Support Vector Machines (SVM)

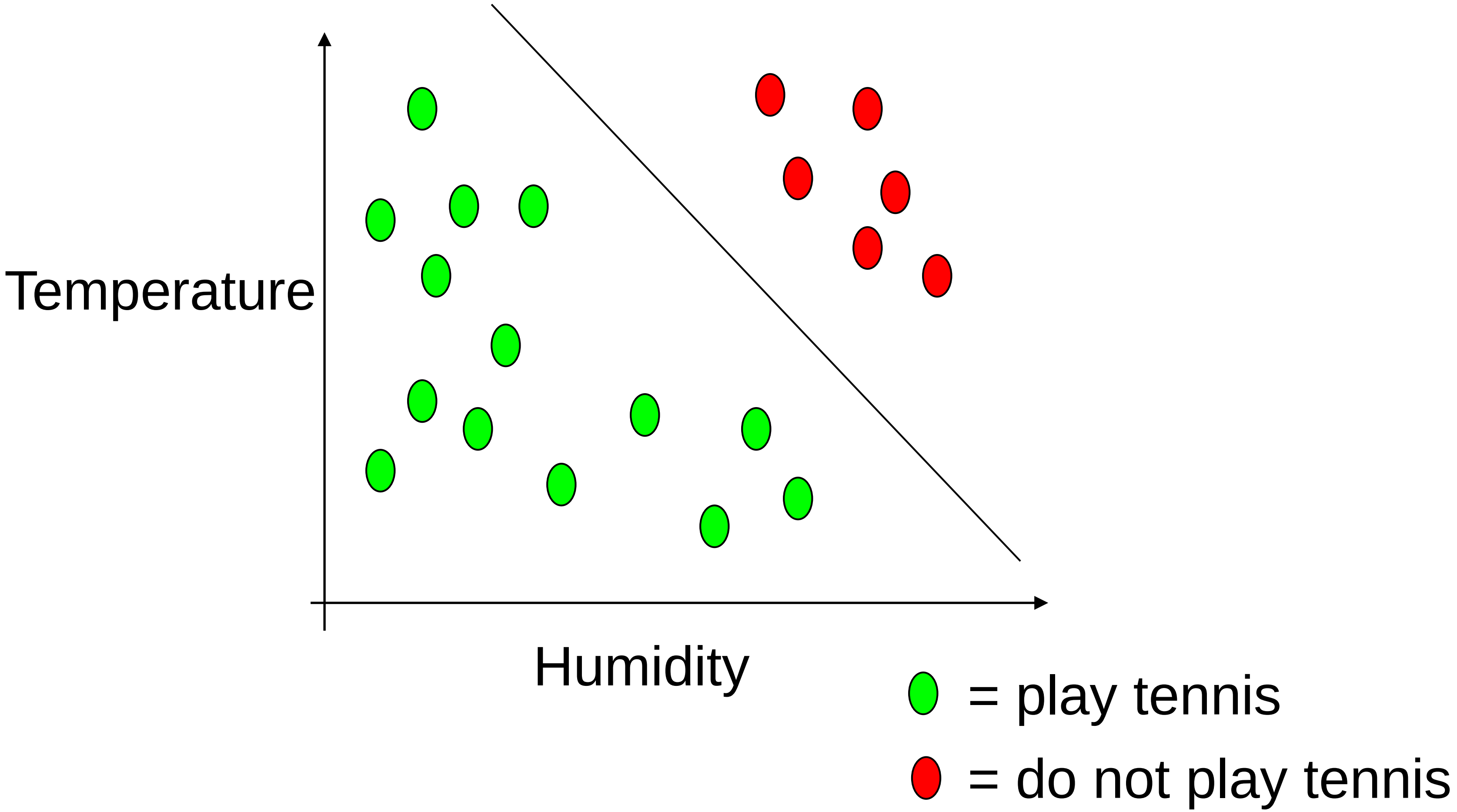
DR. M. SRIDEVI

Problems

- Table below contains data on 12 automobiles tested for emission control. X values are the miles driven and age in months, and the Y values indicate emission passed (Y=+1) or failed (Y=-1). Fit an SVM and obtain the support vectors and margin

Auto age	Mileage	Emission test
8	85000	Pass
11	104000	Fail
7	58000	Pass
9	975000	Fail
16	152000	Fail
8	56500	Pass
10	120000	Fail
13	129000	Fail
14	151000	Fail
10	79000	Pass
15	135000	Fail
12	75000	Pass

Tennis example



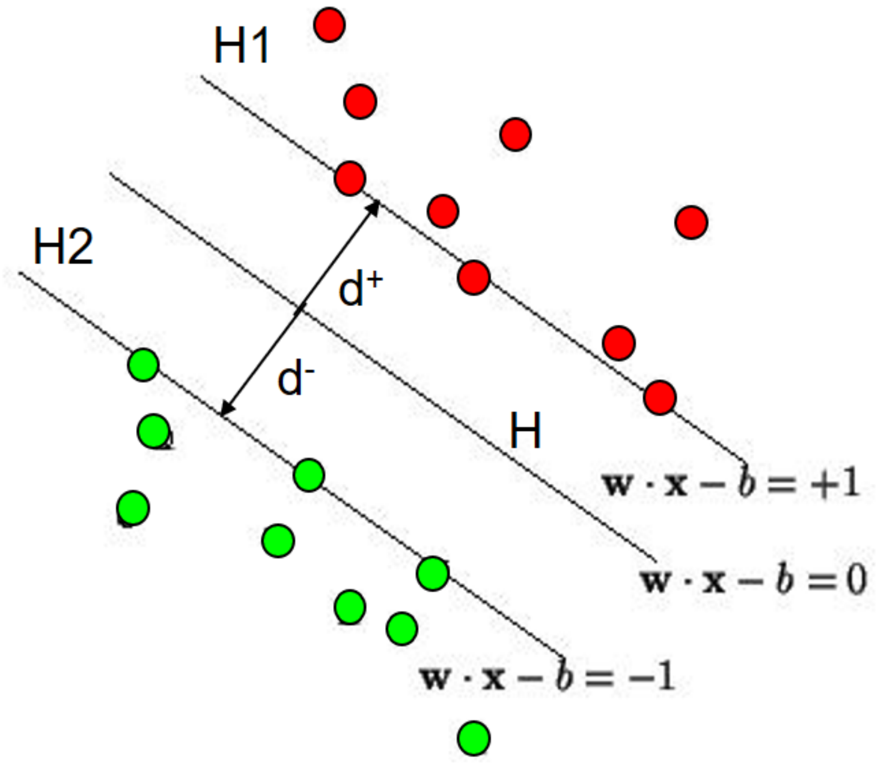
Support Vector Machines

- A popular model for Classification and Prediction
- SVM can be defined as a method for creation of an optimal hyperplane in a multi dimensional space such that the hyperplane separates the two categories and has the lowest possible misclassification error.
- The hyperplane has the lowest misclassification error when it has the largest possible margin between the hyperplane and the nearest plot in the training set on either side of the hyperplane.
- Such a hyperplane can be called the maximum-margin hyperplane.

Define the hyperplane H such that:

$\mathbf{x}_i \bullet \mathbf{w} + b \geq +1$ when $y_i = +1$

$\mathbf{x}_i \bullet \mathbf{w} + b \leq -1$ when $y_i = -1$


H1
H2
d^+
d^-
H
$\mathbf{w} \cdot \mathbf{x} - b = +1$
$\mathbf{w} \cdot \mathbf{x} - b = 0$
$\mathbf{w} \cdot \mathbf{x} - b = -1$

H1 and H2 are the planes:

H1: $\mathbf{x}_i \bullet \mathbf{w} + b = +1$

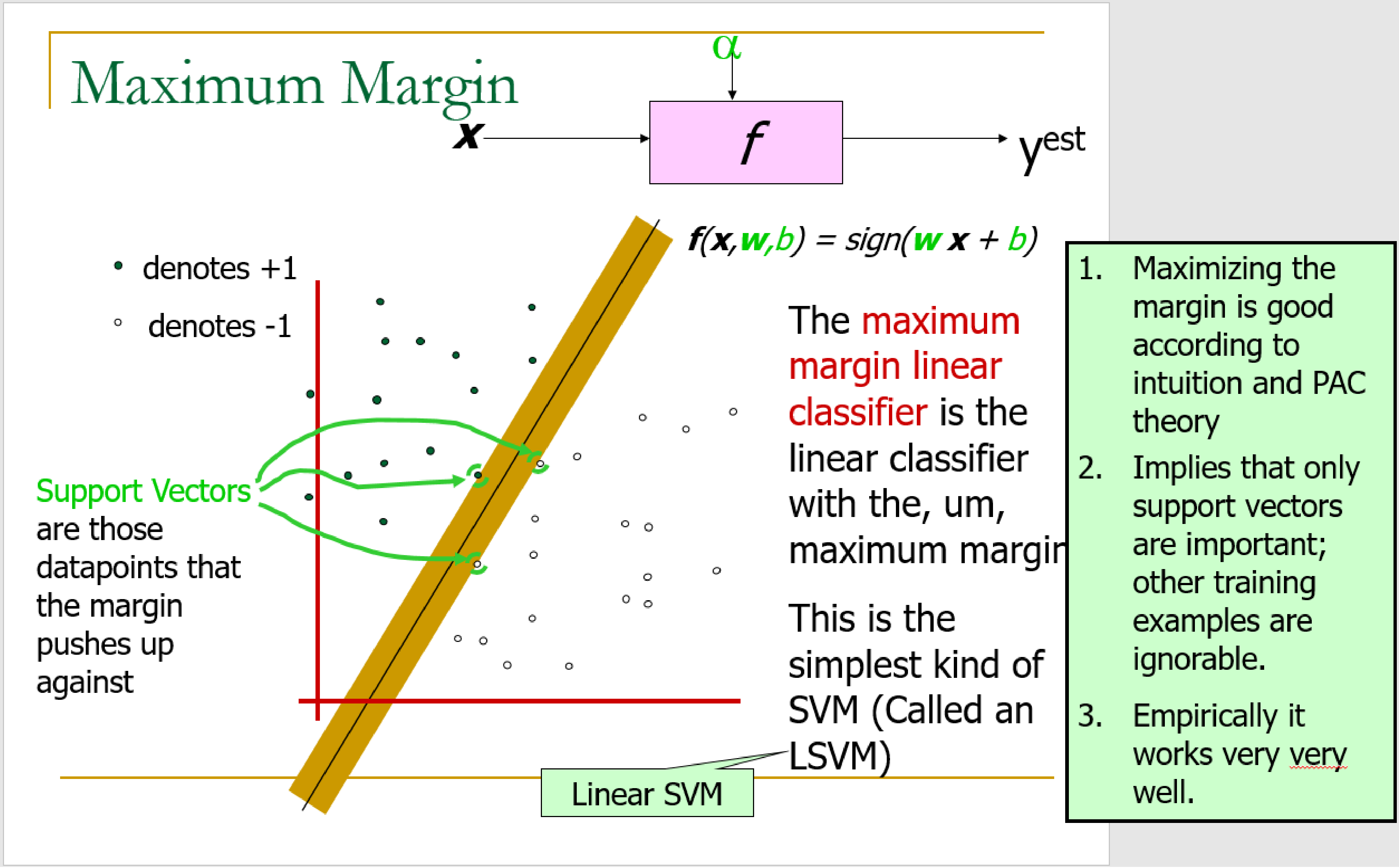
H2: $\mathbf{x}_i \bullet \mathbf{w} + b = -1$

The points on the planes H1 and H2 are the Support Vectors

d+ = the shortest distance to the closest positive point

d- = the shortest distance to the closest negative point

The margin of a separating hyperplane is $d^+ + d^-$.



Maximum Margin

$f(\boldsymbol{x},\boldsymbol{w},b) = sign(\boldsymbol{w}\ \boldsymbol{x} + b)$

Support Vectors are those datapoints that the margin pushes up against

The maximum margin linear classifier is the linear classifier with the, um, maximum margin.

This is the simplest kind of SVM (Called an LSVM)

1. Maximizing the margin is good according to intuition and PAC theory
2. Implies that only support vectors are important; other training examples are ignorable.
3. Empirically it works very very well.

Training a linear SVM

To find the maximum margin separator, we have to solve the following optimization problem:

$$\mathbf{w}.\mathbf{x}^c + b > +1 \quad \textit{for positive cases}$$

$$\mathbf{w}.\mathbf{x}^c + b < -1 \quad \textit{for negative cases}$$

$$\textit{and} \quad \|\mathbf{w}\|^2 \quad \textit{is as small as possible}$$

This is tricky but it’s a convex problem. There is only one optimum and we can find it without fiddling with learning rates or weight decay or early stopping.

- Don’t worry about the optimization problem. It has been solved. Its called quadratic programming.
- It takes time proportional to N^2 which is really bad for very big datasets
    - so for big datasets we end up doing approximate optimization!

Testing a linear SVM

The separator is defined as the set of points for which:

$$\mathbf{w}.\mathbf{x} + b = 0$$

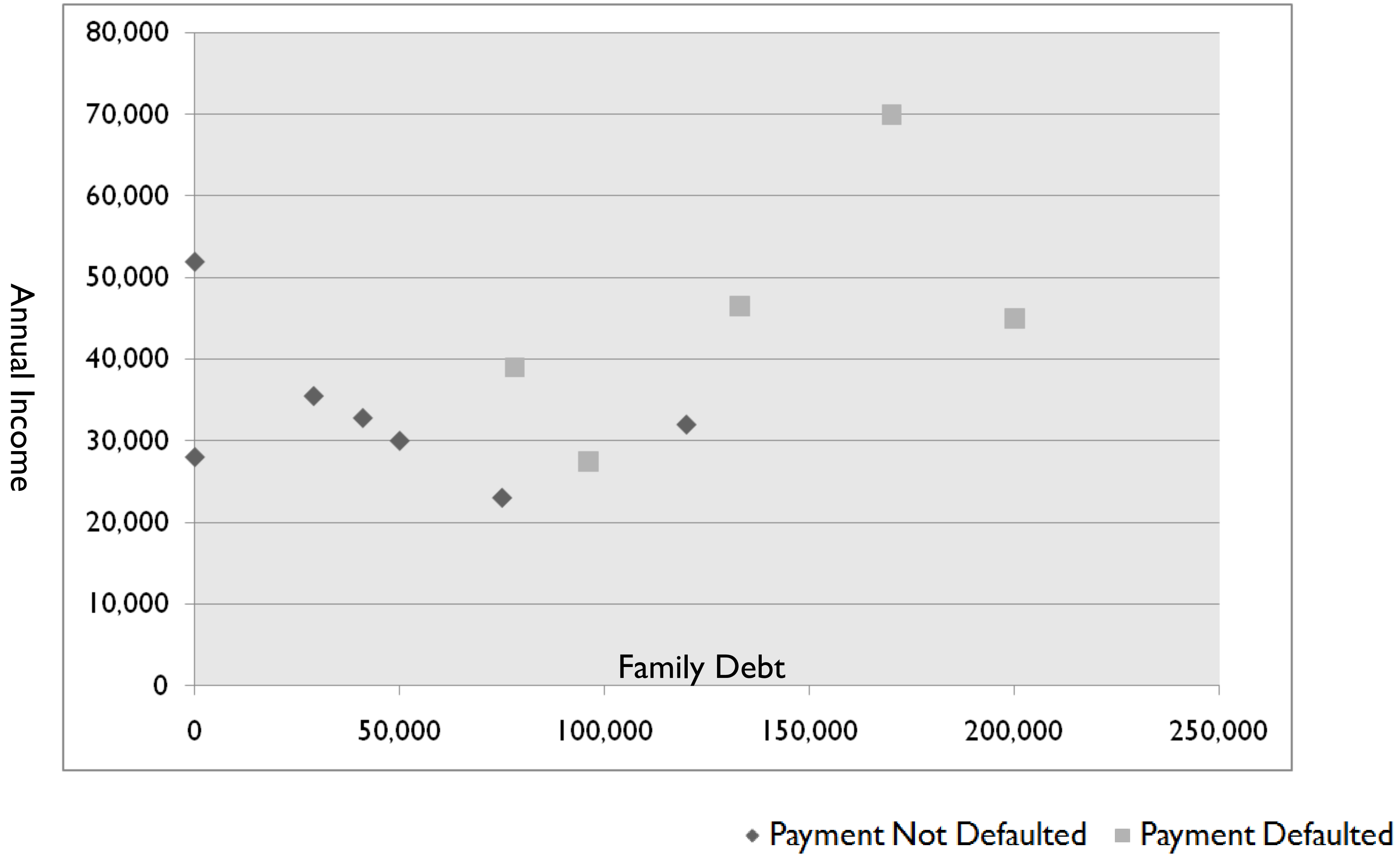
$$so\ if \quad \mathbf{w}.\mathbf{x}^c + b > 0 \ \ say\ its\ a\ positive\ case$$

$$and\ if \quad \mathbf{w}.\mathbf{x}^c + b < 0 \ \ say\ its\ a\ negative\ case$$

Which Customer faults on loan payment?

S.No.	Family Debt	Annual Income	Payment Default	Class Label
1	120,000	32,000	No	-1
2	200,000	45,000	Yes	+1
3	75,000	23,000	No	-1
4	0	52,000	No	-1
5	50,000	30,000	No	-1
6	170,000	70,000	Yes	+1
7	0	28,000	No	-1
8	29,000	35,500	No	-1
9	78,000	39,000	Yes	+1
10	133,000	46,500	Yes	+1
11	96,000	27,500	Yes	+1
12	41,000	32,800	No	-1

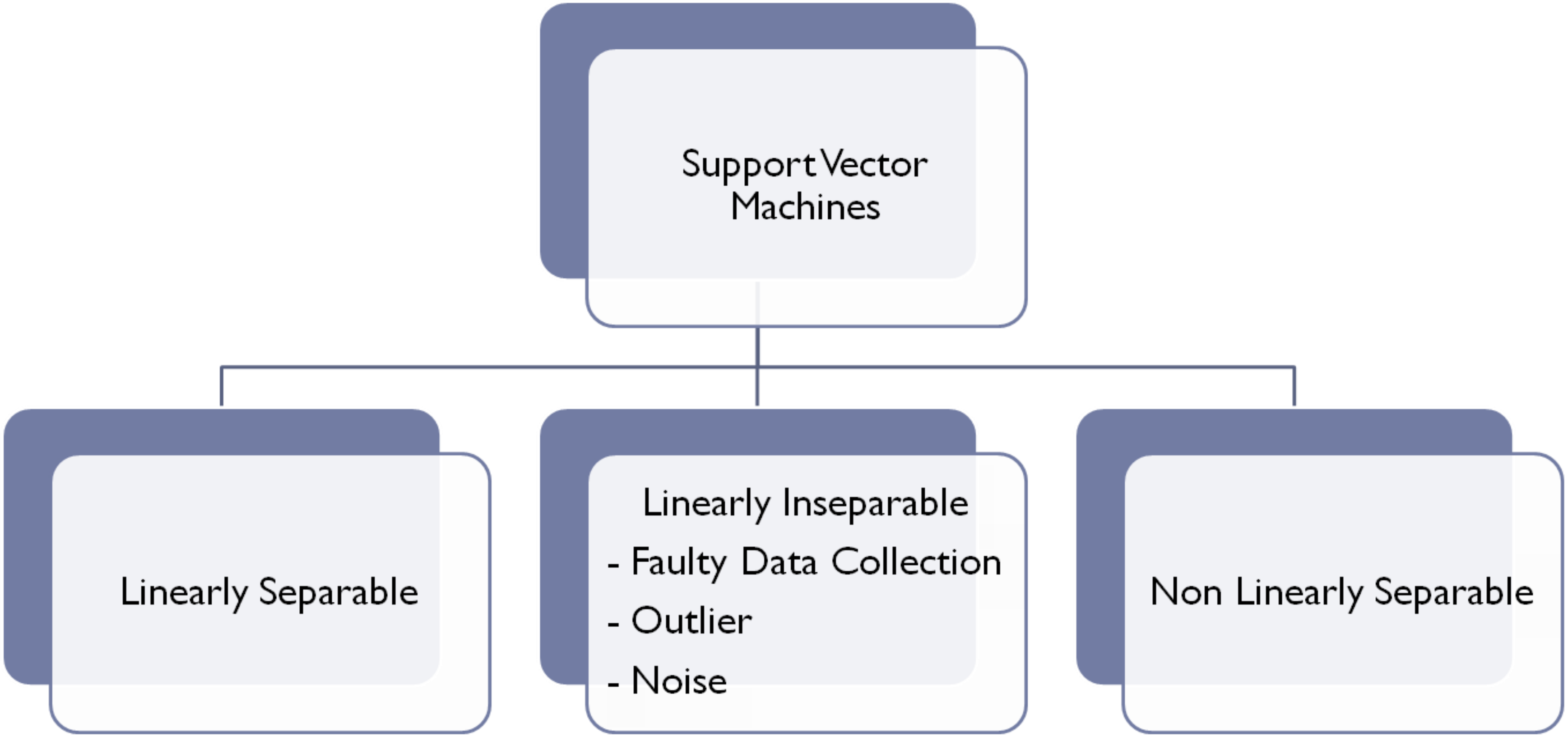
Which Customer faults on loan payment?



High / Low BP Risks

S.No.	Class Label	BMI	W/H Ratio	Sys BP	Dia BP	Chol mg.
1	-1	23.3	0.76	146	68.5	133
2	-1	26.2	0.89	116.5	74	205
3	-1	25.6	0.84	111	69.5	200
4	-1	25.1	1.03	154.5	85.5	218
5	-1	23.3	0.76	108	68	176
6	-1	23.3	1	136.5	84	184
7	-1	23	0.91	146	65	265
8	+1	23.7	0.86	142	84	247
9	-1	33	0.9	104	69.5	170
10	-1	29.7	0.88	127	80.5	201
11	+1	28.5	0.99	138	84	148
12	+1	25.2	0.23	106	62.5	187
13	-1	22.1	0.81	144.5	85.5	238
14	+1	28.4	`.0`	152.5	87.5	184
15	-1	32.2	`.03	117.5	87	187
16	+1	33.6	0.89	126	80	193
17	-1	27.4	0.93	122	80.5	210
18	+1	24.4	0.93	144.5	89	245
19	-1	24.4	0.85	125.5	80.5	221
20	+1	24	0.98	148	92	239

SVM Classification



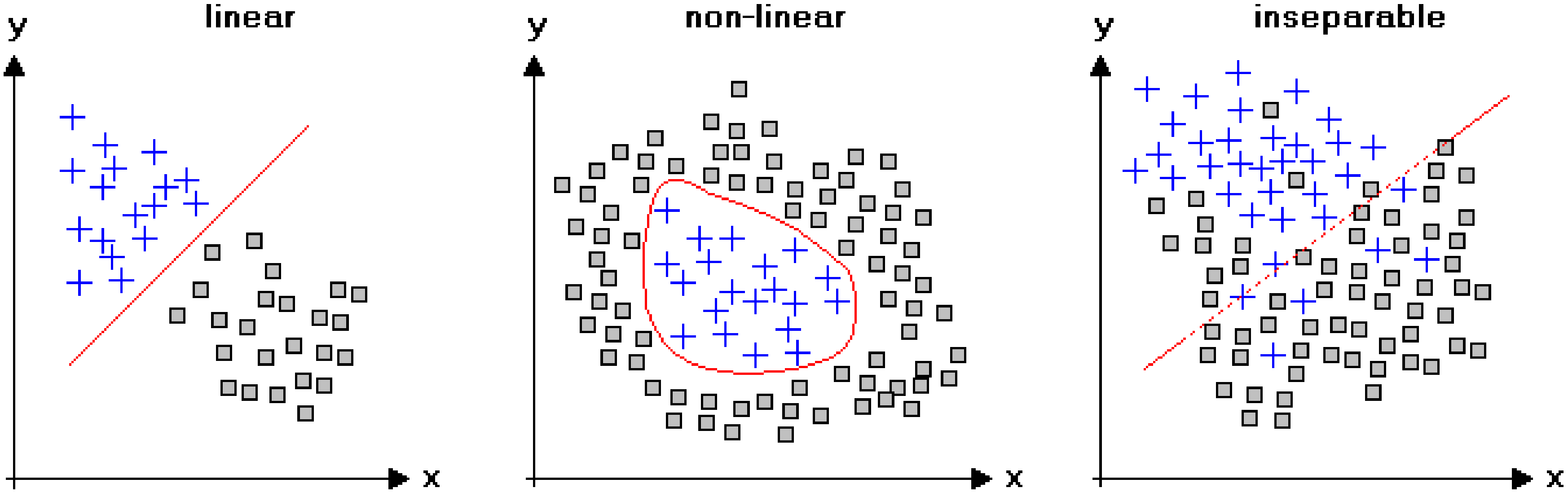
Support Vector Machines

Linearly Separable

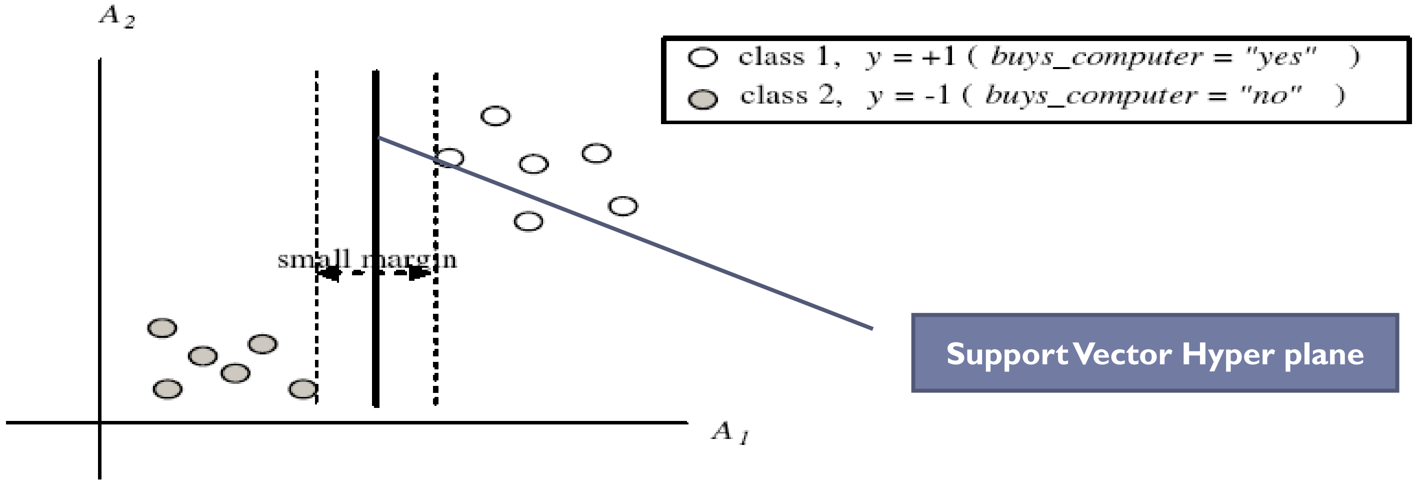
Linearly Inseparable
- Faulty Data Collection
- Outlier
- Noise

Non Linearly Separable

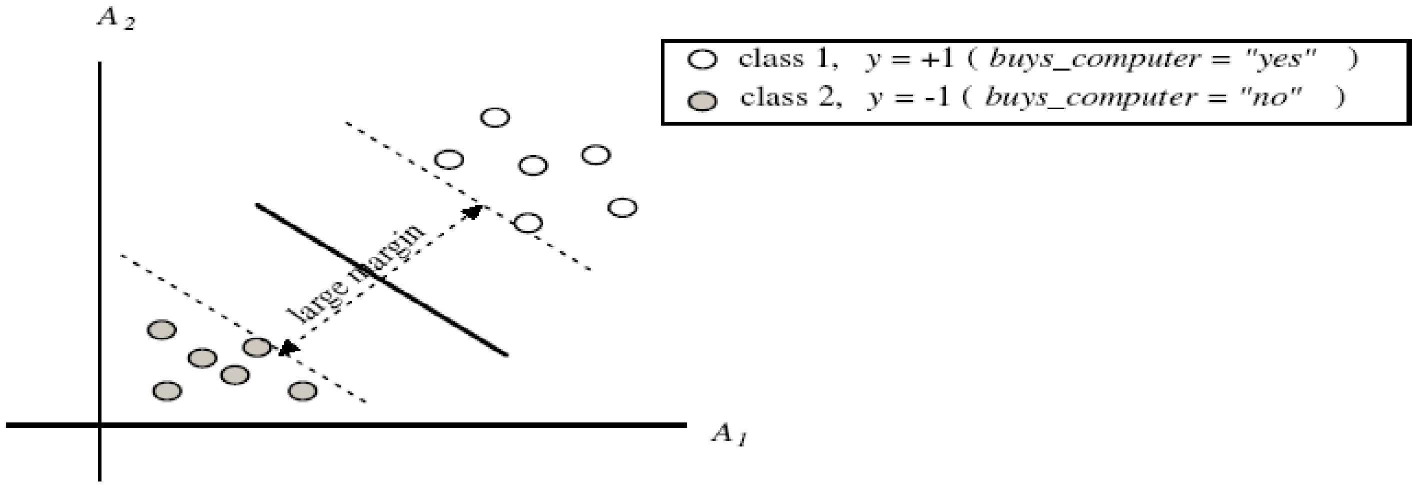
Data



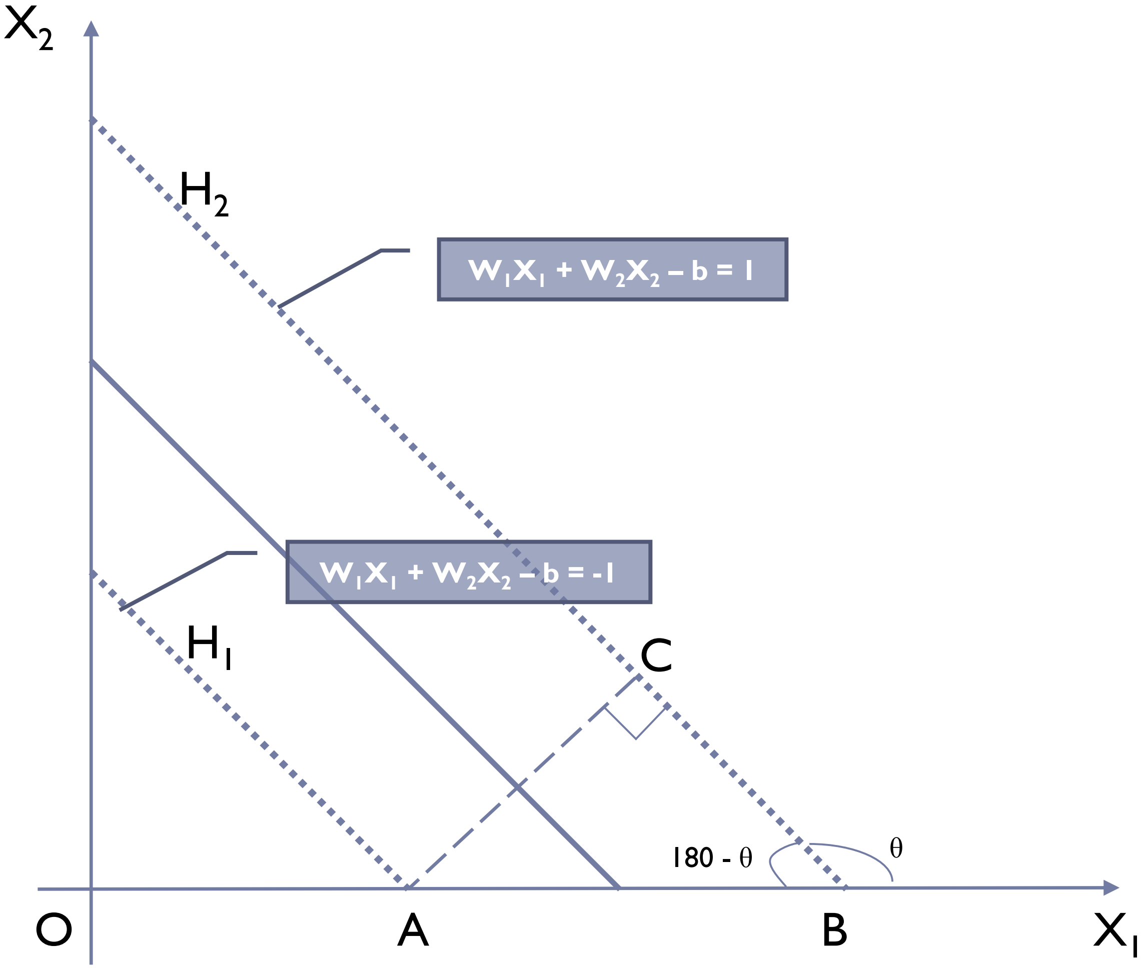
Linearly Separable


A 2
○ class 1, y = +1 (buys_computer = "yes")
● class 2, y = -1 (buys_computer = "no")
small margin
A 1

Support Vector Hyper plane


A 2
○ class 1, y = +1 (buys_computer = "yes")
● class 2, y = -1 (buys_computer = "no")
large margin
A 1

Distance between parallel hyper planes



Linear SVM Mathematically

- Goal: **1) Correctly classify all training data**

$$\left.\begin{aligned} wx_i + b &\geq 1 \quad \textit{if } y_i = +1 \\ wx_i + b &\leq 1 \quad \textit{if } y_i = -1 \end{aligned}\right\}$$

$$y_i(wx_i + b) \geq 1 \quad \text{for all i}$$

2) Maximize the Margin $M = \frac{2}{|w|}$

same as minimize $\frac{1}{2} w^t w$

- **We can formulate a Quadratic Optimization Problem and solve for w and b**

- Minimize $\Phi(w) = \frac{1}{2} w^t w$

  subject to $y_i(wx_i + b) \geq 1 \qquad \forall i$

Solving the Optimization Problem

Find $\mathbf{w}$ and b such that
$\boldsymbol{\Phi}(\mathbf{w}) = ½\, \mathbf{w}^T\mathbf{w}$ is minimized;
and for all $\{(\mathbf{x_i}, y_i)\}$: $y_i\,(\mathbf{w}^T\mathbf{x_i} + b) \geq 1$

- **Need to optimize a *quadratic* function subject to *linear* constraints.**
- **Quadratic optimization problems are a well-known class of mathematical programming problems, and many (rather intricate) algorithms exist for solving them.**
- **The solution involves constructing a *dual problem* where a *Lagrange multiplier* α_i is associated with every constraint in the primary problem:**

Find $\alpha_1 \ldots \alpha_N$ such that
$\mathbf{Q}(\boldsymbol{\alpha}) = \Sigma\alpha_i - ½\Sigma\Sigma\alpha_i\alpha_j y_i y_j \mathbf{x_i}^T\mathbf{x_j}$ is maximized and
(1) $\Sigma\alpha_i y_i = 0$
(2) $\alpha_i \geq 0$ for all α_i

The Optimization Problem Solution

- The solution has the form:

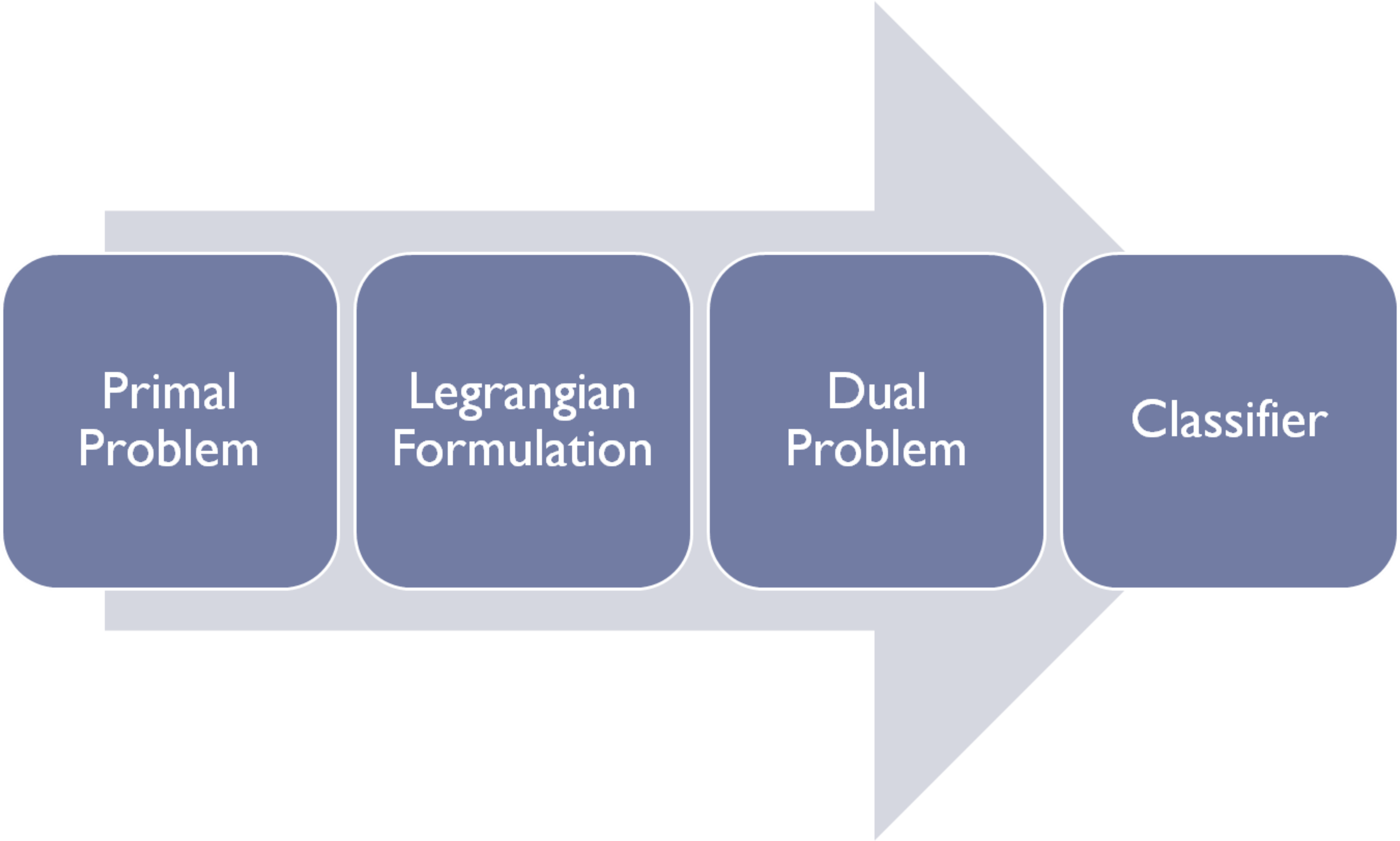
$$\mathbf{w} = \Sigma \alpha_i y_i \mathbf{x_i} \qquad b = y_k - \mathbf{w^T x_k} \text{ for any } \mathbf{x_k} \text{ such that } \alpha_k \neq 0$$

- Each non-zero α_i indicates that corresponding $\mathbf{x_i}$ is a support vector.
- Then the classifying function will have the form:

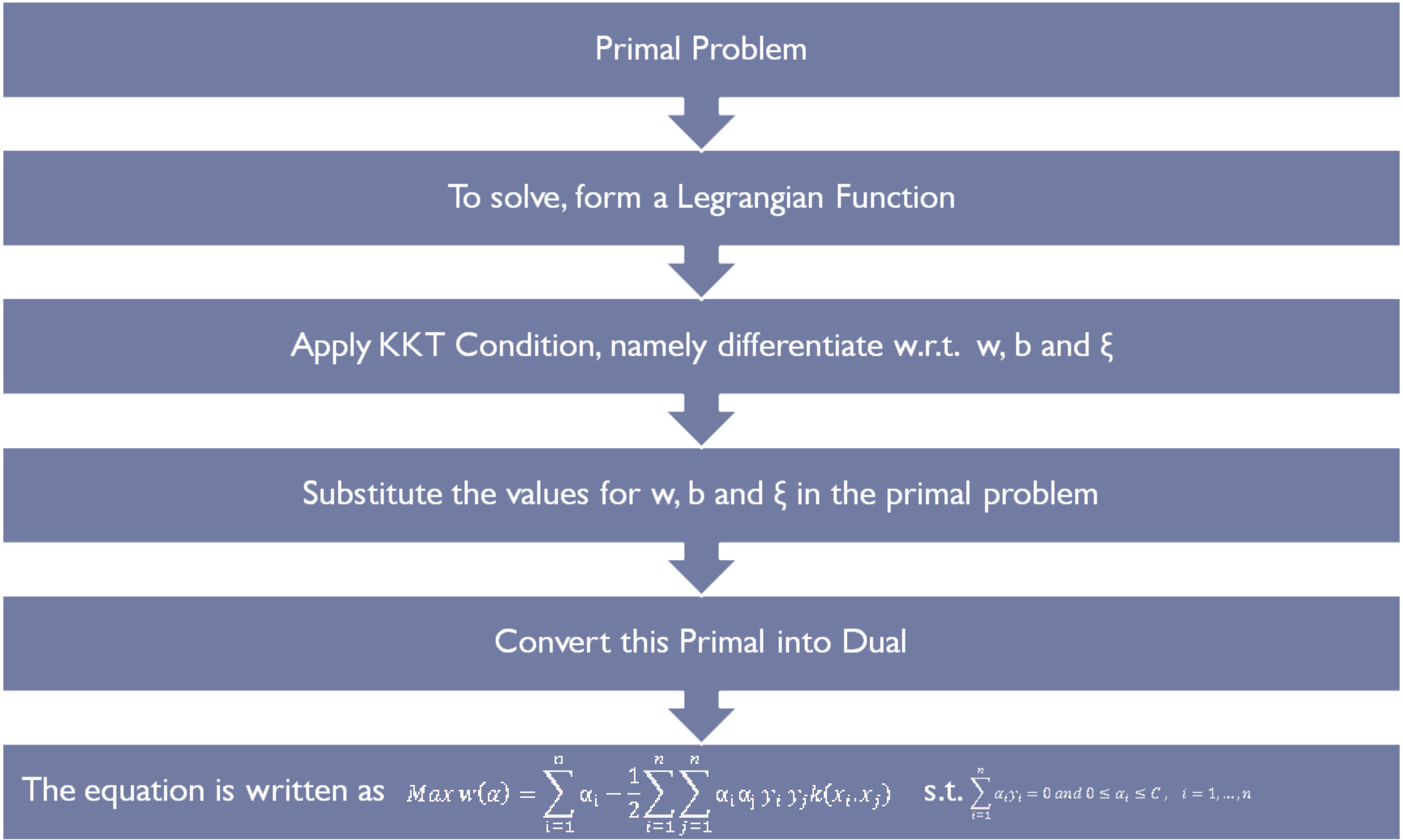
$$f(\mathbf{x}) = \Sigma \alpha_i y_i \mathbf{x_i^T x} + b$$

- Notice that it relies on an *inner product* between the test point $\mathbf{x}$ 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 $\mathbf{x_i^T x_j}$ between all pairs of training points.

SVM


Primal Problem
Legrangian Formulation
Dual Problem
Classifier

SVM



Primal Problem

To solve, form a Legrangian Function

Apply KKT Condition, namely differentiate w.r.t. w, b and ξ

Substitute the values for w, b and ξ in the primal problem

Convert this Primal into Dual

The equation is written as $Max\, w(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j k(x_i.x_j)$ s.t. $\sum_{i=1}^{n} \alpha_i y_i = 0 \text{ and } 0 \le \alpha_i \le C, \quad i = 1, \dots, n$

Cont …

SVM

- The SSVM can be mathematically expressed in **Primal Form** as

  Minimize $\frac{1}{2}||w||^2$, subject to the constraints,

$$(w'.x_i + b) \geq 1 \;\; \forall x_i \in C_1 \; and \; (w'.x_i + b) \leq -1 \;\; \forall x_i \in C_2$$

  - The 2 constraints can be combined into single constraint

$$y_i(w'.x_i + b) \geq 1 \; \forall x_i \in C_1 \cup C_2$$

Where, $y_i \in \{-1, +1\}$ are the class labels

SVM

- The **Lagrangian Formulation** is

$$L(w,b,\alpha)=\frac{1}{2}||\mathrm{w}||^{2}-\sum_{i=1}^{n}\alpha_i[(w'.x_i+b)y_i-1]$$

SVM

- Setting the derivative of L with respect to w and b to zero, we have

$$\nabla_w L(w, b, \alpha) = w - \sum_{i=1}^{n} \alpha_i y_i x_i = 0$$

- This implies that

$$w = \sum_{i=1}^{n} \alpha_i y_i x_i$$

- As for the derivative with respect to b, we obtain

$$\frac{\partial}{\partial b} L(w, b, \alpha) = \sum_{i=1}^{n} \alpha_i y_i = 0$$

Cont …

SVM

- Substituting in the previous equation, we get

$$L(w,b,\alpha)=\sum_{i=1}^{n}\alpha_i-\frac{1}{2}\sum_{i,j=1}^{n}y_iy_j\alpha_i\alpha_jx_i^Tx_j-b\sum_{i=1}^{n}\alpha_iy_j$$

- The last term becomes zero. So

$$L(w,b,\alpha)=\sum_{i=1}^{n}\alpha_i-\frac{1}{2}\sum_{i,j=1}^{n}y_iy_j\alpha_i\alpha_jx_i^Tx_j$$

SVM

- The **Dual Form** for is

$$Max\ L(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} y_i y_j \alpha_i \alpha_j x_i^T x_j$$

$$\alpha_i \geq 0, \sum \alpha_i y_i = 0\ \forall i$$

- OR

$$Min\ L(\alpha) = \frac{1}{2} \sum_{i,j=1}^{n} y_i y_j \alpha_i \alpha_j x_i^T x_j - \sum_{i=1}^{n} \alpha_i$$

$$\alpha_i \geq 0, \sum \alpha_i y_i = 0\ \forall i$$

Cont …

SVM: Finding the Value of b

- Support Vectors α_i not equal to 0
- In the Training set
  - Let SV_R belong to C_1
  - Let SV_S belong to C_2

$$w^T x_{SV_R} + \ b = 1$$

$$w^T x_{SV_s} + \ b = -1$$

$$2b = -\left[w^T x_{SV_s} + w^T x_{SV_R}\right]$$

$$b = -\frac{1}{2}\, w^T\left[x_{SV_s} + x_{SV_R}\right]$$

Cont …

Problem

- If the optimal weight vector $\hat{w}$= [1.2,0.2,0.9]’, and the two support vectors are x'_r=[1,0,-1] and x'_s=[1,1,1], estimate the bias term ‘b’ in the OSH hyperplane equation $w'x+b=0$, and obtain the classifier

- Use the above classifier to classify the new data points [-2,-1,3] & [2,4,-1]. Are any of these new data points support vectors?

Cont …

SVM: Finding w

$$W = \left\{ \begin{matrix} w_1 \\ w_2 \\ \dots \\ w_n \end{matrix} \right\} = \sum_{i=1}^{n} \alpha_i y_i x_i$$

$$X = \left\{ \begin{matrix} x_1 \\ x_2 \\ \dots \\ x_n \end{matrix} \right\}$$

Cont …

SVM

- Now we know w and b.
- The **classifier** can be defined as

$$f(x) = sign\left[\sum_{i=1}^{|SV|} \alpha_i y_i x_i^T x + b\right]$$

Multi – Class SVM

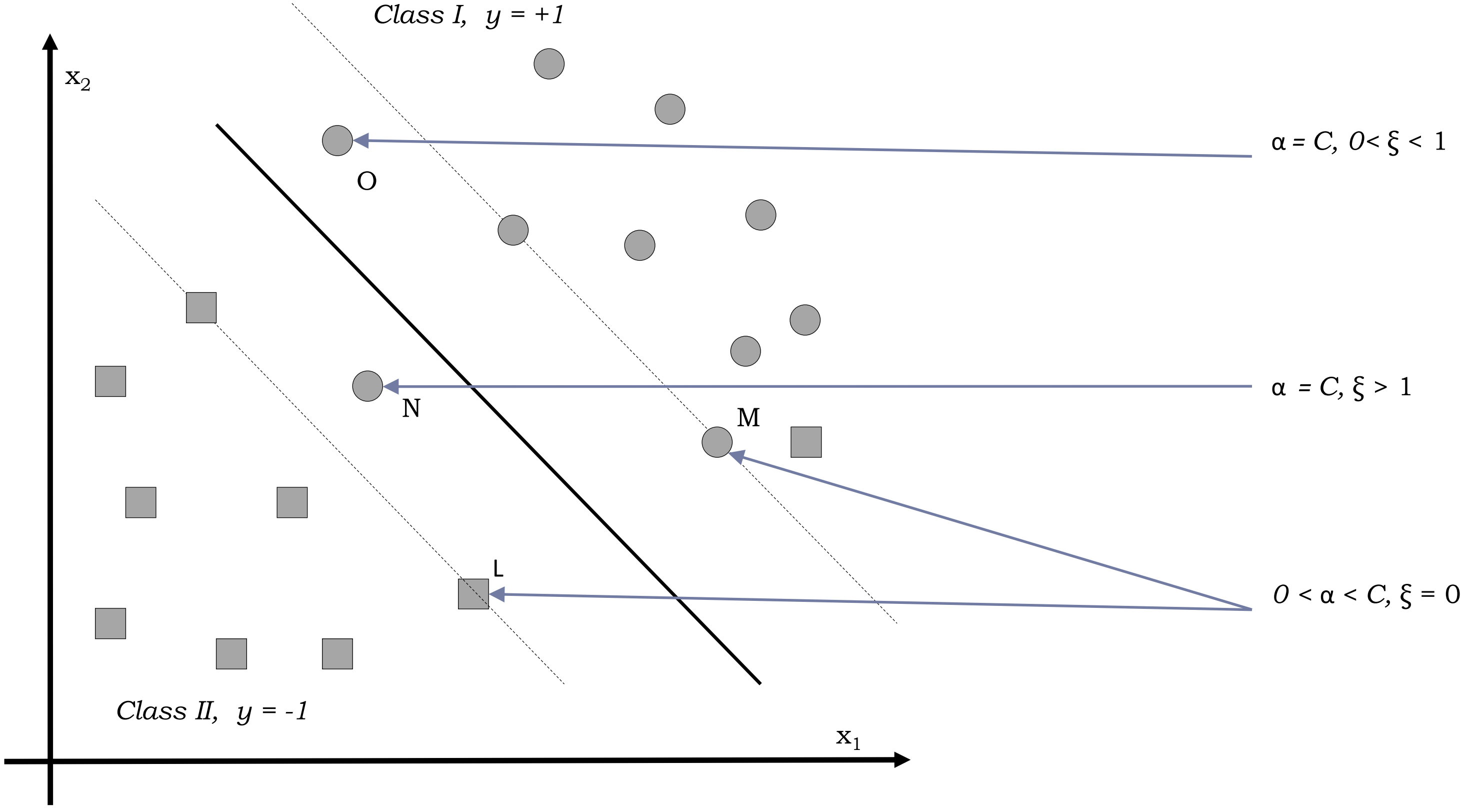
- If there are n classes then ${}^{n}C_2$ combinations are possible
- This means that we can have ${}^{n}C_2$ classifiers
- For a test data, solve it in each classifier equation to find where the new point falls
- Based on the result move the point to the appropriate group

Linearly Non Separable

- There are many instances when linear separating hyperplanes can be a good solution even when data are overlapped
- Sometimes data are not strictly linearly separable because of
  - Faulty Data Collection Mechanism
  - Various Types of Errors
  - Data Outliers etc.
- If data points in training set violating linear separability is small, we can still use a Linear model by penalizing the violating points using slack variables

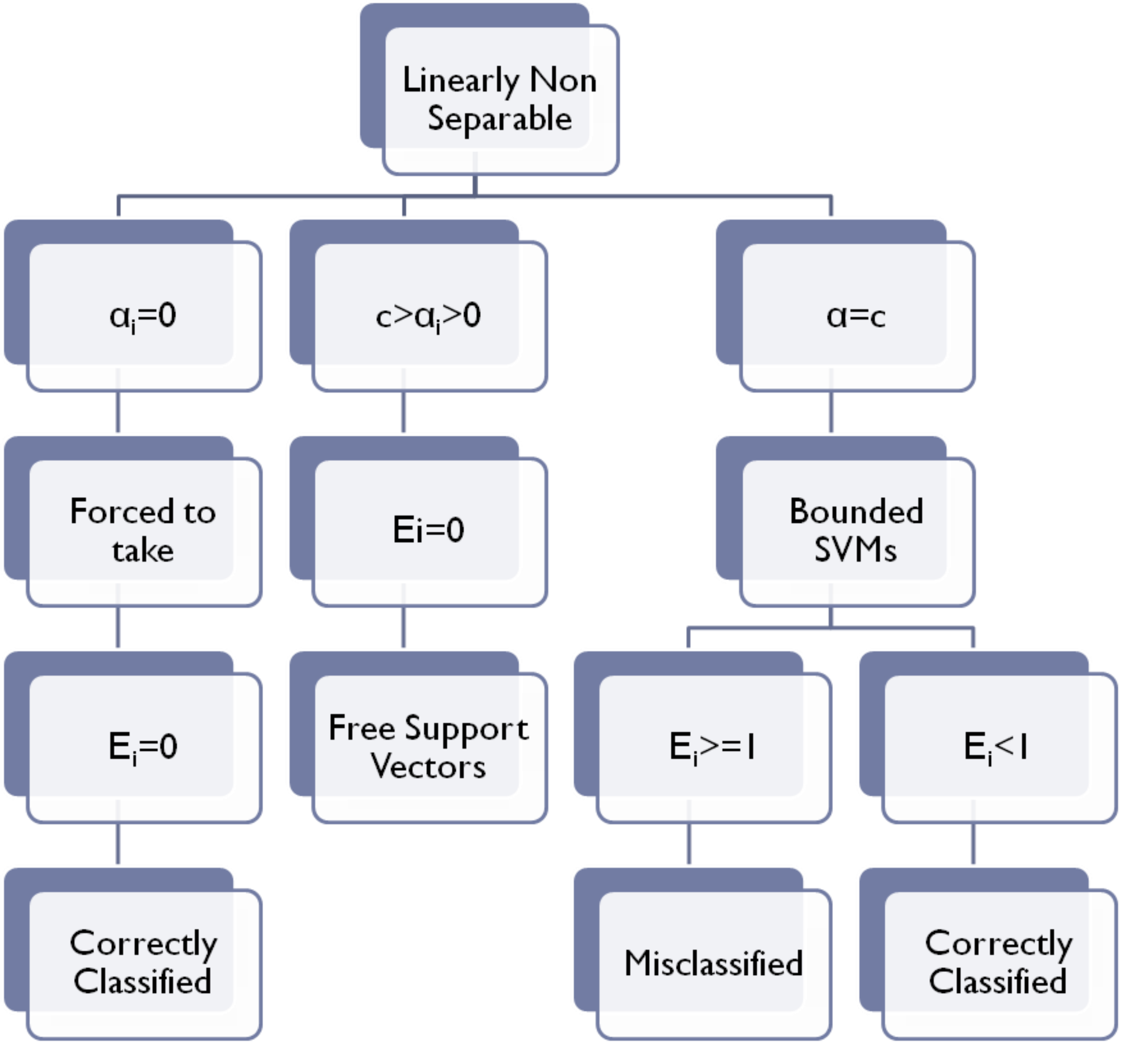
Cont …

Linearly Non-Separable SVM



Cont …

Linearly Non-Separable SVM



Linearly Non Separable

α_i=0

Forced to take

E_i=0

Correctly Classified

c>α_i>0

Ei=0

Free Support Vectors

α=c

Bounded SVMs

E_i>=1

Misclassified

E_i<1

Correctly Classified

Cont …

Linearly Non Separable

- Let $(X_i y_i)$, where $x_i \in R^n$ be the training data
- Let $\xi_i \geq 0$ be the slack variables, ideally close to zero
- The objective is to simultaneously minimize $||w||^2$ & $\sum_{i=1}^{n} \xi_i$
- Hence, it can be represented as

$$J(w, b, \xi): \text{Min}_{w,b} \frac{1}{2} ||w||^2 + C \sum_{i=1}^{n} \xi_i$$

w.Xi – b ≥ +1 - ξi   for yi =+1
w.Xi – b ≤ -1+ ξi   for yi = -1

ξi ≥ 0 , for all i

Where, $(w'.x_i + b)y_i \geq 1 - \xi_i \ and \ \xi_i \geq 0 \ \forall i$

C is a constant to compromise between the size of the margin (Large) and the magnitude of ξ's (small)

Cont …

Linearly Non Separable

- As before we take the Lagrangian to find the dual form

$$Q(w,b,\xi,\Lambda,\beta) = \frac{1}{2}||\mathrm{w}||^2 + C\sum_{i=1}^{n}\xi_i - \sum_{i=1}^{n}\lambda_i[(w'.x_i + b_i)y_i - 1 + \xi_i] - \sum_{i=1}^{n}\beta_i\xi_i$$

- The negative sign is used because we intent to maximize w.r.t. λ_i, β_i and minimize w.r.t. $w, \xi_i \ and \ b$

- Differentiate w.r.t $w, \xi_i \ and \ b$ .Also impose stationarity and we get,

$$\sum \alpha_i y_i x_i \ , \alpha_i + \beta_i = C \qquad \text{and} \qquad \sum \alpha_i y_i = 0$$

- Substituing in the previous equation and simplifying we get

$$Max\ L(\alpha) = -\frac{1}{2}\sum_{i,j=1}^{n}\alpha_i\alpha_j y_i y_j (x_i.x_j) + \sum_{i=1}^{n}\alpha_i \qquad \text{with the constraints,}$$

$$\sum \alpha_i y_i = 0, and\ 0 \le \alpha_i \le C$$

Cont …

Linearly Non Separable

- The bias term b is found by averaging over those sample data points for which

  $0 \leq \alpha_i \leq C$

- The Dual Classifier is

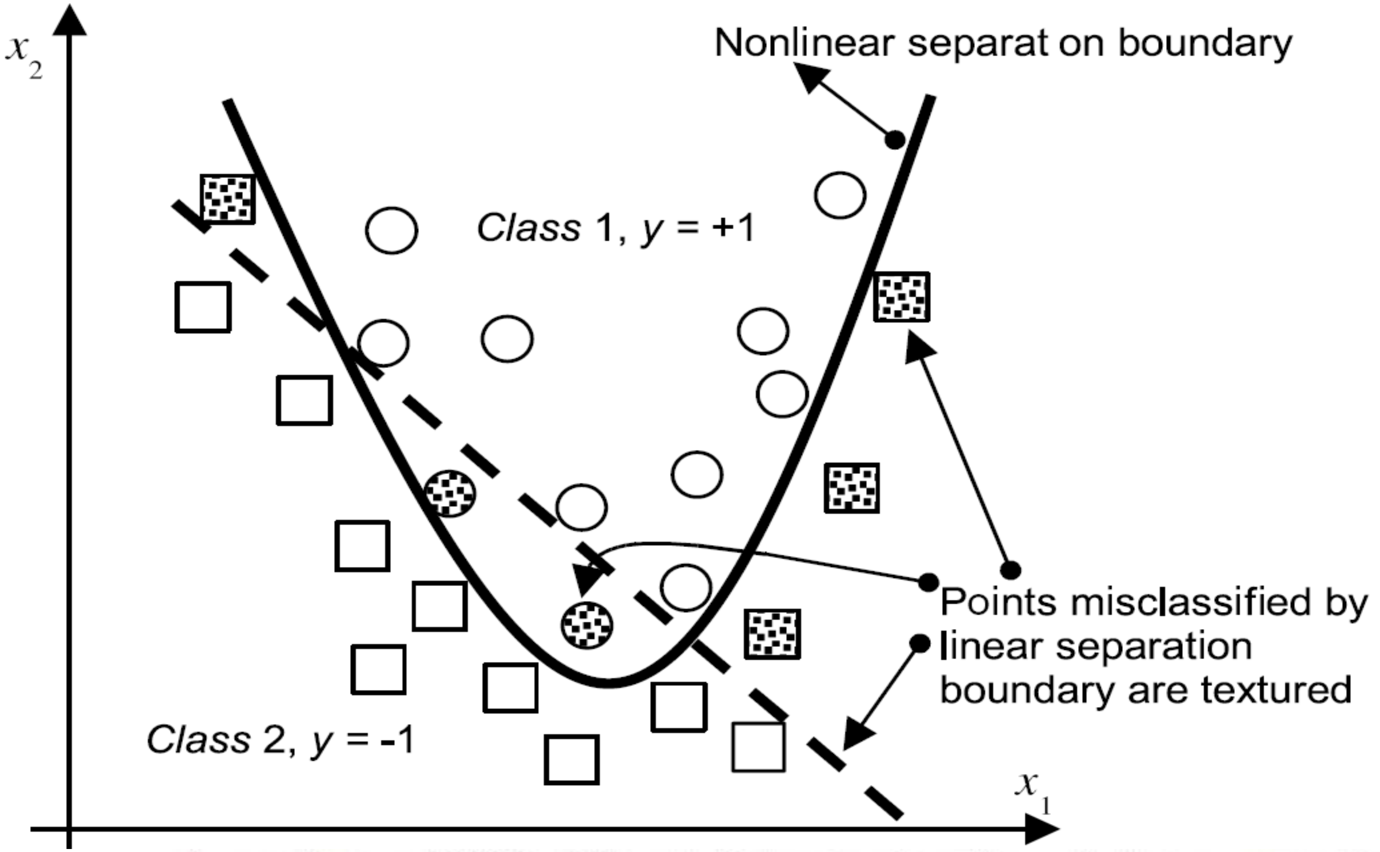
$$D(x) = Sign\left(\sum_{SV's} \hat{\alpha}_i\, y_i[(x_i, x) + \hat{b}]\right)$$

Where,

$$\hat{b} = \frac{1}{2} \sum_{MSV's} \hat{\alpha}_i\, y_i[(x_r, x_i) + (x_s, x_i)]$$

- The Primal and Dual solutions are connected as $w = \sum_{i=1}^{n} \alpha_i y_i x_i$

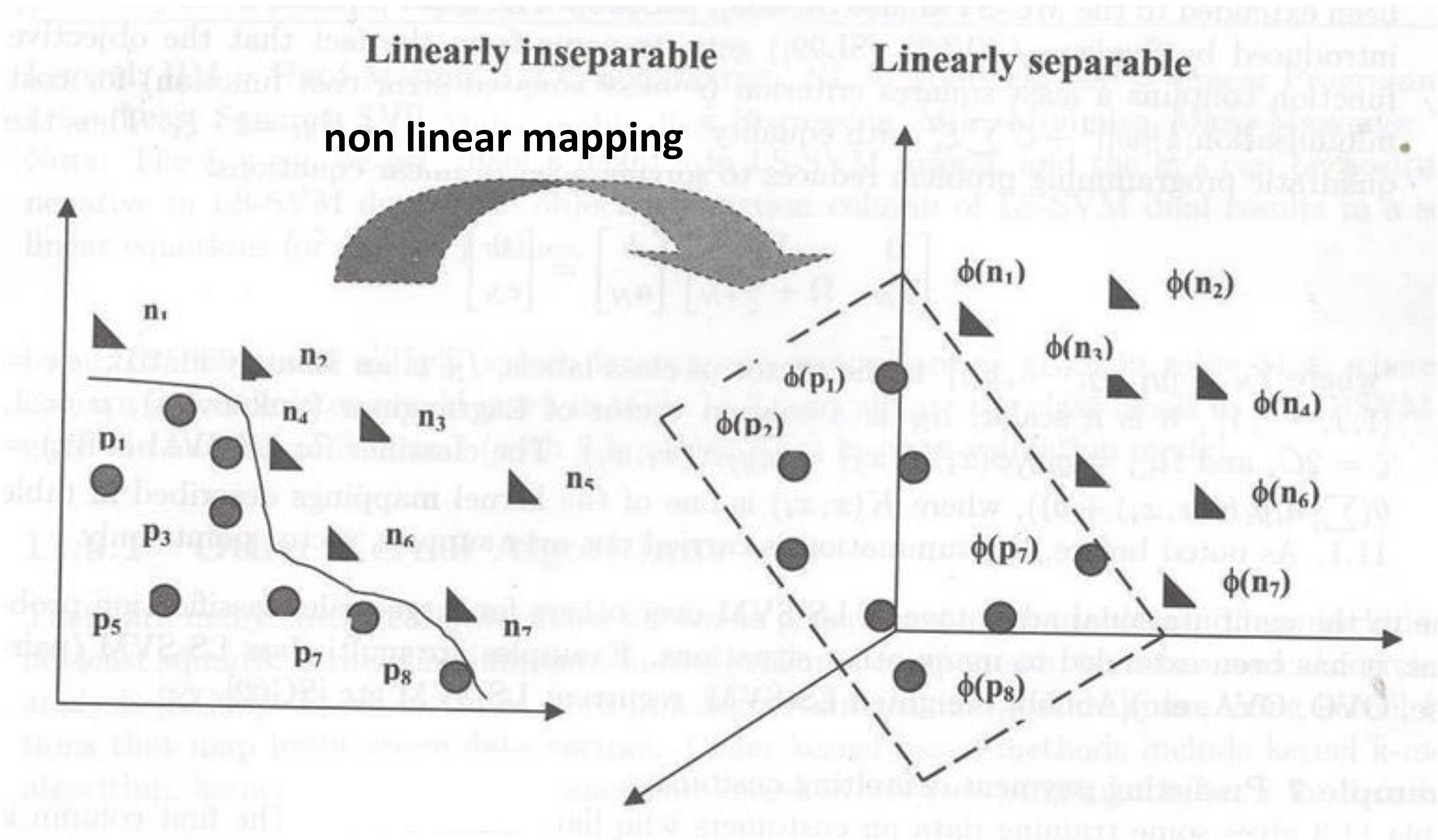
Non-Linearly Separable


x_2
Nonlinear separat on boundary
Class 1, y = +1
Points misclassified by linear separation boundary are textured
Class 2, y = -1
x_1

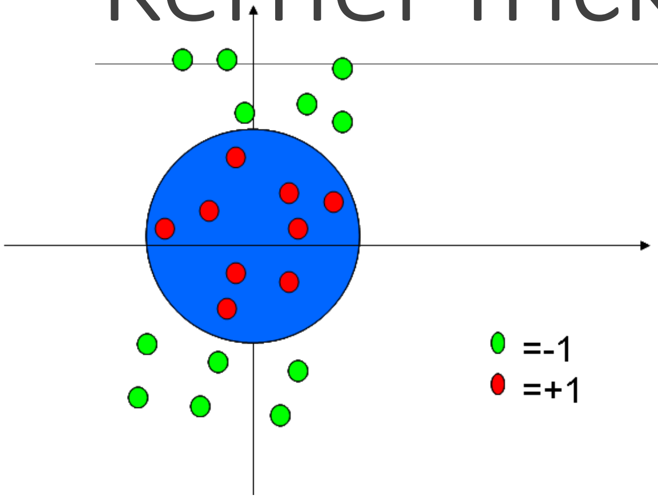
A nonlinear SVM without data overlapping. A true separation is a quadratic curve. The nonlinear separation line (solid), the linear one (dashed) and data points misclassified by the linear separation line (the textured training data points) are shown. There are 4 misclassified negative data and 2 misclassified positive ones. SVs are not shown.

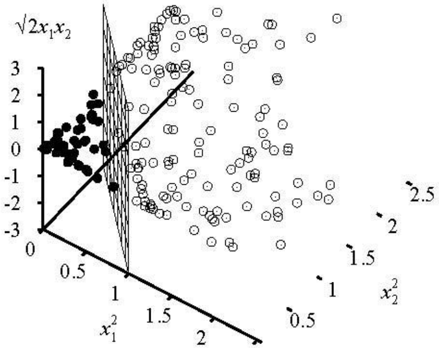
Cont …

Non Linearly Separable SVM



Kernel Trick





Data points are linearly separable in the space $(x_1^2, x_2^2, \sqrt{2}x_1x_2)$

We want to maximize $\sum_i \alpha_i - \frac{1}{2}\sum_{i,j} y_i y_j \alpha_i \alpha_j \left\langle F(\mathbf{x}_i)\cdot F(\mathbf{x}_j)\right\rangle$

Define $K(\mathbf{x}_i, \mathbf{x}_j) = \left\langle F(\mathbf{x}_i)\cdot F(\mathbf{x}_j)\right\rangle$

Cool thing : K is often easy to compute directly! Here,

$K(\mathbf{x}_i, \mathbf{x}_j) = \left\langle \mathbf{x}_i \cdot \mathbf{x}_j \right\rangle^2$

Other Kernels

The polynomial kernel

$K(x_i,x_j) = (x_i \bullet x_j + 1)^p$, where p is a tunable parameter.

Evaluating K only require one addition and one exponentiation more than the original dot product.

Gaussian kernels (also called radius basis functions)

$K(x_i,x_j) = \exp(||x_i-x_j\ ||^2/2\sigma^2)$

Non Linearly Separable SVM

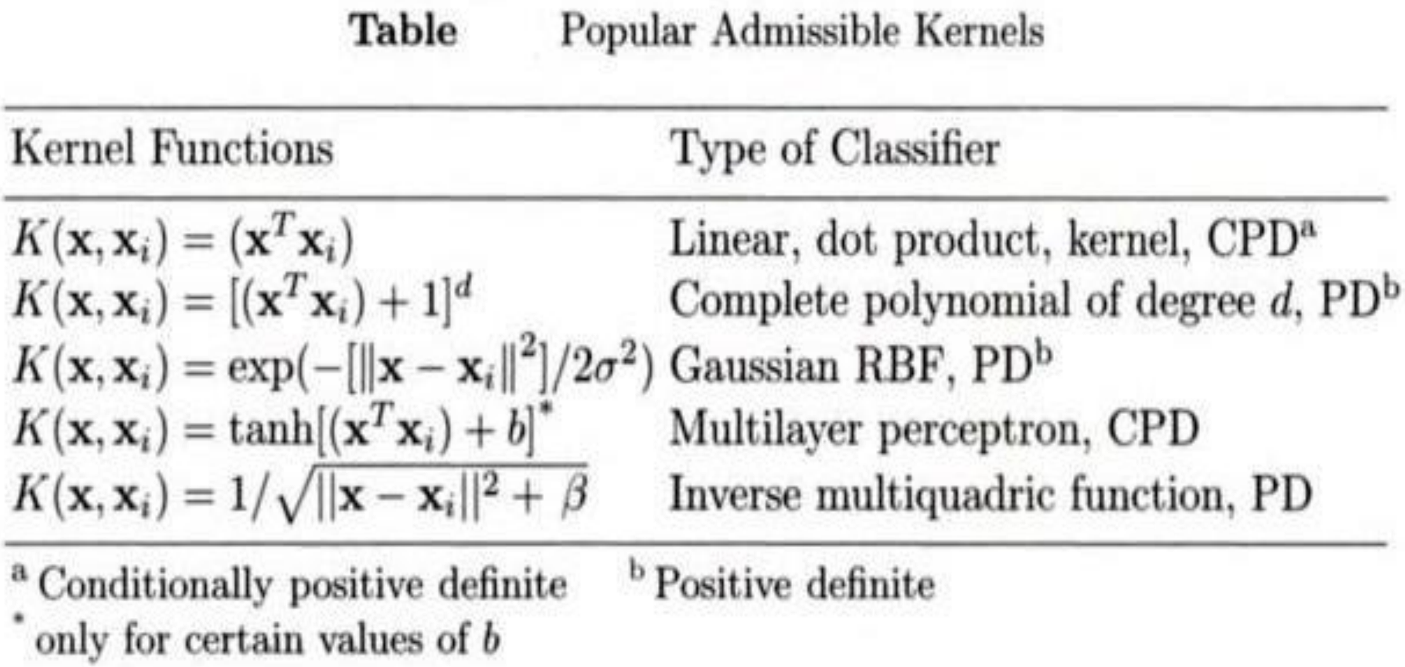


Table Popular Admissible Kernels

Kernel Functions	Type of Classifier
$K(\mathbf{x}, \mathbf{x}_i) = (\mathbf{x}^T \mathbf{x}_i)$	Linear, dot product, kernel, CPD[a]
$K(\mathbf{x}, \mathbf{x}_i) = [(\mathbf{x}^T \mathbf{x}_i) + 1]^d$	Complete polynomial of degree *d*, PD[b]
$K(\mathbf{x}, \mathbf{x}_i) = \exp(-[\lVert \mathbf{x} - \mathbf{x}_i \rVert^2]/2\sigma^2)$	Gaussian RBF, PD[b]
$K(\mathbf{x}, \mathbf{x}_i) = \tanh[(\mathbf{x}^T \mathbf{x}_i) + b]^*$	Multilayer perceptron, CPD
$K(\mathbf{x}, \mathbf{x}_i) = 1/\sqrt{\lVert \mathbf{x} - \mathbf{x}_i \rVert^2 + \beta}$	Inverse multiquadric function, PD

[a] Conditionally positive definite [b] Positive definite

* only for certain values of b

Non Linearly Separable SVM

- Φ(X_i) is simply the non linear mapping function applied to transform the training inputs

$$L_d(\boldsymbol{\alpha}) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,\,j=1}^{n} \alpha_i \alpha_j y_i y_j \boldsymbol{\Phi}_i^T \boldsymbol{\Phi}_j,$$

$$\max\ L_d(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} y_i y_j \alpha_i \alpha_j K(\mathbf{x}_i, \mathbf{x}_j)$$

$$\text{s.t. } \alpha_i \geq 0, \quad i = 1, \ldots, n \quad \text{and}$$

$$\sum_{i=1}^{n} \alpha_i y_i = 0.$$

$$0 \ \leq \alpha_i \leq C \ for\ i = 1 \ldots n$$

Cont …

Non Linearly Separable SVM

- If the Gaussian Radial Basis Function is used

$$\max L_d(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} y_i y_j \alpha_i \alpha_j \exp(-[\|\mathbf{x} - \mathbf{x}_i\|^2]/2\sigma^2)$$

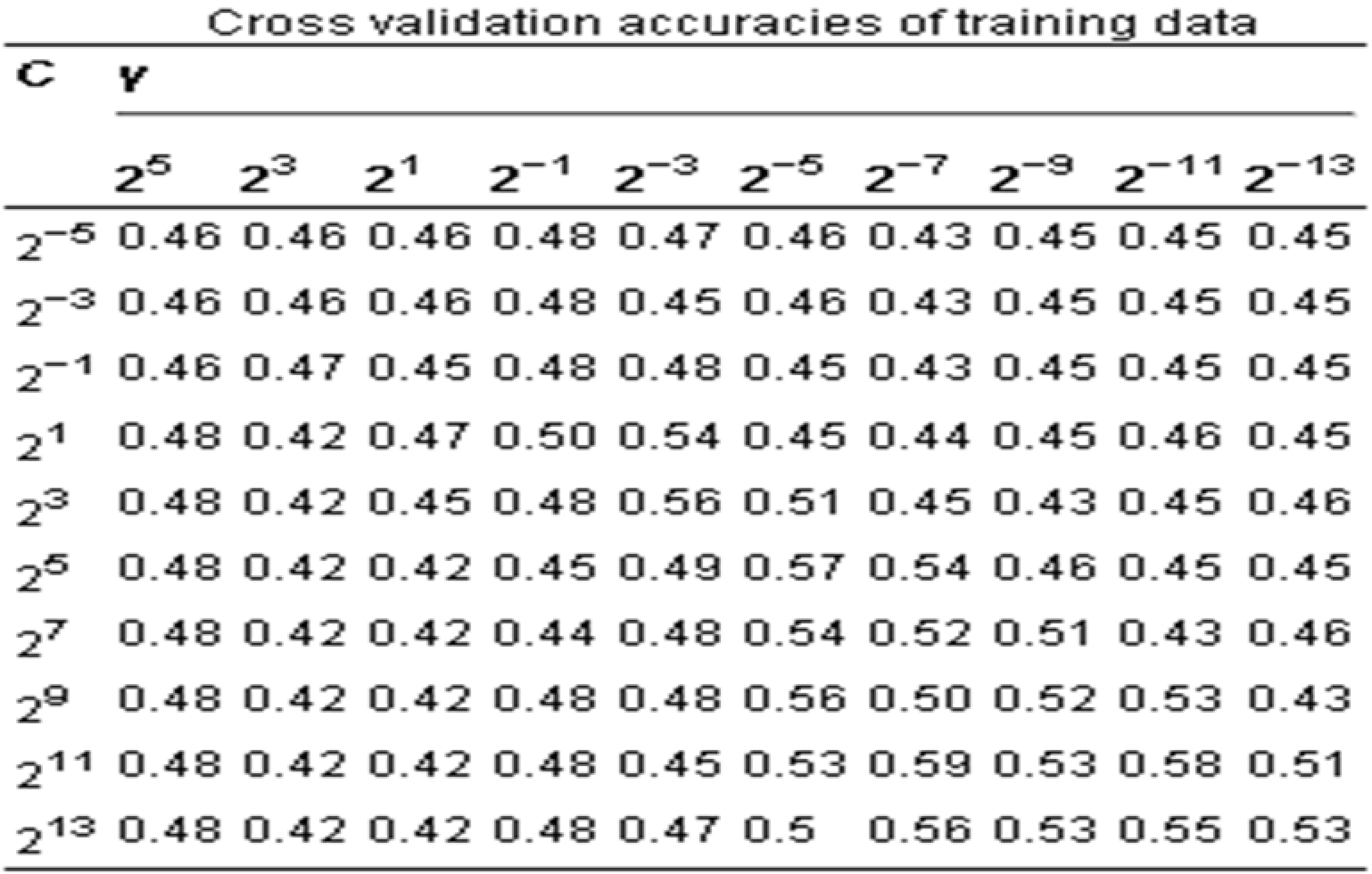
$$\text{s.t. } \alpha_i \geq 0, \quad i = 1, \ldots, n \quad \text{and}$$

$$\sum_{i=1}^{n} \alpha_i y_i = 0.$$

$$0 \leq \alpha_i \leq C \; for \; i = 1 \ldots n$$

Cross Validation Accuracies of Training Data

Cont …



Cross validation accuracies of training data

C	γ									
	2^{5}	2^{3}	2^{1}	2^{-1}	2^{-3}	2^{-5}	2^{-7}	2^{-9}	2^{-11}	2^{-13}
2^{-5}	0.46	0.46	0.46	0.48	0.47	0.46	0.43	0.45	0.45	0.45
2^{-3}	0.46	0.46	0.46	0.48	0.45	0.46	0.43	0.45	0.45	0.45
2^{-1}	0.46	0.47	0.45	0.48	0.48	0.45	0.43	0.45	0.45	0.45
2^{1}	0.48	0.42	0.47	0.50	0.54	0.45	0.44	0.45	0.46	0.45
2^{3}	0.48	0.42	0.45	0.48	0.56	0.51	0.45	0.43	0.45	0.46
2^{5}	0.48	0.42	0.42	0.45	0.49	0.57	0.54	0.46	0.45	0.45
2^{7}	0.48	0.42	0.42	0.44	0.48	0.54	0.52	0.51	0.43	0.46
2^{9}	0.48	0.42	0.42	0.48	0.48	0.56	0.50	0.52	0.53	0.43
2^{11}	0.48	0.42	0.42	0.48	0.45	0.53	0.59	0.53	0.58	0.51
2^{13}	0.48	0.42	0.42	0.48	0.47	0.5	0.56	0.53	0.55	0.53

Cont …

Non Linearly Separable SVM

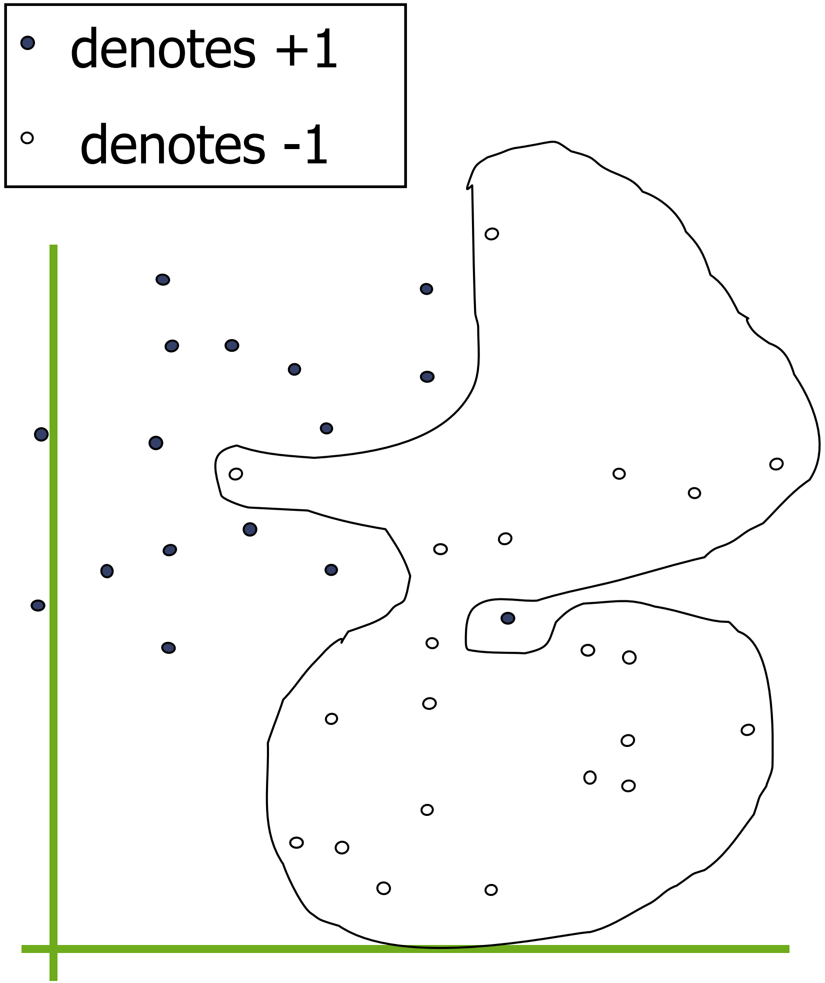
- $\mathrm{Sign}[\sum y_i \alpha_i \phi^T(x_i)\phi(x_i)+b]$ for all $1 \leq i \leq n$

- $\mathrm{Sign}[\sum y_i \alpha_i K(X,X_i)+b]$ for all $1 \leq i \leq n$

Summary

Type	Linearly Separable	Linearly Inseparable	Non Linearly Separable
Dual	$Max\ L(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i\alpha_j y_i y_j x_i x_j$	$Max\ L(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i\alpha_j y_i y_j x_i x_j$	$Max\ L(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i\alpha_j y_i y_j k(x_i . x_j)$
Constraint	$\sum_{i=1}^{n} \alpha_i y_i = 0\ and\ 0 \leq \alpha_i\ ,\quad i = 1, \ldots, n$	$\sum_{i=1}^{n} \alpha_i y_i = 0\ and\ 0 \leq \alpha_i \leq C\ ,\quad i = 1, \ldots, n$	$\sum_{i=1}^{n} \alpha_i y_i = 0\ and\ 0 \leq \alpha_i \leq C\ ,\quad i = 1, \ldots, n$
To Find ‘b’	$b = \frac{1}{N_{sv}} \sum_{sv \in S} \left\{ y_{sv} - \sum_{i \in S} \alpha_i y_i x_i . x_{sv} \right\}$	$b = \frac{1}{N_{sv}} \sum_{sv \in S} \left\{ y_{sv} - \sum_{i \in S} \alpha_i y_i x_i . x_{sv} \right\}$	$b = \frac{1}{N_{sv}} \sum_{sv \in S} \left\{ y_{sv} - \sum_{i \in S} \alpha_i y_i k(x_i . x_{sv}) \right\}$
Classifier	$f(y) = sign\ (\sum_{i=1}^{sv} \alpha_i y_i (x, x_{sv}) + b)$	$f(y) = sign\ (\sum_{i=1}^{sv} \alpha_i y_i (x, x_{sv}) + b)$	$f(y) = sign\ (\sum_{i=1}^{sv} \alpha_i y_i k(x, x_{sv}) + b)$

Dataset with noise



- **Hard Margin: So far we require all data points be classified correctly**

  - No training error

- **What if the training set is noisy?**

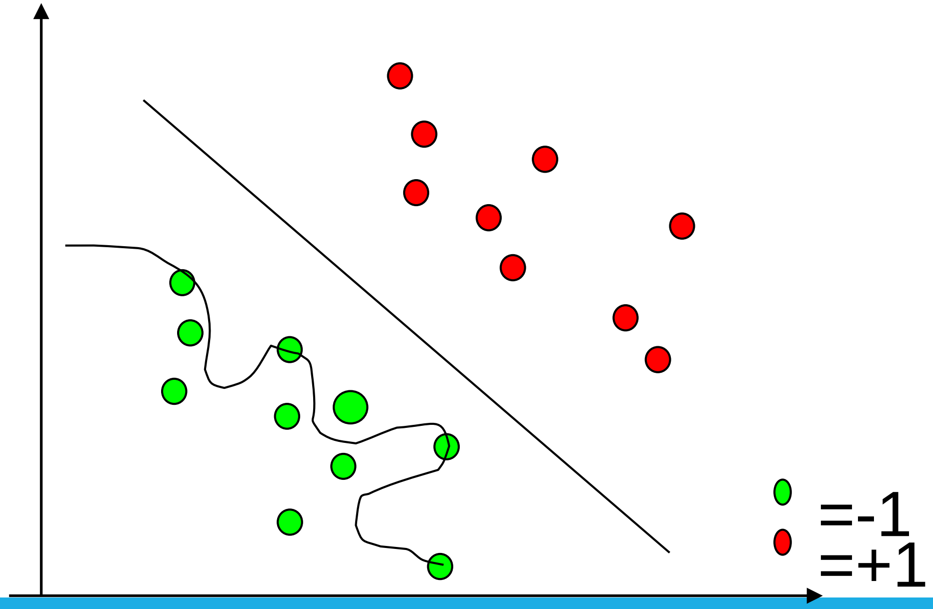
  - Solution 1: use very powerful kernels

OVERFITTING!

Overtraining/overfitting

A well known problem with machine learning methods is overtraining. This means that we have learned the training data very well, but we can not classify unseen examples correctly.

An example: A botanist really knowing trees. Everytime he sees a new tree, he claims it is not a tree.



Overtraining/overfitting 2

A measure of the risk of overtraining with SVM (there are also other measures).

It can be shown that: The portion, n, of unseen data that will be missclassified is bounded by:

$n \leq$ Number of support vectors / number of training examples

Ockham´s razor principle: Simpler system are better than more complex ones.

In SVM case: fewer support vectors mean a simpler representation of the hyperplane.

Example: Understanding a certain cancer if it can be described by one gene is easier than if we have to describe it with 5000.

Hard Margin v.s. Soft Margin

- **The old formulation:**

Find **w** and *b* such that

$\mathbf{\Phi(w)} = \frac{1}{2}\, \mathbf{w}^T\mathbf{w}$ is minimized and for all $\{(\mathbf{x_i}, y_i)\}$

$y_i\, (\mathbf{w}^T\mathbf{x_i} + \mathrm{b}) \geq 1$

- **The new formulation incorporating slack variables:**

Find **w** and *b* such that

$\mathbf{\Phi(w)} = \frac{1}{2}\, \mathbf{w}^T\mathbf{w} + C\Sigma\xi_i$ is minimized and for all $\{(\mathbf{x_i}, y_i)\}$

$y_i\, (\mathbf{w}^T\mathbf{x_i} + b) \geq 1 - \xi_i$ and $\xi_i \geq 0$ for all i

- **Parameter *C* can be viewed as a way to control overfitting.**

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

SVM Applications

SVM has been used successfully in many real-world problems

- text (and hypertext) categorization

- image classification,

- bioinformatics (Protein classification, Cancer classification)

- hand-written character recognition

Problems

- Give two practical examples that result in multi-class SVM with i) 3 classes ii) 4 classes
- Find the margin width for each of the following problems where the support vector hyper-plane equations are given.

  (i) $4x_1 + 3x_2 - 3.5 = 0$, $4x_1 + 3x_2 - 1.5 = 0$ (ii) $1.2x_1 + 1.6\,x_2 + 5 = \pm 1$

Thank you

ADDITIONAL SLIDES

Constrained Optimization Problem

Minimize $\| \mathbf{w} \| = \langle \mathbf{w} \cdot \mathbf{w} \rangle$ subject to $y_i(\langle \mathbf{x}_i \cdot \mathbf{w} \rangle + b) \geq 1$ for all i

Lagrangian method : maximize $\inf_{\mathbf{w}} L(\mathbf{w}, b, \alpha)$, where

$$L(\mathbf{w}, b, \alpha) = \frac{1}{2} \| \mathbf{w} \| - \sum_i \alpha_i \left[\left(y_i (\mathbf{x}_i \cdot \mathbf{w}) + b \right) - 1 \right]$$

At the extremum, the partial derivative of L with respect both $\mathbf{w}$ and b must be 0. Taking the derivative s, setting them to 0, substituting back into L, and simplifyin g yields :

Maximize $\displaystyle \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} y_i y_j \alpha_i \alpha_j \langle \mathbf{x}_i \cdot \mathbf{x}_j \rangle$

subject to $\displaystyle \sum_i y_i \alpha_i = 0$ and $\alpha_i \geq 0$

Quadratic Programming

Why is this reformulation a good thing?

The problem

$$\text{Maximize } \sum_i \alpha_i - \frac{1}{2}\sum_{i,j} y_i y_j \alpha_i \alpha_j \left\langle \mathbf{x}_i \cdot \mathbf{x}_j \right\rangle$$

$$\text{subject to} \sum_i y_i \alpha_i = 0 \text{ and } \alpha_i \geq 0$$

is an instance of what is called a positive, semi-definite programming problem

For a fixed real-number accuracy, can be solved in O(n log n) time = O(|D|^2 log |D|^2)

Linear SVMs: Overview

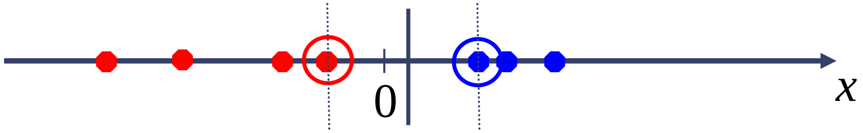
- **The classifier is a *separating hyperplane.***
- **Most "important" training points are support vectors; they define the hyperplane.**
- **Quadratic optimization algorithms can identify which training points $\mathbf{x}_i$ are support vectors with non-zero Lagrangian multipliers α_i.**
- **Both in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

Find $\alpha_1 \ldots \alpha_N$ such that
$Q(\alpha) = \Sigma\alpha_i - ½\Sigma\Sigma\alpha_i\alpha_j y_i y_j \mathbf{x}_i^T\mathbf{x}_j$ is maximized and
(1) $\Sigma\alpha_i y_i = 0$
(2) $0 \leq \alpha_i \leq C$ for all α_i

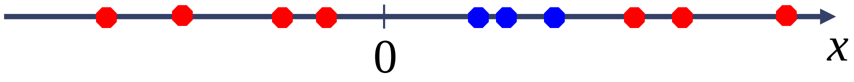
$$f(\mathbf{x}) = \Sigma\alpha_i y_i \mathbf{x}_i^T\mathbf{x} + b$$

Non-linear SVMs

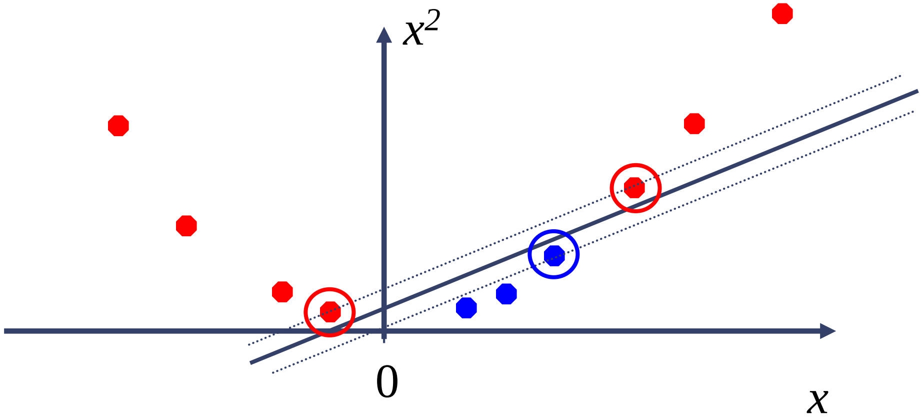
- Datasets that are linearly separable with some noise work out great:



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

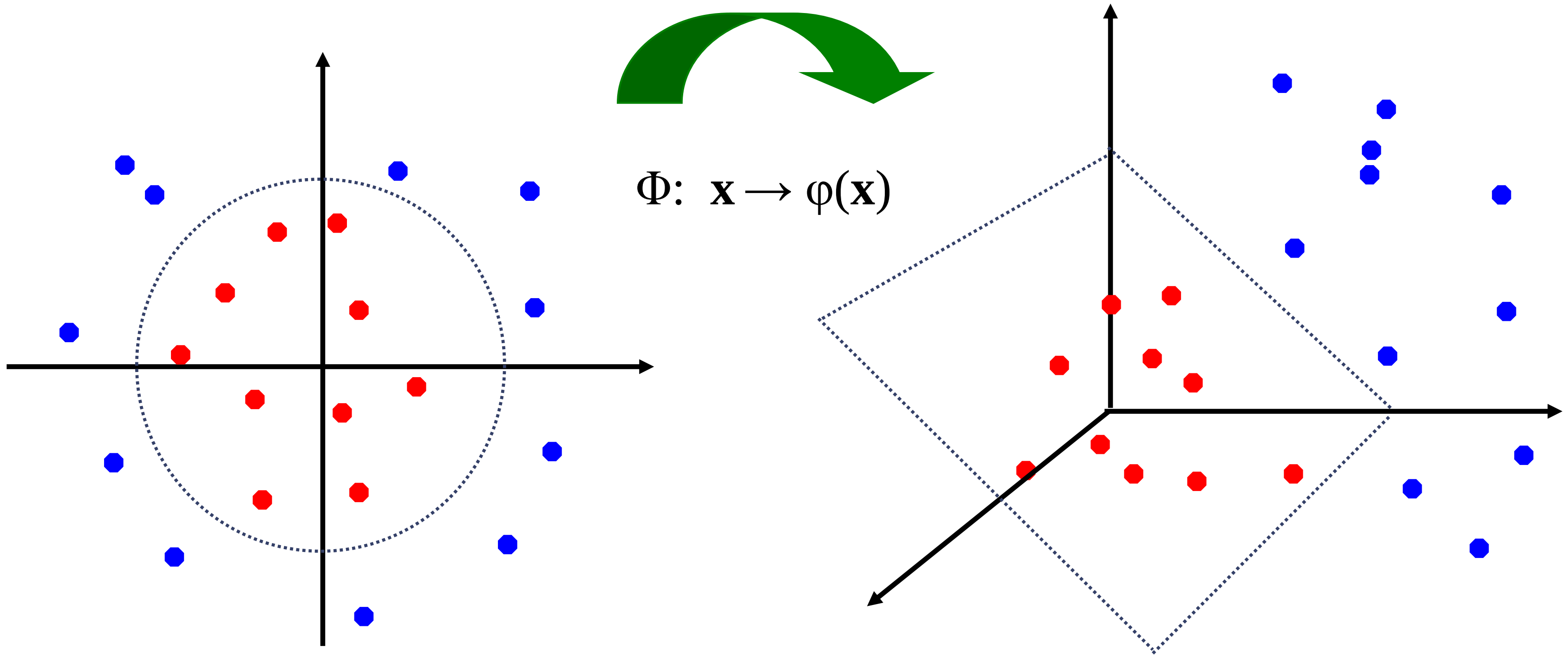


- How about… mapping data to a higher-dimensional space:



Non-linear SVMs: Feature spaces

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



$\Phi: \mathbf{x} \rightarrow \varphi(\mathbf{x})$

The “Kernel Trick”

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

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

- **A *kernel function* is some function that corresponds to an inner product in some expanded feature space.**
- **Example:**

2-dimensional vectors $\mathbf{x}=[x_1\ \ x_2]$; let $K(\mathbf{x}_i,\mathbf{x}_j)=(1 + \mathbf{x}_i^T\mathbf{x}_j)^2$,

Need to show that $K(\mathbf{x}_i,\mathbf{x}_j)= \varphi(\mathbf{x}_i)^T\varphi(\mathbf{x}_j)$:

$$
\begin{aligned}
K(\mathbf{x}_i,\mathbf{x}_j)&=(1 + \mathbf{x}_i^T\mathbf{x}_j)^2,\\
&= 1+ x_{i1}^2x_{j1}^2 + 2\, x_{i1}x_{j1}\, x_{i2}x_{j2}+ x_{i2}^2x_{j2}^2 + 2x_{i1}x_{j1} + 2x_{i2}x_{j2}\\
&= [1\ \ x_{i1}^2\ \ \sqrt{2}\, x_{i1}x_{i2}\ \ x_{i2}^2\ \ \sqrt{2}x_{i1}\ \ \sqrt{2}x_{i2}]^T\, [1\ \ x_{j1}^2\ \ \sqrt{2}\, x_{j1}x_{j2}\ \ x_{j2}^2\ \ \sqrt{2}x_{j1}\ \ \sqrt{2}x_{j2}]\\
&= \varphi(\mathbf{x}_i)^T\varphi(\mathbf{x}_j), \quad \text{where } \varphi(\mathbf{x}) = [1\ \ x_1^2\ \ \sqrt{2}\, x_1x_2\ \ x_2^2\ \ \sqrt{2}x_1\ \ \sqrt{2}x_2]
\end{aligned}
$$

What Functions are Kernels?

- **For some functions $K(x_i,x_j)$ checking that**

  $K(x_i,x_j)= \varphi(x_i)^T\varphi(x_j)$ 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 Kernel Functions

- Linear: $K(\mathbf{x_i},\mathbf{x_j}) = \mathbf{x_i}^T\mathbf{x_j}$

- Polynomial of power *p*: $K(\mathbf{x_i},\mathbf{x_j}) = (1+ \mathbf{x_i}^T\mathbf{x_j})^p$

- Gaussian (radial-basis function network):

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

- Sigmoid: $K(\mathbf{x_i},\mathbf{x_j}) = \tanh(\beta_0\mathbf{x_i}^T\mathbf{x_j} + \beta_1)$

Non-linear SVMs Mathematically

- **Dual problem formulation:**

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

- **The solution is:**

$$f(\mathrm{x}) = \Sigma\alpha_i y_i K(\mathrm{x}_i, \mathrm{x}_j) + b$$

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

Nonlinear 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.