Predavanja VI

Tags Done

Zastitni kod i verovatnoca greske

Zadatak je da unosenjem redudanse smanjimo verovatnocu greske pri slanju kroz kanal.

```
Definicija (M,n) kod za kanal (X,p(y|x),Y)
Poruka je neki od brojeva (1,...M). Potrebno je svaki od tih
brojeva kodirati tako sto mu pridruzimo neku n-torku ulaznih
simbola koje saljemo kroz kanal x(k).
Na prijemu od N primljenih simbola, potrebno je da nadjemo
nacin da otkrijemo koja je poruka poslata.

Ukoliko posaljemo poruku I, kazemo da je doslo do greske
ukoliko nasa funkcija dekodiranja nam kaze da smo dobili pogr
izlaz.

Na koji nacin formitati funkcije x i g, tako da kodiranje bud

Kodni kolicnik:
R = odnos minimalnog broj bitnova za predtavljanje od 1...M s
odnosom stvarnog broj bitnova, sa koliko simbola predstavljam
svaki od brojeva.

Kontantovali smo da vazi, ukoliko je odnos manji od kapacitet
tada mozemo da konstruisemo niz kodova tako da maksimalna ver
tezi nuli kada n -> beskonacno.
Ako je R < C, za dovoljno dugacke kodne reci da postignemo
proizvoljno malu verovatnocu greske.

Druga Shannonova teorema:
Za slucajno generisan kod verovatnoca je proizvoljno mala za
dovoljno veliko n. Mane jesu sto je potrebno dovoljno veliko
za realizaciju, moze n da bude veliko toliko da je nemoguce k
```

```
takav kod, a cak i da je moguce operacije kodiranja i dekodir
trebaju da se odvijaju brzo.

Problemi u praksi sa konkretnim kodovima:
- tesko se konstuisu oni koji imaju dobre performanse
- tesko se racuna verovatnoca greske
- komplikovana i neefikasna implementacija kodera i dekodera

Moguce je dostici Shannonovu granicu.
```

Kodovi u praksi

```
Pretpostavke:
- q-arni simetricni kanal:
X = Y, |X| = q, p(x|x) = 1 - (q-1)e, p(y|x) = e za x!=y
- struktura (algebarska) nad X, tj nad kodom C

Ove pretpostavke su ensencijalne da x i g
mozemo da konstruisemo.

Hammingovo rastojanje predstavlja broj
pozicija na kojima se ti vektori razlikuju.
C = {x(0),x(1),x(2),x(3)} = {0000,0011,1100,1111}
i neka je y = 1000. Tada je:

d_n(x(0),y) = d_n(0000,1000) = 1
d_n(x(1),y) = d_n(0011,1000) = 3
d_n(x(2),y) = d_n(1100,1000) = 1
d_n(x(3),y) = d_n(1111,1000) = 3

Ukoliko smo primili y = 1000, najverovatnije
je poslato ili x(0) ili x(2). U oba slucaja
(g(y)) = 0 ili g(y) = 2 dobijamo najbolji mogci
dekoder. Implementacija dekodera na nacin da redom
pretrazuje kodne reci i racuna minamalno.
Hammingovo rastojanje najcesce nije isplativo
```

```
zato sto je broj kodnih reci M suvise veliki.

lema:
ML dekoder za q-arni simetricni kanal dat je sa
g(y) = i, tako da je d_h(x(i),y) minimalno.

d_h(x,y) = 0 akko x = y
d_h(x,y) = d_h(y,x) - simetricno
d_h(x,z) <= d_h(x,y) + d_h(y,z) -
nejednakost trougla
```

ML dekoder

Na koji nacin cemo najbolje konstruisati funkciju g.

Kada dobijemo rec Y, trebamo da procenimo sta

je poslato. Ukoliko pretpostavimo da su pocetne

poruke od 1...M - uniformno raspodeljeno.
Kada primimo Y, nekako je najlogicnije je da pogledamo uslovne verovatnoce.

```
g(y) = w
p(y|x(w)) = max p(y|x(w))
```

Koja je to rec najverovatnije posalta ukoliko je Y primljena. Ako zelimo da budemo u potunosti precizni je uslovna verovatnoca :

```
g(y) = w
p(w|y) = max p(w|y)
```

Tada se dekoder naziva MAp. Ako je ulazna raspodela uniformna onda mozemo da koristimo formulu:

```
p(y|x(w))
```

Ukoliko ne znamo nista od ulaznoj i izlaznoj reci onda za svako omega moramo da racunamo uslovnu verovatnocu, tada moze M da bude veliko.

Umesto da racunamo verovatnoce iz kanala, racunacemo samo Hammingovo rastojanje kako bismo izbegli pretpostavke.

```
p(y|x) = alfa - doslo je do greske
i 1-q-1 ako nije doslo do greske

ako je bez memorije, znaci da svako q koje
dolazi je nezavisno, ne zavisi od prethodnog
koji je dosao.
Kada su kordinate iste, dobijamo alfa.
Kada su razlicite, dobijamo delta.

Broj kordinata koje su iste za x i y, alfa se
pojavljuje odredjeni broj puta, a delta na
mestima gde to nije isto.

alfa < 1 - q =>
Ako je alfa / delta < 1 da je veca
verovatnoca da nije doslo do greske vec da
se pojavila neka od greska.
Tada dobijamo da je kolicnik manji od 1.

p(y|x) = p(y1|x1)p(y2|x2)...p(yn|xn) =
             = alfa^d_h(x,y) * delta^n-d_h(x,y)
             = delta^n * (alfa/delta) ^ d_h(x,y)
Kako je delta^n konstanta u funkciji od n
imamo da je p(y|x) maksimalan ukoliko je
d_h(x,y) minimalan.

g(y) = argmax p(y|x(i)) = argmin d_h(x(i),y)

Argmin i argmax smo oznacili indeks za
koji se dostize maksimum odnosno minimum
odgovarajuceg indeksa.
```

Sfera oko (kodne) reci x e X^n. Skup reci koje nastaju od x usled ne vise od s gresaka.

```
Z_s(x) = {y e X^n | d_h(x,y) <= s)
ovde y oznacava sve reci koje smo dobili
```

```
ako je doslo do manje od s gresaka
```

U kontekstu gresaka Hammingovo rastojanje,
Ako smo primili Y, a poslali X i hammingovo rastojanje je jednako s. To znaci da se desilo s gresaka.

Mozemo da kazemo ukoliko sve n-torke Y iz sfere
se slikaju u i pomocu funkcije g(Z_s(x(i))) = {i}
da tada kod ispravlja s gresaka. To znaci da koje god Y dobijeno sa manje od s gresaka od x_i, dekoder ce to Y slikati u indeks i, korektno ce zakljuciti koji indeks je poslat.

Za kod kazemo da detektuje s gresaka ukoliko neka druga kodne rec ne pripada skupu g(Z_s).

Ako posaljemo x_i i desi se s greska, znaci da nema sanse da ne zakljucimo da se nije dogodila ni jedna greska. Primljeno je y, da li je to tacna rec.

Ukoliko neka druga kodna rec x_j pripada kodnoj sferi
oko kodne reci X_i, onda greske mogu da se sfere tako podese da y bude x_j, tada moze da se zakljuci da ne postoji s gresaka, a zapravo to nam je lose tvrdjenje jer se desilo s gresaka.
Ukoliko je to nemoguce, kazemo da kod detektuje ukupno s gresaka.

Kodno rastojanje je minimalno Hemmingovo rastojanje izmedju dve kodne reci.

```
d(C) = min d_h(x(i),x(j))
```

Ako imamo jmbg i promenimo neke brojeve. Ako uzmemo jedan moguc i ubacimo neke brojeve, formiramo dekoder koji ce sve pogresne da slika u one od kojih je nastao - tada dekoder ispravlja gresku. A detektuje, ako mi ne dobijemo neki drugitacan jmbg.

Drugo vazno svojstvo, ondosi se na performanse koda i na kodno rastojanje izmedju dve kodne reci. Kod detektuje s greska ukoliko je kodne rastojanje vece od s, a ispravlja s gresaka ukoliko je kodno rastojanje vece od 2s.

```
Ako je kodno rastojanje d(C) > s, ne postoje
x_i i x_j koje se ne nalaze na rastojanju
manje i jednakom od s.
```

```
Z_s je skup reci koje se nalaze na rastojanju
od s.

Ako je D(C)>s, onda znamo da je x_j nije
element skupa Z_s.

Ukoliko je D(C) > 2s i pretpostavimo
da tvrdjenje ne vazi, tj postoji y iz
sfere Z_s, tako da se g(y) = j koje je razlicito
od i, mozemo da posmatramo rastojanje
izmedju x_i, x_j.
d_h(x(i),x(j)) <= d_h(x(i),y) + d_h(y,x(i)) <=2s

g(y) vraca najblizu kodnu rec za koju je d_h
minimalna. Koristimo definiciju g, funkcije g
koji vraca kodnu rec najblizu pod hamm
rastojanjem. Dolazimo do kontradikcije
i znaci da je d(C) >= 2s.
```

```
Primer:
C = {000,111} spada u grupu perfektnih kodova.
Z1(000) = {000,001,010,110}
Z1(111) = {111,110,101,011}
Posto je d_h(000,111) = 3, ovaj kod ispravlja s = 1 gresku.
Na slican nacin, ponavljajuci kod duzina 2k+1, ispravlja k
gresaka. Mejdutim i dalje ostaje cinjenica da se ovako dobre
placaju malim brojem kodnih reci (M = 2), odnosno kodnim koli
koji tezi 0.
```

Hammingov uslov

Ako kazemo da kod ispravlja s gresaka onda je ogranicenje:

```
M <= q^n / sum [(nk)(q-1)^k]
```

M broj poruka koji mozemo da posaljemo je maksimalan broj kodnih reci koje mozemo da postavimo u prostoru tako da kod ispravlja s greska.

Uslov je da rastojanje izmedju bilo koje dve reci bude 2s.
Zelimo da vidimo koliko sfera mozemo da upakujemo, sto je manj sfera to je manje M i kolicnik je manji - kod je neefikasniji.

Potrebno nam je da sfere budu medjusobno disjunktne. Da bismo videli koliko reci imamo u svakoj od njih:

```
1 2 3 4
nemamo gresaka
1 2 3 4

ukoliko imamo jednu gresku
X 2 3 4
1 X 3 4
1 2 X 4
1 2 3 X
na n mesta mozemo da odaberemo mesto za greku

X moze da bude bilo koji simbol q, ali bez onog koji je tacan
tako da X moze da bude q-1 nacin odabran.
```

Kada sve sfere za kodne reci uzmemo imamo M * A ≤ q^n.
Perfektni kodovi, za njih vazi jednakost da su sfere idealno upakovane.
Izvlacimo maksimum performansi.

Pretpostavka da je kanal kroz koji saljemo podatke q-narni.

Sada trebamo da vidimo stukturu nad skupovima kodnih reci X^n.

Pretpostavljamo da je struktura konacno polje. Skup F na kome su definisane + i * koje zadovoljavaju sve uslove vezane za + i * i broj elemata ovog skupa je konacan. F = {0,1} pri cemu operacija + i * su operacije po mofulu 2.

```
0 + 0 mod 2 = 0
1 + 1 mod 2 = 0
0 + 1 mod 2 = 1
1 + 0 mod 2 = 1
```

U principu operacija + predstvlja ekskluzivno ili.
A operacija * :

```
0 * 0 = 0
1 * 1 = 1
1 * 0 = 0
0 * 1 = 0
```

Jednaka je operaciji logocko ili.
Ako kroz kanal saljemo 0, odnosno jedinice. Ukoliko je kanal veci od 2, tada imamo slozenija polja. Ukoliko imamo p elemenata onda je x+y mode p i x* y mod p. Postoje takodje i konacna polja koja su p^m.

Deljenje je racunanje inverznog elementa:

```
po modulu 7:
5 / x = 1. Direktnom proverom za x = 1..6 nalazimo da je to x
Dakle x = 3 = 5^-1.
```

Ogranicenje je da p mora da bude prost broj. X i Y moraju da budu jendaki GF(p^m). Polja GF(p^m) formiraju se komplikovanije (elementi su polinomi a nad GF(p) a operacije su po modulu ireducibilnog polinoma f(x)).

U daljem radu radimo sa pretpostavkom da kroz kanal saljemo 0,1. Direktno de uopstava ako se radi o nekim drugim vrednostima.

Linearni blok kodovi

Skup X^n je vektorski prostor nad poljem X = Fq. Operacije x+y i alfa*x definisu de pokordinatno:

```
x+y = (x1+y1, ... x_n+y_n)
alfa*x = (alfa*x1...alfa*x_n) - mnozenje vektora

Kod C je linearni blok kod nad X ukoliko je C vektorski podpr
```

Kodne reci su n-torke koje se ponasaju kao geometrijski vektori.
Ukoliko je dim C = k onda je u pitanju (n,k)-kod.
Linearni blok kod C je generisan sa k linearno nezavisno kodnih reci g1,g2...gk.
Svaka kodne rec moze da se predstavi kao linearna kombinacija ovih reci:

```
x = alfa_1 * g_1+...+alfa_k * g_k
g - generatori
gde su alfa_1...alfa_k e X, moze da uzme q razlicith vrednost
=> sto znaci da je M = q^k elemenata - broj kodnih reci

Svaki indeks u = 0,1...q^k - 1 (M=q^k) pisemo u sistemu sa os
u = (v1...vk) e X^k
Kodiranje u -> u1*g1+ ... + uk*qk.

Skup indeksa poistovecujemo sa X^k, a skup kodnih reci je
podskup od X^n.

Primer:
g1 = (1,1,1,0,0)
g2 = (0,0,1,1,0)
g3 = (1,1,1,1,1)

kodiranje:
u = (u1,u2,u3) => u1g1 + u2g2 + u3g3
=> (u1 + u3, u1 + v3, v1 + u2 + u3, u2 + u3,u3)

Kodne reci g1...gk, formiraju vise generatorske matrice G for
k x n.

G1 = [1 1 1 1 1] ,
G2 = [
1 1 1 0 0
0 0 1 1 0
1 1 1 1 1
]

Kodna rec se sastoji od informacionih i parity check -> siste

Informacioni simboli u na pocetku kodne reci x(u)
x(u) = [u, uA]. Elementarnim transformacijama, matrica G moze
da se svede na G = [I,A] -> u -> [u,uA]

Kodiranje mozemo da promenim tako sto radimo transformaciju o
```

```
tako da se sam skup kodnih reci ne menja. Tada se u menja, al
skup kodnih reci ne menja.
```