LECTURER: Nghia Duong-Trung

MACHINE LEARNING

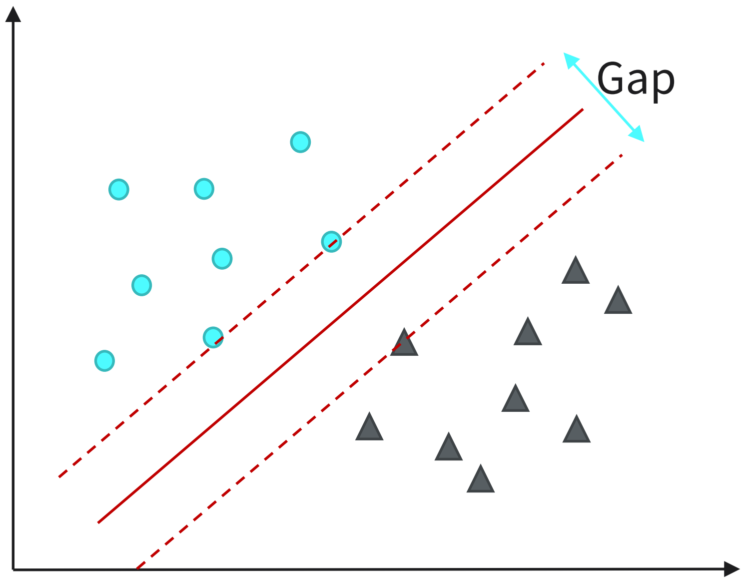
UNIT 4

SUPPORT VECTOR MACHINES

STUDY GOALS

- Know the definitions and terms used for Support Vector Machines (SVM)
- Comprehend common applications of SVM
- Understand different methods for SVM classifier and regressor
- Implement SVM methods in Python

- Support vector machine (SVM) is a **mathematical** model within the class of **supervised** learning for:
  - accurate classification & prediction
  - both continuous & categorical data
  - both linear & non-linear problems
  - high efficiency & effectivity



An example of SVM

Source of the text: Zöller , 2022, p. 88.
Source of the image: File: Normdist_regression.png (2023).

SVM applications

- Text categorization
- Image classification
- Geographic data classification
- Hand-writing recognition
- Protein classification
- Face detection
- Sentiment analysis

SVM methods

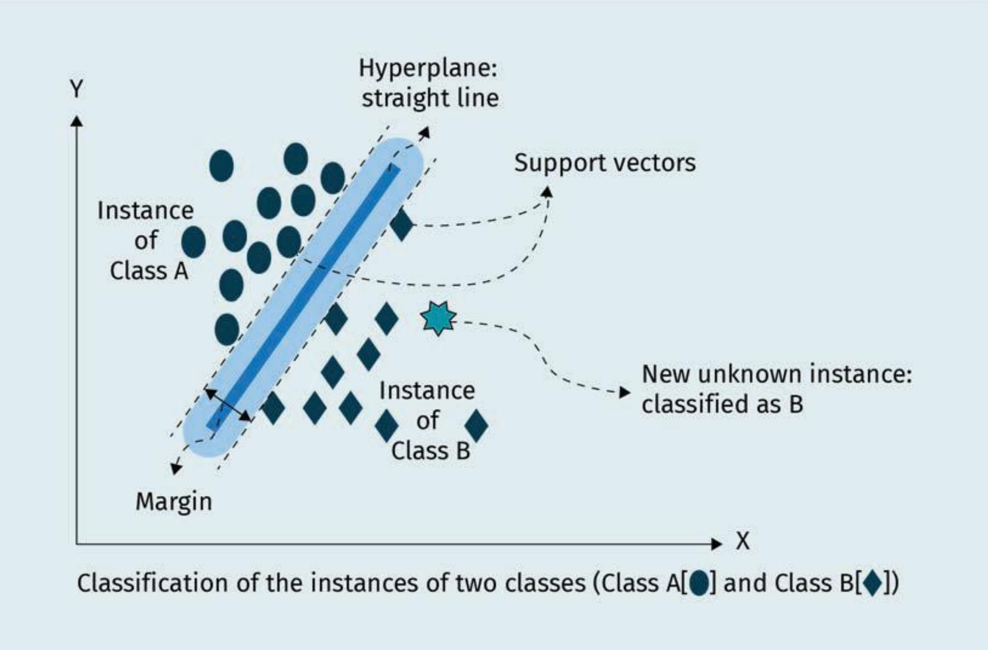
- **CLASSIFICATION HARD** MARGIN
- **CLASSIFICATION SOFT** MARGIN **PRIMAL** FORM
- **CLASSIFICATION SOFT** MARGIN **DUAL** FORM
- **REGRESSION** (SVR)

Source of the text: Zöller, 2022; Schölkopf, & Smola, 2002; Misra, 2019; Rajput, 2021.

Definition

- **Hyperplane**: a separating boundary that help to classify the points
- **Margin**: the boundary gap between two point-sets
- **Support vectors**: the points that determine the margin

Hyperplane in SVM

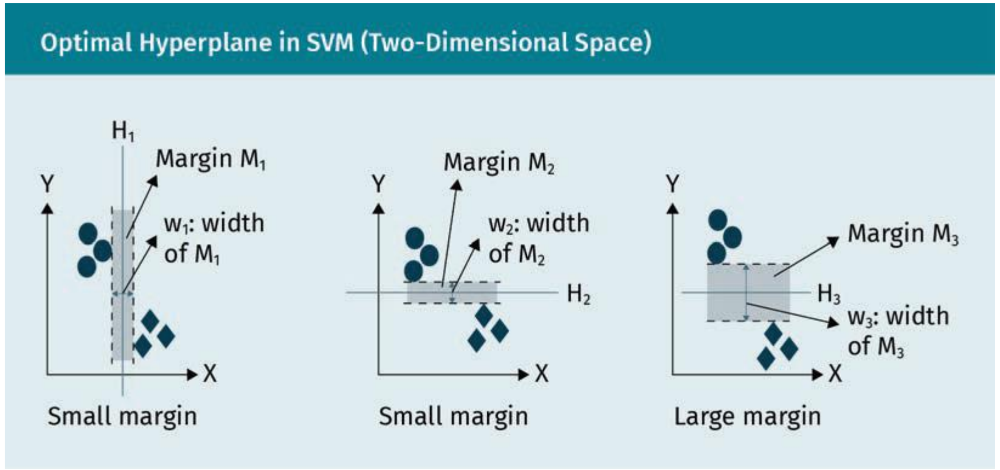


An example of SVM

Source of the text : Zöller, 2022, pp. 89-90.
Source the image: Zöller (2022, p. 89).

SVM **algorithm**:

- The separating hyperplane is detected (learned) by **maximizing the margin** width
- The margin width is measured based on the **perpendicular distance** from the separating hyperplane to the support vectors
- Large margin ensures good **generalization**
- This method is **memory efficient**, as it uses only a subset of training points (support vectors)



Optimization of Hyperplane in SVM

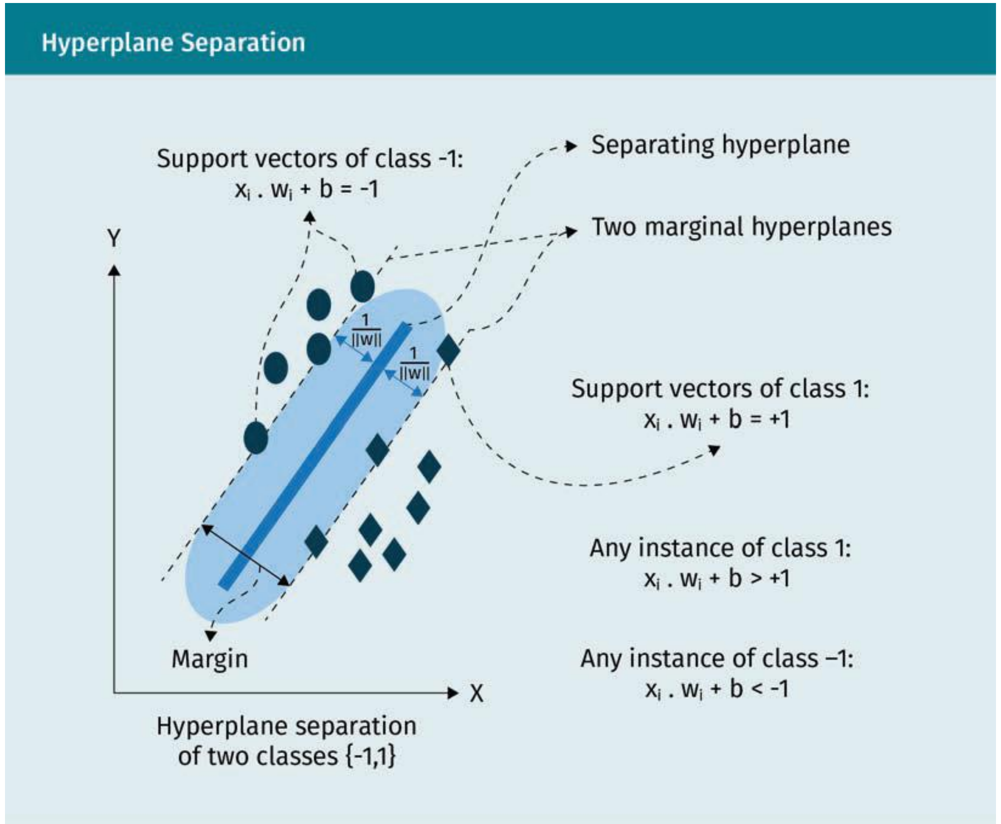
Source of the text : Zöller, 2022, p. 90.
Source of the image : Zöller (2022, p. 90).

SVM classifier with **Hard Margin:**

- Only suitable for **Linearly Separable** data
- A separating hyperplane (i.e., decision function f(x)=0) can be detected **without training errors**
- For a dataset of multiple features:

$$f(x) = \sum_{i=1}^{n} w_i \cdot x_i - b$$

- Optimization problem: **Minimize** $\{||\boldsymbol{w}||\}$
- Hard margin is **Sensitive** to outliers

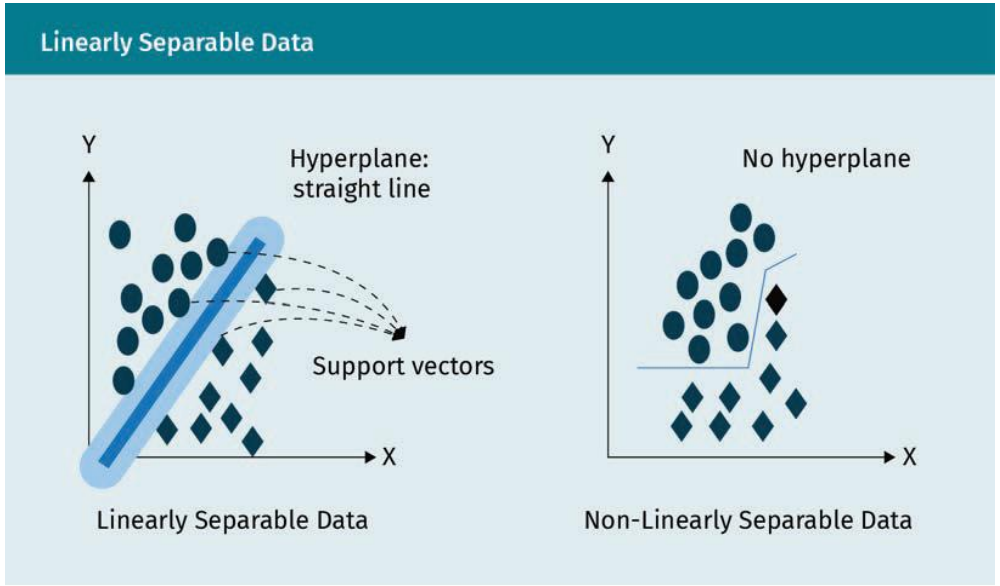


An example of SVM with hard margin for classification

Source of the text : Zöller, 2022, pp. 91 - 94.
Source of the image : Zöller (2022, p. 94).

SVM classifier with **Soft Margin:**

- An **extension** of Hard Margin SVM for **nonlinearly** separable data
- Allow SVM to make a certain number of **mistakes** and keep the margin as wide as possible



An example of linearly and non-linearly separable data

.
Source of the text : Zöller, 2022; Misra, 2019.
Source of the image : Zöller (2022, p. 92).

SVM classifier with **Soft Margin:**

— **Primal Form:**

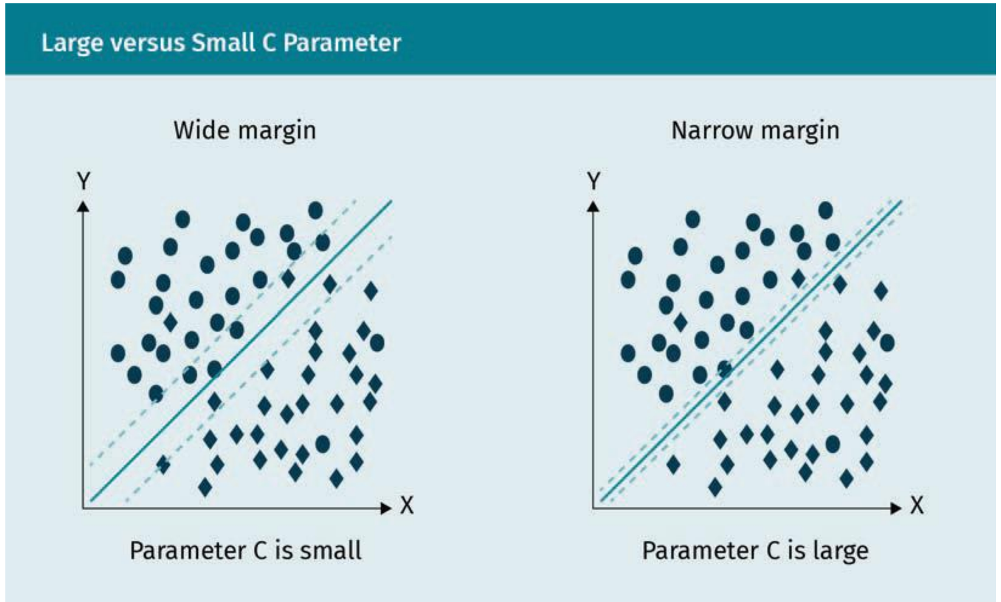
- The data are **almost** linearly separable
- Some data points can lie **inside** the margin area
- Optimization problem:

$$\textbf{Minimize } \frac{C}{n}\sum_{i=1}^{n}\xi_i + \|w\|^2$$

Where:

C – trade-off parameter

ξ_i - slack error



An application of SVM with Soft Margin/Primal form for classification

Source of the text : Zöller, 2022; Misra, 2019; Rajput, 2021.
Source of the image : Zöller (2022, p. 97).

SVM classifier with **Soft Margin:**

– **Dual Form:**

- an **adjusted form** from primal form
- uses a **Lagrangian multiplier** α in defining weight vector:

$$\boldsymbol{w} = \sum_{j=1}^{n} \boldsymbol{\alpha_j y_j x_j}$$

- Optimization problem: **Maximize** quadratic function:

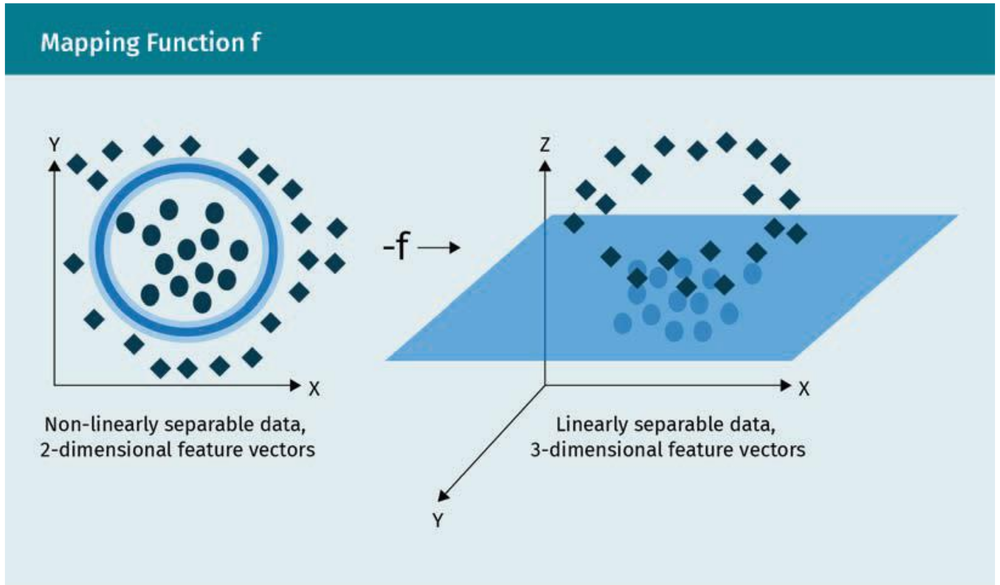
$$Q(\alpha_i) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j (x_i \cdot x_j)\, y_i y_j$$

- Dual form is preferred when data has a **huge dimension**, and the **kernel trick** is needed to reduce the computational cost

Source of the text: Zöller, 2022); Schölkopf, & Smola, 2002; Misra, 2019; Rajput, 2021.

Transformation

- **Real-life datasets** are usually not linearly separable
- The nonlinearly separable data can be transformed to linearly separable data by **adding a new dimension** of the feature space
- **Problems**: Overfitting & high computational cost



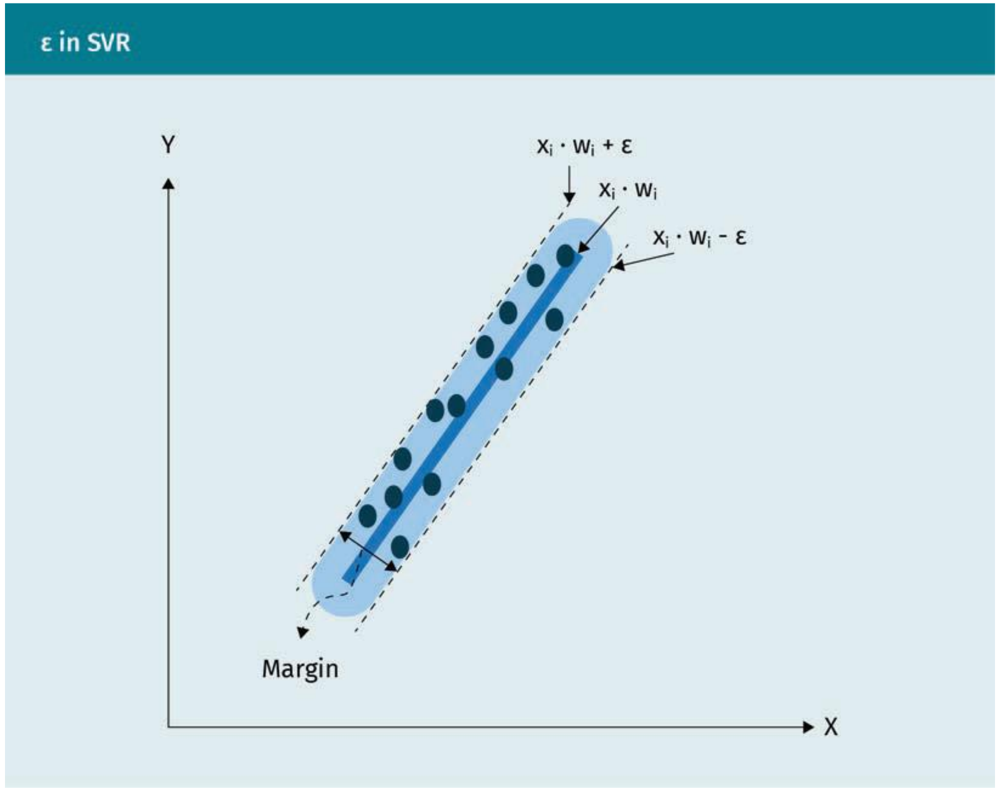
An example of Transformation from non-linearly to linearly separable data

Source of the text : Zöller, 2022.
Source of the image : Zöller (2022, p. 98).

SVM Regression (SVR): a version of the SVM classifier

- SVR **predicts** numerical output values with some deviation (**epsilon error ε**)
- Objective function: **Minimize** the model coefficient **w** and introduce the cost C (**trade-off** parameter) and two slack variables ξ_i, ξ_i^*:

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

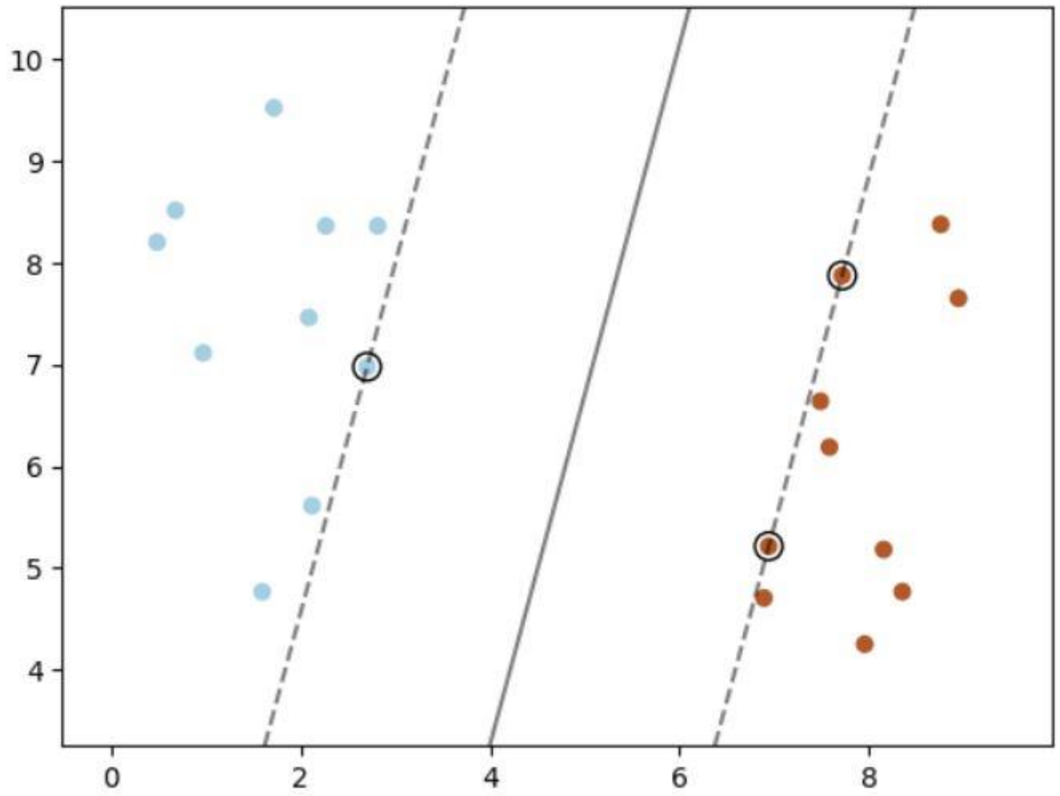


An example of SVR

Source of the text : Zöller, 2022.
Source of the image : Zöller (2022, p. 101).

3.4 SVM WITH PYTHON

```
# SVM classifier with linear kernel
import matplotlib.pyplot as plt
from sklearn import svm
from sklearn.datasets import make_blobs
from sklearn.inspection import DecisionBoundaryDisplay
# we create 20 separable points
X, y = make_blobs(n_samples=20, centers=2, random_state=20)
# fit the model, don't regularize for illustration purposes
clf = svm.SVC(kernel="linear", C=1000)
clf.fit(X, y)
plt.scatter(X[:, 0], X[:, 1], c=y, s=30, cmap=plt.cm.Paired)
# plot the decision function
ax = plt.gca()
DecisionBoundaryDisplay.from_estimator( clf, X, plot_method="contour",
  colors="k", levels=[-1, 0, 1], alpha=0.5, linestyles=["--", "-", "--"], ax=ax,)
# plot support vectors
ax.scatter(clf.support_vectors_[:, 0],clf.support_vectors_[:, 1],
  s=100, linewidth=1, facecolors="none", edgecolors="k", )
plt.show()
```

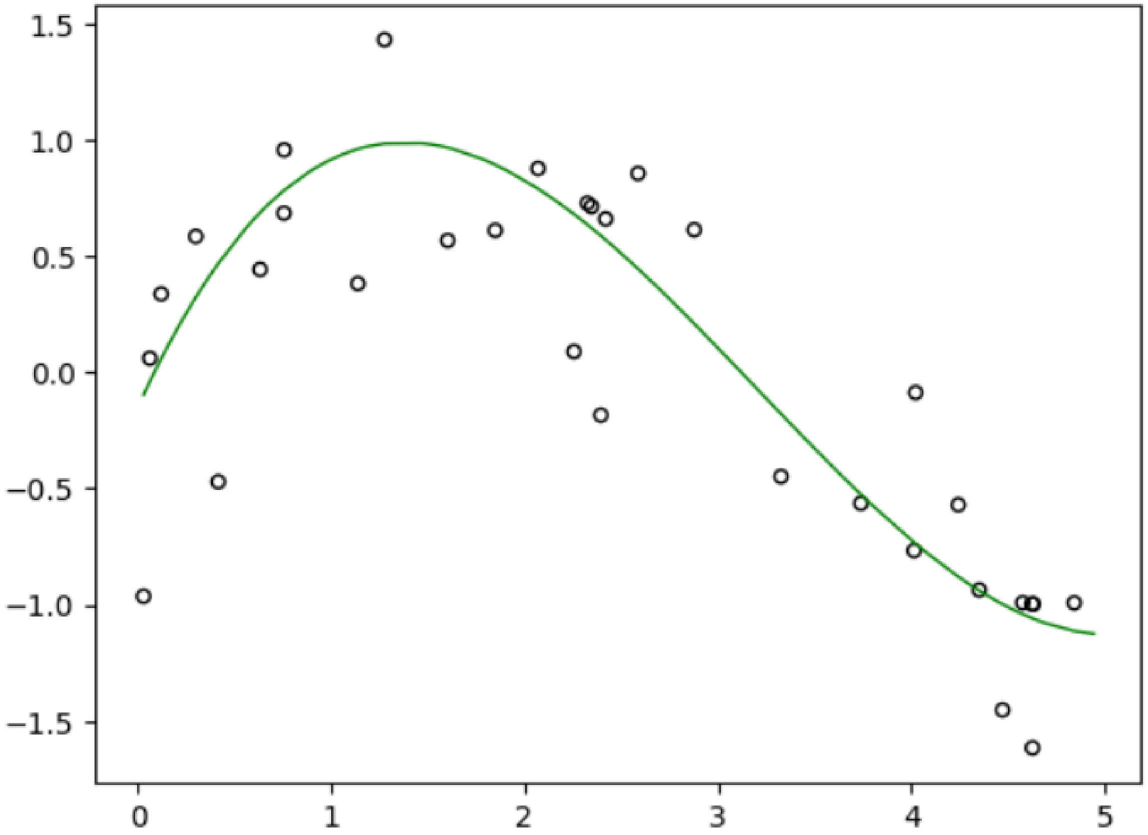


Maximum margin separating hyperplane within a two-class separable dataset using SVM classifier

Source of the text: Zöller, 2022.

3.4 SVM WITH PYTHON

```
# SVM regression with poly kernel
import numpy as np
from sklearn.svm import SVR
import matplotlib.pyplot as plt
X = np.sort(5 * np.random.rand(100, 1), axis=0)
y = np.sin(X).ravel()
# add noise to targets
y[::5] += 2 * (0.5 - np.random.rand(20))
svr = SVR(kernel="poly", C=100, gamma="auto", degree=3, epsilon=0.1,
coef0=1)
ax = plt.gca()
#plot regression curve
ax.plot(X,svr.fit(X, y).predict(X),color=model_color[ix],lw=1)
ax.scatter(X[svr.support_],y[svr.support_],facecolor="none",
edgecolor="k",s=20)
plt.show()
```



SVR with poly kernel

REVIEW STUDY GOALS

- Know the definitions and terms used for Support Vector Machines (SVM)
- Comprehend common applications of SVM
- Understand different methods for SVM classifier and regressor
- Implement SVM methods in Python

SESSION 4

TRANSFER TASK

1. Create the dataset using the following code:

```
from sklearn.datasets import make_blobs
X, y = make_blobs(n_samples=100, centers=2, random_state=12)
```

Implement SVM classifier using

a) Poly kernel

b) RBF kernel

2. Create the dataset using the following code:

```
import numpy as np
X = np.sort(5 * np.random.rand(100, 1), axis=0)
y = np.sin(X).ravel()
y[::5] += 2 * (0.5 - np.random.rand(20)) # adding noises
```

Implement SVM regression using

a) linear kernel

b) RBF kernel

TRANSFER TASK
PRESENTATION OF THE RESULTS

Please present your results.
The results will be discussed in plenary.

1. The idea of SVM classification is to find a hyperplane that__________ the margin between classes.?

   a) Removes
   b) Minimizes
   c) Maximizes
   d) None of these

2. The What is/are true about kernel in SVM?

(1): Kernel function maps low dimensional data to a high dimensional space.

(2): It’s a similarity function.

a) Neither option is correct
b) Options 1 and 2 are correct
c) Option 1 is correct
d) Option 2 is correct

3. The regularization parameter in the SVM cost function determines…

a) …the tradeoff between misclassification and simplicity of the model.
b) …the number of cross-validations to be made.
c) …the kernel to be used.
d) None of the above.

4. If I am using all features of my dataset and I achieve 100% accuracy on my training set, but ~70% on testing set, what should I look out for?

a) Underfitting
b) Nothing, the model is perfect.
c) Include more testing data
d) Overfitting

5. The constraint of the Support Vector Regression model is that the absolute error is ____ the value of error term epsilon.

a) Less than or equal to
b) Not related to
c) Greater than
d) Exactly equal to

LIST OF SOURCES

Text:

Schölkopf, B., & Smola, A. J. (2002). Learning with kernels: support vector machines, regularization, optimization, and beyond. *MIT Press.*

Misra, R. (2019). Support Vector Machines – Soft Margin Formulation and Kernel Trick. https://towardsdatascience.com/support-vector-machines-soft-margin-formulation-and-kernel-trick-4c9729dc8efe

Rajput, V. (2021). The Optimization Behind SVM: Primal and Dual Form. https://medium.com/aiguys/the-optimization-behind-svm-primal-and-dual-form-5cca1b052f45

Images:

File: Normdist_regression.png (2023)

Zöller (2022, p. 89)

Zöller (2022, p. 90)

Zöller (2022, p. 92)

Zöller (2022, p. 94)

Zöller (2022, p. 97)

Zöller (2022, p. 98)

Zöller (2022, p. 101)