Naiv Bayes

1 Bevezető

Az alábbiakban a következő jelöléseket használjuk: $\{(\mathbf{d}_i, y_i)|i = 1, \ldots, \ell\}$ a tanulási adathalmaz, ahol $\mathbf{d}_i$ az i. dokumentumot jelöli, y_i pedig annak címkéjét (spam vagy ham, azaz -1 vagy 1). Minden dokumentumot úgy fogunk fel, mint egymástól független szavak sorozatát, és annyiszor tekintünk egy szót, ahányszor szerepel a dokumentumban; a szavakat a w_k szimbólummal jelöljük.

2 Naiv Bayes

A következőt akarjuk kiszámítani/megbecsülni:

$$c^* = \arg\max_i P(c_i|\mathbf{d}_j)$$

vagyis meg akarjuk határozni, melyik a legvalószínűbb osztály, ahol i végigfut az összes lehetséges osztályon, kategórián. Ez a Bayes-tétel értelmében felírható úgy, mint

$$P(c_i|\mathbf{d}_j) = \frac{P(\mathbf{d}_j|c_i)P(c_i)}{P(\mathbf{d}_j)}$$

ahonnan – az arg max miatt – a nevező elhagyható, mivel adott dokumentum esetén ez ugyanaz lesz minden osztályra, vagyis a köv. kifejezéssel dolgozunk tovább:

$$P(\mathbf{d}_j|c_i)P(c_i)$$

A modell "naivitása" onnan jön, hogy minden feature-t (jelen esetben szót) egymástól függetlennek tekint, azaz a fenti összefüggésből a köv. lesz:

$$P(\mathbf{d}_j|c_i)P(c_i) = P(c_i) \prod_{w_k \in \mathbf{d}_j} P(w_k|c_i)$$

(A $w_k \in \mathbf{d}_j$-t úgy kell értelmezni, hogy a szót többször is figyelembe vesszük, annyiszor, ahányszor a dokumentumban megjelenik; ezért nem a leghelyesebb az $\in$ jelölésmód. Ha jobban tetszik, akkor lehet ténylegesen halmazoknak tekinteni a dokumentumokat, ekkor $P(w_k|c_i)^{f(w_k,\mathbf{d}_j)}$ a helyes kifejezés, ahol $f(w_k, \mathbf{d}_j)$ a w_k szó előfordulásainak számát jelöli a $\mathbf{d}_j$ dokumentumban.)

Valószínűségeket nem jó dolog szorozni, mivel nagyon kicsi szám lehet a végeredmény, amit nem tudunk megfelelő pontossággal ábrázolni, ezért döntések

esetén, ha lehetséges, jobb összegzést használni. Ezt megtehetjük úgy, hogy alkalmazzuk a log függvényt (pl. 2-es alapút) – amit megtehetünk, mivel monoton növekvő függvény. A naiv Bayes osztályozónk tehát a következőképpen fog kinézni:

$$c^* = \arg\max_i \left[\log P(c_i) + \sum_{w_k \in \mathbf{d}_j} \log P(w_k|c_i) \right]$$

2.1 Bináris osztályozás

Két osztály esetén (amilyen a tartalom alapú spamszűrés is) elegendő, ha a $P(c_i|\mathbf{d}_j)$ valószínűségek arányát vesszük, azaz

$$\frac{P(c=1|\mathbf{d}_j)}{P(c=-1|\mathbf{d}_j)} = \prod_{w_k \in \mathbf{d}_j} \frac{P(w_k|c=1)}{P(w_k|c=-1)} \cdot \frac{P(c=1)}{1-P(c=1)}$$

Ha ez az arány 1-nél nagyobb, a dokumentum ham (azaz nem spam), ellenkező esetben spam. Vagyis, a logaritmusát véve ennek (pl. 2-es alapút), a kifejezés előjele adja meg a prediktált osztályt:

$$c^* = \sum_{w_k \in \mathbf{d}_j} \log \frac{P(w_k|c=1)}{P(w_k|c=-1)} + \log \frac{P(c=1)}{1-P(c=1)}$$

2.2 Paraméterek becslése

A modell paramétereit a tanulási adatok alapján a következőképpen tudjuk megbecsülni:

$$\begin{aligned} P(w_k|c_i) &= \frac{c(w_k, c_i)}{\sum_{w_k \in V} c(w_k, c_i)} \\ P(c_i) &= \frac{|c_i \text{ osztály}|}{|\text{összes tanulási adat}|} \end{aligned}$$

ahol $c(w_k, c_i)$ a w_k szó és a c_i osztály együttes előfordulásainak számát jelenti (= összesen hányszor fordult elő a w_k szó c_i dokumentumaiban), V pedig a tanulási adatok szótára (= az összes különböző szó halmaza, amely megjelent a tanulási adatokban).

3 Additív simítás (*Additive/Laplace/Lidstone smoothing*)

A simítás lényege, hogy zérónál nagyobb valószínűségeket rendeljünk a tanulási halmazban nem látott adatokhoz, szavakhoz, így jobb becslést kapva. Additív simítás esetén a paraméterek a következőképpen néznek ki ($P(c_i)$ nem változik):

$$P(w_k|c_i) = \frac{c(w_k, c_i) + \alpha}{\alpha|V| + \sum_{w_k \in V} c(w_k, c_i)}$$

ALG 1 Félig felügyelt naiv Bayes.

$\mathcal{D}_0$ = címkézett tanulási adatok
$\mathcal{D}_1$ = címkézetlen tanulási adatok
while nem változnak a paraméterek **do**
    Tanítsuk be, azaz számoljuk ki a naiv Bayes paramétereit $\mathcal{D}_0$ alapján.
    $\mathcal{D}_2 = \emptyset$
    for $\mathbf{d} \in \mathcal{D}_1$ **do**
        if $P_{\text{nagyobb}}/P_{\text{kisebb}} \geq \theta$ **then**
            $\mathcal{D}_2 = \mathcal{D}_2 \cup \{(\mathbf{d}, \text{prediktált címke})\}$
        end if
    end for
    $\mathcal{D}_0 = \mathcal{D}_0 \cup \mathcal{D}_2$
    $\mathcal{D}_1 = \mathcal{D}_1 \setminus \mathcal{D}_2$
end while

ahol $\alpha \in (0, 1]$.

(A feladat: keressünk meg α optimális értékét kereszt-validálással.)

4 Félig felügyelt tanulás naiv Bayes-szel

A félig felügyelt tanulás alapötlete az, hogy ha vannak címke nélküli adataink (amiből általában több van, mint címkézettből, könnyen beszerezhető stb.), használjuk fel azokat, javítva ezáltal a predikciókat. A kérdés az, hogy hogyan is tudjuk ezeket felhasználni?

Naiv Bayes esetén az egyik alkalmazható módszer a következő: tanítsuk be (azaz határozzuk meg a paramétereket) a címkézett tanulási adatok alapján, majd határozzuk meg a címkézetlen tanulási adatok osztályait. Ha eléggé biztos a döntés (azaz $P_{\text{nagyobb}}/P_{\text{kisebb}} \geq \theta$), akkor adjuk az illető dokumentumot a prediktált címkéjével együtt a tanulási adathalmazhoz. Ezután pedig tanítsuk újra a naiv Bayes osztályozót az új tanulási halmaz alapján. Az eljárást addig folytatjuk, amíg a paraméterek nem változnak (= $\mathcal{D}_0$ a halmaz nem bővül). A pszeudokód az ALG 1 algoritmusban látható.