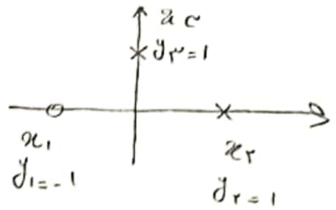
فاطمه ناشیان ۸۱۰۱۹۸۴۷۹ تمرین ۴ ماشین لرنینگ

(۱



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

$$\text{st.} \quad y_i(w^T x_i + b) \geq 1$$

$$\sum_{i=1}^{n} \lambda_i y_i = 0 \Rightarrow -\lambda_1 + \lambda_2 + \lambda_3 = 0 \Rightarrow \boxed{\lambda_1 = \lambda_2 + \lambda_3}$$

$$w = \sum_{i=1}^{n} y_i x_i \lambda_i \Rightarrow w = -\lambda_1 \begin{bmatrix}-1\\0\end{bmatrix} + \lambda_2 \begin{bmatrix}1\\0\end{bmatrix} + \lambda_3 \begin{bmatrix}0\\1\end{bmatrix}$$

$$L(w,b,\lambda) = \frac{1}{2} w^T w + \sum \lambda_i \left(1 - y_i(w^T x_i + b)\right)$$

$$\min_{\lambda} \frac{1}{2} \sum_{i,j} \lambda_i \lambda_j y_i y_j x_i^T x_j - \sum_{i=1}^{n} \lambda_i$$

$x_1^T x_2 = -1 \quad x_1^T x_3 = 0 \quad x_1^T x_1 = 1 \quad x_2^T x_1 = -1 \quad x_2^T x_3 = 0 \quad x_2^T x_2 = 1$

$x_3^T x_1 = 0 \quad x_3^T x_2 = 0 \quad x_3^T x_3 = 1$

$$\min \frac{1}{2}\left[\lambda_1^2 + \lambda_2^2 + \lambda_3^2 + \lambda_1\lambda_2 + \lambda_1\lambda_2\right] - \lambda_1 - \lambda_2 - \lambda_3$$

$$\frac{\partial J}{\partial \lambda_1} = \lambda_1 + \lambda_2 - 1 = 0$$

$$\frac{\partial J}{\partial \lambda_3} = \lambda_3 - 1 = 0 \longrightarrow \boxed{\lambda_3 = -1} \rightarrow \lambda_1 = \lambda_2 - 1$$

$$\boxed{\begin{matrix}\lambda_1 = 0\\ \lambda_2 = 1\end{matrix}}$$

$$\begin{cases}\lambda_1 = 0\\ \lambda_2 = 1\\ \lambda_3 = -1\end{cases} \rightarrow w = \begin{bmatrix}1\\0\end{bmatrix} + \begin{bmatrix}0\\1\end{bmatrix} = \begin{bmatrix}1\\1\end{bmatrix}$$

$$\begin{matrix}x_2\\ y_2 = 1\end{matrix} \longrightarrow y_2(w^T x_2 + b) = 1 \longrightarrow 1\left(\begin{bmatrix}1 & 1\end{bmatrix}\begin{bmatrix}1\\0\end{bmatrix} + b\right) = 1 \rightarrow \boxed{b = 0}$$

خط $0 = w^T x + b \rightarrow \begin{bmatrix}1 & 1\end{bmatrix} x + 0 = 0 \longrightarrow x + y = 0 \rightarrow \boxed{x = -y}$

۲)

$$k(x_i,x_j) = \exp\left(-\frac{1}{\alpha}\|x_i-x_j\|^c\right) = \phi(x_i)^T\phi(x_j)$$

$$\|\phi(x_i)-\phi(x_j)\|^2 = (\phi(x_i)-\phi(x_j))^T(\phi(x_i)-\phi(x_j))$$

$$= \varphi(x_i)^T\varphi(x_i) - \varphi(x_i)^T\varphi(x_j) - \varphi(x_j)^T\varphi(x_i) + \varphi(x_j)^T\varphi(x_j)$$

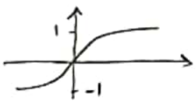
$$= K(x_i,x_i) - K(x_i,x_j) - K(x_j,x_i) + K(x_j,x_j)$$

$$= \exp(0) - 2\exp\left(-\frac{1}{\alpha}\|x_i-x_j\|^c\right) + \exp(0) = 2 - 2\exp\left(-\frac{1}{\alpha}\|x_i-x_j\|^c\right) \leq K$$

ماکسیمم این عبارت ۲ است پس

$$\|\phi(x_i)-\phi(x_j)\|^2 \leq 2$$

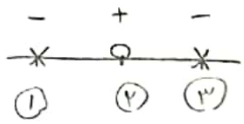
$$K(x_i,x_j) = \tanh(a x_i^T x_j + b)$$



$$\Rightarrow \underbrace{\tanh(a x_i^T x_i + b)}_{\text{ماکسیمم } 1} - \underbrace{\tanh(a x_i^T x_j + b)}_{\text{مینیمم } -1} - \underbrace{\tanh(a x_j^T x_i + b)}_{\text{مینیمم } -1} + \underbrace{\tanh(a x_j^T x_j + b)}_{\text{ماکسیمم } 1}$$

$$\|\phi(x_i)-\phi(x_j)\|^2 \leq 4$$

۳) الف)



داده‌ها به صورت خطی جدایی پذیر نیستند

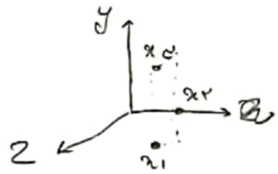
ب)

$\varphi(x) = [1, \sqrt{2}x, x^2]^T$

$\varphi(x_1) = [1, -\sqrt{2}, 1]^T \quad y=-1$

$\varphi(x_2) = [1, 0, 0]^T \quad y=+1$

$\varphi(x_3) = [1, 0, 0]^T \quad y=-1$



داده‌ها به صورت خطی جدایی پذیر هستند. همانطور که دیده می‌شود داده‌ها دارای x یکسان هستند و در راستای y نیز قابل جداشدن نیستند ولی در راستای z قابل جداشدن هستند و اگر بیشترین margin را بخواهیم صفحه $z=\frac{1}{2}$ می‌تواند پاسخ خوبی دهد.

ج)

$$w = \sum y_i x_i \lambda_i = -\lambda_1\begin{bmatrix}1\\-\sqrt{2}\\1\end{bmatrix} + \lambda_2\begin{bmatrix}1\\0\\0\end{bmatrix} - \lambda_3\begin{bmatrix}1\\\sqrt{2}\\1\end{bmatrix}$$

$$\sum \lambda_i y_i = -\lambda_1 + \lambda_2 - \lambda_3 = 0$$

$x_1^Tx_1 = 4 \quad x_2^Tx_2 = 1 \quad x_3^Tx_3 = 4 \quad x_1^Tx_2 = 1 \quad x_1^Tx_3 = 0 \quad x_2^Tx_3 = 1$

$$\min_{\lambda} \frac{1}{2}\left[4\lambda_1^2 + \lambda_2^2 + 4\lambda_3^2 - 2\lambda_1\lambda_2 - 2\lambda_2\lambda_3\right] - \lambda_1 - \lambda_2 - \lambda_3$$

$$\frac{\partial J}{\partial \lambda_1} = 4\lambda_1 - \lambda_2 - 1 = 0$$

$$\frac{\partial J}{\partial \lambda_3} = 4\lambda_3 - \lambda_2 - 1$$

$$\lambda_2 = \lambda_1 + \lambda_3 \Rightarrow \begin{cases} \lambda_1 = \frac{1}{2} \\ \lambda_2 = 1 \\ \lambda_3 = \frac{1}{2} \end{cases}$$

$$w = -\frac{1}{2}\begin{bmatrix} -1 \\ -\sqrt{2} \\ 1 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} - \frac{1}{2}\begin{bmatrix} 1 \\ \sqrt{2} \\ 1 \end{bmatrix} = \begin{bmatrix} 2 \\ 0 \\ 1 \end{bmatrix}$$

$$y_2(w^T x_2 + b) = 1 \rightarrow 1\left(\begin{bmatrix} 2 & 0 & 1 \end{bmatrix}\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} + b\right) = 1 \rightarrow \boxed{b = -1}$$

خط $\quad w^T x + b = 0 \rightarrow \begin{bmatrix} 2 & 0 & 1 \end{bmatrix} x - 1 = 0 \rightarrow \boxed{2x + z - 1 = 0}$

$$\text{margin} = \frac{2}{\|w\|} = \frac{2}{\sqrt{5}}$$

۴) الف)

$$\min \frac{1}{2}\|w\|^2 + C\sum \varepsilon_i$$

$$L(w, b, \varepsilon, \lambda, \mu) = \frac{1}{2}\|w\|^2 + C\sum\varepsilon_i + \sum\lambda_i\left(1 - \varepsilon_i - y_i(w^T x + b)\right)^2 - \sum \mu_i \varepsilon_i$$

$$g(\lambda, \mu) = \min L(w, b, \varepsilon, \lambda, \mu)$$

$$\left.\begin{array}{l} \nabla_w L = 0 \rightarrow w = \sum \lambda_i y_i x_i \\ \nabla_b L = 0 \rightarrow \sum \lambda_i y_i = 0 \\ \nabla_\varepsilon L = 0 \rightarrow \mu_j = C - \lambda_j \end{array}\right\} \quad \begin{array}{l} \min\limits_{\lambda} \frac{1}{2}\sum\limits_{i,j} \lambda_i \lambda_j y_i y_j x_i^T x_j + \sum \lambda_i \\ \text{s.t} \quad \sum \lambda_i y_i = 0 \qquad 0 \leq \lambda_i \leq C \end{array}$$

برای اینکه misclassified رخ دهد درحالت $y_i = +1$ باید $+1 - \varepsilon_i < 0$ شود پس $1 < \varepsilon_i$ می‌شود
درحالت $y_i = -1$ نیز باید $-1 + \varepsilon_i > 0$ شود پس بازهم $1 < \varepsilon_i$ می‌شود
از طرفی برای همه داده‌ها $\sum \varepsilon_i$ داریم که از $\sum \varepsilon_i$ فقط برای Misclassified ها بزرگتر است
پس حد بالای خطای طبقه بندی $\sum_{i=1}^{M} \varepsilon_i$ است.

ب)

① $K(x_i, x_j) = \langle \varphi(x_i), \varphi(x_j) \rangle$

② $\langle \varphi(x), \varphi(y) \rangle^2 \leq \langle \varphi(x), \varphi(x) \rangle \langle \varphi(y), \varphi(y) \rangle$

③ $w = \dfrac{\langle \varphi(x), \varphi(y) \rangle}{\langle \varphi(y), \varphi(y) \rangle} \varphi(y) \quad , \quad z = \varphi(x) - w$

با کمک ۳ رابطه‌ای که بالا نوشته‌ام به اثبات می‌پردازم

$$\langle z, \varphi(y) \rangle = \left\langle \varphi(x) - \frac{\langle \varphi(x), \varphi(y) \rangle}{\langle \varphi(y), \varphi(y) \rangle} \varphi(y), \varphi(y) \right\rangle = \langle \varphi(x), \varphi(y) \rangle - \frac{\langle \varphi(x), \varphi(y) \rangle}{\langle \varphi(y), \varphi(y) \rangle} \langle \varphi(y), \varphi(y) \rangle = 0$$

$$\langle \varphi(x), \varphi(x) \rangle = \frac{\langle \varphi(x), \varphi(y) \rangle}{\langle \varphi(y), \varphi(y) \rangle} \frac{\langle \varphi(x), \varphi(y) \rangle}{\langle \varphi(y), \varphi(y) \rangle} \langle \varphi(y), \varphi(y) \rangle = \frac{\langle \varphi(x), \varphi(y) \rangle^2}{\langle \varphi(y), \varphi(y) \rangle}$$

$$\langle \varphi(x), \varphi(x) \rangle \geqslant \frac{\langle \varphi(x), \varphi(y) \rangle^2}{\langle \varphi(y), \varphi(y) \rangle} \longrightarrow \langle \varphi(x), \varphi(y) \rangle^2 \leqslant \langle \varphi(x), \varphi(x) \rangle \langle \varphi(y), \varphi(y) \rangle$$

راه دوم

$$K(x, y)^2 = \begin{bmatrix} K(x,x) & K(x,y) \\ K(y,x) & K(y,y) \end{bmatrix}$$

$$\Downarrow$$

$$\det \rightarrow \quad K(x,x)\,K(y,y) - K(y,x)\,K(x,y) \geqslant 0$$

$$K(x,y)^2 \leqslant K(x,x)\,K(y,y).$$

5)

الف)

Rbf = محبوب ترین کرنل است که در طبقه بندی svm استفاده می شود . بر روی دیتاست های بزرگ بهتر جواب میدهد و روی دیتا ست کوچک احتمال اورفیت شدن را زیاد میکند. وقتی استفاده می شود که شناخت قبلی از دیتا ها نداریم و به صورت radial است.

Linear = وقتی داده ها به صورت خطی جدایی پذیر باشند با یک خط داده ها را جدا میکند. وقتی استفاده می شود که تعداد ویژگی های زیادی داشته باشیم. ساده ترین کرنل است و یک بعد دارد.

Polynomial = معمولا در svm استفاده می شود. میتوان کرنل های غیر خطی را با ان مدل کرد.مناسب پردازش تصویر است.

ب)

	C=1		C=100		C=1000	
	train	test	train	test	train	test
RBF	0.978767	0.982681	0.984215	0.98044	0.986101	0.97653
Linear	0.978628	0.98156	0.978767	0.98156	0.978767	0.98156
Polynomial	0.976952	0.979608	0.980583	0.979050		
Sigmoid	0.870931	0.872625	0.870652	0.87262		

با توجه به داده های جمع اوری شده بالا بهترین کرنل rbf با c=1000 است.

پ)

با الگوریتم گریدسرچ بهترین پارامتر ها شامل کرنل rbf و c=100 و gamma=0.1 است.

```
from sklearn.model_selection import GridSearchCV

param_grid = [
    {'C': [1, 10, 100, 500], 'gamma': [0.1,0.3,0.5,0.7,0.9], 'kernel': ['rbf']},
    {'C': [1,10,100,1000], 'kernel': ['linear']},
    {'C': [1, 10, 100, 1000], 'gamma': [0.01,0.03,0.05], 'degree': [2,3,4] , 'kernel': ['poly']},
    ]

grid = GridSearchCV(SVC(), param_grid, refit = True, verbose = 3)
```

```
from sklearn.metrics import classification_report, confusion_matrix
print(grid.best_params_)
print(grid.best_estimator_)
```

```
{'C': 100, 'gamma': 0.1, 'kernel': 'rbf'}
SVC(C=100, gamma=0.1)
```

ت)

حال بهترین طبقه بند ها را روی داده ها اعمال میکنیم و کانفیوژن ماتریس را گزارش میکنیم.

کانفیوژن ماتریس بخش ب

```
svm_clf = Pipeline([
            ("scaler", StandardScaler()),
            ("svm_clf", SVC(kernel="rbf", C=1000))
        ])
svm_clf.fit(X_train, y_train)
Ypred_test = svm_clf.predict(X_test)

print(classification_report(y_test, Ypred_test))
```

```
              precision    recall  f1-score   support

           0       0.98      0.99      0.99      3251
           1       0.90      0.84      0.87       329

    accuracy                           0.98      3580
   macro avg       0.94      0.92      0.93      3580
weighted avg       0.98      0.98      0.98      3580
```

کانفیوژن ماتریس بخش پ

```
grid_predictions = grid.predict(X_test)

print(classification_report(y_test, grid_predictions))
```

```
              precision    recall  f1-score   support

           0       0.99      0.99      0.99      3251
           1       0.93      0.86      0.90       329

    accuracy                           0.98      3580
   macro avg       0.96      0.93      0.94      3580
weighted avg       0.98      0.98      0.98      3580
```

مشاهده میشود به صورت کلی کرنل rbf و c=100 و gamma=0.1 دارای دقت و صحت بیشتری است .

6)

الف)

در این روش به دنبال پیش بینی بهتر هستیم که از طریق ترکیب کردن چند مدل به دست می اید. سه روش اصلی bagging, stacking, boosting, دارد و لازم است این سه روش را بشناسیم و در پیش بینی ها اعمال کنیم. در نهایت برای نتیجه گیری نهایی روی مدل ها رای گیری میکنیم.

ب)

برای جایگذاری داده های از دست رفته از میانگین هر ستون استفاده شده است.

```
updated_df = df
updated_df['Bare Nuclei']=updated_df['Bare Nuclei'].fillna(updated_df['Bare Nuclei'].mean())
updated_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 699 entries, 0 to 698
Data columns (total 11 columns):
 #   Column                       Non-Null Count  Dtype  
---  ------                       --------------  -----  
 0   Sample Code Number           699 non-null    int64  
 1   Clump Thickness              699 non-null    int64  
 2   Uniformity of Cell Size      699 non-null    int64  
 3   Uniformity of Cell Shape     699 non-null    int64  
 4   Marginal Adhesion            699 non-null    int64  
 5   Single Epithelial Cell Size  699 non-null    int64  
 6   Bare Nuclei                  699 non-null    float64
 7   Bland Chromatin              699 non-null    int64  
 8   Normal Nucleoli              699 non-null    int64  
 9   Mitoses                      699 non-null    int64  
 10  Class                        699 non-null    int64  
dtypes: float64(1), int64(10)
memory usage: 60.2 KB
```

In [138]:
```python
X = updated_df
y = updated_df['Class']
del X['Class']
del X['Sample Code Number']
```

In [139]:
```python
X = preprocessing.normalize(X)
scaler = StandardScaler()
X = scaler.fit_transform(X)
```

In [140]:
```python
LR = LogisticRegression()
DT = DecisionTreeClassifier()
SVM = SVC()
```

In [141]:
```python
fold = KFold(n_splits = 10)
```

In [142]:
```python
voting_clf = VotingClassifier(estimators=[('lr', LR), ('dt', DT), ('svm', SVM)])
accuracy = cross_val_score(voting_clf, X, y, scoring='accuracy', cv=fold)

accuracy
```

Out[142]:
```
array([0.81428571, 0.88571429, 0.91428571, 0.92857143, 0.84285714,
       0.92857143, 0.8       , 0.95714286, 0.98571429, 0.97101449])
```

In [143]:
```python
mean_accuracy = sum(accuracy)/len(accuracy)
print('mean accuracy = ',mean_accuracy)
```

```
mean accuracy =  0.902815734989648
```

میانگین دقت برابر با 90.28 درصد است.