



ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΛΟΠΟΝΝΗΣΟΥ,
ΤΜΗΜΑ ΤΕΧΝΟΛΟΓΙΑΣ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

ΒΕΣ 04: ΣΥΜΠΙΕΣΗ ΚΑΙ ΜΕΤΑΔΟΣΗ ΠΟΛΥΜΕΣΩΝ

Ακαδημαϊκό Έτος 2007 – 2008, Χειμερινό Εξάμηνο

6 Νοεμβρίου 2007

Φροντιστηριακή Άσκηση 2: (I) Εντροπία,

(II) Κωδικοποίηση Huffman εικόνων αποχρώσεων του γκρι

(III) Προβλεπτική κωδικοποίηση εικόνων

(IV) Αριθμητική Κωδικοποίηση

I. Εντροπία

Ας υποθέσουμε ότι ένα αρχείο κειμένου περιέχει μόνο τους χαρακτήρες «a» και «b». Η συχνότητα που εμφανίζεται ο χαρακτήρας «a» στο κείμενο αναπαριστάται με το $P(a)$. Όμοια έχουμε το $P(b)$ για τον χαρακτήρα «b». Συνήθως αναπαριστάουμε τα $P(a)$ και $P(b)$ σαν πιθανότητες, έτσι έχουμε $P(a) + P(b) = 1$. Η εντροπία (ή η περιεχόμενη πληροφορία) του αρχείου κειμένου αυτού, μπορεί να υπολογιστεί όπως φαίνεται πιο κάτω:

$$H = -(P(a) \log_2 P(a) + P(b) \log_2 P(b))$$

Γενικά, αν έχουμε ένα σύνολο από n χαρακτήρες s_1, s_2, \dots, s_n , που ο κάθε ένας συμβαίνει με πιθανότητα $P(s_i)$, τότε:

$$H = -\sum_{i=1}^n P(s_i) \log_2 P(s_i)$$

- (1) Υπολογίστε το H για $P(a) = 0.2$, και $P(b) = 0.8$.
- (2) Αν χρησιμοποιηθεί η κωδικοποίηση Huffman για την αποθήκευση του κειμένου, πόσα bits χρειαζόμαστε μέσο όρο (συνήθως ονομάζεται μέσο μήκος κωδικοποιημένης λέξης (codeword));
- (3) Εναλλακτικά μπορούμε να κωδικοποιήσουμε το ίδιο αρχείο κειμένου χρησιμοποιώντας ζεύγη χαρακτήρων, δηλαδή, (a, a) , (a, b) , (b, a) , (b, b) . Εάν υποθέσουμε ότι οι εμφανίσεις των «a» και «b» είναι ανεξάρτητες, υπολογίστε τα $P(a, a)$, $P(a, b)$, $P(b, a)$, και $P(b, b)$. Ποια είναι η αντίστοιχη τιμή της εντροπίας (bits/σύμβολο); Ποια είναι η τιμή της εντροπίας σε bit ανά χαρακτήρα;
- (4) Βασισμένοι στο (3), ποια θα ήταν η κωδικοποίηση Huffman για κάθε ζεύγος χαρακτήρων; Ποιο είναι το αντίστοιχο μέσο μήκος κωδικοποιημένης λέξης για αυτό το σχήμα κωδικοποίησης; Ποιο είναι το μέσο μήκος κωδικοποίησης ανά χαρακτήρα;
- (5) Τι παρατηρήσεις μπορείτε να κάνετε, με βάση τις απαντήσεις σας στις πιο πάνω τέσσερις ερωτήσεις;
- (6) Τι συμβαίνει αν χρησιμοποιήσουμε τριάδες, δηλαδή, (a, a, a) , (a, a, b) ... (b, b, b) κ.τ.λ. για την κωδικοποίηση του κειμένου; Ποιο θα είναι το μέσο μήκος κωδικοποιημένης λέξης; Ποιο είναι το μέσο μήκος κωδικοποίησης ανά χαρακτήρα.

II. Κωδικοποίηση Huffman

Σας δίνεται η εικόνα *'pouf3.tif'* με αποχρώσεις του γκρι (intensity image) (κάντε download από <http://eclass.uop.gr/TST128/document/assignments/pouf3.tif>). Χρησιμοποιώντας τη Matlab διαβάστε την εικόνα (εντολή *imread*) και περάστε τη στο χώρο εργασίας της Matlab.

- (1) Πόσο χώρο μνήμης καταλαμβάνει η ασυμπίεστη εικόνα (χρησιμοποιήστε την εντολή *whos*);
- (2) Αποθηκεύστε την εικόνα ως *jpg* (χρησιμοποιήστε την εντολή *imwrite*)
- (3) Βρείτε πόσες διακριτές τιμές φωτεινότητας περιέχει η εικόνα (Υπόδειξη: Χρησιμοποιήστε τη συνάρτηση *hist* ή *imhist* για τη δημιουργία του ιστογράμματος της εικόνας)
- (4) Υπολογίστε τις συχνότητες εμφάνισης των παραπάνω τιμών και εκτιμήστε την αντίστοιχη πιθανότητα. Δημιουργήστε το σχετικό ραβδοδιάγραμμα (εντολή *bar*) και σώστε το ως εικόνα.
- (5) Κατασκευάστε τις κωδικές λέξεις (codewords) Huffman για τις μη μηδενικές τιμές φωτεινότητας της εικόνας.
- (6) Υπολογίστε την εντροπία και το μέσο μήκος κωδικής λέξης της εικόνας.
- (7) Υπολογίστε το βαθμό συμπίεσης που επιτυγχάνεται με χρήση του κώδικα Huffman. Συγκρίνετε το βαθμό συμπίεσης με το αντίστοιχο της συμπίεσης JPEG (μπορείτε να χρησιμοποιήσετε την εντολή *imfinfo* για το σκοπό αυτό). Αιτιολογήστε τη διαφορά στο βαθμό συμπίεσης.

III. Προβλεπτική κωδικοποίηση εικόνων

Συνεχίστε από το βήμα (1) της προηγούμενης άσκησης (έχετε επομένως φορτώσει στο χώρο εργασίας της Matlab την εικόνα 'rouf3.tif').

- (1) Αφού μετατρέψτε την εικόνα σε μονοδιάστατο διάνυσμα (π.χ. $g=f(:)$), υπολογίστε τις διαφορές ανάμεσα σε γειτονικά pixel της εικόνας (μπορείτε να χρησιμοποιήσετε την εντολή *diff* - βεβαιωθείτε ότι έχετε μετατρέψει το διάνυσμα g σε *double*) δημιουργώντας το διάνυσμα d των διαφορών των pixels.
- (2) Βρείτε πόσες διακριτές τιμές περιέχει η εικόνα διαφορών (το διάνυσμα d αντιστοιχεί στην πραγματικότητα στην εικόνα διαφορών) υπολογίζοντας το ιστόγραμμα του διανύσματος d (εντολή *hist*).
- (3) Με βάση τη μορφή των ιστογραμμάτων των ερωτημάτων II.(3) και III.(2) ποια από τις εικόνες f και d συμπιέζεται περισσότερο;
- (4) Υπολογίστε την εντροπία του διανύσματος d (δηλαδή της εικόνας διαφορών).
- (5) Εφαρμόστε κωδικοποίηση μήκους διαδρομής για το διάνυσμα d . Κάθε μη μηδενική τιμή του d στα διαστήματα $[-127 -1]$, $[1 127]$ κωδικοποιείται ως προσημασμένος ακέραιος. Ο κωδικός 00000000 χρησιμοποιείται ως σύμβολο που δηλώνει ακολουθία μηδενικών ακολουθούμενη από το πλήθος των συνεχόμενων μηδενικών. Δώστε την κωδικοποίηση των πρώτων 26 τιμών του διανύσματος d .
- (6) Υπολογίστε τη συμπίεση που επιτυγχάνεται με την πιο πάνω τεχνική.
- (7) Κάτω από ποιες προϋποθέσεις η ανωτέρω τεχνική είναι χωρίς απώλειες;
- (8) Περιγράψτε τη μεθοδολογία αποκωδικοποίησης της εικόνας.

Παράδειγμα: Σύμφωνα με τα παραπάνω η ακολουθία 3 0 0 0 0 0 -12 0 0 0 0 0 0 8 8 θα κωδικοποιηθεί ως:

3 !5 -12 !6 8 8 (ο κωδικός για το ! είναι ο 00000000)

IV. Αριθμητική κωδικοποίηση

Δίνονται τα παρακάτω τμήματα ψευδοκώδικα τα οποία επιδεικνύουν τη λειτουργία της Αριθμητικού Κωδικοποιητή. Γράψτε τις ανάλογες συναρτήσεις (m-files) σε Matlab που υλοποιούν τον κωδικοποιητή. Υποθέστε ότι ο κωδικοποιητής διαθέτει ένα αρχείο με της πιθανότητες των συμβόλων και τα αντίστοιχα διαστήματα (το \$ είναι το σύμβολο τερματισμού της ακολουθίας).

Symbol	Probability	Range
A	0.2	[0 0.2)
B	0.1	[0.2 0.3)
C	0.2	[0.3 0.5)
D	0.05	[0.5 0.55)
E	0.3	[0.55 0.85)
F	0.05	[0.85 0.9)
\$	0.1	[0.9 1.0)

Ελέγξτε την αποκωδικοποίηση της συμβολοσειράς CDEF\$.

<u>Estimation of Low and High values</u>	<u>Generating Codewords</u>
<pre>BEGIN low=0.0; high=1.0; range=1.0; while (symbol !=terminator) { get(symbol); low = low + range*Range_low(symbol); high = low + range*Range_high(symbol); range = high - low; } output a code so that low <= code < high END</pre>	<pre>BEGIN code=0; k=1; while (value(code) < low) { assign 1 to the k-th binary fraction bit; if (value(code) > high) replace the k-th bit by 0; k = k+1; } END</pre>

Απαντήσεις

MCL= Mean Codeword Length, MChL= Mean Character Length

I.(1): H =

I.(2): MCL =

I.(3): H= (bits/symbol) H= (bits/character)

I.(4): aa =

ba =

ab =

bb =

MCL =

MChL =

I.(5): Παρατηρήσεις:

I.(6): aaa =>

abb =>

bba =>

aab =>

baa =>

bbb =>

aba =>

bab =>

MCL =

MChL =

II.(1): Μνήμη ασυμπίεστης εικόνας:

bytes

II.(2): Εντολή για αποθήκευση εικόνας σε jpg:

II.(3): Αριθμός διακριτών τιμών φωτεινότητας:

Ιστόγραμμα:

II.(4): Συχνότητες εμφάνισης:

Πιθανότητες:

Ραβδόγραμμα:

II.(5): Κωδικές Λέξεις:

II.(6): $H =$

$MCL =$

II.(7): Βαθμός συμπίεσης με Huffman:

Βαθμός συμπίεσης με JPEG:

Η διαφορά στο βαθμό συμπίεσης οφείλεται:

III.(1): Εντολή υπολογισμού διανύσματος διαφοράς:

III.(2): Αριθμός διακριτών τιμών διαφοράς:

Ιστόγραμμα:

III.(3): Καλύτερα συμπιέζεται η εικόνα

επειδή:

III.(4): $H =$

III.(5): Κωδικοποίηση πρώτων 26 τιμών του d :

III.(6): Βαθμός Συμπίεσης:

III.(7): Προϋποθέσεις για να είναι η τεχνική χωρίς απώλειες:

III.(8): Περιγραφή αποκωδικοποίησης: