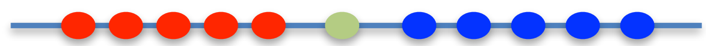
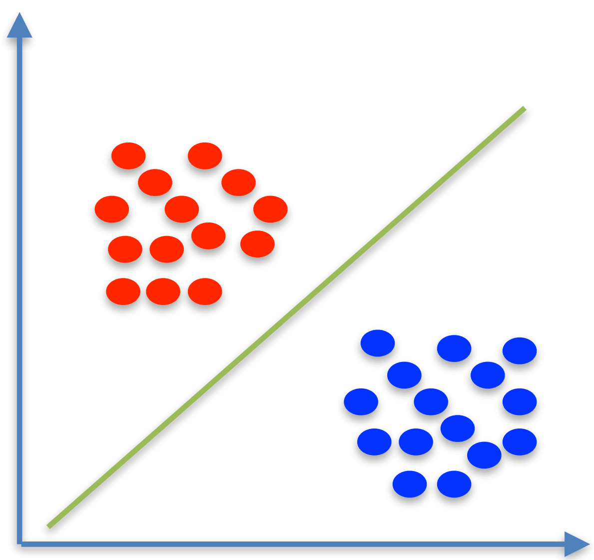
Support Vector Machine

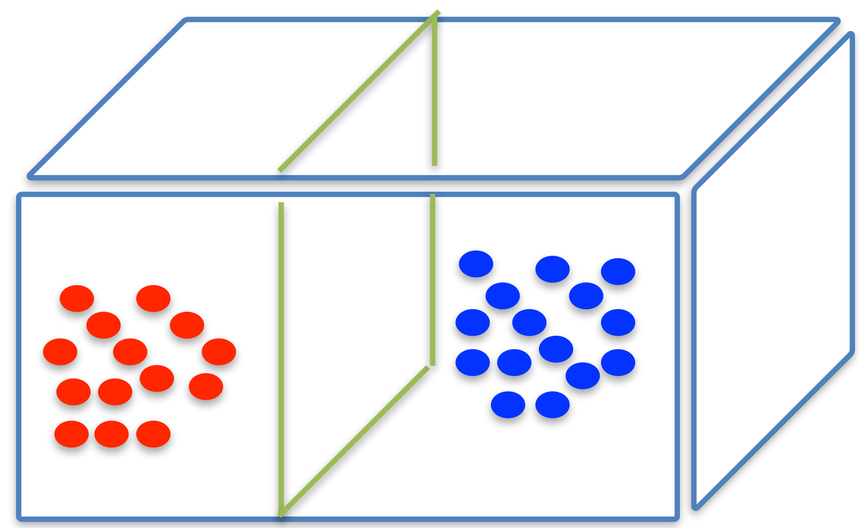
Raman Khurana
PhD in Physics
Researcher at Centre for Deep Learning

Northwestern

Classification Problem in n-dimension

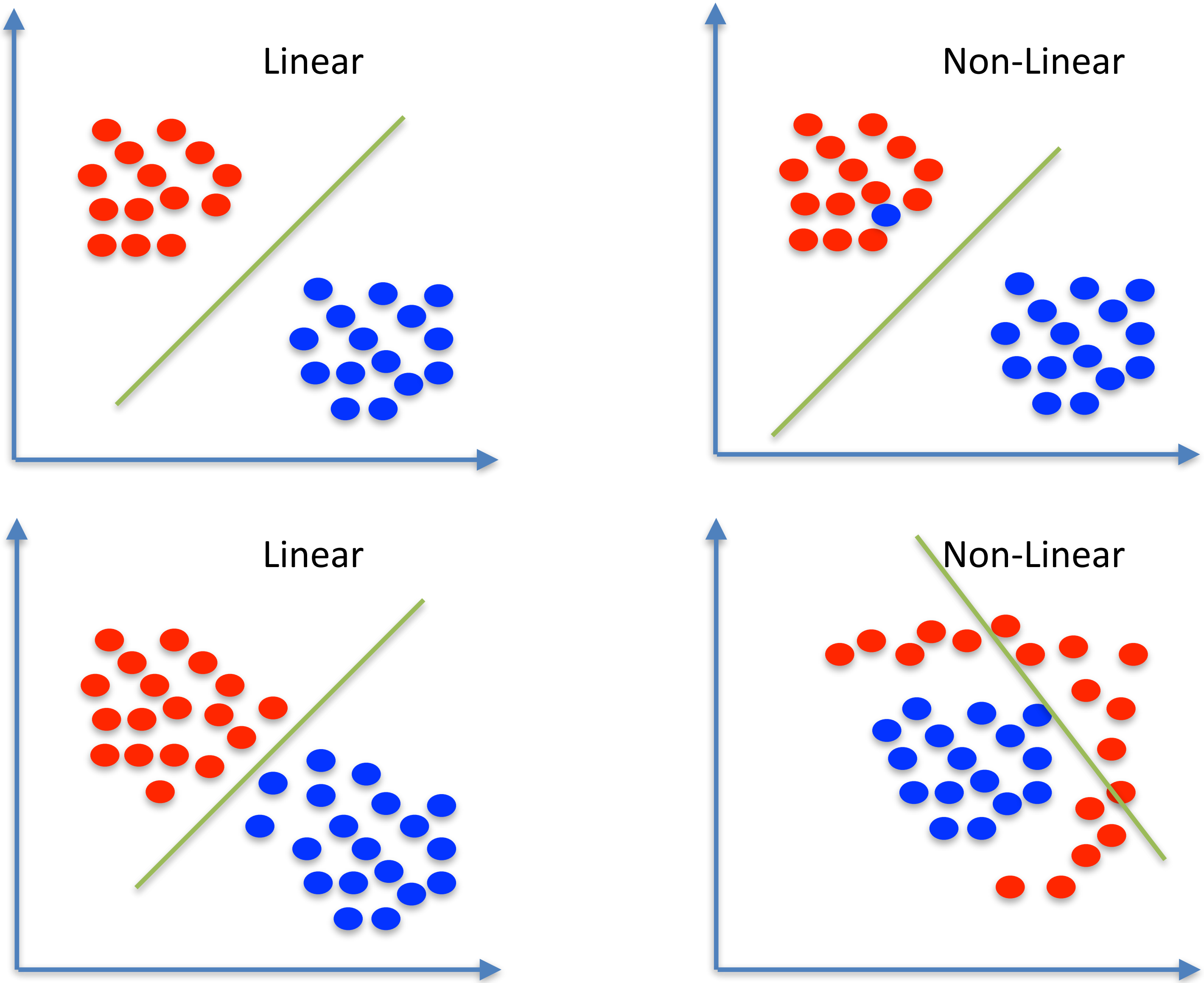




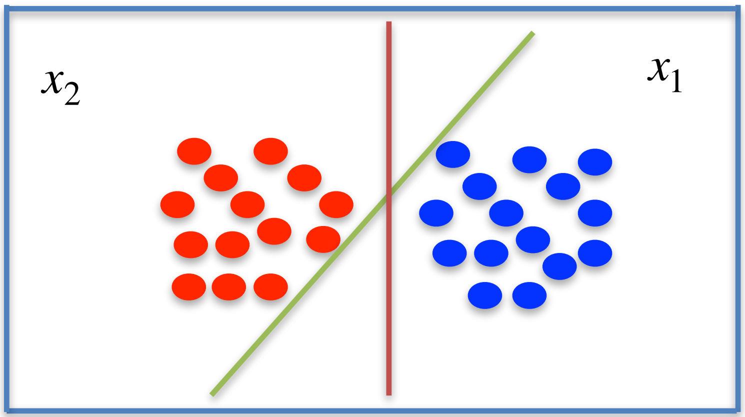


For n-dimensional feature space one can find a n-1 dimensional hyperplane to classify data.

Linear separability



Find the best line

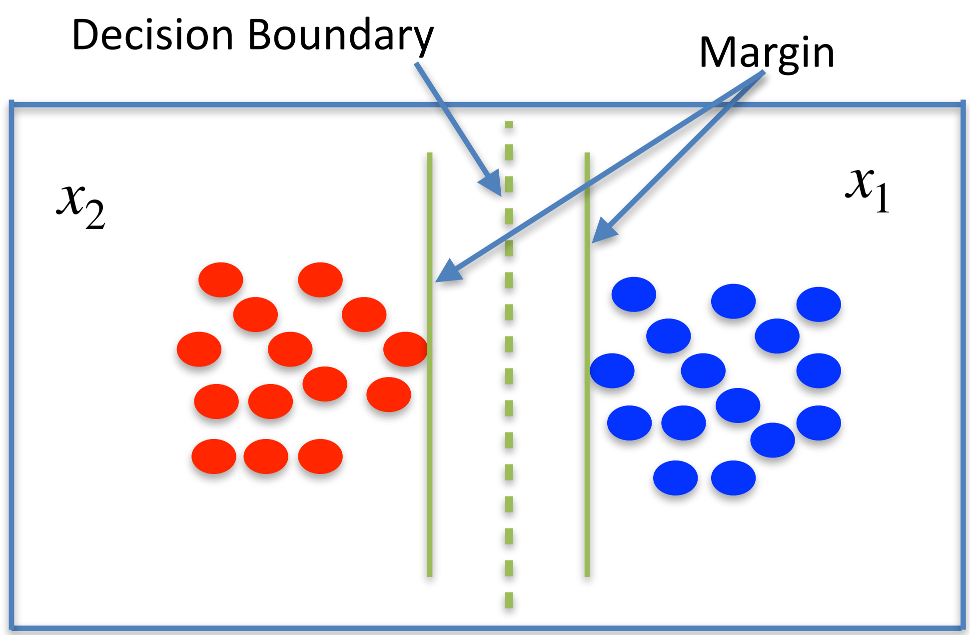


One can draw an infinite straight lines that will classify this data.

But How do I find the best one?

Support vector machine (SVM): Idea

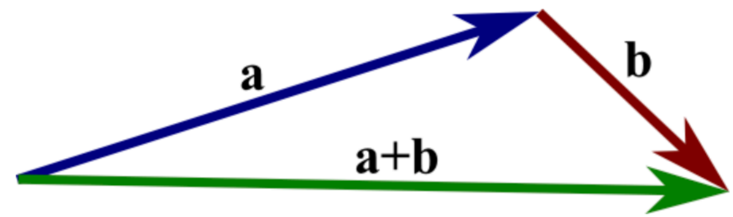
- Vladimir Vapnik wanted to separate two classes using a straight line.
- Used the widest street approach.
- Place the decision boundary in such a way that the separation b/w classes is the widest.



SVM: Prerequisite

What is a vector?

Triangle law of vector addition.

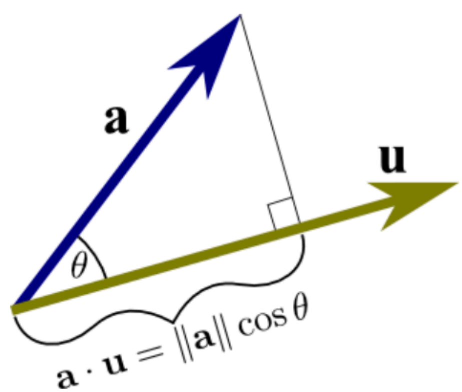


Unit vector.

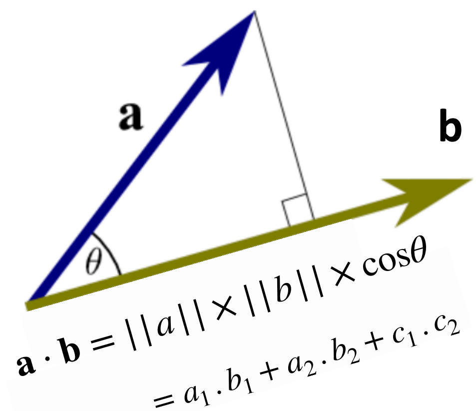
$$||b|| = magnitude\ of\ \overrightarrow{b}$$

$$\hat{b} = \frac{\overrightarrow{b}}{||b||} = \mathbf{u}$$

Projection on Unit vector

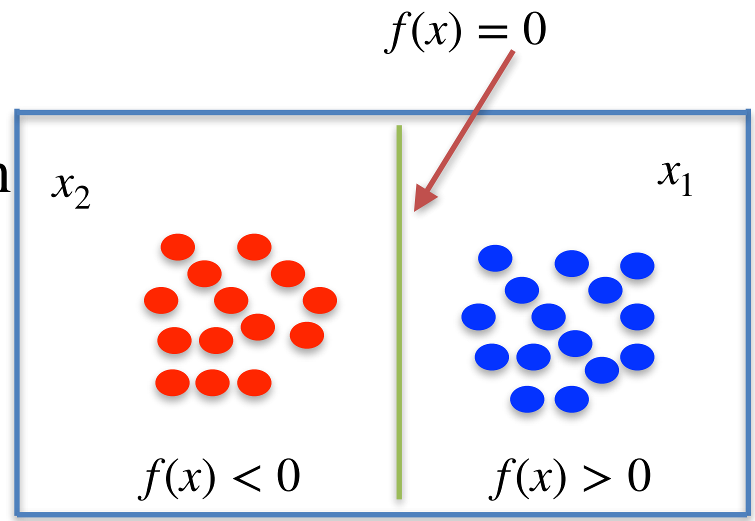


Scalar or dot or inner product.



Linear classifiers

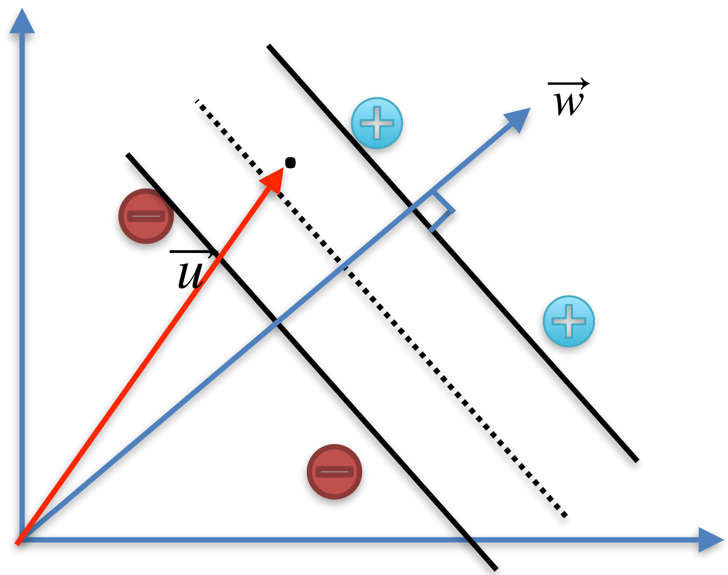
- A Linear classifier in 2d is straight line of the form: $y = mx + c$; eqn of line
  - $\implies ax + by + c = 0$; standard eqn
  - If $w = (a, b)^T$ and $\mathbf{x} = (x, y)$;
  - $w^T \cdot \mathbf{x} + c = 0$



- In 3d space, classifier is a 2d plane with a generic equation $ax + by + cz + d = 0.$
  - $w = (a, b, c)^T$ and $\mathbf{x} = (x, y, z)$;
  - $w^T \cdot \mathbf{x} + d = 0$

- In n-dimensional space,
  - $f(x) = w^T \cdot \mathbf{x} + b$; b is bias term

SVM: Decision Rule

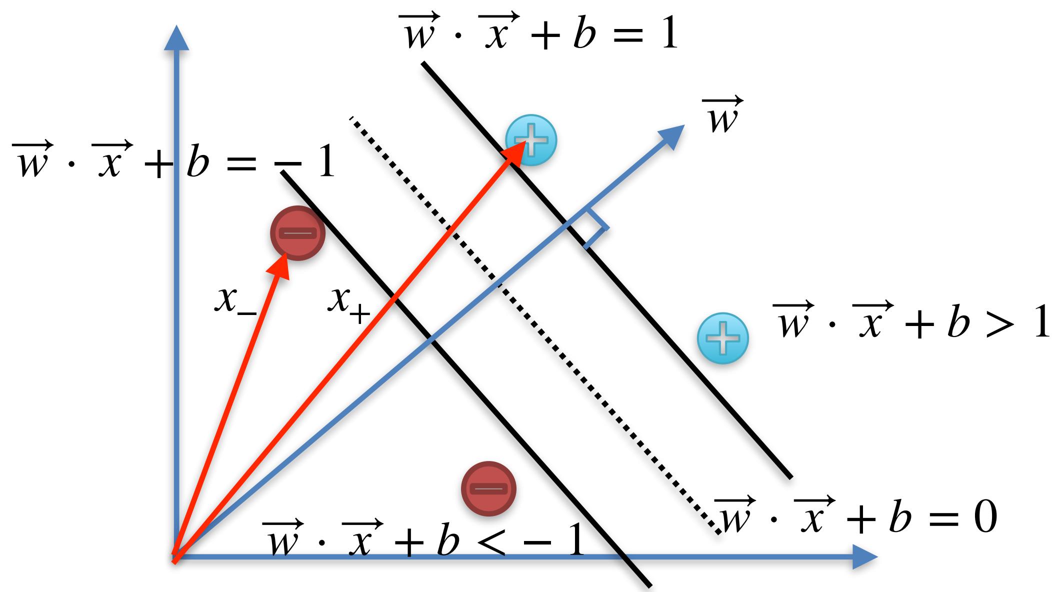


$$\Longrightarrow \quad \vec{w} \cdot \vec{u} + b \geq 0, \quad \text{then it is } +$$

Decision Rule

We don't know what are $\vec{w}$ and b

SVM: Constraints



$$\vec{w} \cdot \overrightarrow{x_+} + b \geq +1,$$
$$\vec{w} \cdot \overrightarrow{x_-} + b \leq -1$$

For mathematical convenience assume that the separation between x_- and x_+ is from -1 to +1.

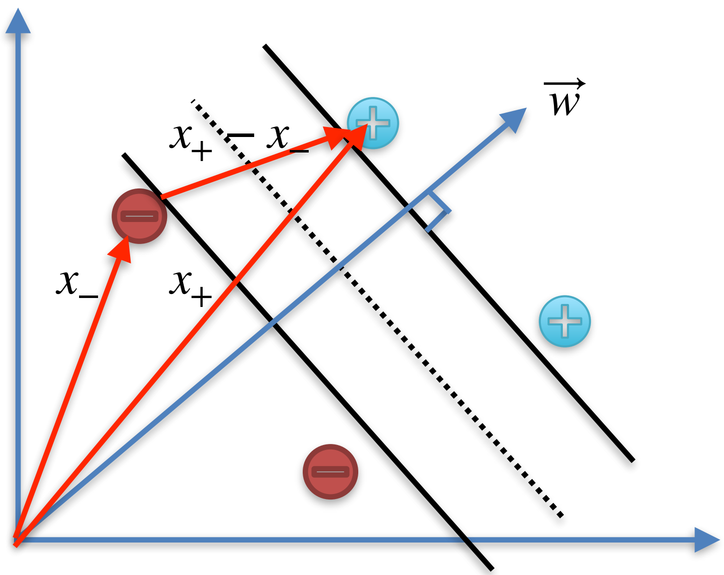
$$y_i(\vec{w} \cdot \vec{x_i} + b) \geq +1, \ where\ x_i\ is\ x_+, x_-$$
$$y_i = +1\ for\ \ +ve\ samples$$
$$y_i = -1\ for\ \ -ve\ samples$$

$$\boxed{y_i(\vec{w} \cdot \vec{x_i} + b) - 1 = 0}$$

For x_i to be exactly on the margin line.

SVM: Width of the street

$$\boxed{y_i(\vec{w} \cdot \vec{x_i} + b) - 1 = 0}$$



Width = $(x_+ - x_-) \cdot \dfrac{\vec{w}}{||w||}$

$1 - b$ $\quad$ $1 + b$

Width = $\dfrac{2}{||w||}$ ← Maximize $\Longrightarrow$ Minimize $\dfrac{||w||}{2}$

$\Longrightarrow$ Minimize $\dfrac{||w||^2}{2}$

Revisit

$$\Longrightarrow \ \vec{w} \cdot \vec{u} + b \geq 0, \ \ where \ \ b = -c$$ then it is ⊕

Decision Rule

..................Equation 1

$$y_i(\vec{w} \cdot \vec{x_i} + b) - 1 \geq 0$$

For x_i to be correctly classified.

..................Equation 2

$\Longrightarrow$ Minimize $\dfrac{||w||^2}{2}$

..................Equation 3

Aim: minimize Equation 3 with constraint in Equation 2.

Lagrange Multiplier

- Find extremum of a function with constraints
- Formulate Lagrangian using objective function and all the constraints.

Plan of action

L = function to maximize (or minimize) - $\Sigma \alpha_i \,.\, R_i$

↓

Take partial derivative w.r.t each variable and set to ZERO

↓

Rearrange expression of L using all the new information

Simplifying the optimization problem

$$L = \frac{1}{2}||w||^2 - \Sigma\alpha_i[y_i(\vec{w}\cdot\vec{x}+b)-1]$$

α's are non-zero for samples on the margin. For all other samples α are zero. These are called support vectors.

$$\frac{\delta L}{\delta \vec{w}} = \vec{w} - \Sigma\alpha_i y_i \vec{x_i} = 0$$

$$\implies \vec{w} = \Sigma\alpha_i y_i \vec{x_i}$$

$$\frac{\delta L}{\delta b} = \Sigma\alpha_i y_i = 0$$

$$L = \frac{1}{2}(\Sigma\alpha_i y_i \vec{x_i})\cdot(\Sigma\alpha_j y_j \vec{x_j}) - (\Sigma\alpha_i y_i \vec{x_i})\cdot(\Sigma\alpha_j y_j \vec{x_j}) - \Sigma\alpha_i y_i + \Sigma\alpha_i$$

$$\boxed{L = \Sigma\alpha_i - \frac{1}{2}\Sigma\Sigma\alpha_i\alpha_j\ y_i y_j\ \vec{x_i}\cdot\vec{x_j}}$$

Exercise 1: Find expression for bias term, b

$$b = y_i - \sum_j \alpha_j y_j(\mathbf{x_j}\cdot\mathbf{x_i})$$

What do we have so far?

$$\Longrightarrow \ \vec{w} \cdot \vec{u} + b \geq 0, \quad where \quad b = -c$$ then it is ⊕

Decision Rule

..................Equation 1

$$y_i(\vec{w} \cdot \vec{x_i} + b) - 1 \geq 0$$

For x_i to be correctly classified.

..................Equation 2

$\Longrightarrow$ Minimize $\dfrac{||w||^2}{2}$

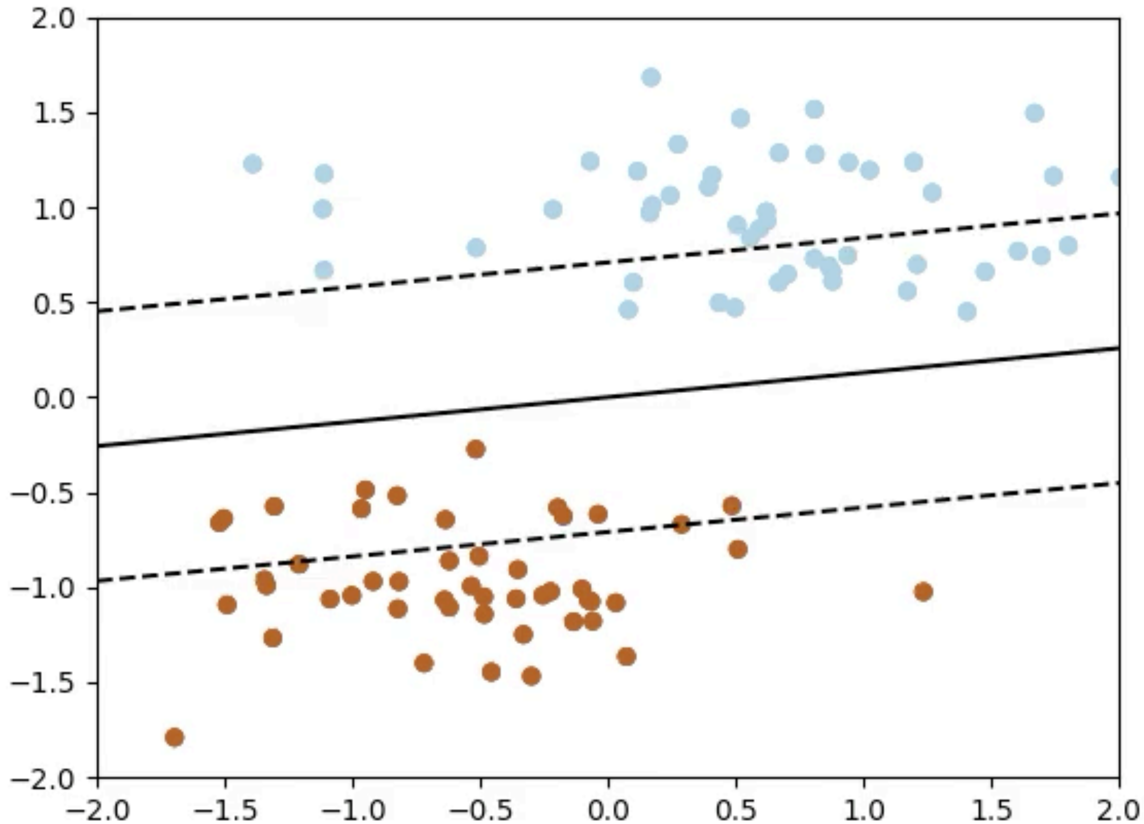
..................Equation 3

$$L = \Sigma\alpha_i - \frac{1}{2}\Sigma\Sigma\alpha_i\alpha_j \ y_i y_j \ \vec{x_i} \cdot \vec{x_j}$$

....Equation 4

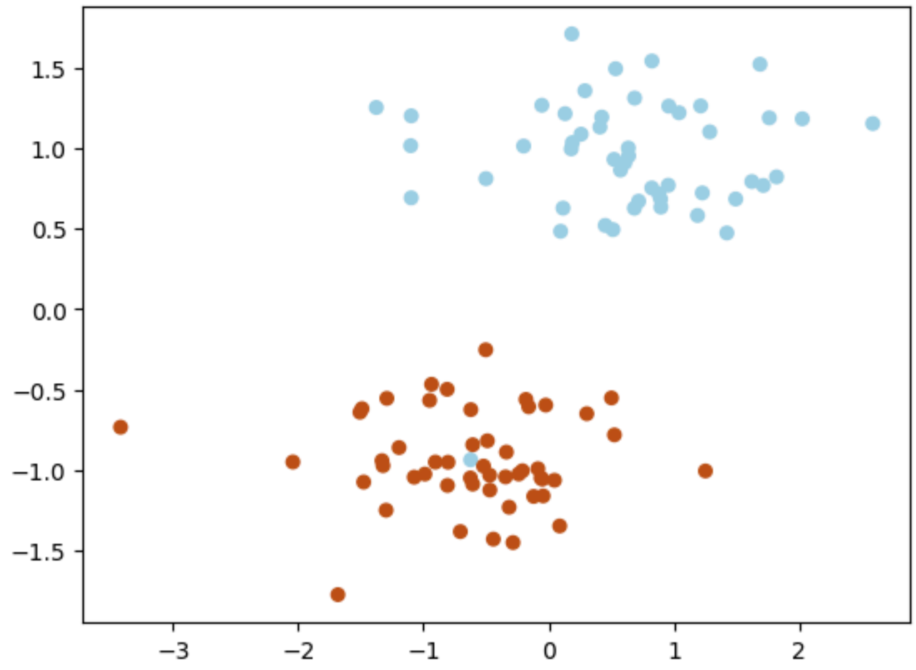
The problem is now simplified to find maximum of equation 4, i.e. optimization depends only on the dot product of pair of samples

SVM at action



Support vector are the data points/samples which lie on the boundary of margin.

Noisy/Non-linear data



- Hard margin will never converge due to shifting of data point.
- Model will never find a minima and hence no decision boundary to completely separate two classes.

Practical solution: Ignore this one data point, by allowing a certain degree of misclassification.

Soft Margin

$$\Longrightarrow \quad \vec{w} \cdot \vec{u} + b \geq 0, \quad where \quad b = -c$$ then it is ⊕

Decision Rule

$$y_i(\vec{w} \cdot \vec{x_i} + b) - 1 \geq 0$$

$$y_i(\vec{w} \cdot \vec{x_i} + b) - (1 - \zeta_i) \geq 0$$
$$\zeta_i > 0 \ \forall \ i$$

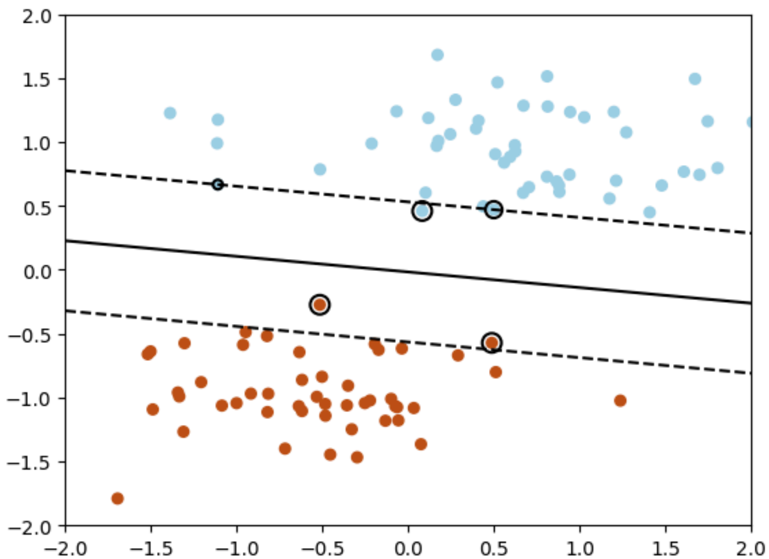
$\Longrightarrow$Minimize $\dfrac{||w||^2}{2}$

$\Longrightarrow$Minimize $\dfrac{||w||^2}{2} + C \sum_{i=1}^{n} \zeta_i$

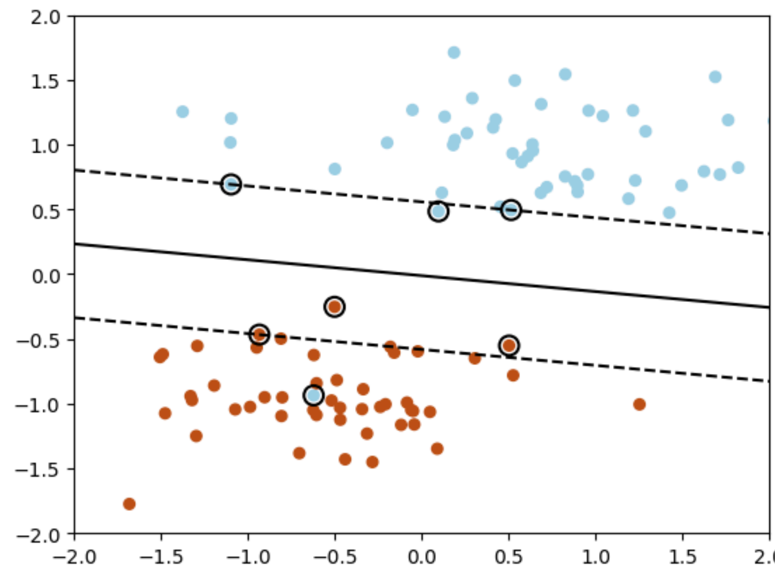
$$L = \Sigma\alpha_i - \frac{1}{2}\Sigma\Sigma\alpha_i\alpha_j \ y_i y_j \ \vec{x_i} \cdot \vec{x_j}$$

Soft Margin

- Two scenarios
  - Correctly classified data point within the margin
    - $0 < \zeta_i < 1$



  - Misclassified data point on wrong side of margin
    - $\zeta_i \geq 1$



Support vector are the data points/samples which lie **on the decision boundary** or **in the margin** or **mis-classified**.

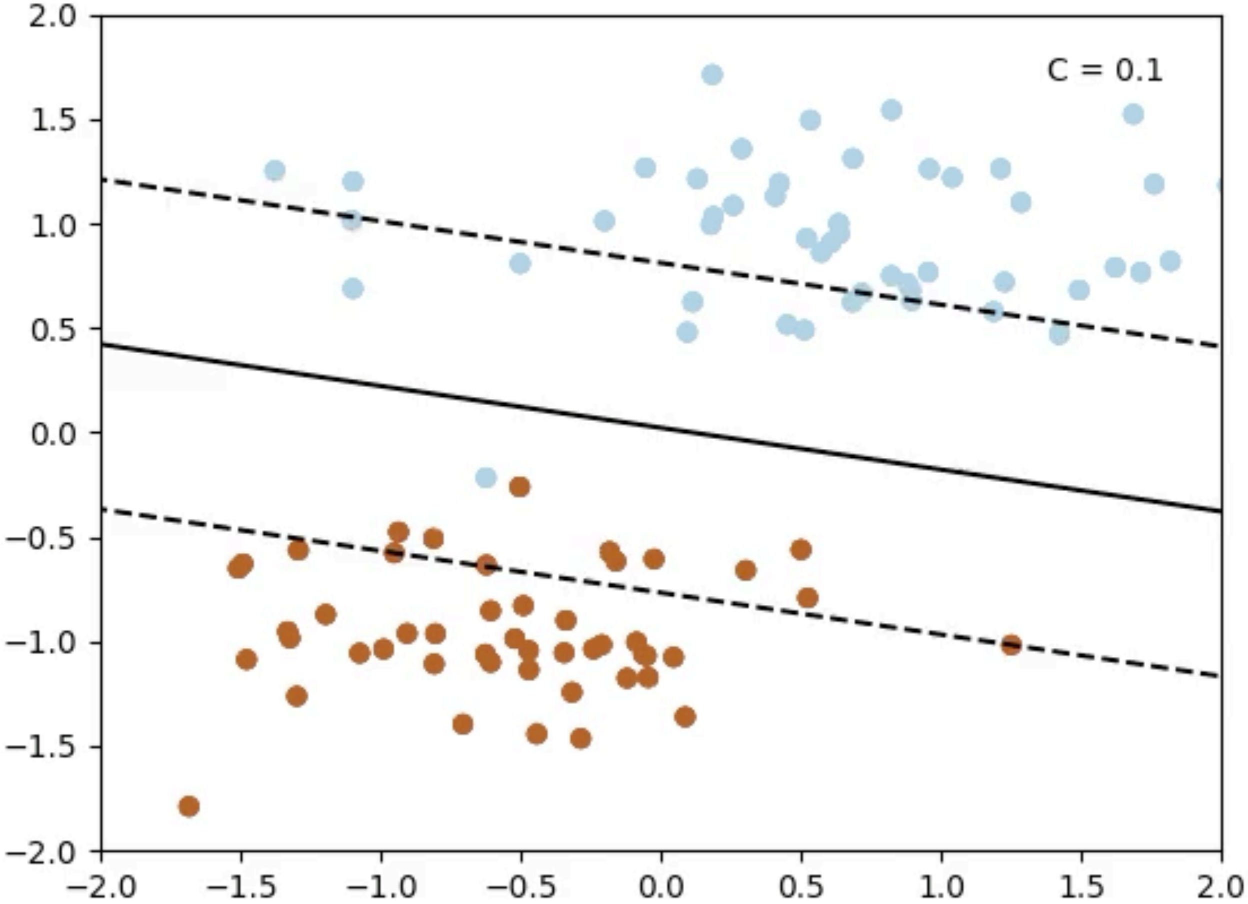
Importance of C

- C is the hyper parameter of model which controls tradeoff b/w margin width and classification error.

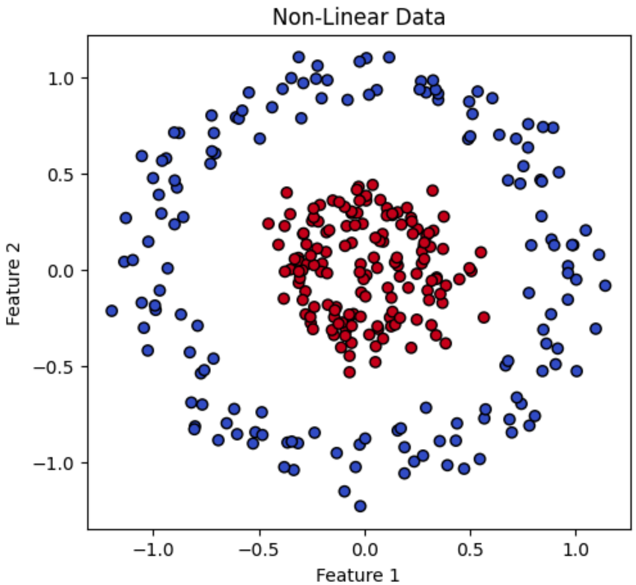
Minimize $\frac{||w||^2}{2} + C\sum_{i=1}^{n}\zeta_i$

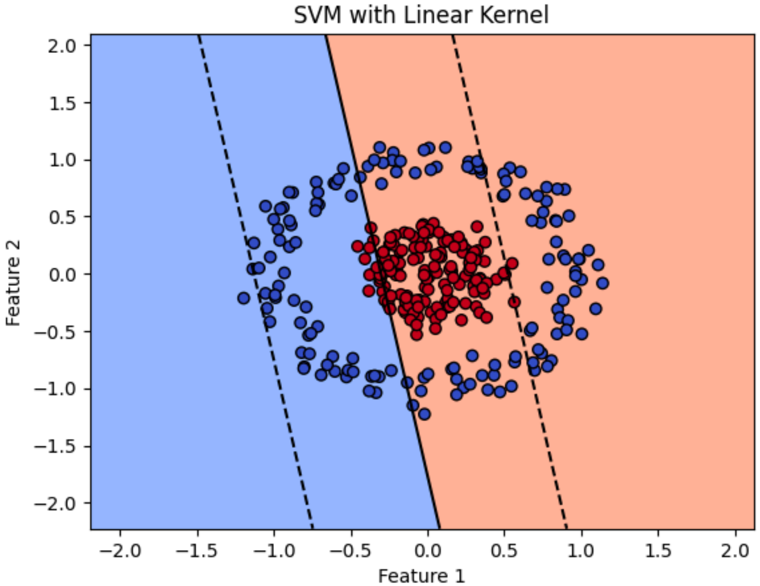
- After introducing slack variable in the Lagrangian it is feasible to find optimized solution even if some data points are misclassified.
- Large C: classification error has more weightage $\Longrightarrow$model tries to minimize mis-classification of classes $\Longrightarrow$the margin width can be smaller $\Longrightarrow$sensitive to small variation in the (unseen) data $\Longrightarrow$can not be generalized on unseen data [**Overfitting**]
- Small C: allows misclassification of class $\Longrightarrow$model will tolerate errors for larger margin width $\Longrightarrow$better generalization [**less likely to overfit, but more likely to underfit**]

Importance of C



Kernel Trick





A linear classifier will not be generalized enough to deal with non-linear data.

$$L = \Sigma\alpha_i - \frac{1}{2}\Sigma\Sigma\alpha_i\alpha_j \; y_i y_j \; \overrightarrow{x_i} \cdot \overrightarrow{x_j}$$

$$\overrightarrow{x_i} \rightarrow \phi(\overrightarrow{x_i}) \; and \; \overrightarrow{x_j} \rightarrow \phi(\overrightarrow{x_j})$$

$$\overrightarrow{x_i} \cdot \overrightarrow{x_j} \rightarrow \phi(\overrightarrow{x_i}) \cdot \phi(\overrightarrow{x_j})$$

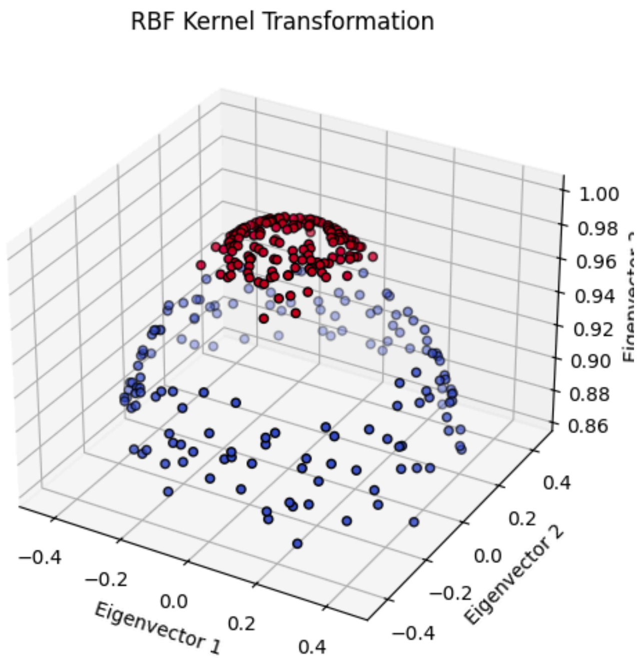
$$\boxed{\overrightarrow{x_i} \cdot \overrightarrow{x_j} \rightarrow K(\overrightarrow{x_i}, \overrightarrow{x_j}) = \phi(\overrightarrow{x_i}) \cdot \phi(\overrightarrow{x_j})}$$

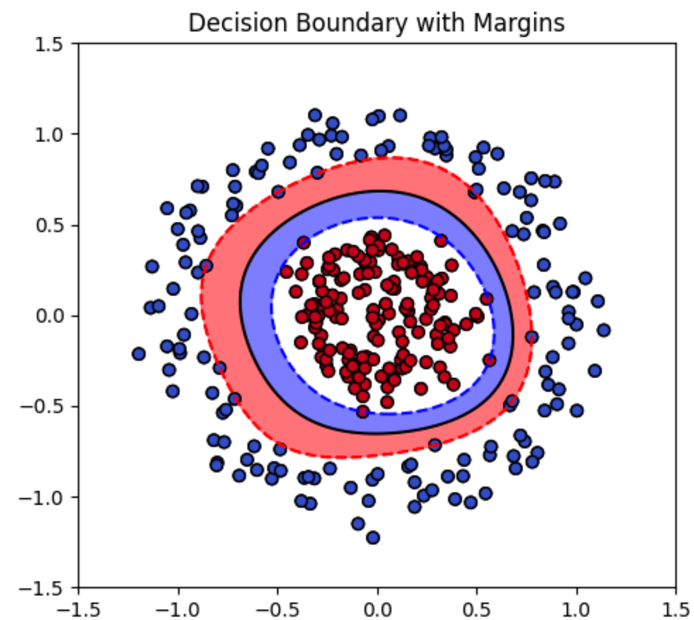
Kernel Trick

$$\boxed{\overrightarrow{x_i} \cdot \overrightarrow{x_j} \rightarrow K(\overrightarrow{x_i}, \overrightarrow{x_j}) = \phi(\overrightarrow{x_i}) \cdot \phi(\overrightarrow{x_j})}$$

Kernel function

- **Kernel Trick**: Kernel Function computes the dot product of two vectors in a potentially higher dimensional feature space without explicitly performing the transformation into that higher dimensional space.





Common Kernels

- **Linear**:

$$K(\overrightarrow{x_i}, \overrightarrow{x_j}) = \overrightarrow{x_i} \cdot \overrightarrow{x_j}$$

  - The simplest kernel to be used when data is linearly separable.
  - # of features > # of data points.
- **Polynomial**:

$K(x_i, x_j) = (x_i \cdot x_j + c)^d$; c is constant and d is degree of polynomial.

  - Use for non-linear case; complexity of model controlled by parameter d
- **Radial Basis Function (RBF) or Gaussian Function**:

$$K(x_i, x_j) = \exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right)$$

  - Another non-linear kernel transforming data to infinite dimensional space.
  - Transformation to Infinite dimensional space allows the modeling of complex data.

Exercise 2: Try different kernel for circular ring dataset and find which kernel is the most suitable.

Key points

- Pros
  - Dataset size is small
  - # of features > # of samples
  - Robust (not immune) to overfitting; if used carefully
  - Memory efficient; uses only a subset of samples
  - Kernel trick; handles non-linear data.
  - Effective for binary classification
  - Better interpretability
- Cons
  - When dataset is very large
  - # of feature >> # of samples
  - Choice of right Kernel
  - Computational expensive for large dataset or large number of features.

Further reading

- Support Vector Networks, Corinna Cortes & Vladimir Vapnik (1995)
- The Nature of Statistical Learning Theory :
- SVM in sklearn
- A tutorial on Support Vector Regression: Alex J Smola
- Support Vector Machine: Hype or Hallelujah
- Code and Slides

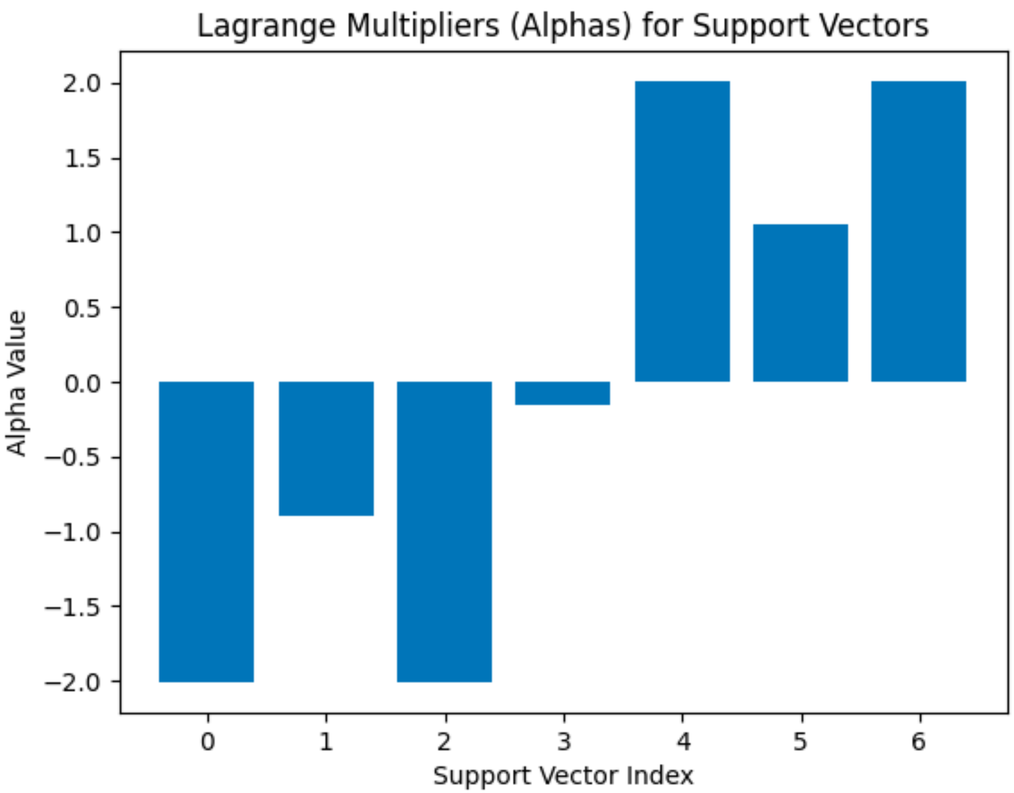
Exercise 3: Take MNIST handwritten digit data and compare the results of linear vs RBF kernel.

Exercise 4: Apply SVM for a regression task.

Quick reminder

- A powerful supervised learning algorithm used primarily for classification task.
- Finds the best hyperplane that classify the data.
  - Maximum width street ensures the generalization of the model
- The Lagrangian/Objective function depends on the dot product of pair of vectors.
- Kernel trick transform the non-linearly separable data into higher dimensional space where it become linearly separable.
- A slack variable (ζ) allows for limited misclassification, making the model more robust against noisy data or a few non-linearly separable data points.
- Regularization parameter (C) controls the tradeoff between margin width and classification error.
- Support vectors are datapoints which are on the margin, in the street, or mis-classified.

Dual coefficients




```
1  dual_coefs
✓ 0.0s
array([-2.01      , -0.89680413, -2.01      , -0.15962996,  2.01      ,
        1.05643409,  2.01      ])
```

```
1  np.sum(dual_coefs)
✓ 0.0s
4.440892098500626e-16
```

$$\frac{\delta L}{\delta b} = \Sigma \alpha_i y_i = 0$$

Class weighted sum of Lagrange Multipliers is zero.

Kernel Trick

$$\overrightarrow{x_i} \cdot \overrightarrow{x_j} \rightarrow K(\overrightarrow{x_i}, \overrightarrow{x_j}) = \phi(\overrightarrow{x_i}) \cdot \phi(\overrightarrow{x_j})$$

- The kernel trick is to perform the dot product of pair of vector in a higher dimensional space without actually transforming the data points itself.
  - Saves computation when there is huge data to be transformed.
- Why do we need generic kernels?
  - Data is not always simple to transform using known/simple mathematical equations.
  - Generalized kernels help them separate in higher dimensional without knowing the exact mathematical formula.
    - However, you still need to know something about the data to make decision about the kernel.

Dimensionality in Kernel space

- Polynomial:
  - For n number of features and d as degree of polynomial the dimension of feature space after polynomial transformation is:

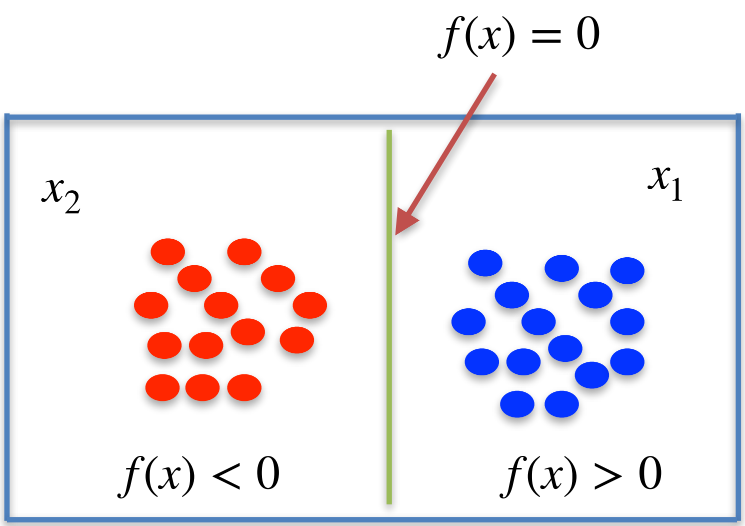
$$\frac{(n+d)!}{d!}$$

- Gaussian:
  - For any value of n and width of gaussian function the dimension of feature space after RBF transformation is INFINITE.
  - Gaussian function can be expressed as infinite sum using Taylor Series.

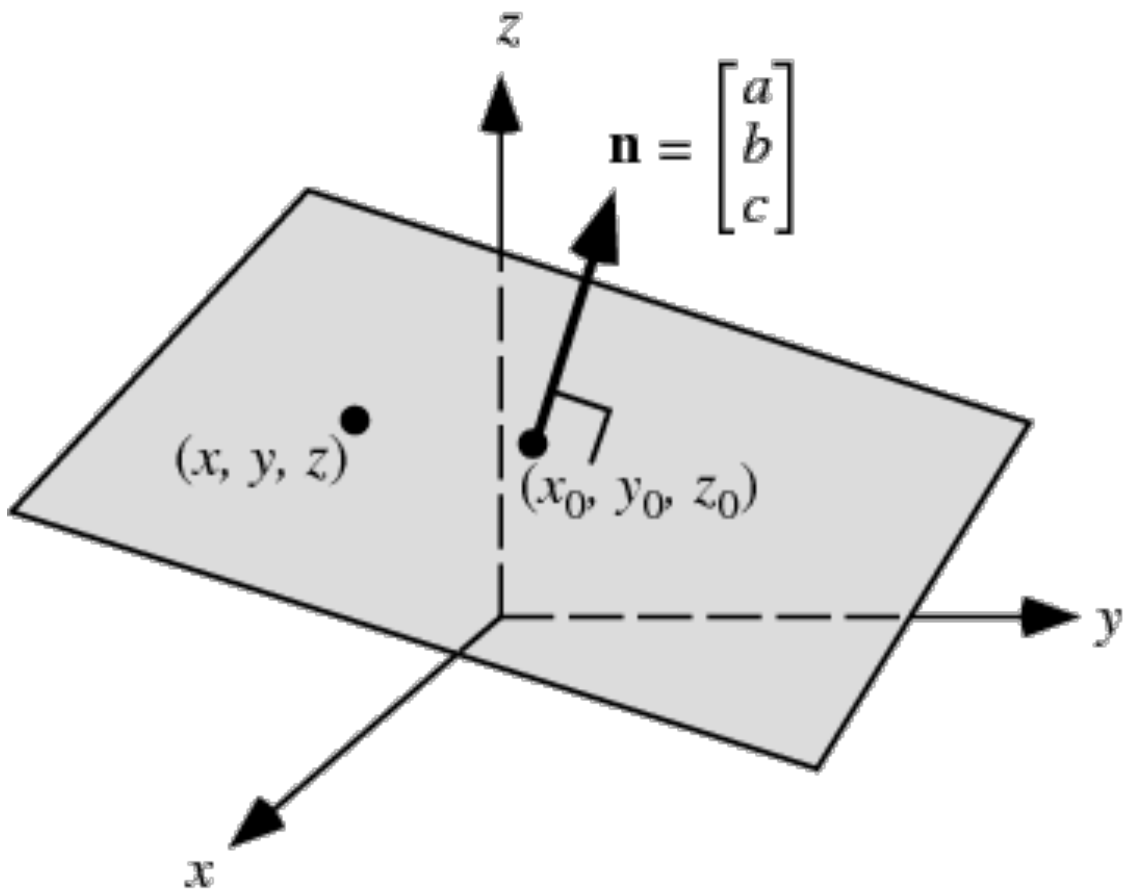
$$e^{-x^2} = \sum_{n=0}^{\infty} \frac{(-1)^n (x^2)^n}{n!}$$

Linear classifiers

- A Linear classifier has the form
  $$f(x) = w^T x + b$$



- In 2d the discriminant is a line.
- w is the normal to the discriminant line, and b is bias.
- w is also known as weight vector.


z
$\mathbf{n} = \begin{bmatrix} a \\ b \\ c \end{bmatrix}$
(x, y, z)
(x_0, y_0, z_0)
y
x

Find the best line