ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

SUPPORT VECTOR MACHINES

Module - 4

Session - 16

AIM

To familiarize students with the concept of Support Vector Machines in Machine Learning.

INSTRUCTIONAL OBJECTIVES

This unit is designed to:

1. Describe the nature and feature of Support Vector Machines.
2. Demonstrate how to use Support Vector Machines for a classification problem solving.
3. Illustrate in detail how to use Support Vector Machines for regression with suitable example.

LEARNING OUTCOMES

At the end of this unit, you should be able to:

1. Define Support Vector Machines.
2. Describe the nature of ways to construct the Support Vector Machines for a classification problem.
3. Discover in detail how Support Vector Machines used for classification and regression technique with example.

SUPPORT VECTOR MACHINES

- The **support vector machine** or SVM framework is currently the most popular approach for "off-the-shelf" supervised learning in modern machine learning technique.
- It was introduced by Vapnik in 1992 and has taken off radically since then, principally because it often provides very impressive classification performance on reasonably sized datasets.
- Support Vector Machine is used for **Classification as well as Regression** problems. However, primarily, it is used for Classification problems in Machine Learning.
- This approach is fairly simple: we choose a kernel and then for given data, assemble the relevant quadratic problem and its constraints as matrices, and then pass them to the solver, which finds the **decision boundary and necessary support vectors** for us.

Cont.,

- The goal of the SVM algorithm is to create the **best line** or **decision boundary** that can be segregate n-dimensional space into classes.
- So that we can easily **put the new data point in the correct category** in the future. This best decision boundary is called **a hyperplane.**
- SVM chooses the extreme points/vectors that help in creating the hyperplane. **These extreme cases are called as support vectors**, and hence algorithm is termed as Support Vector Machine.
- If you don't have any specialized prior knowledge about a domain, then the SVM is an excellent method to try first.

PROPERTIES OF SVM

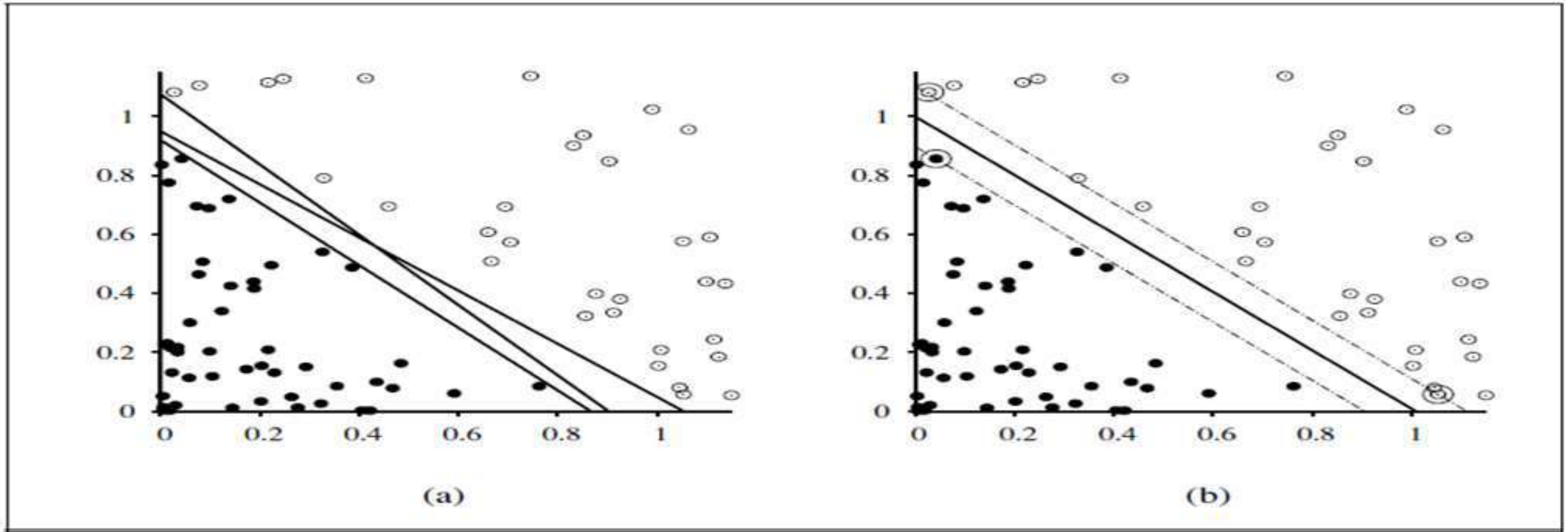
- There are three properties that make SVMs attractive:
  - SVMs construct a **maximum margin separator.**
  - SVMs create a **linear separating hyperplane**, but they have the ability to embed the data into a higher-dimensional space.
  - SVMs are a **nonparametric method**—they retain training examples and potentially need to store them all., using the so-called **kernel trick**.

Cont.,

- **Note:** A **nonparametric model** is one that cannot be characterized by a bounded set of parameters.
- **For example**, suppose that each hypothesis we generate simply retains within itself all of the training examples and uses all of them to predict the next example.
- Such a hypothesis family would be nonparametric because the effective number of parameters is unbounded it grows with the number of examples.

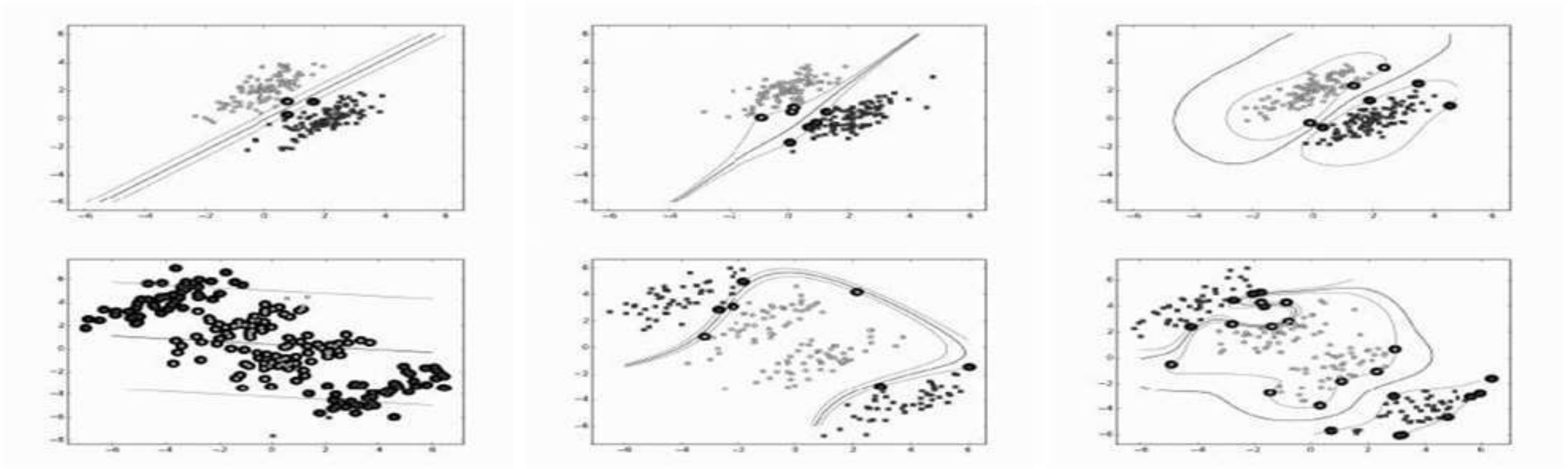
- You could say that SVMs are successful because of one key insight and one neat trick.

Two Different Categories That Are Classified Using A Decision Boundary Or Hyperplane



Support vector machine classification: (a) Two classes of points (black and white circles) and three candidate linear separators. (b) The maximum margin separator (heavy line), is at the midpoint of the **margin** (area between dashed lines). The **support vectors** (points with large circles) are the examples closest to the separator.

CONT.,



The SVM learning about a linearly separable dataset (*top row*) and a dataset that needs two straight lines to separate in 2D (*bottom row*) with *left* the linear kernel, *middle* the polynomial kernel of degree 3, and *right* the RBF kernel. $C = 0.1$ in all cases.

THE SUPPORT VECTOR MACHINE ALGORITHM

Initialisation

- For the specified kernel, and kernel parameters, compute the kernel of distances between the datapoints
  - The main work here is the computation $\mathbf{K} = \mathbf{X}\mathbf{X}^T$
  - For the linear kernel, return **K**, for the polynomial of degree *d* return $\frac{1}{\sigma}\mathrm{K}^{d}$
  - For the RBF kernel, compute $\mathbf{K} = \exp(-(\mathbf{x} - \mathbf{x}')^2 / 2\sigma^2)$

CONT.,

Training

- Assemble the constraint set as matrices to solve:

$$\underset{\mathbf{x}}{\text{Min}}\ \frac{1}{2}\,\mathbf{x}^T\, t_i t_j \mathbf{K}_\mathbf{x} + \mathbf{q}^T\,\mathbf{x} \text{ subject to } \mathbf{G}_\mathbf{x} \leq \mathbf{h}, \mathbf{A}\mathbf{x} = b$$

- Pass these matrices to the solver
- Identify the support vectors as those that are within some specified distance of the closest point and dispose of the rest of the training data
- Compute b* using the equation $b^* = \frac{1}{Ns} \sum_{\text{support vectors } j} (tj - \sum_{i=1}^{n} \lambda_i ti\, X_i^T\, X_j)$

CONT.,

Classification

- For the given test data **z**, use the support vectors to classify the data for the relevant kernel using:
    - Compute the inner product of the test data and the support vectors
    - Perform the classification as $\sum_{i=1}^{n} \lambda_i t_i \mathbf{K}(\mathbf{x}_i, \mathbf{z}) + b^*$, returning either the label (hard classification) or the value (soft classification)

MULTI-CLASS CLASSIFICATION

- The SVM only works for two classes. This might seem like a major problem, but with a little thought it is possible to find ways around the problem.
- For the problem of *N*-class classification, you train an SVM that learns to classify class one from all other classes, then another that classifies class two from all the others. So for *N*-classes, we have *N* SVMs.
- This still leaves one problem: how do we decide which of these SVMs is the one that recognises the particular input?
- The answer is just to choose the one that makes the strongest prediction, that is, the one where the basis vector input point is the furthest into the positive class region. It might not be clear how to work out which is the strongest prediction.

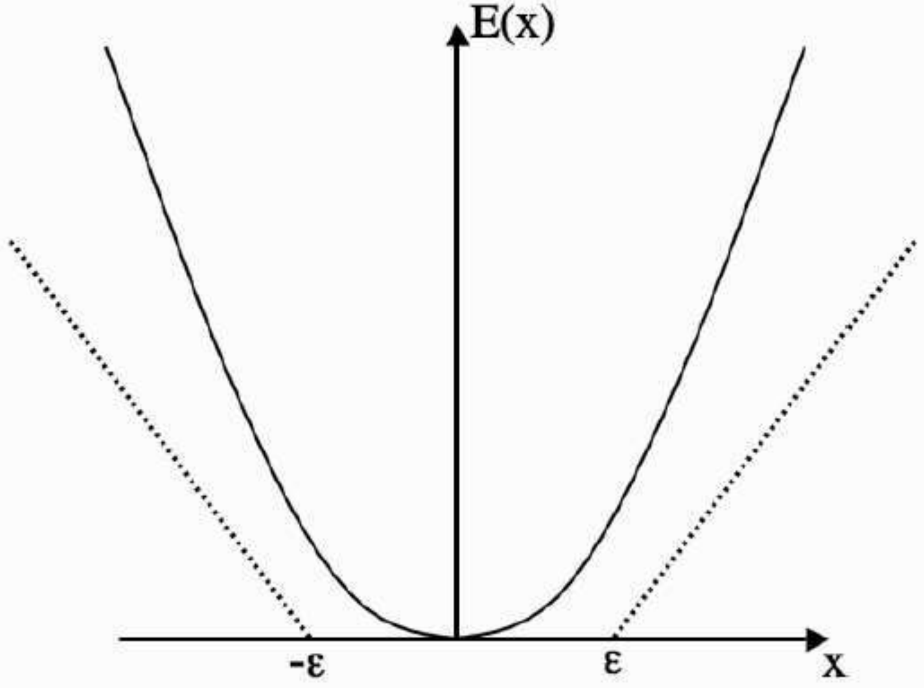
SVM REGRESSION

- Perhaps rather surprisingly, it is also possible to use the SVM for regression. The key is to take the usual least-squares error function (with the regulariser that keeps the norm of the weights small):

$$\frac{1}{2}\sum_{i=1}^{N}(t_i - y_i)^2 + \frac{1}{2}\lambda\|\mathbf{w}\|^2,$$

- and transform it using what is known as an **$\in$-insensitive error function** ($E\in$) that gives 0 if the difference between the target and output is less than $\in$ (and subtracts $\in$ in any other case for consistency). The reason for this is that we still want a small number of support vectors, so we are only interested in the points that are not well predicted.

THE $\in$-INSENSITIVE ERROR FUNCTION IS ZERO FOR ANY ERROR BELOW $\in$



The ϵ-insensitive error function is zero for any error below ϵ.

1. The goal of the SVM algorithm is to create the ____________ that can be segregate n-dimensional space into classes.

(a) Selected boundary
(b) Bad line
(c) Decision boundary
(d) Good line

2. Is SVMs are a nonparametric method?

(a) Yes
(b) No

3. Is SVM support multi class classification?

(a) Yes

(b) No

4. SVM regression using -insensitive error function, because of

(a) Want a small number of support vectors

(b) Want a big number of support vectors

(c) Want a average number of support vectors

(d) None of the above

THANK YOU

OUR TEAM