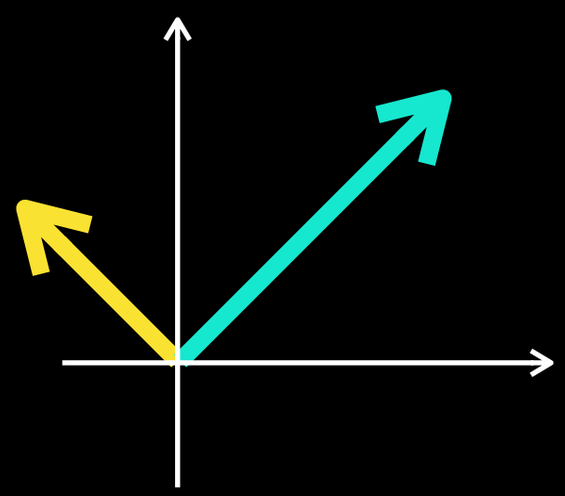
Support vector machines

Linear Algebra Essentials

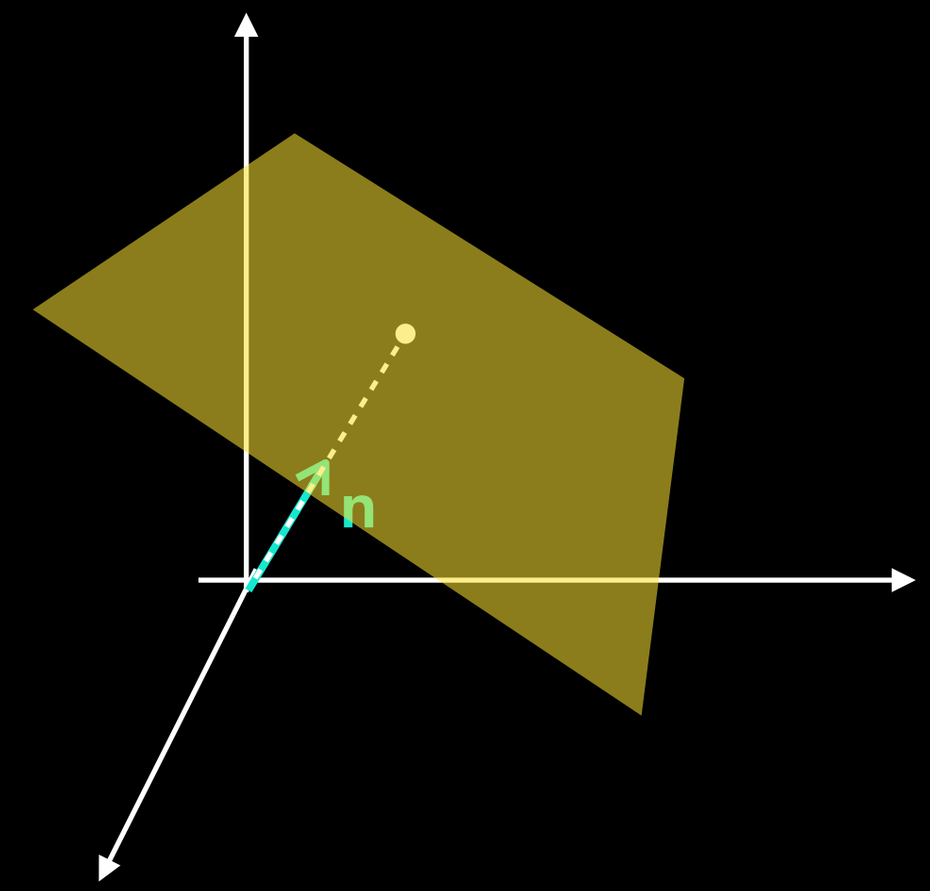


Hyperplane

$(x, n) = d$

$(x, n) > d$ - class 1

$(x, n) < d$ - class 2

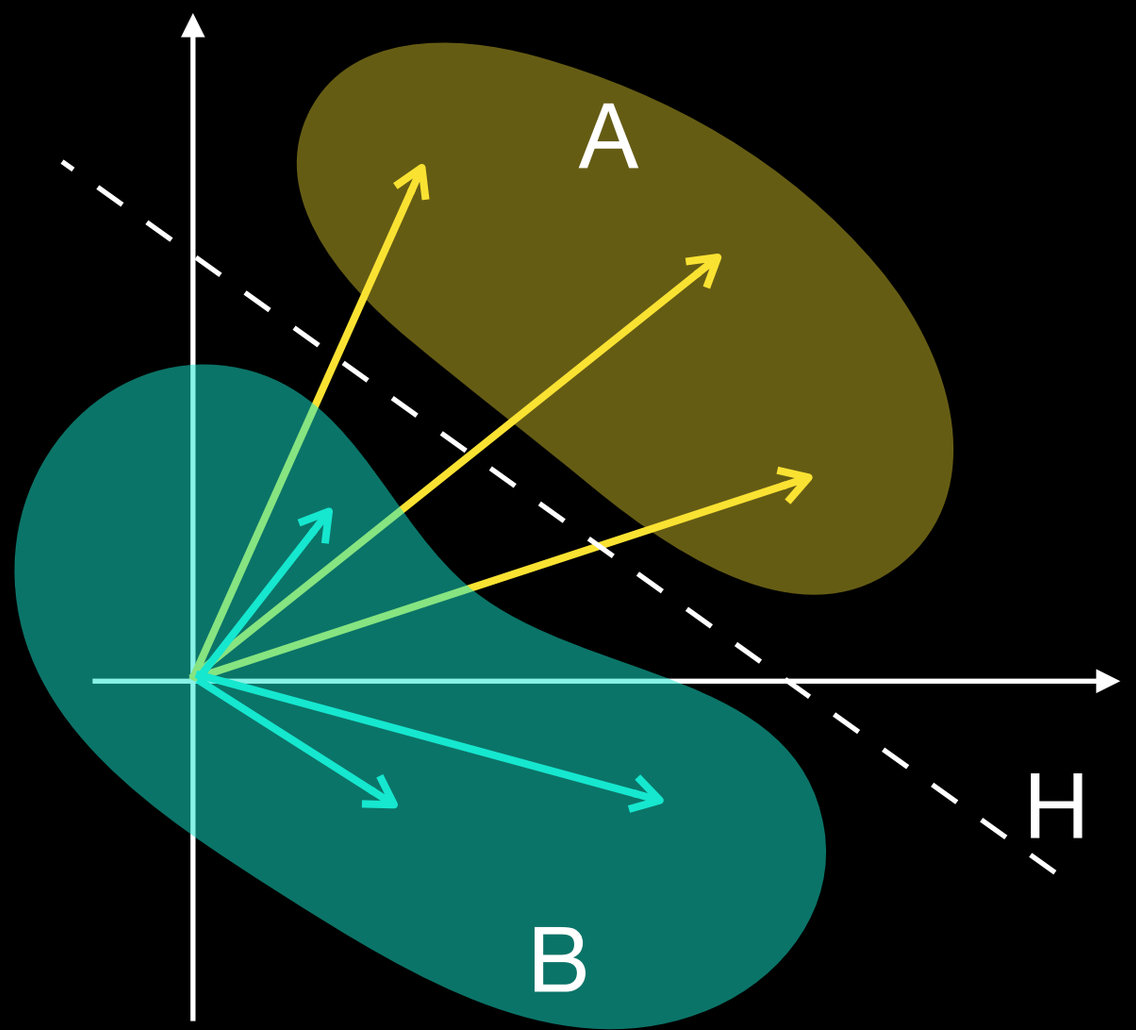


Classification with SVM

$\{V_a\} \in A$

$\{V_b\} \in B$

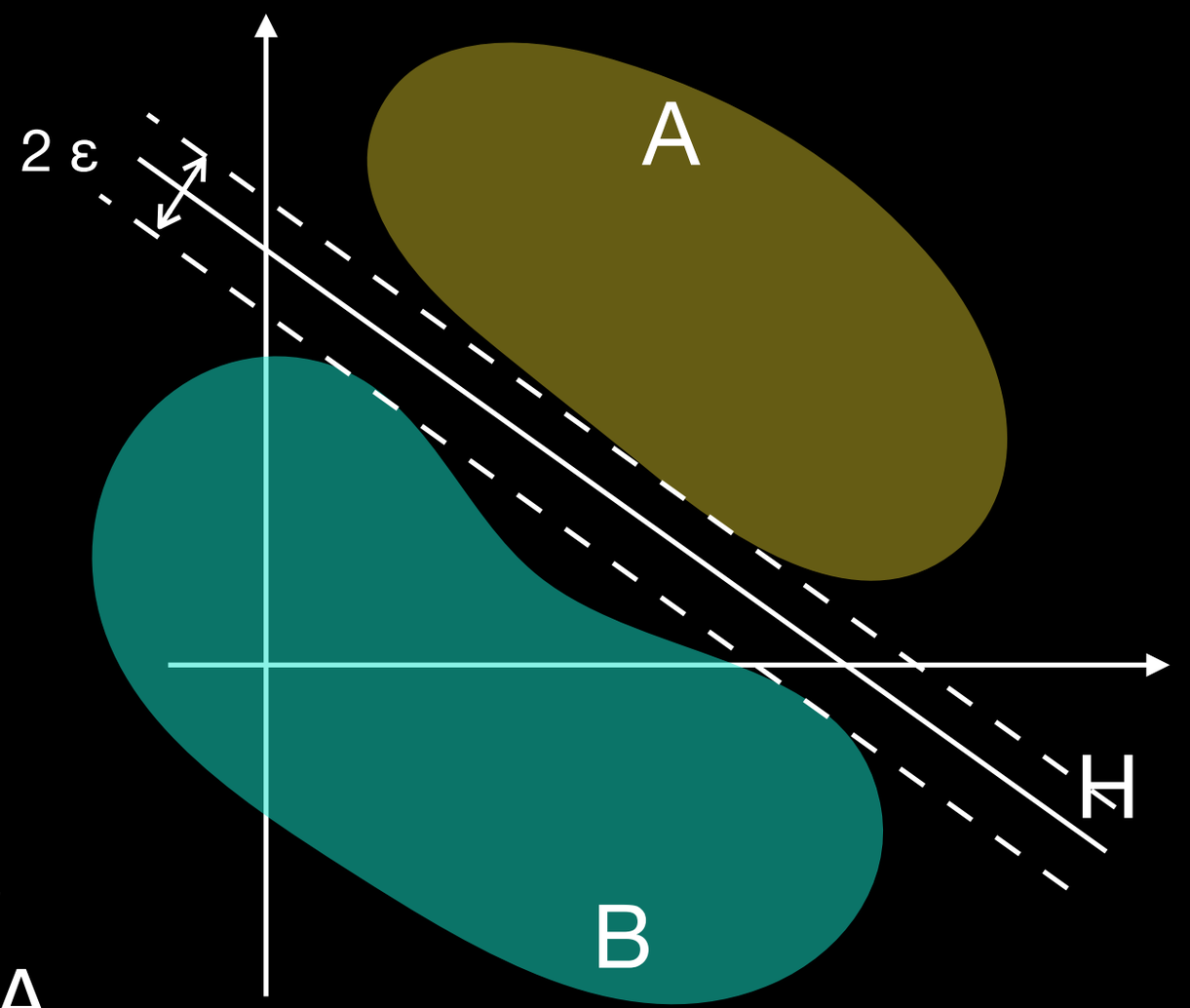
Find hyperplane H that separates A and B



Classification with SVM

$\{V_a\} \in A$

$\{V_b\} \in B$



Find hyperplane H that separates A and B

and maximizes margin ε

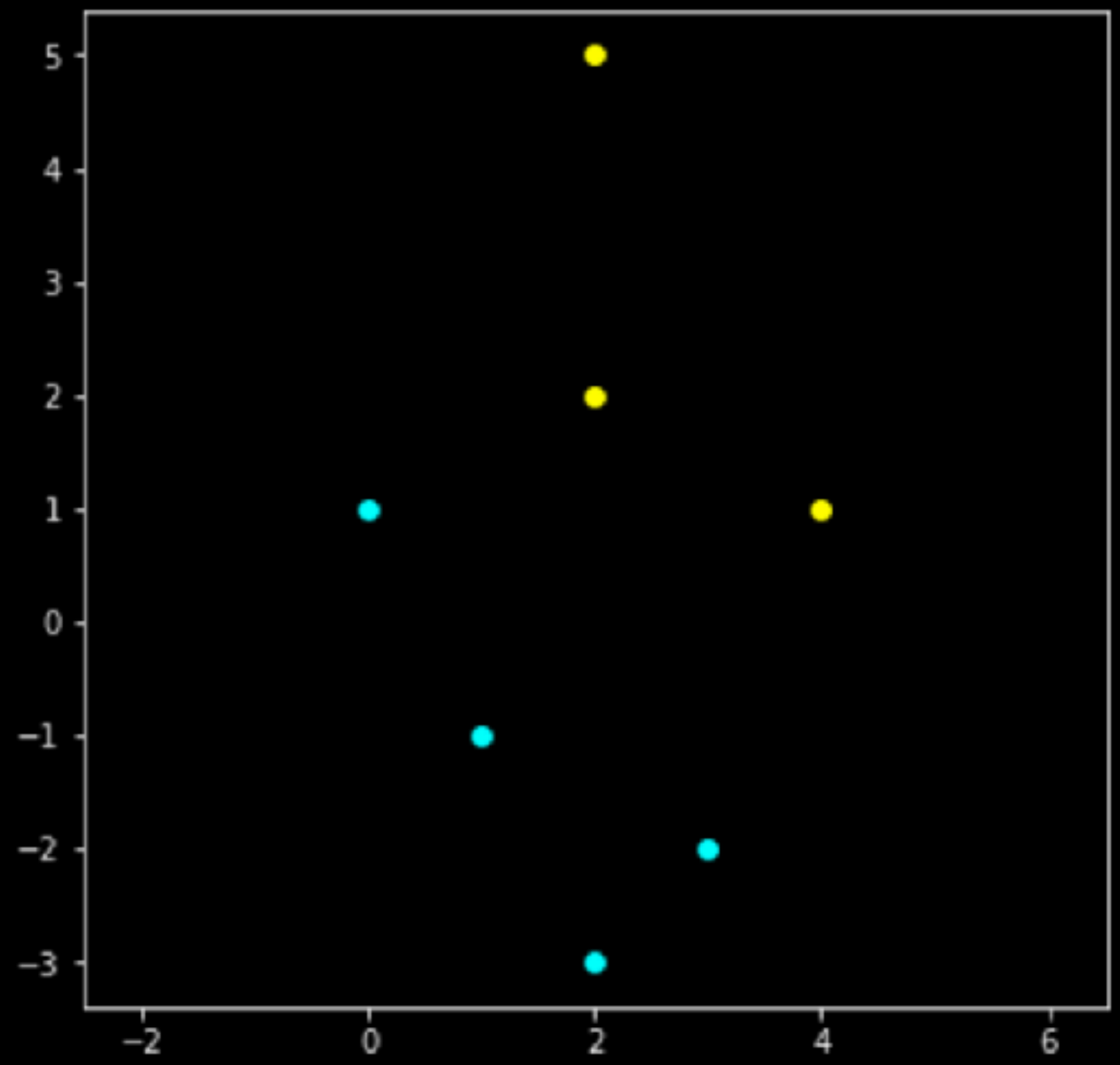
$(x, n) > d + \varepsilon$ - for class A

$(x, n) < d - \varepsilon$ - for class B

Example

```
import numpy as np
import matplotlib.pyplot as plt
from sklearn.svm import SVC
```

```
class_A = [np.array([2,2]), np.array([2,5]),
           np.array([4,1])]
class_B = [np.array([0,1]), np.array([1,-1]),
           np.array([3,-2]), np.array([2,-3])]
```



Example

```
cla = SVC(kernel='linear')
X = np.vstack([class_A, class_B])
X
```

```
array([[ 2,  2],
       [ 2,  5],
       [ 4,  1],
       [ 0,  1],
       [ 1, -1],
       [ 3, -2],
       [ 2, -3]])
```

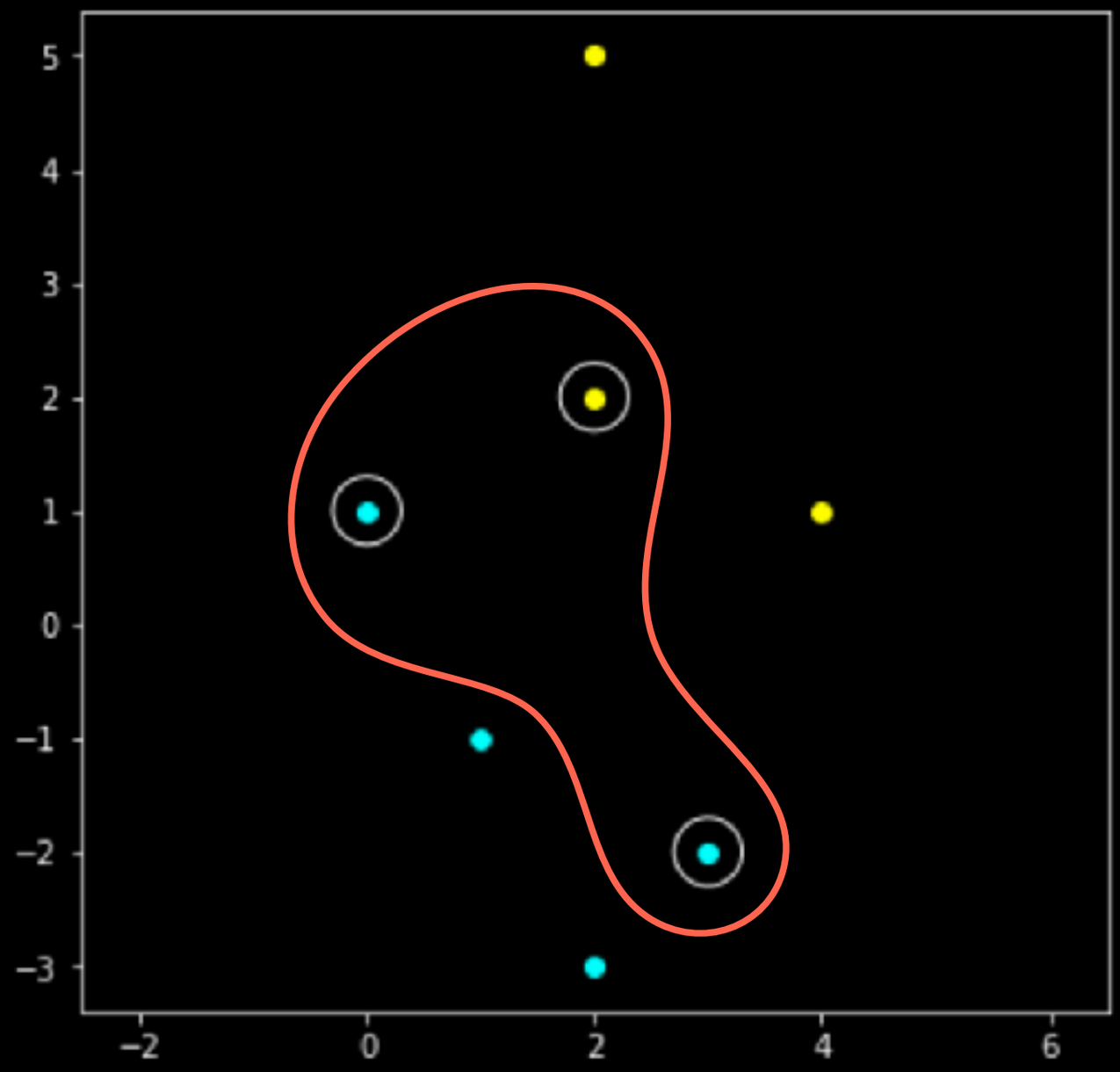
```
y = [0, 0, 0, 1, 1, 1, 1]
```

```
cla.fit(X,y)
```

```
SVC(kernel='linear')
```

```
supp = cla.support_vectors_
supp
```

```
array([[ 2.,  2.],
       [ 0.,  1.],
       [ 3., -2.]])
```



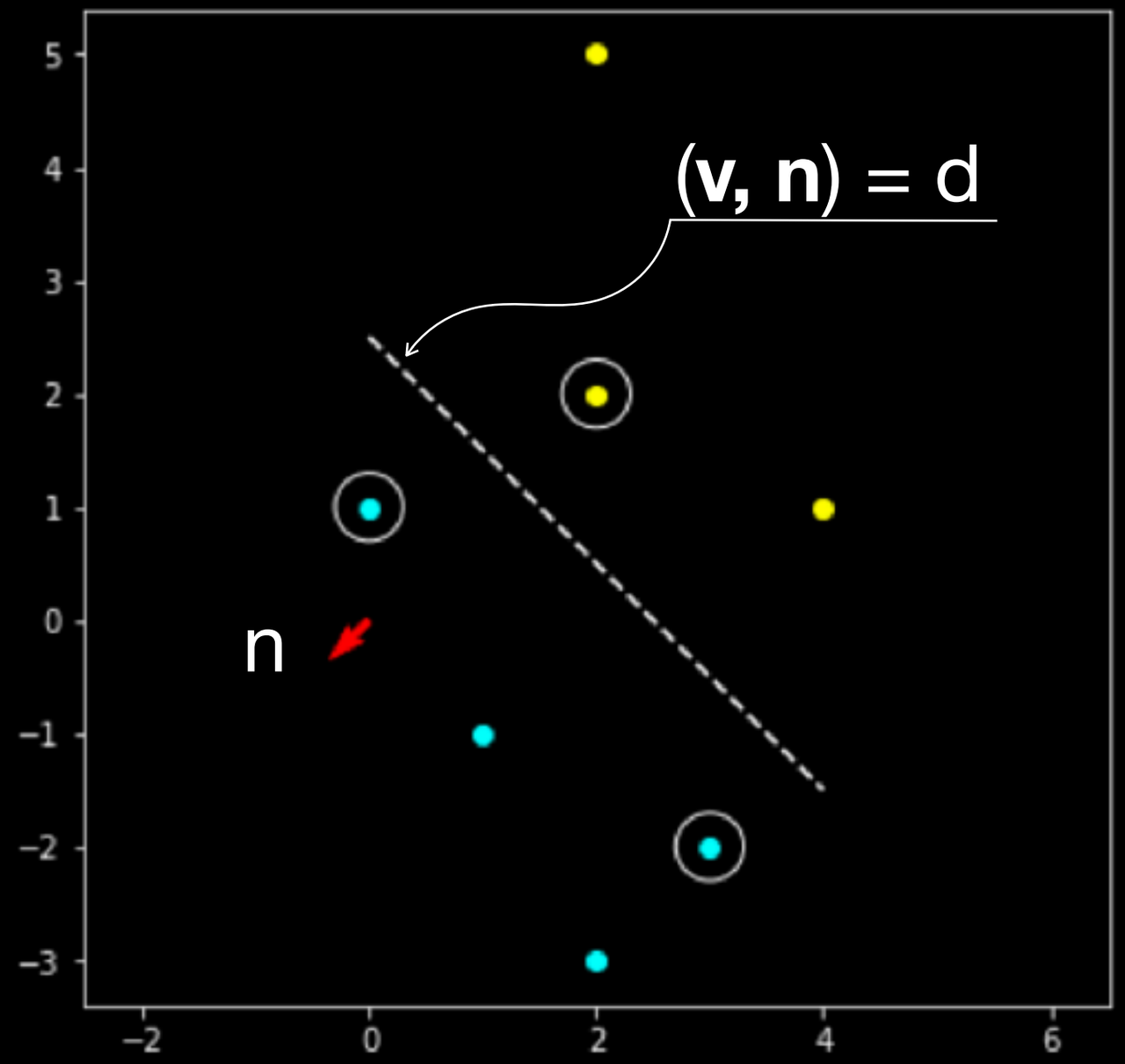
Example

```
1  n = cla.coef_[0]
2  n
```

```
array([-0.6667418, -0.6665164])
```

```
1  d = -cla.intercept_
2  d
```

```
array([-1.6667418])
```



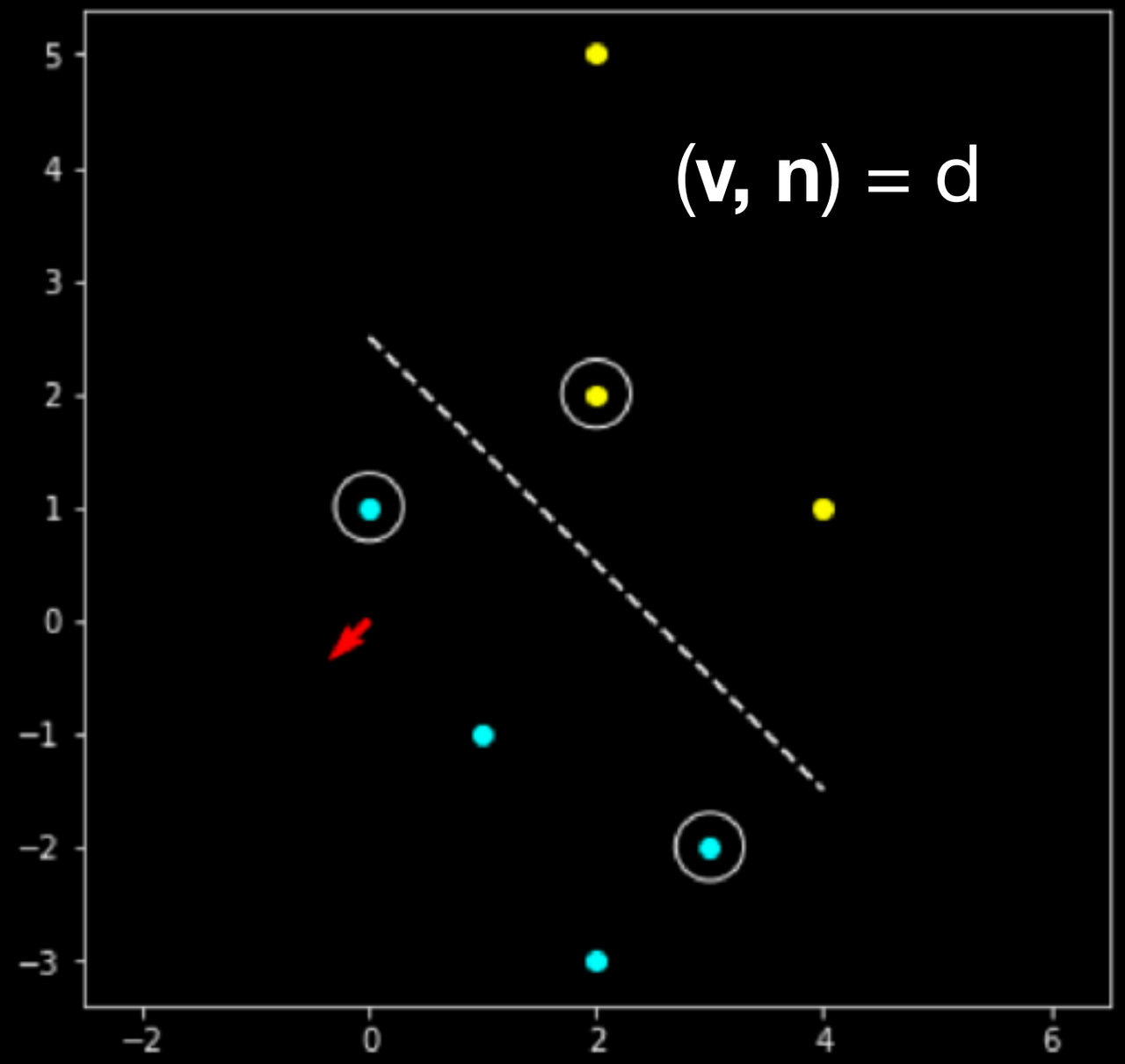
Example

```
for v in class_A:
    print(v.dot(n) - d)
```

```
[-0.9997746]
[-2.9993238]
[-1.6667418]
```

```
for v in class_B:
    print(v.dot(n) - d)
```

```
[1.0002254]
[1.6665164]
[0.9995492]
[2.3328074]
```



(x, n) - d > ε

(x, n) - d < -ε

ε = 1