1 Osnove

1.1 Ponovitev logaritmov

- $log_a x = \frac{log_b x}{log_b a}$
- $log_b(\frac{x}{y}) = log_b x - log_b y$
- $x = b^y \implies log_b x = y$
- $log_2 x = log x$
- $0 log 0 = 0$

1.2 Entropija je povprecje vseh lastnih informacij:

$$H(X) = \sum_{i=1}^{n} p_i I_i = -\sum_{i=1}^{n} p_i log p_i$$

Lastnosti: je zvezna, simetricna funckija (vrsni red p_i ni pomemben, sestevanje je komutativno). Je vedno vecja od 0 ($p_i \geq 0 \to -p_i \log p_i \geq 0 \to H(X) \geq 0$) in navzgor omejena z $\log n$.
Ce sta dogodka **neodvisna** velja aditivnost: $H(X,Y) = H(X) + H(Y)$.
Vec zaporednih dogodkov neodvisnega vira: $X^l = X \times \cdots \times X \to H(X^l) = lH(X)$.

2 Kodi

2.1 Uvod

Kod sestavljajo *kodne zamenjave*, ki so sestavljene iz znakov **kodne abecede**. Stevilo znakov v kodni abecedi oznacujemo z **r**.
Ce so $\{p_1, \ldots, p_n\}$ verjetnosti znakov $\{s_1, \ldots, s_n\}$ osnovnega sporocila in $\{l_1, \ldots, l_n\}$ dolzine prejetih kodnih zmanjav, je povprecna dolzina kodne zamenjave

$$L = \sum_{i=1}^{n} p_i l_i$$

2.2 Tipi kodov

- **optimalen** - ce ima najmanjso mozno dolzino kodnih zamenjav
- **idealen** - ce je povprecna dolzina kodnih zamenjav enaka entropiji
- **enakomeren** - ce je dolzina vseh kodnih zamenjav enaka
- **enoznacen** - ce lahko poljuben niz znakov dekodiramo na en sam nacin
- **trenuten** - ce lahko osnovni znak dekodiramo takoj, ko sprejmemo celotno kodno zamenjavo

2.3 Kraftova neenakost Za dolzine kodnih zamenjav $\{l_1, \ldots, l_n\}$ in r znaki kodne abecede obstaja trenutni kod, iff

$$\sum_{i=1}^{n} r^{-li} \leq 1$$

2.4 Povprecna dolzina in ucinkovitost

Najkrajse kodne zamenjave imamo, ce velja:

$$H_r(X) = L \to l_i = \lceil -\log_r p_i \rceil$$

Ucinkovitost koda:

$$\eta = \frac{H(X)}{L \log_r}, \eta \in [0, 1]$$

Kod je **gospodaren**, ce je L znotraj:

$$H_r(X) \leq L < H_r(X) + 1$$

kjer je $H_r(X)$:

$$H_r(X) = -\sum_{i=1}^{n} \frac{\log_{p_i}}{\log_r} = \frac{H(X)}{\log_r}$$

2.5 Shannonov prvi teorem

Za nize neodvisnih znakov dozline n obstajajo kodi, za katere velja:

$$\lim_{n\to\infty} \frac{L_n}{n} = H(X)$$

pri cemer je $H(X)$ entropija vira X.
Postopek kodiranja po Shannonu:

1. znake razvrstimo po padajocih verjetnostih
2. dolocimo stevilo znakov v vsaki kodni zamenjavi (l_k)
3. za vse simbole izracunamo komulativne verjetnosti ($P_k = \sum_{i=1}^{k-1} p_i$)
4. P_k pretvorimo v bazo r. Kodno zamenjavo predstavlja prvih l_k znakov necelega dela stevila

2.6 Fanojev kod

Postopek kodiranja:

1. znake razvrstimo po padajocih verjetnostih
2. znake razdelimo v r cim bolj enako verjetnih skupin
3. Vsaki skupini priredimo enega od r znakov kodne abecede
4. Deljenje ponovimo na vsaki od skupin. Postopek ponavljamo, dokler je mogoce

2.7 Huffmanov kod

Huffmanov postopek kodiranja poteka od spodaj navzgor (Pri Fanoju je ravno obratno). Pri huffmanovem kodu imamo dve fazi:

1. Zdruzevanje
   (a) Posici r najmanj verjetnih znakov in jih zdruzi v sestavljeni znak, katerega verjetnost je vsota verjetnosti vseh znakov
   (b) Preostale znake skupaj z novo sestavljenim znakom spet razvrsti
   (c) Postopek ponavljaj dokler ne ostane samo r znakov
2. Razdruzevanje
   (a) Vsakemu od preostalih znakov priredi po en znak kodirne abecede
   (b) Vsak sestavljeni znak razstavi in mu priredi po en znak kodirne abecede
   (c) Ko zmanjka sestavljenih znakov, je postopek zakljucen

Pred kodiranjem, je vedno pametno preveriti, ce imamo zadostno stevilo znakov. Veljati mora:

$$n = r + k(r - 1), k \geq 0$$

Ce imamo premalo znakov, jih po potrebi dodamo s verjetnostjo $p = 0$.
Huffmanov kod lahko razsirimo tako, da vec osnovnih znakov zdruzujemo v sestavljene znake $\to$ bolj ucinkoviti kodi. Vendar naletimo na nevarnost kombinacijske eksplozije.

2.9 Aritmeticni kod

Je **hiter** in **blizu optimalnemu** kodu. Je manj ucinkovit kot Huffmanov, vendar se izogne kombinacijski eksploziji. Vsak niz je predstavljen kot realno stevilo $0 \leq R < 1$, kar nam pove, da daljsi kot bo niz, bolj natancno mora biti podano naravno stevilo R.
Postopek kodiranja(znakov ni potrebno razvrstiti):

1. Zacnemo z intervalom $[0, 1]$
2. Izbrani interval razdelimo na n podintervalov, ki se ne prekrivajo. Sirine podintervalov ustrezajo verjetnostim znakov. Vsak podinterval predstavlja en znak
3. Izberemo podintrval, ki ustreza iskanemu znaku
4. Ce niz se ni koncan, izbrani podinterval ponovno razdelimo (bne 2.tocka)
5. Niz lahko predstavimo s poljubnim realnim stevilom v zadnjem podintervalu

Ko dobimo realni interal, ga samo se pretvorimo v binarnega s pomocjo klasicnega pretvarjanja iz dec v bin stevilski sistem.