Predavanje II

Tags	Done

Kraftova nejednakost

Da li su prefiksni kodovi najznacajniji kodovi ako je jedini uslov duzina reci kodne reci.

McMillanova teorema

Svaki jednoznacni dekodabilni kod zadovoljava Kraftovu nejednakost, ne samo prefiksni kod, vec svaki drugi koji zadovoljava jednoznacno dekodabilan kod.

Spajanjem kodnih reci gde:

```
x = u_1u_2..u_k
= v_1v_2..v_l
=> k = l, u_i = v_i
```

Ukoliko rec predstavlja poruku (dobijena kodnim recima) to predstavljanje je jedinstveno.

```
|vk| = nk
=> zadovoljava Kraftovu nejednakost
```

za dokaz predpostavljamo da postoji k koji je prirodan broj gde:

```
(sum b^-ni)^k = sum b^(-ni1...+nik)

k = 2
b^(n1-n1)+b^(-n1-n1)

(sum b^ni)^k = sum(nj^b-j)

x = vi1...vik = vi1...vij
=> Ukoliko bi postajale dve
```

```
razlicite katorke koje
cine istu rec, znaci
da nisu jednoznacno dekodabilne.



mj <= b^j

(sum b^0nj) <= sum b&j - b^-j = kn
k-ti koren od k^n -> 1 (k->beskonacnosti)

kada k pustimo da tezi beskonacnosti

Kod takav da ne postoji poruka koja
se dobija istim brojem kodnih
reci na dva razlicita nacina.

Zakljucak:
Imamo kod koji je jednoznacno
dekodabilan
odredjene duzine reci, tada vazi
jednakost na osnovu McMillanove teoreme.
A na osnovu obrata, postoji pref
kod sa istim duzinama kodnih reci.

Ako neko tvrdi da neki kod
koji nije prefiksan, mozemo da
konstruisemo isti kod koji
je prefiksan iste duzine kodnih reci.

Prefiksni kodovi i njihova prednost
se osnosi na trenutne kodove.
Gde u trenutku kada nam stigne
kodna rec na osnovu azbuke,
mozemo u jendom prolazu da
dekodiramo kodne reci.
Algoritmi koji koriste kodne
reci u trenutku se nazivaju
online algoritmi.
```

Performanse koda

Pored duzine kodnih reci (niz duzine kodnih reci), bitnija je srednja duzina kodnih reci. Da bismo nju definisali potrebno je definisati da se kodovi na pocetku javljaju da nekom verovatnocom.

Pretpostavljamo za svkai simbol pojavljinu u kodnoj reci.

Srednja duzina kodnih reci bi bila srednja vrednost duzine svih kodnih reci, gde uracunavamo i verovatnocu pojavljivanja simbola kodne reci.

Ukupan broj simbola koji trebamo da prenesemo je jednak ukupnoj duzini kodne reci i verovatnoca pojavljivanja simbola kodne reci.

Potrebno je da kod konstruisemo tako da je srednja duzina kodne reci minimalna -nacin na koji se radi kompresija.

Prefiksni kod koji ima minimalnu duzinu kodnih reci za zadate ulazne verovatnoce.

Na osnovu ulaznih verovatnoca, na koji mozemo pronaci teorijsku donju granicu od koje ce ona svakako biti veca.

Diskretna slucajna promenljiva uzima prebrojiv skup vrednosti sa prebrojivim vrednostima.

Visedimenzionalne diskretne promenljve, gde imamo udruzene verovatnoce gde je x = xk i y = yj. Dva dogadjaja P(X = xk, Y = yj) mogu biti medjisobno nezavisni.

Bacanje dve kockice → dva medjusobno nezavisna ishoda. Ucenje i ocena na ispitu jesu dva medjusobno zavisna dogadjaja.

Potrebno je da utvrdimo na koji nacin mozemo da opisemo kolicinu informacija koju nosi neki dogadjaj. Kolicina informacija treba da bude neprekidna i opadajuca. Dva podatka koja su nezavisna dogadjaja i kolicina informacija koju oni nose zajendo treba da bude jednaka pojedinacno jednaka ponaosob.

```
p(x,y) = p(x)p(y) =>
I(p(x,y)) = I(p(x)) + I(p(y))
I(x) = -log_b p(x)

I(x) - kolicina informacja koju
nosi jedna dogadjaj

Nama je potrebna srednja vrednost
-> primer:
Otvorimo knjigu, nadjemo slovo i
pitanjekoje postavljamo koja je
kolicina informacija koju taj
podatak nosi kada
posmatramo celokupne informacije.

sum I(x)p(x)

entropija - srednja kolicina infromacija
H(X) = - sum p(x)log_b p(x)

b moze da ima proizvoljnu velicinu,
b je baza

Vrednosti za b:
b = 2

H(X) je u jedinicama bit ili shanon.
Posmatramo bit u memoriji,
koliku kolicni on informacija nosi

H(X) = - sum p(x)log_2 p(x)

H(1/2,1/2) = 1
ako imamo podjenake verovatnoce

kada p != 1/2, mozemo da ustedimo na
```

```
memoriji kada koristimo kompresiju
podataka.
```

Entropija u najjednostavnijem slucaju ima max vrednost, ako slucajna promenljiva p ima uniformnu raspodelu.

Minimalni srednji broj n se nalazi izmedju H(x) i H(x)+1.

Da bi pokazali kada entropija postize svoji maksimum, potrebna nam je lema.

Ukoliko su data dva niza brojeva pri cemu je zbir q1+...+qn manji od zbira p1+...+pn ⇒ - sum[p_i log_2 p_i] ≤ - sum[p_i log_2 q_i] (samo akko je pi = qi)

Ukolika je leva stana jednakosti p_1+...+p_n = 1, onda je suma jednaka entropiji P(x). Desna strana nejednakosti je krosna nejednakost (kroz entropija), akko je q_1+...+q_n takodje jednako 1.

Nacin na koji se lema primenjuje ⇒ maksimum entropije slucajne entropije koja uzima n vrednsoti, upravo logaritam od n.

Entropija:

```
- sum[p_i log_2 p_i] <= - sum[p_i log_2 q_i]
=> H(x) <= - sum[p_i log_2 q_i] <= - sum[p_i log_2 1/n]= log(
=> log(n) akko p_i = 1/n

Trebamo da odredimo q_i:
1 = p_1+..+p_n >= q_1+..+q_n
akko q_i = 1/n -> uniformna raspodela

Entropija postize maksimum kada je n jednaka uniformnoj rasod
```

Granice za srednju duzinu kodnih reci:

```
Srednja duzina kodne reci proizvoljnog jendoznacno dekodabiln
je veca jednaka od entropije kroz log_2 od broja simbola u iz
alfabetu.
```

```
n_v >= H(X)/ log_2(b)

- sum[p_i log_2 p_i] <= - sum[p_i log_2 q_i]
q_i = b^-n_i

p_1+...+p_a >= q_1+...q_a = b^-n_1+...+b^n_a

Na osnovu megmilove teoreme vazi kraftova nejednakost gde
je izraz manji jedna od 1.

Hb(X) = -sum[p_i log_b p_i] <= - sum[p_i log_b q_i]
= - sum[p_i log_b b^-n_i = sum[n_i p_i] = n_v

Pokazali smo ovu nejednakost. A jednakost vazi kada su polazn
verovatnoce jednake b^-n_i gde su n_i celi brojevi.
```