Machine Learning
Teknik Informatika - UNIKOM

Learning –basic concept

Learning

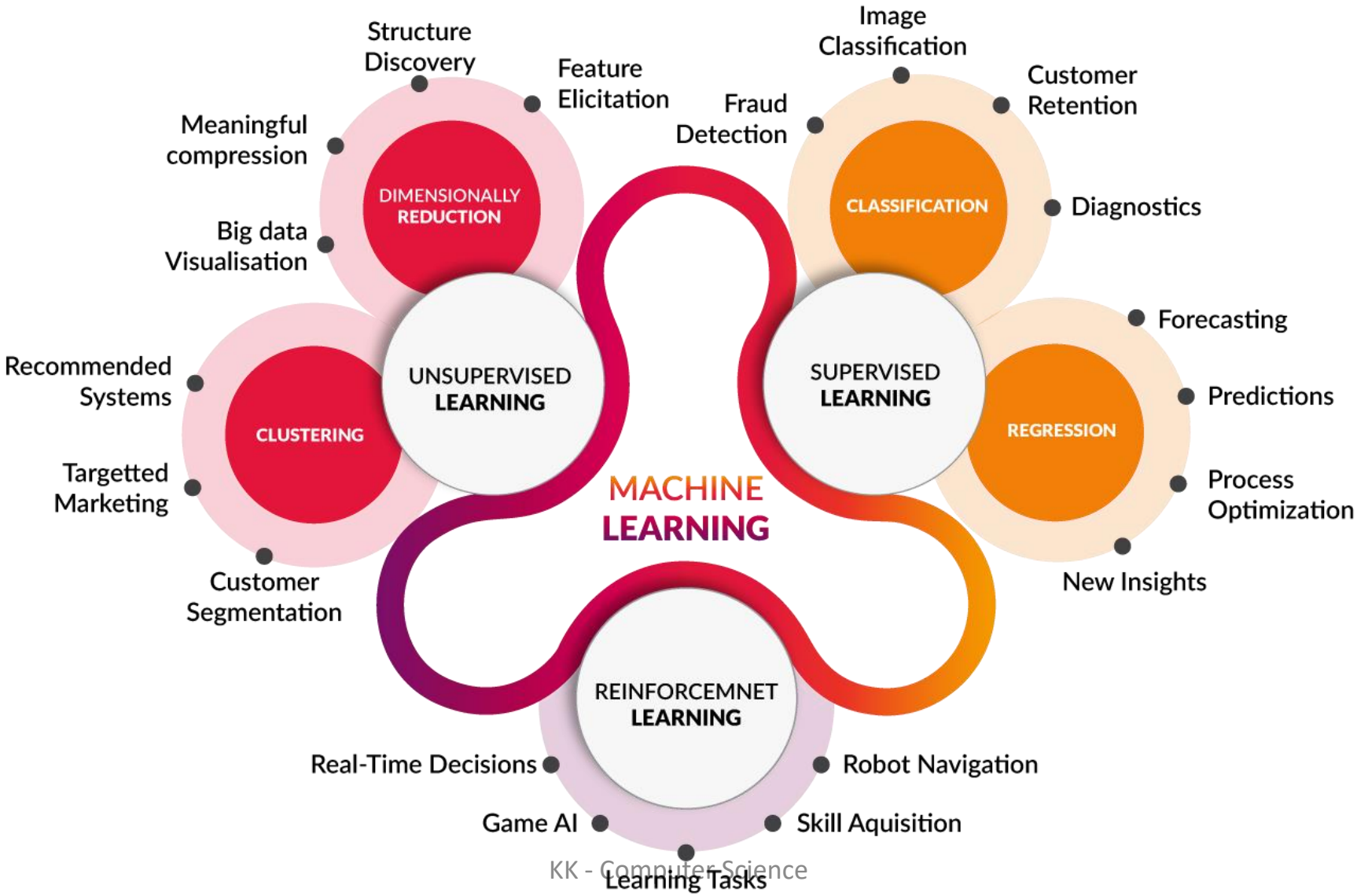
- Learning adalah hal penting untuk lingkungan yg tidak diketahui

- Learning berguna untuk metode konstruksi sistem
  - Mendekatkan agen dengan realitas daripada mendeskripsikan lingkungan untuk agen
- Learning memodifikasi mekanisme keputusan agen untuk meningkatkan performa

Learning objectives

- Persepsi agen seharusnya digunakan bukan hanya untuk beraksi, tapi juga untuk meningkatkan performa di masa datang.
- Tugas-tugas belajar untuk sebuah agen:
  - Kondisi yang ingin dicapai untuk sebuah aksi
  - Perubahan lingkungan yang berpengaruh
  - Nilai untuk setiap kondisi
  - Kondisi yang memiliki nilai tinggi (rendah) → solusi
  - Informasi mana yang relevan
- Learning task – estimations of functions $y=f(\mathbf{x})$: $\mathbf{x} \rightarrow y$

Taxonomi of Machine Learning



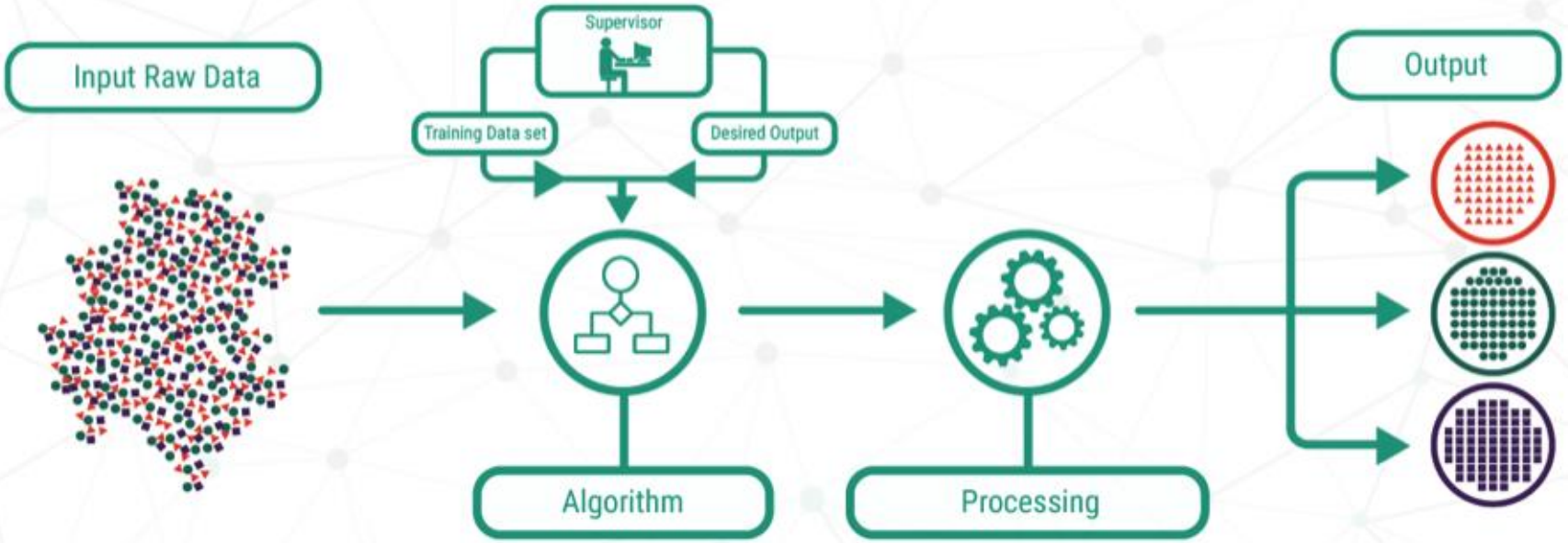
Supervised Learning

Supervised learning is the machine learning task of inferring a function from labeled training data

Mehryar mohri

(foundations of machine learning, 2012, MIT Press)

SUPERVISED LEARNING



- Supervised learning menggunakan training data yang sudah diberi label untuk mempelajari mapping function, dari input variable (x) ke output variable (y).

$$y = f(x)$$

- Beberapa algoritma supervised learning
  - Linear regression
  - Logistic regression
  - Multiclass classification
  - Support vector machines

- Classification
  - Classification bertujuan untuk memprediksi outcome dari input yang diberikan , dimana output variable berbentuk kategori-kategori. Misal pria/wanita, sakit/sehat, tinggi/rendah
- Regression
  - Regression bertujuan untuk memprediksi outcome dari input, dimana output variable berbentuk nilai actual (real values), misal tinggi badan seseorang, curah hujan.

Algoritma Supervised Learning

- Decision tree
- Naïve bayes classifier
- Artificial neural network
- Support vector machine
- Linear regression
- Logistic regression
- CART
- KNN (K- Nearest Neighbor)

Algoritma kNN
(k-Nearest Neighbor)

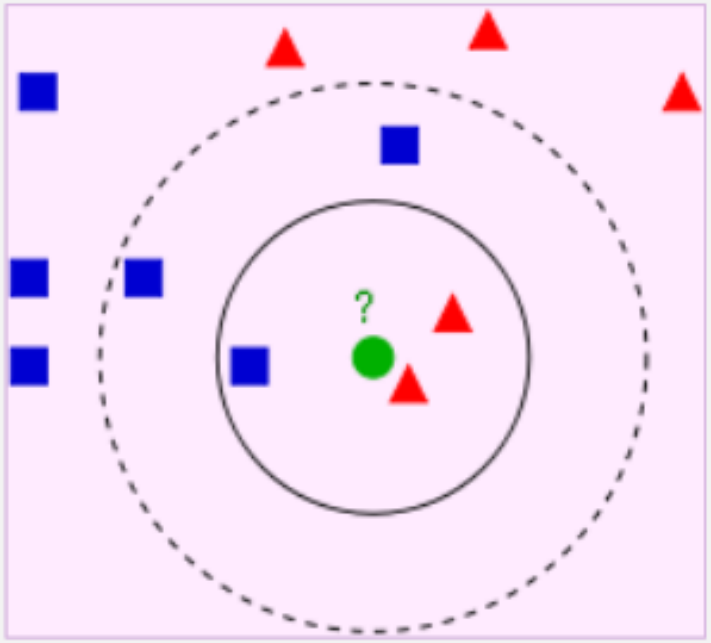
Deskripsi kNN

- KNN adalah sebuah metode klasifikasi terhadap sekumpulan data berdasarkan pembelajaran data yang sudah terklasifikasikan sebelumya.
- Termasuk dalam *supervised learning*, dimana hasil *query instance* yang baru diklasifikasikan berdasarkan mayoritas kedekatan jarak dari kategori yang ada dalam KNN.

Deskripsi kNN

- Diberikan titik query, akan ditemukan sejumlah k obyek atau (titik training) yang paling dekat dengan titik query.
- Klasifikasi menggunakan voting terbanyak diantara klasifikasi dari k obyek
- Algoritma k-nearest neighbor (KNN) menggunakan klasifikasi ketetanggaan sebagai nilai prediksi dari query instance yang baru.

- Jika k=3 maka lingkaran hijau akan masuk dalam kategori segitiga merah
- Jika k=5 maka lingkaran hijau akan masuk dalam kategori segiempat biru.



Ukuran Jarak

- Dekat atau jauhnya tetangga biasanya dihitung berdasarkan *Euclidean Distance.*

$$D(a,b) = \sqrt{\sum_{k=1}^{d}(a_k - b_k)^2},$$

- Dimana D(a,b) adalah jarak skalar dari dua buah vektor data a dan b yang berupa matrik berukuran d dimensi.

Beberapa macam jarak yang dapat digunakan

- Jarak Euclidean

$$d(x,y) = \|x - y\| = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$

- Jarak Manhattan atau Cityblock

$$d(x,y) = \sum_{i=1}^{n} (|x_i - y_i|)$$

- Jarak Minkowski

$$d(x,y) = \|x - y\|_q = (\sum |x - y|^q)^{\frac{1}{q}}$$

- Jarak Mahalanobis

$$d(\vec{x},\vec{y}) = \sqrt{(\vec{x} - \vec{y})^T S^{-1} (\vec{x} - \vec{y})}$$

Kelebihan

Kelebihan KNN:

- Simpel
- Efektif jika data besar
- Intuitif
- Peforma cukup baik
- Tahan terhadap data latih yang noisy

Kekurangan

Kekurangan KNN :

- Waktu komputasi tinggi jika data latih besar. Disebabkan oleh semua data diukur jaraknya untuk setiap data uji.
- Sangat sensitive dengan ciri yang redundan atau tidak relevan. Ditanggulangi dengan seleksi ciri atau pembobotan ciri.
- Tidak diketahui perhitungan jarak apa yang paling sesuai untuk dataset tertentu.

Algoritma

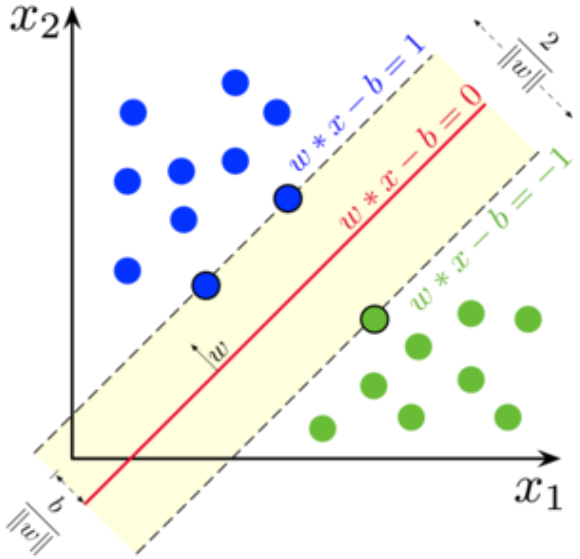
1. Menentukan parameter k (jumlah tetangga paling dekat).
2. Menghitung kuadrat jarak eucliden objek terhadap data training yang diberikan.
3. Mengurutkan hasil no 2 secara ascending
4. Mengumpulkan kategori Y (Klasifikasi nearest neighbor berdasarkan nilai k)
5. Dengan menggunakan kategori nearest neighbor yang paling mayoritas maka dapat dipredisikan kategori objek .

Support Vector Machine

Kenapa SVM?

- SVM dapat menemukan solusi optimum global
- Dalam setiap kasus, memiliki solusi yang hampir selalu sama.

Support Vector Classifier



https://upload.wikimedia.org/wikipedia/commons/thumb/7/72/SVM_margin.png/300px-SVM_margin.png

Inti dari SVC

- mencari hyperplane yang terbaik
- memaksimalkan margin
- Penentuan suatu data termasuk kelas yang mana dengan cara

$$f(\overline{x_i}) = \text{sgn}(w^T \overline{x_i} + b)$$

Pelatihan

Untuk menemukan$\alpha_1, \alpha_2, \ldots, \alpha_n$

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

Dengan syarat:

1. $\sum_{i=1}^{n} \alpha_i y_i = 0$
2. $\alpha_i \geq 0$, untuk setiap α_i

$w = \sum \alpha_i y_i \overline{x_i}$ dan $b = y_k - \overline{w}^T \overline{x_k}$ untuk setiap $\overline{x_k}$ sedemikian hingga $\alpha_k \neq 0$

Untuk setiap α_i yang tidak nol, mengidentifikasikan $\bar{x}$ adalah support vector.

Pengujian SVC

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

Jika $f(\bar{x}) > 0$ maka kelas akan sama dengan +1

Jika $f(\bar{x}) < 0$ maka kelas akan sama dengan -1

Sequential Minimal Optimization

- Pilih α_i dan α_j
- Tentukan nilai L dan H sehingga $L \leq \alpha_j \leq H$
- $L = \begin{cases} \max(0, \alpha_j - \alpha_i) & jika\ y_i = y_j \\ \max(0, \alpha_i + \alpha_j - C) & jika\ y_i \neq y_j \end{cases}$
- $H = \begin{cases} \min(C, C + \alpha_j - \alpha_i) & jika\ y_i = y_j \\ \min(C, \alpha_i + \alpha_j) & jika\ y_i \neq y_j \end{cases}$
- www.stanford.edu/class/cs229/materials/smo.pdf

SMO 2

- Update α_j jika α_j tidak terletak antara L dan H

$$\alpha_j = \alpha_j - \frac{y^{(j)}\left(E_i - E_j\right)}{\eta}$$

Dimana

$$E_k = f\left(x^{(k)}\right) - y^{(k)}$$

$$\eta = 2\left\langle x^{(i)}, x^{(j)}\right\rangle - \left\langle x^{(i)}, x^{(i)}\right\rangle - \left\langle x^{(j)}, x^{(j)}\right\rangle$$

$$f\left(x^{(k)}\right) = \sum_{i=1}^{m} \alpha_i y_i x^{(i)} x^{(k)} + b$$

Untuk menghitung $\left\langle x^{(i)}, x^{(j)}\right\rangle$ bisa menggunakan fungsi kernel

SMO 3

- Update α_j jika α_j terletak antara L dan H

$$\alpha_j = \begin{cases} H & ,jika\ \alpha_j > H \\ \alpha_j & ,jika\ L \leq \alpha_j \leq H \\ L & jika\ \ \alpha_j < L \end{cases}$$

SMO 4

- Update α_i

$$\alpha_i = \alpha_i + y^{(i)} y^{(j)} \left(\alpha_j^{(old)} - \alpha_j \right)$$

SMO b

- Setelah α_i dan α_j diupdate maka hitung

$$b_1 = b - E_i - y^{(i)}\left(\alpha_i - \alpha_i^{(old)}\right)\left\langle x^{(i)}, x^{(i)}\right\rangle - y^{(j)}\left(\alpha_j - \alpha_j^{(old)}\right)\left\langle x^{(i)}, x^{(j)}\right\rangle$$

Dan

$$b_2 = b - E_j - y^{(i)}\left(\alpha_i - \alpha_i^{(old)}\right)\left\langle x^{(i)}, x^{(j)}\right\rangle - y^{(j)}\left(\alpha_j - \alpha_j^{(old)}\right)\left\langle x^{(j)}, x^{(j)}\right\rangle$$

Maka b diupdate dengan cara

$$b = \begin{cases} b_1 & ,jika\ 0 < \alpha_i < C \\ b_2 & ,jika\ 0 < \alpha_j < C \\ \dfrac{b_1 + b_2}{2} & lainnya \end{cases}$$

- Input: C, tol, max_passes, data training $\{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \ldots, (x^{(m)}, y^{(m)})\}$
- Output: $\alpha \in R^m$ dan $b \in R$
- Inisialisasi $\alpha_i = 0, \forall i$ dan $b = 0$
- Inisialisasi $passes = 0$
- Selama $(passes < maxpasses)$
- Num_changed_alpha=0
- Untuk i = 1..0
- Hitung $E_i = f(x^{(i)}) - y^{(i)}$

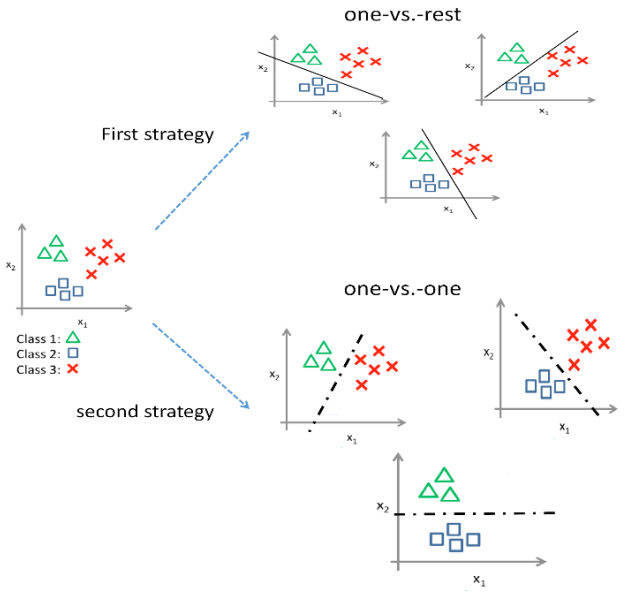
- Jika $y^{(i)}E_i < -tol \ \&\& \ \alpha_i < C$ atau $y^{(i)}E_i > tol \ \&\& \ \alpha_i > 0$
- Pilih $j \neq i$ secara random
- Hitung $E_j = f\left(x^{(j)}\right) - y^{(j)}$
- $\alpha_i^{(old)} = \alpha_i$ dan $\alpha_j^{(old)} = \alpha_j$
- Hitung L dan H
- Jika L=H maka i=i+1
- Hitung η
- Jika $\eta \geq 0$ lanjut i = i +1

- Hitung α_j
- Jika $\left|\alpha_j - \alpha_j^{(old)}\right| < 10^{-5}$
- i=i+1
- Hitung α_i
- Hitung b_1 dan b_2
- Hitung b
- Num_changed_alphas = num_changed_alpha+1
- End if
- End for
- Jika num_changed = 0
- Passes = passes +1
- Else
- Passes = 0
- End while

MULTICLASS

- SVC hanya dapat mengklasifikasikan untuk 2 kelas, tetapi dapat dimodifikasi, yaitu multiclass SVM

1. One against All
2. One against One



https://www.researchgate.net/profile/Shervan_Fekri_Ershad2/publication/332370066/figure/fig22/AS:746765703720962@1555054226099/Sample-example-of-a-multiclass-support-vector-machine-The-SVM-algorithms-applied-to-a.ppm

Perbandingan

One Against All

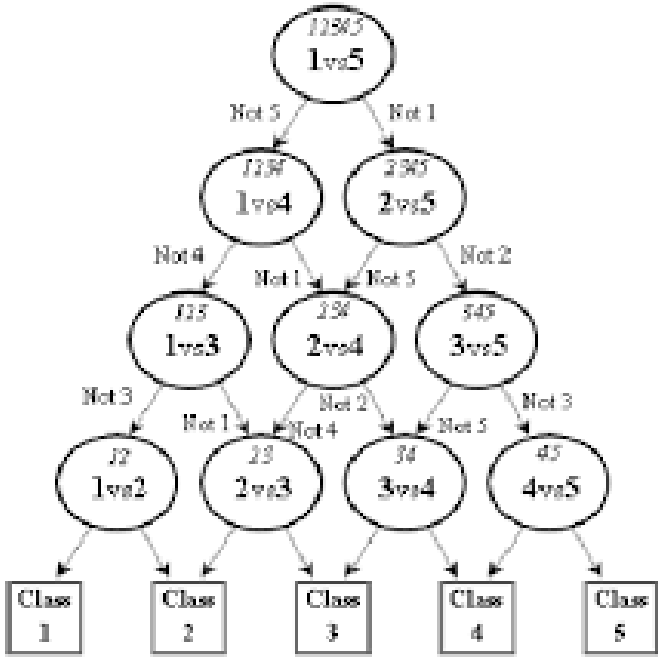
- Banyak hyperplane yang harus dibanyak sebanyak kelas yang akan diklasifikasi
- Kerugian terlalu banyak daerah yang tidak terdefinisi
- Penentuan kelas dengan cara maksimum nilai $f(\bar{x})$

One Against One

- Banyak hyperplane yang harus dibangun sebanyak $\frac{k(k+1)}{2}$
- Daerah yang tidak terdefinisi hanya sedikit
- Penentuan kelas dengan cara voting

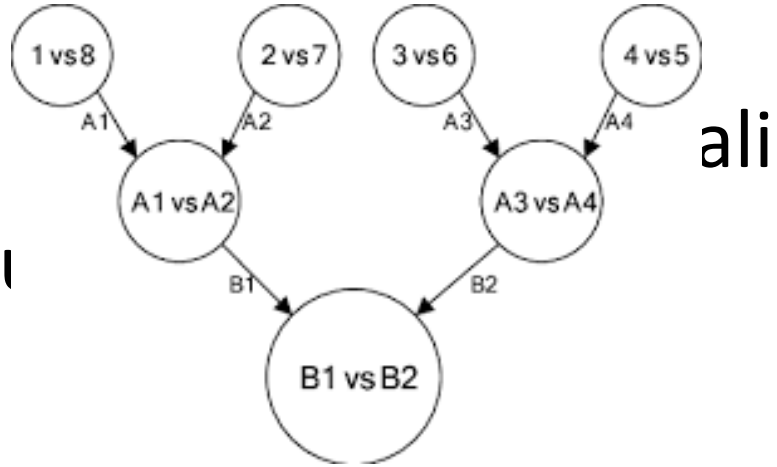
Decesion Directed Acyclic Graph

- Jumlah hyperplane yang digunakan sebanyak one against one
- Penentuan kelas menggunakan system gugur

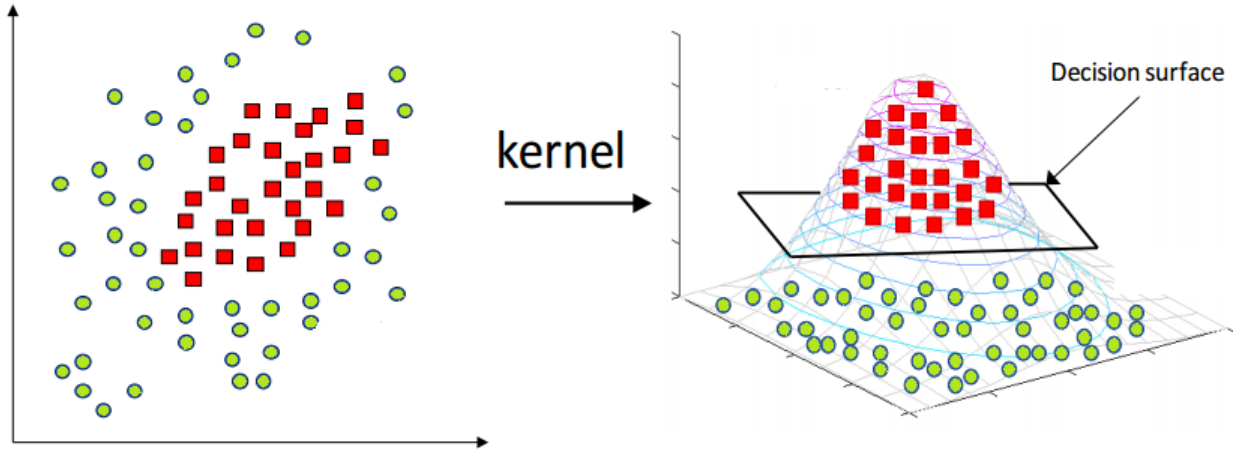


Adaptive Directed Acyclic Graph

- Perbaikan dari DDAG, dengan cara membalikan grafik, missal ada 8 kelas yang akan diklasifikasi maka diawal akan ada 4 klasifikasi
- Kelas yang terpilih telah ali
- Lebih tahan terhadap uru



Kernel



https://miro.medium.com/max/1676/0*5AtpuJ7X08V4zc9f.png

Jenis-jenis Kernel

1. Kernel Linier, $K(x, x_k) = x_k^T x$
2. Kernel Polynomial, $K(x, x_k) = \left(x_k^T x + 1\right)^d$
3. Kernel Gaussian (radial basis function, RBF),

   $K(x, x_k) = \exp\left(\frac{-\|x - x_k\|^2}{\sigma^2}\right)$
4. Kernel sigmoid, $K(x, x_k) = \tanh\left(\kappa x_k^T x + \theta\right)$

Dimana $\tanh(\alpha) = \frac{e^{2\alpha} - 1}{e^{2\alpha} + 1}$