Πολυτεχνειο Κρητης
Σχολη Η.Μ.Μ.Υ
ΤΗΛ513 - Δορυφορικες Επικοινωνιες

Παράδοση: 15/5/2018
Προθεσμία: 29/5/2018

Project 2 Δορυφορικών Επικοινωνιών

Σκοπός είναι να εφαρμόσετε τον αλγόριθμο Viterbi, με τον οποίο ασχοληθήκατε στο προηγούμενο project, στην αποκωδικοποίηση συνελικτικού κώδικα. Ο τελευταίος θα χρησιμοποιηθεί για την διόρθωση σφαλμάτων (forward error correction - FEC) κατά την μετάδοση ψηφιακής πληροφορίας (bits) μέσα από κανάλι BSC (binary symmetric channel). Στόχος είναι να καταλάβετε σε βάθος μια απλή εφαρμογή FEC, καθώς αποτελεί συστατικό στοιχείο οποιασδήποτε δορυφορικής ζεύξης. Η παράδοση του σχετικού κώδικα και της αναφοράς θα γίνει μέσω courses. Επιτρέπεται η συνεργασία, αρκεί να αναγραφεί στην αναφορά. Ωστόσο, ο κώδικας και η αναφορά πρέπει να γίνουν ατομικά.

Εισαγωγή

Έστω bits πληροφορίας $m_1, m_2, \ldots, m_N$, $m_i \in \{0,1\}$, τα οποία αντιστοιχούν με κατάλληλη FEC κωδικοποίηση, σε μία νέα σειρά από bits $b_1, b_2, \ldots, b_{2N-1}$, $b_i \in \{0,1\}$. Συνεπώς, η κωδικοποίηση εισάγει επιπλέον bits και για κάθε N bits πληροφορίας απαιτούνται $2N-1$ bits, δηλ. ο *ρυθμός* της κωδικοποίησης είναι $\rho \triangleq N/(2N-1) \overset{\text{large N}}{\approx} 1/2$.

Μπορούν να υπάρξει FEC με διαφορετικό ρυθμό ρ, ωστόσο εδώ θα εστιάσουμε σε συνελικτικούς κώδικες (convolutional codes) με $\rho = 1/2$. Οι τελευταίοι χρησιμοποιήθηκαν στις διαστημικές αποστολές Voyager I (Άρης, Δίας και Κρόνος) και II (Ουρανός) [1], ενώ χρησιμοποιούνται ακόμη και σήμερα, π.χ. στον Αυστραλιανό μικροδορυφόρο (cubeSat) AU03 i-Inspire II (NORAD ID: 42731) [2], με τον οποίο θα ασχοληθούμε στο επόμενο project. Οι συνελικτικοί κώδικες χρησιμοποιούνται σε αλυσιδωτούς κώδικες (concatenation codes), όπου η κωδικοποιημένη λέξη της εξόδου ενός (εσωτερικού) κωδικοποιητή χρησιμοποιείται ως είσοδος σε νέο (εξωτερικό) κωδικοποιητή. Τέτοια δομή χρησιμοποιήθηκε στο Voyager II, με εξωτερικό κώδικα έναν Reed Solomon και εσωτερικό, έναν συνελικτικό.[1]

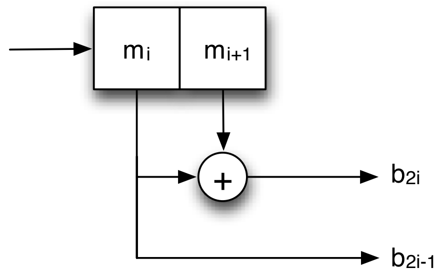
1 Κωδικοποίηση

Θα εστιάσουμε σε μία απλή μορφή συνελικτικού κώδικα με *μήκος εξαναγκασμού* (constraint length) L ίσο με $L = 2$ και ρυθμό 1/2, ο οποίος περιγράφεται από τις παρακάτω εξισώσεις:

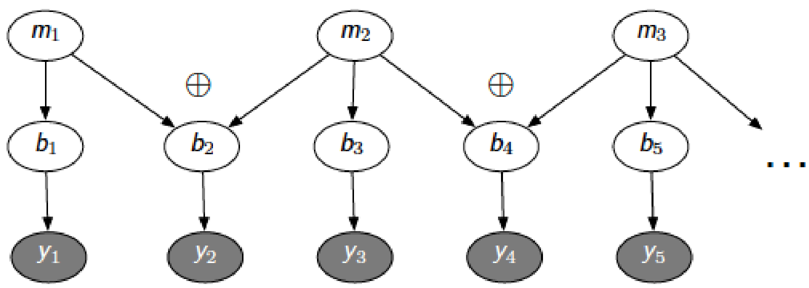
$$b_{2i-1} = m_i, \tag{1}$$
$$b_{2i} = m_i \oplus m_{i+1}, \tag{2}$$

όπου $\oplus$ συμβολίζει xor (δηλ. modulo 2 addition) και $i = 1, 2, \ldots, N$. Προσέξτε ότι η σειρά των N data bits μετατρέπεται σε μια σειρά $2N-1$ κωδικοποιημένων bits, της οποίας τα περιττά bits είναι αυτούσια τα bits πληροφορίας, ενώ τα άρτια προκύπτουν ως xor διαδοχικών data bits.

[1]Σημειώνεται πως οι αλυσιδωτοί κώδικες είναι η βάση των turbo κωδίκων, ενώ η δομή του Voyager 2 χρησιμοποιείται και στον AU03.



Σχήμα 1: Σχηματική αναπαράσταση συνελικτικής κωδικοποίησης μήκους εξαναγκασμού $L = 2$.



Σχήμα 2: Κωδικοποίηση με μία μορφή συνελικτικού κώδικα και στην συνέχεια μεταφορά μέσα από BSC κανάλι (Σχήμα από [3], [4]).

Σχηματικά, η κωδικοποίηση αυτή περιγράφεται στο Σχήμα 1, όπου φαίνονται οι $L = 2$ θέσεις των data bits . Για τις αποστολές Voyager χρησιμοποιήθηκε μήκος εξαναγκασμού $L = 7$.

2 Μεταφορά σε κανάλι BSC

Στην συνέχεια τα κωδικοποιημένα bits $b_1, b_2, \ldots, b_{2N-1}$ "περνούν" μέσα από ένα κανάλι BSC, το οποίο αλλάζει το κάθε bit (δηλ. το κάνει flip, μετατρέποντας το 1 σε 0 και το 0 σε 1), με πιθανότητα $\epsilon \in (0, 1/2)$ ή ισοδύναμα, δεν το μεταβάλλει, με πιθανότητα $1 - \epsilon$. Η έξοδος του καναλιού αυτού συμβολίζεται ως y_i, για είσοδο καναλιού το κωδικοποιημένο bit b_i, και $i = 1, 2, \ldots, 2N - 1$. Προφανώς, ισχύει:

$$\Pr(y_i = 1 | b_i = 0) = \Pr(y_i = 0 | b_i = 1) = \epsilon, \tag{3}$$

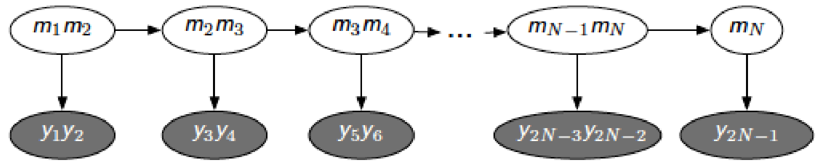
$$\Pr(y_i = 1 | b_i = 1) = \Pr(y_i = 0 | b_i = 0) = 1 - \epsilon, \tag{4}$$

και η συμμετρία στις παραπάνω πιθανότητες δικαιολογεί το όνομα του καναλιού (binary symmetric channel), για το οποίο γνωρίζουμε πως η χωρητικότητα δίνεται από την παρακάτω σχέση:

$$C_{\text{BSC}} = 1 - H(\epsilon), \tag{5}$$

όπου $H(\epsilon)$ η εντοπία μιας δυαδικής πηγής πληροφορίας: $H(\epsilon) = -\epsilon \log_2(\epsilon) - (1 - \epsilon) \log_2(1 - \epsilon)$.

Η διαδικασία κωδικοποίησης του συνελικτικού κώδικα και η μεταφορά μέσω του καναλιού BSC φαίνεται στο Σχήμα 2.



Σχήμα 3: Απλοποίηση της παραπάνω κωδικοποίησης και μεταφοράς σε ένα ισοδύναμο HMM (Σχήμα από [3], [4]).

Από θεωρία πιθανοτικών γραφικών μοντέλων (probabilistic graphical models), η διαδικασία αυτή μπορεί να απλοποιηθεί στο hidden Markov model (HMM) του Σχήματος 3. Παρατηρήστε στο Σχήμα 3 ότι διαδοχικές κρυμμένες καταστάσεις έχουν υποχρεωτικά κοινό bit.

Από την παρατήρηση των $y_1, y_2, \ldots, y_{2N-1}$, ο δέκτης πρέπει να εκτιμήσει την (κρυμμένη) ακολουθία $m_1, m_2, \ldots, m_N$.

3 Αποκωδικοποίηση

Η αποκωδικοποίηση έχει ως στόχο την επίλυση του παρακάτω προβλήματος μέγιστης πιθανοφάνειας (maximum likelihood - ML):

$$m_1^*, m_2^*, \ldots, m_N^* = \arg \max_{m_1, m_2, \ldots, m_N} \Pr\left(y_1, y_2, \ldots y_{2N-1} \middle| m_1, m_2, \ldots, m_N\right) \tag{6}$$

Με βάση τα χαρακτηριστικά του προβλήματος και το HMM του Σχήματος 3, αποδεικνύεται ότι:

$$\Pr\left(y_1, y_2, \ldots y_{2N-1} \middle| m_1, m_2, \ldots, m_N\right) \propto \Pr\left(y_{2N-1} \middle| m_N\right) \prod_{i=1}^{N-1} \Pr\left(y_{2i-1}, y_{2i} \middle| m_i, m_{i+1}\right), \tag{7}$$

και επομένως, το παραπάνω πρόβλημα γίνεται ισοδύναμο με:

$$m_1^*, m_2^*, \ldots, m_N^* = \arg \max_{m_1, m_2, \ldots, m_N} \log_2 \left[\Pr\left(y_{2N-1} \middle| m_N\right)\right] + \sum_{i=1}^{N-1} \log_2 \left[\Pr\left(y_{2i-1}, y_{2i} \middle| m_i, m_{i+1}\right)\right], \tag{8}$$

δηλ. ο εκτιμητής ML μετατρέπεται από max-product σε max-sum.

Σχετικά εύκολα (όπως δείξαμε στο μάθημα) μπορεί να αποδειχθούν τα παρακάτω:

$$i = N,\ \log_2 \left[\Pr\left(y_{2N-1} = u \middle| m_N = a\right)\right] \propto \mathbb{1}(u == a) \ln\left(\frac{1-\epsilon}{\epsilon}\right) \triangleq \log \phi_{m_N}(a)\Big|_{y_{2N-1}=u}, \tag{9}$$

$$i = 1, 2, \ldots, N-1,\ \log_2 \left[\Pr\left(y_{2i-1} = u, y_{2i} = v \middle| m_i = a, m_{i+1} = b\right)\right] \propto$$
$$\mathbb{1}(u == a) \ln\left(\frac{1-\epsilon}{\epsilon}\right) + \mathbb{1}(v == a \oplus b) \ln\left(\frac{1-\epsilon}{\epsilon}\right) \triangleq \log \phi_{m_{i,i+1}}(ab)\Big|_{y_{2i-1}=u, y_{2i}=v}, \tag{10}$$

όπου $a, b, u, v \in \{0,1\}$ και $\mathbb{1}(\cdot)$ είναι η indicator function, η οποία επιστρέφει 1 όταν το όρισμα είναι αληθές, και 0 διαφορετικά. Σημειώνεται πως $\frac{1-\epsilon}{\epsilon} > 1$ για $\epsilon \in (0, 1/2)$. Ορίζουμε το παρακάτω βάρος:

$$w_\epsilon = \ln\left(\frac{1-\epsilon}{\epsilon}\right) > 0, \epsilon \in (0, 1/2). \tag{11}$$

Για παράδειγμα, η Εξ. (10) για $i = 2$, $y_3 = 0, y_4 = 1, a = 0, b = 1$ δίνει:

$$\log \phi_{m_{2,3}}(ab) = \log \phi_{m_{2,3}}(01) = \mathbb{1}(y_3 == a)\ w_\epsilon + \mathbb{1}(y_4 == a \oplus b)\ w_\epsilon = 2w_\epsilon,$$

ενώ για $y_3 = 0, y_4 = 1, a = 0$ και $b = 0$, δίνει :

$$\log \phi_{m_{2,3}}(ab) = \log \phi_{m_{2,3}}(00) = \mathbb{1}(y_3 == a)\ w_\epsilon + \mathbb{1}(y_4 == a \oplus b)\ w_\epsilon = w_\epsilon.$$

Συνεπώς, οι Εξ. (9), (10) και οι αντίστοιχες συναρτήσεις $\log \phi$ "ζυγίζουν" τις κρυμμένες καταστάσεις και ορίζουν το αθροιστικό βάρος της κάθε πιθανής ακολουθίας $m_1, m_2, \ldots, m_N$, σύμφωνα με όλες τις παρατηρήσεις $y_1, y_2, \ldots, y_{2N-1}$.

Απλοποιώντας τον συμβολισμό από m_i, m_{i+1} σε $m_{i(i+1)}$, το Σχήμα 4 δείχνει παράδειγμα του διαγράμματος trellis για $N = 3$ και παρατηρήσεις $y_1 = 0, y_2 = 0, y_3 = 0, y_4 = 1, y_5 = 0, y_6 = 0, y_7 = 0$. Παρατηρήστε ότι οι μεταβάσεις μεταξύ συγκεκριμένων τιμών των κρυμμένων καταστάσεων εξασφαλίζουν ότι το κοινό bit μεταξύ διαδοχικών κρυμμένων καταστάσεων έχει την ίδια τιμή, π.χ. η κατάσταση $m_{12} = 10$ οδηγεί υποχρεωτικά μόνο στις καταστάσεις $m_{23} = 00$ ή $m_{23} = 01$, καθώς μόνο για αυτές τις τιμές της κατάστασης m_{23} το bit m_2 είναι ίσο με $m_2 = 0$.

Επίσης, παρατηρήστε ότι κάθε κρυμμένη κατάσταση, έχει συγκεκριμένο βάρος, που ορίζεται από την τιμή της κατάστασης ($m_{i(i+1)} = ab$), τις αντίστοιχες παρατηρήσεις ($y_{2i-1}y_{2i} = uv$) και τις εξισώσεις Εξ. (9), (10). Τα συγκεκριμένα παραδείγματα που υπολογίστηκαν παραπάνω αντιστοιχούν στην κρυμμένη κατάσταση m_{23}, όταν παίρνει τις τιμές $ab = 01$ ή $ab = 00$, αντίστοιχα.

Υλοποιώντας τον αλγόριθμο Viterbi στο παρακάτω διάγραμμα trellis, μπορεί να βρεθεί ότι το μονοπάτι-ακολουθία $m_1^* = 0, m_2^* = 0, m_3^* = 0, m_4^* = 0$ μεγιστοποιεί το αθροιστικό βάρος, σε σχέση με όλα τα υπόλοιπα μονοπάτια του διαγράμματος, πετυχαίνοντας συνολικό βάρος 6 w_ϵ.

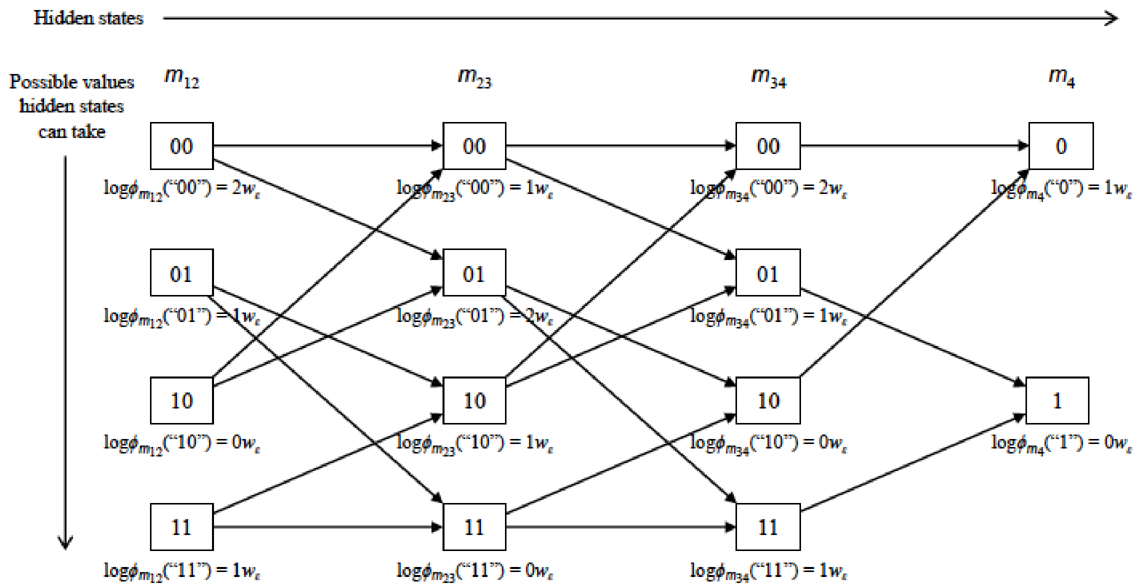
4 Ερωτήσεις

1. Υποθέστε $\epsilon = 1/5$ και $N = 128$. Προσομοιώστε μια ακολουθία $y_1, y_2, \ldots y_2 55$, και υλοποιήστε τον αλγόριθμο Viterbi. Σημειώστε τον αριθμό των bits που εκτιμήθηκαν λανθασμένα. Επαναλάβετε άλλες 10^4 φορές και εκτιμήστε το BER για το συγκεκριμένο ϵ.

2. Επανάλαβετε το παραπάνω για $\epsilon = 1/6, \epsilon = 1/8, \epsilon = 1/10, \epsilon = 1/20$ και σχεδιάστε το BER ως συνάρτηση του ϵ σε λογαριθμική κλίμακα.

3. Σχολιάστε συνοπτικά τα αποτελέσματα, λαμβάνοντας υπόψη την χωρητικότητα του BSC.

Παραδώστε τις παραπάνω απαντήσεις, καθώς και τον κώδικα (π.χ. αρχείο Matlab ή Java).



Σχήμα 4: Διάγραμμα trellis για $N = 4$ bits και $(y_1, y_2, y_3, y_4, y_5, y_6, y_7) = (0, 0, 0, 1, 0, 0, 0)$ (Σχήμα από [4]).

Βιβλιογραφία

[1] J. G. Proakis, M. Salehi (Μετ. Καρούμπαλος, Ζέρβας, Καραμπογιάς, Σαγκριώτης), Συστήματα Τηλεπικοινωνιών, Εκδόσεις ΕΚΠΑ, 2002.
[2] https://www.n2yo.com/satellite/?s=42731
[3] MIT 6.438 Algorithms for Inference, Fall 2014, lecture 9, available online at opencourseware.
[4] MIT 6.438 Algorithms for Inference, Fall 2014, lecture 11, available online at opencourseware.