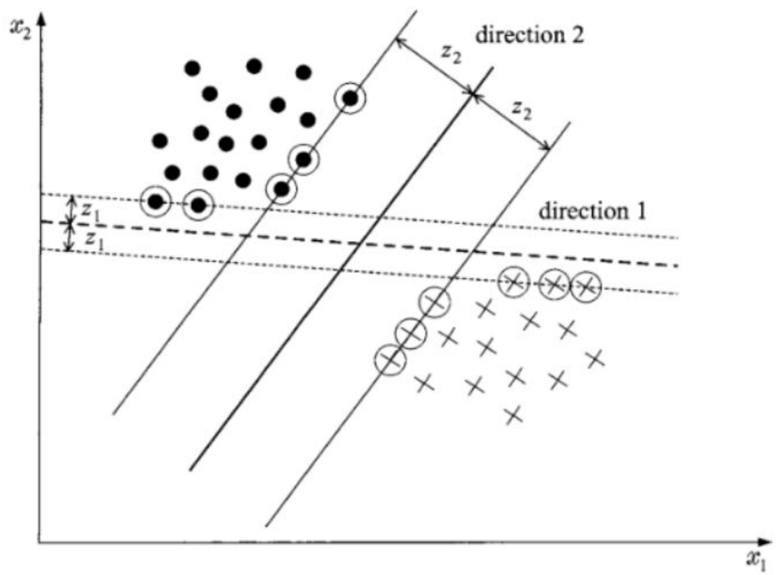
2019

Applied Physics 186
Activity 16

Support Vector Machines

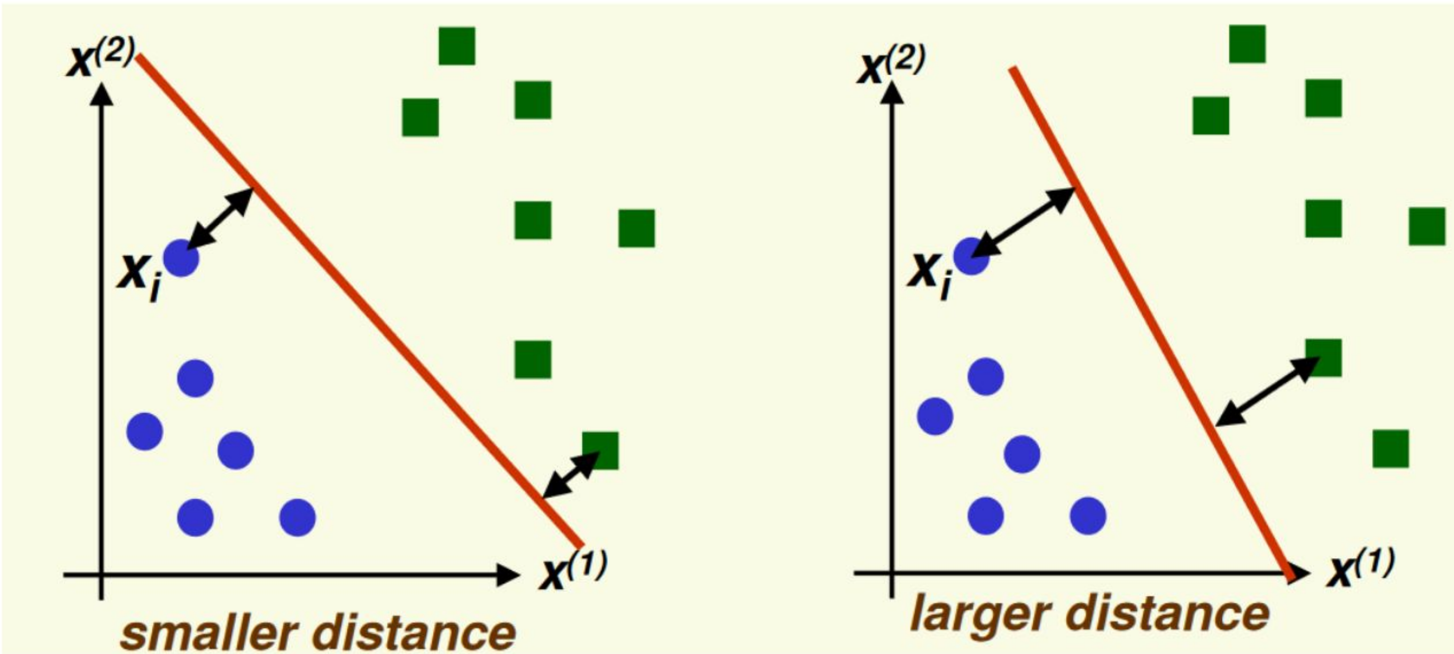
Castro, Marc Jerrone R.
B.S. Applied Physics
2015-07420

Support Vector Machines



Common Use & Purpose: Support Vector Machines are discriminative classifiers uniquely described by separating hyperplanes.

Given an input of pairs of the features of positive and negative inputs (e.g. features of apples and bananas) outputs an optimized hyperplane which acts a plane in two parts which splits the two classes to corresponding sides.



Support Vector Machines

Objective: Support Vector Machines are designed such that a corresponding margin between the determined separating plane and data points called support vectors is maximized

Algorithm

The best decision line to separate each type of data is given by:

$$g(\boldsymbol{x}) = w_o + \boldsymbol{x}^T \boldsymbol{w} = 0$$

where **w** is the optimized weight for our initial bias, **x** corresponding to the feature space with **w** weights. We then subject this line to the following constraints as to maximize the distance:

$$g(\boldsymbol{x}) = \begin{cases} w^t x_i + w_0 \geq 1 & \text{if } x_i \text{ is positive example} \\ w^t x_i + w_0 \leq -1 & \text{if } x_i \text{ is negative example} \end{cases}$$

and set corresponding classification array **y** where it is equal to 1 for one class, and -1 for the other. Using the Kuhn-Tucker theorem to simplify the conditions of our optimizer, we have the following problem:

$$\text{maximize} \quad L_D(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j z_i z_j x_i^t x_j$$

$$\text{constrained to} \quad \alpha_i \geq 0 \quad \forall i \quad \text{and} \quad \sum_{i=1}^{n} \alpha_i z_i = 0$$

where alpha is given by the solution of the quadratic equation **L_d** whereas it corresponds to coefficients for each feature value pair presented by our input.

Algorithm

Each value of **alpha i**s given by a certain pre-determined threshold whereas, all values lower than it are equated to 0. The zero values of **alpha** are considered as the support vectors of our SVM model.

We can then determine **w** by using the following equation:

$$\mathbf{w} = \sum_{i=1}^{n} \alpha_i z_i \mathbf{x}_i = (\alpha .* \mathbf{z})^t \mathbf{x}$$

and the corresponding bias weight, **w_0,** via,

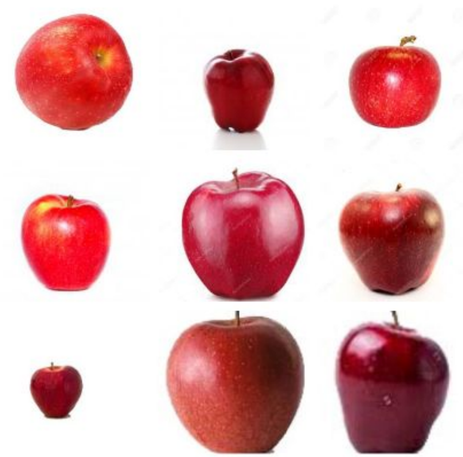
$$w_0 = \frac{1}{z_1} - \mathbf{w}^t \mathbf{x}_1$$

Given these variables, we can now calculate for the slope and y-intercept of the line. Where **A = w_0** , **B = w_1** , and **C = w_2**.

$$Ax+By=C \quad \text{or} \quad -C+ax+By=0 \ .$$

In terms of slope and y-intercept form, $y=\frac{C}{B}-\frac{A}{B}x$ where $m=\frac{-A}{B}$ and $b=\frac{C}{B}$.

Data and Inputs



The data used for this study is a set of images of fruits (apples and bananas) taken from Google Images.

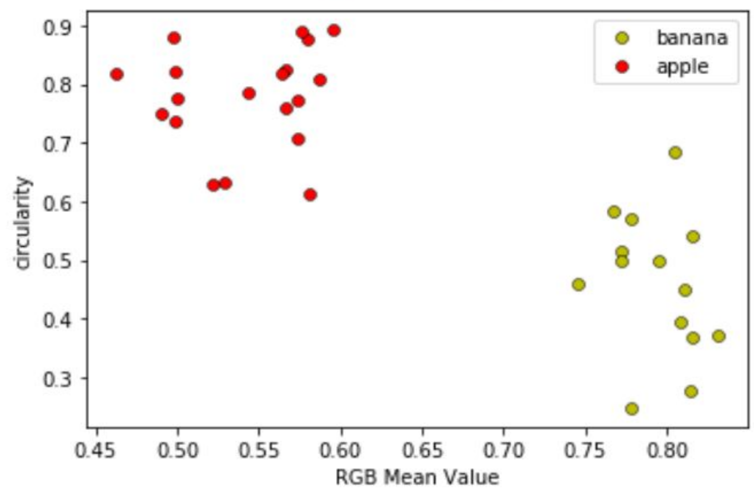
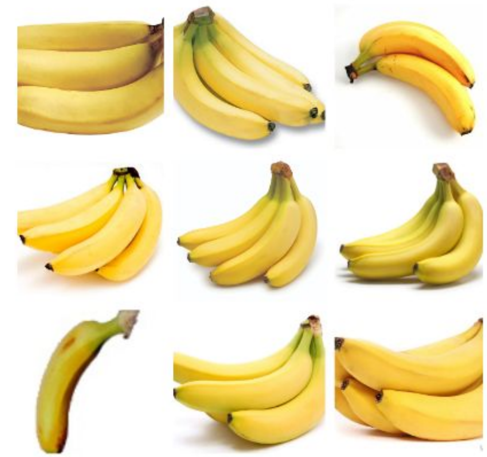


Figure 1. Feature space of banana and apple properties. This particular figure presents the comparison of the average pixel value and circularity for each sample image.

Using the same pre-processing steps from Activity 11 - Feature Extraction, image features such as average pixel value and circularity were extracted from each image.



Support Vector Machine Results

```python
xdat = np.concatenate((apple['rgbmean'], banana['rgbmean']))
ydat = np.concatenate(( apple['circularity'], banana['circularity']))
X = np.column_stack((xdat, ydat))
z = np.concatenate((np.ones(len(apple['rgbmean'])), -np.ones(len(banana['rgbmean']))))
n_samples, n_features = X.shape
```

```python
H = ((X.dot(X.T))*(z.dot(z.T)))
f = -np.ones(len(X))
A = -np.identity(len(X))
a = np.zeros(len(X))
B = np.vstack((z, np.zeros((len(z)-1, len(z)))))
b = np.zeros(len(X))
```

```python
alpha = cvx.Variable(len(X))
objective = cvx.Minimize(0.5*cvx.quad_form(alpha, H) + f.T*alpha)
constraints = [A*alpha <= a, B*alpha == b]
problem = cvx.Problem(objective, constraints)
problem.solve(verbose=True)
alpha = alpha.value
print(alpha)
alpha[alpha < 1e-12] = 0
alpha
```

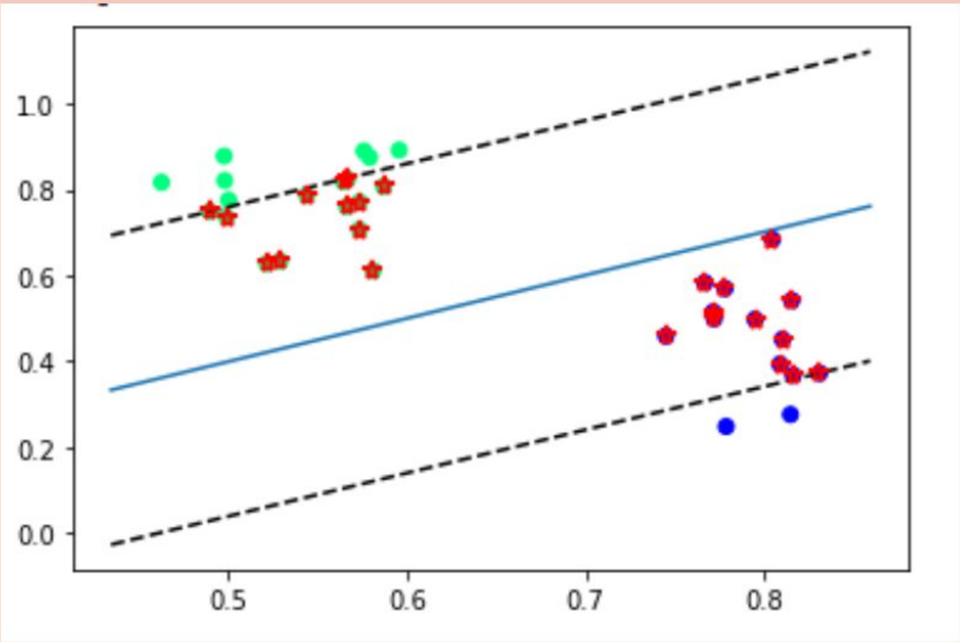


Figure 1. Feature space of banana and apple properties. This particular figure presents the comparison of the average pixel value and circularity for each sample image.

You could include up to 20 lines of text,
as long as you break them up into paragraphs to make it easier to read.

The photo on the left can be replaced with other photos that are relevant to your topic. We recommend having one to capture the attention of your reader.

Support Vector Machine Predictions.

```
xdat = np.concatenate((apple['rgbmean'], banana['rgbmean']))
ydat = np.concatenate(( apple['circularity'], banana['circularity']))
X = np.column_stack((xdat, ydat))
z = np.concatenate((np.ones(len(apple['rgbmean'])), -np.ones(len(banana['rgbmean']))))
n_samples, n_features = X.shape
```

```
H = ((X.dot(X.T))*(z.dot(z.T)))
f = -np.ones(len(X))
A = -np.identity(len(X))
a = np.zeros(len(X))
B = np.vstack((z, np.zeros((len(z)-1, len(z)))))
b = np.zeros(len(X))
```

```
alpha = cvx.Variable(len(X))
objective = cvx.Minimize(0.5*cvx.quad_form(alpha, H) + f.T*alpha)
constraints = [A*alpha <= a, B*alpha == b]
problem = cvx.Problem(objective, constraints)
problem.solve(verbose=True)
alpha = alpha.value
print(alpha)
alpha[alpha < 1e-12] = 0
alpha
```

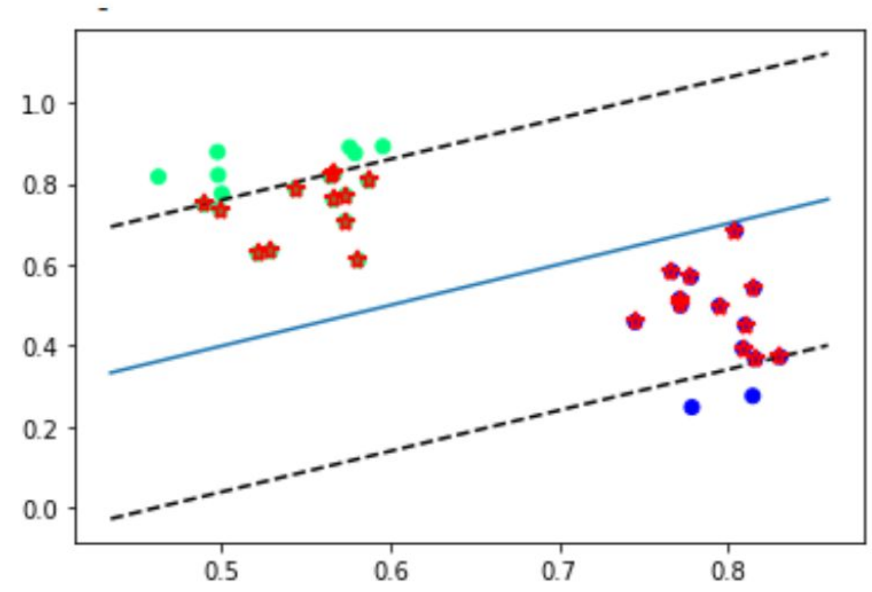


Figure 2. Decision plane along the apple and banana feature space with margins equal to **2/||w||.** The data points annotated by red stars are members of the support vectors which suggests that these **alpha v**alues are equal to zero. As one may observe, these were located within the margins of each datatype. However, my results failed to depict maximum distance in between each class as data points for the banana were found closer to decision line in contrast to the data points for the apples. Further optimization may be performed to lessen this distance or maximize the distance in between with the corresponding constraints mentioned earlier in the report. Increasing the number of datapoints would also assist in the optimization of the support vector machine.

Applied Physics 186
Activity 16

QUALITY OF PRESENTATION: 5/5
TECHNICAL CORRECTNESS: 5/5
INITIATIVE: 0

Reference

1. Theodoridis, Chapter 3, Pattern Recognition
2. Veksler, O. , CS 434a/541a:Pattern Recognition Lecture 11 slides.