University of Nottingham
UK | CHINA | MALAYSIA

COMP3055 Machine Learning

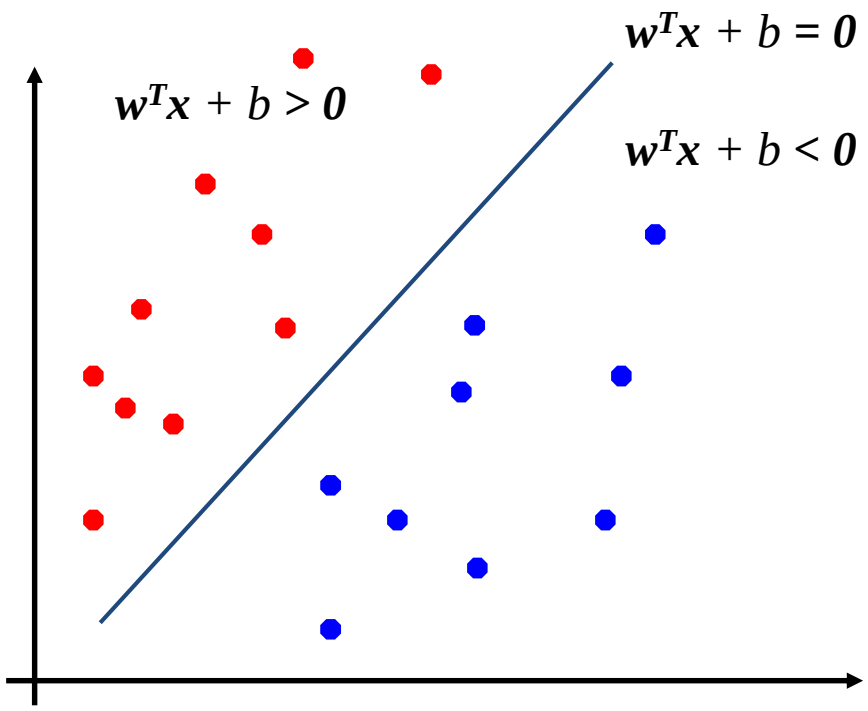
Topic 13 – Support Vector Machine

Zheng Lu

2024 Autumn

Perceptron Revisited: Linear Separators

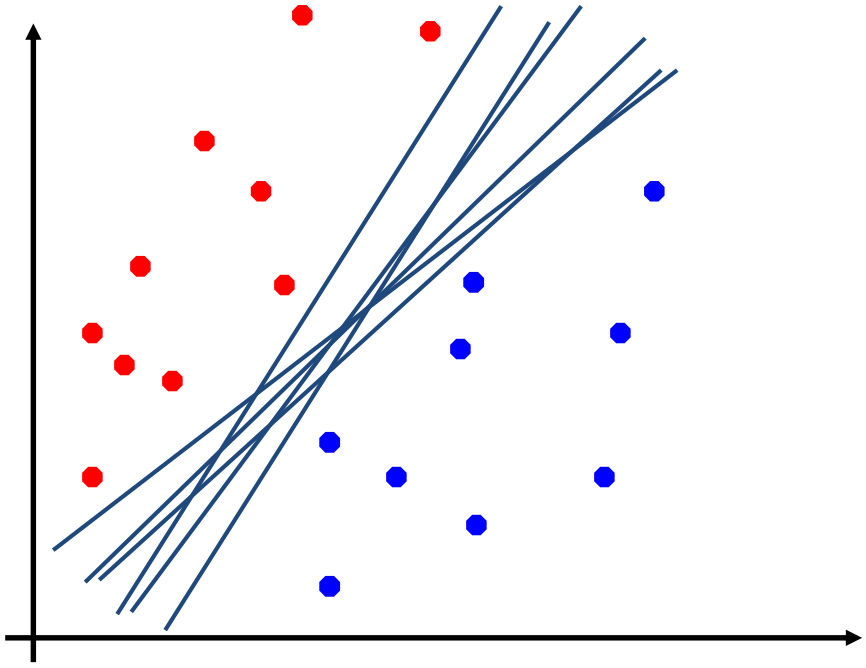
Binary classification can be viewed as the task of separating two classes in feature space:



$$f(x) = sign(w^T x + b)$$

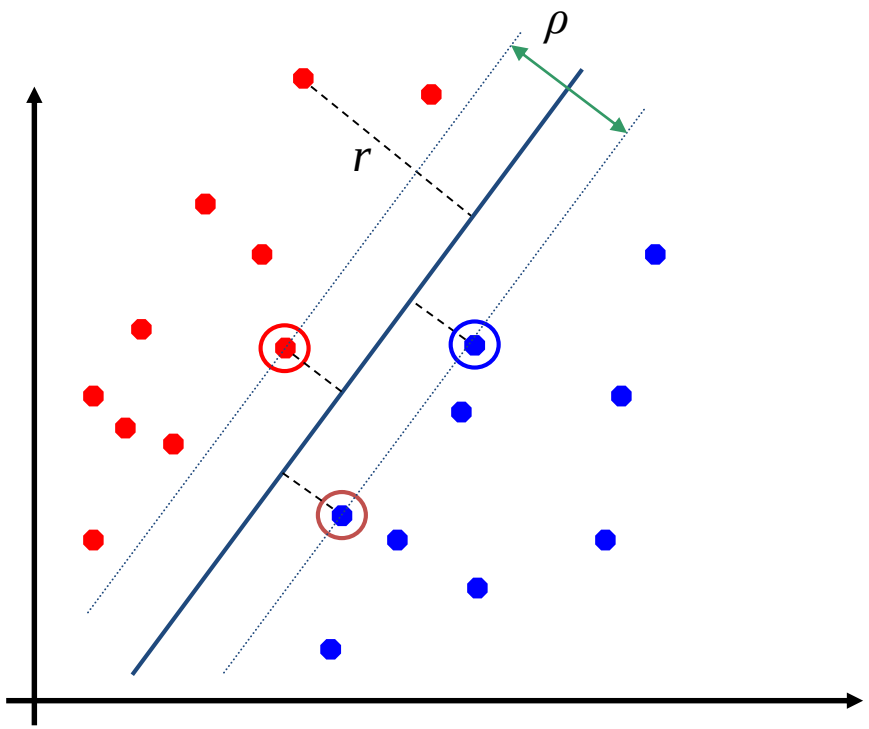
Linear Separator

Which of the linear separators is optimal?



Classification Margin

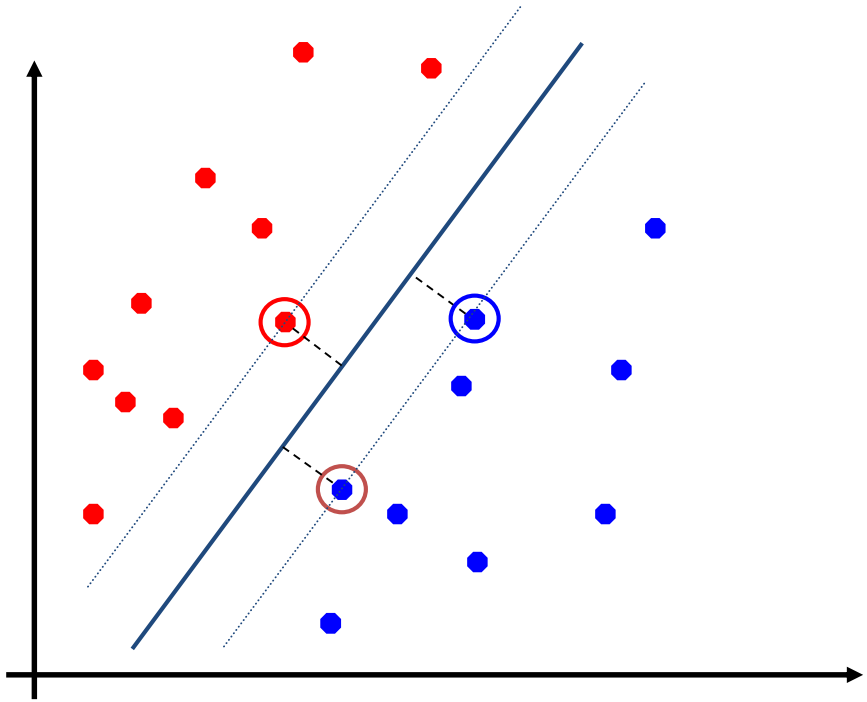
- Distance from example x_s to the separator is $r = \frac{|w^T x_s + b|}{\|w\|}$.
- Examples closest to the hyperplane are ***support vectors***.
- ***Margin*** ρ of the separator is the distance between support vectors.



Maximum Margin Classification

- Maximizing the margin is good according to intuition.

- Implying that only support vectors matter; other training examples are ignorable.



Linear SVM Mathematically

- Let training set $\{(x_i, y_i)\}_{i=1..w}, x_i \in R^d, y_i \in \{-1,1\}$ be separated by a hyperplane with margin *r=2d*. Then for each training example $(\boldsymbol{x}_i, y_i)$:

$$\frac{\boldsymbol{w^T x_i}+b}{\|\boldsymbol{w}\|} \leq -d \quad if\ y_i = -1$$

$$\frac{\boldsymbol{w^T x_i}+b}{\|\boldsymbol{w}\|} \geq d \quad if\ y_i = 1$$

$$\Leftrightarrow \quad y_i\left(\frac{\boldsymbol{w^T x_i}+b}{\|\boldsymbol{w}\|}\right) \geq d$$

- For every support vector $\boldsymbol{x}_s$ the above inequality is an equality. After rescaling, we can obtain

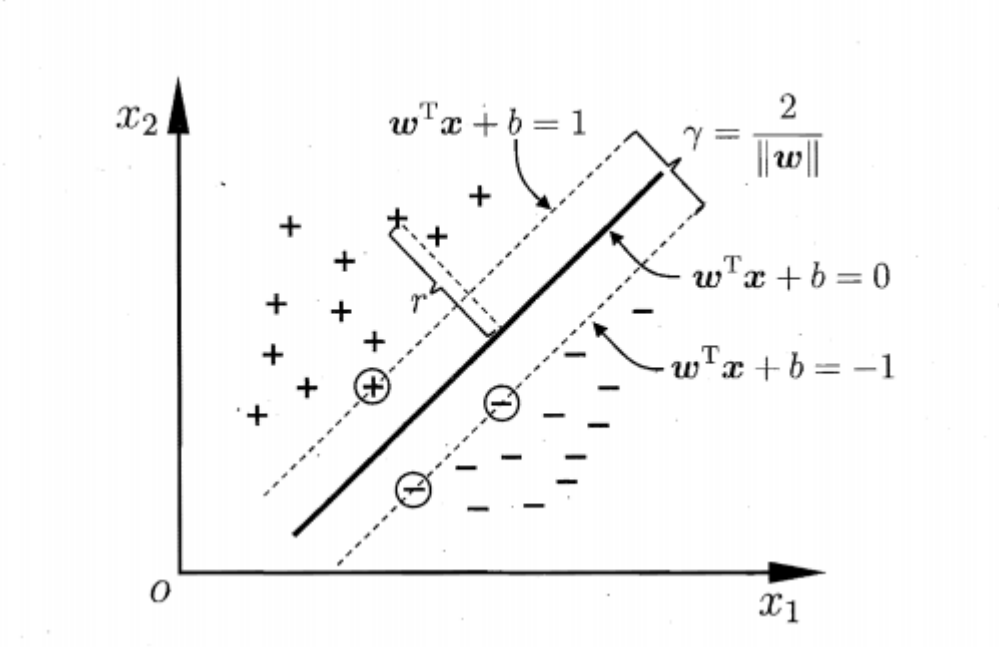
$$y_s(\boldsymbol{w^T x_s} + b) = 1$$

- Alternatively, we have

$$(\boldsymbol{w^T x_s} + b) = 1 \quad if\ y_s = 1$$

$$(\boldsymbol{w^T x_s} + b) = -1 \quad if\ y_s = -1$$

Linear SVM Mathematically



- Then the margin can be expressed as:

$$d = \frac{|\boldsymbol{w}^{\boldsymbol{T}}\boldsymbol{x}_{\boldsymbol{s}} + b|}{\|\boldsymbol{w}\|} = \frac{1}{\|\boldsymbol{w}\|}$$

$$r = 2d = \frac{2}{\|\boldsymbol{w}\|}$$

Linear SVM Mathematically

- Then we can formulate the quadratic optimization problem:

Find $\boldsymbol{w}$ and b such that
$r = \frac{2}{\|\boldsymbol{w}\|}$ is maximized,
and for all $(\boldsymbol{x_i}, y_i), i = 1..n$: $y_i(\boldsymbol{w^T x_i} + b) \geq 1$

Which can be reformulated as:

Find $\boldsymbol{w}$ and b such that
$\boldsymbol{\Phi}(\boldsymbol{w}) = \frac{1}{2}\|\boldsymbol{w}\|^2$ is minimized,
and for all $(\boldsymbol{x_i}, y_i), i = 1..n$: $y_i(\boldsymbol{w^T x_i} + b) \geq 1$

Solving the Optimization Problem

Find $\boldsymbol{w}$ and b such that
$\boldsymbol{\Phi(w)} = \frac{1}{2}\|\boldsymbol{w}\|^2$ minimized,
and for all $(\boldsymbol{x_i}, y_i), i = 1..n$: $y_i(\boldsymbol{w^T x_i} + b) \geq 1$

- Need to optimize a ***quadratic* function subject to *linear* constraints**.
- Quadratic optimization problems are a well-known class of mathematical programming problems for which several (non-trivial) algorithms exist.
- The solution involves constructing a ***dual problem*** where a ***Lagrange multiplier*** α_i is associated with every inequality constraint in the primal (original) problem:

Find α_i such that
$\boldsymbol{L}(w, b, \alpha) = \frac{1}{2}\|\boldsymbol{w}\|^2 - \sum \alpha_i (y_i(\boldsymbol{w^T}.\boldsymbol{x_i} + b) - 1)$ is maximized for all $\alpha_i \geq 0$

The Optimization Problem Solution

- Given a solution $\alpha_1 \ldots \alpha_n$ to the dual problem, solution to the primal is:

$$\boldsymbol{w} = \sum \alpha_i y_i \boldsymbol{x_i}\ , \quad b = y_j - \sum \alpha_i y_i \boldsymbol{x_i}^T \boldsymbol{x_j}, \ \text{ for any } \alpha_j > 0$$

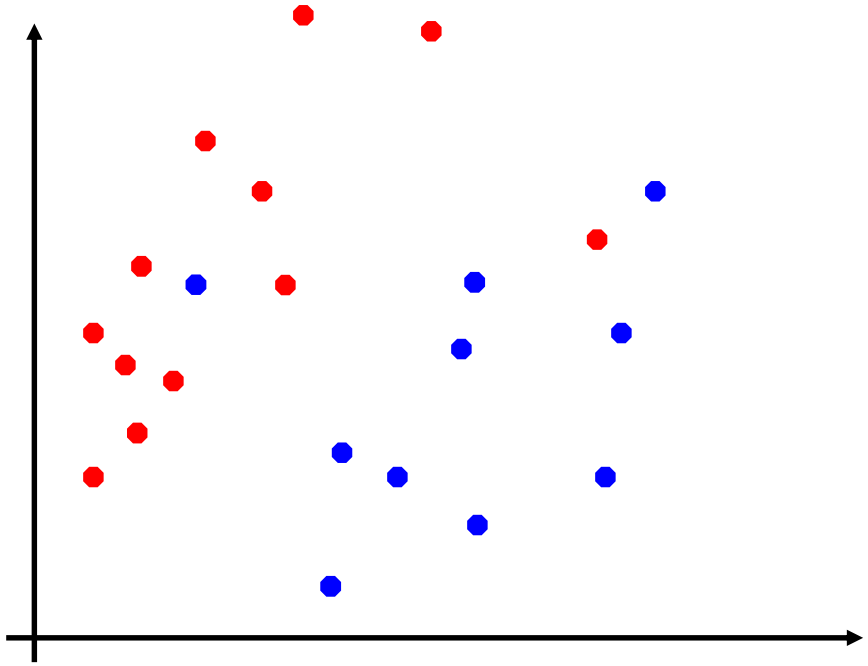
- Each non-zero α_i indicates that corresponding $\boldsymbol{x_i}$ is a support vector.
- Then the classifying function is (note that we don't need $\boldsymbol{w}$ explicitly):

$$f(x) = \sum \alpha_i y_i \boldsymbol{x_i}^T \boldsymbol{x} + b$$

- Notice that it relies on an *inner product* between the test point $\boldsymbol{x}$ and the support vectors $\boldsymbol{x}_i$ – we will return to this later.

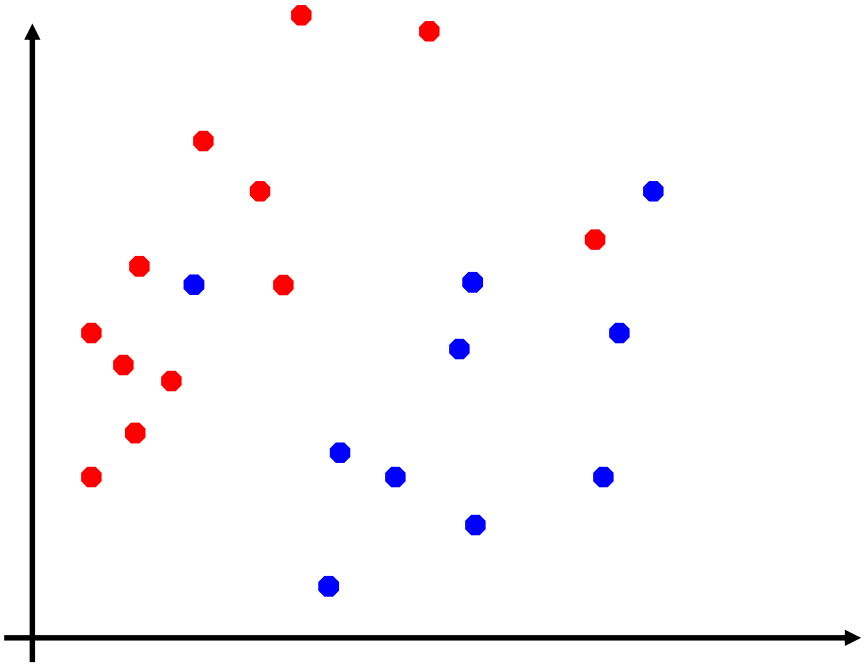
Soft Margin Classification

- What if the training set is not linearly separable?



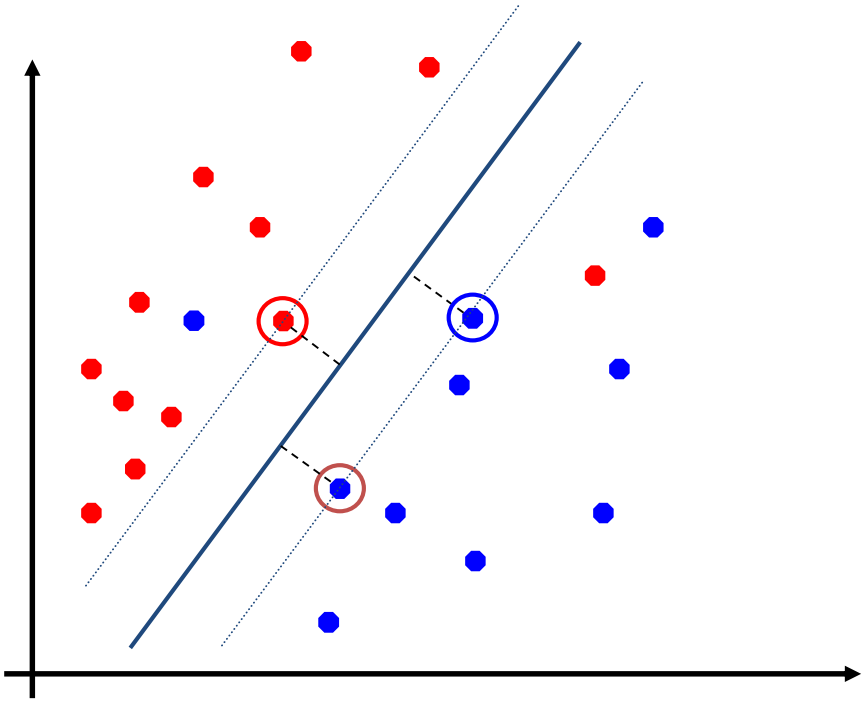
Soft Margin Classification

- What if the training set is not linearly separable?
- ***Slack variables*** ξ_i can be added to allow misclassification of difficult or noisy examples, resulting margin called ***soft***.



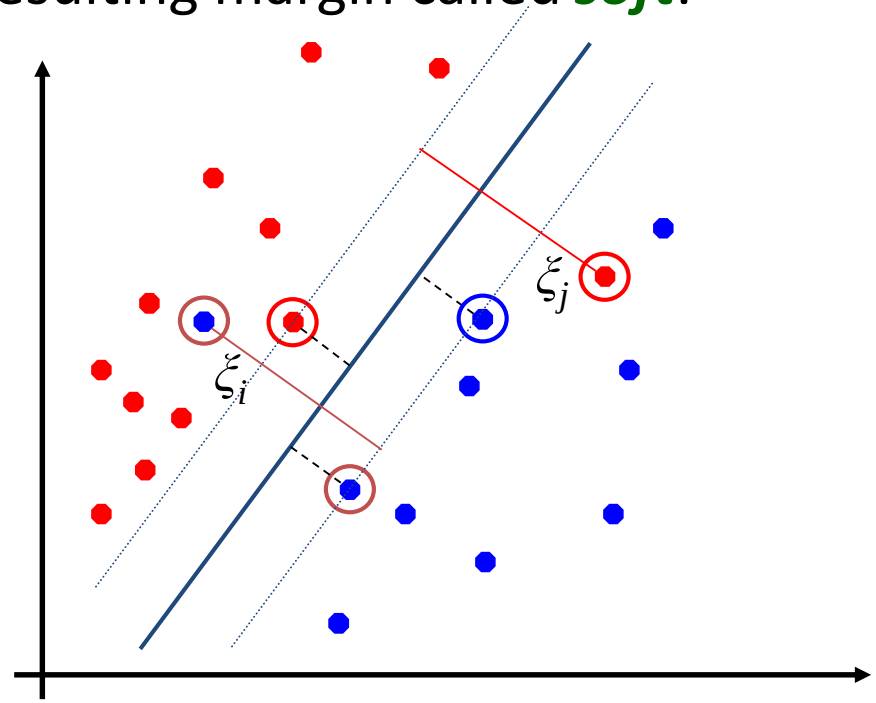
Soft Margin Classification

- What if the training set is not linearly separable?
- ***Slack variables*** ξ_i can be added to allow misclassification of difficult or noisy examples, resulting margin called ***soft***.



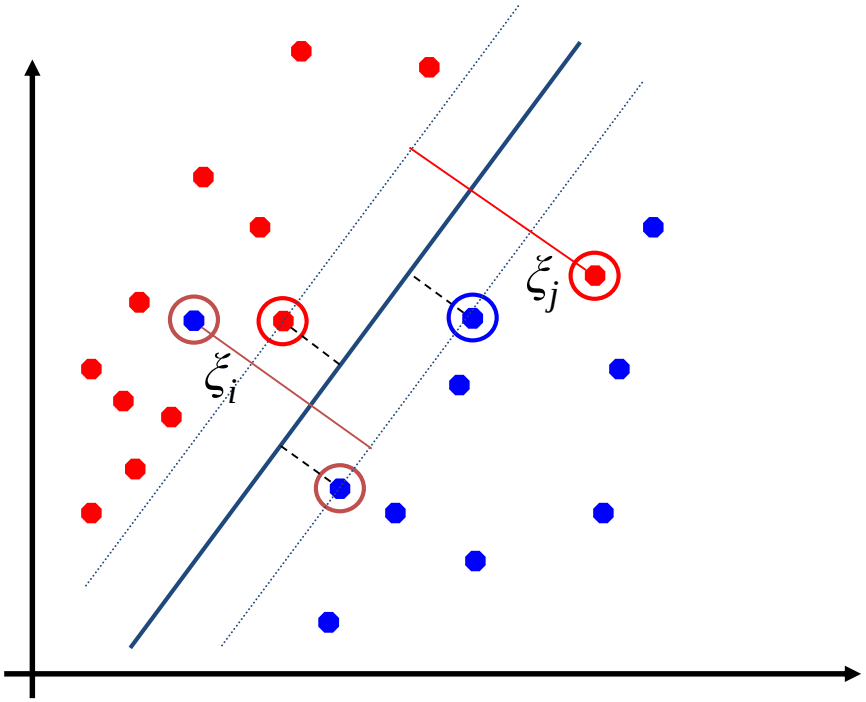
Soft Margin Classification

- What if the training set is not linearly separable?
- ***Slack variables*** ξ_i which measures the distance of the point to its marginal hyperplane if it is on the wrong side, otherwise 0, can be added to allow misclassification of difficult or noisy examples, resulting margin called ***soft***.



Soft Margin Classification

- Applying Soft Margin, SVM tolerates a few dots to get misclassified and tries to balance the trade-off between finding a line that maximizes the margin and minimizes the misclassification.



Soft Margin Classification Mathematically

- The old formulation:

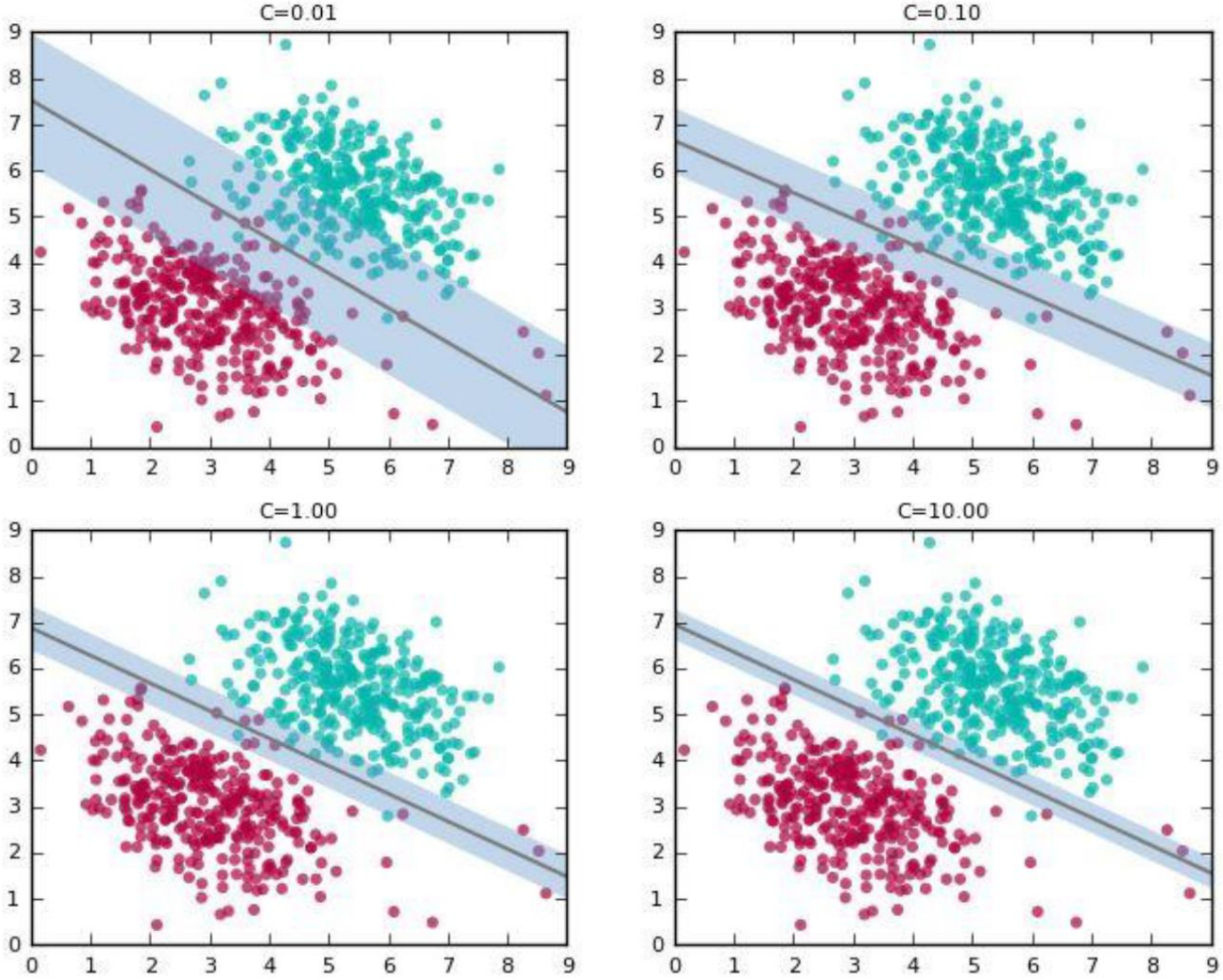
Find $\boldsymbol{w}$ and b such that
$\boldsymbol{\Phi}(\boldsymbol{w}) = \frac{1}{2}\|\boldsymbol{w}\|^2$ is minimized,
and for all $(\boldsymbol{x_i}, y_i), i = 1..n$: $y_i(\boldsymbol{w^T x_i} + b) \geq 1$

- Modified formulation incorporates slack variables:

Find $\boldsymbol{w}$ and b such that
$\boldsymbol{\Phi}(\boldsymbol{w}) = \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum \xi_i$ is minimized,
and for all $(\boldsymbol{x_i}, y_i), i = 1..n$: $y_i\left(\boldsymbol{w^T x_i} + b\right) \geq 1 - \xi_i, \quad \xi_i > 0$

- Parameter C can be viewed as a way to control overfitting: it "trades off" the relative importance of maximizing the margin and fitting the training data.

SVM Parameter Tuning



Soft Margin Classification Solution

- Dual problem:

Find α_i such that
$\boldsymbol{L}(w, b, \alpha) = \frac{1}{2}\|\boldsymbol{w}\|^2 - \sum \alpha_i (y_i(\boldsymbol{w^T}.\boldsymbol{x_i} + b) - \xi_i), \xi_i > 0$ is maximized for all $0 \leq \alpha_i \leq C$

- Again, $\boldsymbol{x}_i$ with non-zero α_i will be support vectors.
- Solution to the dual problem is:

$$\boldsymbol{w} = \sum \alpha_i y_i \boldsymbol{x_i}\ , \quad b = y_j(1 - \xi_j) - \sum \alpha_i y_i \boldsymbol{x_i^T x_j}, \text{ for any } \alpha_j > 0$$

- Again, we do not need to compute $\boldsymbol{w}$ explicitly for classification:

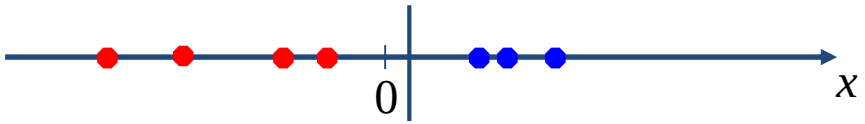
$$f(x) = \sum \alpha_i y_i \boldsymbol{x_i^T x} + b$$

Linear SVM - Overview

- The classifier is a ***separating hyperplane***.
- Most "important" training points are support vectors; they define the hyperplane.
- Quadratic optimization algorithms can identify which training points $\boldsymbol{x}_i$ are support vectors with non-zero Lagrangian multipliers α_i.

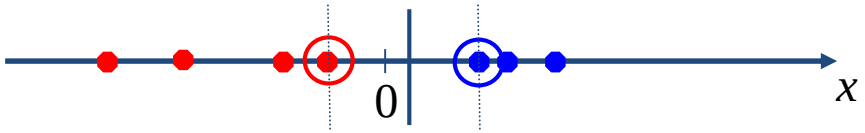
Non-Linear SVM

- Datasets that are linearly separable with some noise work out great:



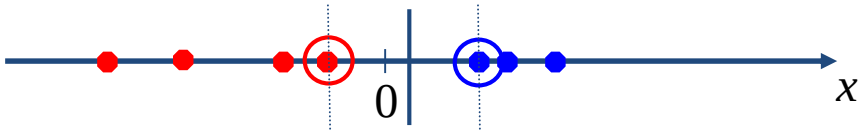
Non-Linear SVM

- Datasets that are linearly separable with some noise work out great:



Non-Linear SVM

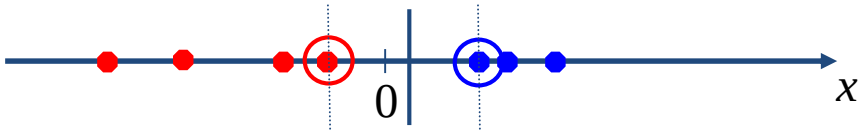
- Datasets that are linearly separable with some noise work out great:



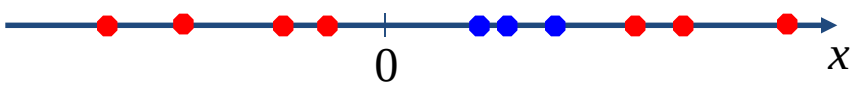
- But what are we going to do if the dataset is just too hard?

Non-Linear SVM

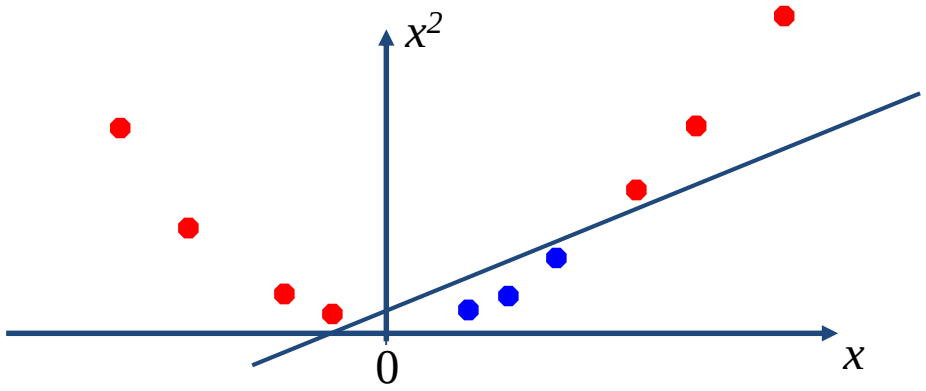
- Datasets that are linearly separable with some noise work out great:



- But what are we going to do if the dataset is just too hard?

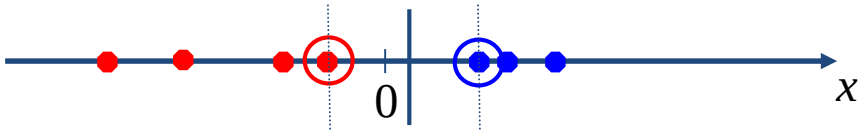


- How about… mapping data to a higher-dimensional space:

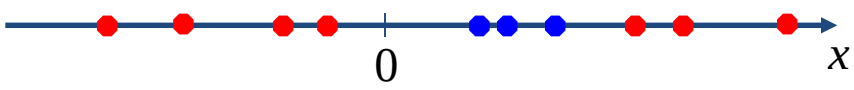


Non-Linear SVM

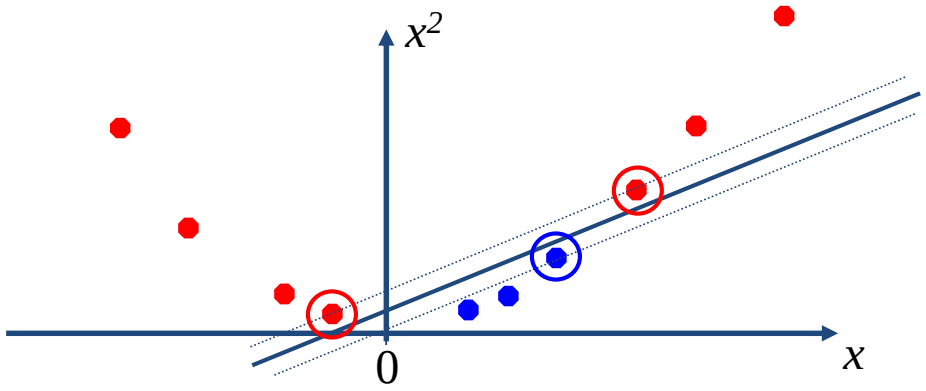
- Datasets that are linearly separable with some noise work out great:



- But what are we going to do if the dataset is just too hard?

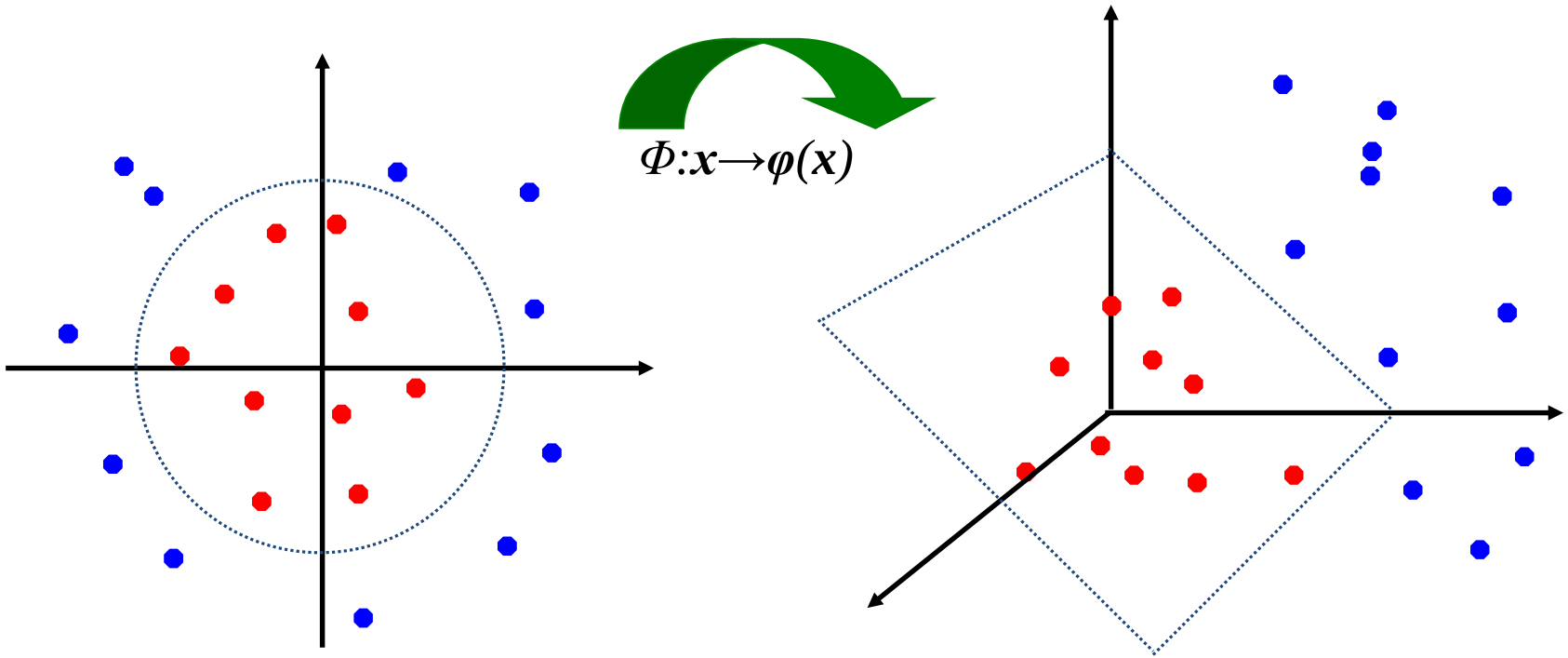


- How about… mapping data to a higher-dimensional space:



Non-Linear SVM: Feature Spaces

- **General idea**: the original feature space can always be mapped to some higher-dimensional feature space where the training set is separable:



The "Kernel Trick"

- The linear classifier relies on inner product between vectors $K(\boldsymbol{x}_i,\boldsymbol{x}_j)=\boldsymbol{x}_i^T\boldsymbol{x}_j$
- If every data point is mapped into high-dimensional space via some transformation $\Phi:\mathbf{x}\rightarrow\varphi(\mathbf{x})$, the inner product becomes:

$$K(\boldsymbol{x}_i,\boldsymbol{x}_j)= \boldsymbol{\varphi}(\boldsymbol{x}_i)^T\boldsymbol{\varphi}(\boldsymbol{x}_j)$$

- A ***kernel function*** is a function that is equivalent to an inner product in some feature space.
- Thus, a kernel function ***implicitly*** maps data to a high-dimensional space (without the need to compute each $\boldsymbol{\varphi}(\boldsymbol{x})$ explicitly).

Kernel Functions

- What functions are kernel functions?
  - For some functions $K(\boldsymbol{x}_i,\boldsymbol{x}_j)$ checking that $K(\boldsymbol{x}_i,\boldsymbol{x}_j)= \varphi(\boldsymbol{x}_i)^{\boldsymbol{T}}\varphi(\boldsymbol{x}_j)$ can be cumbersome.

Kernel Functions

- What functions are kernel functions?
  - For some functions $K(\boldsymbol{x}_i,\boldsymbol{x}_j)$ checking that $K(\boldsymbol{x}_i,\boldsymbol{x}_j)= \varphi(\boldsymbol{x}_i)^T\varphi(\boldsymbol{x}_j)$ can be cumbersome.
- Can we use any function?
  - No! A function $K(\boldsymbol{x}_i,\boldsymbol{x}_j)$ is a valid kernel if it corresponds to an inner product in some (perhaps infinite dimensional) feature space.
- Mercer's theorem
  - ***Every semi-positive definite symmetric function is a kernel.***

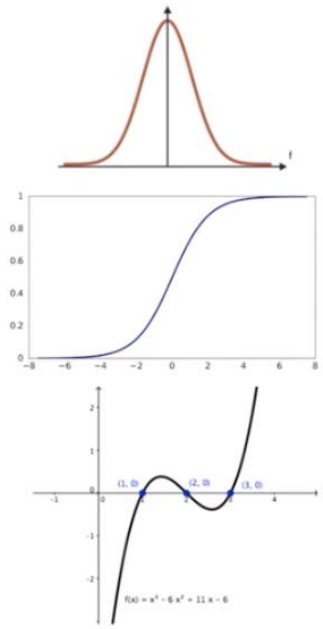
Examples of Kernel Function

$$K(x_i, x_j) = (x_i \cdot x_j + 1)^p; \text{ polynomial kernel.} \quad (1.22)$$

$$K(x_i, x_j) = e^{\frac{-1}{2\sigma^2}(x_i - x_j)^2}; \text{ Gaussian kernel; Special case of Radial Basis Function.}$$

$$K(x_i, x_j) = e^{-\gamma(x_i - x_j)^2}; \text{ RBF Kernel}$$

$$K(x_i, x_j) = \tanh(\eta\, x_i \cdot x_j + \nu); \text{ Sigmoid Kernel; Activation function for NN.}$$



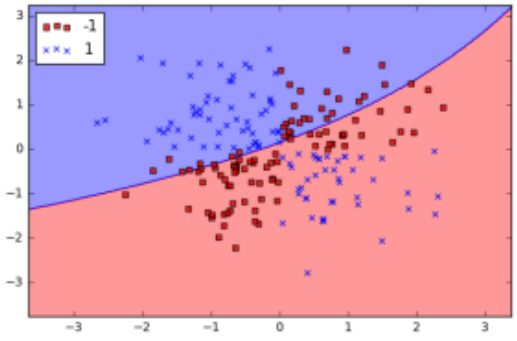
Gaussian RBF Kernel $K(\vec{x}, \vec{l}^i) = e^{-\frac{\|\vec{x} - \vec{l}^i\|^2}{2\sigma^2}}$

Sigmoid Kernel $K(X, Y) = \tanh(\gamma \cdot X^T Y + r)$

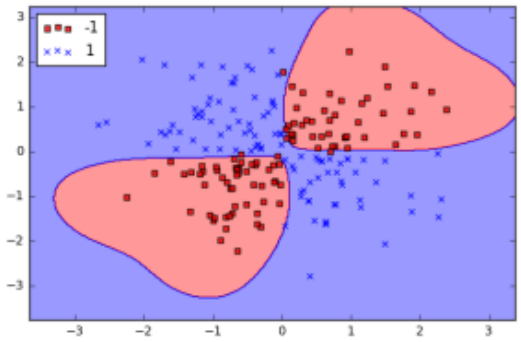
Polynomial Kernel $K(X, Y) = (\gamma \cdot X^T Y + r)^d, \gamma > 0$

SVM Parameter Tuning

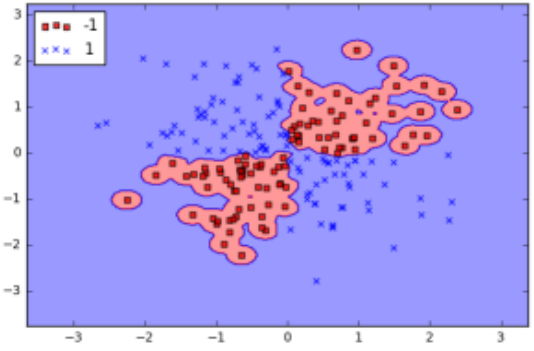
- The gamma parameter is the inverse of the standard deviation of the RBF kernel (Gaussian function), which is used as similarity measure between two points.
- Small gamma, large variance and vice versa.
- Large may cause complicated decision boundary or give rise to over-fitting.



$\gamma = 0.01$



$\gamma = 1$



$\gamma = 100$

Non-Linear SVM Mathematically

- Dual problem formulation:

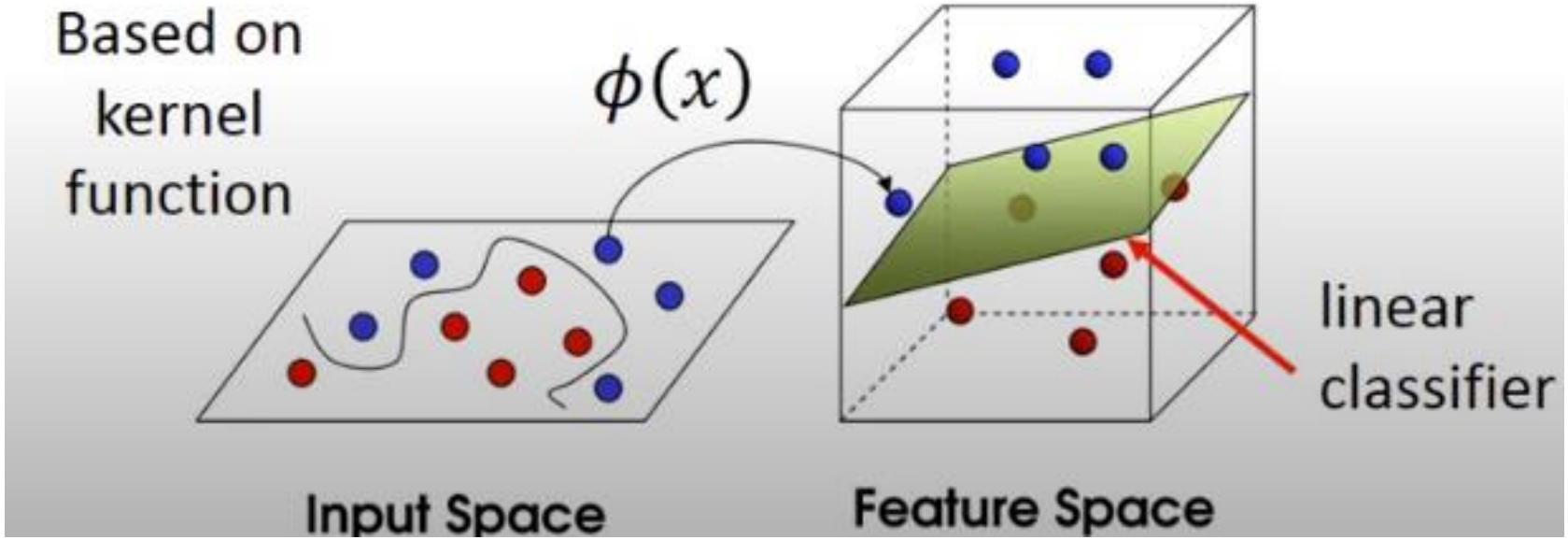
Find α_i such that
$\boldsymbol{L}(w, b, \alpha) = \frac{1}{2}\|\boldsymbol{w}\|^2 - \sum \alpha_i (y_i(\boldsymbol{w^T}.\varphi(\boldsymbol{x_i}) + b) - 1)$ is maximized for all $\alpha_i \geq 0$

- The solution is:

$$f(x) = \sum \alpha_i y_i K(\boldsymbol{x_i}, \boldsymbol{x}) + b$$

- Optimization for finding α_i remains the same!

SVM and Kernel Methods



- Mapping data points from low dimensional space to a higher dimensional space can make it possible to apply SVM even for non-linear data sample.
- We don't need to know the mapping function itself, as long as we know the Kernel function (***Kernel Trick)***
- How the tuning parameter gamma can lead to over fitting or bias in RBF kernel.

Multi-Class Classification

- Some algorithms are designed for binary classification problems:
  - Logistic Regression
  - Perceptron
  - Support Vector Machines
- Instead, heuristic methods can be used to split a multi-class classification problem into multiple binary classification datasets and train a binary classification model each
  - One-vs-All (OVA)
  - One-vs-One (OvO)

One-vs-All (OVA)

- Every class is paired with the remaining classes
- The base classifier needs to produce a real-valued confidence score for its decision, rather than just a class label
- Making decisions means applying all classifiers to an unseen sample *x* and predicting the label *k* for which the corresponding classifier reports the highest confidence score

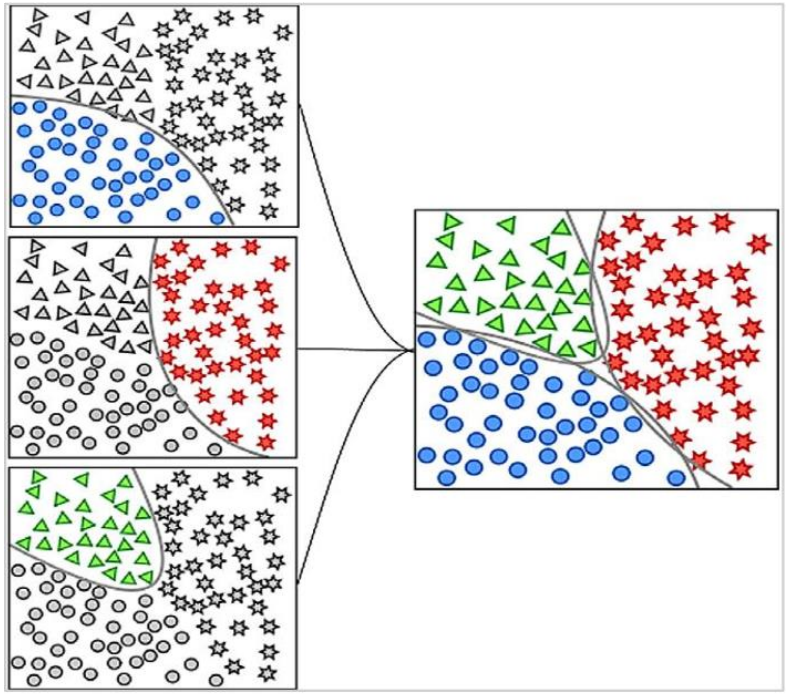


Figure 3. One vs all Strategy using 3 class problem

One-vs-One (OVO)

- Every class is paired with the every other class
- At prediction time, a voting scheme is applied: all classifiers are applied to an unseen sample and the class that got the highest number of "+1" predictions gets predicted by the combined classifier

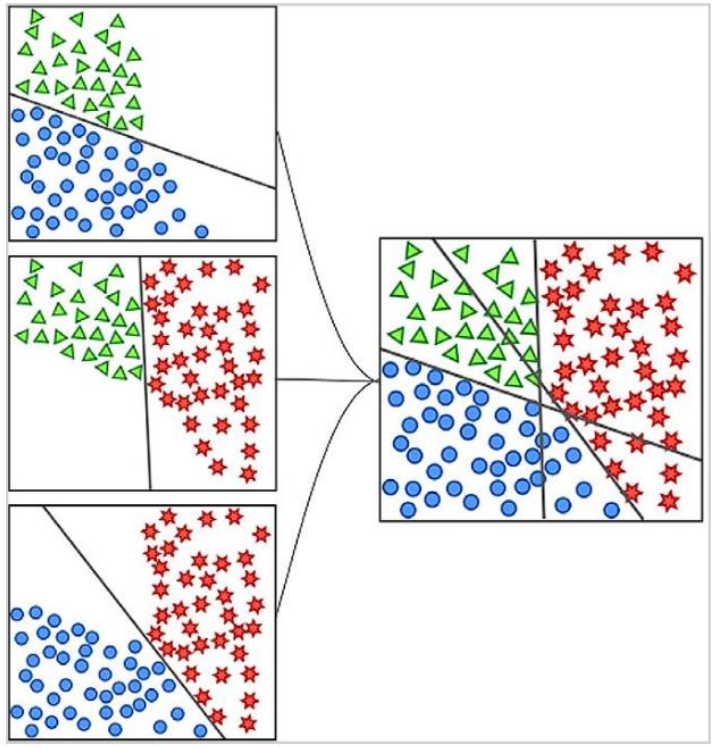


Figure 2. One vs one strategy using 3 class problem

SVM Software and Resourses

- http://www.svms.org/tutorials/
- http://www.csie.ntu.edu.tw/~cjlin/libsvm/
  - LIBSVM -- A Library for Support Vector Machines by Chih-Chung Chang and Chih-Jen Lin