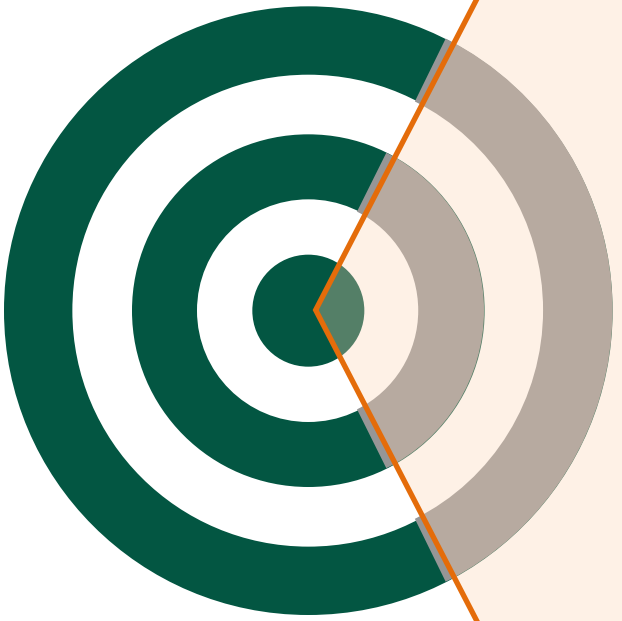
Support Vector Machines

Agenda



In this session, you will learn about:

- Classification as hyper plane location problem
- Motivation for linear Support Vectors (SV) in classification problem
- SVM as a quadratic optimization problem
- Real life application using SVM
- Non-linear SVM
- Introduction to Kernel methods
- Summary
- References

Classification Tasks

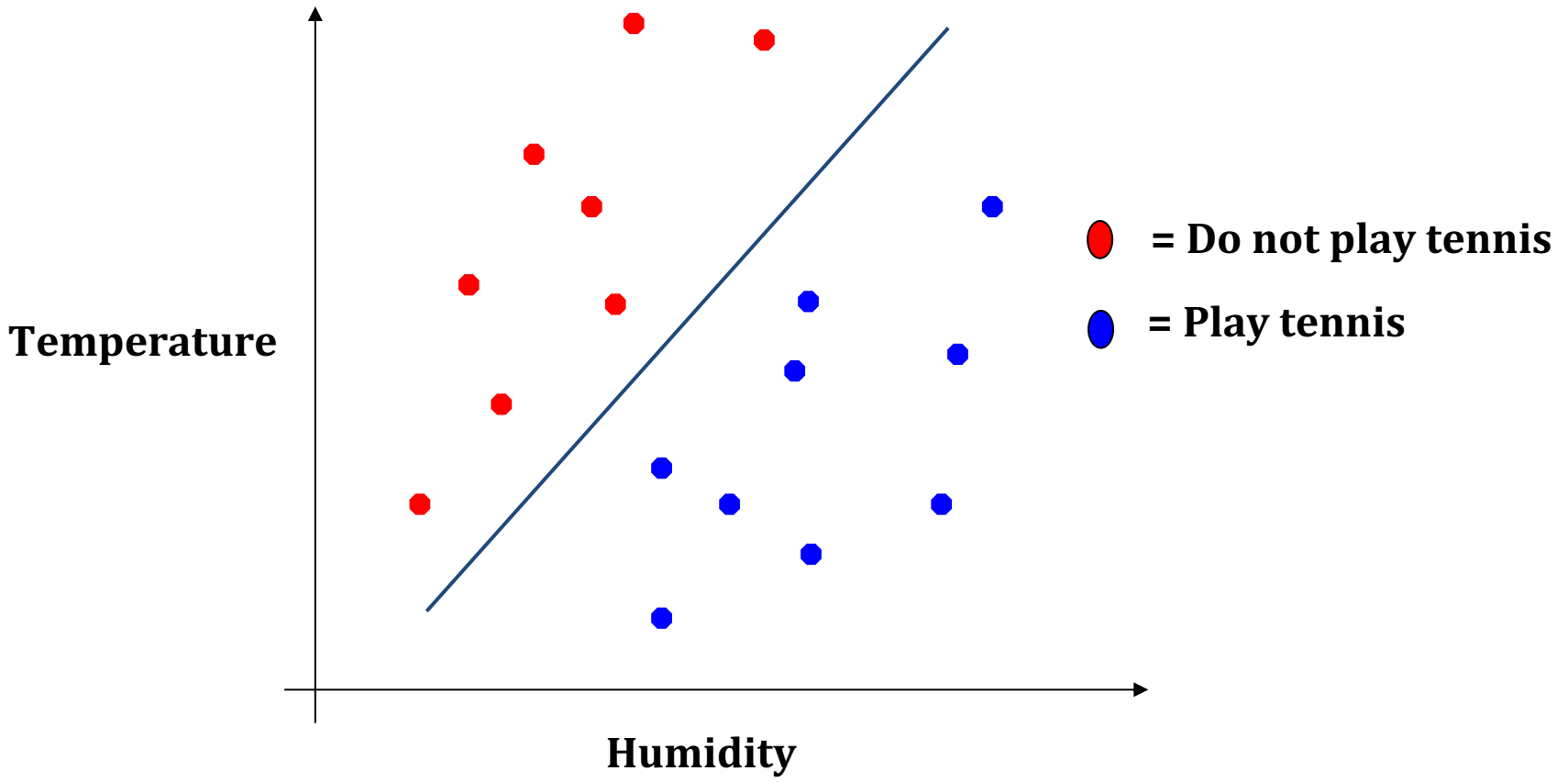
Learning Task

- Given: Expression profiles of cancer patients and healthy persons.
- Compute: A model distinguishing if a person has cancer from expression data.

Classification Task

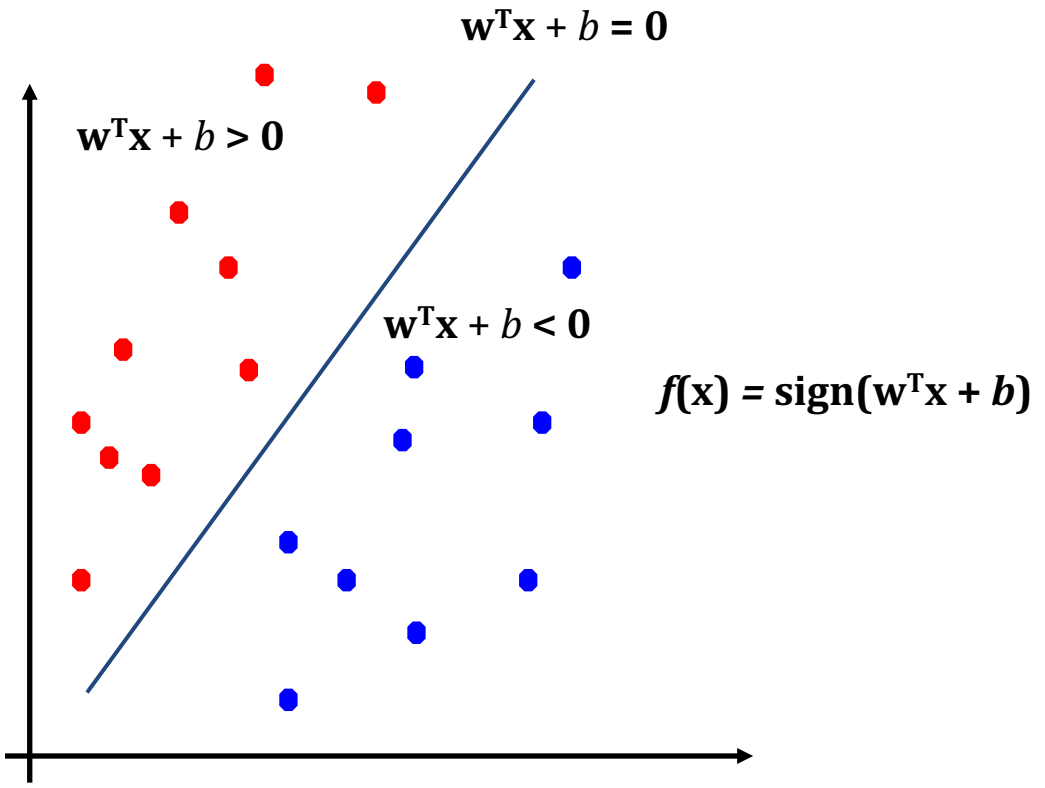
- Given: Expression profile of a new patient + a learned model
- Determine: If a patient has cancer or not.

Tennis Example



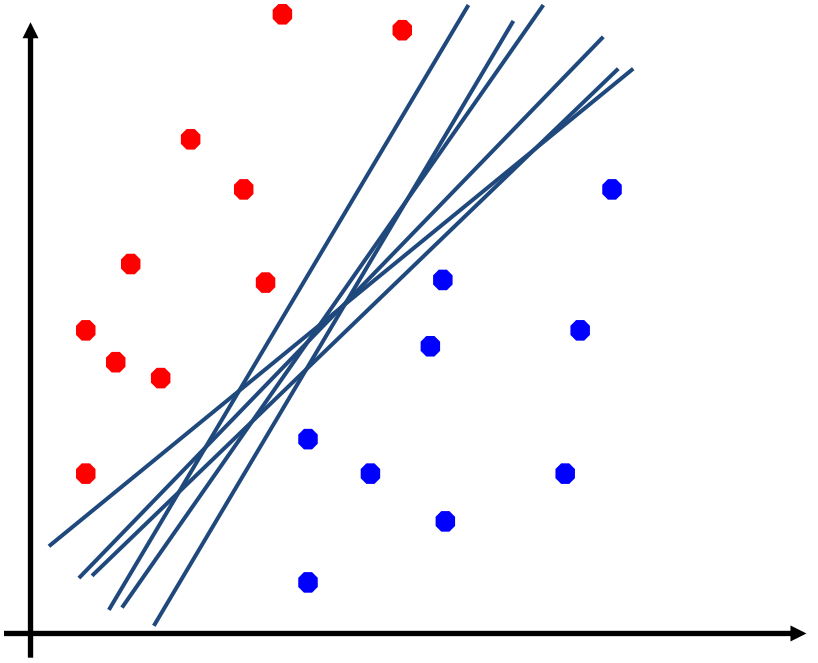
Introdution to Linear Separators

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



Linear Separators

Which of the Linear Separators is Optimal?

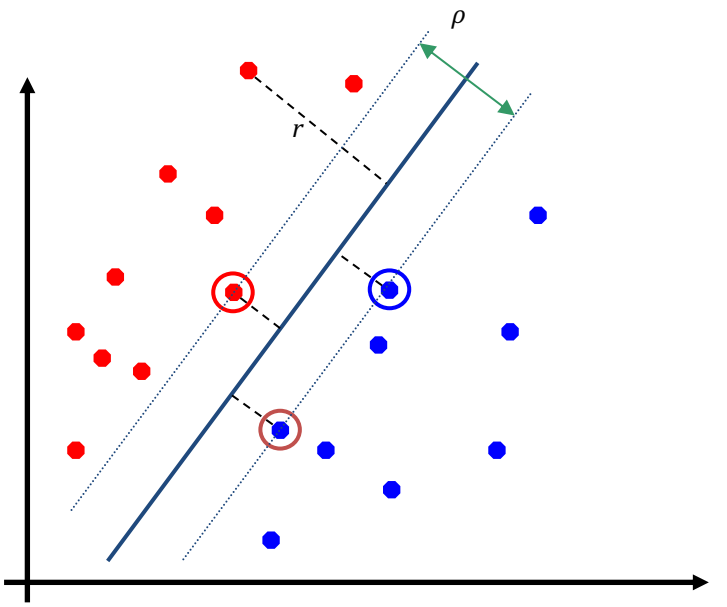


- All hyperplanes in R^d are parameterized by a vector (**w**) and a constant b.
- Can be expressed as **w**•**x**+b=0 (remember the equation for a hyperplane from algebra!).
- Our aim is to find such a hyperplane f(x)=sign(**w**•**x**+b), that correctly classify our data.

Selection of a Good Hyper-Plane

- **Objective**:
  - Select a `good' hyper-plane using only the data!

- **Intuition**:
  - (Vapnik 1965) - assuming linear seperability.
    - Separate the data.
    - Place hyper-plane `**far**' from data.



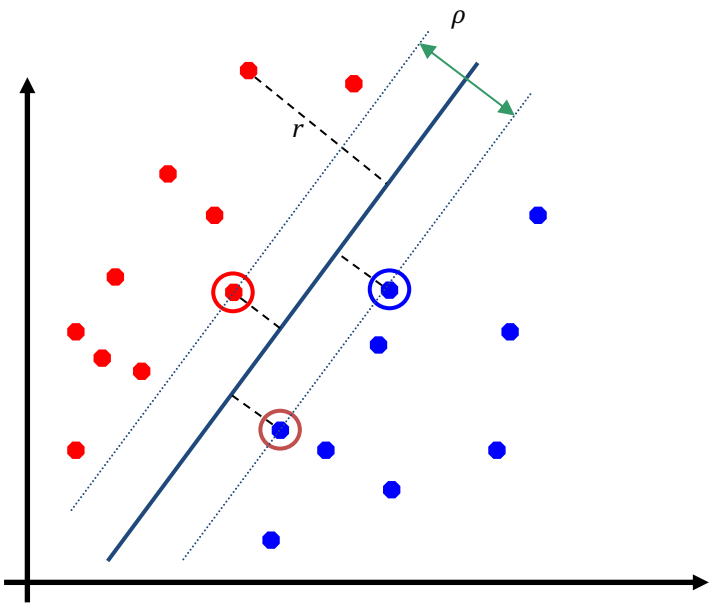
Maximizing the Margin

- Recall: The distance from a point (x_0, y_0) to a line $Ax + By + c = 0$ is

$$\frac{|Ax_0 + B\,y_0 + c|}{\sqrt{A^2 + B^2}}$$

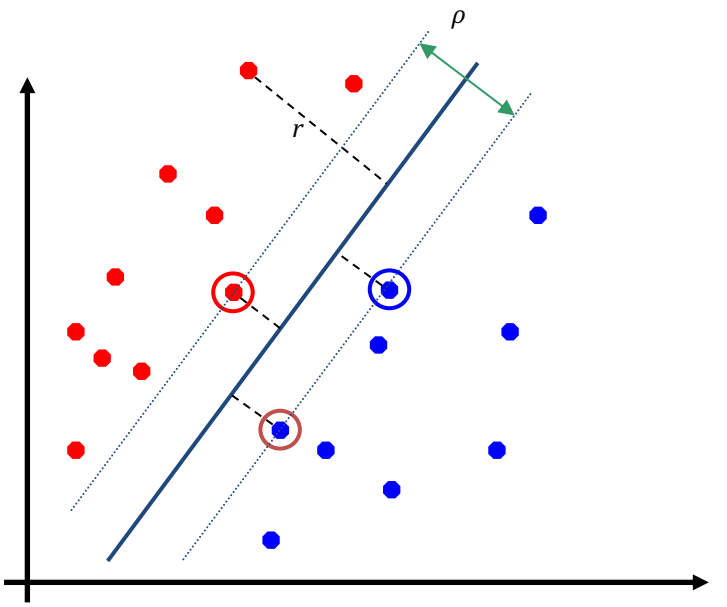
- Distance from example xi to the separator is

$$\mathrm{r} = \frac{|w^T x_i + b|}{\|w\|}$$



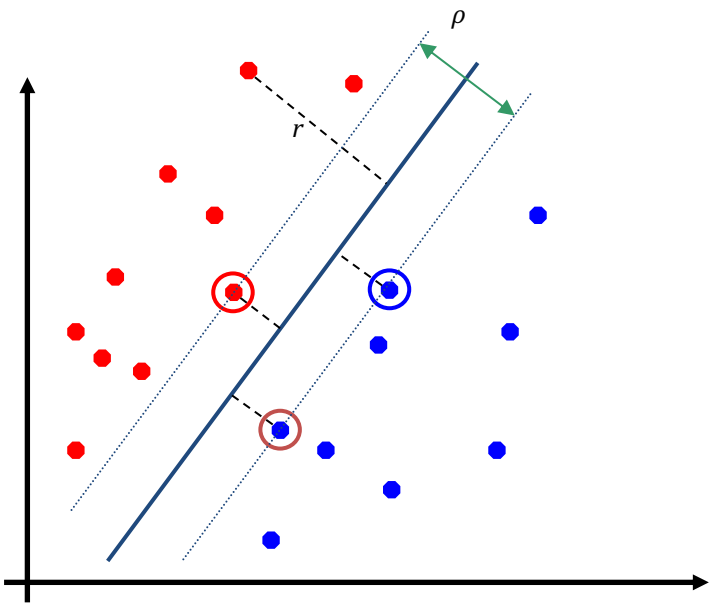
Classification Margin

- Examples closest to the hyperplane are **support vectors**.
- **Margin** ρ of the separator is the distance between support vectors.



Maximum Margin Classification(Hard Margin Classifier)

- Maximizing the margin is good according to intuition.
- Implies that only support vectors matter; other training examples are ignorable.



Linear SVM Mathematically (1/2)

- Let training set $\{(\mathbf{x}_i, y_i)\}_{i=1..n}$, $\mathbf{x}_i \in \mathbf{R}^d$, $y_i \in \{-1, 1\}$ be separated by a hyperplane with margin ρ. Then for each training example $(\mathbf{x}_i, y_i)$:

$$w^T x_i + b \geq +\rho/2 \ \ when\ y_i = +1$$
$$w^T x_i + b \leq -\frac{\rho}{2} \ \ when\ y_i = -1$$
$$\Leftrightarrow \quad y_i\,(w^T x_i) \geq \rho/2$$

- For every support vector $\mathbf{x}_s$ the above inequality is an equality.
- After rescaling **w** and b by $\rho/2$ in the equality, we obtain that distance between each $\mathbf{x}_s$ and the hyperplane is :

$$r = \frac{y_s(\mathbf{w}^T\mathbf{x}_s + b)}{\|\mathbf{w}\|} = \frac{1}{\|\mathbf{w}\|}$$

- Then the margin can be expressed through (rescaled) **w** and b as:

$$\rho = 2r = \frac{2}{\|\mathbf{w}\|}$$

Linear SVM Mathematically (2/2)

Then we can formulate the quadratic optimization problem.

Find $\mathbf{w}$ and b such that $\rho = \frac{2}{\|\mathbf{w}\|}$

is maximized

and for all $(\mathbf{x}_i, y_i)$, i=1..n : $y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1$

Which can be reformulated as:

Find $\mathbf{w}$ and b such that

$\Phi(\mathbf{w}) = \|\mathbf{w}\|^2 = \mathbf{w}^T\mathbf{w}$ is minimized

and for all $(\mathbf{x}_i, y_i)$, i=1..n : $y_i (\mathbf{w}^T\mathbf{x}_i + b) \geq 1$

Solving the Optimization Problem

- 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 $\mathbf{w}$ and b such that:
$\mathbf{\Phi(w)} = \mathbf{w}^T\mathbf{w}$ is minimized

and for all $(\mathbf{x}_i, y_i)$, $i=1..n$: $y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1$

Find $\alpha_1 \ldots \alpha_n$ such that:
$\mathbf{Q(\alpha)} = \sum \alpha_i - 1/2 \sum_i \sum_j \alpha_i \alpha_j y_i y_j \boldsymbol{x}_i^T \boldsymbol{x}_j^T$ is maximized and

(1) $\sum \alpha_i y_i = 0$
(2) $\alpha_i \geq 0$ for all α_i

Optimization Problem Solution

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

$$\mathbf{w} = \Sigma \alpha_i y_i \mathbf{x}_i \qquad b = y_k - \Sigma \alpha_i y_i \mathbf{x}_i^{\mathrm{T}} \mathbf{x}_k \quad \text{for any } \alpha_k > 0$$

- Each non-zero α_i indicates that corresponding $\mathbf{x_i}$ is a support vector.

- Then the classifying function is (note that we don't need **w** explicitly):

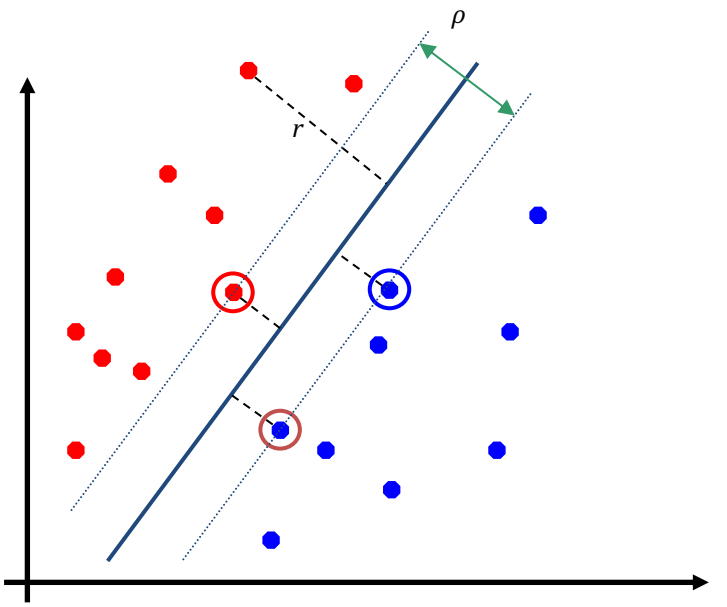
$$f(\mathbf{x}) = \sum \alpha_i y_i \mathbf{x}_i^{\mathrm{T}} \mathbf{x} + b$$

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

- Also keep in mind that solving the optimization problem involved computing the inner products $\mathbf{x}_i^{\mathrm{T}} \mathbf{x}_j$ between all training points.

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 Mathematically

- The old formulation:

  Find $\mathbf{w}$ and b such that
  $\mathbf{\Phi(w)} = \mathbf{w}^T\mathbf{w}$ is minimized
  and for all $(\mathbf{x}_i, y_i)$, $i=1..n$: $y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1$

- Modified formulation incorporates slack variables:

  Find $\mathbf{w}$ and b such that
  $\mathbf{\Phi(w)} = \mathbf{w}^T\mathbf{w} + C\Sigma\xi_i$ is minimized
  and for all $(\mathbf{x}_i, y_i)$, $i=1..n$: $y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 - \xi_i$, $\xi_i \geq 0$

- 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.

Soft Margin Classification - Solution

- Dual problem is identical to separable case (would not be identical if the 2-norm penalty for slack variables $C\Sigma\xi_i^2$ was used in primal objective, we would need additional Lagrange multipliers for slack variables):

  Find $\alpha_1 \ldots \alpha_N$ such that
  $\mathbf{Q}(\boldsymbol{\alpha}) = \Sigma\alpha_i - \frac{1}{2}\Sigma\Sigma\alpha_i\alpha_j y_i y_j \mathbf{x}_i^T\mathbf{x}_j$ is maximized and
  (1) $\Sigma\alpha_i y_i = 0$
  (2) $0 \leq \alpha_i \leq C$ for all α_i

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

$\mathbf{w} = \Sigma\alpha_i y_i \mathbf{x}_i$
$b = y_k(1 - \xi_k) - \Sigma\alpha_i y_i \mathbf{x}_i^T\mathbf{x}_k$ for any k s.t. $\alpha_k > 0$

Again, we don't need to compute $\mathbf{w}$ explicitly for classification:

$$f(\mathbf{x}) = \Sigma\alpha_i y_i \mathbf{x}_i^T\mathbf{x} + b$$

Theoretical Justification for Maximum Margins

What has Vapnik has proved?

The class of optimal linear separators has VC dimension h bounded from above as :

$$h \leq \min\left\{\left\lceil \frac{D^2}{\rho^2} \right\rceil, m_0\right\} + 1$$

Where ρ is the margin, D is the diameter of the smallest sphere that can enclose all of the training examples, and m_0 is the dimensionality.

- Intuitively, this implies that regardless of dimensionality m_0 we can minimize the VC dimension by maximizing the margin ρ.
- Thus, **complexity of the classifier is kept small regardless of dimensionality**.

Linear SVM - Overview

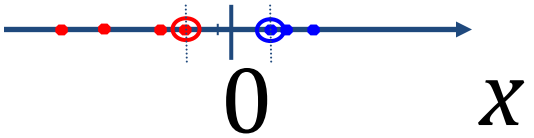
- The classifier is a separating hyperplane.
- Most "important" training points are support vectors; they define the hyperplane.
- Quadratic optimization algorithms can identify which training points $\mathbf{x}_i$ are support vectors with non-zero Lagrangian multipliers α_i.
- Both in the dual formulation of the problem and in the solution training points appear only inside inner products:

Find $\alpha_1 \ldots \alpha_N$ such that
$\mathbf{Q}(\boldsymbol{\alpha}) = \Sigma\alpha_i - \frac{1}{2}\Sigma\Sigma\alpha_i\alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j$ is maximized
and
(1) $\Sigma\alpha_i y_i = 0$
(2) $0 \leq \alpha_i \leq C$ for all α_i

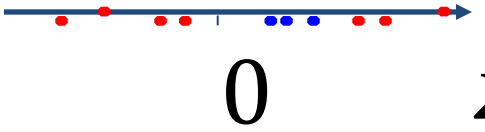
$f(\mathbf{x}) = \Sigma\alpha_i y_i \mathbf{x}_i^T \mathbf{x} + b$

Non-Linear SVM

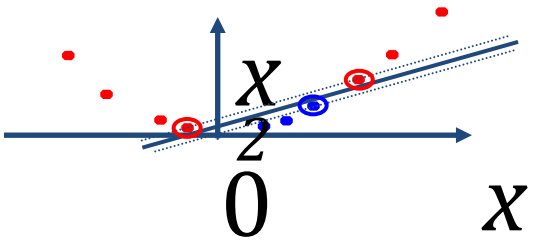
- Datasets that are linearly separable with some noise work out great:



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

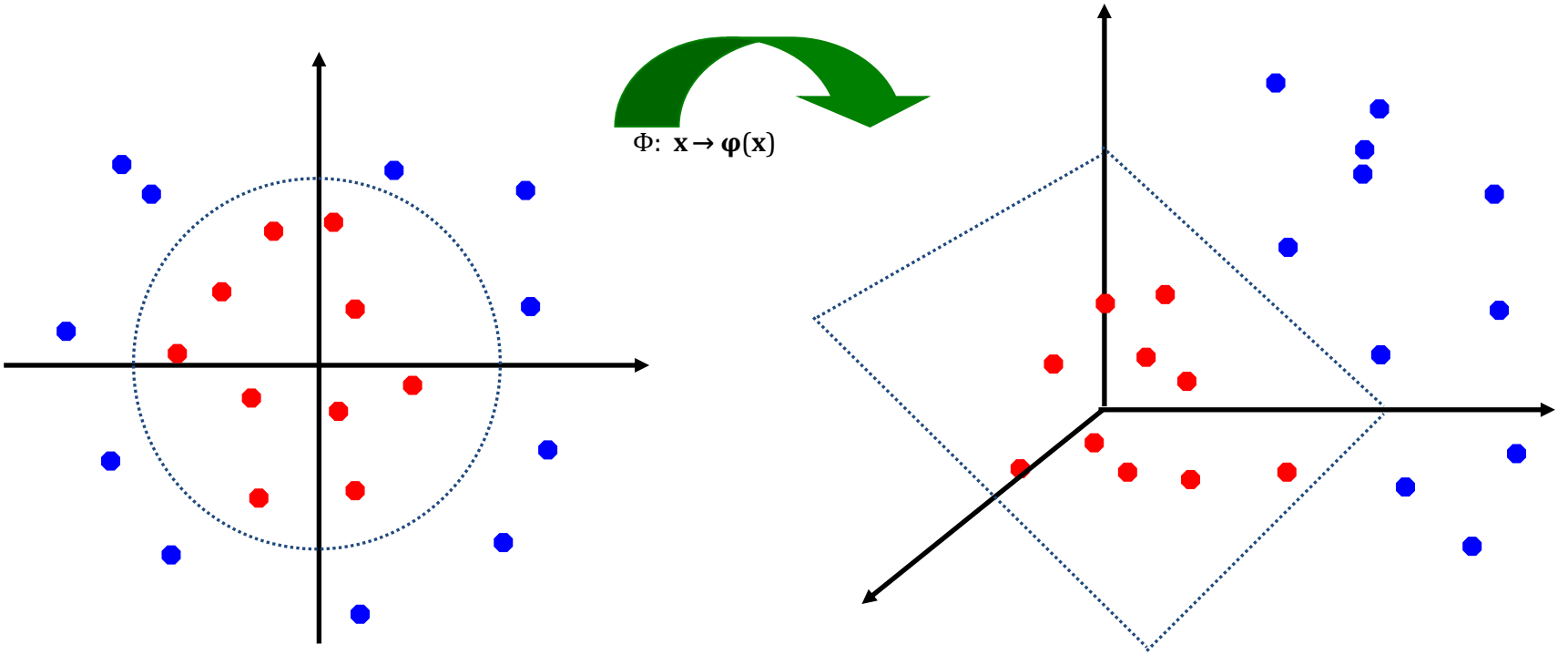


- How about… mapping data to a higher-dimensional space:



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 $K(\mathbf{x}_i,\mathbf{x}_j)=\mathbf{x}_i^{\mathbf{T}}\mathbf{x}_j$
- If every data point is mapped into high-dimensional space via some transformation Φ: $\mathbf{x} \rightarrow \varphi(\mathbf{x})$, the inner product becomes:

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

- A *kernel function* is a function that is equivalent to an inner product in some feature space.
- Example:
  - 2-dimensional vectors $\mathbf{x}=[x_1 \ x_2]$; let $K(\mathbf{x}_i,\mathbf{x}_j)=(1+\mathbf{x}_i^{\mathbf{T}}\mathbf{x}_j)^2$,
    Need to show that $K(\mathbf{x}_i,\mathbf{x}_j)= \boldsymbol{\varphi}(\mathbf{x}_i)^{\mathbf{T}}\boldsymbol{\varphi}(\mathbf{x}_j)$:

The "Kernel Trick"

$$K(\mathbf{x}_i,\mathbf{x}_j)=(1+\mathbf{x}_i^T\mathbf{x}_j)^2,= 1+ x_{i1}^2x_{j1}^2 + 2\, x_{i1}x_{j1}\, x_{i2}x_{j2}+ x_{i2}^2x_{j2}^2 + 2x_{i1}x_{j1} + 2x_{i2}x_{j2}$$

$$= [1 \;\; x_{i1}^2 \;\; \sqrt{2}\, x_{i1}x_{i2} \;\; x_{i2}^2 \;\; \sqrt{2}x_{i1} \;\; \sqrt{2}x_{i2}]^T\, [1 \;\; x_{j1}^2 \;\; \sqrt{2}\, x_{j1}x_{j2} \;\; x_{j2}^2 \;\; \sqrt{2}x_{j1} \;\; \sqrt{2}x_{j2}]$$

$$= \boldsymbol{\varphi}(\mathbf{x}_i)^T\boldsymbol{\varphi}(\mathbf{x}_j), \quad \text{where } \boldsymbol{\varphi}(\mathbf{x}) = [1 \;\; x_1^2 \;\; \sqrt{2}\, x_1x_2 \;\; x_2^2 \;\; \sqrt{2}x_1 \;\; \sqrt{2}x_2]$$

- Thus, a kernel function implicitly maps data to a high-dimensional space (without the need to compute each $\boldsymbol{\varphi}(\mathbf{x})$ explicitly).

What Functions are Kernels?

- For some functions $K(\mathbf{x}_i,\mathbf{x}_j)$ checking that $K(\mathbf{x}_i,\mathbf{x}_j)= \varphi(\mathbf{x}_i)^T\varphi(\mathbf{x}_j)$ can be cumbersome.
- Mercer's theorem:

  Every semi-positive definite symmetric function is a kernel
- Semi-positive definite symmetric functions correspond to a semi-positive definite symmetric Gram matrix:

K=

$K(\mathbf{x}_1,\mathbf{x}_1)$	$K(\mathbf{x}_1,\mathbf{x}_2)$	$K(\mathbf{x}_1,\mathbf{x}_3)$	…	$K(\mathbf{x}_1,\mathbf{x}_n)$
$K(\mathbf{x}_2,\mathbf{x}_1)$	$K(\mathbf{x}_2,\mathbf{x}_2)$	$K(\mathbf{x}_2,\mathbf{x}_3)$		$K(\mathbf{x}_2,\mathbf{x}_n)$
…	…	…	…	…
$K(\mathbf{x}_n,\mathbf{x}_1)$	$K(\mathbf{x}_n,\mathbf{x}_2)$	$K(\mathbf{x}_n,\mathbf{x}_3)$	…	$K(\mathbf{x}_n,\mathbf{x}_n)$

Examples of Kernels Functions

- Linear: $K(\mathbf{x}_i,\mathbf{x}_j)= \mathbf{x}_i^T\mathbf{x}_j$
  - Mapping Φ: $\mathbf{x} \rightarrow \boldsymbol{\varphi}(\mathbf{x})$, where $\boldsymbol{\varphi}(\mathbf{x})$ is $\mathbf{x}$ itself

- Polynomial of power p: $K(\mathbf{x}_i,\mathbf{x}_j)= (1+ \mathbf{x}_i^T\mathbf{x}_j)^p$
  - Mapping Φ: $\mathbf{x} \rightarrow \boldsymbol{\varphi}(\mathbf{x})$, where $\boldsymbol{\varphi}(\mathbf{x})$ has $\binom{d+p}{p}$ dimensions

- Gaussian (radial-basis function): $K(\mathbf{x}_i,\mathbf{x}_j) = e^{-\frac{\|\mathbf{x}_i-\mathbf{x}_j\|^2}{2\sigma^2}}$
  - Mapping Φ: $\mathbf{x} \rightarrow \boldsymbol{\varphi}(\mathbf{x})$, where $\boldsymbol{\varphi}(\mathbf{x})$ is infinite-dimensional: every point is mapped to a function (a Gaussian); combination of functions for support vectors is the separator.

- Higher-dimensional space still has intrinsic dimensionality d (the mapping is not onto), but linear separators in it correspond to non-linear separators in original space.

Non-linear SVMs Mathematically

- Dual problem formulation:

  Find $\alpha_1 \ldots \alpha_n$ such that
  $\mathbf{Q}(\boldsymbol{\alpha}) = \Sigma\alpha_i - \frac{1}{2}\Sigma\Sigma\alpha_i\alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j)$ is maximized and
  (1) $\Sigma\alpha_i y_i = 0$
  (2) $\alpha_i \geq 0$ for all α_i

- The solution is: $f(x) = \Sigma\alpha i y i K(xi, xj) + b$

- Optimization techniques for finding α_i's remain the same!

Non Linear SVM - Overview

- SVM locates a separating hyperplane in the feature space and classify points in that space
- It does not need to represent the space explicitly, simply by defining a kernel function
- The kernel function plays the role of the dot product in the feature space.

Properties of SVM

- Flexibility in choosing a similarity function
- Sparseness of solution when dealing with large data sets
  - Only support vectors are used to specify the separating hyperplane
- Ability to handle large feature spaces
  - Complexity does not depend on the dimensionality of the feature space
- Overfitting can be controlled by soft margin approach
- Nice math property: a simple convex optimization problem which is guaranteed to converge to a single global solution
- Feature Selection

Weakness of SVM

- It is sensitive to noise
  - A relatively small number of mislabeled examples can dramatically decrease the performance

- It only considers two classes: How to do multi-class classification with SVM?

Answer

- With output arity m, learn m SVM's
  - 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 input, just predict with each SVM and find out which one puts the prediction the furthest into the positive region.

Demo Using R code

SVM Applications

- SVMs were originally proposed by Boser, Guyon and Vapnik in 1992 and gained increasing popularity in late 1990s.
- SVMs are currently among the best performers for a number of classification tasks ranging from text to genomic data.
- SVMs can be applied to complex data types beyond feature vectors (e.g. graphs, sequences, relational data) by designing kernel functions for such data.
- SVM techniques have been extended to a number of tasks such as regression [Vapnik *et al.* '97], principal component analysis [Schölkopf *et al.* '99], etc.
- Most popular optimization algorithms for SVMs use *decomposition* to hill-climb over a subset of α_i's at a time, e.g. SMO [Platt '99] and [Joachims '99]
- Tuning SVMs remains a black art: selecting a specific kernel and parameters is usually done in a try-and-see manner.

References and Resources Used

- "Statistical Learning Theory", Vladimir Vapnik, Wiley-Interscience; 1998
- "Support Vector Machine Classification of Microarray Gene Expression Data", Michael P. S. Brown William Noble Grundy, David Lin, Nello Cristianini, Charles Sugnet, Manuel Ares, Jr., David Haussler
- Text categorization with Support Vector Machines: learning with many relevant features , T. Joachims, ECML - 98
- An excellent tutorial on VC-dimension and Support Vector Machines: C.J.C. Burges.
  - A tutorial on support vector machines for pattern recognition. Data Mining and Knowledge Discovery, 2(2):955-974, 1998.
- Majority of the slides are form Prof. Andrews SVM tutorial
  - http://www.cs.cmu.edu/~awm/tutorials
- www.cs.utexas.edu/users/mooney/cs391L/svm.ppt
- http://www.labs.iro.umontreal.ca/~pift6080/H09/documents/papers/svm_tutorial.ppt
- http://cis-linux1.temple.edu/~latecki/Courses/AI-Fall10/Lectures/ch5SVM.ppt

Which to Use: SVM or Logistic Regression?

- When classes are (nearly) separable, SVM does better than LR. So does LDA.
- When not, LR (with ridge penalty) and SVM very similar.
- If you wish to estimate probabilities, LR is the choice.
- For nonlinear boundaries, kernel SVMs are popular. Can use kernels with LR and LDA as well, but computations are more expensive.

IMARTICUS LEARNING

Thank you

Mumbai | Bangalore | Pune | Chennai | Jaipur

ACCREDITED TRAINING PARTNER:

CISI
CHARTERED INSTITUTE FOR SECURITIES & INVESTMENT

IIBA International Institute of Business Analysis