# Introduction to Data Science

## Support Vector Machine


Arijit Mondal

Dept. of Computer Science & Engineering

Indian Institute of Technology Patna

arijit@iitp.ac.in

# Support Vector Machine

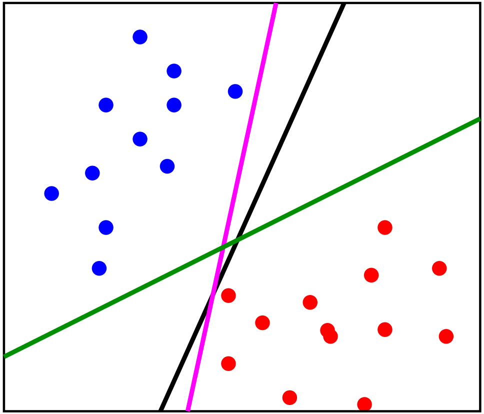
- An approach for classification
- Developed in 1990s
- Generalization of maximum margin classifier
  - Mostly limited to linear boundary
- Support vector classifier — broad range of classes
- SVM — Non-linear class boundary

# Hyperplane

- In $n$ dimensional space a hyperplane is a flat affine subspace of dimension $n-1$
- Mathematically it is defined as
  - For $2$ dimensions — $w_0 + w_1x_1 + w_2x_2 = 0$
  - For $n$ dimensions — $w_0 + w_1x_1 + \ldots + w_nx_n = 0$

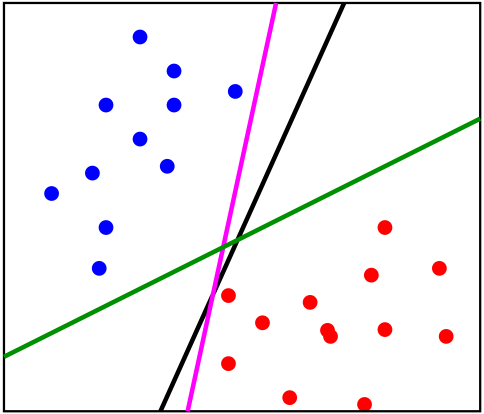
# Classification using Hyperplane

- Assume, $m$ training observation in $n$ dimensional space



# Classification using Hyperplane

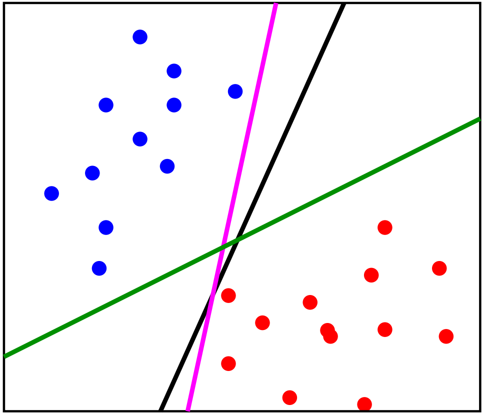
- Assume, $m$ training observation in $n$ dimensional space
- Separating hyperplane has the property
  - $w_0 + w_1 x_1 + \ldots + w_n x_n > 0$ if $y_i = 1$
  - $w_0 + w_1 x_1 + \ldots + w_n x_n < 0$ if $y_i = -1$



# Classification using Hyperplane

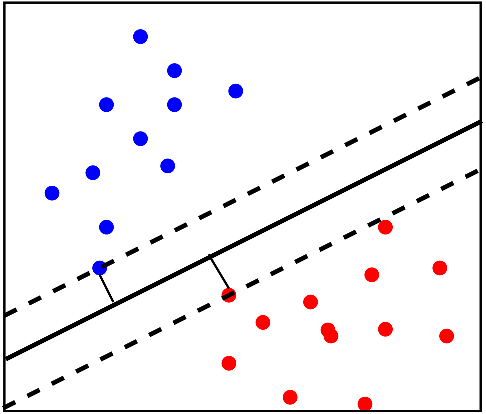
- Assume, $m$ training observation in $n$ dimensional space
- Separating hyperplane has the property
  - $w_0 + w_1 x_1 + \ldots + w_n x_n > 0$ if $y_i = 1$
  - $w_0 + w_1 x_1 + \ldots + w_n x_n < 0$ if $y_i = -1$
- Hence, $y_i(w_0 + w_1 x_1 + \ldots + w_n x_n) > 0$
- Classification of test observation $x^*$ is done based on the sign of

  $f(x^*) = w_0 + w_1 x_1^* + \ldots + w_n x_n^*$
- Magnitude of $f(x^*)$
  - Far from 0 — Confident about prediction
  - Close to 0 — Less certain



# Maximal margin classifier

- Also known as optimal separating hyperplane
- Separating hyperplane farthest from training observation
  - Compute perpendicular distance from training point to the hyperplane
  - Smallest of these distances represents the margin
- Target is to find the hyperplane for which the margin is the largest



# Construction of maximal margin classifier

- Input — $m$ points in $n$ dimension space ie. $\mathrm{x}_1, \mathrm{x}_2, \ldots, \mathrm{x}_m$
- Input — labels $y_1, y_2, \ldots, y_m$ for each point $\mathrm{x}_i$ where $y_i \in \{-1, 1\}$
- Need to solve the following optimization problem

$$\max_{w_0, w_1, \ldots, w_n, M} \quad M$$

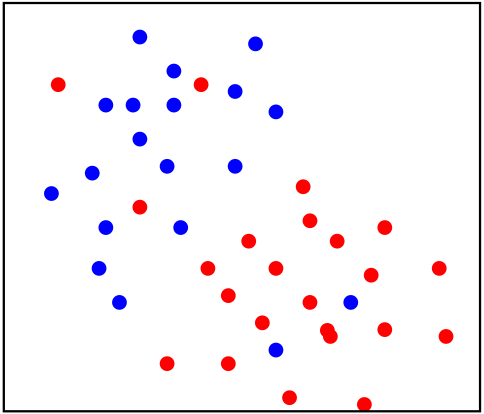
subject to

$$y_i(w_0 + w_1 x_{i1} + w_{i2} + \ldots + w_{in} x_{in}) \geq M \quad \forall i = 1, \ldots, m$$

$$\sum_{i=1}^{n} w_i^2 = 1$$

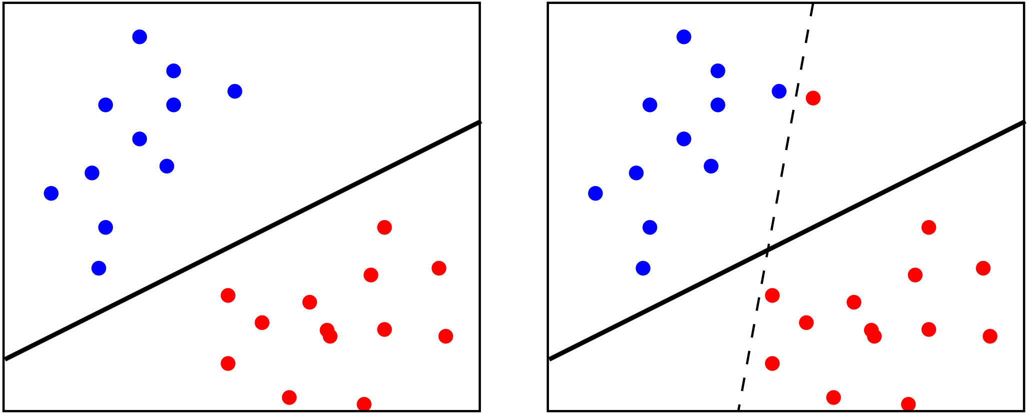
# Issues

- Maximal margin classifier fails to provide classification in case of overlap



# Issues

- Single observation point can change the hyperplane drastically

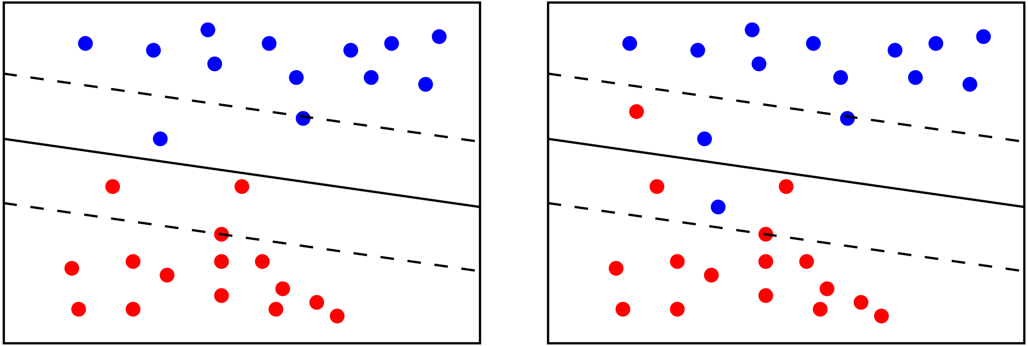


# Support Vector Classifier

- Provides greater robustness to individual observations
- Better classification of most of the training observations
- Worthwhile to misclassify a few training observations
- Also known as soft margin classifier

# Support Vector Classifier

- Points can lie within the margin or wrong side of hyperplane



# Optimization with misclassification

- Input — $\mathrm{x}_1, \mathrm{x}_2, \ldots, \mathrm{x}_m$ and $y_1, y_2, \ldots, y_m$
- Need to solve the following optimization problem

$$\max_{w_0, w_1, \ldots, w_n, M} \quad M$$

subject to

$$y_i(w_0 + w_1 x_{i1} + \ldots + w_{in} x_{in}) \geq M(1 - \epsilon_i) \quad \forall i = 1, \ldots, m$$

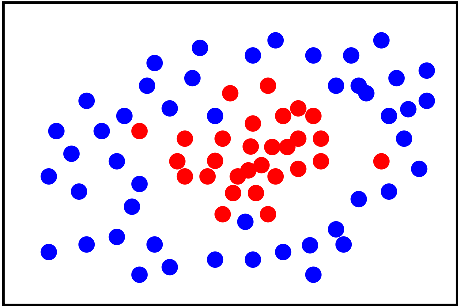
$$\sum_{i=1}^{n} w_i^2 = 1, \quad \sum_{i=1}^{m} \epsilon_i = C$$

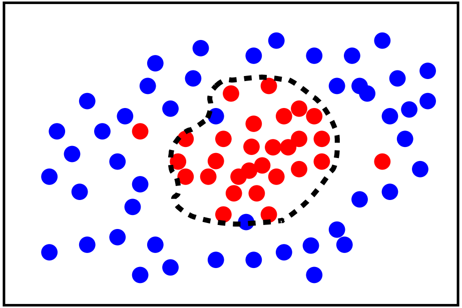
- $C$ is non-negative tuning parameter, $\epsilon_i$ - slack variable
- Classification of test observation remains the same

# Observations

- $\epsilon_i = 0$ — $i$th observation is on the correct side of margin
- $\epsilon_i > 0$ — $i$th observation is on the wrong side of margin
- $\epsilon_i > 1$ — $i$th observation is on the wrong side of hyperplane
- $C$ — budget for the amount that the margin can be violated by $m$ observations
  - $C = 0$ — No violation, ie. maximal margin classifier
  - $C > 0$ — No more than $C$ observation can be on the wrong side of hyperplane
  - $C$ is small — Narrow margin, highly fit to data, low bias and high variance
  - $C$ is large — Fitting data is less hard, more bias and may have less variance

# Classification with non-linear boundaries





# Classification with non-linear boundaries

- Performance of linear regression can suffer for non-linear data
- Feature space can be enlarged using function of predictors
  - For example, instead of fitting with $x_1, x_2, \ldots, x_n$ features we could use $x_1, x_1^2, x_2, x_2^2 \ldots, x_n, x_n^2$ as features
- Optimization problem becomes

$$\max_{w_0, w_{11}, w_{12} \ldots, w_{n1}, w_{n2}, \epsilon_i, M} \quad M$$

subject to

$$y_i \left( w_0 + \sum_{j=1}^{n} w_{j1} x_{ij} + \sum_{j=1}^{n} w_{j2} x_{ij}^2 \right) \geq M(1 - \epsilon_i) \quad \forall i = 1, \ldots, m$$

$$\sum_{i=1}^{n} \sum_{j=1}^{2} w_{ij}^2 = 1, \quad \sum_{i=1}^{m} \epsilon_i \leq C, \quad \epsilon_i \geq 0$$

# Support Vector Machine

- Extension of support vector classifier that results from enlarging feature space
- It involves inner product of the observations $f(x) = w_0 + \sum_{i=1}^{m} \alpha_i \langle \mathrm{x}, \mathrm{x}_i \rangle$ where $\alpha_i$ - one per training example
  - To estimate $\alpha_i$ and $w_0$, we need $m(m-1)/2$ inner products, $\langle \mathrm{x}_i, \mathrm{x}_{i'} \rangle$
- It turns out that $\alpha_i \neq 0$ for support vectors

  $f(x) = w_0 + \sum_{i \in S} \alpha_i \langle \mathrm{x}, \mathrm{x}_i \rangle$ where $S$ - set of support vectors

# Support Vector Machine

- Inner product is replaced with kernel, $K$ or $K(\mathrm{x}_i, \mathrm{x}_{i'})$
- Kernel quantifies similarity between observations $K(\mathrm{x}_i, \mathrm{x}_{i'}) = \sum_{j=1}^{n} x_{ij} x_{i'j}$
  - Above one is Linear kernel ie. Pearson correlation
- Polynomial kernel $K(\mathrm{x}_i, \mathrm{x}_{i'}) = \left(1 + \sum_{j=1}^{n} x_{ij} x_{i'j}\right)^d$ where $d$ is positive integer $> 1$
- Support vector classifier with non-linear kernel is known as support vector machine and the function will look

$$f(x) = w_0 + \sum_{i \in S} \alpha_i K(\mathrm{x}, \mathrm{x}_i)$$

- Radial kernel: $K(\mathrm{x}_i, \mathrm{x}_{i'}) = \exp\left(-\gamma \sum_{i=1}^{n} (x_{ij} - x_{i'j})^2\right)$ where $\gamma > 0$