SUPERVISED LEARNING

~Naïve Bayes~

Metode Klasifikasi Bayes

- Pengklasifikasi Bayesian adalah pengklasifikasi statistik untuk memprediksi probabilitas keanggotaan kelas seperti probabilitas bahwa tupel tertentu milik kelas tertentu.
- Klasifikasi Bayesian didasarkan pada teorema Bayes.
- Pengklasifikasi Bayesian sederhana atau dikenal sebagai pengklasifikasi Bayesian naif (*naïve Bayesian classifier*) juga menunjukkan akurasi dan kecepatan tinggi ketika diterapkan ke database besar.
- Pengklasifikasi Naïve Bayes mengasumsikan bahwa efek dari nilai atribut pada kelas tertentu tidak tergantung pada nilai atribut lainnya, yaitu disebut *independensi bersyarat* kelas. Hal ini dibuat untuk menyederhanakan perhitungan yang terlibat sehingga dianggap "naif."

Teorema Bayes

- Teorema Bayes dinamai Thomas Bayes, seorang pendeta Inggris nonkonformis yang melakukan pekerjaan awal dalam teori probabilitas dan keputusan selama abad ke-18.
- Misalkan $\boldsymbol{X}$ menjadi tupel data dengan pengukuran yang dilakukan pada satu set *n* atribut. Dalam istilah Bayesian, $\boldsymbol{X}$ dianggap sebagai "bukti". Biarkan $\boldsymbol{H}$ menjadi beberapa hipotesis yang mana data tupel $\boldsymbol{X}$ milik kelas tertentu C. Untuk masalah klasifikasi, kami ingin menentukan $P(H|\boldsymbol{X})$, probabilitas bahwa hipotesis H dengan diberikan "bukti" atau data yang diamati tupel $\boldsymbol{X}$. Dengan kata lain, kita mencari probabilitas bahwa tupel $\boldsymbol{X}$ termasuk ke dalam kelas C, mengingat kita mengetahui deskripsi atribut dari $\boldsymbol{X}$.
- **Teorema Bayes** adalah

$$P(H|\boldsymbol{X}) = \frac{P(\boldsymbol{X}|H)P(H)}{P(\boldsymbol{X})} \quad (1)$$

Teorema Bayes: contoh

Misalkan dunia tupel data kita terbatas pada pelanggan yang masing-masing dijelaskan oleh atribut umur dan pendapatan, dan bahwa $\boldsymbol{X}$ adalah pelanggan berusia 35 tahun dengan pendapatan \$40.000. Misalkan H adalah hipotesis bahwa pelanggan kita akan membeli komputer.

$P(H|\boldsymbol{X})$ mencerminkan probabilitas bahwa pelanggan $\boldsymbol{X}$ akan membeli komputer jika kita mengetahui usia dan pendapatan pelanggan.

$P(H)$ adalah probabilitas bahwa setiap pelanggan tertentu akan membeli komputer, tanpa memandang usia, pendapatan, atau informasi lainnya.

- $P(H|\boldsymbol{X})$ adalah *posterior probability* dari H yang dikondisikan pada $\boldsymbol{X}$. Sebaliknya, $P(H)$ adalah *prior probability* dari H.
- *Posterior probability*, $P(H|\boldsymbol{X})$, didasarkan pada lebih banyak informasi daripada *prior probability*, $P(H)$, yang tidak bergantung pada X.
- Demikian pula, $P(\boldsymbol{X}|H)$ adalah *posterior probability* $\boldsymbol{X}$ yang dikondisikan pada H. Yaitu, probabilitas bahwa seorang pelanggan, $\boldsymbol{X}$, berusia 35 tahun dan menghasilkan \$40.000, dengan diketahui bahwa pelanggan tersebut akan membeli sebuah komputer.
- $P(\boldsymbol{X})$ adalah *prior probability* dari $\boldsymbol{X}$. Yaitu, probabilitas bahwa seseorang dari kumpulan pelanggan kita berusia 35 tahun dan menghasilkan \$40.000.

Klasifikasi Naïve Bayes (1)

1. Misalkan D adalah training set dari tuple dan label kelas terkaitnya. Setiap tuple diwakili oleh vektor atribut n-dimensi,$\boldsymbol{X} = (x_1, x_2, \dots, x_n)$, menggambarkan n pengukuran yang dilakukan pada tuple dari n atribut, masing-masing, $A_1, A_2, \dots, A_n$.
2. Misalkan ada m kelas, $C_1, C_2, \dots, C_m$. Diberikan sebuah tuple, $\boldsymbol{X}$, pengklasifikasi akan memprediksi bahwa $\boldsymbol{X}$ milik kelas yang memiliki *posterior probability* tertinggi, dikondisikan pada $\boldsymbol{X}$. Artinya, pengklasifikasi naïve Bayesian memprediksi bahwa tupel $\boldsymbol{X}$ milik kelas C_i **jika dan hanya jika**

$$P(C_i|\boldsymbol{X}) > P(C_j|\boldsymbol{X}) \text{ untuk } i \leq j \leq m, j \neq i$$

Jadi, kita maksimalkan $P(C_i|\boldsymbol{X})$. Kelas C_i di mana $P(C_j|\boldsymbol{X})$ dimaksimalkan disebut *maximum posteriori hypothesis*. Dengan teorema Bayes (persamaan 1),

$$P(C_i|\boldsymbol{X}) = \frac{P(\boldsymbol{X}|C_i)P(C_i)}{P(\boldsymbol{X})} \qquad (2)$$

Klasifikasi Naïve Bayes (2)

3. Karena $P(\boldsymbol{X})$ konstan untuk semua kelas, hanya $P(\boldsymbol{X}|C_i)P(C_i)$ yang perlu dimaksimalkan. Jika *prior probabilities* kelas tidak diketahui, maka biasanya diasumsikan bahwa kelas-kelas tersebut memiliki peluang yang sama, yaitu $P(C_1) = P(C_2) = \cdots = P(C_m)$, dan karena itu kita akan memaksimalkan $P(\boldsymbol{X}|C_i)$. Jika tidak, kita maksimalkan $P(\boldsymbol{X}|C_i)P(C_i)$.

4. Mengingat kumpulan data dengan banyak atribut, akan sangat mahal secara komputasi untuk menghitung $P(\boldsymbol{X}|C_i)$. Untuk mengurangi komputasi dalam mengevaluasi $P(\boldsymbol{X}|C_i)$, dibuat asumsi naif tentang independensi bersyarat kelas. Ini mengasumsikan bahwa nilai atribut secara bersyarat independen satu sama lain, mengingat label kelas dari tupel (yaitu, bahwa tidak ada hubungan ketergantungan di antara atribut). Jadi,

$$P(\boldsymbol{X}|C_i) = \prod_{k=1}^{n} P(x_k|C_i) = P(x_1|C_i) \times P(x_2|C_i) \times \cdots \times P(x_n|C_i) \quad (3)$$

Kita dapat dengan mudah memperkirakan probabilitas $P(x_1|C_i) \times P(x_2|C_i) \times \cdots \times P(x_n|C_i)$ dari *training tuples*. Nilai x_k mengacu pada nilai atribut A_k untuk tuple $\boldsymbol{X}$. Untuk setiap atribut, kita melihat apakah atribut tersebut bernilai kategorikal atau kontinu.

Klasifikasi Naïve Bayes (3)

Misalnya, untuk menghitung $P(\boldsymbol{X}|C_i)$, kami mempertimbangkan hal berikut:

(a) Jika A_k adalah kategorikal, maka $P(x_k|C_i)$ adalah banyaknya tupel dari kelas C_i pada D yang bernilai x_k untuk A_k, dibagi $|C_{i,D}|$, banyaknya tupel dari kelas C_i pada D.

(b) Jika A_k bernilai kontinu, maka kita perlu melakukan pekerjaan lainnya. Sebuah atribut bernilai kontinu biasanya diasumsikan memiliki distribusi Gaussian dengan mean μ dan standar deviasi σ, yang didefinisikan oleh

$$g(x, \mu, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \quad (4)$$

sehingga

$$P(x_k|C_i) = g(x_k, \mu_{C_i}, \sigma_{C_i}) \quad (5)$$

Kita perlu menghitung μ_{C_i} dan σ_{C_i} dari nilai atribut A_k untuk *training tuples* kelas C_i. Kemudian memasukkan dua besaran ini ke Persamaan (4), bersama dengan x_k, untuk memperkirakan $P(x_k|C_i)$.

Klasifikasi Naïve Bayes (4)

Sebagai contoh, misalkan $\boldsymbol{X} = (35, \$40.000)$, di mana A_1 dan A_2 masing-masing adalah atribut *usia* dan *pendapatan*. Atribut label kelas menjadi *beli_komputer*. Label kelas terkait untuk $\boldsymbol{X}$ adalah *ya* (yaitu, *beli_komputer=ya*). Misalkan *usia* belum didiskritisasi sehingga atribut bernilai kontinu.

Misalkan dari *training set*, kita menemukan bahwa pelanggan di D yang membeli komputer adalah berusia 38 ± 12 tahun. Dengan kata lain, untuk atribut *usia* dan kelas ini, kita memiliki $\mu = 38$ tahun dan $\sigma = 12$ tahun. Kita dapat memasukkan besaran-besaran ini, bersama dengan $x_1 = 35$ untuk tuple X kita, ke dalam Persamaan (4) untuk memperkirakan $P(usia = 35|beli\_komputer = ya)$.

5. Untuk memprediksi label kelas $\boldsymbol{X}$, $P(\boldsymbol{X}|C_i)P(C_i)$ dievaluasi untuk setiap kelas C_i. Pengklasifikasi memprediksi bahwa label kelas dari tuple $\boldsymbol{X}$ adalah kelas C_i **jika dan hanya jika**

$$P(\boldsymbol{X}|C_i)P(C_i) > P\left(\boldsymbol{X}\middle|C_j\right)P(C_j) \text{ untuk } i \leq j \leq m, j \neq i \qquad (6)$$

Dengan kata lain, label kelas yang diprediksi adalah kelas C_i dengan nilai maksimum $P(\boldsymbol{X}|C_i)P(C_i)$.

“Seberapa efektif pengklasifikasi Bayesian?”

Berbagai studi empiris pengklasifikasi ini dibandingkan dengan pohon keputusan dan pengklasifikasi jaringan saraf telah menemukan hal itu sebanding di beberapa domain. **Secara teori**, pengklasifikasi Bayesian memiliki tingkat kesalahan minimum dibandingkan dengan semua pengklasifikasi lainnya. Namun, dalam praktiknya hal ini **tidak selalu terjadi, karena ketidakakuratan dalam asumsi** yang dibuat untuk penggunaannya, seperti independensi bersyarat kelas, dan **kurangnya data probabilitas** yang tersedia.

Pengklasifikasi Bayesian juga berguna karena memberikan pembenaran teoretis untuk pengklasifikasi lain yang tidak secara eksplisit menggunakan teorema Bayes. Misalnya, di bawah asumsi tertentu, dapat ditunjukkan bahwa banyak jaringan saraf dan algoritma pencocokan kurva menghasilkan hipotesis posteriori maksimum, seperti halnya pengklasifikasi naïve Bayes.

Contoh 1: Memprediksi label kelas menggunakan klasifikasi naïve Bayes

Kita ingin memprediksi label kelas dari sebuah tuple menggunakan klasifikasi naïve Bayes dengan *training data* dalam Tabel 1.

Tuple data tersebut dideskripsikan dengan atribut *usia*, *pendapatan*, *mahasiswa*, dan *peringkat_kredit*. Atribut label kelas, *beli_komputer*, memiliki dua nilai yang berbeda (yaitu, {ya, tidak}). Misalkan C_1 untuk kelas *beli_komputer=ya* dan C_2 untuk *beli_komputer=tidak*. Tuple yang ingin kita klasifikasikan adalah
$\boldsymbol{X}$ = (usia=remaja, pendapatan=sedang, mahasiswa=ya, peringkat_kredit=cukup)

Kita perlu memaksimalkan $P(\boldsymbol{X}|C_i)P(C_i)$, untuk $i = 1, 2$.

$P(C_i)$ adalah *prior probability* dari setiap kelas

$$P(beli\_komputer = ya) = \frac{9}{14} = 0{,}643$$

$$P(beli\_komputer = tidak) = \frac{5}{14} = 0{,}357$$

Probabilitas bersyarat

$$P(usia = remaja \mid beli\_komputer = ya) = \frac{2}{9} = 0{,}222$$

$$P(usia = remaja \mid beli\_komputer = tidak) = \frac{3}{5} = 0{,}600$$

$$P(pendapatan = sedang \mid beli\_komputer = ya) = \frac{4}{9} = 0{,}444$$

$$P(pendapatan = sedang \mid beli\_komputer = tidak) = \frac{2}{5} = 0{,}400$$

$$P(mahasiswa = ya \mid beli\_komputer = ya) = \frac{6}{9} = 0{,}667$$

$$P(mahasiswa = ya \mid beli\_komputer = tidak) = \frac{1}{5} = 0{,}200$$

$$P(peringkat\_kredit = cukup \mid beli\_komputer = ya) = \frac{6}{9} = 0{,}667$$

$$P(peringkat\_kredit = cukup \mid beli\_komputer = tidak) = \frac{2}{5} = 0{,}400$$

Tabel 1

usia	pendapatan	mahasiswa	peringkat_kredit	kelas: beli_komputer
remaja	tinggi	tidak	cukup	tidak
remaja	tinggi	tidak	baik	tidak
lansia	tinggi	tidak	cukup	ya
dewasa	tidak ada	tidak	cukup	ya
dewasa	rendah	ya	cukup	ya
dewasa	rendah	ya	baik	tidak
lansia	rendah	ya	baik	ya
remaja	sedang	tidak	cukup	tidak
remaja	rendah	ya	cukup	ya
dewasa	sedang	ya	cukup	ya
remaja	sedang	ya	baik	ya
lansia	sedang	tidak	baik	ya
lansia	tinggi	ya	cukup	ya
dewasa	tidak ada	tidak	baik	tidak

Dengan nilai probabilitas-probabilitas yang sudah dihitung sebelumnya, kita dapat menghitung $P(\boldsymbol{X}|C_i), i = 1, 2$.

$$P(\boldsymbol{X} \mid beli\_komputer = ya) = P(usia = remaja \mid beli\_komputer = ya) \times P(pendapatan = sedang \mid beli\_komputer = ya) \times P(mahasiswa = ya \mid beli\_komputer = ya) \times P(peringkat\_kredit = cukup \mid beli\_komputer = ya)$$

$$= 0{,}222 \times 0{,}444 \times 0{,}667 \times 0{,}667 = 0{,}044$$

$$P(\boldsymbol{X} \mid beli\_komputer = tidak) = 0{,}600 \times 0{,}400 \times 0{,}200 \times 0{,}400 = 0{,}019$$

Untuk mendapatkan kelas, C_i, yang memaksimumkan $P(\boldsymbol{X}|C_i)\, P(C_i)$, kita menghitung

$$P(\boldsymbol{X} \mid beli\_komputer = ya)P(beli\_komputer = ya) = 0{,}044 \times 0{,}643 = 0{,}028$$
$$P(\boldsymbol{X} \mid beli\_komputer = tidak)P(beli\_komputer = tidak) = 0{,}019 \times 0{,}357 = 0{,}007$$

Jadi, 0,028 > 0,007 sehingga pengklasifikasi naïve Bayes memprediksi *beli_komputer=ya* untuk tuple $\boldsymbol{X}$

“Bagaimana jika menemukan nilai probabilitas nol?”

Ada trik sederhana untuk menghindari masalah ini. Kita dapat **berasumsi** bahwa **database pelatihan kita, *D*, sangat besar** sehingga **menambahkan satu ke setiap hitungan yang kita butuhkan hanya akan membuat perbedaan** yang dapat diabaikan dalam nilai probabilitas yang diperkirakan, tetapi akan dengan mudah menghindari kasus nilai probabilitas nol.

Teknik untuk estimasi probabilitas ini dikenal sebagai **Laplacian correction atau Laplace estimator**, dinamai Pierre Laplace, seorang matematikawan Perancis. Jika kita memiliki, katakanlah, *q* jumlah yang masing-masing kita tambahkan satu, maka kita harus ingat untuk tambahkan *q* ke penyebut yang sesuai yang digunakan dalam perhitungan probabilitas. Perhatikan contoh 2.

Contoh 2: Menggunakan koreksi Laplacian

Misalkan untuk kelas *beli_komputer=ya* di beberapa database pelatihan, *D*, berisi 1000 tupel, kita memiliki

0 tupel dengan *pendapatan=rendah*,

990 tupel dengan pendapatan=sedang, dan

10 tupel dengan pendapatan=tinggi.

Probabilitas kejadian ini, tanpa koreksi Laplacian, masing-masing adalah

$0, \frac{990}{1000} = 0{,}990$, dan $\frac{10}{1000} = 0{,}010$.

Menggunakan koreksi Laplacian untuk tiga kuantitas, kita berpura-pura bahwa kita **memiliki 1 tupel lagi** untuk setiap pasangan pendapatan-nilai. Dengan cara ini, kita memperoleh probabilitas berikut (dibulatkan), masing-masing:

$$\frac{1}{1003} = 0{,}001\,; \frac{991}{1003} = 0{,}988\,; dan\, \frac{11}{1003} = 0{,}011.$$

Perkiraan probabilitas "dikoreksi" mendekati yang "tidak dikoreksi", tetapi nilai probabilitas nol dihindari.

Terima Kasih