

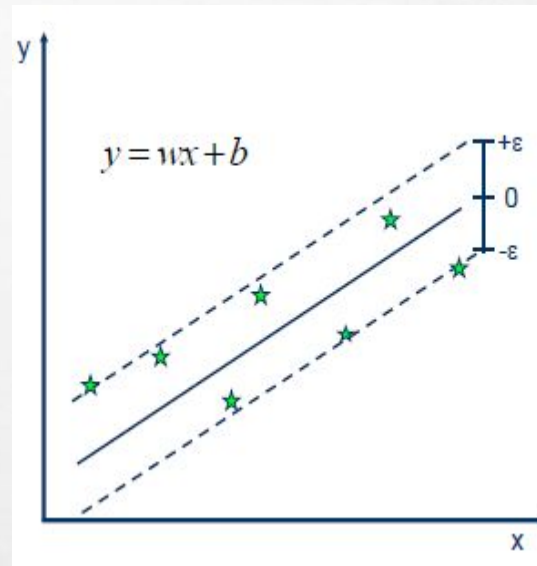
Трөк векторлар көмегімен классификациялау

Орындаған: Болатов Назарбек
Қабылдаған: Қабдрахова Сымбат

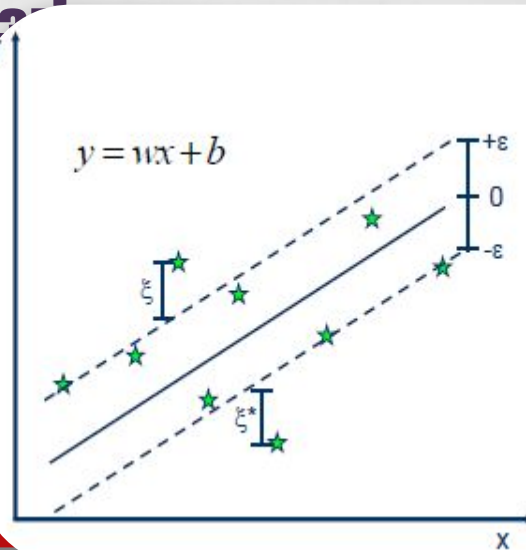
Тірек векторлар машинасы дегеніміз не?



Тірек векторлар машинасы, сондай-ақ алгоритмді сипаттайтын барлық негізгі функцияларды сақтай отырып, регрессия әдісі ретінде пайдаланылуы мүмкін. Негізгі векторлардың регрессиясы (SVM) жіктеу үшін SVM сияқты принциптерді қолданады.

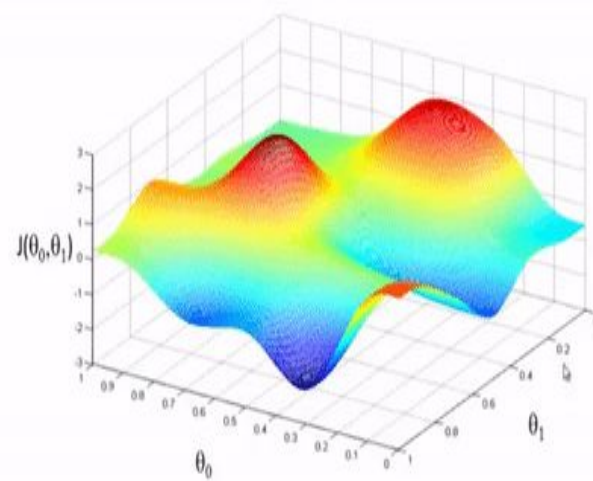


- Solution:
$$\min \frac{1}{2} \|w\|^2$$
- Constraints:
$$y_i - wx_i - b \leq \varepsilon$$
$$wx_i + b - y_i \leq \varepsilon$$

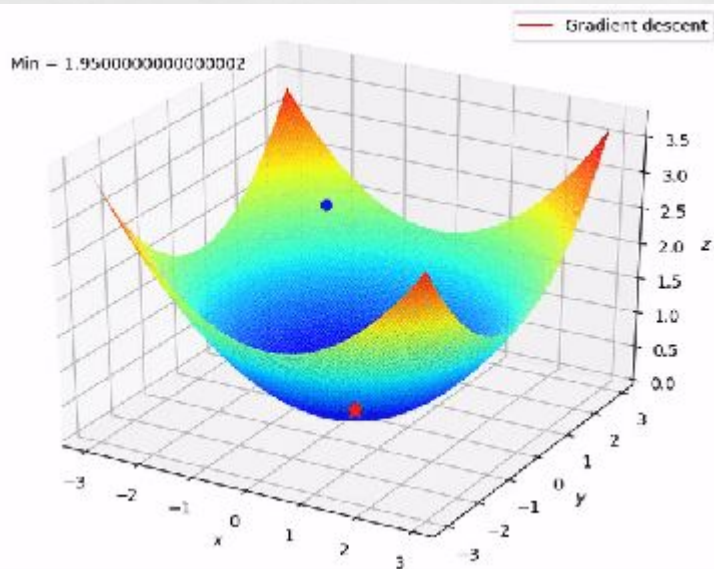


- Minimize:
$$\frac{1}{2} \|w\|^2 + C \sum_{i=1}^N (\xi_i + \xi_i^*)$$
- Constraints:
$$y_i - wx_i - b \leq \varepsilon + \xi_i$$
$$wx_i + b - y_i \leq \varepsilon + \xi_i^*$$
$$\xi_i, \xi_i^* \geq 0$$

Ең алдымен, қорытынды нақты сан болғандықтан, шексіз мүмкіндіктері бар ақпаратты болжау өте қиын болады. Регрессия жағдайында рұқсат шегі (эпсилон) тапсырмадан сұралатын SVM-ге жақындауда орнатылады.



Andrew Ng



Бірақ бұл фактіден басқа күрделі себеп бар, сондықтан алгоритм күрделі, сондықтан оны назарға алу керек. Дегенмен, негізгі идея әрқашан бірдей: қатені азайту, қорды максималдайтын гиперплоскость дараландыру...

СВР

СЫЗЫҚТЫҚ

СЫЗЫҚТЫҚ
емес

СЫЗЫҚТЫҚ
СВР



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

СЫЗЫҚТЫҚ емес
СВР

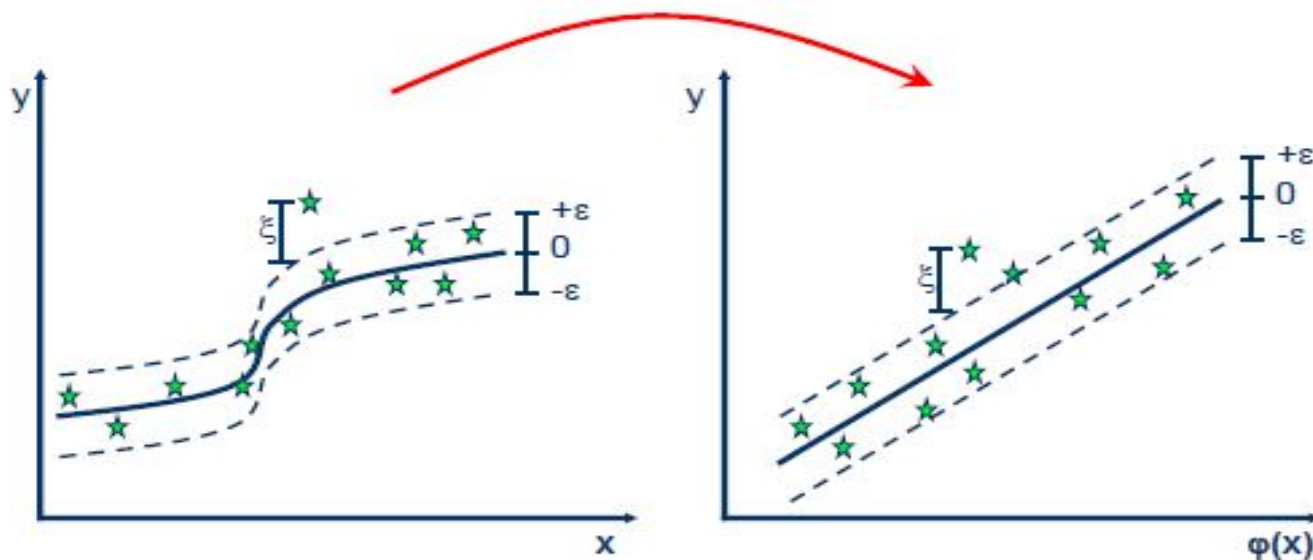


ядроның функциялары
деректерді сызықтық бөлуге
мүмкіндік беру үшін көп
өлшемді белгілер кеңістігіне
түрлендіреді.

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

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

**Сызықтық емес СВР-
дің формулалары**



ERP код

jupyter Kabdrachova 13 apta Last Checkpoint: В прошлое воскресенье в 18:33 (autosaved)



Logout

File Edit View Insert Cell Kernel Widgets Help

Not Trusted

Python 3

Save + Copy Paste Undo Redo Run Stop Refresh Run Cell Markdown

```
import pandas as pd import numpy as np
```

(Бұл жолдың кодында csv файлын оқимын)

```
data=pd.read_csv("C:/Users/hp/Desktop/Сабак/3 курс/2 СЕМЕСТР/Python/heart.csv")
```

(Бұл жолдың кодында 1-ші 5 жолды аламын және қараймын)

```
data.head()
```

(Бұл жолда датасеттердің ақпараттарын оқи аламын)

```
data.info()
```

```
In [5]: data.shape #Бұл жерде қанша жол және баған бардың өлшемін анықтай аламын
```

```
Out[5]: (303, 14)
```

```
In [6]: data.describe() #Датасеттің ортақ статикалық ішіндегі заттарды көре аламы
```



In [6]: `data.describe()` *#Датасеттің ортақ статикалық ішіндегі заттарды көре аламы*

Out[6]:

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak
count	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000
mean	54.366337	0.683168	0.966997	131.623762	246.264026	0.148515	0.528053	149.646865	0.326733	1.039604
std	9.082101	0.466011	1.032052	17.538143	51.830751	0.356198	0.525860	22.905161	0.469794	1.161075
min	29.000000	0.000000	0.000000	94.000000	126.000000	0.000000	0.000000	71.000000	0.000000	0.000000
25%	47.500000	0.000000	0.000000	120.000000	211.000000	0.000000	0.000000	133.500000	0.000000	0.000000
50%	55.000000	1.000000	1.000000	130.000000	240.000000	0.000000	1.000000	153.000000	0.000000	0.800000
75%	61.000000	1.000000	2.000000	140.000000	274.500000	0.000000	1.000000	166.000000	1.000000	1.600000
max	77.000000	1.000000	3.000000	200.000000	564.000000	1.000000	2.000000	202.000000	1.000000	6.200000

In [7]: `y=data['target']` *#Бұл жолда бағанда тұрған таргетті жолға меншіктеп жатырмыз*

In [8]: `x=data.drop(['target'], axis=1)` *#Мына жерде x -ке датасетті мекшиктеймін және 'target' өшіремін*

In [9]: `x.shape, y.shape` *#размер x және y караймыз*

Out[9]: ((303, 13), (303,))

```
In [10]: from sklearn.model_selection import train_test_split #train_test_split Библиотекадан шакырдым . Себеби
```

```
In [20]: x_train, x_test, y_train, y_test = train_test_split(x, y, test_size = 0.3)
```

```
In [21]: from sklearn.svm import SVC  
svm = SVC(kernel='linear')  
svm.fit(x_train, y_train)
```

Out[21]: SVC(C=1.0, break_ties=False, cache_size=200, class_weight=None, coef0=0.0, decision_function_shape='ovr', degree=3, gamma='scale', kernel='linear', max_iter=-1, probability=False, random_state=None, shrinking=True, tol=0.001, verbose=False)

```
In [22]: y_pred = svm.predict(x_test)
```

```
In [23]: from sklearn import metrics
```


Run Cell Stop Cell Restart Cell Markdown

Out[38]: 0.7692307692307693

Gaussian Kernel

```
In [43]: svm2=SVC(kernel='rbf', degree=8) #Гаусс формуласы аркылы питру-ды қолданып Гаусс ядросын шыгарамыз  
svm2.fit(x_train, y_train) #Ядрога полиді меншіктейміз.  
#Ал степень 8 ге тең болды.  
#X және Y трейндерді экранға шыгарамыз
```

Out[43]: SVC(C=1.0, break_ties=False, cache_size=200, class_weight=None, coef0=0.0,
decision_function_shape='ovr', degree=8, gamma='scale', kernel='rbf',
max_iter=-1, probability=False, random_state=None, shrinking=True,
tol=0.001, verbose=False)

```
In [45]: y_pred2=svm2.predict(x_test)
```

```
In [47]: from sklearn import metrics  
metrics.accuracy_score(y_pred2, y_test) #Дайын болған жауапты экранға шығару керекпіз.  
#Ең маңыздысы ассигасу 0-ден үлкен болу керек.
```

Out[47]: 0.6483516483516484

Sigmoid Kernel

#Ең маңыздысы ассигасу 0-ден үлкен болу керек.

Out[47]: 0.6483516483516484

Sigmoid Kernel

In [48]: `svm3=SVC(kernel='sigmoid', degree=8) #Дәл осылай Синмондтық ядроны шығарамыз.
svm3.fit(x_train, y_train)`

Out[48]: `SVC(C=1.0, break_ties=False, cache_size=200, class_weight=None, coef0=0.0,
decision_function_shape='ovr', degree=8, gamma='scale', kernel='sigmoid',
max_iter=-1, probability=False, random_state=None, shrinking=True,
tol=0.001, verbose=False)`

In [49]: `y_pred3=svm3.predict(x_test)`

In [51]: `from sklearn import metrics
metrics.accuracy_score(y_pred3, y_test) #ең маңыздысы 0-ден үлкен болу керек`

Out[51]: 0.5604395604395604

In []:

Назарыңызға рахмет!!!