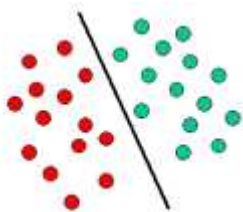


## Support Vector Machines (SVM)

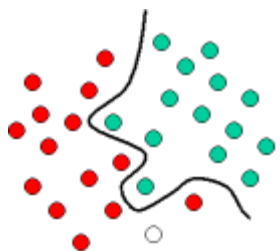
- [Introductory Overview](#)
- [Technical Notes](#)
- [Classification SVM](#)
- [Regression SVM](#)
- [Kernel Functions](#)

### Support Vector Machines (SVM) Introductory Overview

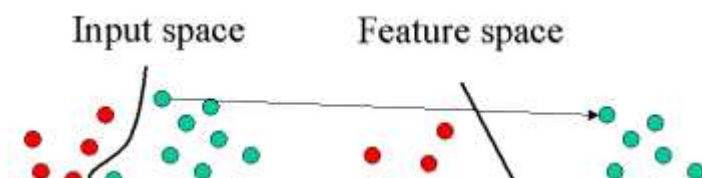
Support Vector Machines are based on the concept of decision planes that define decision boundaries. A decision plane is one that separates between a set of objects having different class memberships. A schematic example is shown in the illustration below. In this example, the objects belong either to class GREEN or RED. The separating line defines a boundary on the right side of which all objects are GREEN and to the left of which all objects are RED. Any new object (white circle) falling to the right is labeled, i.e., classified, as GREEN (or classified as RED should it fall to the left of the separating line).

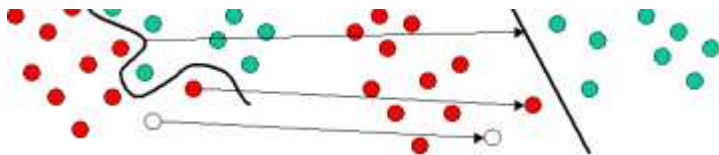


The above is a classic example of a linear classifier, i.e., a classifier that separates a set of objects into their respective groups (GREEN and RED in this case) with a line. Most classification tasks, however, are not that simple, and often more complex structures are needed in order to make an optimal separation, i.e., correctly classify new objects (test cases) on the basis of the examples that are available (train cases). This situation is depicted in the illustration below. Compared to the previous schematic, it is clear that a full separation of the GREEN and RED objects would require a curve (which is more complex than a line). Classification tasks based on drawing separating lines to distinguish between objects of different class memberships are known as hyperplane classifiers. Support Vector Machines are particularly suited to handle such tasks.



The illustration below shows the basic idea behind Support Vector Machines. Here we see the original objects (left side of the schematic) mapped, i.e., rearranged, using a set of mathematical functions, known as kernels. The process of rearranging the objects is known as mapping (transformation). Note that in this new setting, the mapped objects (right side of the schematic) is linearly separable and, thus, instead of constructing the complex curve (left schematic), all we have to do is to find an optimal line that can separate the GREEN and the RED objects.




[To index](#)

## Technical Notes

Support Vector Machine (SVM) is primarily a classifier method that performs classification tasks by constructing hyperplanes in a multidimensional space that separates cases of different class labels. SVM supports both regression and classification tasks and can handle multiple continuous and categorical variables. For categorical variables a dummy variable is created with case values as either 0 or 1. Thus, a categorical dependent variable consisting of three levels, say (A, B, C), is represented by a set of three dummy variables:

A: {1 0 0}, B: {0 1 0}, C: {0 0 1}

To construct an optimal hyperplane, SVM employs an iterative training algorithm, which is used to minimize an error function. According to the form of the error function, SVM models can be classified into four distinct groups:

- Classification SVM Type 1 (also known as C-SVM classification)
- Classification SVM Type 2 (also known as nu-SVM classification)
- Regression SVM Type 1 (also known as epsilon-SVM regression)
- Regression SVM Type 2 (also known as nu-SVM regression)

Following is a brief summary of each model.

[To index](#)

## Classification SVM

### Classification SVM Type 1

For this type of SVM, training involves the minimization of the error function:

$$\frac{1}{2} w^T w + C \sum_{i=1}^N \xi_i$$

subject to the constraints:

$$y_i (w^T \phi(x_i) + b) \geq 1 - \xi_i \text{ and } \xi_i \geq 0, \quad i = 1, \dots, N$$

where C is the capacity constant, w is the vector of coefficients, b a constant and  $\xi_i$  are parameters for handling nonseparable data (inputs). The index i labels the N training cases. Note that  $y \in \pm 1$  is the class labels and  $x_i$  is the independent variables. The kernel  $\phi$  is used to transform data from the input (independent) to the feature space. It should be noted that the larger the C, the more the error is penalized. Thus, C should be chosen with care to avoid over fitting.

### Classification SVM Type 2

In contrast to Classification SVM Type 1, the Classification SVM Type 2 model minimizes the error function:

$$\frac{1}{2} w^T w - \nu \rho + \frac{1}{N} \sum_{i=1}^N \xi_i$$

subject to the constraints:

$$y_i(w^T \phi(x_i) + b) \geq \rho - \xi_i, \quad \xi_i \geq 0, \quad i = 1, \dots, N \text{ and } \rho \geq 0$$

[To index](#)

## Regression SVM

In a regression SVM, you have to estimate the functional dependence of the dependent variable  $y$  on a set of independent variables  $x$ . It assumes, like other regression problems, that the relationship between the independent and dependent variables is given by a deterministic function  $f$  plus the addition of some additive noise:

$$y = f(x) + \text{noise}$$

The task is then to find a functional form for  $f$  that can correctly predict new cases that the SVM has not been presented with before. This can be achieved by training the SVM model on a sample set, i.e., training set, a process that involves, like classification (see above), the sequential optimization of an error function. Depending on the definition of this error function, two types of SVM models can be recognized:

### Regression SVM Type 1

For this type of SVM the error function is:

$$\frac{1}{2} w^T w + C \sum_{i=1}^N \xi_i + C \sum_{i=1}^N \xi_i^*$$

which we minimize subject to:

$$\begin{aligned} w^T \phi(x_i) + b - y_i &\leq \varepsilon + \xi_i^* \\ y_i - w^T \phi(x_i) - b_i &\leq \varepsilon + \xi_i \\ \xi_i, \xi_i^* &\geq 0, i = 1, \dots, N \end{aligned}$$

### Regression SVM Type 2

For this SVM model, the error function is given by:

$$\frac{1}{2} w^T w - C \left( v\varepsilon + \frac{1}{N} \sum_{i=1}^N (\xi_i + \xi_i^*) \right)$$

which we minimize subject to:

$$\begin{aligned} (w^T \phi(x_i) + b) - y_i &\leq \varepsilon + \xi_i \\ y_i - (w^T \phi(x_i) + b_i) &\leq \varepsilon + \xi_i^* \\ \xi_i, \xi_i^* &\geq 0, i = 1, \dots, N, \varepsilon \geq 0 \end{aligned}$$

## Kernel Functions

There are number of kernels that can be used in Support Vector Machines models. These include linear, polynomial, radial basis function (RBF) and sigmoid:

$$\phi = \left\{ \begin{array}{ll} x_i * x_j & \text{Linear} \\ (px_i x_j + \text{coefficient})^{\text{degree}} & \text{Polynomial} \\ \exp\{-\gamma |x_i - x_j|^2\} & \text{RBF} \end{array} \right\}$$

$$\left\{ \tanh(\gamma x_1 x_2 + \text{coefficient}) \quad \text{Sigmoid} \right\}$$

The RBF is by far the most popular choice of kernel types used in Support Vector Machines. This is mainly because of their localized and finite responses across the entire range of the real x-axis.

[To index](#)



© Copyright StatSoft, Inc., 1984-2004  
STATISTICA is a trademark of StatSoft, Inc.

---