

bayesian :

- RADAR di setiap tempat T1, T2, T3 dan T4 dipasang 40%, 30%, 20% dan 30% dari waktu sehari
- Bila seseorang yang ngebut ke kantor berpeluang masing-masing 0.2, 0.1, 0.5 dan 0.2 melalui tiap-tiap tempat

$P(A) = P(B_1)P(A|B_1) + P(B_2)P(A|B_2) + P(B_3)P(A|B_3) + P(B_4)P(A|B_4)$

$P(A) = 0.2 \times 40\% + 0.1 \times 30\% + 0.5 \times 20\% + 0.2 \times 30\%$

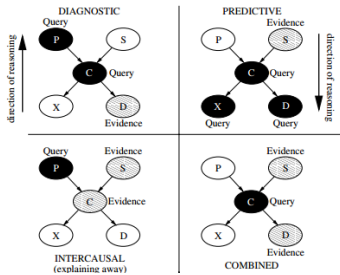
$P(A) = 0.27$

$P(B_2|A) = \frac{P(B_2)P(A|B_2)}{P(A)}$

$P(B_2|A) = \frac{0.1 \times 30\%}{0.27} = \frac{1}{9}$

- Tipe Reasoning (pertimbangan)

P(h|e) = P(e|h) P(h) / P(e) , dimana h=hipotesis, e=eviden



Day	Outlook	Temperature	Humidity	Wind	Play Tennis
Day1	Sunny	Hot	High	Weak	No
Day2	Sunny	Hot	High	Strong	No
Day3	Overcast	Hot	High	Weak	Yes
Day4	Rain	Mild	High	Weak	Yes
Day5	Rain	Cool	Normal	Weak	Yes
Day6	Rain	Cool	Normal	Strong	No
Day7	Overcast	Cool	Normal	Strong	Yes
Day8	Sunny	Mild	High	Weak	No
Day9	Sunny	Cool	Normal	Weak	Yes
Day10	Rain	Mild	High	Weak	Yes
Day11	Sunny	Mild	Normal	Strong	Yes
Day12	Overcast	Mild	High	Strong	Yes
Day13	Overcast	Hot	Normal	Weak	Yes
Day14	Rain	Mild	High	Strong	No

Misal untuk x = (Outl=Sunny, Temp=Cool, Hum=High, Wind=strong) >> Play or No??

$$h_{NB} = \arg \max_{h \in \{yes, no\}} P(h)P(x|h) = \arg \max_{h \in \{yes, no\}} P(h) \prod_i P(a_i|h)$$

$$= \arg \max_{h \in \{yes, no\}} P(h)P(Outlook = sunny|h)P(Temp = cool|h)$$

$$P(Humidity = high|h)P(Wind = strong|h)$$

P(Play = yes) = 9/14=0,64
P(Play= no) = 5/14=0,36

P(wind=strong | Play=yes) = 3/9 = 0.33
P(wind=strong | Play=no) = 3/5 = 0.6, dll

P(yes)* P(sunny|yes)* P(cool|yes)* P(high|yes)* P(strong|yes)
= 9/14 * 2/9 * 3/9 * 3/9 * 3/9
= 0.64 * 0.22 * 0.33 * 0.33 * 0.33
= **0,0053**

P(no) * P(sunny|no) * P(cool|no) * P(high|no) * P(strong|no)
= 5/14 * 3/5 * 1/5 * 4/5 * 3/5
= 0.36 * 0.6 * 0.2 * 0.8 * 0.6
= **0,0206** (lebih tinggi)
Answer: Play(x) = no

Fungsi Pembeda (Discriminant Functions)

$$p(x) = \frac{1}{2\pi^{\frac{1}{2}}|\Sigma|^{\frac{1}{2}}} \exp\left[-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)\right]$$

$$p(\omega_1) = p(\omega_2) = p(\omega_3) = \frac{1}{3}$$

$$\mu_1 = \begin{bmatrix} 0 \\ 2 \end{bmatrix}, \mu_2 = \begin{bmatrix} 4 \\ 1 \end{bmatrix}, \mu_3 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

$$\Sigma^{-1} = \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix}$$

$$w_1 = \Sigma^{-1}\mu_1 = \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix} \begin{bmatrix} 0 \\ 2 \end{bmatrix} = \begin{bmatrix} 0 \\ 4 \end{bmatrix}$$

$$w_{10} = -\frac{1}{2}\mu_1^T \Sigma^{-1}\mu_1 = -\frac{1}{2} \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix} \begin{bmatrix} 0 \\ 2 \end{bmatrix} = -\frac{1}{2} \begin{bmatrix} 0 \\ 4 \end{bmatrix} = \begin{bmatrix} 0 \\ -2 \end{bmatrix}$$

$$g_1(x) = \begin{bmatrix} 0 \\ 4 \end{bmatrix}^T x - 4 = \begin{bmatrix} 1 & 4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} - 4 = 4x_2 - 4$$

$$w_2 = \Sigma^{-1}\mu_2 = \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix} \begin{bmatrix} 4 \\ 1 \end{bmatrix} = \begin{bmatrix} 4 \\ 2 \end{bmatrix}$$

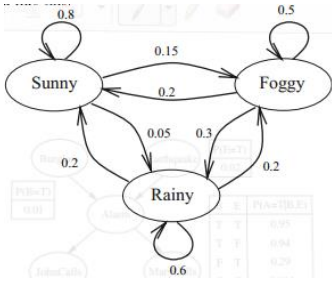
$$w_{20} = -\frac{1}{2}\mu_2^T \Sigma^{-1}\mu_2 = -\frac{1}{2} \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix} \begin{bmatrix} 4 \\ 1 \end{bmatrix} = -\frac{1}{2} \begin{bmatrix} 4 \\ 2 \end{bmatrix} = \begin{bmatrix} -2 \\ -1 \end{bmatrix}$$

$$g_2(x) = \begin{bmatrix} 4 \\ 2 \end{bmatrix}^T x - 9 = \begin{bmatrix} 4 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} - 9 = 4x_1 + 2x_2 - 9$$

$$g_3(x) = x_1 - \frac{1}{2}$$

HMM

Contohnya digunakan untuk perkiraan cuaca.



		Tomorrow's Weather		
Today's Weather	Sunny	0.8	0.05	0.15
	Rainy	0.2	0.6	0.2
	Foggy	0.2	0.3	0.5

	Probability of Umbrella
Sunny	0.1
Rainy	0.8
Foggy	0.3

jika hari ini sunny, maka probabilitas besok sunny dan lusa rainy adalah:

$$P(w_2 = \text{Sunny}, w_3 = \text{Rainy} | w_1 = \text{Sunny})$$

$$= P(w_3 = \text{Rainy} | w_2 = \text{Sunny}, w_1 = \text{Sunny}) * P(w_2 = \text{Sunny} | w_1 = \text{Sunny})$$

$$= P(w_3 = \text{Rainy} | w_2 = \text{Sunny}) * P(w_2 = \text{Sunny} | w_1 = \text{Sunny})$$

$$= (0.05)(0.8)$$

$$= 0.04$$

$$P(w_3 = \text{Rainy} | w_1 = \text{Foggy})$$

$$= P(w_2 = \text{Foggy}, w_3 = \text{Rainy} | w_1 = \text{Foggy}) + P(w_2 = \text{Rainy}, w_3 = \text{Rainy} | w_1 = \text{Foggy}) + P(w_2 = \text{Sunny}, w_3 = \text{Rainy} | w_1 = \text{Foggy})$$

$$= P(w_3 = \text{Rainy} | w_2 = \text{Foggy})P(w_2 = \text{Foggy} | w_1 = \text{Foggy}) + P(w_3 = \text{Rainy} | w_2 = \text{Rainy})P(w_2 = \text{Rainy} | w_1 = \text{Foggy}) + P(w_3 = \text{Rainy} | w_2 = \text{Sunny})P(w_2 = \text{Sunny} | w_1 = \text{Foggy})$$

$$= (0.3)(0.5) + (0.6)(0.3) + (0.05)(0.2)$$

$$= 0.34$$

- Tiga masalah pokok dari HMM :
- 1). Memberikan model λ, akan digunakan untuk mengevaluasi probabilitas dari urutan pengamatan yang diberikan, O = {O, O2... Oτ}, dinamakan, P(O|λ).
 - 2). Memberikan model λ dan urutan pengamatan O, untuk menemukan urutan keadaan Q = {q1, q2... qτ}, dimana memiliki probabilitas tertinggi yang menghasilkan O, yaitu, diinginkan untuk menemukan Q maksimum, P(Q|O,λ).
 - 3). Memberikan training set dari urutan pengamatan, X = {O^k}_k, ingin mempelajari model probabilitas maksimum dari hasil X, yaitu menemukan λ, dengan memaksimalkan P(X|λ).

JST / NN

- tujuannya untuk menemukan hubungan non-linear antara data numeric
- **Kemampuan JST →**
 - 1). Fungsi Boolean : Setiap fungsi boolean dapat diwakili oleh jaringan dengan lapisan tersembunyi tunggal tetapi mungkin memerlukan unit tersembunyi eksponensial.
 - 2). Fungsi kontinu : Setiap fungsi kontinu dibatasi dapat diperkirakan dengan kesalahan kecil sewenang-wenang, dengan jaringan dengan satu hidden layer. Setiap fungsi dapat didekati dengan akurasi sewenang-wenang oleh jaringan dengan dua lapisan tersembunyi.
- JST ditentukan oleh 3 hal, yaitu :
 - Pola hubungan antar neuron (disebut arsitektur jaringan)
 - Metode untuk menentukan bobot penghubung (disebut metode learning)
 - Fungsi aktivasi
- **Kekurangan**
 - Ketidakakuratan hasil yang diperoleh.
 - Hanya bekerja berdasarkan pola yang terbentuk pada inputnya.

X1	X2	AND	NAND	OR	NOR	XOR	XNOR
0	0	0	1	0	1	0	1
0	1	0	1	1	0	1	0
1	0	0	1	1	0	1	0
1	1	1	0	1	0	0	1

Suatu single layer network seperti gambar berikut terdiri dari 2 input $x_1=0.7$ dan $x_2=2.1$ dan memiliki bias. Bobot $w_1=0.5$ dan $w_2=-0.3$ dan bobot bias $b = 1.2$. Tentukan keluaran neuron jika fungsi aktivasinya adalah fungsi threshold bipolar.

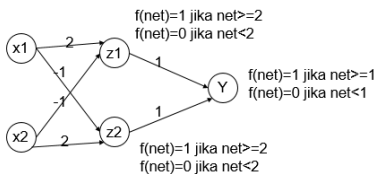
- $net = b + \sum w_i x_i = 1.2 + (0.7 \cdot 0.5) + (2.1 \cdot -0.3) = 0.92$
- Karena $net > 0$ maka keluaran jaringan $y=f(net) = 1$

Berdasarkan arsitekturnya JST dibagi menjadi :

- Single layer network
ADALINE, Hopfield, Perceptron, LVQ
- Multi layer network
MADALINE, Backpropagation, Neocognitron.
- Recurrent
BAM, Boltzman Machine, Hopfield
- Aplikasi dengan menggunakan JST :
 - Klasifikasi
ADALINE, LVQ, Backpropagation, dll
 - Pengenalan pola
ART, LVQ, Backpropagation, Neocognitron
 - Peramalan
ADALINE, MADALINE, Backpropagation
 - Optimasi
ADALINE, Hopfield, Boltzman, Backpropagation

• Buat model neuron McCulloch Pitts untuk menyatakan logika “xor”

• Penyelesaian :



• Untuk semua kemungkinan masukan x_1 dan x_2 , nilai fungsi aktivasi ditunjukkan pada tabel :

x_1	x_2	net_1	$z_1=f(net_1)$	net_2	$z_2=f(net_2)$
1	1	$1.2+1 \cdot 1=2.1$	1	$1.2+1 \cdot 1=2.1$	1
1	0	$1.2+0 \cdot 1=1.2$	0	$1.2+1 \cdot 1=2.1$	1
0	1	$1.2+1 \cdot 1=2.1$	1	$1.2+0 \cdot 1=1.2$	0
0	0	$1.2+0 \cdot 1=1.2$	0	$1.2+0 \cdot 1=1.2$	0

• Hasil akhir :

z_1	z_2	net	$Y=f(net)$
1	1	$1+1+1=3$	1
1	0	$1+0+1=2$	1
0	1	$0+1+1=2$	1
0	0	$0+0+1=1$	0

• Selanjutnya, nilai z_1 dan z_2 ini diteruskan ke fungsi aktivasi Y dengan bobot dari $z_1=1$ dan $z_2=1$ dan threshold = 1. Hasilnya tampak pada tabel berikut :

• Tampak bahwa keluaran jaringan tepat sama dengan tabel logika “xor” berarti jaringan dapat dengan tepat merepresentasikan fungsi “xor”

clustering

Clustering melakukan penge-lompokan data tanpa berdasarkan kelas data tertentu. Bahkan clustering dapat dipakai untuk memberikan label pada kelas data yang belum diketahui itu. Karena itu clustering sering digolongkan sebagai metode unsupervised learning. Prinsip dari clustering adalah memaksimalkan kesamaan antar anggota satu kelas dan meminimumkan kesamaan antar kelas/cluster. Clustering dapat dilakukan pada data yang memiliki beberapa atribut yang dipetakan sebagai ruang multidimensi. Banyak algoritma clustering memerlukan fungsi jarak untuk mengukur kemiripan antar data, diperlukan juga metode untuk normalisasi bermacam atribut yang dimiliki data. DBSCAN merupakan clustering berdasarkan density, yaitu kerapatan, jumlah titik yang diperoleh dalam radius tertentu.

decision tree

$$InformationGain(a_i, S) = Entropy(y, S) - \sum_{v_{i,j} \in dom(a_i)} \frac{|S_{a_i=v_{i,j}}|}{|S|} \cdot Entropy(y, S_{a_i=v_{i,j}})$$

where:

$$Entropy(y, S) = \sum_{c_j \in dom(y)} - \frac{|S_{y=c_j}|}{|S|} \cdot \log_2 \frac{|S_{y=c_j}|}{|S|}$$

$$GainRatio(a_i, S) = \frac{InformationGain(a_i, S)}{Entropy(a_i, S)}$$

$$Entropy(t) = -\sum p(j|i) \log_2 p(j|i)$$

$$Error(t) = 1 - \max P(i|i)$$

C1	0	P(C1) = 0/6 = 0	P(C2) = 6/6 = 1
C2	6	Entropy = $-0 \log_2 0 - 1 \log_2 1 = -0 - 0 = 0$	

C1	0	P(C1) = 0/6 = 0	P(C2) = 6/6 = 1
C2	6	Error = $1 - \max(0, 1) = 1 - 1 = 0$	

C1	1	P(C1) = 1/6	P(C2) = 5/6
C2	5	Entropy = $-(1/6) \log_2 (1/6) - (5/6) \log_2 (5/6) = 0.65$	

C1	1	P(C1) = 1/6	P(C2) = 5/6
C2	5	Error = $1 - \max(1/6, 5/6) = 1 - 5/6 = 1/6$	

C1	2	P(C1) = 2/6	P(C2) = 4/6
C2	4	Entropy = $-(2/6) \log_2 (2/6) - (4/6) \log_2 (4/6) = 0.92$	

C1	2	P(C1) = 2/6	P(C2) = 4/6
C2	4	Error = $1 - \max(2/6, 4/6) = 1 - 4/6 = 1/3$	

Algoritma Hierarchical clustering

Hierarchical Clustering & Dataset

- Single Linkage**
Metode ini sangat cocok untuk dipakai pada kasus shape independent clustering, karena kemampuannya untuk membentuk pattern tertentu dari cluster. Untuk kasus condensed clustering, metode ini tidak bagus.
 - Centroid Linkage**
Metode ini baik untuk kasus clustering dengan normal data set distribution. Akan tetapi, metode ini tidak cocok untuk data yang mengandung outlier.
 - Complete Linkage**
Metode ini sangat ampuh untuk memperkecil variance within cluster karena meletakkan centroid pada saat penggabungan antar cluster. Metode ini juga baik untuk data yang mengandung outlier.
 - Average Linkage**
Metode ini relatif yang terbaik dari metode-metode hierarchial. Namun, ini harus dibayar dengan waktu komputasi yang paling tinggi dibandingkan dengan metode-metode hierarchial yang lain.
1. Tentukan k sebagai jumlah cluster yang ingin dibentuk
 2. Setiap data dianggap sebagai cluster. Kalau N =jumlah data dan c =jumlah cluster, berarti ada $c=N$.
 3. Hitung jarak antar cluster
 4. Cari 2 cluster yang mempunyai jarak antar cluster yang paling minimal dan gabungkan (berarti $c=c-1$)
 5. Jika $c > k$, kembali ke langkah 3

Algoritma K-means

Karakteristik K-means

1. Tentukan k sebagai jumlah cluster yang ingin dibentuk
 2. Bangkitkan k centroids (titik pusat cluster) awal secara random
 3. Hitung jarak setiap data ke masing-masing centroids
 4. Setiap data memilih centroids yang terdekat
 5. Tentukan posisi centroids baru dengan cara menghitung nilai rata-rata dari data-data yang memilih pada centroid yang sama
 6. Kembali ke langkah 3 jika posisi centroids baru dengan centroids lama tidak sama.
- K-means sangat cepat dalam proses clustering
 - K-means sangat sensitif pada pembangkitan centroids awal secara random
 - Memungkinkan suatu cluster tidak mempunyai anggota
 - Hasil clustering dengan K-means bersifat tidak unik (selalu berubah-ubah) - terkadang baik, terkadang jelek.
 - K-means sangat sulit untuk mencapai global optimum

6. GROUND TRUTH

	+	-
+	TP	FP
-	FN	TN

Akurasi / efficiency = $(TP+TN) / \text{Total}$
 Presisi / predictive = $TP / (TP+FP)$ (+) | $TN / (FN+TN)$ (-)
 Recall = $TP / (TP+FN)$
 Specificity = $TN / (FP+TN)$

Contoh pencarian kata pada google.

Misal ditemukan 5 yang relevan & 1000 page yang muncul. maka : presisi = $5/1000$

Kemudian terdapat 100 page yang tidak relevan. maka recall = $5/100$

A merupakan total page yang dimiliki google. misal 1.000.000.000, maka $A = 1.000.000.000 - 995-95$.

#Tambahan NN

