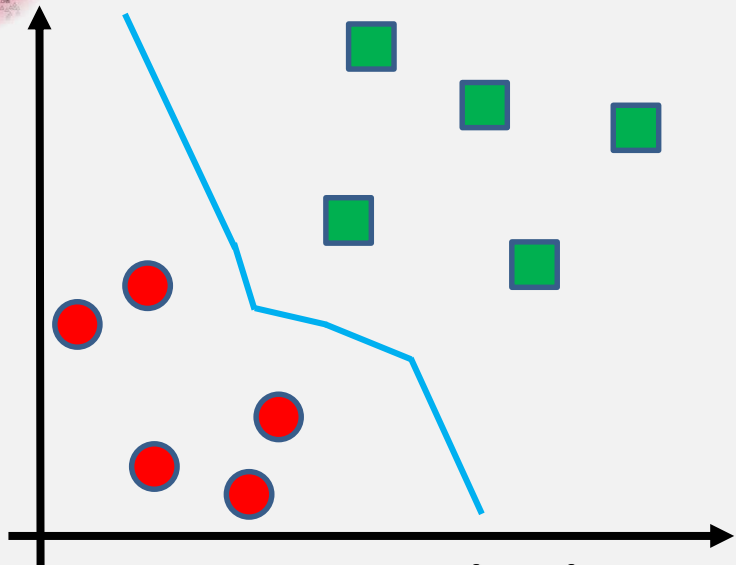


Essence of SVM

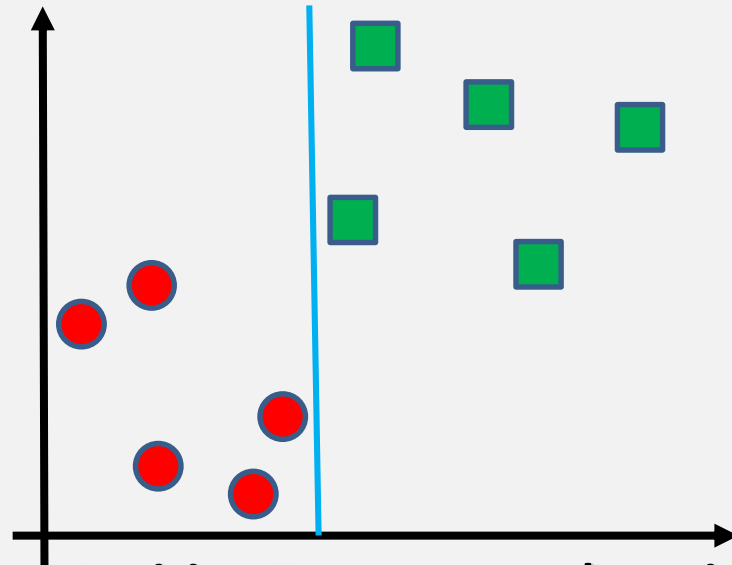


- The essence of the proposed change is to perform the learning process on the basis of **weight selection**, during which the **separation margin** between objects of the chosen and all other classes is maximized.
- This margin is defined between the most difficult separable objects (points of space), which define the so-called **support vectors**.
- SVM networks form a specific two-layer neural structure that uses different types of **activation functions (linear, polynomial, radial, or sigmoidal)**.
- There is used a learning technique based on **square programming**, which is characterized by only one global minimum.
- SVM networks are mainly dedicated to classification issues, where objects of one class are separated by the greatest possible margin from the objects of the other classes.
- It can be also adapter to some **regression tasks**.

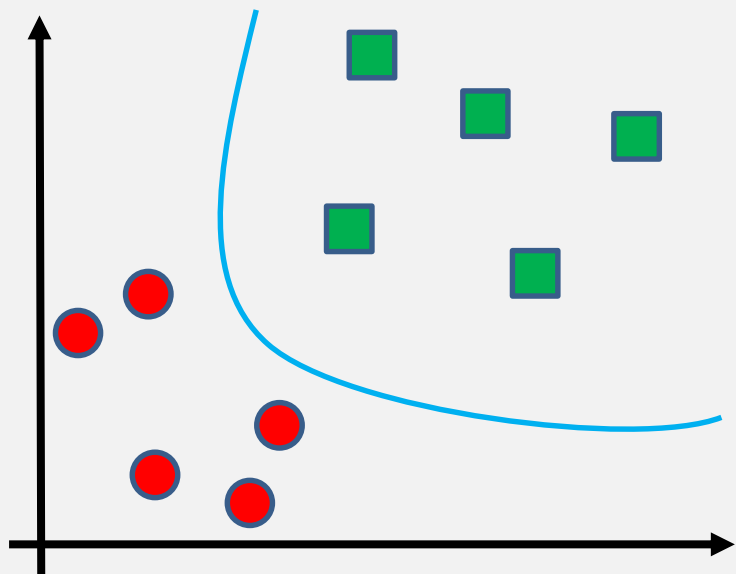
Discrimination and Classification Problems



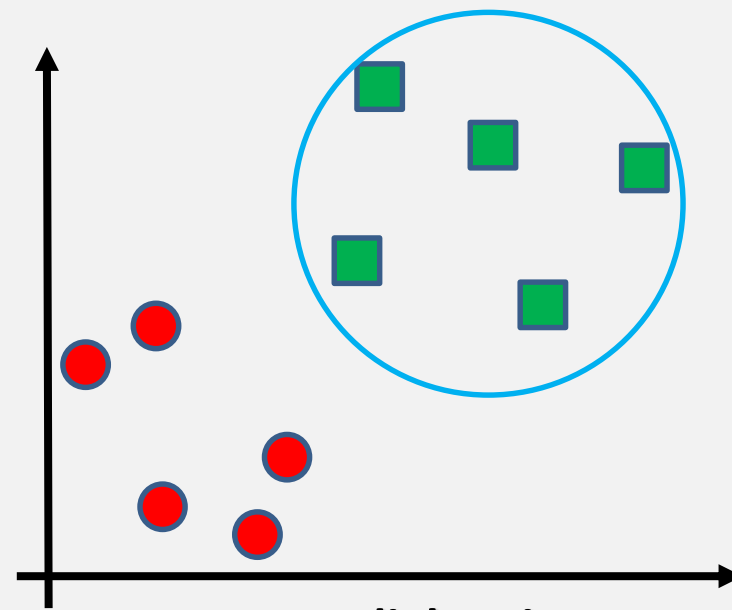
K-NN – Voronoi regions



Decision Tree – rectangle regions



MLP – non-linear hyperplanes

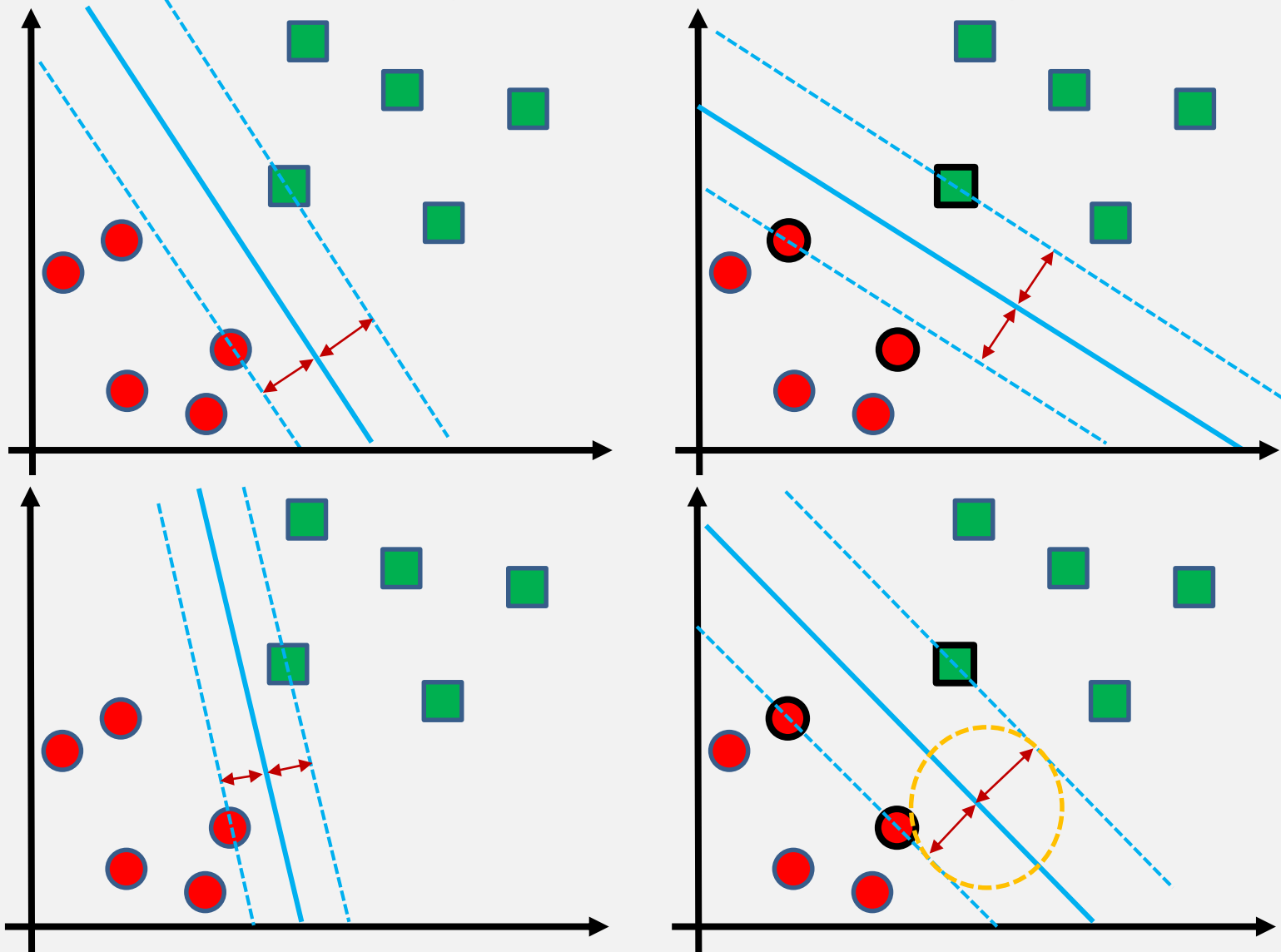


RBF – radial regions

The widest separation margin



The SVM method is designed to determine the **widest margin of separation** of objects of various classes. It discriminates objects of one selected class from objects of all other classes:



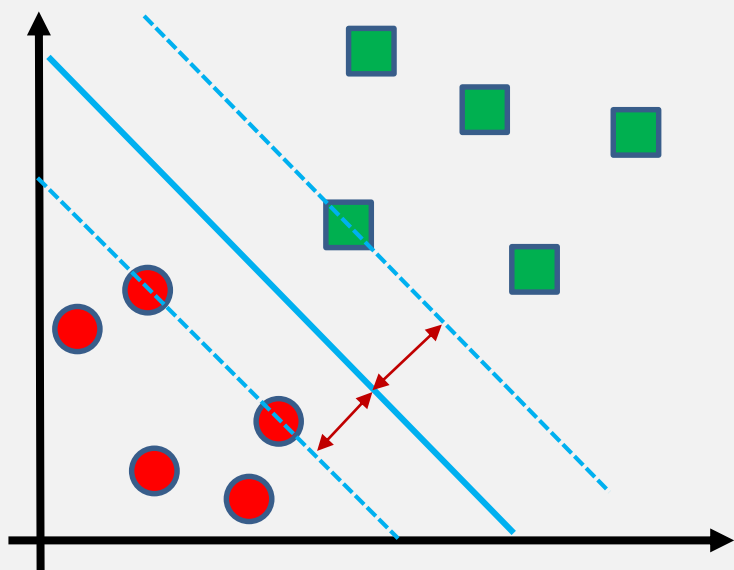
How to separate and discriminate?



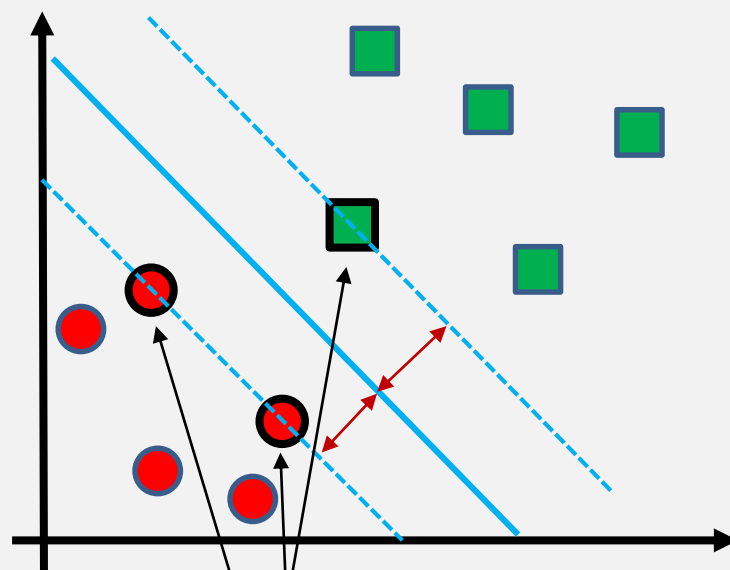
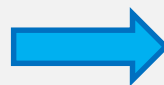
The goal is to maximize the margin separating objects (training patterns) of each class through determination of the optimal hyperplane that discriminates objects of one class against the objects of other classes.

This method takes into account only **these objects which are most difficult to separate and discriminate**, i.e. the objects that are close to objects of other classes.

The produced model should simplify representation, reduce the number of calculations, and supply us with satisfactory generalization.



SVM discrimination



Support Vectors

Support Vector Machine – SVM



Suppose that we have a set of learning pairs:

(x_i, d_i) for $i = 1, 2, \dots, p$

where x_i – input vector (training pattern, object)

$d_i \in \{-1; +1\}$ – discrimination pointer:

$d_i = +1$ – is used for the discriminated class,

$d_i = -1$ – is used for all other classes.

Assuming that it is possible to separate classes of objects of $d_i = +1$ class from the objects of $d_i = -1$ class linearly, it is possible to determine the equation of the hyperplane that separates these patterns: $y(x) = w^T x + b = 0$

where w – weight vector, x – input data vector, b – polarization

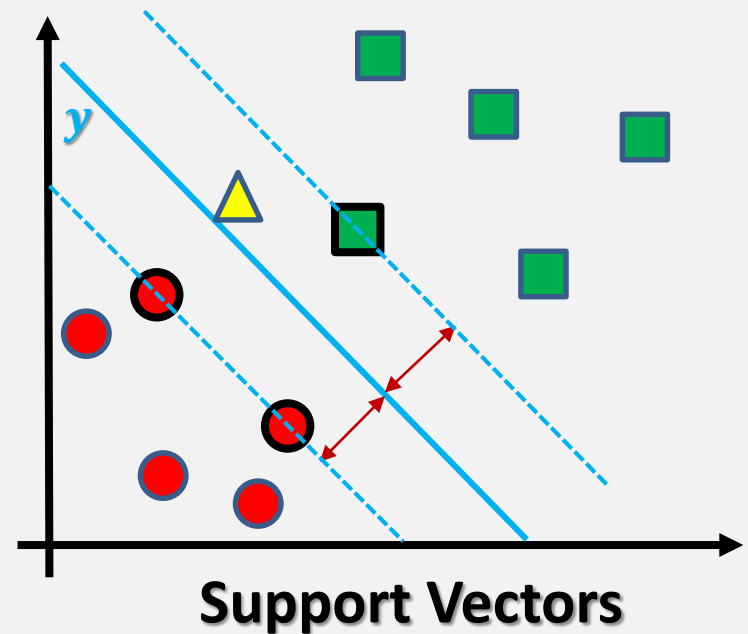
Thus, we can define decision-making inequalities:

If $w^T x + b \geq 0$ then $d_i = +1$ (for our discriminated class)

If $w^T x + b \leq 0$ then $d_i = -1$ (for the other classes)

On this basis we define inequality: $d_i(w^T x + b) \geq 1$

If this inequality is true for pairs (x_i, d_i) that define **support vectors**, which determine the hyperplane position and the width of the **separation margin**. Therefore, it is necessary to calculate b and w to **determine the decision**.



Crossing Separation Limits



Sometimes, it is impossible to use such a separation margin, especially for problems that are non-linearly separable where some pairs (x_i, d_i) lie within the separation margin zone. This can be expressed using the following inequality:

$$d_i(w^T x_i + b) \geq 1 - \delta_i$$

where

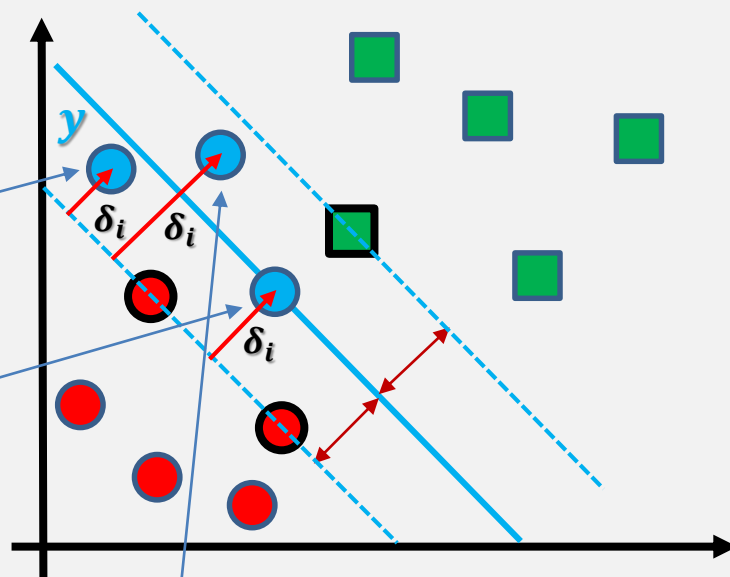
$\delta_i \geq 0$ makes this separation margin smaller:

If $0 \leq \delta_i < 1$ then (x_i, d_i) lies **on the right side of the separation hyperplane**, so the decision about classification will be **correct**.

If $\delta_i = 1$ then (x_i, d_i) lies **exactly on the hyperplane**, so the classification will be **undetermined**.

If $1 < \delta_i$ then (x_i, d_i) lies **on the wrong side of the separation hyperplane**, so the classification will be **incorrect**.

When determining the decision boundary, the value δ_i should be minimized as far as possible.



Width of Separation Margin



The **width of the separation margin** can be determined as Cartesian product of the weight vector and the **difference** of two support vectors belonging to the opposite classes:

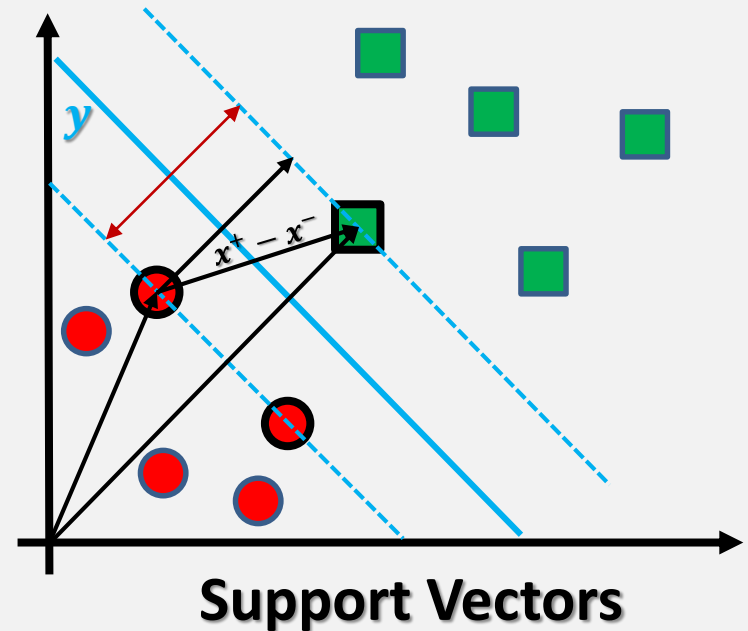
$$\rho = (x^+ - x^-) \cdot \frac{w}{\|w\|} = \frac{2}{\|w\|} = 2 \cdot r(x_{SV})$$

Because the distance between the support vectors and the hyperplane are defined as:

$$r(x_{SV}) = \frac{y(x_{SV})}{\|w\|} = \begin{cases} \frac{1}{\|w\|} & \text{for } y(x_{SV}) = 1 \\ \frac{-1}{\|w\|} & \text{for } y(x_{SV}) = -1 \end{cases}$$

In order to maximize the margin of separation between the support vectors of different classes $\rho = \frac{2}{\|w\|}$ it is necessary to minimize $\|w\|$, which is equivalent to minimizing the expression $\frac{1}{2} \|w\|^2$ with some linear constraints resulting from the defined decision inequality.

In such cases, we use [Lagrange multipliers](#) to find the extrema of a multivariate function subject to the defined constraints, so we minimize the Lagrange function.



Minimization of Lagrange Function



Determination of the Lagrange function for the problem of maximizing separation margin:

$$\min_w \frac{1}{2} \|w\|^2 + \vartheta \sum_{i=1}^p \delta_i$$

subject to the defined constraints:

$$\begin{aligned} d_i(w^T x_i + b) &\geq 1 - \delta_i \\ \delta_i &\geq 0 \end{aligned}$$

where ϑ – is the weight with which testing errors are weighted in comparison to the separation margin, determine the complexity of the network, which is selected by the user in an experimental manner, using e.g. cross-validation.

Finally, we get the following Lagrange function:

$$L(w, b, \alpha, \delta, \mu) = \frac{1}{2} w^T w + \vartheta \sum_{i=1}^p \delta_i - \sum_{i=1}^p \alpha_i [d_i(w^T x_i + b) - (1 - \delta_i)] - \sum_{i=1}^p \mu_i \delta_i$$

where α_i is a Lagrange multiplier vector with non-negative values corresponding to the particular functional constraints, μ_i is a Lagrange multiplier vector corresponding to the inequality constraints imposed on the variables δ_i .

Lagrange's minimization solution consists in determining the saddle point on the basis of the partial derivatives relative to multipliers.

Minimization of Lagrange Function



Conditions of optimal solution are determined by the following relationships:

$$\frac{\partial L(w, b, \alpha, \delta, \mu)}{\partial w} = 0 \rightarrow w = \sum_{i=1}^p \alpha_i d_i x_i$$

$$\frac{\partial L(w, b, \alpha, \delta, \mu)}{\partial b} = 0 \rightarrow \sum_{i=1}^p \alpha_i d_i = 0$$

$$\frac{\partial L(w, b, \alpha, \delta, \mu)}{\partial \mu} = 0 \rightarrow \mu_i = \vartheta - \alpha_i$$

Which now we substitute in the Lagrange function:

$$\begin{aligned} L(w, b, \alpha, \delta, \mu) &= \frac{1}{2} w^T w + \vartheta \sum_{i=1}^p \delta_i - \sum_{i=1}^p \alpha_i [d_i (w^T x_i + b) - (1 - \delta_i)] - \sum_{i=1}^p \mu_i \delta_i \\ &= \frac{1}{2} \sum_{i=1}^p \alpha_i d_i x_i \sum_{j=1}^p \alpha_j d_j x_j + \vartheta \sum_{i=1}^p \delta_i - \sum_{i=1}^p \alpha_i \left[d_i \left(\sum_{j=1}^p \alpha_j d_j x_j x_i + b \right) - (1 - \delta_i) \right] - \sum_{i=1}^p \mu_i \delta_i \\ &= \frac{1}{2} \sum_{i=1}^p \alpha_i d_i x_i \sum_{j=1}^p \alpha_j d_j x_j + \vartheta \sum_{i=1}^p \delta_i - \sum_{i=1}^p \alpha_i d_i x_i \sum_{j=1}^p \alpha_j d_j x_j + b \sum_{i=1}^p \alpha_i d_i + \sum_{i=1}^p \alpha_i (1 - \delta_i) - \sum_{i=1}^p (\vartheta - \alpha_i) \delta_i \\ &= \frac{1}{2} \sum_{i=1}^p \alpha_i d_i x_i \sum_{j=1}^p \alpha_j d_j x_j + \vartheta \sum_{i=1}^p \delta_i - \sum_{i=1}^p \alpha_i d_i x_i \sum_{j=1}^p \alpha_j d_j x_j + b \sum_{i=1}^p \alpha_i d_i + \sum_{i=1}^p \alpha_i - \sum_{i=1}^p \alpha_i \delta_i - \vartheta \sum_{i=1}^p \delta_i + \sum_{i=1}^p \alpha_i \delta_i \\ &= \sum_{i=1}^p \alpha_i - \frac{1}{2} \sum_{i=1}^p \sum_{j=1}^p \alpha_i \alpha_j d_i d_j x_i x_j \end{aligned}$$

Dual Problem



At the saddle point, the quotient of the Lagrange multiplier d_{SV} and the corresponding boundary constraints δ_{SV} with the support vector x_{SV} is equal to zero ($d_{SV}\delta_{SV} = 0$), because $\delta_{SV}=0$, so the relation:

$$d_i(w^T x_i + b) \geq 1 - \delta_i$$

at the point of support vector comes down to :

$$w^T x_i + b = \pm 1$$

This helps to determine the value b :

$$b = \pm 1 - w^T x_i$$

So we got a dual problem defined as:

$$\max_{\alpha} Q(\alpha) = \sum_{i=1}^p \alpha_i - \frac{1}{2} \sum_{i=1}^p \sum_{j=1}^p \alpha_i \alpha_j d_i d_j x_i x_j$$

For the defined constraints for $i = 1, 2, \dots, p$ defined as follows:

$$0 \leq \alpha_i \leq \vartheta \quad \sum_{i=1}^p \alpha_i d_i = 0$$

The solution of the dual problem allows us to find the desired hyperplane:

$$y(x) = \sum_{i=1}^p \alpha_i d_i x_i^T x_j + b$$

Conclusions and Remarks



The complementary variable δ_i nor Lagrange multipliers associated with it do not appear in the formulation of **the dual problem**.

Multipliers must meet the basic condition that the **product of multipliers and values of the constraints' function for each pair of learning data is equal to zero**.

If the constraint is satisfied with the excess for the non-support vectors, then the multipliers must be equal to zero. Non-zero multiplier values exist for the support vectors, so they determine support vectors which number is denoted as $N_{SV} \leq p$, and therefore the equation of the optimal-weighted linear SVM network defines a hyperplane dependent on the support vectors:

$$y(x) = \sum_{i=1}^{N_{SV}} \alpha_i d_i x_i^T x_j + b$$

Most of the classification problems are not linearly separable, so there is necessary to use the non-linear projection of original data into another functional space where the patterns become linearly separable and it is possible to use hyperplane to separate vectors.

There is necessary to use non-linear transformation with a sufficiently high dimension K of the feature space $K \geq N$.

Non-linear SVM

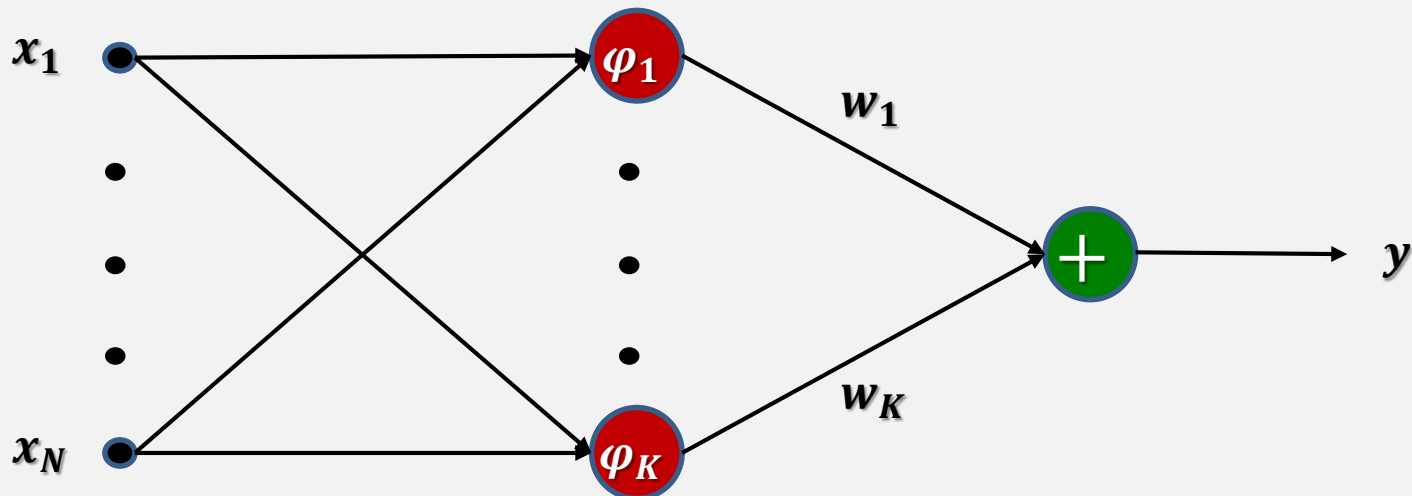


For non-linearly separable tasks, we do the linear projection of each pattern from its N-dimensional feature space to the K-dimensional feature space $\varphi_j(x)$, $j = 1, 2, \dots, K$. As a result of this non-linear transformation, the hyperplane equation will be defined by the following formula:

$$y(x) = w^T \varphi(x) + b = \sum_{j=1}^K w_j \varphi_j(x) + b = 0$$

where w_i denotes the weights of connections from the neuron of **non-linear activation function** φ_j computed on the input vector x to the output linear neuron.

Finally, we get a two-layer neural network structure containing one hidden layer:



Non-linear SVM Network



We get the solvation of the original problem by substituting the variable x_i by $\varphi_i(x)$:

$$\max_{\alpha} Q(\alpha) = \sum_{i=1}^p \alpha_i - \frac{1}{2} \sum_{i=1}^p \sum_{j=1}^p \alpha_i \alpha_j d_i d_j K(x_i, x_j)$$

where K is called a **kernel function**, defined as follows:

$$K(x_i, x_j) = \varphi^T(x_i) \varphi(x_j)$$

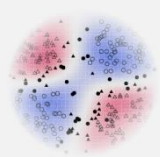
The solution to this problem is achieved by determination of the weight values:

$$w = \sum_{i=1}^p \alpha_i d_i \varphi(x_i)$$
$$b = \pm 1 - w^T \varphi(x_i)$$

Obtaining the output for the non-linear SVM:

$$y(x) = w^T \varphi(x) + b = \sum_{i=1}^{N_{SV}} \alpha_i d_i K(x_i, x) + b = 0$$

For the kernel function candidates K , we can select functions satisfying the condition of Mercator's theorem, e.g. Gaussian functions, polynomial, splines, and even sigmoidal functions with certain restrictions.



Non-linear Functions of SVM Kernel



The most commonly used kernel functions include:

➤ Linear functions:

$$K(x_i, x) = x^T x_i + \gamma$$

➤ Polynomial functions:

$$K(x_i, x) = (x^T x_i + \gamma)^p$$

➤ Gaussian functions:

$$K(x_i, x) = \exp(-\gamma \|x - x_i\|^2)$$

➤ Sigmoidal functions:

$$K(x_i, x) = \text{tgh}(\beta x^T x_i + \gamma)$$

Where β , γ are the fixed constants, and p is the degree of the polynomial.

The SVM radial base function network is very similar to the RBF network, although the way it is created and weights are computed differs.

Similarly, with the use of sigmoidal functions, we get a MLP double layer network. If you want to use SVM network to **discriminate more than two classes** of patterns, you have to construct a few SVM networks, which will discriminate patterns of each class from the others separately. In the end, results are added and combined.

Striving for Correctness of SVM

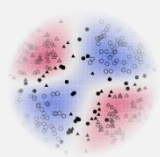


The penalties for failing constraints are often used, which forces the network to optimize for the adapted constants. Kuhn-Tucker's optimality conditions for the optimization problem formulated for SVM are as follows:

$$\begin{aligned}\alpha_i [d_i (w^T \varphi(x_i) + b) - (1 - \delta_i)] &= 0 \\ 0 &\leq \alpha_i \leq \vartheta \\ \mu_i \delta_i &= 0 \\ \alpha_i + \mu_i &= \vartheta \\ \delta_i &\geq 0\end{aligned}$$

Depending on the Lagrange coefficients, we can consider three cases:

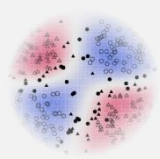
- $\alpha_i = 0$ – means that if $\alpha_i + \mu_i = \vartheta$ then $\mu_i = \vartheta$, so from the dependence $\mu_i \delta_i = 0$ comes up that $\delta_i = 0$, hence the learning pair (x_i, d_i) meets the restriction with the excess, so without reducing the width of the separation margin.
- $0 < \alpha_i < \vartheta$ – means that $\mu_i = \vartheta - \alpha_i$, hence also $\delta_i = 0$, hence the learning pair (x_i, d_i) defines the support vector, which lies exactly on the separation margin.
- $\alpha_i = \vartheta$ – means that $\mu_i = \vartheta - \alpha_i = 0$, so $\delta_i \geq 0$, which means that the learning pattern is within the separation margin causing narrowing of the separation margin or even on the wrong side $\delta_i > 1$.



Solving Dual Problem for Large Data Sets



- Regardless of the used kernel and the type of a task, the main computational problem in SVM networks is reduced to the quadratic programming task with linear constraints.
- The problem is a huge number of optimized variables, i.e. the Lagrange multipliers, which causes memory and computational complexity problems. This eliminates the ability to use the classical quadratic programming approach, e.g. MINOS, OSL, LOQO, and Matlab.
- As an alternative, there is used the decomposition of learning set to a number of subsets and the strategy of active constraints resulting from equality, neglecting those inactive with a sigh of greater inequality. This allows to move a part of patterns from the active set to the inactive set in the subsequent iterations.
- There are also used different versions of the SMO algorithm of sequential programming, the Platt's BSVM, or the suboptimal Joachims SVM_{Light} algorithm.



References and Bibliography



- T. Joachims, Making large scale SVM learning practical, in Advances in kernel methods – support vector learning, B. Scholkopf, C. Burges, A. Smola eds., MIT Press, pp. 41-56, Cambridge 1998.
- Lin C.J., Chang C.C, LIBSVM: a library for support vector machines:
<http://www.csie.ntu.edu.tw/~cjlin/libsvm>
- Open MIT Lectures about SVM :
<https://www.youtube.com/watch?v=PwhiWxHK8o>
- Caltech Lectures about SVM:
<https://www.youtube.com/watch?v=eHsErIPJWUU>