

SVM Python Code - Cal Poly Humboldt

Kali Chowdhury¹

¹*e-mail:* kchowdh1@jh.edu.

1. SVM Maximum Margin Separating Hyperplane

These examples are available [here](#).

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

# Create 100 separable points
X, y = make_blobs(n_samples=100, centers=2, random_state=1)

# fit the model, don't regularize for illustration purposes
hbdt = svm.SVC(kernel="linear", C=1000)
hbdt.fit(X, y)

plt.scatter(X[:, 0], X[:, 1], c=y, s=30, cmap=plt.cm.Paired)

# plot the decision function
ax = plt.gca()
xlim = ax.get_xlim()
ylim = ax.get_ylim()

# create grid to evaluate model
xx = np.linspace(xlim[0], xlim[1], 90)
yy = np.linspace(ylim[0], ylim[1], 90)
YY, XX = np.meshgrid(yy, xx)
xy = np.vstack([XX.ravel(), YY.ravel()]).T
Z = hbdt.decision_function(xy).reshape(XX.shape)

# plot decision boundary and margins
ax.contour(
    XX, YY, Z, colors="k", levels=[-1, 0, 1], alpha=0.5, linestyles=["--", "-", "--"]
)

# plot support vectors
ax.scatter(
    hbdt.support_vectors_[0],
    hbdt.support_vectors_[1],
```

```

s=100,
linewidth=1,
facecolors="none",
edgecolors="k",
)
plt.show()

```

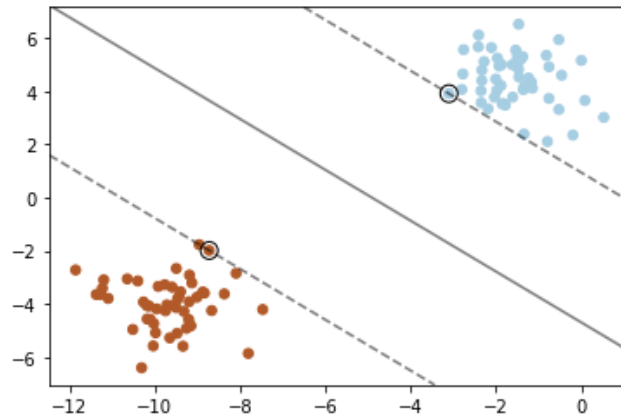


Fig 1: Linear SVM

2. Nonlinear SVM

These examples are available [here](#).

```

#####Non-linear SVM.#####
xx, yy = np.meshgrid(np.linspace(-3, 3, 500), np.linspace(-3, 3, 500))
np.random.seed(0)
X = np.random.randn(300, 2)
Y = np.logical_xor(X[:, 0] > 0, X[:, 1] > 0)

# fit the model
hbdt = svm.NuSVC(gamma="auto")
hbdt.fit(X, Y)

# plot the decision function for each datapoint on the grid
Z = hbdt.decision_function(np.c_[xx.ravel(), yy.ravel()])
Z = Z.reshape(xx.shape)

plt.imshow(
Z,
interpolation="nearest",

```

```
extent=(xx.min(), xx.max(), yy.min(), yy.max()),
aspect="auto",
origin="lower",
cmap=plt.cm.PuOr_r,
)
contours = plt.contour(xx, yy, Z, levels=[0], linewidths=2, linestyle="dashed")
plt.scatter(X[:, 0], X[:, 1], s=30, c=Y, cmap=plt.cm.Paired, edgecolors="k")
plt.xticks(())
plt.yticks(())
plt.axis([-3, 3, -3, 3])
plt.show()
```

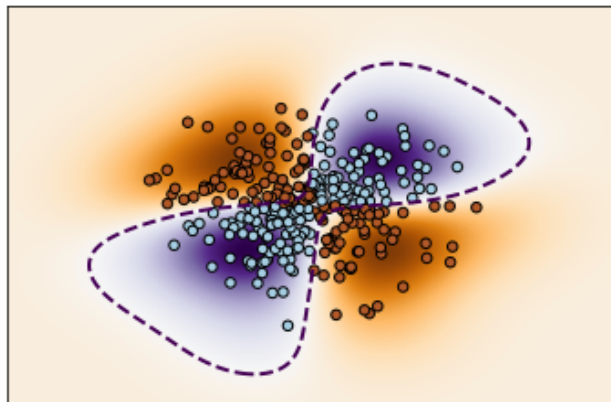


Fig 2: Nonlinear SVM