Support Vector Machines

IF-3170 Inteligensi Artifisial

Teknik Informatika ITB

Modul: Supervised Learning

01 SVM: What & Why?

IF3170 - Inteligensi Artifisial

Dr. Fariska Z. Ruskanda, S.T., M.T.
(fariska@informatika.org)

KK IF -Teknik Informatika - STEI ITB

Outline

Sejarah SVM

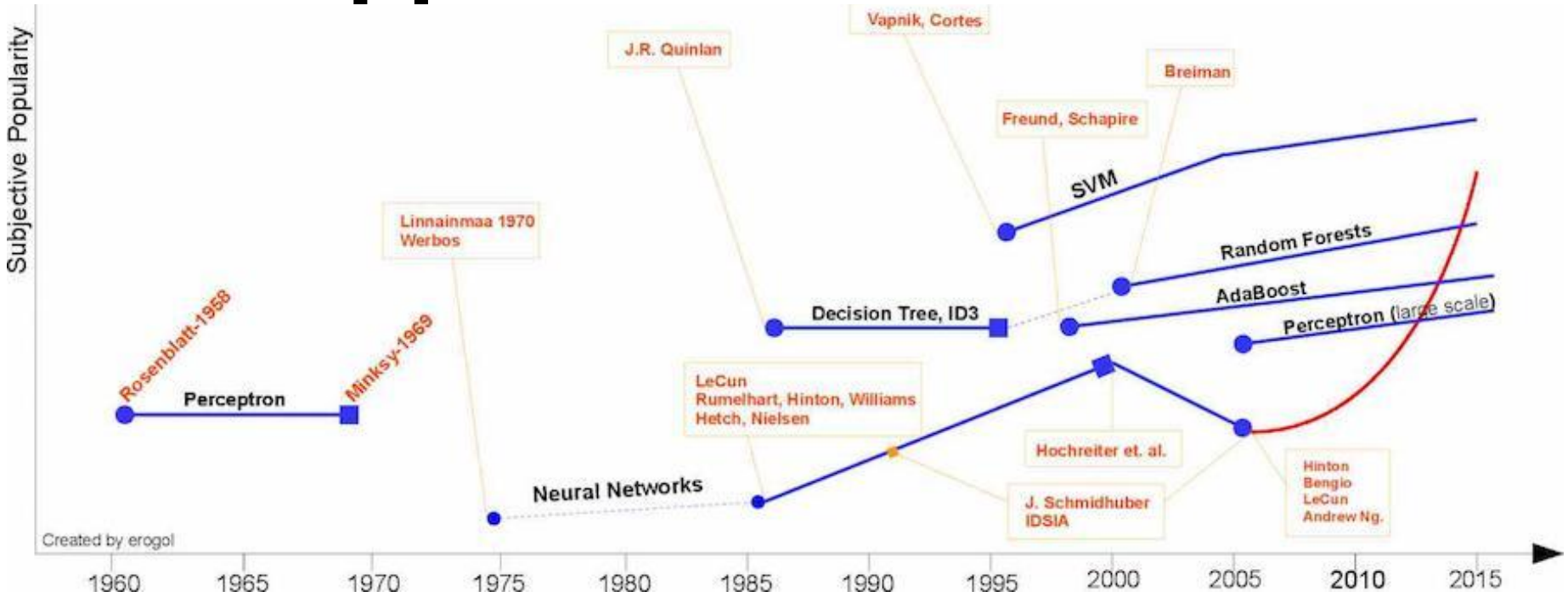
Bidang Pemisah Terbaik

Tujuan SVM

Klasifikasi Biner – Linear Separability

Hyperplane Classifier

Support Vector Machine



Eren Golge: https://www.kdnuggets.com/2014/10/deep-learning-make-machine-learning-algorithms-obsolete.html

- SVM diperkenalkan tahun 1992 oleh Vapnik, Boser, & Guyon
- Kinerja baik di berbagai aplikasi seperti *bioinformatics*, klasifikasi teks, pengenalan tulisan tangan dan lain-lain.

SVM

- 1980an
  - DTL dan NN memungkinkan pembelajaran nonlinear yang efisien
  - Kurang didukung dasar teoritis dan memungkinkan terjadinya local minima
- 1990an
  - Algoritma pembelajaran yang efisien untuk fungsi non linier berbasis teori komputasi

SVM Introduction

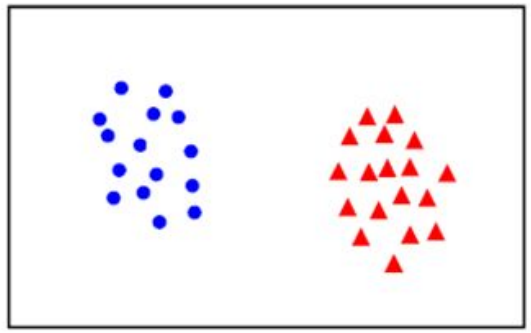
- Boser, B. E., Guyon, I. M., & Vapnik, V. N. (1992, July). A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory* (pp. 144-152). ACM.
- Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine learning*, *20*(3), 273-297.

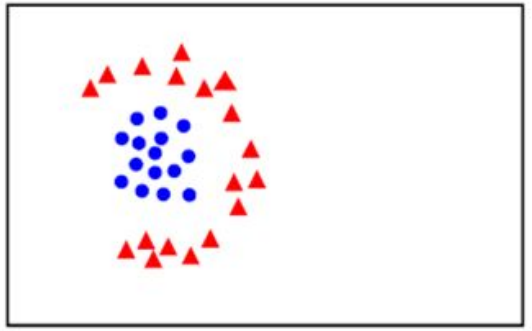
Klasifikasi Biner

Given training data $(\mathbf{x}_i, y_i)$ for $i = 1 \ldots N$, with $\mathbf{x}_i \in \mathbb{R}^d$ and $y_i \in \{-1, 1\}$, learn a classifier $f(\mathbf{x})$ such that

$$f(\mathbf{x}_i) \begin{cases} \geq 0 & y_i = +1 \\ < 0 & y_i = -1 \end{cases}$$

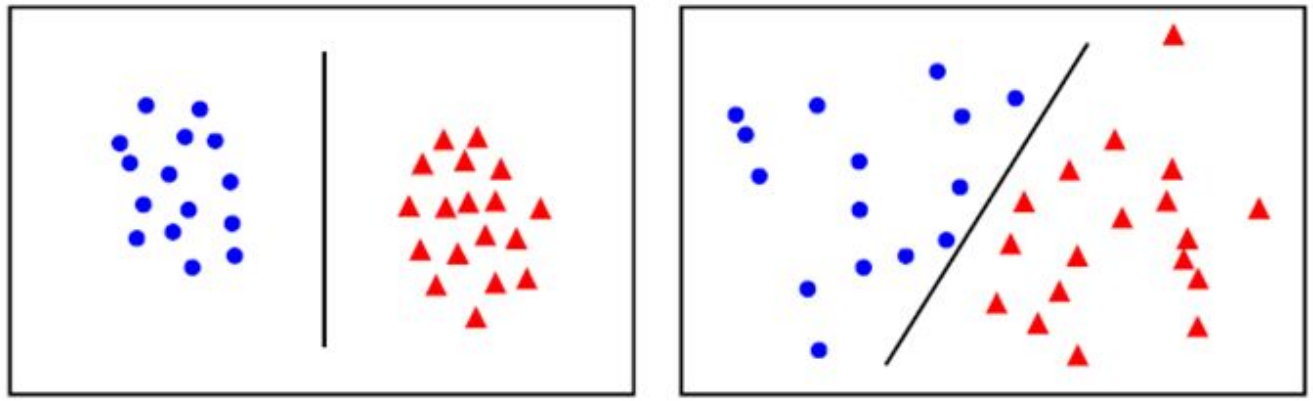
i.e. $y_i f(\mathbf{x}_i) > 0$ for a correct classification.



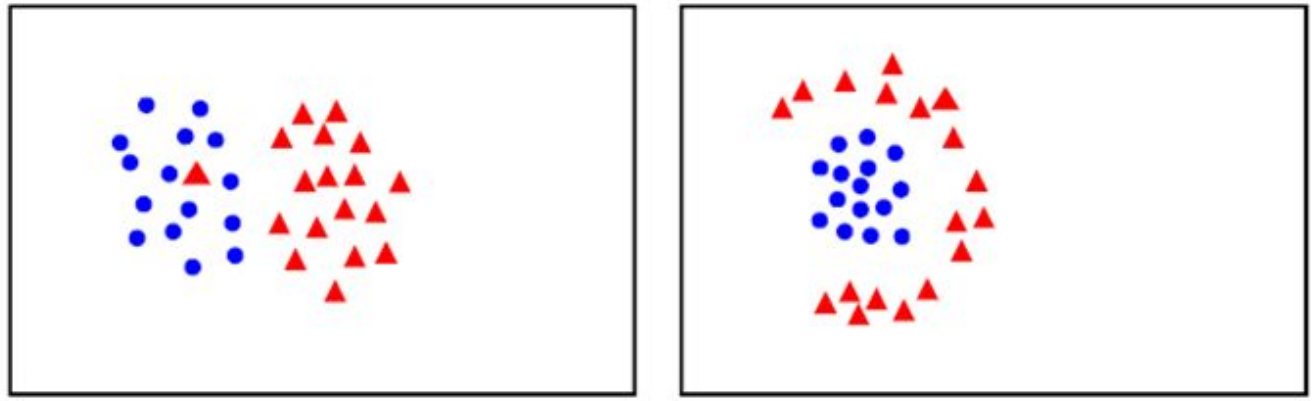


Linear Separability

linearly separable



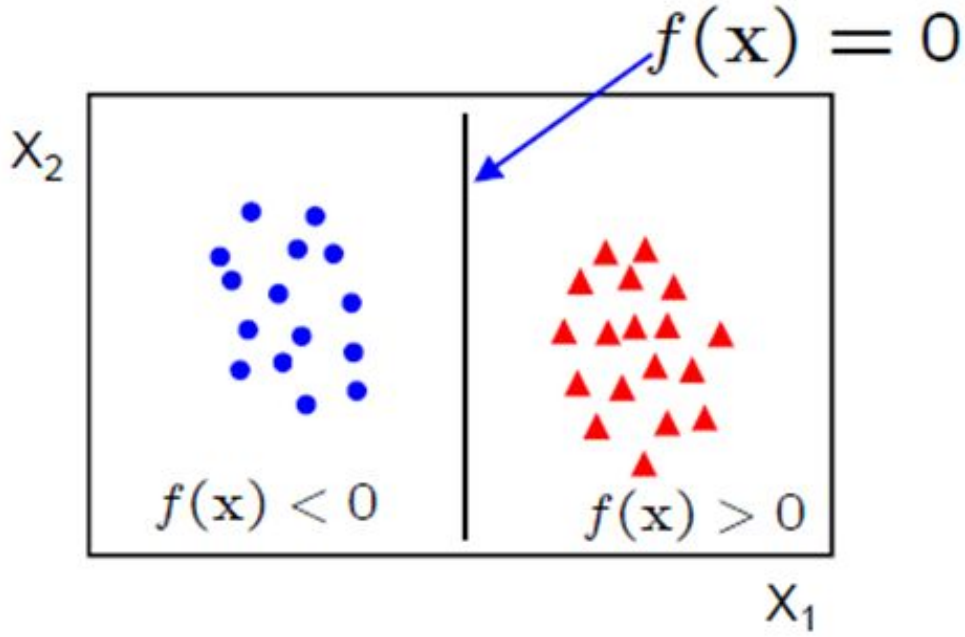
not linearly separable



Linear Classifier

A linear classifier has the form

$$f(\mathbf{x}) = \mathbf{w}^\top \mathbf{x} + b$$

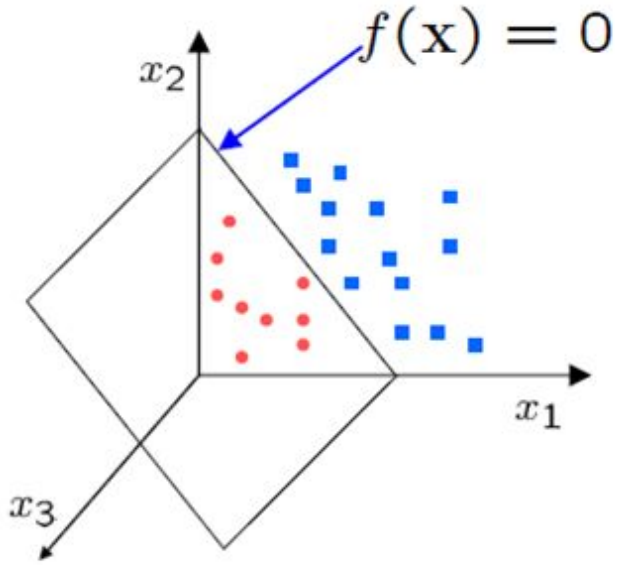


- in 2D the discriminant is a line
- $\mathbf{w}$ is the normal to the line, and b the bias
- $\mathbf{w}$ is known as the weight vector

Linear Classifier

A linear classifier has the form

$$f(\mathbf{x}) = \mathbf{w}^\top \mathbf{x} + b$$



- in 3D the discriminant is a plane, and in nD it is a hyperplane

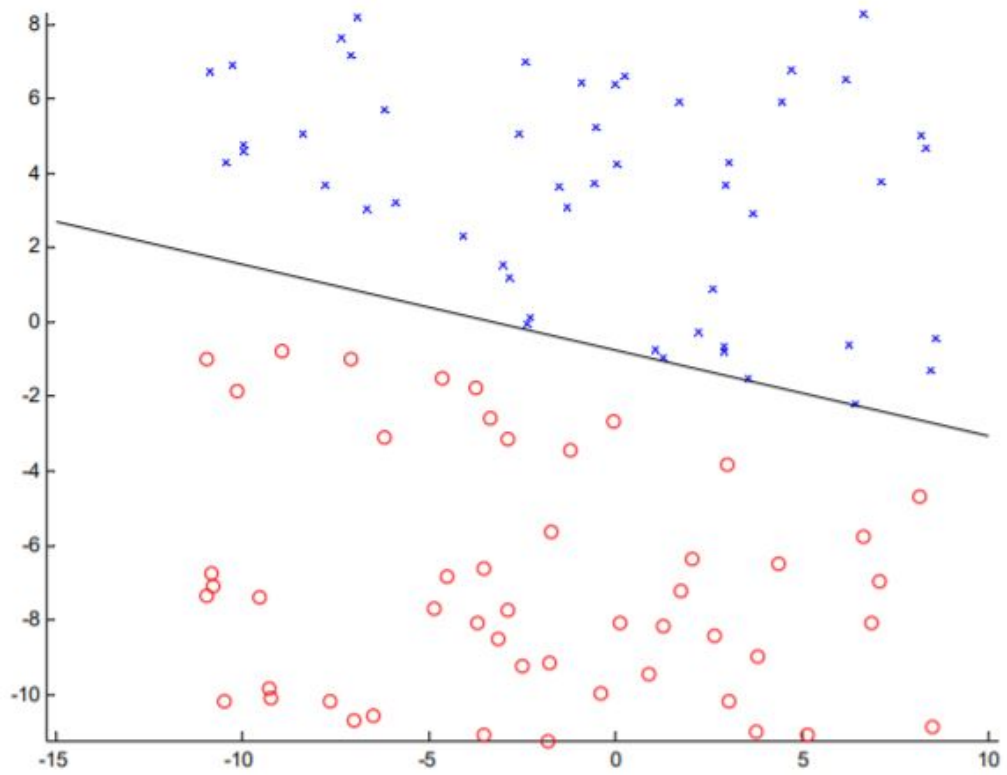
For a K-NN classifier it was necessary to `carry' the training data

For a linear classifier, the training data is used to learn **w** and then discarded

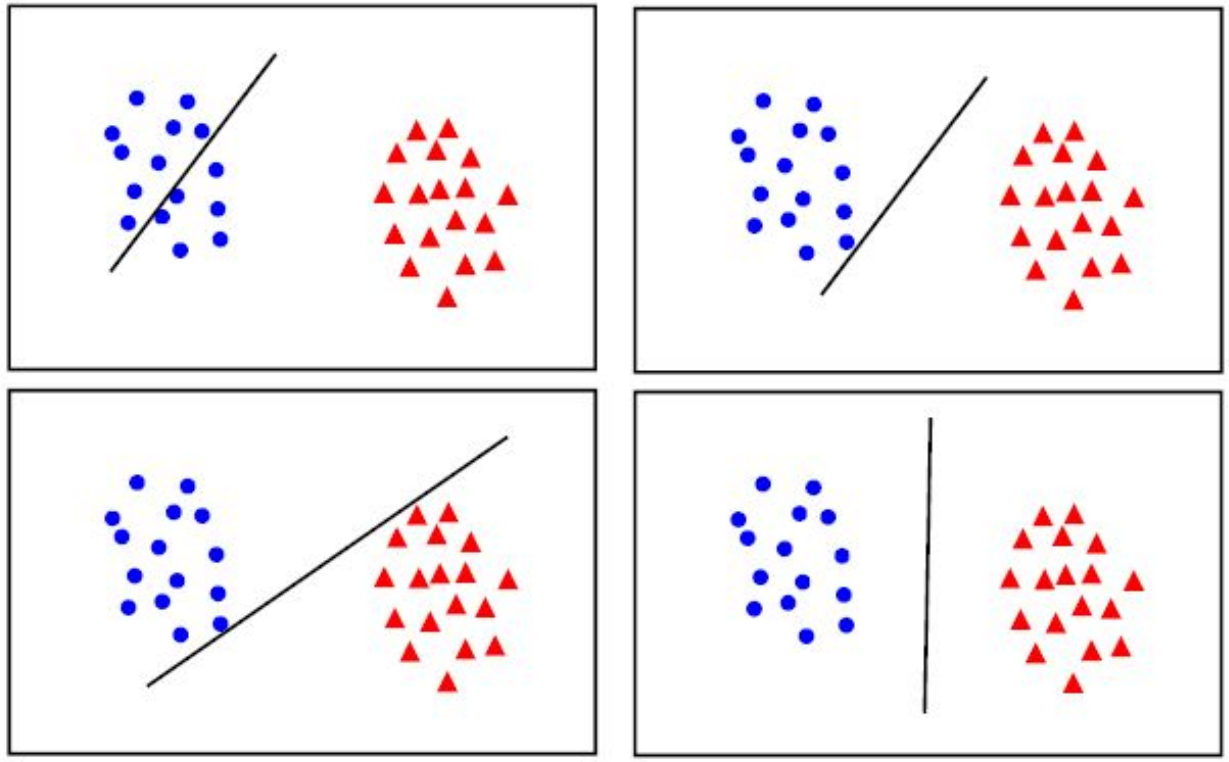
Only **w** is needed for classifying new data

Perceptron Weakness

- Perceptron biggest weakness is that it will not find the same hyperplane every time.
  - Not all separating hyperplanes are equals.
  - If the Perceptron gives you a hyperplane that is very close to all the data points from one class, you have a right to believe that it will generalize poorly when given new data.
- After an accurate hyperplane is found, the training process will stop and it is considered to have converged.



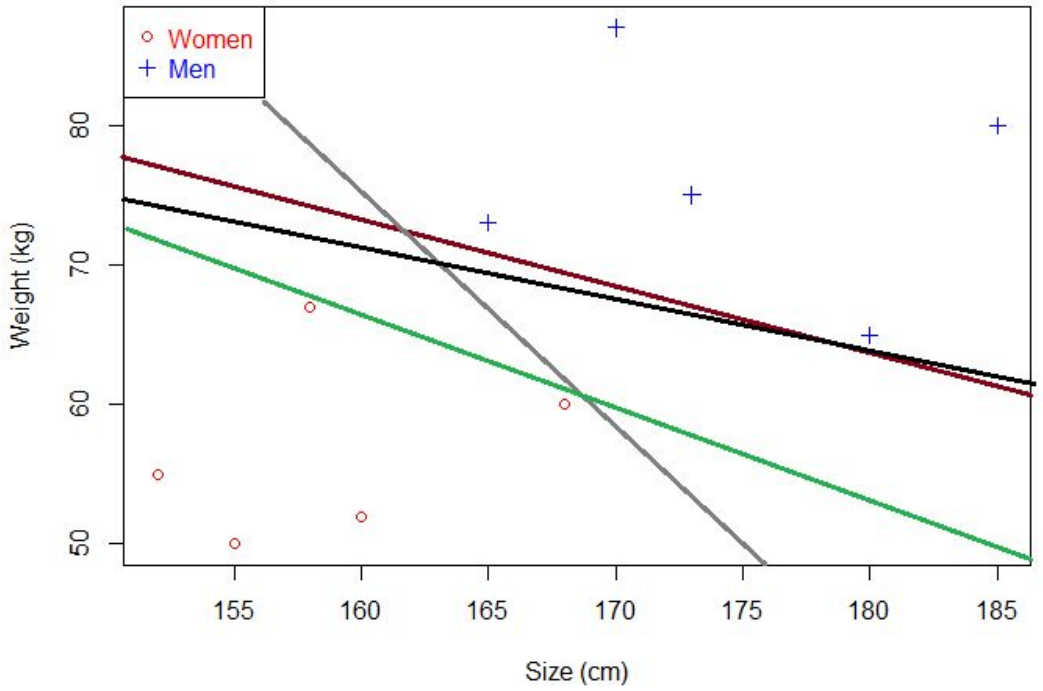
Bidang Pemisah Terbaik?

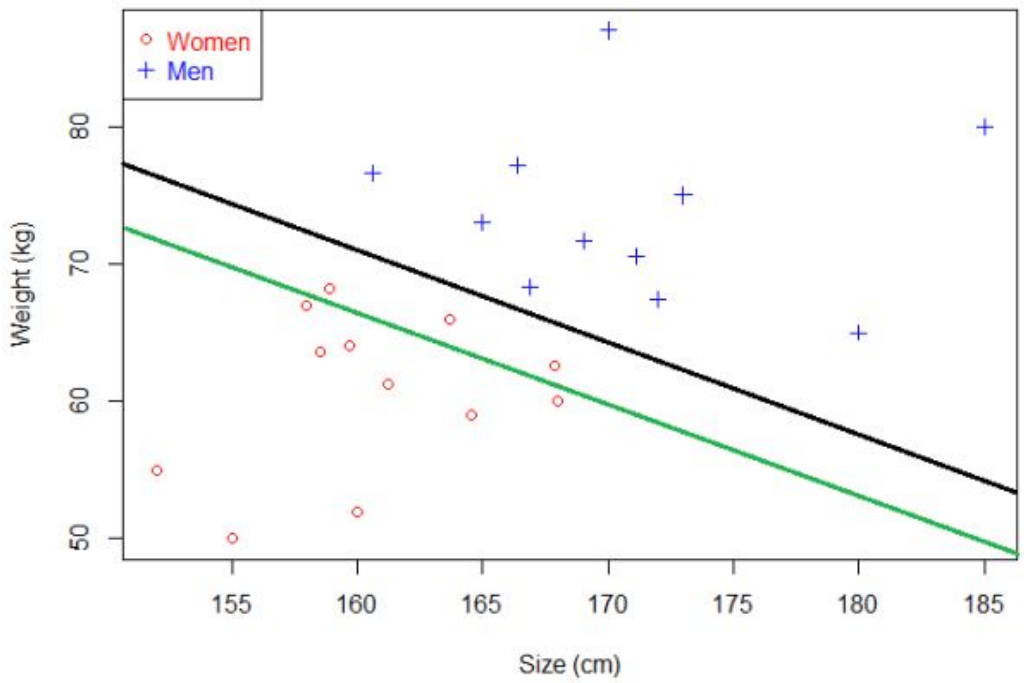


- Mengapa?

https://www.robots.ox.ac.uk/~az/lectures/ml/lect2.pdf

Bidang Pemisah Terbaik (lanj)





Kiri: semua bidang pemisah valid karena memisahkan kedua kelas pada training data.
Kanan: real-life data. Bidang pemisah hitam lebih baik daripada hijau.

SVM Objective

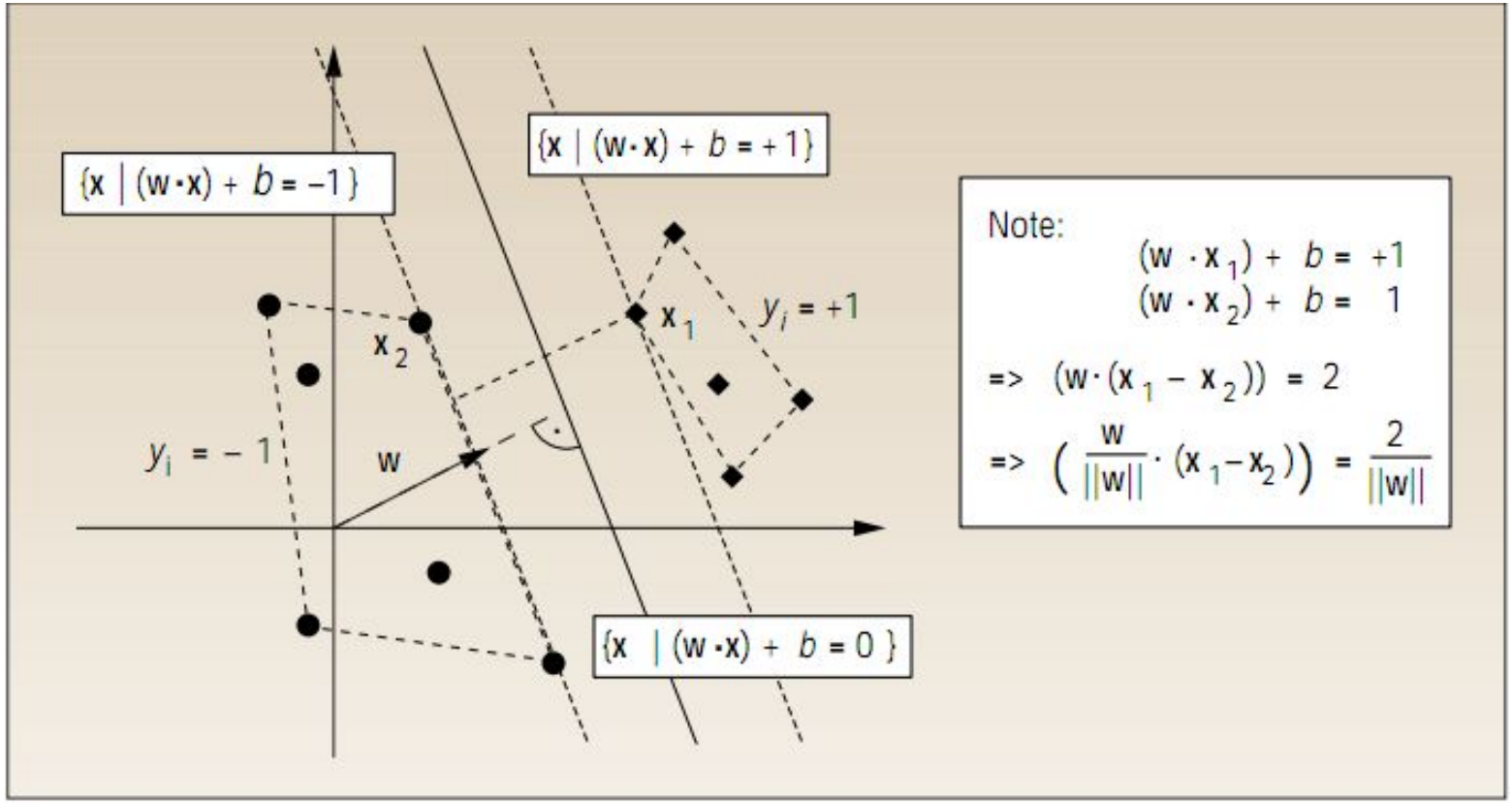
- Objective of the SVM **is to find the optimal separating hyperplane which maximizes the margin of the training data.** There will never be any data point inside the margin.
- Menggunakan optimasi kuadratik untuk menghindari ‘local minimum’ isu yang ada pada NN (Greedy)
- Menggunakan fungsi kernel untuk memisahkan non-linear region

https://www.svm-tutorial.com/2015/06/svm-understanding-math-part-3/

Hyperplane Classifier

- Hipotesis:
- x1, x2 ∈ training data

$$f(x) = sign(\vec{w}.\vec{x} + b); \vec{w}, \vec{x} \in \Re^N; b \in \Re$$



Vector Direction

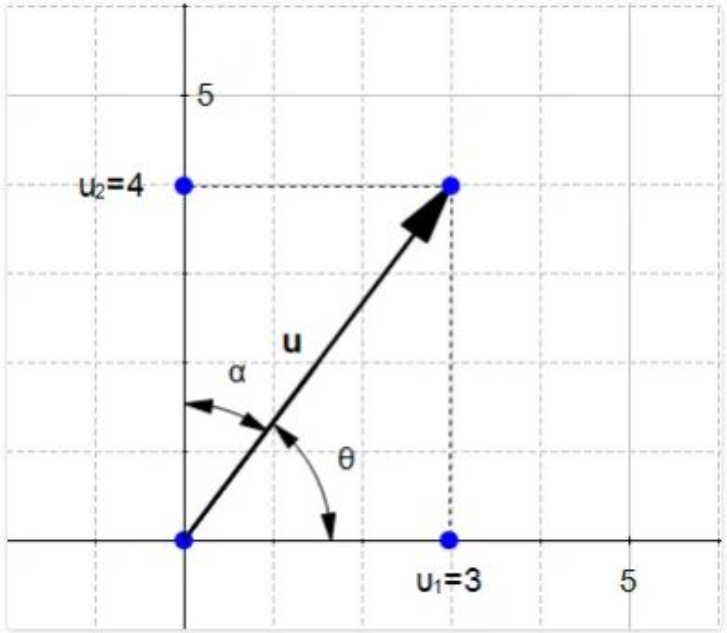


Figure 4 - direction of a vector

u(u1,u2) with u1=3 and u2=4

$$cos(\theta) = \frac{u_1}{\|u\|} \qquad cos(\alpha) = \frac{u_2}{\|u\|}$$

- Naive definition 1: The direction of the vector u is defined by the angle θ with respect to the horizontal axis, and with the angle α with respect to the vertical axis.
- Naive definition 2: The direction of the vector u is defined by the cosine of the angle θ and the cosine of the angle α.

02 SVM for Linearly Separable Data

IF3170 Artificial Intelligence

Modul: Supervised Learning

02 SVM for Linearly Separable Data

IF3170 - Inteligensi Artifisial

Dr. Fariska Z. Ruskanda, S.T., M.T.
(fariska@informatika.org)

KK IF -Teknik Informatika - STEI ITB

Outline

Support Vectors

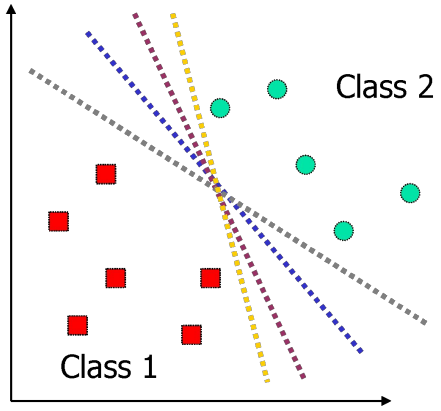
Optimal Hyperplane

Quadratic Optimization Problem

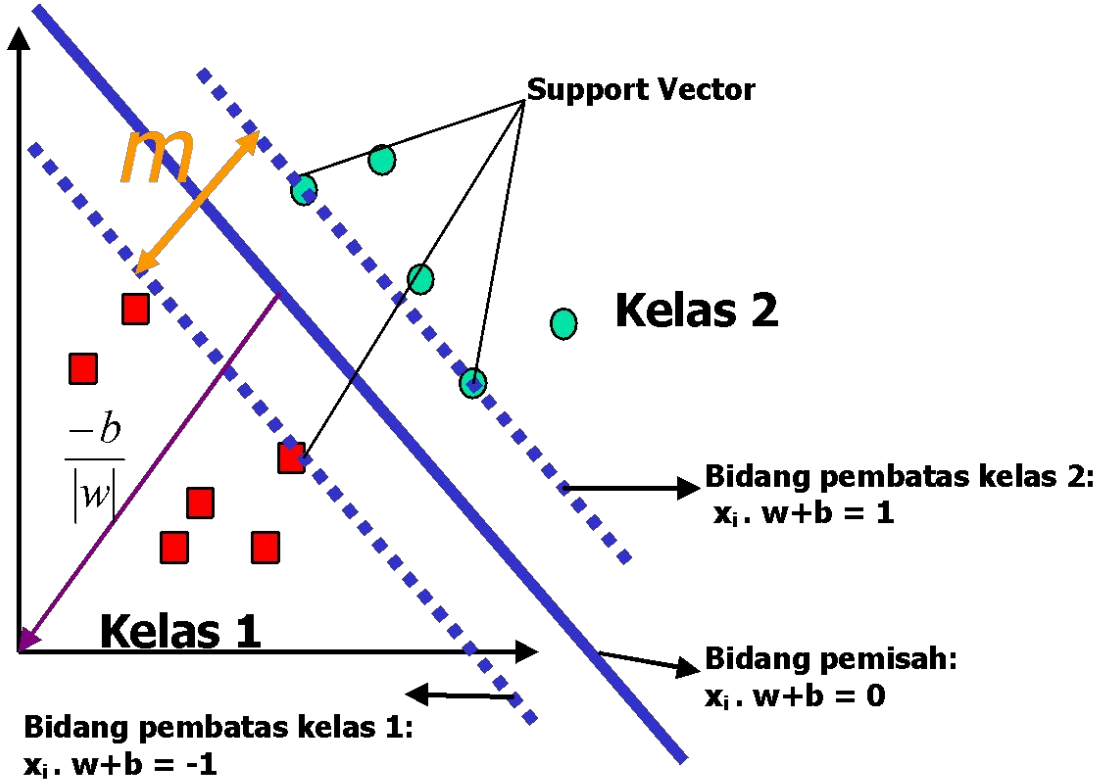
Bidang Pemisah Terbaik

Calculation Example

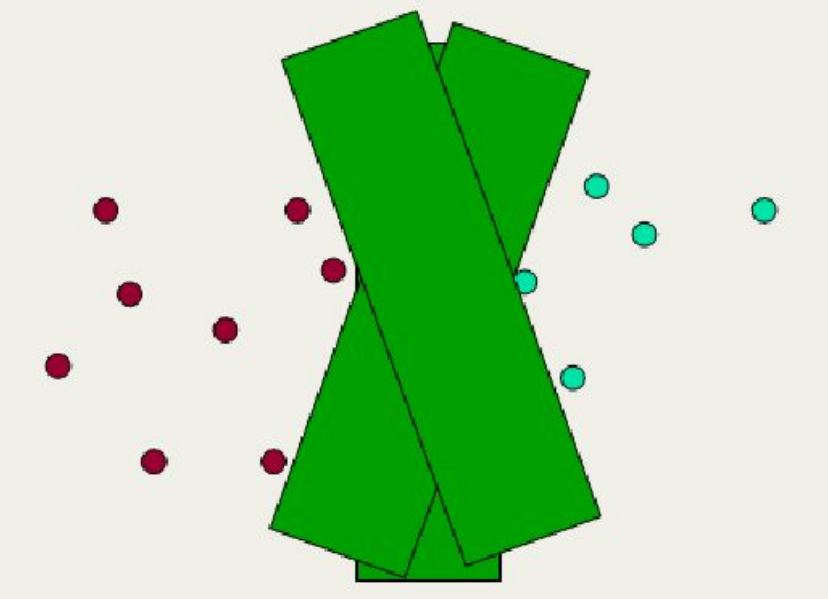
Mencari Bidang Pemisah Terbaik (1)



Mencari bidang pemisah dengan margin (m) terbesar



Mencari Bidang Pemisah Terbaik (2)



- 2 kelas dapat dipisahkan oleh sepasang bidang pembatas yang sejajar.
  - Bidang pembatas pertama membatasi kelas pertama
  - Bidang pembatas kedua membatasi kelas kedua
- ***Support Vector*** : Vector pada training data yang men-*support* bidang pemisah

Mengapa mencari margin terbesar?

- Bidang pemisah terbaik dengan margin terbesar memiliki generalisasi yang lebih baik
  - Titik yang dekat dengan bidang pemisah merepresentasikan ketidakyakinan klasifikasi : 50% peluang pengambilan keputusan oleh classifier
- Kapasitas memori untuk menyimpan model menjadi lebih sedikit
  - Hanya support vector yang mempengaruhi pengambilan keputusan

Support Vectors

- The data points that lie closest to the decision surface (or hyperplane)
- They are the data points most difficult to classify
- They have direct bearing on the optimum location of the decision surface

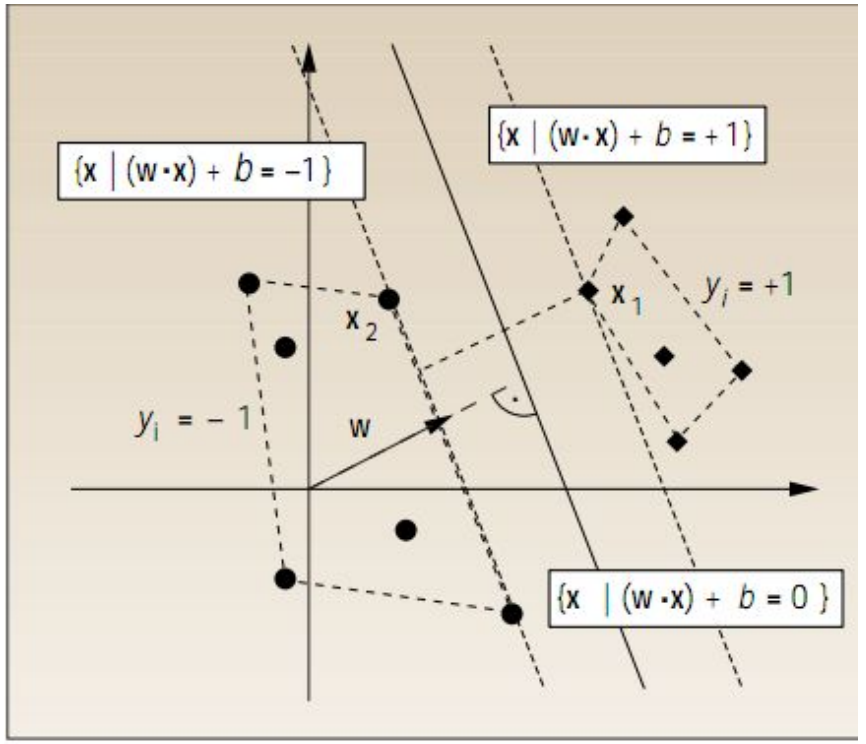
Support Vectors

- Support vectors memenuhi:

$$|\vec{w}.\vec{x}_i + b| = 1$$

- Untuk itu, semua training data berlabel “+” dan “–” memenuhi juga:

$$y_i(\vec{w}.\vec{x}_i + b) \geq 1$$



Optimal Hyperplane

- Optimal hyperplane:
  - *maximum-margin hyperplane*
  - atau hyperplane with maximal margin separation between two classes.
- Maximal margin (2/||w||) ≈ minimize ||w|| dengan batasan konsistensi training data tetap terjaga (semua klasifikasi benar).
- Minimize: $V(\vec{w},b)=\frac{1}{2}\vec{w}.\vec{w}$

  subject to: $y_i(\vec{w}.\vec{x}_i+b)\geq 1$

Recall the distance from a point(x_0,y_0) to a line: $Ax+By+c = 0$ is: $|Ax_0+By_0+c|/\text{sqrt}(A^2+B^2)$, so,
The distance between H_0 and H_1 is then: $|w\bullet x+b|/||w||=1/||w||$, so
The total distance between H_1 and H_2 is thus: $2/||w||$

Optimal Hyperplane (2)

- Pencarian bidang pemisah terbaik dapat dirumuskan menjadi:

$$\min \frac{1}{2}|w|^2$$
$$s.t\ \ y_i(x_i.w+b)-1 \geq 0$$

- Want to look for solution point p where
$$\nabla f(p) = \nabla \lambda g(p)$$
$$g(x) = 0$$
- Or, combining these two as the *Langrangian* L & requiring derivative of L be zero:
$$L(x,a) = f(x) - ag(x)$$
$$\nabla(x,a) = 0$$

- Supaya lebih mudah diselesaikan ubah ke formula Lagrangian

$$\min_{w,b} L_p(w,b,\alpha) \equiv \frac{1}{2}|w|^2 - \sum_{i=1}^{n} \alpha_i y_i (x_i.w+b) + \sum_{i=1}^{n} \alpha_i$$
$$\alpha \geq 0$$

- Minimumkan Lp terhadap b dan w (saddle point), diperoleh:

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

$$\frac{\partial}{\partial w} L_p(w,b,\alpha) = 0 \Rightarrow \frac{\partial L_P}{\partial \mathbf{w}} = \mathbf{w} - \sum_{i=1}^{l} a_i y_i \mathbf{x}_i = 0$$

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

Langrangian Dual Problem

The Lagrangian Dual Problem: instead of minimizing over **w**, b, subject to constraints involving a's, we can maximize over a (the dual variable) subject to the relations obtained previously for **w** and b

Our solution must satisfy these two relations:

$$\mathbf{w} = \sum_{i=1}^{l} a_i y_i \mathbf{x}_i, \quad \sum_{i=1}^{l} a_i y_i = 0$$

By substituting for **w** and b back in the original eqn we can get rid of the dependence on **w** and b.

Note first that we already now have our answer for what the weights **w** must be: they are a linear combination of the training inputs and the training outputs, x_i and y_i and the values of a. We will now solve for the a's by differentiating the dual problem wrt a, and setting it to zero. Most of the a's will turn out to have the value zero. The non-zero a's will correspond to the support vectors

Substitute the Primal Problem

Primal problem:

$$\min L_P = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{l} a_i y_i \left(\mathbf{x}_i \cdot \mathbf{w} + b\right) + \sum_{i=1}^{l} a_i$$

s.t. $\forall i \; a_i \geq 0$

$$\mathbf{w} = \sum_{i=1}^{l} a_i y_i \mathbf{x}_i, \quad \sum_{i=1}^{l} a_i y_i = 0$$

Dual problem:

$$\max L_D(a_i) = \sum_{i=1}^{l} a_i - \frac{1}{2} \sum_{i=1}^{l} a_i a_j y_i y_j \left(\mathbf{x}_i \cdot \mathbf{x}_j\right)$$

s.t. $\sum_{i=1}^{l} a_i y_i = 0 \;\&\; a_i \geq 0$

(note that we have removed the dependence on $\mathbf{w}$ and b)

Quadratic Optimization Problem

- Minimize: $$W(\vec{\alpha}) = -\sum_{i=1}^{n} \alpha_i + \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} y_i y_j \alpha_i \alpha_j (\vec{x}_i . \vec{x}_j)$$

  subject to: $$\sum_{i=1}^{n} y_i \alpha_i = 0; \forall i \in [1..n] : \alpha_i \geq 0$$

- Support vectors: $\alpha_i > 0$
- Hyperplane: $$\vec{w}.\vec{x} = (\sum_{i=1}^{n} \alpha_i y_i \vec{x}_i).\vec{x} = \sum_{i=1}^{n} \alpha_i y_i (\vec{x}_i . \vec{x})$$

$$b = y_{sv} - \vec{w}.\vec{x}_{sv}$$

Optimization Problem Example

Suppose we have two 2D data points: [(x11, x12), (x21, x22)] with labels [y1, y2]

- Minimize: W(a) = -(a1+a2) + 0.5*
  [y1*y1*a1*a1*(x11*x11 + x12*x12) +
  y1*y2*a1*a2*(x11*x21 + x12*x22) +
  y2*y1*a2*a1*(x21*x11 + x22*x12) +
  y2*y2*a2*a2*(x21*x21 + x22*x22)]

  subject to: (y1*a1 + y2*a2) = 0
- Support vectors: a1> 0, a2> 0
- Hyperplane:

w.x = a1*y1*((x11,x12)*x) + a2*y2*((x21,x22)*x)

b = ysv – w.xsv

$$W(\vec{\alpha}) = -\sum_{i=1}^{n}\alpha_i + \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} y_i y_j \alpha_i \alpha_j (\vec{x}_i . \vec{x}_j)$$

$$\sum_{i=1}^{n} y_i \alpha_i = 0; \forall i \in [1..n] : \alpha_i \geq 0$$

$$\vec{w}.\vec{x} = (\sum_{i=1}^{n} \alpha_i y_i \vec{x}_i).\vec{x} = \sum_{i=1}^{n} \alpha_i y_i (\vec{x}_i . \vec{x})$$

$$b = y_{sv} - \vec{w}.\vec{x}_{sv}$$

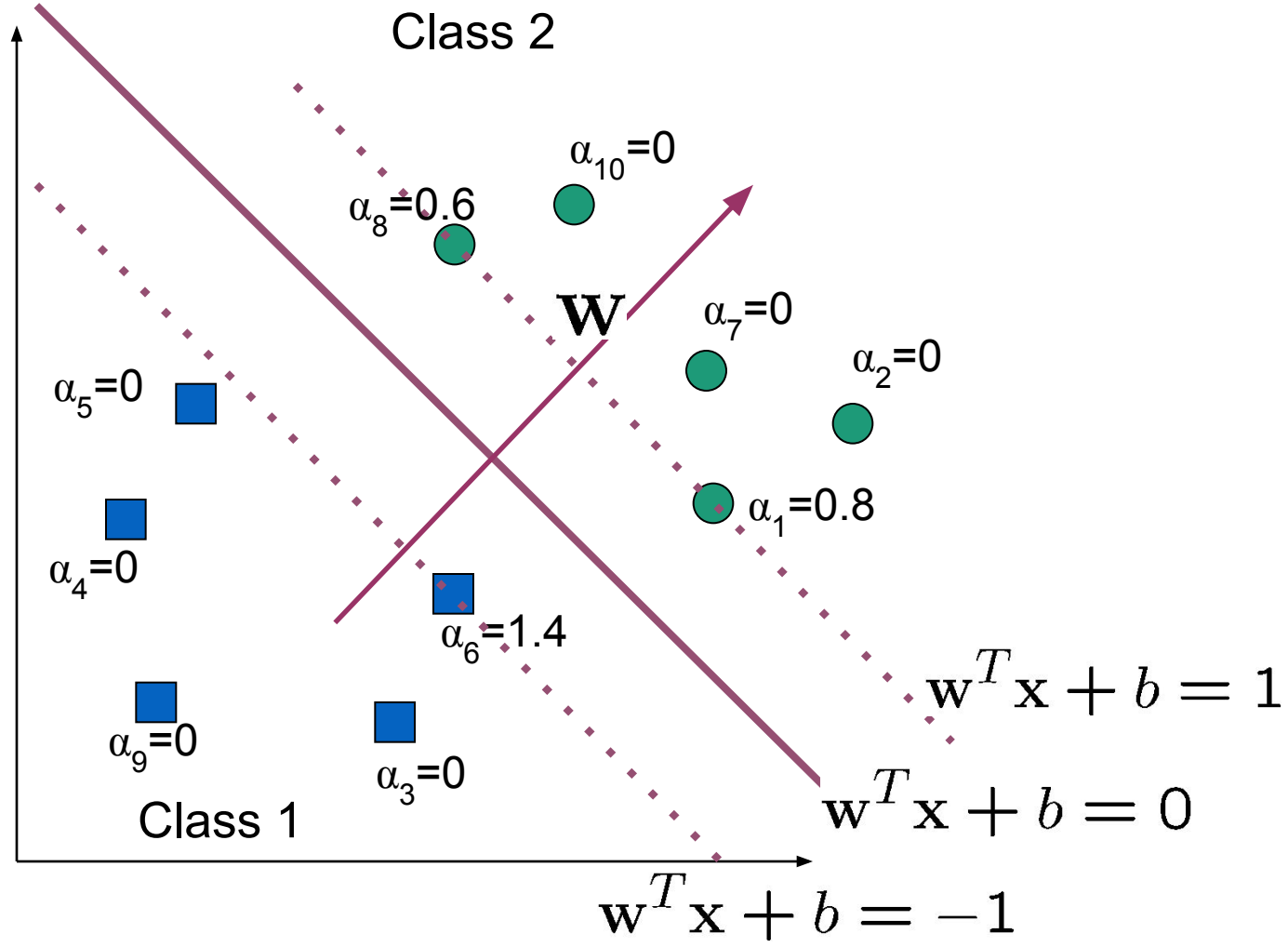
Quadratic Programming

- Quadratic programming (QP) is the problem of optimizing a quadratic objective function and is one of the simplest form of non-linear programming.
- The objective function can contain bilinear or up to second order polynomial terms, and the constraints are linear and can be both equalities and inequalities.

$$\min f(x) = q^T x + \frac{1}{2} x^T Q x$$
$$s.t. Ax = a$$
$$Bx \leq b$$
$$x \geq 0$$

https://optimization.mccormick.northwestern.edu/index.php/Quadratic_programming

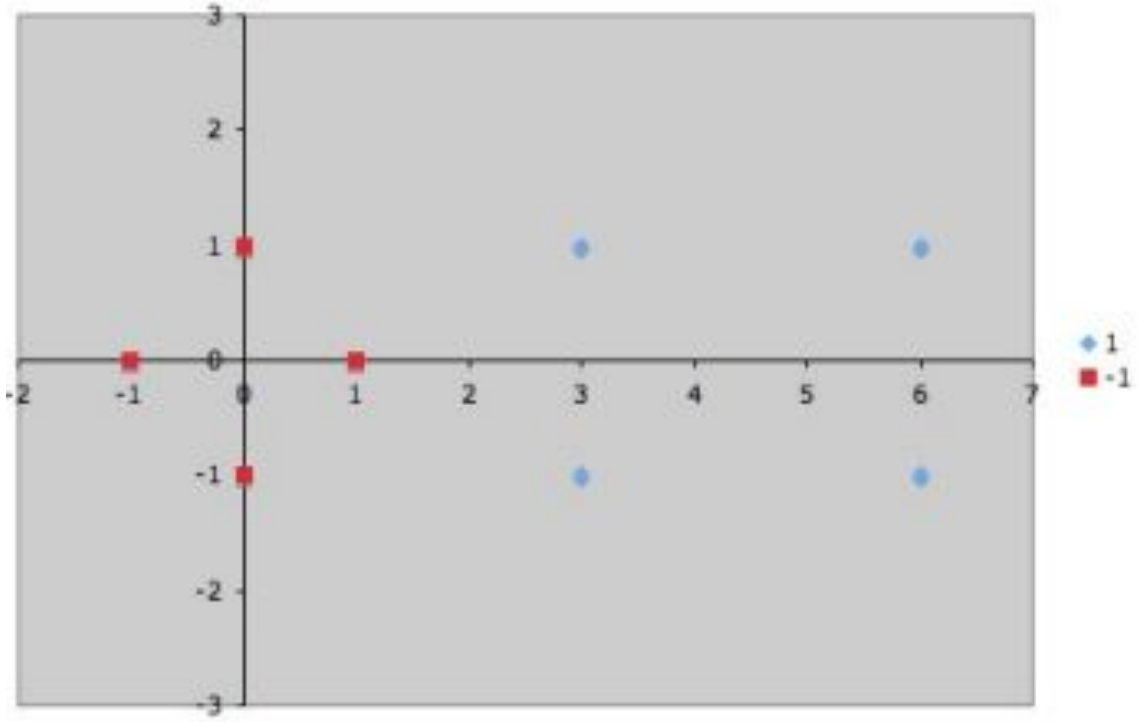
A Geometrical Interpretation



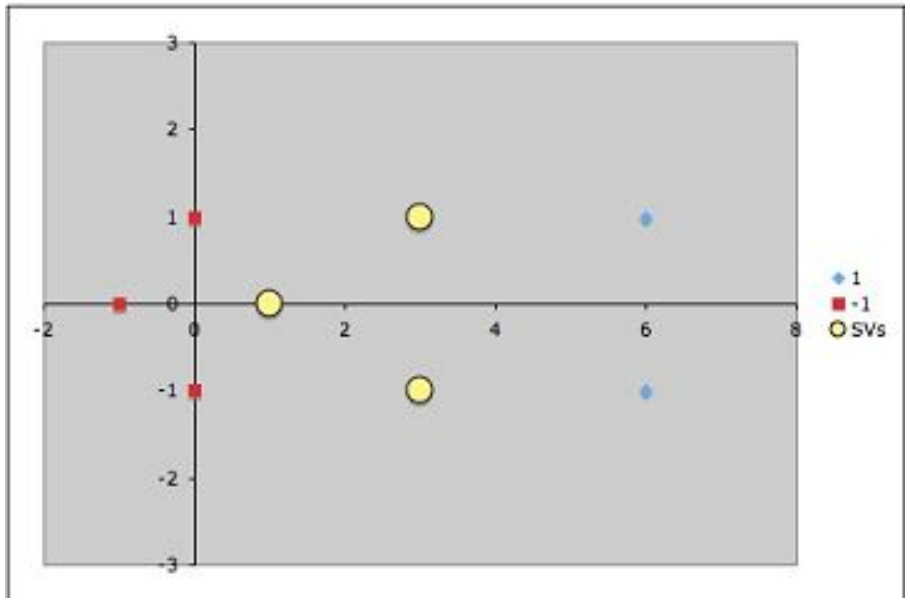
A Simple Introduction to Support Vector Machines; Martin Law; Michigan State Univ.

Contoh

x1	x2	Kelas
3	1	+1
3	-1	+1
6	1	+1
6	-1	+1
1	0	-1
0	1	-1
0	-1	-1
-1	0	-1



- Taken from: http://axon.cs.byu.edu/Dan/678/miscellaneous/SVM.example.pdf



Support Vectors

$$f(\vec{x}) = \sum_{i=1}^{nsv} (\alpha_i y_i \vec{x}_i \vec{x}) + b$$

$$\begin{pmatrix}1\\0\end{pmatrix} \Rightarrow -1 = \alpha_1 . -1 . \begin{pmatrix}1\\0\end{pmatrix} . \begin{pmatrix}1\\0\end{pmatrix} + \alpha_2 . 1 . \begin{pmatrix}3\\1\end{pmatrix} . \begin{pmatrix}1\\0\end{pmatrix} + \alpha_3 . 1 . \begin{pmatrix}3\\-1\end{pmatrix} . \begin{pmatrix}1\\0\end{pmatrix} + b = -\alpha_1 + 3\alpha_2 + 3\alpha_3 + b \ldots (1)$$

$$\begin{pmatrix}3\\1\end{pmatrix} \Rightarrow 1 = \alpha_1 . -1 . \begin{pmatrix}1\\0\end{pmatrix} . \begin{pmatrix}3\\1\end{pmatrix} + \alpha_2 . 1 . \begin{pmatrix}3\\1\end{pmatrix} . \begin{pmatrix}3\\1\end{pmatrix} + \alpha_3 . 1 . \begin{pmatrix}3\\-1\end{pmatrix} . \begin{pmatrix}3\\1\end{pmatrix} + b = -3\alpha_1 + 10\alpha_2 + 8\alpha_3 + b \ldots (2)$$

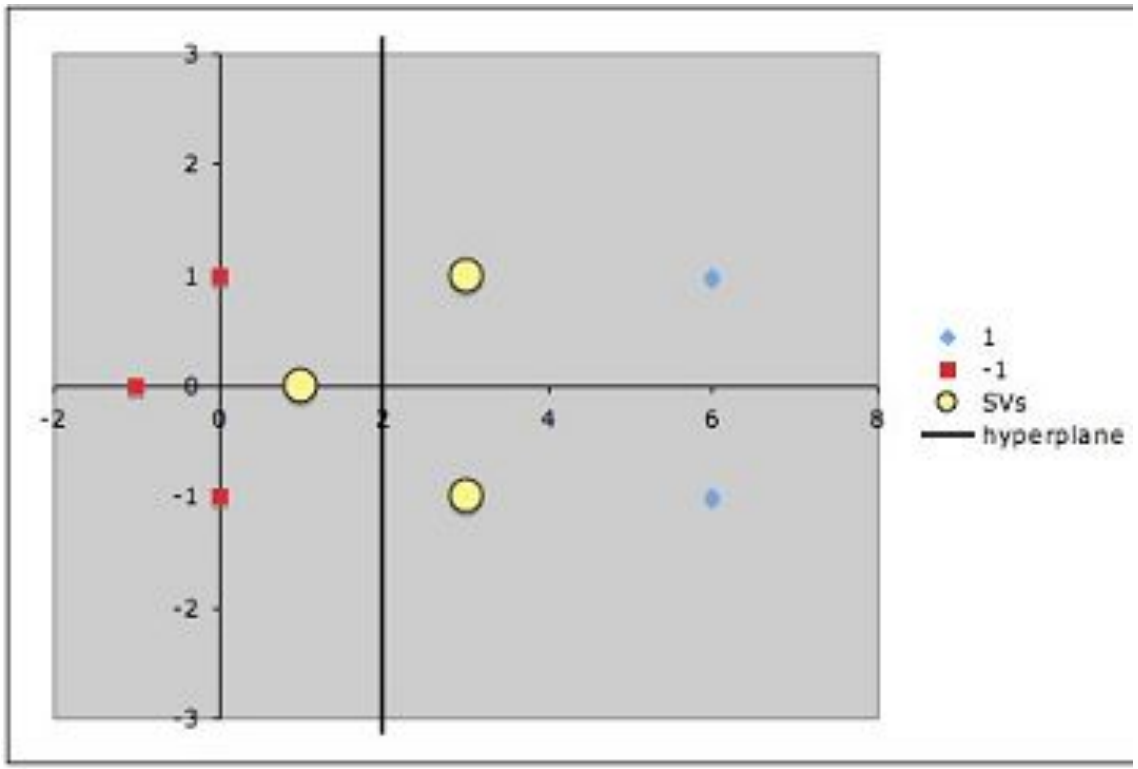
$$\begin{pmatrix}3\\-1\end{pmatrix} \Rightarrow 1 = \alpha_1 . -1 . \begin{pmatrix}1\\0\end{pmatrix} . \begin{pmatrix}3\\-1\end{pmatrix} + \alpha_2 . 1 . \begin{pmatrix}3\\1\end{pmatrix} . \begin{pmatrix}3\\-1\end{pmatrix} + \alpha_3 . 1 . \begin{pmatrix}3\\-1\end{pmatrix} . \begin{pmatrix}3\\-1\end{pmatrix} + b = -3\alpha_1 + 8\alpha_2 + 10\alpha_3 + b \ldots (3)$$

$$-\alpha_1 + 3\alpha_2 + 3\alpha_3 + b = -1 \ldots (1)$$
$$-3\alpha_1 + 10\alpha_2 + 8\alpha_3 + b = 1 \ldots (2)$$
$$-3\alpha_1 + 8\alpha_2 + 10\alpha_3 + b = 1 \ldots (3)$$
$$-\alpha_1 + \alpha_2 + \alpha_3 = 0 \ldots (4)$$

$$(2) - (3): 2\alpha_2 - 2\alpha_3 = 0 \rightarrow \alpha_2 = \alpha_3 \ldots (5)$$
$$(5) subs (1): -\alpha_1 + 6\alpha_2 + b = -1 \ldots (6)$$
$$(5) subs (2): -3\alpha_1 + 18\alpha_2 + b = 1 \ldots (7)$$
$$(6) - (7): 2\alpha_1 - 12\alpha_2 = -2$$
$$\alpha_1 - 6\alpha_2 = -1$$
$$\alpha_1 = 6\alpha_2 - 1 \ldots (8)$$
$$(8,5) subs (4): -6\alpha_2 + 1 + \alpha_2 + \alpha_2 = 0$$
$$-4\alpha_2 = -1 \rightarrow \alpha_2 = \frac{1}{4} = 0.25$$

$$\alpha_1 = 0.5; \alpha_2 = 0.25; \alpha_3 = 0.25$$
$$substitusi (6): -0.5 + 6 * 0.25 + b = -1$$
$$b = -2$$

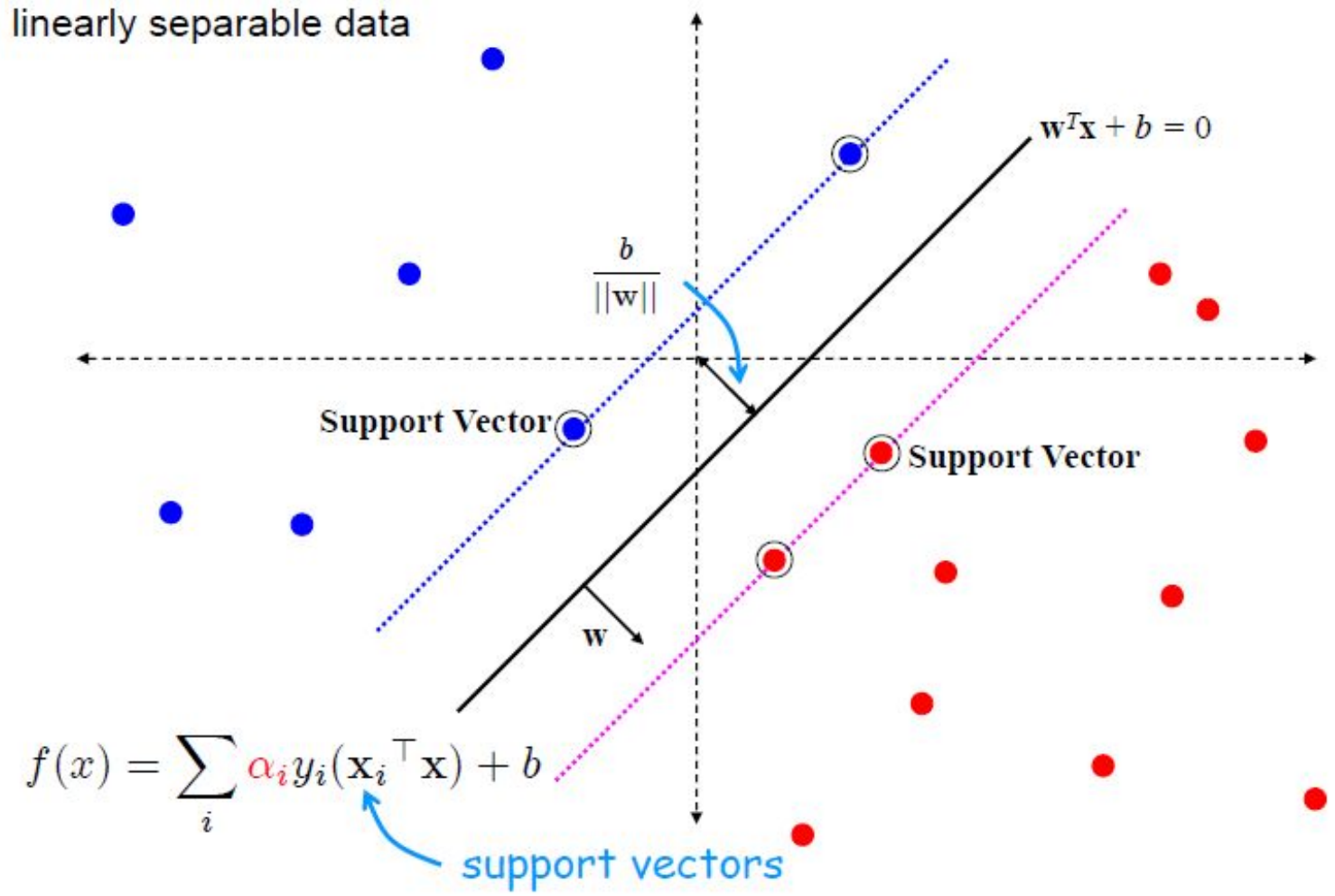
Hipotesis



$$f(\vec{x}) = \sum_{i=1}^{nsv} (\alpha_i y_i \vec{x}_i \vec{x}) - 2; \alpha_1 = 0.5; \alpha_2 = \alpha_3 = 0.25$$

$$f\left(\begin{pmatrix}6\\1\end{pmatrix}\right) = sign((0.5)(-1)\begin{pmatrix}1\\0\end{pmatrix}.\begin{pmatrix}6\\1\end{pmatrix} + (0.25)(1)\begin{pmatrix}3\\1\end{pmatrix}.\begin{pmatrix}6\\1\end{pmatrix} + (0.25)(1)\begin{pmatrix}3\\-1\end{pmatrix}.\begin{pmatrix}6\\1\end{pmatrix} - 2) = sign(-3 + 4.75 + 4.25 - 2) = sign(4) = 1$$

Summary


linearly separable data
$\mathbf{w}^T\mathbf{x} + b = 0$
$\frac{b}{||\mathbf{w}||}$
Support Vector
Support Vector
$\mathbf{w}$
$f(x) = \sum_i \alpha_i y_i (\mathbf{x}_i{}^\top \mathbf{x}) + b$
support vectors

Inner Product and Similarity

Why should inner product kernels be involved in pattern recognition using SVMs, or at all?

– Intuition is that inner products provide some measure of 'similarity'

– Inner product in 2D between 2 vectors of unit length returns the cosine of the angle between them = how 'far apart' they are

e.g. $\mathbf{x} = [1, 0]^T$, $\mathbf{y} = [0, 1]^T$

i.e. if they are parallel their inner product is 1 (completely similar)

$$\mathbf{x}^T \mathbf{y} = \mathbf{x} \bullet \mathbf{y} = 1$$

If they are perpendicular (completely unlike) their inner product is 0 (so should not contribute to the correct classifier)

$$\mathbf{x}^T \bullet \mathbf{y} = \mathbf{x} \bullet \mathbf{y} = 0$$

03 SVM for Non-linearly Separable Data

IF3170 Artificial Intelligence

Modul: Supervised Learning

03 SVM for Non-linearly Separable Data

IF3170 - Inteligensi Artifisial

Dr. Fariska Z. Ruskanda, S.T., M.T.
(fariska@informatika.org)

KK IF -Teknik Informatika - STEI ITB

Outline

Slack Variable

Non-linearly Separable

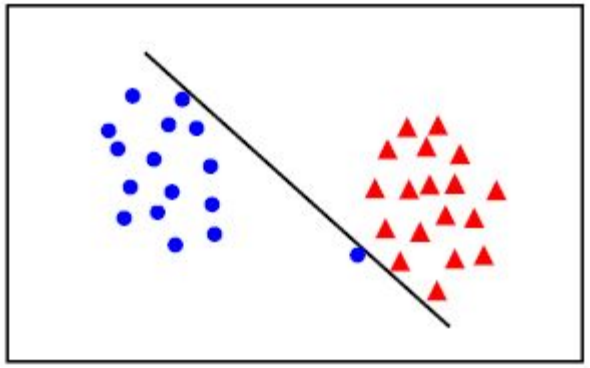
Optimization Problem

SVM for Non-linearly Separable Data

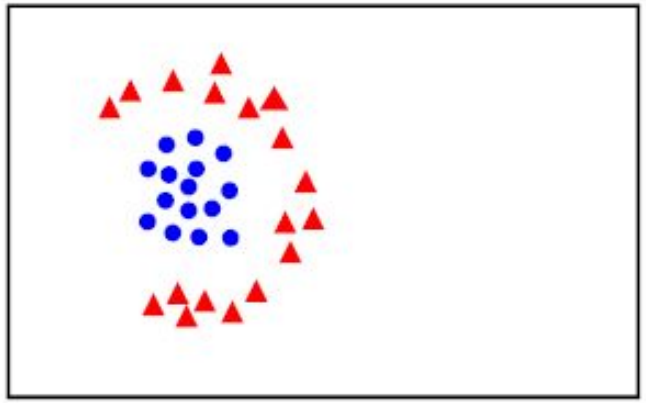
Non-linear Boundary Transformation

Non-Linearly Separable

- Existence of the noise



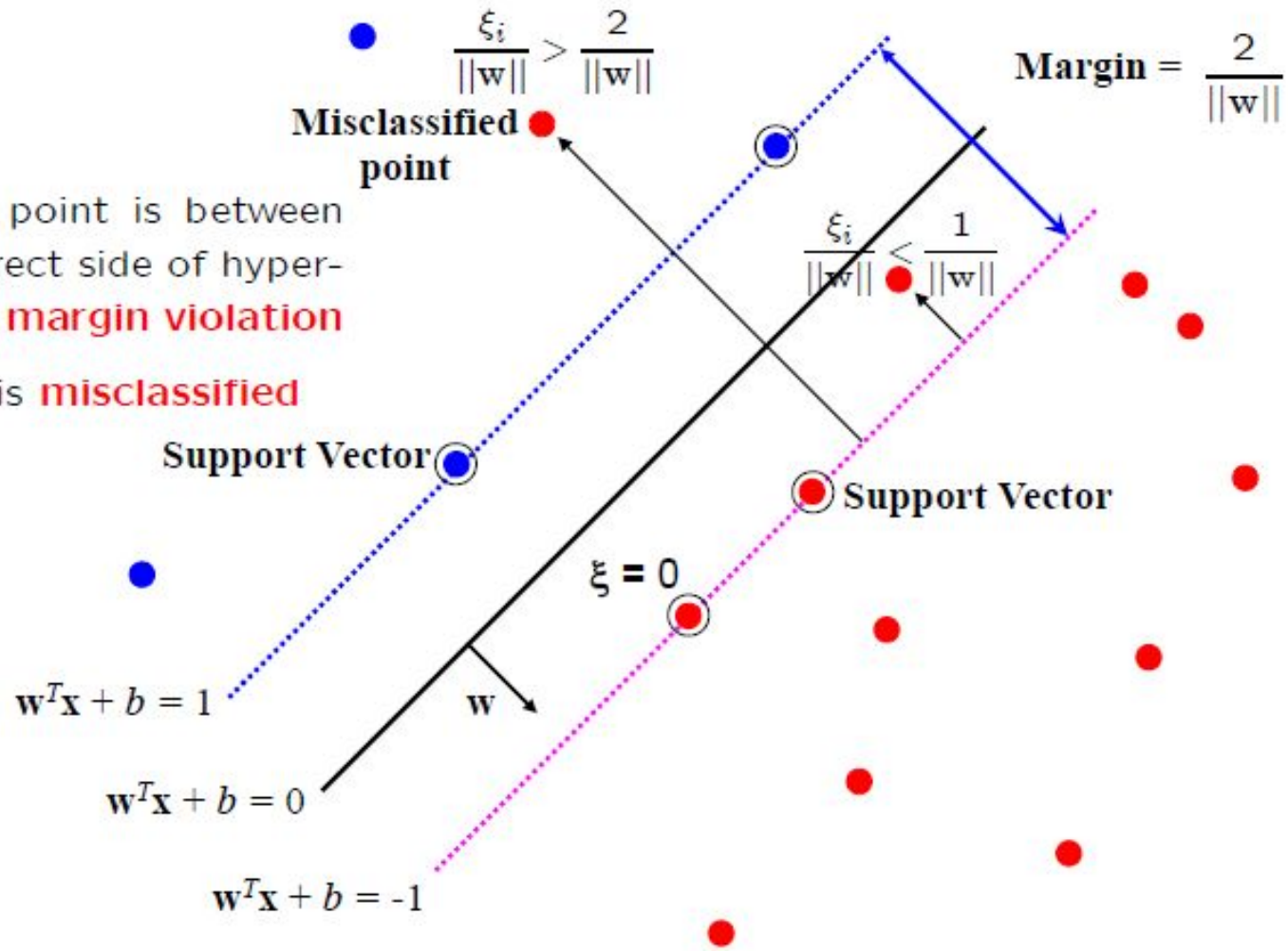
- Nature of data : Non linear boundary



Slack Variable

$\xi_i \geq 0$

- for $0 < \xi \leq 1$ point is between margin and correct side of hyper-plane. This is a **margin violation**
- for $\xi > 1$ point is **misclassified**



https://www.robots.ox.ac.uk/~az/lectures/ml/lect2.pdf

Noise in Data

$$\text{Minimize} : \Phi(\mathbf{w}) = \tfrac{1}{2}\mathbf{w}^T\mathbf{w}$$
$$\text{Subject to} : d_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 \quad \forall i$$

↓

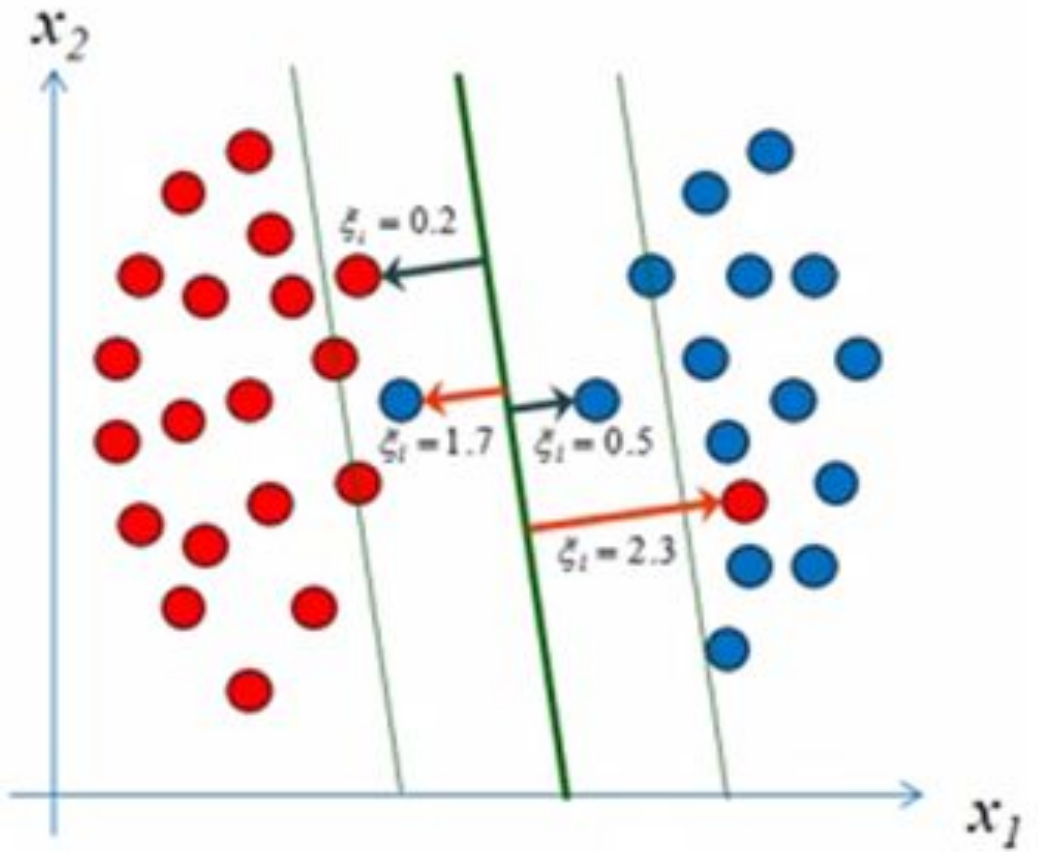
Introduce slack variables $\xi_i \geq 0$

↓

$$\text{Minimize} : \Phi(\mathbf{w}) = \tfrac{1}{2}\mathbf{w}^T\mathbf{w}$$
$$\text{Subject to} : d_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 - \xi_i \quad \forall i$$

Also minimize training error $\sum_{i=1}^{N} \mathrm{I}(\xi_i \geq 1)$ or $\sum_{i=1}^{N} \xi_i$

$$\text{Minimize} : \Phi(\mathbf{w}) = \tfrac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{i=1}^{N} \xi_i$$
$$\text{Subject to} : d_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 - \xi_i ; \quad \xi_i \geq 0 , \quad \forall i$$



Dual Form dengan Slack

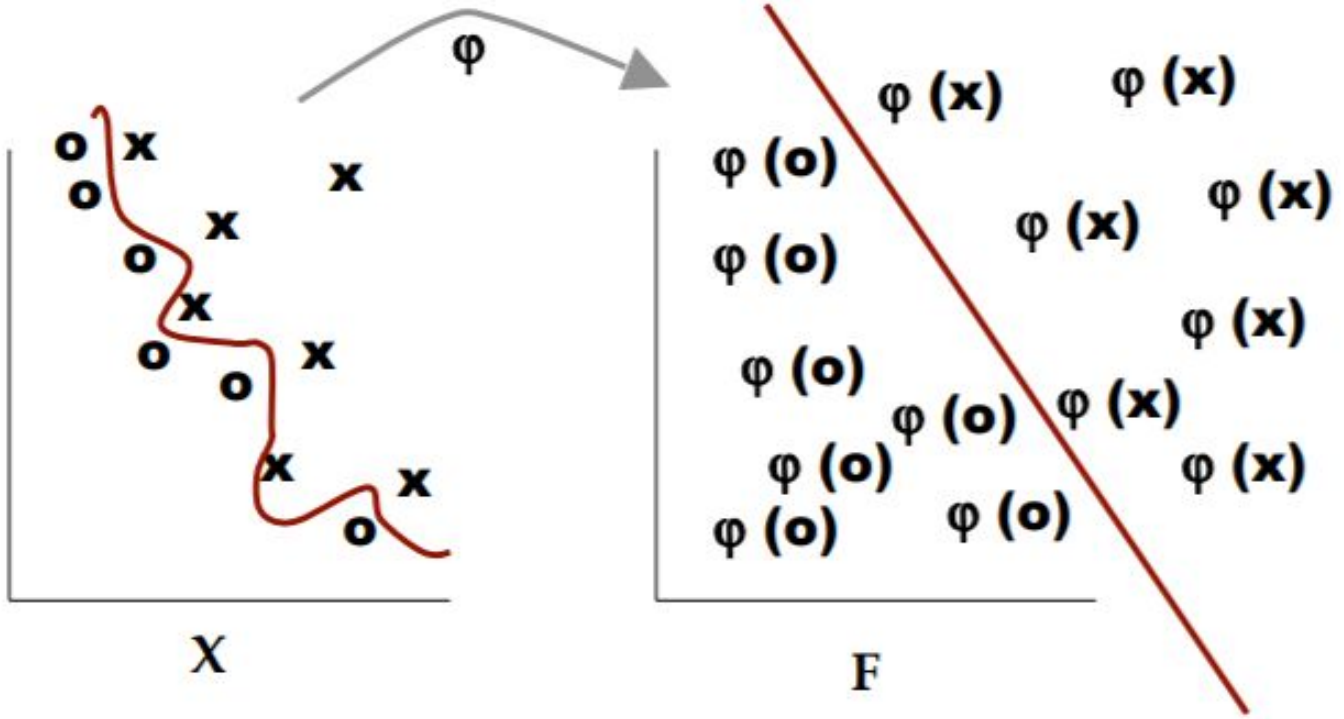
- Forming the Lagrangian and converting to dual, we get:

$$Q(\boldsymbol{\alpha}) = \sum_{i=1}^{N} \alpha_i - \tfrac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j d_i d_j \mathbf{x_i}^T \mathbf{x}_j$$

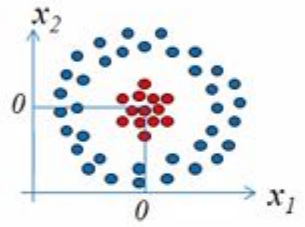
$$\text{Subject to} \quad 0 \le \alpha_i \le C \quad \forall_i \quad \text{and} \quad \sum_{i=1}^{N} \alpha_i d_i = 0$$

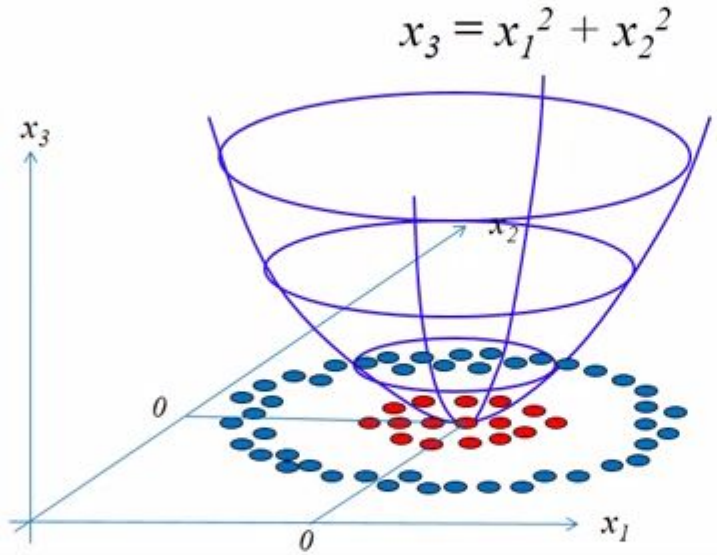
- Note that neither the slack variables, nor their Lagrange multipliers appear in the dual.
- The only change is the additional constraint on α_i
- The parameter C controls the relative weight between training error and the VC dimension.

Non-Linear Boundary Transformation



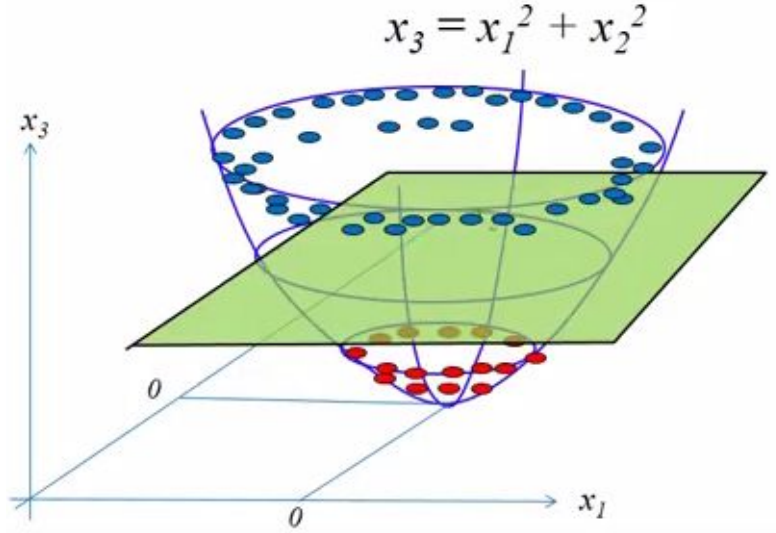
Non-Linear Boundary Transformation





Lower Dimension: Non-linearly Separable Data

Higher Dimension: Linearly Separable Data

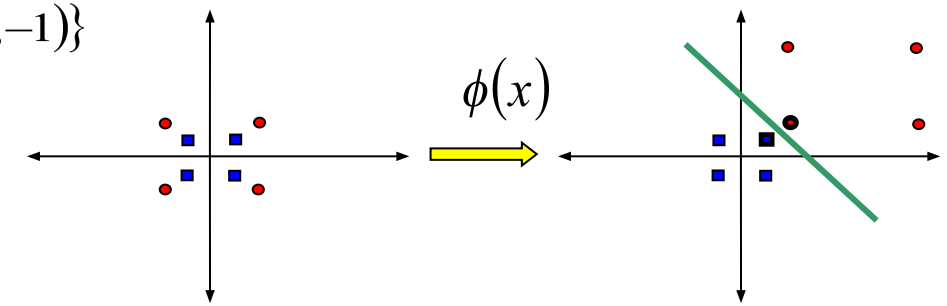


SVM pada *non-linearly separable data (2)*

- Contoh:

Misalkan dataset

- Data kelas positif $\{(2,2),(2,-2),(-2,2),(-2,-2)\}$
- Data kelas negatif $\{(1,1),(1,-1),(-1,1),(-1,-1)\}$



$$\phi(x_1, x_2) = \begin{cases} \sqrt{{x_1}^2 + {x_2}^2} > 2 \rightarrow (4 - x_2 + |x_1| - |x_2|, 4 - x_1 + |x_1| - |x_2|) \\ \sqrt{{x_1}^2 + {x_2}^2} \leq 2 \rightarrow (x_1, x_2) \end{cases}$$

- Dengan transformasi diperoleh
  - Data kelas positif $\{(2,2),(6,2),(6,6),(2,6)\}$
  - Data kelas negatif $\{(1,1),(1,-1),(-1,1),(-1,-1)\}$

SVM pada *non-linearly separable data (3)*

Klasifikasi: $f(x) = \sum_{i=1}^{ns} \alpha_i y_i x_i . x + b$ ➡ $f(x) = \sum_{i=1}^{ns} \alpha_i y_i \phi(x_i) \phi(x) + b$

- Sulit untuk mengetahui $\phi(x)$ dan feature space biasanya memiliki dimensi yang lebih besar
- Solusinya "*kernel trick*", yang perlu diketahui adalah $K(x_i, x) = \phi(x_i) \phi(x)$
- Dengan fungsi K (fungsi Kernel), maka fungsi $\phi(x)$ tidak perlu diketahui

Klasifikasi: $f(x) = \sum_{i=1}^{ns} \alpha_i y_i K(x_i, x) + b$

Fungsi Kernel yang umum digunakan:

Linear Kernel $K(x_i, x_j) = x_i^T x_j$

Polynomial kernel $K(x_i, x_j) = (\gamma . x_i^T x_j + r)^p, \gamma > 0$

RBF kernel $K(x_i, x_j) = \exp(-\gamma |x_i - x_j|^2), \gamma > 0$

Sigmoid kernel $K(x_i, x_j) = \tanh(\gamma x_i^T x_j + r)$

Example

- Suppose we have 5 (1D) data points
  - x_1=1, x_2=2, x_3=4, x_4=5, x_5=6,
  - with 1, 2, 5 as class 1 and 3, 4 as class 2 ⇒ y_1=1, y_2=1, y_3=-1, y_4=-1, y_5=1
- We use the polynomial kernel of degree 2
  - $K(x,y) = (xy+1)^2$
  - C is set to 100
- We first find a_i (i=1, …, 5) by

$$\text{max.} \sum_{i=1}^{5} \alpha_i - \frac{1}{2} \sum_{i=1}^{5} \sum_{j=1}^{5} \alpha_i \alpha_j y_i y_j (x_i x_j + 1)^2$$

$$\text{subject to } 100 \geq \alpha_i \geq 0, \sum_{i=1}^{5} \alpha_i y_i = 0$$

Example

- x_1=1, x_2=2, x_3=4, x_4=5, x_5=6,
- y_1=1, y_2=1, y_3=-1, y_4=-1, y_5=1

- By using a QP solver, we get
  - α_1=0, α_2=2.5, α_3=0, α_4=7.333, α_5=4.833
  - Note that the constraints are indeed satisfied
  - The support vectors are {x_2=2, x_4=5, x_5=6}
- The discriminant function is

α_5 y_5 $K(z, x_5)$

$$f(z) = 2.5(1)(2z+1)^2 + 7.333(-1)(5z+1)^2 + 4.833(1)(6z+1)^2 + b$$
$$= 0.6667z^2 - 5.333z + b$$

- *b* is recovered by solving f(2)=1 or by f(5)=-1 or by f(6)=1, as x_2 and x_5 lie on the line $\phi(\mathbf{w})^T\phi(\mathbf{x}) + b = 1$ and x_4 lies on the line $\phi(\mathbf{w})^T\phi(\mathbf{x}) + b = -1$
- All three give b=9

$$\rightarrow \quad f(z) = 0.6667z^2 - 5.333z + 9$$

04 SVM for Multi-class Data

IF3170 Artificial Intelligence

Modul: Supervised Learning

04 SVM for Multi-class Data

IF3170 - Inteligensi Artifisial

Dr. Fariska Z. Ruskanda, S.T., M.T.
(fariska@informatika.org)

KK IF -Teknik Informatika - STEI ITB

Multi Class SVM

- SVM dirancang untuk mengklasifikasikan data ke dalam dua kelas → SVM biner
- Untuk klasifikasi multiclass gabungkan beberapa SVM biner.
- Ada 3 metode umum:
  - *One-against-all*
  - *One-against-one*
  - DAGSVM

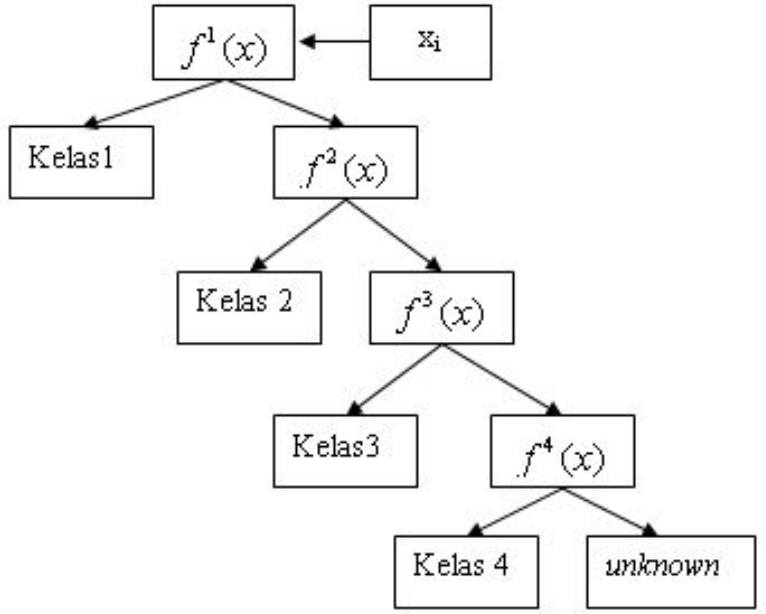
Multi Class SVM→*one-against-all*

- Dibangun k model klasifikasi (k adalah jumlah kelas)
- Setiap pelatihan model menggunakan data dari semua kelas
- Prediksi kelas data umumnya seperti gambar sebelah kanan atau berdasarkan nilai maksimum f(X)

Pembelajaran:

$y_i = 1$	$y_i = -1$	Hipotesis
Kelas 1	Bukan kelas 1	$f^1(x) = (w^1)x + b^1$
Kelas 2	Bukan kelas 2	$f^2(x) = (w^2)x + b^2$
Kelas 3	Bukan kelas 3	$f^3(x) = (w^3)x + b^3$
Kelas 4	Bukan kelas 4	$f^4(x) = (w^4)x + b^4$

Klasifikasi:



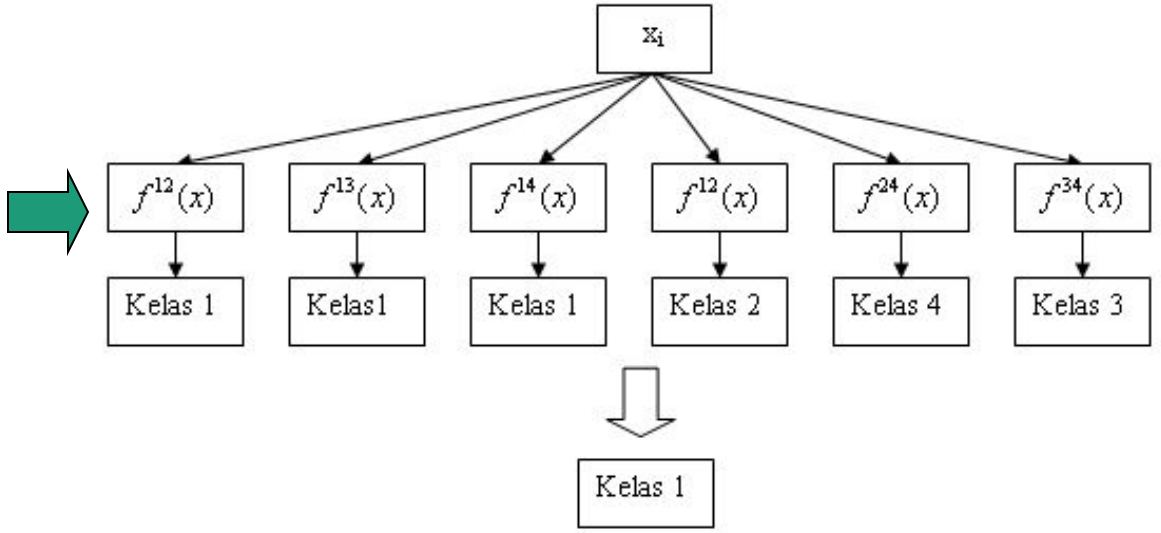
Multi Class SVM→*one-against-one*

- Dibangun $\frac{k(k-1)}{2}$ model klasifikasi (k adalah jumlah kelas)
- Setiap pelatihan model menggunakan data dari dua kelas
- Prediksi Kelas data dengan metode voting

Pembelajaran:

$y_i = 1$	$y_i = -1$	Hipotesis
Kelas 1	Kelas 2	$f^{12}(x) = (w^{12})x + b^{12}$
Kelas 1	Kelas 3	$f^{13}(x) = (w^{13})x + b^{13}$
Kelas 1	Kelas 4	$f^{14}(x) = (w^{14})x + b^{14}$
Kelas 2	Kelas 3	$f^{23}(x) = (w^{23})x + b^{23}$
Kelas 2	Kelas 4	$f^{24}(x) = (w^{24})x + b^{24}$
Kelas 3	Kelas 4	$f^{34}(x) = (w^{34})x + b^{34}$

Klasifikasi:



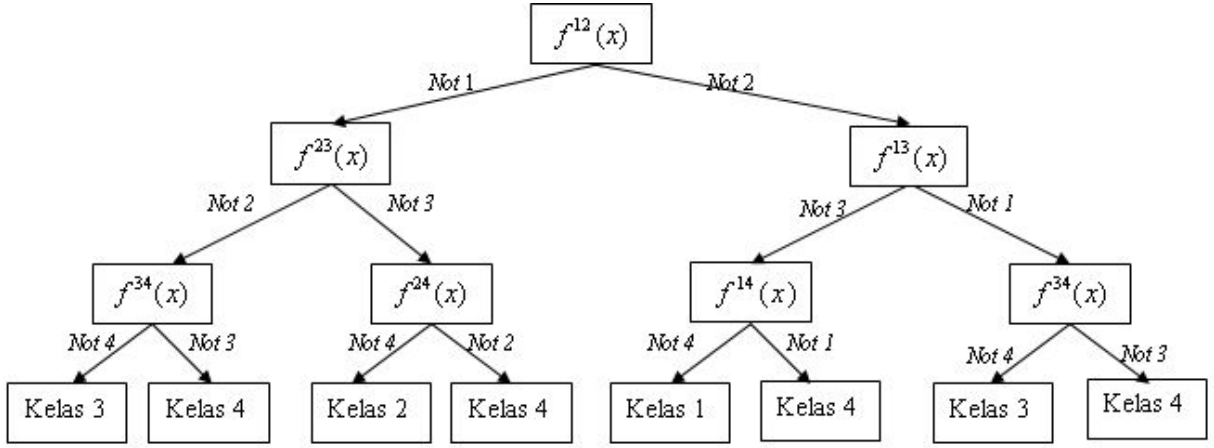
Multi Class SVM→DAG (*Directed Acyclic Graph*) SVM

- Proses pembelajaran sama dengan One-Against-One

Pembelajaran:

$y_i = 1$	$y_i = -1$	Hipotesis
Bukan Kelas 2	Bukan Kelas 1	$f^{12}(x) = (w^{12})x + b^{12}$
Bukan Kelas 3	Bukan Kelas 1	$f^{13}(x) = (w^{13})x + b^{13}$
Bukan Kelas 4	Bukan Kelas 1	$f^{14}(x) = (w^{14})x + b^{14}$
Bukan Kelas 3	Bukan Kelas 2	$f^{23}(x) = (w^{23})x + b^{23}$
Bukan Kelas 4	Bukan Kelas 2	$f^{24}(x) = (w^{24})x + b^{24}$
Bukan Kelas 4	Bukan Kelas 3	$f^{34}(x) = (w^{34})x + b^{34}$

Klasifikasi:



SVM Software

- LibSVM
  - Umum, dapat digunakan untuk berbagai aplikasi, tidak dioptimasi untuk SVM Linier (fungsi kernel linier)
  - Mendukung Multi Class SVM One-Against-One, One Class SVM dan Support Vector Regression
  - C++, Java, Phyton, C#, Matlab
- LibLinear
  - versi LibSVM yang dioptimasi untuk kernel linier
  - Mendukung Multi Class SVM One-Against-All
  - C++
- SVMLight
  - C++, populer dalam aplikasi klasifikasi teks
- Info tambahan tentang SVM:
  - http://www.kernel-machines.org
  - www.svms.org
  - http://agbs.kyb.tuebingen.mpg.de
  - http://support-vector-machines.org

Terima Kasih