Model Linear untuk Klasifikasi

Dr. rer. nat. Hendri Murfi

Intelligent Data Analysis (IDA) Group

Departemen Matematika, Universitas Indonesia – Depok 16424

Telp. +62-21-7862719/7863439, Fax. +62-21-7863439, Email. hendri@ui.ac.id

Machine Learning

Input	Model/Metode	Output
x_1, x_2, :, x_D	$y(\mathbf{x},\mathbf{w})$	t

Diberikan data pelatihan (*training data*), yaitu $\mathbf{x}_i$ dan/atau $\mathbf{t}_i$, i = 1 sd N

- *Preprocessing*: pemilihan/ekstraksi fitur dari data, misal $\mathbf{x}_i = (x_1, x_2, .., x_D)^T$
- *Learning*: penentuan parameter metode, misal **w**, berdasarkan data pelatihan
- *Testing*: pengujian metode dengan data baru. Data penguji (*testing data*) tersebut harus dilakukan *preprocessing* yang sama dengan data pembelajaran sebelum dieksekusi oleh metode

Learning

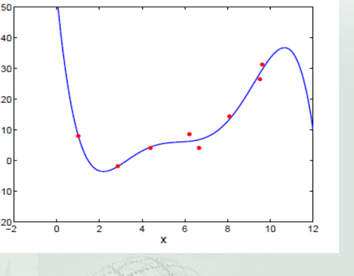
Diberikan data pelatihan $\boldsymbol{x}_i$, *i = 1 sd N,* dan/atau $\boldsymbol{t}_i$, *i = 1 as N*

- Supervised Learning. Data pelatihan disertai target, yaitu {$\mathbf{x}_i$, $\mathbf{t}_i$}, i = 1 sd N. Tujuan pembelajaran adalah membangun model yang dapat menghasilkan output yang benar untuk suatu data input, misal untuk regresi, klasifikasian, regresi ordinal, ranking, dll

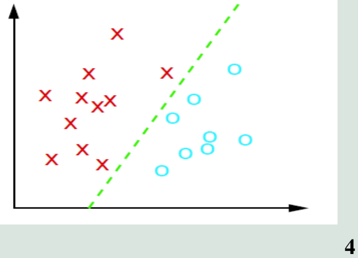
- Unsupervised Learning. Data pelatihan tidak disertai target, yaitu $\mathbf{x}_i$, i = 1 sd N. Tujuan pembelajaran adalah membagun model yang dapat menemukan komponen/variabel/fitur tersembunyi pada data pelatihan, yang dapat digunakan untuk: pengelompokan (*clustering*), reduksi dimensi (*dimension reduction*), rekomendasi, dll

Supervised Learning

- Regresi
  - Nilai output t_i bernilai kontinu (riil)
  - Bertujuan memprediksi output dari data baru dengan akurat

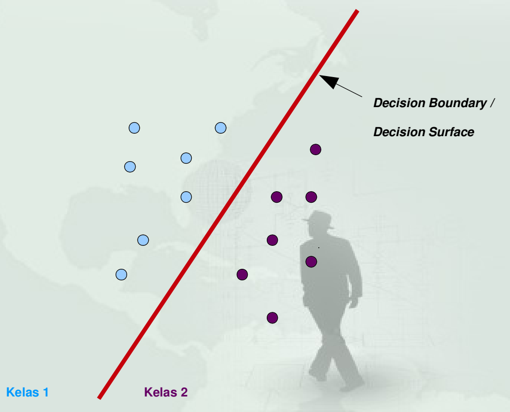


- Klasifikasi
  - Nilai output t_i bernilai diskrit (kelas)
  - Bertujuan mengklasifikasi data baru dengan akurat



Klasifikasi

Masalah Dua Kelas



- Diberikan vektor input **x** berdimensi *D*, bagaimana mengklasifikasikannya pada salah satu kelas
- Salah satu cara adalah mencari batas (*decision boundary/decision surface*) antara area kelas (*decision region*) bedasarkan data pembelajaran
- Misal dengan menggunakan fungsi linear (model linear) yang dikenal juga dengan istilah fungsi diskriminan (*discriminant function*)

Klasifikasi

Model Linear

- Model linear adalah kombinasi linear dari fungsi nonlinear dari variabel input:

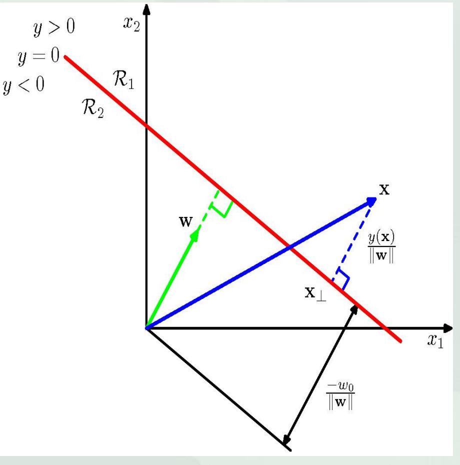
$$y(\mathbf{x}, \mathbf{w}) = \sum_{j=0}^{M-1} w_j \phi_j(\mathbf{x}) = \mathbf{w}^{\mathrm{T}} \boldsymbol{\phi}(\mathbf{x})$$

Dimana $\mathbf{x} = (x_1, x_2, ..., x_D)^T$ adalah variabel input, dan $\mathbf{w} = (w_0, w_1, ..., w_D)^T$ adalah parameter, $\phi(\mathbf{x})$ adalah *fungsi basis*, *M* adalah jumlah total parameter dari model

- Biasanya, $\phi_0(\mathbf{x}) = 1$, sehingga w_0 berfungsi sebagai bias
- Ada banyak pilihan yang mungkin untuk fungsi basis $\phi(\mathbf{x})$, misal fungsi linear, fungsi polinomial, fungsi gaussian, fungsi sigmoidal, dll

Klasifikasi

Hyperplane



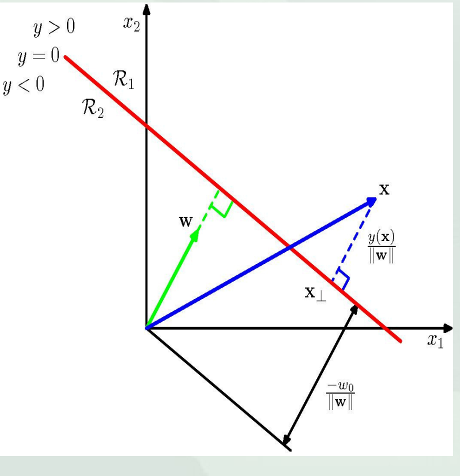
- Bentuk model linear yang paling sederhana untuk fungsi diskriminan linear:

  $$y(\boldsymbol{x}) = \boldsymbol{w}^T\boldsymbol{x} + w_0$$

  Dimana $\boldsymbol{x}$ adalah vektor input, $\boldsymbol{w}$ adalah vektor bobot dan w_0 adalah bias.
- Sehingga, *decision boundary* adalah $y(\boldsymbol{x})=0$, yaitu suatu *hyperplane* berdimensi *(D-1)*
- Suatu vektor input $\boldsymbol{x}$ akan diklasifikasikan ke kelas 1 (R_1) jika $y(\boldsymbol{x})\geq 0$, dan kelas 2 (R_2) jika $y(\boldsymbol{x})<0$

Klasifikasi

Sifat-Sifat Hyperplane



- Jika $\boldsymbol{x}_A$ dan $\boldsymbol{x}_B$ terletak pada *decision boundary* (DS), maka $y(\boldsymbol{x}_A)=y(\boldsymbol{x}_B)=0$ atau $\mathbf{w}^T(\boldsymbol{x}_A-\boldsymbol{x}_B)=0$, sehingga **w** tegak lurus terhadap semua vektor di DS. Dengan kata lain **w** menentukan orientasi dari DS
- Jarak titik awal ke DS adalah $-w_0/||\mathbf{w}||$. Dengan kata lain w_0 menentukan lokasi DS.
- Jarak sembarang vektor **x** ke DS dan searah **w** adalah $y(\mathbf{x})/||\mathbf{w}||$

Perceptron

Bentuk Umum

- Asumsikan kita memiliki data pembelajaran $\{\mathbf{x}_n, \mathbf{t}_n\}$, $n = 1$ sd N. Suatu vektor input **x** pertama-tama ditransformasi menggunakan suatu transformasi nonlinear $\phi(.)$ untuk membentuk vektor fitur $\phi(\mathbf{x})$ yang digunakan untuk membentuk model linear dengan bentuk:

$$y(\mathbf{x}) = f(\mathbf{w}^T\phi(\mathbf{x}))$$

dimana fungsi aktifasi nonlinear $f(.)$ diberikan oleh fungsi tangga dengan bentuk:

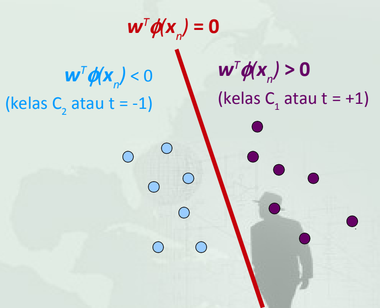
$$f(a)=+1 \text{ jika } a\geq 0$$

$$f(a)=-1 \text{ jika } a<0.$$

- Vektor $\phi(\mathbf{x})$ biasanya mengandung komponen bias $\phi_0(\mathbf{x})=1$

Perceptron

Kriteria Perceptron



- Diberikan data pembelajaran $\mathbf{x}_n$ dan t_n, dimana $t_n = +1$ untuk kelas C_1 dan $t_n = -1$ untuk kelas C_2
- Untuk setiap vektor $\mathbf{x}_n$ di kelas C_1 akan membuat $\mathbf{w}^T\phi(\mathbf{x}_n) > 0$, dan untuk setiap vektor $\mathbf{x}_n$ di kelas C_2 akan membuat $\mathbf{w}^T\phi(\mathbf{x}_n) < 0$
- Sehingga, semua vektor $\mathbf{x}_n$ akan diklasifikasi dengan benar jika $\mathbf{w}^T\phi(\mathbf{x}_n)\, t_n > 0$

Perceptron

Fungsi Error

- Untuk setiap vektor ***x*** yang tidak diklasifikasikan dengan benar, maka metode perceptron akan meminimumkan fungsi error, disebut juga kriteria perceptron (*perceptron criterion*), sbb:

$$E_p(w) = -\sum_{n \in M} w^T \phi(x_n) t_n$$

Dimana *M* adalah himpunan indeks vektor ***x*** yang tidak diklasifikasikan dengan benar

- Sehingga, bobot diperbarui secara berulang sbb:

$$w^{(\tau+1)} = w^{(\tau)} - \eta \nabla E_P(w) = w^{(\tau)} + \eta \phi(x_n) t_n$$

dimana η adalah parameter *learning rate,* dan τ ada langkah dari iterasi algoritma

Perceptron

Algoritma

1) $\{(\mathbf{x}_1, t_1), \ldots, (\mathbf{x}_N, t_N)\}$ adalah data pembelajaran
2) Inisialisasi $w^{(0)}$ dengan bilangan random
3) Untuk setiap data pembelajan (x_n, t_n), Jika data diklasifikasikan dengan benar maka tidak ada perubahan terhadap **w**
4) Untuk setiap data pembelajan (x_n, t_n), Jika data pembelajaran tidak diklasifikasikan dengan benar maka **w** diperbarui sbb:

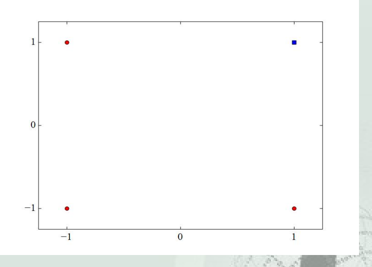
$$w^{(\tau+1)} = w^{(\tau)} - \eta \nabla E_P(w) = w^{(\tau)} + \eta \phi(x_n) t_n$$

Perceptron

Contoh

Diberikan data sebagai berikut

No	x_1	x_2	Kelas (t)
1	1	1	+1
2	1	-1	-1
3	-1	1	-1
4	-1	-1	-1



Tentukan *hyperplane* yang menjadi batas (*decision boundary*) dari ke dua kelas dari data tersebut dengan menggunakan metode perceptron!

Perceptron

Contoh

Misal $\eta = 1,\ \phi(\mathbf{x}) = \mathbf{x},\ \phi_0(\mathbf{x}) = 1,\ \mathbf{w}^{(0)} = (0, 0, 0)$

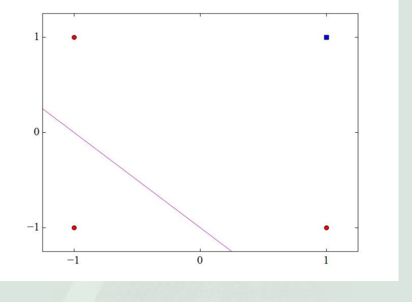
1) Ambil $\mathbf{x}_1 = (1,1)^T$ dan $t_1 = +1$ sebagai data pembelajaran:

$\mathbf{w}^{(1)} = \mathbf{w}^{(0)} + \eta\, \phi(\mathbf{x}_1)\, t_1$

$= \mathbf{w}^{(0)} + \eta\, (1, x_1, x_2)\, t_1$

$= (0, 0, 0)^T + 1 \,.\, (1, 1, 1)^T \,.\, 1$

$= (1, 1, 1)^T$



sehingga *decision boundary* adalah $1 + x_1 + x_2 = 0$

Perceptron

Contoh

2) Ambil $\mathbf{x_2} = (1,-1)^T$ dan $t_2 = -1$ sebagai data pembelajaran:

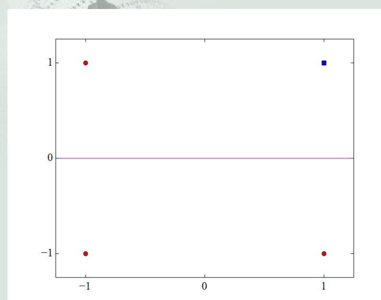
$$\mathbf{w}^{(2)} = \mathbf{w}^{(1)} + \eta\, \phi(\mathbf{x_2})\, t_2$$

$$= \mathbf{w}^{(1)} + \eta\, (1, x_1, x_2)\, t_2$$

$$= (1, 1, 1)^T + 1 \,.\, (1, 1, -1)^T \,.\, -1$$

$$= (0, 0, 2)^T$$

sehingga *decision boundary* adalah $2x_2 = 0$



Perceptron

Contoh

3) Ambil $\mathbf{x}_3 = (-1,1)^T$ dan $t_3 = -1$ sebagai data pembelajaran:

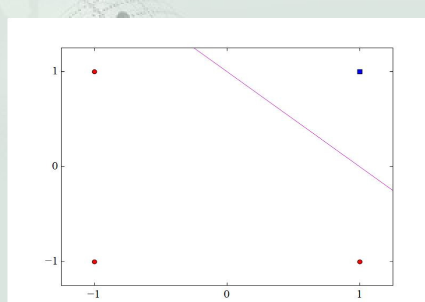
$$\mathbf{w}^{(3)} = \mathbf{w}^{(2)} + \eta\, \phi(\mathbf{x_3})\, t_3$$

$$= \mathbf{w}^{(2)} + \eta\, (1, x_1, x_2)^T\, t_3$$

$$= (0, 0, 2)^T + 1 \,.\, (1, -1, 1)^T \,.\, -1$$

$$= (-1, 1, 1)^T$$

sehingga *decision boundary* adalah $-1 + x_1 + x_2 = 0$



Perceptron

Beberapa Catatan

- Pada tahun 1962, Novikoff telah mebuktikan teorema pertama tentang perceptron. Teorema ini memulai teori tentang pembelajaran (*learning*) yang menyatakan bahwa jika
  - Norm dari vektor input $\boldsymbol{x}$ dibatasi oleh konstanta R $(|\mathbf{x}| \leq R)$
  - Data pembelajaran dapat dipisahkan dengan margin ρ:

  Maka setelah paling banyak $K \leq [R^2/\rho^2]$ koreksi, *hyperplane* yang memisahkan data pembelajaran akan dibentuk
- Perceptron telah menunjukkan kapabilitas generalisasi pada contoh kasus sederhana

Referensi

- Bishop, C. H., *Pattern Recognition and Machine Learning*, Springer, 2006 (Bab 4.1.1, Bab 4.1.7)