Machine Learning
Practical -2
Support Vector Machine (SVM)

Lecturer:
Dr Hamid Khayyam (Australia)
Email: hamid.khayyam@rmit.edu.au

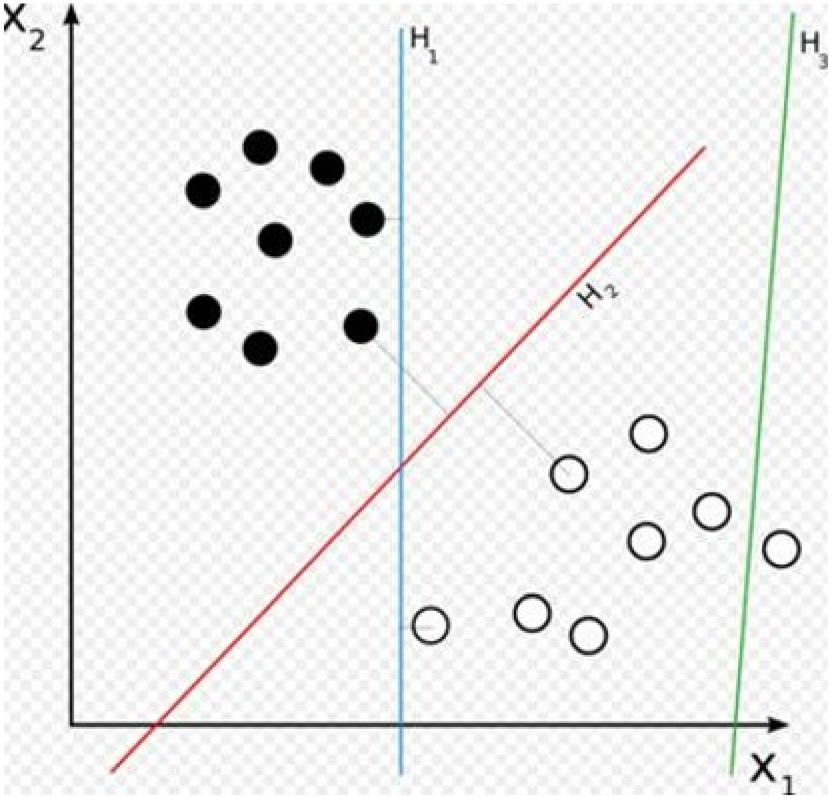
RMIT UNIVERSITY

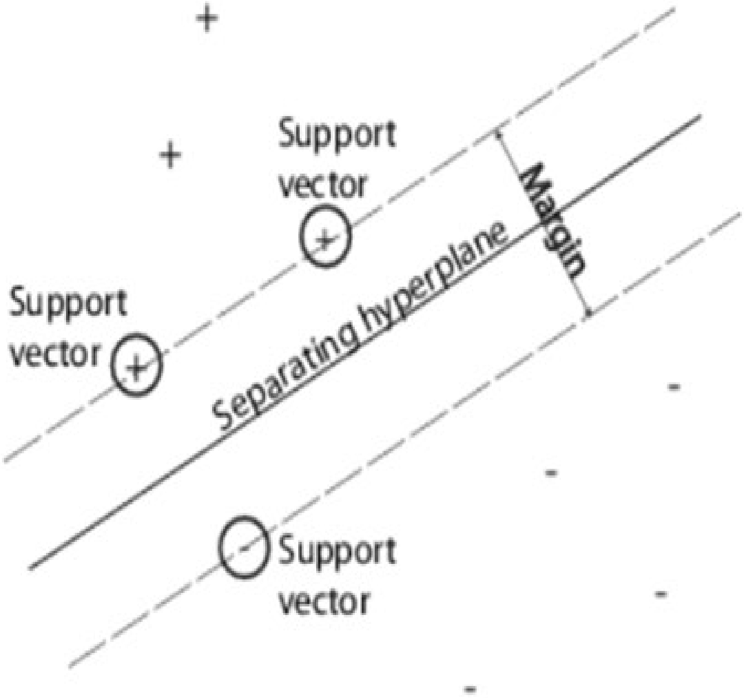
The aim of Machine learning methods (recap)

- To find a relationship between some inputs and outputs from different engineering problems when the model is unknown (black box modelling).

- To build mathematical models of engineering systems from observed input–output data (system identification) regardless of what the inputs and outputs are and make predictions based on some unseen new inputs.

Support vector Machine (SVM) (Recap)





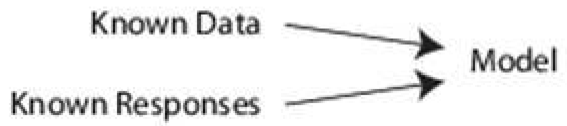
Steps of An Application of SVM (Recap)

1. Data pre-processing (check for missing data ,standardization)

2. Model development and training

3. Mdl = fitcsvm(X, Y) (Classification)   % Mdl is the developed model

   Mdl = fitrsvm(X, Y) (Regression)



3. Simulation (prediction)

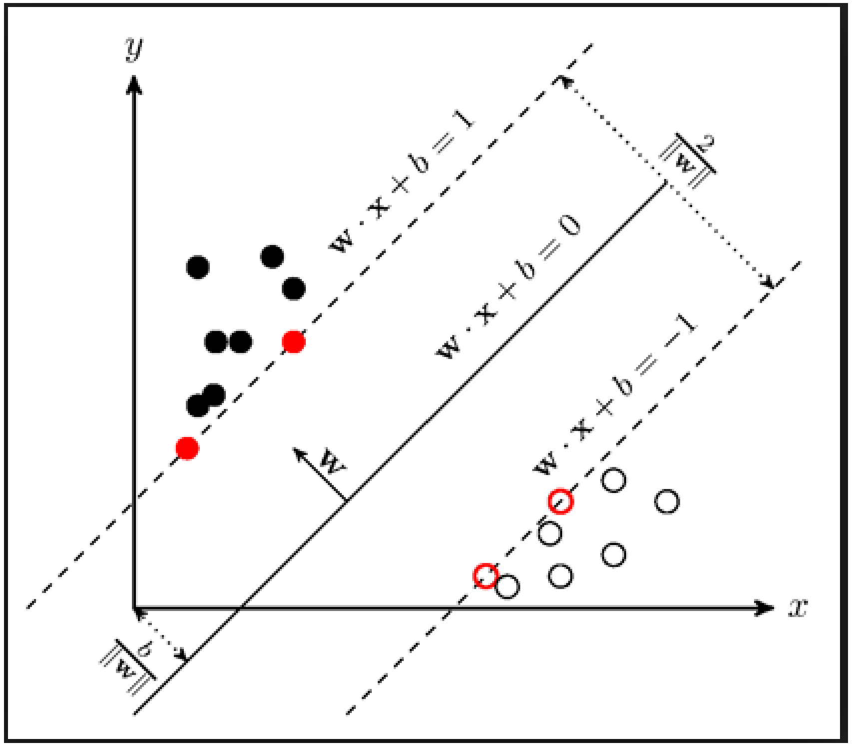
   label= predict(Mdl,X) (classification)   % label : predicted labels

   Y_predicted= predict(Mdl,X) (Regression) %Y_predicted is the predicted responses

4. Post-processing

   - MSE,RMSE,R (Regression)

Support vector Machine (SVM) (Recap)



$$D = 2d = \frac{2}{\|W\|}$$

Training (solving) hard-margin problem in Matlab:

$$\text{Minimize} \quad L = \frac{1}{2} W^{\mathrm{T}} W$$

Subject to

$$y\left(W^{\mathrm{T}} X + b\right) - 1 \geq 0$$

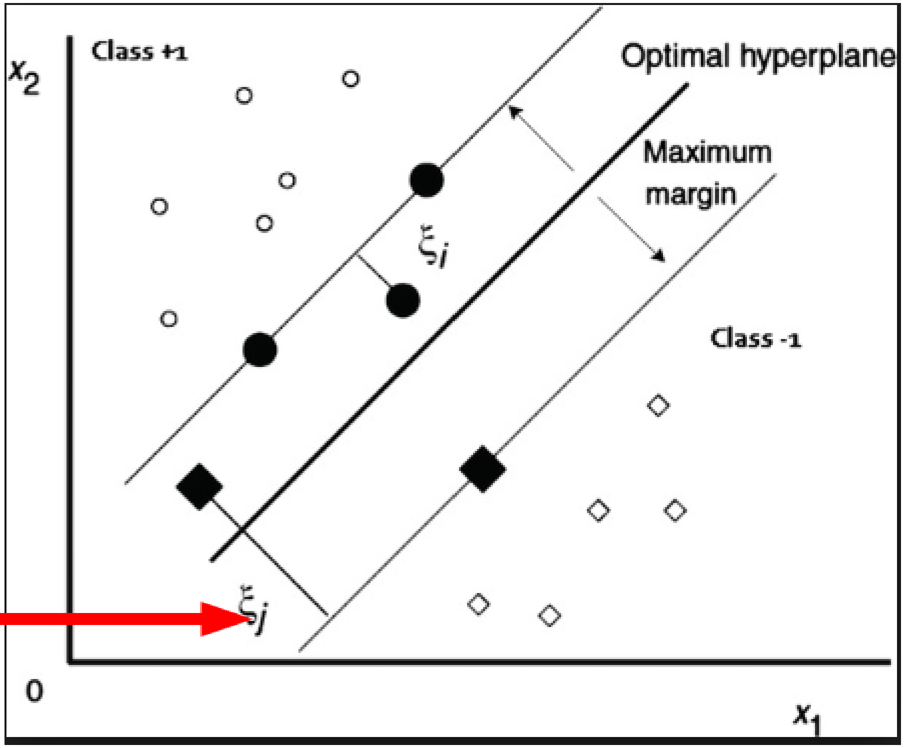
$$\begin{aligned} &\text{if} \quad y = 1 \quad \text{then} \quad W^{\mathrm{T}} X + b \geq 1 \\ &\text{if} \quad y = -1 \quad \text{then} \quad W^{\mathrm{T}} X + b \leq -1 \end{aligned}$$

Support vector Machine (SVM) (Recap)

- **Non separable data (soft-margin)**

.

Error



- **Training(solving) Soft-margin problem in Matlab :**

$$\text{Minimize} \quad \frac{1}{2}W^{\mathrm{T}}W + C\sum_{i=1}^{n}\xi_i \quad i = 1, \ldots, n$$

$$\begin{aligned} \text{Subject to :} \quad & y_i\left(W^{\mathrm{T}}X + b\right) \geq 1 - \xi_i, \quad && \forall i \in \{1, \ldots, n\} \\ & \xi_i \geq 0, && \forall i \in \{1, \ldots, n\} \end{aligned}$$

Support vector Machine (SVM) (Recap)

Solution (Lagrangian multiplier) :

$$\text{Minimize} \quad L = \frac{1}{2}W^{\mathrm{T}}W - \sum_i \alpha_i\left[y\left(W^{\mathrm{T}}X + b\right) - 1\right] \quad i = 1, \ldots, n$$

α : the multiplier of the constraint.

Primal problem of SVM method

$$\begin{cases} \dfrac{dL}{dW} = 0 \Rightarrow W - \sum_i \alpha_i y_i x_i \Rightarrow W = \sum \alpha_i y_i x_i \\ \dfrac{dL}{db} = 0 \Rightarrow \sum_i \alpha_i y_i = 0 \end{cases}$$

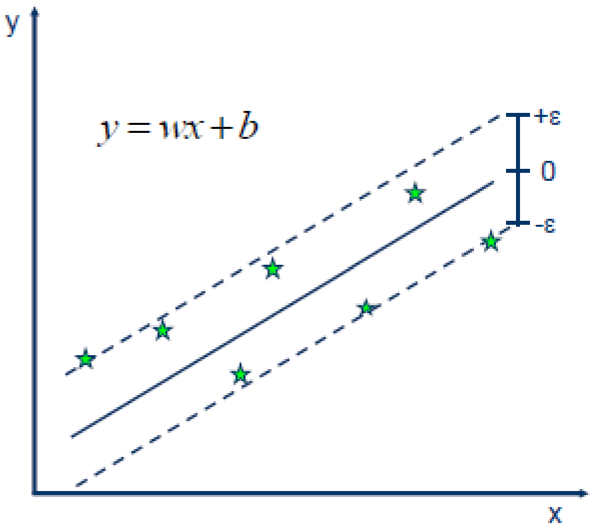
Dual problem of SVM method

$$\text{Maximize} \quad L_D = -\frac{1}{2}\sum_i \sum_j \alpha_i \alpha_j y_i y_j x_i^{\mathrm{T}} x_j + \sum_i \alpha_i \quad i = 1, \ldots, n$$

$$\text{Subject to} \quad \sum_i \alpha_i y_i = 0 \quad \alpha_i \geq 0$$

Support vector Machine (SVM) (Recap)

- **Support Vector Machine - Regression (SVR)**



- Solution:

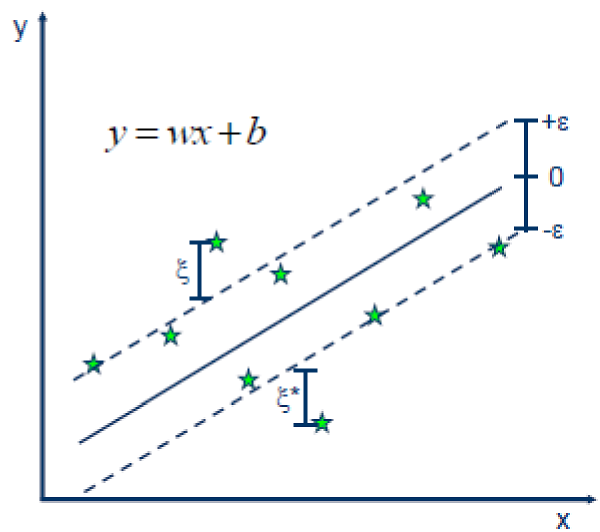
$$\min \frac{1}{2}\|w\|^2$$

- Constraints:

$$y_i - wx_i - b \le \varepsilon$$
$$wx_i + b - y_i \le \varepsilon$$

Hard-Margin Solution

ε: **Margin of tolerance**



- Minimize:

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

- Constraints:

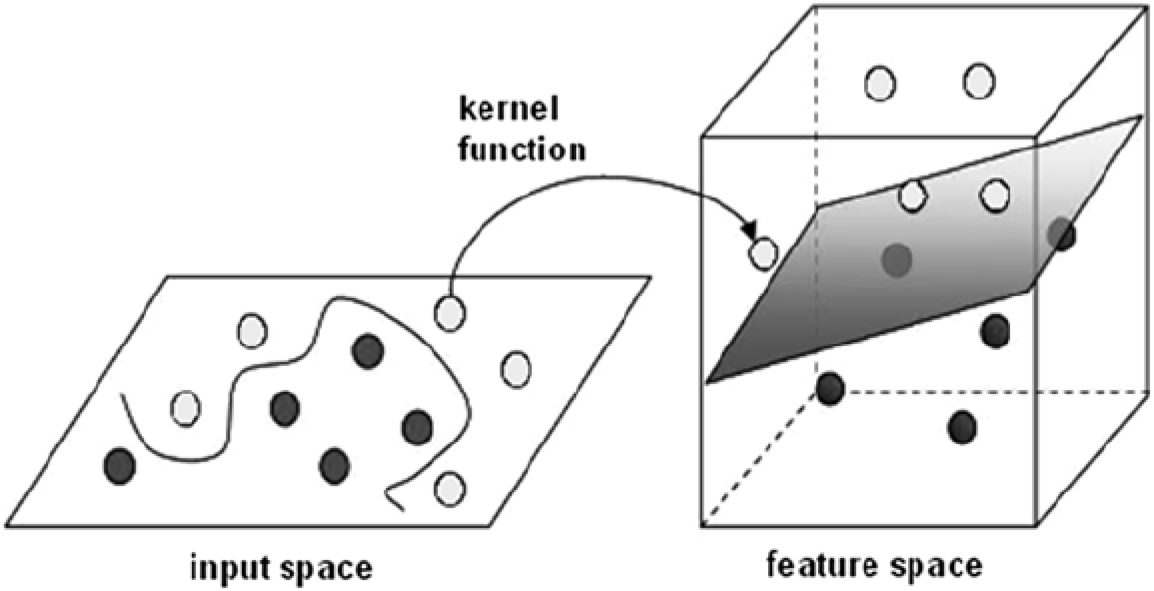
$$y_i - wx_i - b \le \varepsilon + \xi_i$$
$$wx_i + b - y_i \le \varepsilon + \xi_i^*$$
$$\xi_i, \xi_i^* \ge 0$$

Soft-Margin Solution

Linear SVR: $y = \sum_{i=1}^{N}\left(\alpha_i - \alpha_i^*\right)\cdot\langle x_i, x\rangle + b$

Support vector Machine (SVM) (Nonlinear)

- **Kernel Trick (Nonlinear SVM)**



Training (solving) problem with Kernel function in Matlab:

$$\text{Maximize} \quad \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j . k(x_i, x_j)$$

$$\text{Subject to :} \quad \alpha_i \geq 0, \quad \forall i \in \{1, \ldots, n\}$$

$$\sum_{i=1}^{n} \alpha_i y_i = 0$$

$$k(x_i, x_j) = (\phi(x_i) . \phi(x_j))$$

Support vector Machine (SVM) (cont.)

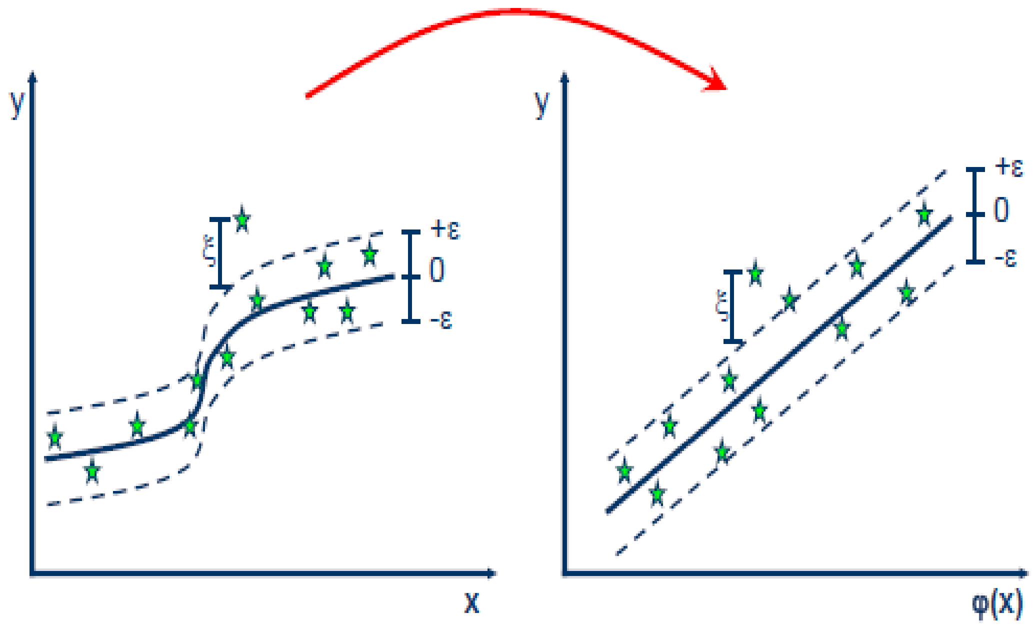
- **Kernel functions**
- Linear $$k(x_i x_j) = x_i^{\mathrm{T}} x_j$$
- Polynomial $$k(x_i x_j) = (\gamma \ x_i^{\mathrm{T}} x_j + r)^d, \quad \gamma > 0$$
- RBF(Radial basis function) $$k(x_i x_j) = \exp\left(-\gamma \|x_i - x_j\|^2\right), \quad \gamma > 0$$

where, γ, r, and d are *kernel* parameters.

Support vector Machine (SVM) (cont.)

- **Nonlinear SVR**

$$y = \sum_{i=1}^{N} \left(\alpha_i - \alpha_i^*\right) \cdot K\left(x_i, x\right) + b$$



Support vector machine model: fitcsvm

- **fitcsvm**

Syntax

Mdl = fitcsvm(x, t)   %trains a two-class (binary) classification .

Example :

```
Mdl=fitcsvm(x,t);
```

- **Predict**

Syntax

ylabel = predict(Mdl,x) (Regression)

Example :

```
y_predicted=predict(mdl,x);
```

Support vector machine model: fitcsvm (cont.)

fitcsvm additional options:

- 'Standardize': false | true (Default: false) % Standardize data
- 'Solver': 'ISDA' | 'L1QP' | 'SMO' (Default: SMO) % Solver for objective functions
- KernelFunction

  'gaussian' or 'rbf': Gaussian or Radial Basis Function (RBF) kernel

  'linear': Linear kernel (default)

  'polynomial': Polynomial kernel % Use 'PolynomialOrder', q, to specify a polynomial kernel of order q.
- 'PolynomialOrder': positive integer (Default:3)
- 'KernelScale': 1 (default) | 'auto' | positive scalar % gamma in RBF kernel
- 'BoxConstraint': positive scalar (Default:1) % C ,the cost of misclassification

Support vector machine model: fitcsvm

- **fitrsvm**

Syntax

Mdl = fitrsvm(x, t) %trains a two-class (binary) classification .

Example :

```
Mdl = fitrsvm(x,t);
```

- **Predict**

Syntax

y= predict(Mdl,x) (Regression)

Example :

```
y_predicted=predict(mdl,X);
```

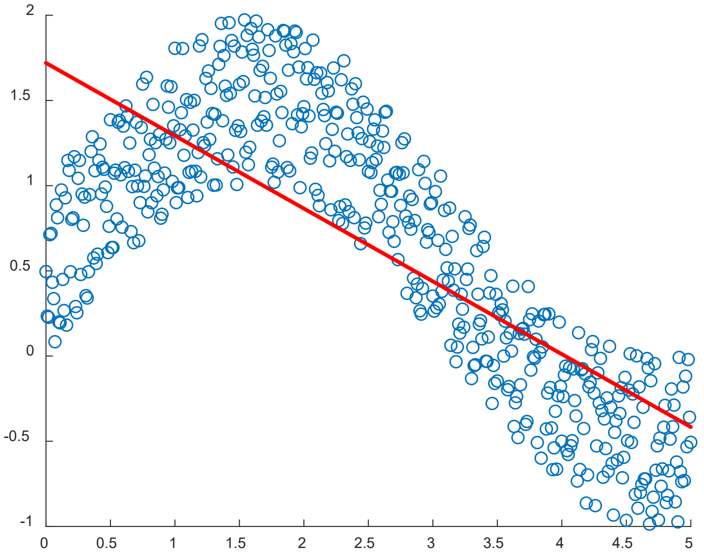
Support vector regression (SVR): fitrsvm (cont.)

fitrsvm additional options:

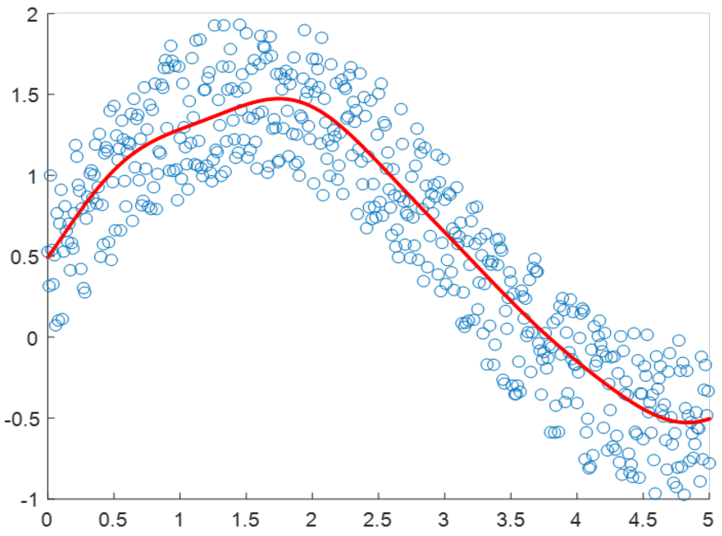
- 'Standardize': false | true (Default:false)
- 'Solver': 'ISDA' | 'L1QP' | 'SMO' (Default: SMO)
- KernelFunction:
  - 'gaussian' or 'rbf': Gaussian or Radial Basis Function (RBF) kernel
  - 'linear': Linear kernel (default)
  - 'polynomial': Polynomial kernel. % Use 'PolynomialOrder', q, to specify a polynomial kernel of order q.
- 'BoxConstraint': positive scalar (Default:1) % C the cost of wrong prediction
- 'KernelScale': 1 (default) | 'auto' | positive scalar % gamma in RBF kernel

Example of regression (Linear kernel): fitrsvm

```
clear;

clc;

rng(1);

x = 0:0.01:5 ;

t = sin(x)+rand(1, length(x)) ;

x = x' ;

t = t' ;

Mdl = fitrsvm(x,t,'Standardize',true);

y= predict(Mdl,x);%y is the predicted output based on model Mdl and input x

scatter(x,t); % Scatter plot

hold on

plot(x,y,'r.')
```



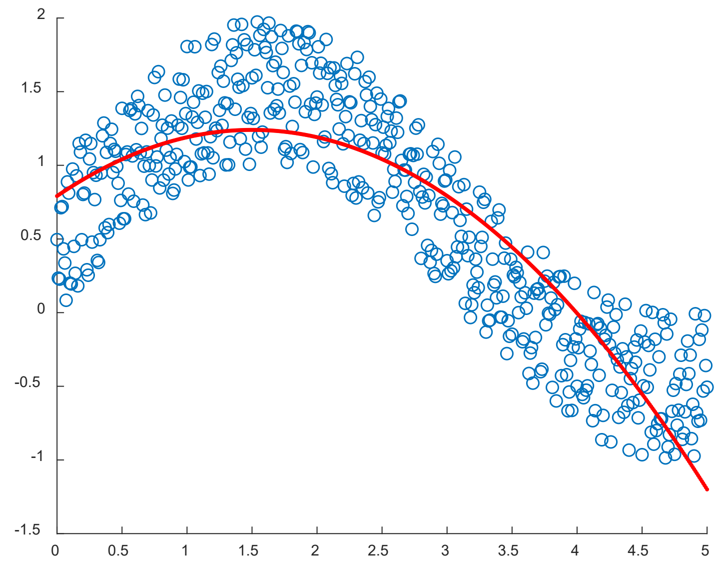
Example of regression (Gaussian kernel): fitrsvm




```
clear;clc;

rng(1);

x = 0:0.01:5 ;

t = sin(x)+rand(1, length(x)) ;

x = x' ;

t = t' ;

Mdl = fitrsvm(x,t,'KernelFunction','gaussian','Standardize',true);

y= predict(Mdl,x);%y is the predicted output based on model Mdl and input x

scatter(x,t); % Scatter plot

hold on

plot(x,y,'r.')
```

Example of regression (Polynomial kernel): fitrsvm

```
clear;clc;

rng(1);

x = 0:0.01:5 ;

t = sin(x)+rand(1, length(x)) ;

x = x' ;

t = t' ;

Mdl =
fitrsvm(x,t,'KernelFunction','polynomial','polynomialorder',2,'Stand
ardize',true);

y= predict(Mdl,x);%y is the predicted output based on
model Mdl and input x

scatter(x,t); % Scatter plot

hold on

plot(x,y,'r.')
```



Example of classication-1: fitcsvm

```
clear ;

clc;

load ionosphere.mat

rng(1);  % random number generation to reproduce results

 mdl = fitcsvm(X,Y, 'KernelFunction', 'gaussian', ...

    'KernelScale','auto',...

    'BoxConstraint', 1, ...

    'Solver','L1QP',...

    'Standardize', true);% use fitcsvm with gaussian
kernel and solver L1QP

 y_expected=predict(mdl,X);

table( Y( 20:30 ),y_expected( 20:30 ), 'VariableNames',...

    {' TrueLabel',' PredictedLabel'}) %Show the results of
20th to 30th data of the output and predicted output.
```

12×2 table

TrueLabel	PredictedLabel
'b'	'b'
'g'	'g'
'b'	'g'
'g'	'g'
'b'	'b'
'g'	'g'
'b'	'b'
'g'	'g'
'b'	'b'
'g'	'g'
'b'	'b'
'g'	'g'

Example of regression-1: fitrsvm

```
clear;clc;
rng(1);
Filename='SVR1.xlsx';
Sheetread='x';
Input1='A1:M252';
Sheetread1='t';
output1='A1:A252';
Input=xlsread(Filename,Sheetread,Input1); %Read Microsoft Excel
Target=xlsread(Filename,Sheetread1,output1 );
x=Input;
t=Target;
```

Example of regression-1: fitrsvm (cont.)

```
mdl = fitrsvm(x,t, 'KernelFunction', 'polynomial', ...
    'polynomialorder',2,'Standardize', true); %To
standardize the data and use polynomial function as the
kernel with order 2.

yfit=predict(mdl,x); % prediction based on the developed
SVR model and x as the input.

table(t(40:50,:),yfit(40:50,:),'VariableNames',{'ObservedV
alue',' PredictedValue'}) % show 40th to 50th data in
output and predicted output

MSE_training=sum((yfit-t).^2)/numel(t); % Calculate MSE
for training data

RMSE_training=sqrt(sum((yfit-t).^2)/numel(t)); % Calculate
RMSE for training data
```

Example of regression-1: fitrsvm (cont.)

```
ans =

  11×2 table

    ObservedValue    PredictedValue
    _____________    ______________

        32.6             31.644
        34.5             35.489
        32.9             33.888
        31.6             32.131
          32             30.394
         7.7             8.6728
        13.9             14.648
        10.8             9.8293
         5.6             7.0346
        13.6             14.864
           4             4.0518
```

MSE training = 11.9979

RMSE training = 3.4638

Example of regression-2: fitrsvm

```
clear;clc;
rng(1);
Filename='SVR2.xlsx';
Sheetread='x';
Input1='A1:A94';
Sheetread1='t';
output1='A1:A94';
Input=xlsread(Filename,Sheetread,Input1); %Read Microsoft Excel
Target=xlsread(Filename,Sheetread1,output1 );
x=Input;
t=Target;
```

Example of regression-2: fitrsvm (cont.)

```
mdl = fitrsvm(x,t, 'KernelFunction', 'gaussian', ...
    'Solver','L1QP',...
    'Standardize', true); %standardize the data and use
Gaussian kernel.

yfit=predict(mdl,x); % prediction based on the developed
SVR model and x as the input

table(t(20:30,:),yfit(20:30,:),'VariableNames',{'ObservedV
alue',' PredictedValue'}) % show 20th to 30th data in
output and predicted output

MSE_training=sum((yfit-t).^2)/numel(t); % Calculate MSE
for data ;numel :number of elements

RMSE_training=sqrt(sum((yfit-t).^2)/numel(t)); % Calculate
RMSE for data ;numel :number of elements
```

Example of regression-2: fitrsvm (cont.)

```
ans =

  11×2 table

    ObservedValue    PredictedValue
    _____________    ______________

    9.8589           9.3745
    9.6876            9.347
    9.4722           9.2794
    9.2283           9.1763
    8.9701           9.0433
    8.7099           8.8865
    8.4579           8.7125
    8.2217           8.5285
    8.0065           8.3412
    7.8153           8.1577
    7.6494           7.9841
```

MSE_training = 0.6481

RMSE_training = 0.8051

Example of regression-3: fitrsvm

```
clear;clc;

rng(1);

Filename='SVR3.xlsx';

Sheetread='Sheet1';

Input1='A1:B89';

output1='C1:C89';

Input=xlsread(Filename,Sheetread,Input1); %Read Microsoft Excel

Target=xlsread(Filename,Sheetread,output1 );

Sheetread1='Sheet2';

Input2='A1:B11';

Target2 ='C1:C11';

Inputnew=xlsread(Filename,Sheetread1,Input2);

Targetnew=xlsread(Filename,Sheetread1,Target2 );
```

Example of regression-3: fitrsvm (cont.)

```
x=Input;

t=Target;

xnew=Inputnew;

tnew=Targetnew;

x=fillmissing(x,'spline'); %fill in the missing input data

t= fillmissing(t,'spline'); %fill in the missing output
data

mdl=fitrsvm(x,t,'Standardize',true,'KernelFunction','gauss
ian','epsilon',0.3); %standardize the data and use
gaussian kernel to develope and model the data

yfit=predict(mdl,x); % prediction based on the developed
SVR model and x as the input

MSE_training=sum((yfit-t).^2)/numel(t); % Calculate MSE
for data
```

Example of regression-3: fitrsvm (cont.)

```
RMSE_Training=sqrt(sum((yfit-t).^2)/numel(yfit)); % Calculate RMSE for training data

table(t(60:70,:),yfit(60:70,:),'VariableNames',{'ObservedValue',' PredictedValue'}) % show 60th to 70th data in output and predicted output

ynew=predict(mdl,xnew);% prediction based on new data

MSE_testing=sum((ynew-tnew).^2)/numel(tnew); % Calculate MSE for new data

RMSE_Testing=sqrt(sum((ynew-tnew).^2)/numel(ynew)); % Calculate RMSE for new data
```

Example of regression-3: fitrsvm (cont.)

Without kernel (linear kernel)

```
11×2 table

ObservedValue      PredictedValue
_____________      ______________

    26.5               26.211
      20               21.763
      13               15.464
      19               21.642
      19               22.758
    16.5               17.312
    16.5               11.827
      13               15.017
      13                16.64
      13               16.498
      28               25.359
```

RMSE_training = 3.5404

RMSE_testing   = 7.7213

Example of regression-3: fitrsvm (cont.)

With kernel (Gaussian kernel)

```
11×2 table

ObservedValue    PredictedValue
_____________    ______________

26.5             28.769
  20             20.415
  13             14.942
  19             20.878
  19             19.911
16.5             16.355
16.5             15.869
  13             14.975
  13             15.262
  13             14.873
  28             25.599
```

RMSE_training = 2.8585

RMSE_testing = 7.5712

Example of regression-4: fitrsvm

```
clear;clc;
rng(1);
Filename='SVR4.xlsx';
Sheetread='Sheet1';
Input1='A1:H72';
output1='I1:I72';
Input=xlsread(Filename,Sheetread,Input1); %Read Microsoft Excel
Target=xlsread(Filename,Sheetread,output1 );
x=Input;
t=Target;
Sheetread1='Sheet2';
Input2='A1:H3';
Target2 ='I1:I3';
```

Example of regression-4: fitrsvm (cont.)

```
Inputnew=xlsread(Filename,Sheetread1,Input2);

Targetnew=xlsread(Filename,Sheetread1,Target2 );

xnew=Inputnew;

tnew=Targetnew;

mdl = fitrsvm(x,t, 'KernelFunction', 'gaussian', ...
        'Standardize', true); %standardize the data
%standardize the data and use gaussian kernel to develope
and model the data

conv = mdl.ConvergenceInfo.Converged; % Shows whether the
program reach an answer

iter = mdl.NumIterations; % number of iteration to reach
the answer

yfit=predict(mdl,x); % prediction based on the developed
SVR model and x as the input
```

Example of regression-4: fitrsvm (cont.)

```
table(t(20:30,:),yfit(20:30,:),'VariableNames',{'ObservedValue',' PredictedValue'}) % show 20th to 30th data in output and predicted output

MSE_training=sum((yfit-t).^2)/numel(t); % Calculate MSE for data

RMSE_training=sqrt(sum((yfit-t).^2)/numel(t)); % Calculate RMSE for data

ynew=predict(mdl,xnew);

table(tnew(:),ynew(:),'VariableNames',{'ObservedValue_Newdata',' PredictedValue_newdata'}) % show data in output and predicted output

MSE_testing=sum((tnew-ynew).^2)/numel(tnew); % Calculate MSE for new data

RMSE_testing=sqrt(sum((tnew-ynew).^2)/numel(tnew)); % Calculate RMSE for new data

Errorpercentage=((ynew-tnew)./tnew)*100; % Calculate error percentage for tnew and ynew
```

Example of regression-4: fitrsvm (cont.)

Without kernel (linear kernel)

11×2 table

ObservedValue	PredictedValue
514	513.96
518	517
517	516.09
517	517.04
515	514.48
511	511.61
511	512.02
516	512.24
515	512.21
514	512.38
515	515.57

ans =

3×2 table

ObservedValue_Newdata	PredictedValue_newdata
495	495.46
498	497.53
498	497.03

Errorpercentage

3x1 double

	1	2	3
1	0.0934		
2	-0.0951		
3	-0.1942		
4			
5			
6			

MSE_training = 3.8295

MSE_testing = 0.4576

RMSE_training = 1.9569

RMSE_testing = 0.6765

Example of regression-4: fitrsvm (cont.)

With kernel (linear kernel)

```
11×2 table

  ObservedValue    PredictedValue
  _____________    ______________

  514                 513
  518                 517
  517                 516
  517                 516
  515                 514
  511              511.18
  511                 512
  516                 515
  515                 514
  514              513.47
  515                 514
```

MSE_training =2.1296 RMSE_training = 1.4593

MSE_testing=4.4638 RMSE_testing = 2.1128

Example of regression-4: fitrsvm (cont.)

```
ans =

  3×2 table

    ObservedValue_Newdata    PredictedValue_newdata
    _____________________    ______________________

    495                      496.37
    498                      496.21
    498                      500.88
```

Errorpercentage

3x1 double

	1	2	3
1	0.2768		
2	-0.3601		
3	0.5785		
4			
5			

SVM References

- https://au.mathworks.com
- S. Araghinejad, Data-Driven Modeling: Using MATLAB® in Environmental Engineering
- http://www.saedsayad.com/support_vector_machine_reg.htm
- https://digitaltransformationpro.com/data-mining-steps/