Sztring- és *n*-gramm-kernelek

Az SVM esetén az 5. fejezetben láttuk, hogy nemlineáris dimenziónövelő transzformációkkal bizonyos osztályozási feladatok lineárisan szeparálhatóvá válnak.

A szövegosztályozás során van két kifogásolható lépés: a szózsákmodellel elvesztünk információkat a szövegből, amelynek következtében az esetleg nem lesz lineárisan szeparálható, majd nemlineáris transzformációkkal megpróbáljuk újra azzá tenni. Kézenfekvő lenne egy olyan reprezentáció, amely e két lépés kihagyásával több információt őriz meg a szövegből.

A skaláris szorzat, illetve ennek általánosabb változata a kernelfüggvény tulajdonképpen két dokumentum hasonlóságát mondja meg. A kapott modell pedig lényegében azt mondja, hogy a tesztdokumentumhoz hasonlóbb dokumentumok kategoriáját nagyobb súllyal vegyük figyelembe a tesztdokumentum kategóriájának meghatározásakor.

Egy jó reprezentációtól az alábbi három tulajdonságot várjuk el:

- a szöveges dokumentumokat egy vektortérbe transzformálja, és a vektorok közötti skalár szorzat adja meg két dokumentum hasonlóságát. Ezáltal a Mercer-tétel feltételei teljesülnek.
- minél hasonlóbbnak találunk két dokumentumot, annál nagyobb a skalárszorzat.
- a skalárszorzat kiszámításához ne kelljen vektortérbe transzformálni a dokumentumokat, hanem a két dokumentumból közvetlenül gyorsan kiszámítható legyen.

A sztring- és *n*-gramm kernelek megfelelnek ezeknek a követelményeknek.

Az *n*-gramm kernelek (NGK) esetén a szöveg minden *n* darab szomszédos karakteréből (a szóközt is beleértve) külön jellemzőt képezünk, ahol *n* egy rögzített konstans. Pl. a *bányászat* szó trigrammjai: *bán, ány, nyá, yás, ász, sza, zat.* Az *n*-gramm kernelek elsősorban akkor hasznosak, ha a szövegben hibák vannak — pl. elírások vagy egy optikai karakterfelismerő hibái. Természetesen a szomszédos betűk helyett tekinthetünk szomszédos szavakat is, ami a szózsákmodellhez képest több információt őriz meg a szövegből.

A sztringkerneleket (string subsequence kernel; SSK) Lodhi és társai javasolták dokumentum osztályozási feladatokra (Lodhi et al, 2002). A sztringkernelek lényegesen bonyolultabbak, mint az *n*-gramm kernelek. Két szabad paraméterük van: n és λ. Az előbbi a karakterek számát szabályozza, az utóbbi a súlyértéket adja meg.

A szavak (dokumentumok) minden n darab karaktert tartalmazó nem feltétlenül egybefüggő részszavából (dokumentumszegmensből) jellemző lesz: pl. $n = 2$ esetén a $\mathbf{x} =$ „oldal” szóból először az alábbi listát állítjuk elő:

$$[\ ol\_\_\_ : \lambda^2,\ \_ld\_\_ : \lambda^2,\ \_\_da\_ : \lambda^2,\ \_\_\_al : \lambda^2,\ o\_d\_\_ : \lambda^3,\ \_l\_a\_ : \lambda^3,\ \_\_d\_l : \lambda^2,\ o\_\_a\_ : \lambda^4,\ \_l\_\_l : \lambda^4,\ o\_\_\_l : \lambda^5\ ].$$

$n = 3$ esetén az alábbi listát állítjuk elő ugyanebből a szóból:

$$[\ old\_\_ : \lambda^3,\ \_lda\_ : \lambda^3,\ \_\_dal : \lambda^3,\ ol\_a\_ : \lambda^4,\ \_ld\_l : \lambda^4,\ o\_da\_ : \lambda^4,\ \_l\_al : \lambda^4,\ ol\_\_l : \lambda^5,\ o\_\_al : \lambda^5,\ o\_d\_l : \lambda^5].$$

Vagyis $n = 2$ esetén az összes lehetséges módon megtartunk 2 karaktert, $n = 3$ esetén az összes lehetséges módon megtartunk 3 karaktert. Mivel az „oldal” szó 5 hosszú, ezért $\binom{5}{2} = 10$ illetve $\binom{5}{3} = 10$ eleműek a listáink.

A továbbiakban csak az $n = 2$ esetet vizsgáljuk. A kapott lista elemiből töröljük az _ karaktert, és az ismétlődő elemekhez rendelt súlyokat összeadjuk (ismétlődést most csak az *ol* és az *o___l* okoz):

$$[\ al : \lambda^2,\ \ da : \lambda^2,\ \ dl : \lambda^2,\ \ la : \lambda^3,\ \ ld : \lambda^2,\ \ ll : \lambda^4,\ \ oa : \lambda^4,\ \ od : \lambda^3,\ \ ol : \lambda^2 + \lambda^5\ ].$$

Ezt a listát tekinthetjük úgy, mint egy dokumentumot, amelyben az *al* sztring súlya λ^2, a *da* sztring súlya λ^2, …, az *ol* sztring súlya $\lambda^2 + \lambda^5$. Minden szóhoz hozzárendeljük a vektortér egy koordinátáját, a koordináta súlya pedig a listában megadott súly lesz, így adódik $\phi(\mathbf{x})$.

A sztringkernelek használata során először rögzítjük két paramétert, n-et és λ-t. Ezután két dokumentum, d_1 és d_2 hasonlóságának kiszámításához kiszámítjuk mindkettőnek a sztringkerneleknek megfelelő reprezentációját, $\phi(d_1)$-et és $\phi(d_2)$-t, majd ezeket a vektorokat skalárisan összeszorozzuk. A d dokumentum (hosszát jelöle $|d|$) reprezentációja kiszámításának menete általános esetben:

- válasszunk ki az összes lehetséges módon n karaktert, az eredeti sorrendet pontosan megtartva, a nem kiválasztott karaktereket jelölje _. Ez pontosan $\binom{|d|}{n}$ darab kiválasztást jelent. Tegyük ezeket az elemeket egy listába.
- A lista minden eleméhez rendeljünk hozzá egy súlyt: λ^k, ahol k 2-vel több, mint a legelső és a legutolsó kiválasztott karakter között levő karakterek száma.
- Töröljük a lista minden eleméből az _ karaktert. Ha lesznek ismétlődések, adjuk össze a súlyukat, és az ismétlődő elemek helyére egyetlen egy elem kerüljön, az összegzett súllyal.

- Feleltessük meg a lista elemeit (részszavak) egy vektortér koordinátáinak (ugyanaz a megfeleltetés az összes d dokumentum esetén). A vektortér megfelelő koordinátája a lista megfelelő elemének súlyát kapja.

Az itt leírt módszer meglehetősen lassú, $O(\binom{|d|}{n})$ futási idejű, azonban dinamikus programozással a kernelfüggvény értéke (a skalárszorzat) kiszámítható $O(n \cdot |d_1| \cdot |d_2|)$ időben is (Lodhi et al, 2002).

A hivatkozott cikkben leírt kisérletek során az n-gramm kernelek mindig jobbnak bizonyultak a hagyományos tf-idf modellnél a Reuters-21578 korpuszon, $n = 5$ mellett. A sztringkernel tekinthető az n-gramm kernel általánosításának: nagyon kis λ esetén a nem összefüggő részszavak sokkal kisebb súlyt kapnak, mint az összefüggőek, így $\lambda \to 0$ esetén a sztringkernelek és az n-gramm kernelek egyre hasonlóbb eredményt adnak. Így aztán nem is csoda, hogy sztringkernelek jobbnak bizonyultak, mint az n-gramm kernelek. A meglepő az, hogy az SSK az NGK-hoz képest a vizsgált 4 kategória (*earn, acq, crude, corn*) közül mindössze 1 esetben volt lényegesen jobb, és 1 esetben kicsit jobb.

Forrás:

H. Lodhi, C. Saunders, J. Shawe-Taylor, N. Cristianini, and C. Watkins. Text classification using string kernels. *J. of Machine Learning Research*, 2:419–444, 2002.