Naive bayes:

	X_1	X_2	X_3	C
Slučaj 1	Da	A	[0-10]	DA
Slučaj 2	Ne	C	[10-20]	NE
Slučaj 3	Da	A	[10-20]	NE
...	...	...	...	...
Slučaj N	Ne	B	[20-30]	DA

Model (C - klasa, X - atribut):

$$P(C_k|\boldsymbol{X}) = \frac{P(\boldsymbol{X}|C_k) \cdot P(C_k)}{P(\boldsymbol{X})} = \frac{\prod_i P(X_i|C_k) \cdot P(C_k)}{\sum_k \prod_i P(X_i|C_k) \cdot P(C_k)}$$

predviđanje klase se onda radi kao:

$$k^* = arg \max_k \prod_i P(X_i|C_k) \cdot P(C_k)$$

$P(a|b) = \frac{count(a \wedge b)}{count(b)}$, $P(a) = \frac{count(a)}{N}$

Underflow

Zbog proizvoda velikog broja verovatnoća, može doći do underflow-a u računarskoj reprezentaciji brojeva. Zbog toga se u implementacijama najčešće $P(C_k|\boldsymbol{X})$ menja sa monotonom transformacijom $log\left(P(C_k|\boldsymbol{X})\right)$, kako bi se proizvod prebacio u sumu koja je numerički stabilnija.

Smoothing

Empirijske procene verovatnoća nisu stabilne na malim uzorcima (tj. mnogo zavise od uzorka kada je uzorak mali), pa se stoga radi *smoothing* procenjenih verovatnoća (*k* je broj kategorija atributa).

$P(x) = \frac{count(x)+1}{N+k}$, što se zove *Laplace smoothing*

ili generalno:

$P(x) = \frac{count(x)+\alpha}{N+k\cdot\alpha}$, što se zove *Additive smoothing*, a α se ponekad zove *pseudocount*

Dodavanje *pseudocount*-a se tumači kao da za $P(x)$ postoji apriori verovanje da $P(x)$ može da uzme bilo koju vrednost (kada je $\alpha = 1$) ili da imamo predznanje da $P(x)$ treba da bude u određenom opsegu vrednosti (vidi Sliku 1). Veće vrednosti *pseudocount*-a sugerišu naše jaše apriori verovanje i zahtevaju više podataka kako bi se to verovanje promenilo. α može biti različito za različite klase, a ukoliko je isto, to daje apriori jednaku težinu svim klasama.

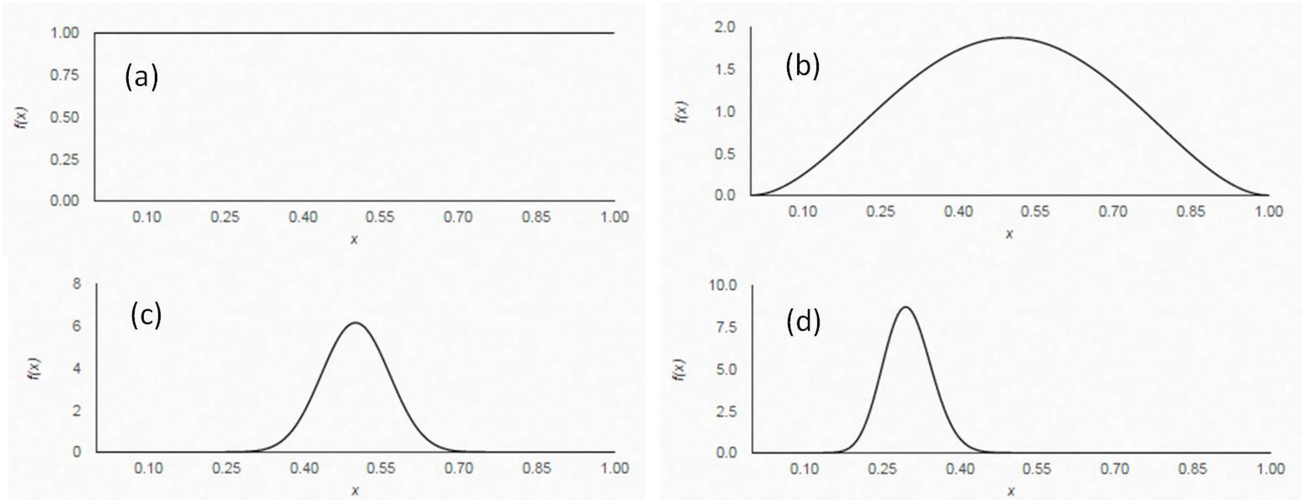
U slučaju binarne promenljive x, ova apriori verovatnoća se može modelovati *Beta* distribucijom (Slika 1), a u slučaju viševrednosne promenljive, ona odgovara *Dirihle* distribuciji.

Ovakva rezonovanja (korišćenje apriori verovanja) su osnov za Bajesov pristup u Mašinskom učenju, i pojavljuju se u velikom broju modela[1].

Smoothing tehnika se uglavnom koristi kada imamo manje podataka, ali ne škodi i kada se koristi sa puno podataka. Najzad, vrlo je korisna kada imamo puno podataka, ali je puno šuma u njima, tj. ne može im se potpuno verovati.

---

[1] ovo je takođe protivno klasičnoj (frekvencionističkoj) statistici, koja ne prihvata takvo subjektivno "verovanje", kao ni da parametre treba tretirati kao slučajne promenljive



Slika 1. Apriori verovanje o verovatnoći $P(x)$ prikazano beta-distribucijom. (a) Neinformativna raspodela sa α=1; (b) Slabo apriori verovanje u jednakost klasa α=3; (c) Jako apriori verovanje u jednakost klasa α=30; (d) Jako apriori verovanje u odnos klasa .3/.7, sa α_1=30 i α_2=70

Numeričke promenljive

U slučaju kada su promenljive X_i numeričke, verovatnoća $P(X_i|C_k)$ se može modelovati nekom kontinualnom distribucijom, a najčešće *Normalnom* distribucijom[2]:

$$P(X_i|C_k) = \frac{1}{\sqrt{2\pi\sigma^2}}\, e^{-\frac{(X_i-\mu)^2}{2\sigma^2}}$$

gde su parametri koje treba proceniti iz podataka (uzorka) μ i σ, i to za svaku klasu posebno.

Naive Bayes model, iako za cilj ima da proceni verovatnoću $P(C|\boldsymbol{X})$ radi predviđanja klase, on to radi računajući (modelujući) verovatnoće $(\boldsymbol{X}|C)$, što ga čini *generativnom modelom*, nasuprot *diskriminacionim modelima* koje ćemo kasnije obraditi. To takođe znači da se sa Naive Bayes modelom mogu generisati slučajni (random, virtuelni) uzorci iz svake klase, što ponekad može biti korisno.

Korisne komande

Matlab	**Python**	**R**
histc	pandas.crosstab	?
HashMap (java)	dict	?
normpdf	scipy.stats.norm.pdf	?
...	...	...

[2] Normalna distribucija je u skladu sa pretpostavkom Naive bayes modela, koja kaže da su atributi X_i međusobno nezavisni ukoliko poznajemo klasu (uslovno nezavisni). Normalna raspodela (uslovljena klasom: $P(X_i\ |C_k)$) takođe pretpostavlja da na X_i najviše utiče klasa, a ostatak čine puno "sitnih" faktora koji konstituišu normalnu rasporedu (po centralnoj graničnoj teoremi).