Curso Data Science

Skill Lab

Prof Me Eng Marcelo Bianchi
Data Scientist | Data Engineer | Data Architect

Aula 9 – SVR: Support Vector Regression

Machine Learning

SVR – Support Vector Regression

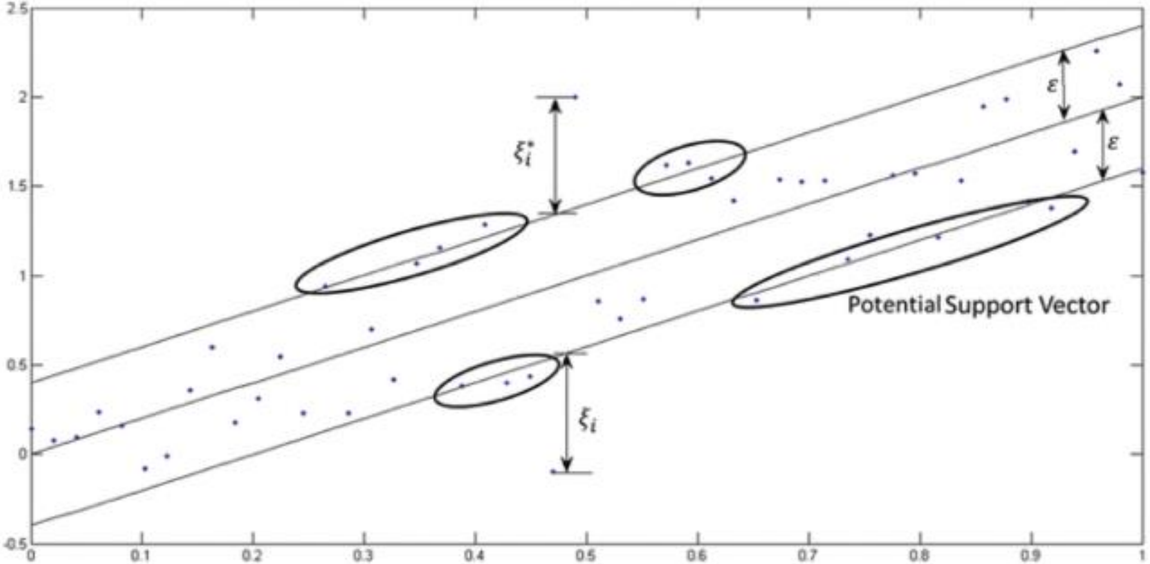
Support Vector Regression - SVR

- Support Vector Machines support linear and nonlinear regression that we can refer to as SVR
- Instead of trying to fit the largest possible street between two classes while limiting margin violations, SVR tries to fit as many instances as possible on the street while limiting margin violations.
- The width of the street is controlled by a hyper parameter Epsilon.

Support Vector Regression - SVR

- SVR performs linear regression in a higher (dimensional space).
- We can think of SVR as if each data point in the training represents it's own dimension. When you evaluate your kernel between a test point and a point in the training set the resulting value gives you the coordinate of your test point in that dimension.
- The vector we get when we evaluate the test point for all points in the training set, $\vec{k}$ is the representation of the test point in the higher dimensional space.
- Once you have that vector you the use it to perform a linear regression.

Support Vector Regression - SVR



https://link.springer.com/chapter/10.1007/978-1-4302-5990-9_4

Support Vector Regression - SVR

It requires a training set: $\mathcal{T} = \{\vec{X}, \vec{Y}\}$ which covers the domain of interest and is accompanied by solutions on that domain.

The work of the SVM is to approximate the function we used to generate the training set

$$F(\vec{X}) = \vec{Y}$$

Support Vector Regression - SVR

Building a SVR

1. Collect a training set $\tau = \{\vec{X}, \vec{Y}\}$

2. Choose a kernel and it's parameters as well as any regularization needed.

3. Form the correlation matrix, $\vec{K}$

4. Train your machine, exactly or approximately, to get contraction coefficients $\vec{\alpha} = \{\alpha_i\}$

5. Use those coefficients, create your estimator $f(\vec{X}, \vec{\alpha}, x^\star) = y^\star$

Support Vector Regression - SVR

Next step is to choose a kernel

- Gaussian

Regularization

- Noise

Support Vector Regression - SVR

Correlation Matrix

$$K_{i,j} = \exp\left(\sum_k \theta_k \left|x_k^i - x_k^j\right|^2\right) + \epsilon\delta_{i,j}$$

Support Vector Regression - SVR

The main part of the algorithm $\bar{K}\vec{\alpha} = \vec{y}$

$\vec{y}$ is the vector of values corresponding to your training set,

$\bar{K}$ is your correlatiōn matrix

$\vec{\alpha}$ is a set of unknowns we need to solve for.

$$\vec{\alpha} = \bar{K}^{-1}\vec{y}$$

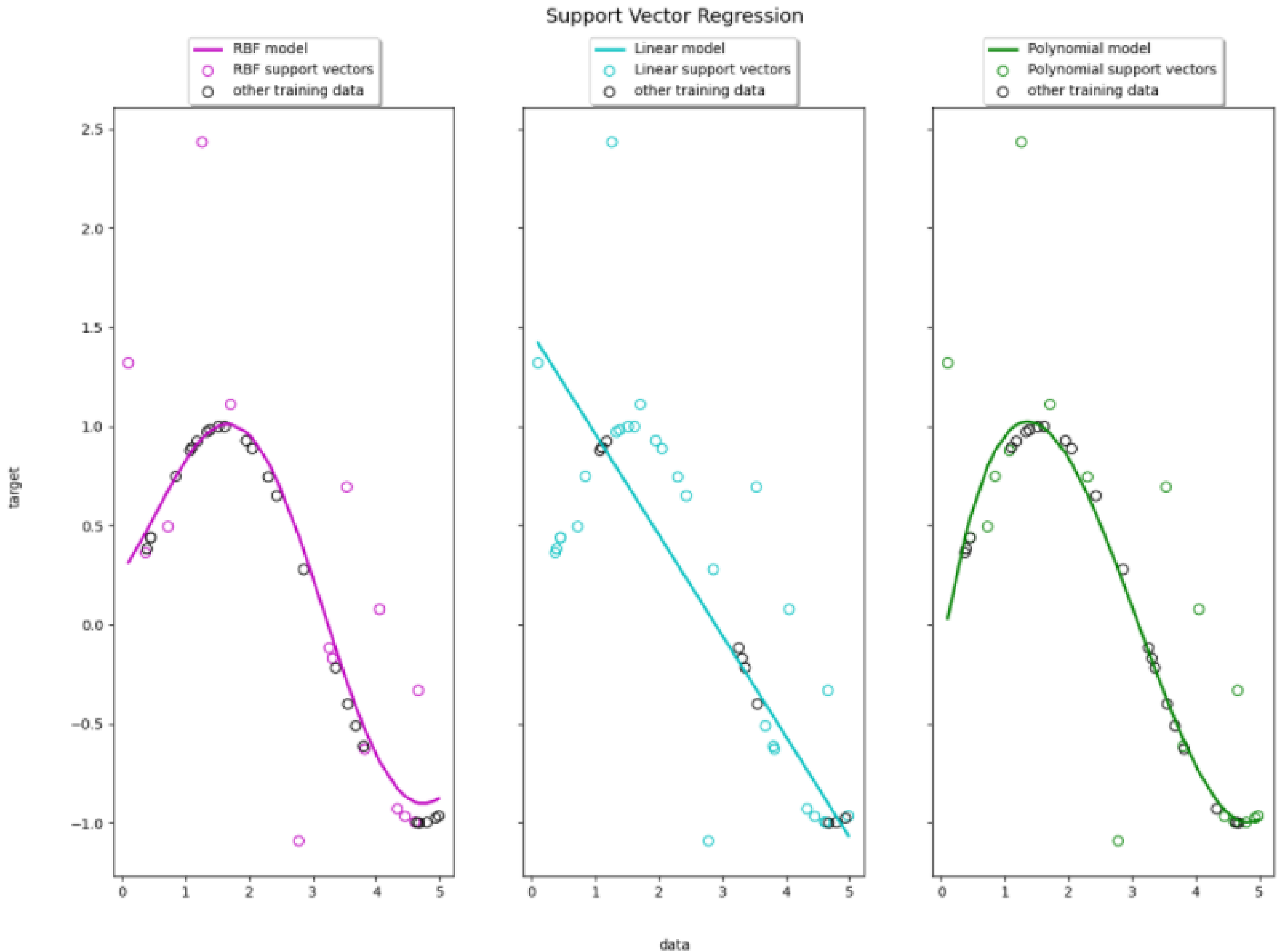
Support Vector Regression - SVR

- Once $\vec{\alpha}$ parameters known - form the estimator
- we use the coefficients we found during the optimization step and the kernel we started off with.

- To estimate the value y* for a test point, $\vec{x^*}$ - compute the correlation vector $\vec{k}$,

- $y^* = \vec{a} \cdot \vec{k}$

$$k_i = \exp\left(\sum_k \theta_k |x_k^i - x_k^\star|^2 \right)$$



Support Vector Regression
RBF model
RBF support vectors
other training data
Linear model
Linear support vectors
other training data
Polynomial model
Polynomial support vectors
other training data
target
data
2.5
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
0
1
2
3
4
5

DataSet and Challenge

Position	Level	Salary
Business Analyst	1	45000
Junior Consultant	2	50000
Senior Consultant	3	60000
Manager	4	80000
Country Manager	5	110000
Region Manager	6	150000
Partner	7	200000
Senior Partner	8	300000
C-level	9	500000
CEO	10	1000000

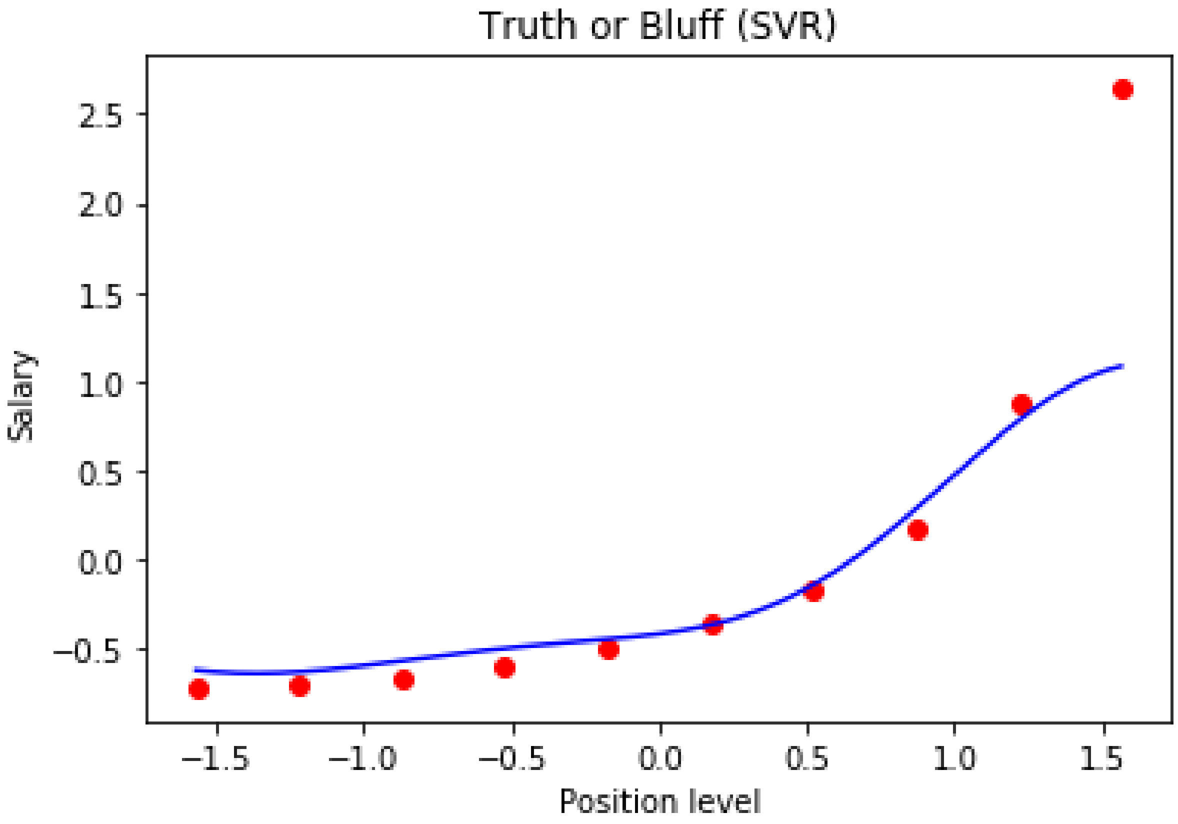
- Determine if the employee is telling the truth or not. We classified the employee like level 6.5 and we need to find out if the salary is 160.000 pounds per year ?

```python
# SVR

# Importing the libraries
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd

# Importing the dataset
dataset = pd.read_csv('Position_Salaries.csv')
X = dataset.iloc[:, 1:2].values
X =X.reshape(-1,1)
y = dataset.iloc[:, 2].values
y =y.reshape(-1,1)

# Splitting the dataset into the Training set and Test set
"""from sklearn.cross_validation import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2, random_state = 0)"""

# Feature Scaling
from sklearn.preprocessing import StandardScaler
sc_X = StandardScaler()
```

```python
sc_y = StandardScaler()
X = sc_X.fit_transform(X)
y = sc_y.fit_transform(y)

# Fitting SVR to the dataset
from sklearn.svm import SVR
regressor = SVR(kernel = 'rbf')
regressor.fit(X, y)

# Predicting a new result
y_pred = regressor.predict([[6.5]])
y_pred = sc_y.inverse_transform(y_pred)

# Visualising the SVR results
plt.scatter(X, y, color = 'red')
plt.plot(X, regressor.predict(X), color = 'blue')
plt.title('Truth or Bluff (SVR)')
plt.xlabel('Position level')
plt.ylabel('Salary')
plt.show()
```

- # Visualising the SVR results (for higher resolution and smoother curve)
- X_grid = np.arange(min(X), max(X), 0.01) # choice of 0.01 instead of 0.1 step because the data is feature scaled
- X_grid = X_grid.reshape((len(X_grid), 1))
- plt.scatter(X, y, color = 'red')
- plt.plot(X_grid, regressor.predict(X_grid), color = 'blue')
- plt.title('Truth or Bluff (SVR)')
- plt.xlabel('Position level')
- plt.ylabel('Salary')
- plt.show()

SVR


Truth or Bluff (SVR)
Salary
2.5
2.0
1.5
1.0
0.5
0.0
−0.5
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
Position level

Muito obrigado!