PERTEMUAN 10

Naive Bayes dan K-Nearest-Neighbor

Copyright © September 2022

Algoritma pembelajaran mesin dapat dibagi menjadi beberapa kategori. Dari sudut pandang apakah algoritma memiliki parameter yang harus dioptimasi, dapat dibagi menjadi:

1. **Parametrik**. Pada kelompok ini, kita mereduksi permasalahan sebagai optimisasi parameter. Kita mengasumsikan permasalahan dapat dilambangkan oleh fungsi dengan bentuk tertentu (e.g., linear, polinomial, dsb). Contoh kelompok ini adalah model linear.

2. **Non parametrik**. Pada kelompok ini, kita tidak mengasumsikan permasalahan dapat dilambangkan oleh fungsi dengan bentuk tertentu. Contoh kelompok ini adalah Naive Bayes, decision tree (ID3) dan K-Nearest Neighbors.

Copyright © September 2022

Dari sudut pandang lainnya, jenis algoritma dapat dibagi menjadi:

1. **Model linear**, contoh regresi linear, regresi logistik, support vector machine.
2. **Model probabilistik**, contoh Naive Bayes, hidden markov model.
3. **Model non-linear**, yaitu (typically) articial neural network.

Copyright © September 2022

Naive Bayes

Naive Bayes adalah algoritma supervised learning yang sangat sederhana. Secara formal, persamaan Naive Bayes untuk klasikasi diberikan pada persamaan:

$$\text{likelihood}(c_i) = P(c_i) \prod_{f=1}^{F} P(t_f|c_i)$$

Dimana: c_i adalah suatu nilai kelas, C adalah kelas (himpunan), t adalah fitur (satu fitur, bukan feature vector) dan F adalah banyaknya fitur.

Copyright © September 2022

Kita memprediksi kelas berdasarkan probabilitas kemunculan nilai fitur pada kelas tersebut. Pertama, kita hitung likelihood suatu feature vector diklasikasikan ke kelas tertentu berdasarkan bagaimana probabilitas korespondensi fitur-fiturnya terhadap kelas tersebut.

$$\text{likelihood}(c_i) = P(c_i) \prod_{f=1}^{F} P(t_f|c_i)$$

Kemudian, kita normalisasi likelihood semua kelas untuk mendapatkan probabilitas class-assignment (softmax).

$$P_{assignment}(c_i) = \frac{\text{likelihood}(c_i)}{\sum_{c_j \epsilon C} \text{likelihood}(c_j)}$$

Akhirnya, kita pilih kelas dengan probabilitas tertinggi

$$\hat{c_i} = \arg\max_{c_i \epsilon C} P_{assignment}(c_i)$$

Copyright © September 2022

Agar mendapatkan gambaran praktis, mari bangun model Naive Bayes untuk Tabel 4.1. Tabel ini disebut sebagai dataset, yaitu memuat entry data (tiap baris disebut sebagai instans/instance). Kita anggap Tabel 4.1 sebagai training data. Untuk menghitung probabilitas, pertama-tama hitung terlebih dahulu frekuensi nilai atribut seperti pada Tabel 4.2, setelah itu bangun model probabilitasnya seperti pada Tabel 4.3.

id	outlook	temperature	humidity	windy	play (class)
1	sunny	hot	high	false	no
2	sunny	hot	high	true	no
3	overcast	hot	high	false	yes
4	rainy	mild	high	false	yes
5	rainy	cool	normal	false	yes
6	rainy	cool	normal	true	no
7	overcast	cool	normal	true	yes
8	sunny	mild	high	false	no
9	sunny	cool	normal	false	yes
10	rainy	mild	normal	false	yes
11	sunny	mild	normal	true	yes
12	overcast	mild	high	true	yes
13	overcast	hot	normal	false	yes
14	rainy	mild	high	true	no

Tabel 4.1. Contoh dataset *play tennis* (UCI machine learning repository)

Copyright © September 2022

	outlook			temperature			humidity			windy		play (class)	
	yes	no		yes	no		yes	no		yes	no	yes	no
sunny	2	3	hot	2	3	high	3	4	false	6	2	9	5
overcast	4	0	mild	4	2	normal	6	1	true	3	3		
rainy	3	2	cool	3	1								

Tabel 4.2. Frekuensi setiap nilai atribut

	outlook			temperature			humidity			windy		play (class)	
	yes	no		yes	no		yes	no		yes	no	yes	no
sunny	2/9	3/5	hot	2/9	3/5	high	3/9	4/5	false	6/9	2/5	9/14	5/14
overcast	4/9	0/5	mild	4/9	2/5	normal	6/9	1/5	true	3/9	3/5		
rainy	3/9	2/5	cool	3/9	1/5								

Tabel 4.3. Probabilitas setiap nilai atribut

Copyright © September 2022

Untuk menguji kebenaran model yang telah dibangun, kita menggunakan testing data, diberikan pada Tabel 4.4. testing data berisi unseen example yaitu contoh yang tidak ada pada training data.

id	outlook	temperature	humidity	windy	play (class)
1	sunny	cool	high	true	no

Tabel 4.4. Contoh testing data *play tennis* [7]

$$\begin{aligned}
\text{likelihood}(play = yes) &= P(yes)P(sunny|yes)P(cool|yes)P(high|yes)P(true|yes) \\
&= \frac{9}{14} * \frac{2}{9} * \frac{3}{9} * \frac{3}{9} * \frac{3}{9} \\
&= 0.0053 \\
\text{likelihood}(play = no) &= P(no)P(sunny|no)P(cool|no)P(high|no)P(true|no) \\
&= \frac{5}{14} * \frac{3}{5} * \frac{1}{5} * \frac{4}{5} * \frac{3}{5} \\
&= 0.0206 \\
P_{assignment}(play = yes) &= \frac{\text{likelihood}(play = yes)}{\text{likelihood}(play = yes) + \text{likelihood}(play = no)} \\
&= \frac{0.0053}{0.0053 + 0.0206} \\
&= 0.205 \\
P_{assignment}(play = no) &= \frac{\text{likelihood}(play = no)}{\text{likelihood}(play = yes) + \text{likelihood}(play = no)} \\
&= \frac{0.0206}{0.0053 + 0.0206} \\
&= 0.795
\end{aligned}$$

Copyright © September 2022

Karena $P_{assignment}$ (play = no) > $P_{assignment}$ (play = yes) maka diputuskan bahwa kelas untuk unseen example adalah play = no.

Proses klasikasi untuk data baru sama seperti proses klasikasi untuk testing data, yaitu kita ingin menebak kelas data. Karena model berhasil menebak kelas pada training data dengan tepat, akurasi model adalah 100% (kebetulan contohnya hanya ada satu).

Copyright © September 2022

K-means

Pada *supervised learning* diketahui kelas data untuk setiap *feature vector*, sedangkan untuk *unsupervised learning* tidak diketahui.

Tujuan *unsupervised learning* salah satunya adalah melakukan ***clustering***. Yaitu mengelompokkan data-data dengan karakter mirip.

Copyright © September 2022

Untuk melakukan pembelajaran menggunakan **K-means** harus mengikuti langkah-langkah sebagai berikut:

1. Tentukan jumlah kelompok yang diinginkan.
2. Inisiasi **centroid** untuk setiap kelompok (secara acak). **Centroid** adalah data yang merepresentasikan suatu kelompok (ibaratnya ketua kelompok)
3. Hitung kedekatan suatu data terhadap *centroid*, kemudian masukkan data tersebut ke kelompok yang **centroid**-nya memiliki sifat terdekat dengan dirinya.
4. Pilih kembali **centroid** untuk masing-masing kelompok, yaitu dari anggota kelompok tersebut (semacam memilih ketua yang baru).
5. Ulangi langkah-langkah sebelumnya sampai tidak ada perubahan anggota untuk semua kelompok.

Copyright © September 2022

id	rich	intelligent	good looking
1	yes	yes	yes
2	yes	no	no
3	yes	yes	no
4	no	no	no
5	no	yes	no
6	no	no	yes

Tabel 4.5. Contoh dataset orang kaya

Perhatikan Tabel 4.5, Kelompokkan data pada tabel tersebut menjadi dua clusters (dua kelompok) yaitu k_1, k_2 menggunakan algoritma K-means. Pertama-tama inisiasi centroid secara acak, id_1 untuk k_1 dan id_6 untuk k_2.

Hitung kedekatan data lainnya terhadap centroid. Untuk mempermudah contoh, hitung perbedaan data satu dan lainnya dengan menghitung perbedaan nilai atribut (nilai atributnya sama atau tidak).
Apabila perbedaan suatu data terhadap kedua centroid bernilai sama, masukkan ke kelas dengan nomor urut lebih kecil.

Copyright © September 2022

Setelah langkah ini, kelompok satu beranggotakan {id_1, id_2, id_3, id_5} sementara kelompok dua beranggotakan {id_4, id_6}
Pilih kembali centroid untuk masing-masing kelompok yang mana berasal dari anggota kelompok itu sendiri. Misal pilih secara acak, centroid untuk kelompok pertama adalah id_2 sementara untuk kelompok kedua adalah id4. Hitung kembali assignment anggota kelompok yang ilustrasinya dapat dilihat pada Tabel 4.7.

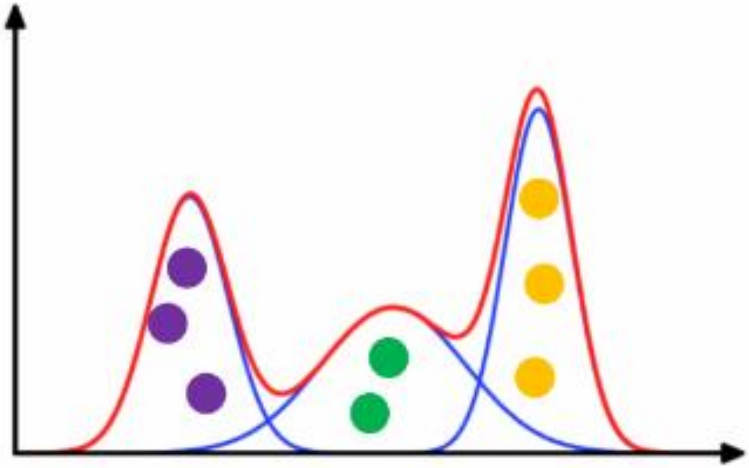
id	perbedaan dengan *centroid*$_1$	perbedaan dengan *centroid*$_2$	assignment
1	2	3	k_1
3	1	3	k_1
5	3	1	k_2
6	2	2	k_1

Tabel 4.7. *Assignment* K-Means langkah 2

Hasil langkah ke-2 adalah perubahan anggota kelompok, k_1 = {id_1, id_2, id_3, id_5} dan k_2 = {id_4, id_6} Anggap pada langkah ke-3 memilih kembali id_2 dan id_4 sebagai centroid masing-masing kelompok sehingga tidak ada perubahan keanggotaan.

Copyright © September 2022

Clustering itu memiliki hubungan erat dengan Gaussian Mixture Model. Secara sederhana, satu cluster (atau satu kelas) sebenarnya seolah-olah dapat dipisahkan dengan kelas lainnya oleh distribusi gaussian.
Perhatikan Gambar 4.1! Suatu cluster atau kelas, seolah-olah “dibungkus“ oleh suatu distribusi gaussian. Distribusi seluruh dataset dapat diaproksimasi dengan Gaussian Mixture Model (GMM).



Gambar 4.1. Ilustrasi hubungan Clustering, kelas, dan Gaussian

Copyright © September 2022

Data memiliki suatu pola (dalam statistik disebut distribusi), GMM dipercaya dapat mengaproksimasi fungsi apapun.

Dengan demikian, machine learning yang mempunyai salah satu tujuan untuk menemukan pola dataset, memiliki hubungan yang sangat erat dengan distribusi gaussian karena pola tersebut dapat diaproksimasi dengan distribusi gaussian.

Copyright © September 2022

K-Nearest-Neighbor

- Ide **K-Nearest-Neighbor** (KNN) adalah mengelompokkan data ke kelompok yang memiliki sifat termirip dengannya.

- Hal ini sangat mirip dengan **K-means**. Bila K-means digunakan untuk *clustering*, KNN digunakan untuk klasifikasi.

Copyright © September 2022

Algoritma klasifikasi ini disebut juga algoritma malas karena tidak mempelajari cara mengkategorikan data, melainkan hanya mengingat data yang sudah ada yang kita telah mengelompokkan data orang kaya menjadi dua kelompok.

id	rich	intelligent	good looking
1	yes	yes	yes
2	yes	no	no
3	yes	yes	no
4	no	no	no
5	no	yes	no
6	no	no	yes

Tabel 4.5. Contoh dataset orang kaya

Copyright © September 2022

KNN mencari K *feature vector* dengan sifat termirip, kemudian mengelompokkan data baru ke kelompok *feature vector* tersebut. Sebagai ilustrasi mudah, lakukan klasikasi algoritma KNN dengan K = 3 untuk data baru {*rich = no; intelligent = yes; good looking = yes*}.

Pada Tabel 4.8. feature vector termirip dimiliki oleh data dengan id_1, id_5, id_6.

id	perbedaan
1	1
2	3
3	3
4	2
5	1
6	1

Tabel 4.8. Perbedaan data baru vs data orang kaya

Copyright © September 2022