

Klasifikasi SVM pada MATLAB

Pendefinisian SVM pada matlab digunakan function templateSVM dengan syntax sebagai berikut:

SYNTAX

```
t = templateSVM()  
t = templateSVM(Name,Value)
```

- o Name, Value, merupakan sepasang argumen dimana name adalah argumen dan value adalah nilai yang sesuai dengan name. Penentuan argumen ini dapat dilakukan lebih dari satu pasang dengan tanpa memperhatikan urutan seperti Name, Value, Name1, Value1 ... NameN, ValueN. Berikut beberapa contoh parameter:

- a. 'KernelFunction' digunakan untuk memilih fungsi kernel pada klasifikasi SVM. Jenis fungsi kernel ditunjukkan pada Tabel

Linear	Kernel jenis ini digunakan secara default pada data dua kelas
Polynomial	Untuk menentukan order pada polynomial dapat digunakan parameter 'PolynomialOrder'
RBF	Kernel jenis ini digunakan secara default pada <i>one-class learning</i>

Contoh penulisan: 'KernelFunction', 'RBF'

- b. 'Standardize' digunakan untuk membakukan kelas data. Tipe data dalam kelas ini berbentuk logical yaitu true(1) dan false(0).

Contoh penulisan: 'Standardize', 1

Setelah didefinisikan, selanjutnya dilakukan proses training menggunakan *function* `fitcecoc` dengan *syntax* sebagai berikut:

SYNTAX

```
Mdl = fitcecoc(Tbl, ResponseVarName)
Mdl = fitcecoc(Tbl, formula)
Mdl = fitcecoc(Tbl, Y)
Mdl = fitcecoc(X, Y, Name, Value)
```

- Tbl, merupakan sampel data untuk melatih model. Tbl ditentukan sebagai tabel. Setiap baris Tbl merupakan data dari satu pengamatan dan tiap kolom merupakan satu dari variabel prediktor. Tbl dapat diberi kolom tambahan yang berisi variabel respons. Pada Tbl tidak diperbolehkan variabel multikolom dan array sel selain array sel vektor karakter. Jika Tbl berisi variabel respons dan semua variabel yang tersisa pada Tbl sebagai variabel prediktor, maka penentuan variabel respons dapat menggunakan `ResponseVarName`. Jika Tbl berisi variabel respons dan sebagian variabel yang tersisa pada Tbl sebagai variabel prediktor, maka penentuan variabel respons dapat menggunakan `formula`. Dan jika Tbl tidak memiliki variabel respons, maka dapat ditentukan dengan variabel Y. Panjang dari variabel respons harus sama dengan jumlah baris.
- `ResponseVarName`, merupakan nama variabel respons pada Tbl. Variabel respons harus berupa kategori, vektor logis, array string, vektor logis, vektor numerik, vektor karakter atau skalar string.
- `formula`, merupakan model penjelasan respons dan subset dari variabel prediktor, ditentukan sebagai vektor karakter atau skalar string dalam bentuk 'Y~X1+X2+X3'. Dalam bentuk ini Y mewakili variabel respons dan X1, X2, dan X3 mewakili variabel prediktor.
- X, merupakan variabel prediktor yang berupa matriks numerik berukuran MxN. Setiap baris X merupakan data dari satu pengamatan dan tiap kolom merupakan satu dari variabel prediktor.

- Y, merupakan label kelas yang ditentukan sebagai vektor numerik, vektor *categorical*, vektor logic, array string atau array sel vektor karakter. Y berukuran Mx1, setiap baris dari Y mewakili label kelas yang dimiliki X.
- Name, Value, merupakan sepasang argumen dimana name adalah argumen dan value adalah nilai yang sesuai dengan name. Pada klasifikasi SVM digunakan name 'Leaners' dengan value merupakan variabel yang menyimpan pendefinisian SVM

Setelah proses training, selanjutnya dilakukan proses testing menggunakan *function* predict dengan *syntax* sebagai berikut:

SYNTAX

```
YPred = predict(t,X)
```

- YPred, merupakan label kelas hasil prediksi
- t, merupakan variabel pendefinisian SVM
- X, merupakan data testing

Contoh penggunaan *function* templateSVM pada MATLAB dapat dilakukan menggunakan data peminat SBMPTN program studi sains (matematika, fisika, kimia, biologi) pada tahun 2017 dengan data diperoleh dari web resmi SBMPTN. Data tersebut akan diklasifikasikan menjadi tiga kelas yaitu tinggi, sedang dan rendah. Dimana kelas 1 berarti tinggi, 2 berarti sedang, dan tiga yang berarti rendah. Data tersebut akan ditampilkan pada Tabel

No.	NAMA PTN	MAT	FIS	KIM	BIO	Kelas
1.	Universitas Indonesia	1.083	761	843	1.080	1
2.	Universitas Islam Negeri Jakarta	464	342	404	666	2
3.	Universitas Negeri Jakarta	685	260	452	930	1
4.	Institut Pertanian Bogor	592	388	560	750	1
5.	Universitas Islam Negeri Gunung Djati	423	160	429	613	2
6.	Universitas Padjajaran	1.014	655	1.110	1.338	1
7.	Universitas Pendidikan Indonesia	612	260	695	853	1
8.	Universitas Diponegoro	645	420	656	797	1
9.	Universitas Islam Negeri Walisongo	120	59	102	211	3
10.	Universitas Jenderal Soedirman	743	342	515	1.126	1

11.	Universitas Negeri Semarang	460	249	516	698	2
12.	Universitas Sebelas Maret	594	463	422	843	1
13.	Universitas Gadjah Mada	663	362	716	965	1
14.	Universitas Islam Negeri Kalijaga	228	100	169	289	3
15.	Universitas Negeri Yogyakarta	871	335	913	1.212	1
16.	Institut Teknologi Sepuluh November	612	477	556	524	1
17.	Universitas Airlangga	557	284	400	586	2
18.	Universitas Brawijaya	689	469	571	867	1
19.	Universitas Islam Negeri Malang	336	158	307	646	2
20.	Universitas Islam Negeri Sunan Ampel Surabaya	249	0	0	406	3
21.	Universitas Jember	390	212	397	566	2
22.	Universitas Negeri Surabaya	553	240	472	713	2
23.	Universitas Negeri Malang	759	391	519	804	1

Data akan disimpan dalam microsoft excell dan akan diolah pada MATLAB, pohon keputusan yang dihasilkan dari klasifikasi data peminat SBMPTN program studi sains tahun 2017 diilustrasikan pada Gambar 3. . Berikut *source code* contoh pohon klasifikasi menggunakan algoritma SVM:

CONTOH

```

clc;clear
data = xlsread('datasbmbaru.xlsx')
datatraining = data(1:floor(0.7*length(data)),1:4);
kelastraining = data(1:floor(0.7*length(data)),5);
datatesting = data(floor(0.7*length(data))+1:end,1:4);
kelastesting = data(floor(0.7*length(data))+1:end,5);

a =
templateSVM('Standardize',1,'KernelFunction','polynomial
');
traini =
fitcecoc(datatraining,kelastraining,'Learners',a);

hasil = predict(traini,datatesting);
cek = [hasil kelastesting]
```

Jika program diatas dijalankan akan menghasilkan tampilan *command window* seperti pada Gambar .

```
Command Window

data =

    1083    761    843    1080     1
     464    342    404     666     2
     685    260    452     930     1
     592    388    560     750     1
     423    160    429     613     2
    1014    655   1110    1338     1
     612    260    695     853     1
     645    420    656     797     1
     120     59    102     211     3
     743    342    515    1126     1
     460    249    516     698     2
     594    463    422     843     1
     663    362    716     965     1
     228    100    169     289     3
     871    335    913    1212     1
     612    477    556     524     1
     557    284    400     586     2
     689    469    571     867     1
     336    158    307     646     2
     249     0      0     406     3
     390    212    397     566     2
     553    240    472     713     2
     759    391    519     804     1

cek =

     2     2
     1     1
     2     2
     3     3
     2     2
     2     2
     1     1

fx
```

Variabel cek merupakan hasil perbandingan kelas hasil prediksi dengan kelas sebenarnya, kolom pertama merupakan kelas hasil prediksi dan kolom kedua merupakan kelas sebenarnya.

Phyton

```
1  import numpy as np
2  import pandas as pd
3  from sklearn import datasets
4  from sklearn.metrics import confusion_matrix
5  from sklearn.metrics import accuracy_score
6  from sklearn.model_selection import train_test_split, cross_val_score
7  from sklearn.svm import SVC
8  #load data
9  iris = datasets.load_iris()
10 df= pd.DataFrame(data= np.c_[iris['data'], iris['target']],
11                  columns= iris['feature_names'] + ['target'])
12
13 df['species'] = pd.Categorical.from_codes(iris.target, iris.target_names)
14 del df['target']
15 print(df)
16 #memisahkan label dan parameter
17 x=df.iloc[:, :-1]
18 y=df.iloc[:, 4]
19 x_train, x_test, y_train, y_test=train_test_split(x, y, test_size=0.30, random_state=100)
20 #pemodelan svm
21 model = SVC()
22 model.fit(x_train, y_train)
23 y_pred = model.predict(x_test)
24 print("hasil prediksi svm")
25 print(y_pred)
26 #evaluasi confusion matrix dan evaluasi akurasi python
27 print("Hasil confusion matrix")
28 print(confusion_matrix(y_test, y_pred))
29 print("Hasil akurasi pemodelan SVM:", accuracy_score(y_test, y_pred))
```

Penjelasan:

1. Import library

```
1  import numpy as np
2  import pandas as pd
3  from sklearn import datasets
4  from sklearn.metrics import confusion_matrix
5  from sklearn.metrics import accuracy_score
6  from sklearn.model_selection import train_test_split, cross_val_score
7  from sklearn.svm import SVC
```

2. Load dataset iris and change dataset

```
8  #load data
9  iris = datasets.load_iris()
10 df= pd.DataFrame(data= np.c_[iris['data'], iris['target']],
11                  columns= iris['feature_names'] + ['target'])
12
13 df['species'] = pd.Categorical.from_codes(iris.target, iris.target_names)
14 del df['target']
15 print(df)
```

Output :

```
   sepal length (cm)  sepal width (cm)  ...  petal width (cm)  species
0                5.1                3.5  ...                0.2    setosa
1                4.9                3.0  ...                0.2    setosa
2                4.7                3.2  ...                0.2    setosa
3                4.6                3.1  ...                0.2    setosa
4                5.0                3.6  ...                0.2    setosa
..                ...                ...  ...                ...    ...
145               6.7                3.0  ...                2.3  virginica
146               6.3                2.5  ...                1.9  virginica
147               6.5                3.0  ...                2.0  virginica
148               6.2                3.4  ...                2.3  virginica
149               5.9                3.0  ...                1.8  virginica

[150 rows x 5 columns]
```

3. Separate variables (X) and class instances (Y)

```
16 #memisahkan label dan parameter
17 x=df.iloc[:, :-1]
18 y=df.iloc[:, 4]
19 print("data variabel".center(100, "="))
20 print(x)
21 print("data kelas".center(100, "="))
22 print(y)
```

Output :

```
=====data variabel=====
   sepal length (cm)  sepal width (cm)  petal length (cm)  petal width (cm)
0                5.1                3.5                1.4                0.2
1                4.9                3.0                1.4                0.2
2                4.7                3.2                1.3                0.2
3                4.6                3.1                1.5                0.2
4                5.0                3.6                1.4                0.2
..                ...                ...                ...                ...
145               6.7                3.0                5.2                2.3
146               6.3                2.5                5.0                1.9
147               6.5                3.0                5.2                2.0
148               6.2                3.4                5.4                2.3
149               5.9                3.0                5.1                1.8

[150 rows x 4 columns]
=====data kelas=====
0      setosa
1      setosa
2      setosa
3      setosa
4      setosa
..      ...
145    virginica
146    virginica
147    virginica
148    virginica
149    virginica
Name: species, Length: 150, dtype: category
Categories (3, object): ['setosa', 'versicolor', 'virginica']
```

4. Split data, 70 % training and 30 % testing.

```
23 #pemisahan training dan testing
24 x_train,x_test, y_train, y_test=train_test_split(x,y,test_size=0.30, random_state=100)
25 print("data variabel training".center(100,"="))
26 print(x_train)
27 print("data kelas training".center(100,"="))
28 print(x_test)
29 print("data variabel testing".center(100,"="))
30 print(y_train)
31 print("data kelas testing".center(100,"="))
32 print(y_test)
```

Output :

Training Data (x train dan y train)

```
=====data variabel training=====
   sepal length (cm)  sepal width (cm)  petal length (cm)  petal width (cm)
6                4.6                3.4                1.4                0.3
25               5.0                3.0                1.6                0.2
21               5.1                3.7                1.5                0.4
92               5.8                2.6                4.0                1.2
9                4.9                3.1                1.5                0.1
..              ...                ...                ...                ...
87               6.3                2.3                4.4                1.3
103              6.3                2.9                5.6                1.8
67               5.8                2.7                4.1                1.0
24               4.8                3.4                1.9                0.2
8                4.4                2.9                1.4                0.2

[105 rows x 4 columns]
=====data kelas training=====
6          setosa
25         setosa
21         setosa
92    versicolor
9          setosa
..          ...
87    versicolor
103   virginica
67    versicolor
24         setosa
8          setosa
Name: species, Length: 105, dtype: category
Categories (3, object): ['setosa', 'versicolor', 'virginica']
```

Testing Data (x test dan y test)

```
=====data variabel testing=====
   sepal length (cm)  sepal width (cm)  petal length (cm)  petal width (cm)
128                6.4                2.8                5.6                2.1
11                4.8                3.4                1.6                0.2
118               7.7                2.6                6.9                2.3
15                5.7                4.4                1.5                0.4
123               6.3                2.7                4.9                1.8
135               7.7                3.0                6.1                2.3
32                5.2                4.1                1.5                0.1
1                4.9                3.0                1.4                0.2
116               6.5                3.0                5.5                1.8
45                4.8                3.0                1.4                0.3
40                5.0                3.5                1.3                0.3
115               6.4                3.2                5.3                2.3
26                5.0                3.4                1.6                0.4
28                5.2                3.4                1.4                0.2
145               6.7                3.0                5.2                2.3
97                6.2                2.9                4.3                1.3
```



```
=====data kelas testing=====
128    virginica
11      setosa
118    virginica
15      setosa
123    virginica
135    virginica
32      setosa
1       setosa
116    virginica
45      setosa
40      setosa
115    virginica
26      setosa
28      setosa
145    virginica
97     versicolor
```

5. SVM Modelling

```
20    #pemodelan svm
21    model = SVC()
22    model.fit(x_train, y_train)
23    y_pred = model.predict(x_test)
24    print("hasil prediksi svm")
25    print(y_pred)
```

Output :

```
hasil prediksi svm
['virginica' 'versicolor' 'setosa' 'versicolor' 'setosa' 'versicolor'
 'versicolor' 'versicolor' 'versicolor' 'versicolor' 'setosa' 'virginica'
 'versicolor' 'versicolor' 'virginica' 'setosa' 'versicolor' 'versicolor'
 'versicolor' 'versicolor' 'setosa' 'versicolor' 'versicolor' 'setosa'
 'versicolor' 'setosa' 'setosa' 'versicolor' 'versicolor' 'setosa'
 'setosa' 'setosa' 'versicolor' 'virginica' 'versicolor' 'virginica'
 'versicolor' 'virginica' 'setosa' 'versicolor' 'virginica' 'virginica'
 'versicolor' 'virginica' 'versicolor']
```

6. Calculate accuracy using confusion matrix

```
26    #evaluasi confusion matrix dan evaluasi akurasi python
27    print("Hasil confusion matrix")
28    print(confusion_matrix(y_test,y_pred))
29    print("Hasil akurasi pemodelan SVM:",accuracy_score(y_test, y_pred))
```

Output :

Confusion Matrix

```
Hasil confusion matrix
[[12  0  0]
 [ 0 23  0]
 [ 0  1  9]]
```

Accuracy

```
Hasil akurasi pemodelan SVM: 0.9777777777777777
```

Latihan

Kerjakan semua source code diatas

Tugas :

1. Kerjakan tugas secara kelompok (Kelompok sama dengan kelompok UTS)
2. Perbaiki File program UTS
3. Tambahkan SVM dengan phyton
4. Buatlah laporan (Format sama seperti UTS)
5. File yang dikumpulkan data asli, program phyton, dan laporan
6. Penamaan Tugas “ Tugas SVM_Kelompok XXX.Zip
7. Dikumpulkan paling lambat Sabtu / 15 April 2023 Pukul 23.59 Wib