Statistical Methods in Artificial Intelligence
CSE471 - Monsoon 2016 : Lecture 17

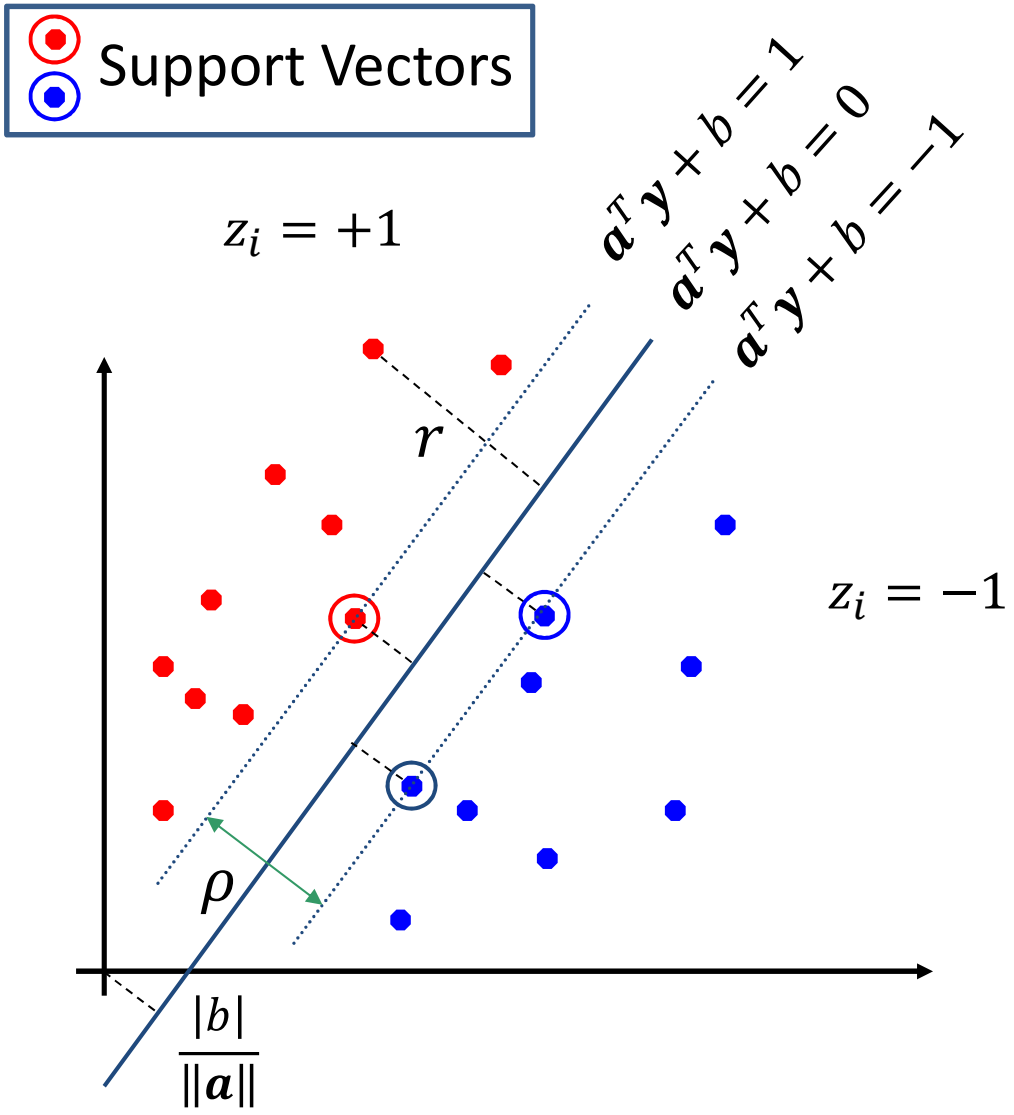
Avinash Sharma

CVIT, IIIT Hyderabad

Lecture Plan

- Revision from Previous Lecture
- Transductive SVM
- Multi-class SVM
- Kernel SVM
- SVM Example
- Kernel Methods, Intro to Clustering (Next Class)

Maximum Margin Classification



For points on boundary

$$r_i = \frac{\boldsymbol{a}^T\boldsymbol{y}_i + b}{\|\boldsymbol{a}\|}, z_i = 1$$

$$r_j = \frac{\boldsymbol{a}^T\boldsymbol{y}_j + b}{\|\boldsymbol{a}\|}, z_j = -1$$

$$z_k(\boldsymbol{a}^T\boldsymbol{y}_k + b) = 1$$

$$\rho = r_i - r_j = \frac{2}{\|\boldsymbol{a}\|}$$

Let $|b|\|\boldsymbol{a}\| = constant$

Linear Support Vector Machine

- Primal Formulation:

  Maximize the margin

$$\arg\max_{\boldsymbol{a}} \left(\frac{2}{\|\boldsymbol{a}\|} \right)$$

  such that $z_i(\boldsymbol{a}^T \boldsymbol{y}_i + b) \geq 1 \quad \forall i \in \{1, \ldots, n\}$

  Or,

$$\arg\min_{\boldsymbol{a},b}(\|\boldsymbol{a}\|^2) = \arg\min_{\boldsymbol{a}} \left(\frac{1}{2} \boldsymbol{a}^T \boldsymbol{a} \right)$$

  such that $z_i(\boldsymbol{a}^T \boldsymbol{y}_i + b) \geq 1 \quad \forall i \in \{1, \ldots, n\}$

Linear Support Vector Machine

- Dual Formulation:

$$\arg \min_{\boldsymbol{a},b} \max_{\alpha_1,\ldots,\alpha_n} \left\{ \frac{1}{2}\boldsymbol{a}^T\boldsymbol{a} - \sum_{i=1}^{n} \alpha_i (z_i(\boldsymbol{a}^T\boldsymbol{y}_i + b) - 1) \right\}$$

such $\alpha_i \geq 0 \quad \forall i \in \{1, \ldots, n\}$

Or,

$$\arg \max_{\alpha_1,\ldots,\alpha_n} \sum_{k=1}^{n} \alpha_k - \frac{1}{2} \sum_{k=1,j=1}^{n} \alpha_k \alpha_j z_k z_j \boldsymbol{y}_k{}^T \boldsymbol{y}_j$$

such that $\sum_{k=1}^{n} \alpha_k z_k = 0$ and $\alpha_k \geq 0 \quad \forall k \in \{1, \ldots, n\}$

Linear Support Vector Machine

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

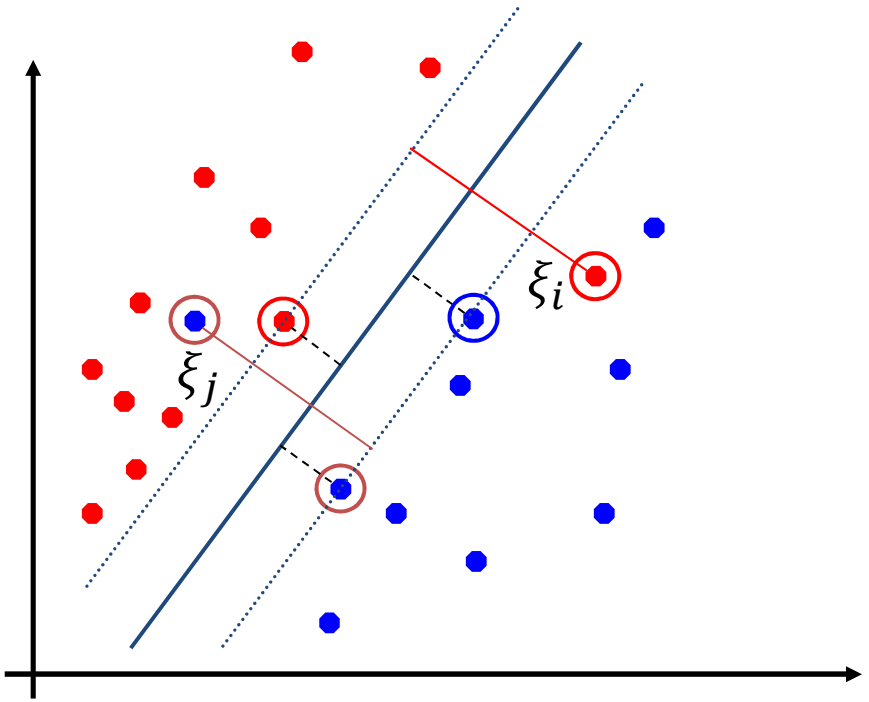
$$\boldsymbol{a} = \sum_{j=1}^{n} \alpha_j z_j \boldsymbol{y}_j \text{ and } b_k = z_k - \sum_{j=1}^{n} \alpha_j z_j \boldsymbol{y}_j{}^T \boldsymbol{y}_k \text{ for } \forall \alpha_k > 0$$

$$b = mean([b_1, \ldots, b_k, \ldots, b_m])$$

- Each non-zero α_k indicates that corresponding $\boldsymbol{y}_k$ is a support vector.
- The classifying function is:

$$f(\boldsymbol{y}) = \sum_{j=1}^{n} \alpha_j z_j \boxed{\boldsymbol{y}_j{}^T \boldsymbol{y}} + b$$

Soft Margin SVM



Let $\xi_i \geq 0 \ \forall i$

$$z_i(\boldsymbol{a}^T \boldsymbol{y}_i + b) \geq 1 - \xi_i$$

Soft Margin SVM

- Primal Formulation:

$$\arg\min_{\boldsymbol{a},\xi,b}\left(\frac{1}{2}\boldsymbol{a}^T\boldsymbol{a} + C\sum_{i=1}^{n}\xi_i\right)$$

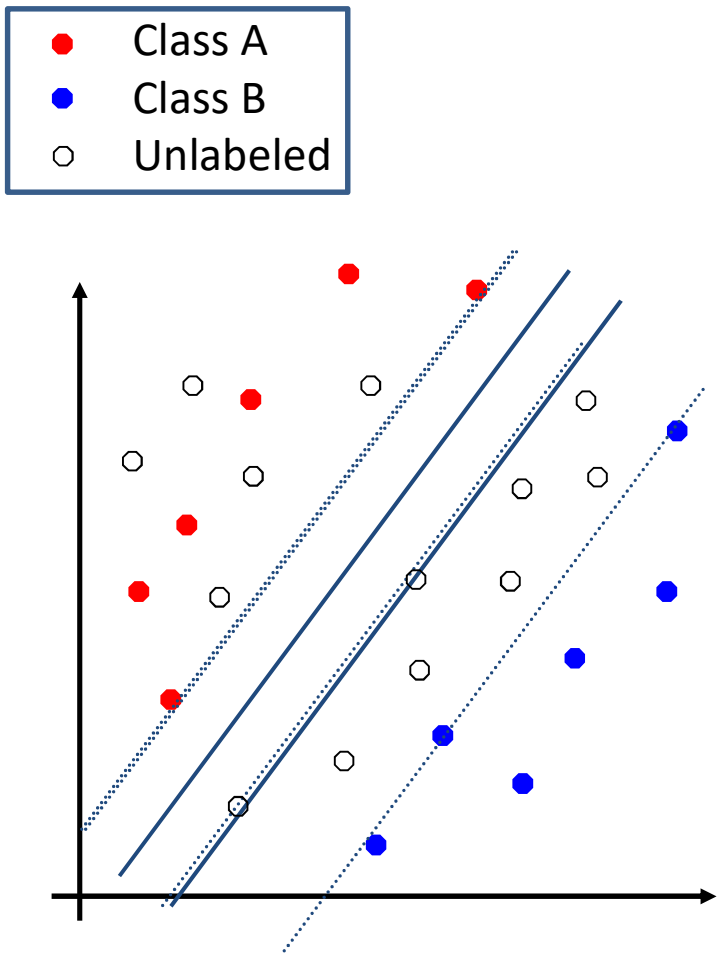
such that $z_i(\boldsymbol{a}^T\boldsymbol{y}_i + b) \geq 1 - \xi_i$ and $\xi_i \geq 0 \quad \forall i \in \{1, \dots, n\}$

- Dual Formulation:

$$\arg\max_{\alpha_1,\dots,\alpha_n}\sum_{k=1}^{n}\alpha_k - \frac{1}{2}\sum_{k=1,j=1}^{n}\alpha_k\alpha_j z_k z_j \boldsymbol{y}_k{}^T\boldsymbol{y}_j$$

such that $\sum_{k=1}^{n}\alpha_k z_k = 0$ and $C \geq \alpha_k \geq 0 \quad \forall k \in \{1, \dots, n\}$

Transductive SVM



- In a semi-supervised setup, transductive SVM consider both labeled and unlabeled data points while learning the maximum margin classifier.
- The idea is to move the decision boundary in the region of low local density.
- The tentative labels for unlabeled data points are inferred and then classifier parameters are estimated, in an iterative manner.

Transductive SVM

$$\arg \min_{z_{n+1},\ldots,z_m} \arg \min_{\boldsymbol{a},\xi,\eta,b} \left(\frac{1}{2}\boldsymbol{a}^T\boldsymbol{a} + C\sum_{i=1}^{n}\xi_i + D\sum_{i=n+1}^{m}\eta_i \right)$$

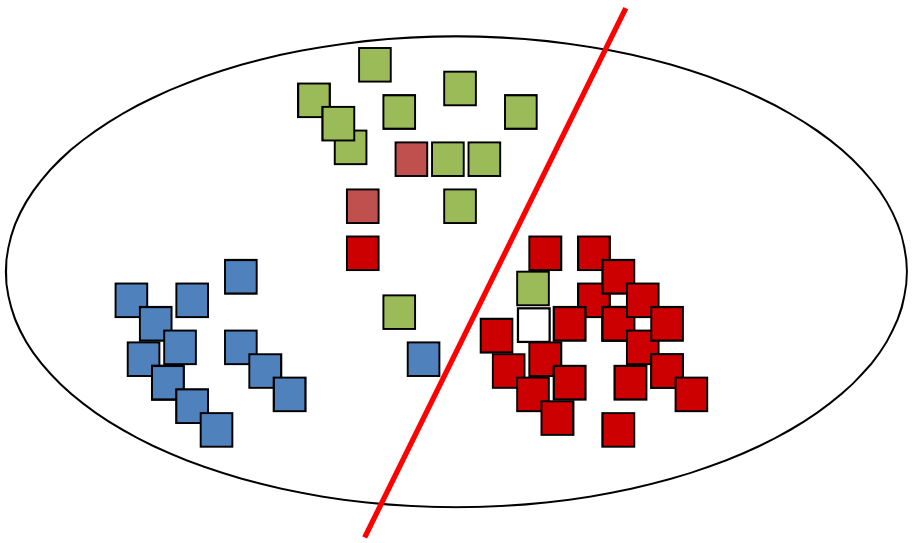
such that $z_i(\boldsymbol{a}^T\boldsymbol{y}_i + b) \geq 1 - \xi_i \;\;\&\;\; \xi_i \geq 0 \quad \forall i \in \{1, \ldots, n\},$

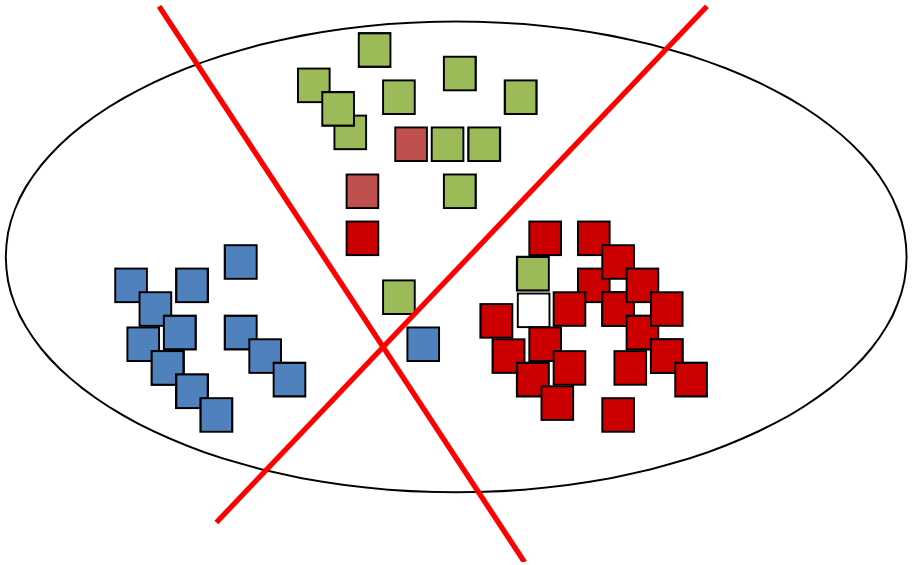
$z_i(\boldsymbol{a}^T\boldsymbol{y}_i + b) \geq 1 - \eta_i \;\;\&\;\; \eta_i \geq 0 \quad \forall i \in \{n+1, \ldots, m\},$

- Do Iteratively:
- Step 1: fix $z_{n+1}, \ldots, z_m$, learn weight vector $\boldsymbol{a}$
- Step 2: fix weight vector $\boldsymbol{a}$, try to predict $z_{n+1}, \ldots, z_m$

Multi-category SVM

- SVM is a binary classifier.
- Two natural multi-class extensions are:
  - One Class v/s All : Learns C classifiers
  - One Class v/s One Class : Learns C*(C-1) Classifiers





Multi-category SVM

- Kesler Construction

$$\hat{\mathbf{a}}_i^t \mathbf{y}_k - \hat{\mathbf{a}}_j^t \mathbf{y}_k > 0, \quad j = 2, ..., c.$$

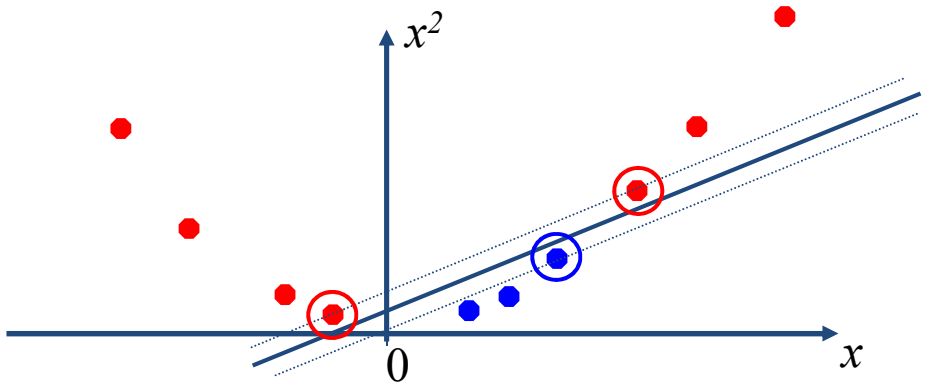
$$\hat{\boldsymbol{\alpha}} = \begin{bmatrix} \mathbf{a}_1 \\ \mathbf{a}_2 \\ \vdots \\ \mathbf{a}_c \end{bmatrix} \qquad \boldsymbol{\eta}_{12} = \begin{bmatrix} \mathbf{y} \\ -\mathbf{y} \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \end{bmatrix}, \quad \boldsymbol{\eta}_{13} = \begin{bmatrix} \mathbf{y} \\ \mathbf{0} \\ -\mathbf{y} \\ \vdots \\ \mathbf{0} \end{bmatrix}, \quad \cdots, \quad \boldsymbol{\eta}_{1c} = \begin{bmatrix} \mathbf{y} \\ \mathbf{0} \\ \mathbf{0} \\ \vdots \\ -\mathbf{y} \end{bmatrix}.$$

- Choose i if $\hat{\boldsymbol{\alpha}}^t \boldsymbol{\eta}_{ij} > 0$ for $j \neq i$,

Non-linear SVM

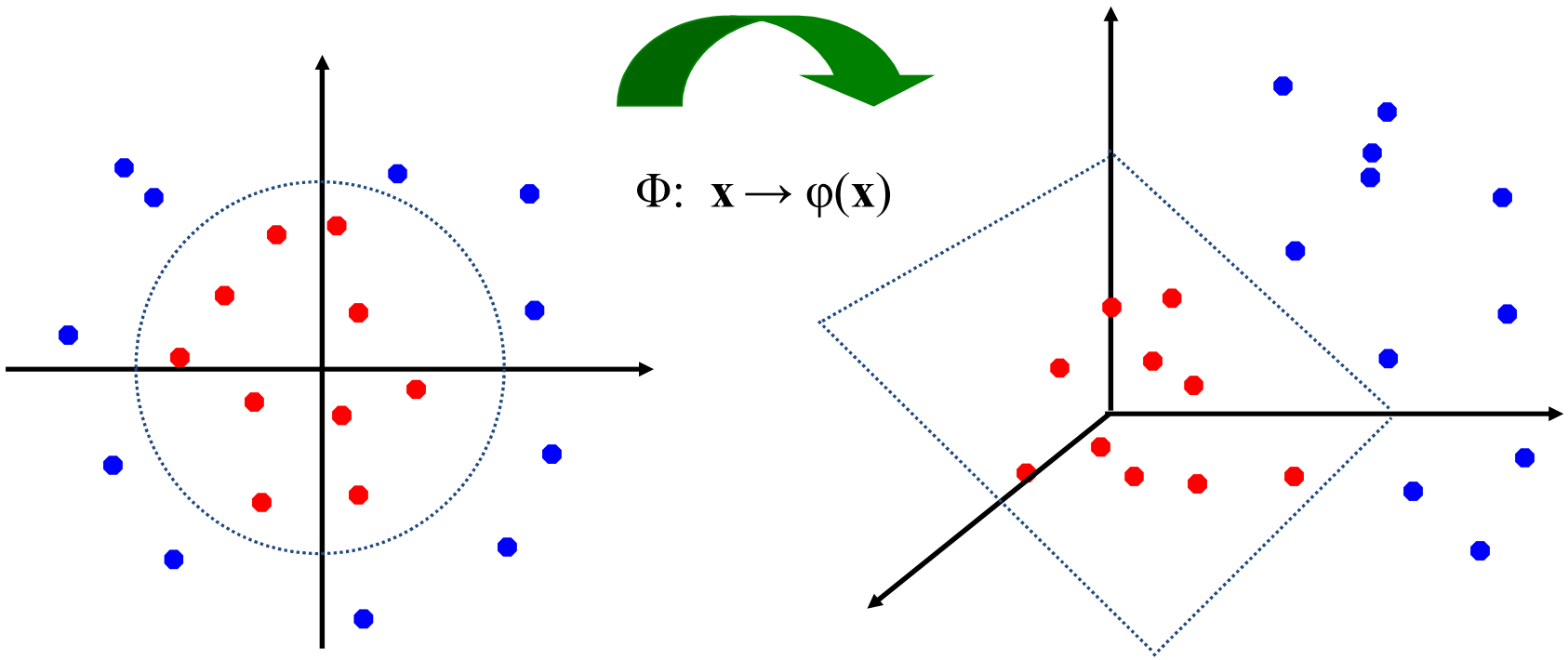
- Non-linear classification scenario

- Project data into higher dimensional space



Non-linear SVM

Linear Classification in Non-linear Space



Non-linear SVM

- Linear SVM

$$\arg\max_{\alpha_1,\ldots,\alpha_n} \sum_{k=1}^{n} \alpha_k - \sum_{k=1,j=1}^{n} \alpha_k \alpha_j z_k z_j \boldsymbol{y}_k{}^T \boldsymbol{y}_j$$

- Non-linear SVM

$$\arg\max_{\alpha_1,\ldots,\alpha_n} \sum_{k=1}^{n} \alpha_k - \sum_{k=1,j=1}^{n} \alpha_k \alpha_j z_k z_j \varphi(\boldsymbol{y}_k)^T \varphi(\boldsymbol{y}_j)$$

$$f(\boldsymbol{y}) = \sum_{j=1}^{n} \alpha_j z_j \, \varphi(\boldsymbol{y}_j)^T \, \varphi(\boldsymbol{y}) \; + b$$

Kernelization

- **Kernels** are functions that return inner products between the images of data points in some space.

$$K(k,j) = \varphi(\boldsymbol{y}_k)^T \varphi(\boldsymbol{y}_j) = < \varphi(\boldsymbol{y}_k), \varphi(\boldsymbol{y}_j) >$$

- K is $n \times n$ square matrix known as Kernel or Gram matrix.
- K is always a symmetric & positive semi-definite matrix (from Mercer's Theorem).
- Or, any symmetric & positive semi-definite matrix can be interpreted as kernel matrix.
- From symmetricity: $K(k,j) = K(j,k)$
- By combining a simple linear discriminant algorithm with this simple Kernel, we can efficiently learn nonlinear separations.
- Any Kernel (PSD) matrix can have both positive and negative entries.

Kernelization

- Commonly used Kernel functions are:
  - Linear Kernel

$$K(k,j) = \boldsymbol{y}_k{}^T \boldsymbol{y}_j$$

  - Polynomial Kernel

$$K(k,j) = (1 + \boldsymbol{y}_k{}^T \boldsymbol{y}_j)^p$$

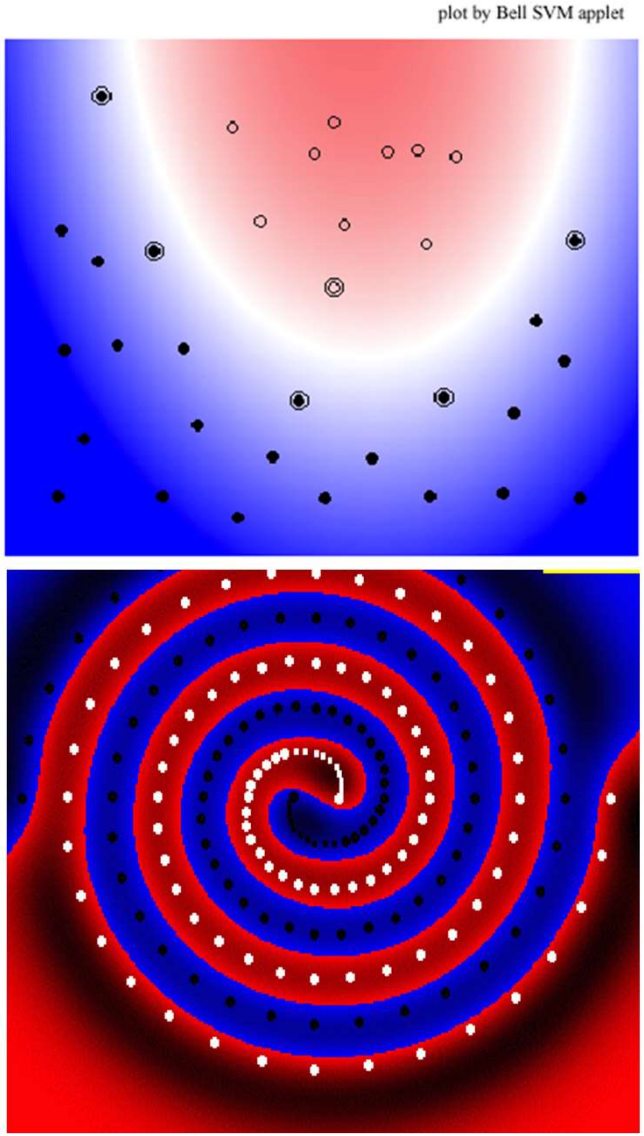
  - Gaussian /Radial Basis Function (RBF) Kernel

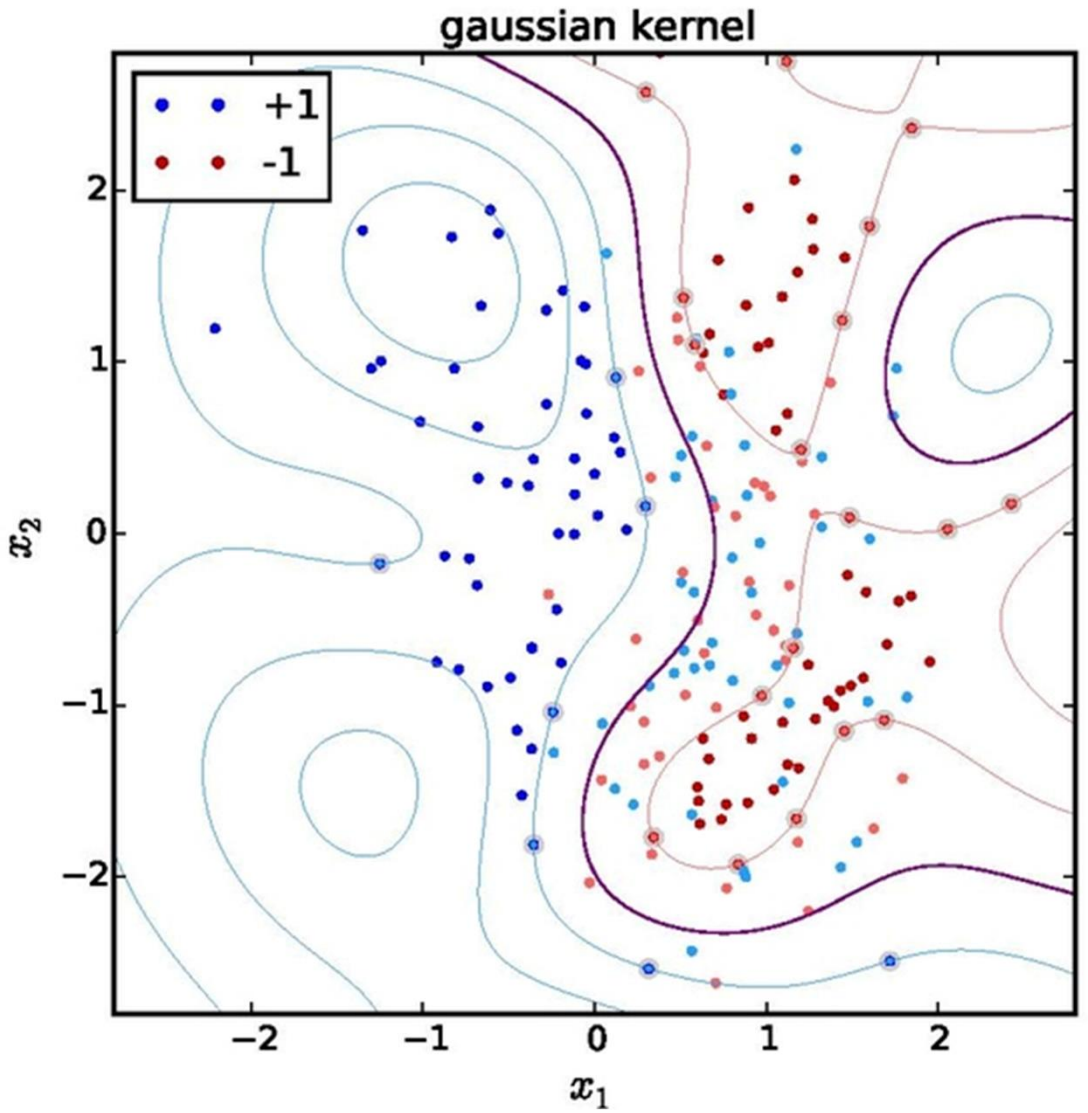
$$K(k,j) = \exp\left(-\frac{\|\boldsymbol{y}_k - \boldsymbol{y}_j\|^2}{2\sigma^2}\right) = \exp\left(-\gamma\|\boldsymbol{y}_k - \boldsymbol{y}_j\|^2\right), \gamma > 0$$

  - Sigmoid Kernel

$$K(k,j) = \tanh(\beta_0 \boldsymbol{y}_k{}^T \boldsymbol{y}_j + \beta_1)$$

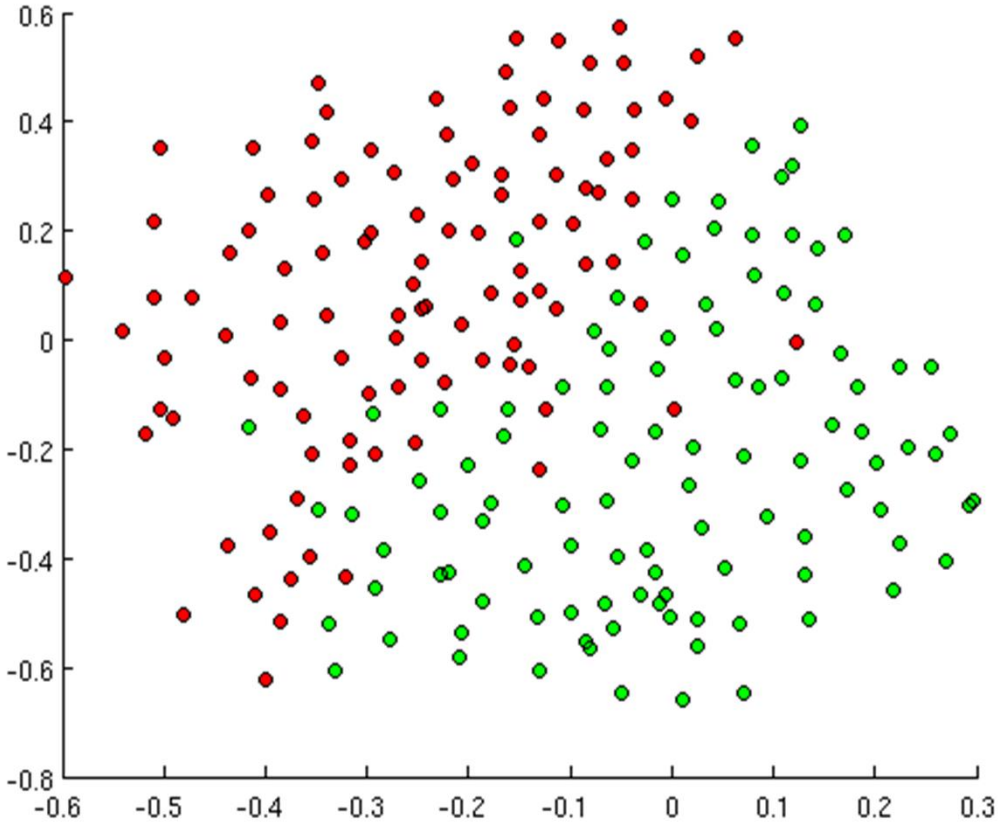
Kernel SVM





KSVM with RBF Kernel

$$K(k,j) = \exp\left(-\gamma \left\| \boldsymbol{y}_k - \boldsymbol{y}_j \right\|^2\right), \gamma > 0$$



Source: http://openclassroom.stanford.edu/

Kernel Trick

- Kernels can be defined on general types of data and many classical algorithms can naturally work with general, non-vectorial, data-types !

- Since the kernelization requires only the inner product (generalization of the dot product in infinite dimensional spaces) matrix, one can avoid defining an explicit mapping function φ.

- For example, kernels on strings, trees and graphs which exploits sequence or topology of the underlying data domain for computing (normalized) similarity which can be represented as inner product.

Kernel SVM

- Choose a kernel function (difficult choice)
- Solve the quadratic programming problem (many software packages available)
- Construct the discriminant function from the support vectors

SVM Pros

- Flexibility in choosing a similarity (kernel) function
- Sparseness of solution when dealing with large data sets due to selective support vectors
- Ability to handle large feature spaces as complexity is independent of feature dimensionality
- Overfitting can be controlled by soft margin approach
- A simple convex optimization problem which is guaranteed to converge to a single global solution

SVM Cons

- It is sensitive to noise
  - A relatively small number of mislabeled examples can dramatically decrease the performance
- It only considers two classes so for Multi-class SVM learn with output arity m:
  - SVM 1 learns “Output==1” vs “Output != 1”
  - SVM 2 learns “Output==2” vs “Output != 2”
  - :
  - SVM m learns “Output==m” vs “Output != m”
- To predict the output for a new test point, just predict with each SVM and find out which one puts the prediction the furthest into the positive region.

SVM Example

- Using the data points, compute the kernel matrix and write the dual formulation.

$$\mathcal{L}(\boldsymbol{\alpha}) \sum_{k=1}^{n} \alpha_k - \frac{1}{2} \sum_{k=1,j=1}^{n} \alpha_k \alpha_j z_k z_j \boldsymbol{y}_k{}^T \boldsymbol{y}_j \qquad G = \begin{pmatrix} 9 & 1 & 1 \\ 1 & 9 & 9 \\ 1 & 9 & 25 \end{pmatrix} \qquad z = [-1 \quad 1 \quad 1]$$

$$\text{Maximize:} \quad \alpha_1 + \alpha_2 + \alpha_3 - \frac{1}{2}\left(9\alpha_1^2 - 2\alpha_1\alpha_2 - 2\alpha_1\alpha_3 + 9\alpha_2^2 + 18\alpha_2\alpha_3 + 25\alpha_3^2\right)$$

$$\text{subject to:} \quad \alpha_1 \geq 0,\ \alpha_2 \geq 0,\ \alpha_3 \geq 0, \quad -\alpha_1 + \alpha_2 + \alpha_3 = 0$$

- Solve for Lagrangian multipliers by setting the partial derivatives of the criterion function to zero and substitutions using the constraints.

$$\alpha_1 = 1/8, \quad \alpha_2 = 1/8, \quad \alpha_3 = 0$$

- Compute the intercept of the boundary hyperplane for each support vector and take the mean as the final value.

$$b_k = z_k - \sum_{j=1}^{n} \alpha_j z_j \boldsymbol{y}_j{}^T \boldsymbol{y}_k \qquad b_k = -1 - (1 * (-1) * 9 + 1 * 1 * 1)/8 = -1 - (-9 + 8/8)/8 = 0$$