# מבוא לתורת האינפורמציה

עפ"י תרגולים עם צור לוריא

סמסטר א', תש"ע

סיכומי התרגולים מתחילים מאמצע הסמסטר בלבד.

## 1 קודים ליניאריים וקודים מתקני שגיאות

### 1.1 הגדרות

**הגדרה 1.1** $\mathbb{F}_q$ הוא השדה עם $q$ איברים. יש שדה יחיד כזה לכל $q$ שהוא חזקה טבעית של מספר ראשוני.

**הגדרה 1.2** קוד ליניארי הוא תת מרחב וקטורי של $\mathbb{F}_q^n$. באופן שקול, קוד ליניארי הוא תמונה של העתקה ליניארית $C:\mathbb{F}_q^k\to\mathbb{F}_q^n$

**הגדרה 1.3** מרחק המינג, עבור $v_1,v_2\in\mathbb{F}_q^n$ מוגדר מרחק המינג להיות מספר הקואורדינטות שבהן $v_1\neq v_2$ ונסמנו ע"י $d\left(v_1,v_2\right)$.

**הגדרה 1.4** המרחק של קוד ליניארי מוגדר להיות המרחק המינימלי בין שתי מילים בקוד, כלומר $\min_{v_1,v_2\in C}d\left(v_1,v_2\right)$, ויסומן ע"י $d\left(C\right)$.

**הערה 1.5** המרחק של קוד ליניארי שווה למרחק המינימלי של מילה בקוד מהאפס $\min_{0\neq v_1\in C}d\left(v_1,0\right)$

### 1.2 פיענוח קוד ליניארי

המרחק בין כל שתי מילים הוא לכל הפחות $d\left(C\right)$, ולכן אם שלחנו מילה ואירעו פחות מ $\frac{d(C)}{2}$ טעויות, פונקציית הפיענוח תמצא את מילת הקוד הקרובה ביותר, ומשיקולי מרחק - יש רק מילה אחת שכזו. מכאן נובע שאם נבנה פונקציית פיענוח שמקבלת מילה ומוצאת את מילת הקוד הקרובה לה ביותר - אנו עשויים לטעות רק כאשר מספר הטעויות גדול מ $\frac{d(C)}{2}$.

### 1.3 חסם היחידון

**טענה 1.6** עבור קוד ליניארי $C:\mathbb{F}_q^k\to\mathbb{F}_q^n$ מתקיים:

$$d\left(C\right)\leq n-k+1$$

**הוכחה:** נתבונן ב $k-1$ הקואורדינטות הראשונות של מילות הקוד, להן יש $q^{k-1}$ אפשרויות, מאידך יש $q^k$ מילות קוד ולכן יש לפחות שתי מילות קוד שמסכימות על $k-1$ הקואורדינטות הראשונות, ולכן המרחק ביניהן הוא לכל היותר $n-(k-1)$, מדוע? יש $n$ קואורדינטות והן מסכימות על לפחות $k-1$ מתוכן. ולכן המרחק של הקוד הוא לכל היותר $n-k+1$. ■

### 1.4 קודי $Reed-Solomon$

#### 1.4.1 כמה עובדות על פולינומים

1. לפולינום מדרגה $k$ יש לכל היותר $k$ שורשים
2. פולינום $g\left(z\right)$ שמקבל אפס בנקודה $\alpha$, מתחלק בפולינום $\left(z-\alpha\right)$.

### 1.4.2 קידוד באמצעות קוד $Reed-Solomon$

יהי $\beta_1,...,\beta_q$ סידור כלשהו ל $\mathbb{F}_q$, נניח $n=q$, $k<n$.
נגדיר קוד $C:\mathbb{F}_q^k\to\mathbb{F}_q^n$ באופן הבא:
בהנתן $M=(m_0,...,m_{k-1})$ הודעה שיש לקודד, נגדיר פולינום

$$P_M(z)=m_0+m_1z+m_2z^2+...+m_{k-1}z^{k-1}$$

כעת:

$$C(M)=(P_M(\beta_1),P_M(\beta_2),...,P_M(\beta_n))$$

**תכונות**

1. קוד $RS$ הוא חח"ע: יהיו $M_1,M_2\in\mathbb{F}_q^k$ נניח $C(M_1)=C(M_2)$, וזה מתקיים אם"ם

$$C(M_1)-C(M_2)=\overrightarrow{0}\Leftrightarrow P_{M_1}(\beta_i)-P_{M_2}(\beta_i)\ \forall i$$

ומכאן נובע שלפולינום $P_{M_1}-P_{M_2}$ יש $n=q$ שורשים, והוא מדרגה לכל היותר $k-1<n$, ולכן מכך נובע שהוא פולינום האפס וכל מקדמיו הם אפס, ולכן $M_1,M_2$ מסכימות בכל קואורדינטה.

2. קוד $RS$ הוא קוד ליניארי: נבנה באופן מפורש את מטריצת ההעתקה:

$$C(M)=\begin{pmatrix}1&\beta_1&\beta_1^2&\cdots&\beta_1^{k-1}\\ \vdots&\ddots&&&\vdots\\ \vdots&&\ddots&&\vdots\\ &&&\ddots&\vdots\\ 1&\beta_n&\beta_n^2&\cdots&\beta_n^{k-1}\end{pmatrix}\begin{pmatrix}m_0\\ m_1\\ \vdots\\ m_{k-1}\end{pmatrix}=\begin{pmatrix}P_M(\beta_1)\\ P_M(\beta_2)\\ \vdots\\ \vdots\\ P_M(\beta_n)\end{pmatrix}$$

3. קוד $RS$ מממש את המרחק המקסימלי שאפשר להשיג בקוד ליניארי, כלומר $d(C)=n-k+1$:
נזכור שלכל פולינום מדרגה שהיא קטנה או שווה ל $k-1$ יש לכל היותר $k-1$ שורשים. לכל לכל $M\neq 0$, ב $C(M)$ יש לכל היותר $k-1$ אפסים. $\Leftarrow$ ב $C(M)$ יש לכל הפחות $n-(k-1)$ קואורדינטות שאינן אפס, ולכן $n-k+1\le d(C(M),0)$ ולכן $n-k+1\le d(C)$, מחסם היחידון ידוע לנו ש $d(C)\le n-k+1$ ולכן מתקיים שוויון ממש כנטען.

**דוגמא:** $\mathbb{F}_q=\mathbb{F}_3$ $\quad\mathbb{F}_3=\left\{\underbrace{0}_{\beta_1},\underbrace{1}_{\beta_2},\underbrace{2}_{\beta_3}\right\}$

$C:\mathbb{F}_3^2\to\mathbb{F}_3^3$
ואז:

$$C((1,2))=?$$

$$P_M(z)=1+2\cdot z$$

והקידוד יתקבל ע"י הצבת ה $\beta_i$ השונים בפולינום שהתקבל מההודעה:

$$\Rightarrow C((1,2))=(P_M(\beta_1),P_M(\beta_2),P_M(\beta_3))=(1,0,2)$$

### 1.4.3 פיענוח של קודי $RS$

כרגיל, בהנתן $y \in \mathbb{F}_q^n$ נחפש את מילת הקוד הקרובה ביותר, כלומר נחפש מילת קוד $w$ כך ש $d(w,y)$ מינימלי.
מספיק לחפש פולינום $f$ מדרגה לכל היותר $k-1$, כך שיתקיים $f(\beta_i) = y_i$ למספר $i$־ים מקסימלי.
כלומר - נותנים לנו $(y_1, ..., y_n)$ ואנחנו מחפשים $(w_1, ..., w_n) = (f(\beta_1), ..., f(\beta_n))$ כך ש $f(\beta_i) = y_i$ עבור מה שיותר $i$־ים.
באופן שקול, נחפש פולינום ב 2 משתנים מהצורה $p(y,z) = y - f(z)$ כך ש:

1. $f(z)$ פולינום מדרגה לכל היותר $k-1$
2. $0 = p(y_i, \beta_i)$ לכמה שיותר $i$־ים

נייצג את $y$ כסכום של מילת קוד $w$ ווקטור $e$ (וקטור טעויות).

**הגדרה 1.7** עבור וקטור טעויות $e$, נגדיר פולינום זיהוי טעויות להיות פולינום $\Lambda \neq 0$ כך ש $\Lambda(\beta_i) = 0$ בכל מקום בו $e_i \neq 0$.

**הערה 1.8** קל לראות שתמיד קיים כזה, למשל:

$$\Lambda_0(z) = \prod_{j:e_j \neq 0} (z - \beta_j)$$

וגם, בעצם, וקטור האפס.

אם $\Lambda(z)$ פולינום זיהוי טעויות עבור $e$, נתבונן בפולינום

$$q(y,z) = \Lambda(z) \cdot p(y,z) = \Lambda(z) \cdot (y - f(z))$$

פולינום זה מתאפס בכל הנקודות $(y_i, \beta_i)$, מדוע? אם $y = w + e$ כמקודם אז:

$$\Lambda(\beta_i)\left(w_i + e_i - \underbrace{f(\beta_i)}_{=w_i}\right) = \Lambda(\beta_i) \cdot e_i = 0$$

השוויון האחרון הוא מהגדרות פולינום זיהוי טעויות.
ולכן כנטען מתקיים

$$q(y_i, \beta_i) = 0$$

נגדיר: $g(z) = \Lambda(z) \cdot f(z)$ ונקבל מיידית כי

$$q(y,z) = \underbrace{\Lambda(z) \cdot (y - f(z))}_{by\,definition} = y \cdot \Lambda(z) - g(z)$$

אם מניחים שמספר הטעויות קטן או שווה ל $t$, אז קיים $\Lambda$ מדרגה קטנה או שווה ל $t$.
מכאן נובע ש

$$deg(g(z)) \leq deg(f(z)) + deg(\Lambda(z)) \leq k - 1 + t$$

ואז:

$$g(z) = a_0 + a_1 z + ... + a_{k+t-1} z^{k+t-1}$$

$$\Lambda(z) = b_0 + b_1 z + ... + b_t z^t$$

יש $k+2t+1$ משתנים.
ידוע

$$q\left(y_i,\beta_i\right)=0\Leftrightarrow\forall 1\leq i\leq n\ :\ y_i\Lambda\left(\beta_i\right)-g\left(\beta_i\right)=0$$

כלומר:

$$\underbrace{y_1b_0+y_1\beta_1b_1+y_1\beta_1^2b_2+...+y_1\beta_1^tb_t}_{y_1\Lambda(\beta_1)}-\underbrace{a_0-\beta_1a_1-\beta_1^2a_2-...-\beta_1^{k+t-1}a_{k+t-1}}_{g(\beta_1)}=0$$

וכן הלאה עוד $n-1$ משוואות לכל ה־$i$־ים.
$d\left(C\right)=n-k+1$, ו $2t+1=d\left(C\right)$ ולכן מספר המשתנים $k+d\left(C\right)=n+1$
נמצא $0\neq\Lambda,g$ שמקיימים את המשוואות ונחלק ע"מ למצוא את $f$ שיתקבל מ:

$$f\left(z\right)=\frac{g\left(z\right)}{\Lambda\left(z\right)}$$

# 2 אלגוריתם דינמי לפיענוח קודים מתקני שגיאות

החומר בתרגול זה מתוך ספרו של מקיי - $Information\,Theory,\ Inference,\ and\,Learning\,Algorithms$

## 2.1 אלג' דינמי לספירת מסלולים ב $DAG$ (גרף מכוון חסר מעגלים)

נתון $DAG$ עם קודקוד $s$ (מקור) וקודקוד $t$ (בור/מטרה), רוצים למנות את המסלולים מ $s$ ל $t$.

**הגדרה 2.1** $P\left(a,b\right)$ מספר המסלולים מ $a$ ל $b$.

דוגמא: מספר מסלולים בגרף סריג $(Grid)$, שבו אפשר מכל קודקוד ללכת לקודקוד שמימינו או מתחתיו.
אלגוריתם דינמי מתקבל כאשר פתרון לבעיה גדולה תלוי באופן רקורסיבי בפתרון תתי־בעיות.

**הערה 2.2** עבור $DAG$ מתקיים שלכל קודקוד $a$ בגרף מתקיים:

$$P\left(s,a\right)=\sum_{v:v\rightarrow a}P\left(s,v\right)$$

כלומר מספיק לסכום את כל המסלולים מ $s$ שמגיעים לקודקודים שמהם ניתן בצעד אחד להגיע ל $a$.

פתרון לדוגמא שהוצגה - נעבור על הקודקודים לפי מרחקם מ $s$, על $s$ נשים את המספר 1 , ולכל קודקוד אחר נסכום את המספר שרשמנו על הקודקוד שמעליו (אם יש) והקודקוד שמשמאלו (אם יש).
באופן דומה קל להגדיר את האלגוריתם הכללי, לעבור על הקודקודים לפי מרחקם מ $s$ ולסכום את המספרים שהתקבלו בצעד הקודם על קודקודים שמהם ניתן להגיע לקודקוד הנוכחי.

## 2.2 הערכת ההסתברות שמסלול עובר בקודקוד

נניח שמגרילים מסלול מ $s$ ל $t$ באקראי בהסתברות אחידה על כל המסלולים, מה ההסתברות שעוברים בקודקוד מסויים $v$?

**הערה 2.3** שקול למניית המסלולים העוברים ב $v$, כי אז נקבל את התשובה ע"י חלוקה במספר המסלולים הכולל.

לפי ההערה מספיק לחשב $\frac{P(s,v)\cdot P(v,t)}{P(s,t)}$ ואופן הביצוע של זה נתון לנו מאלגוריתם ספירת המסלולים הקודם.

## 2.3 מציאת מסלול בעל מחיר מינימלי

נניח שנתונה פונקציית מחיר על הצלעות:

$$C : E \to \mathbb{R}$$

נגדיר מחיר של מסלול להיות סכום המחירים של הצלעות במסלול.
שאלה: מהו המסלול מ $s$ ל $t$ בעל המחיר המינימלי.

**הגדרה 2.4** נסמן את המסלול המינימלי מ $a$ ל $b$ על ידי $M(a,b)$, ונסמן את מחירו $W(a,b)$.

**הערה 2.5** למינימליות של מסלול יש את אותה תכונה של "אופטימליות של תת בעיה" שאנחנו מחפשים באלגוריתמים דינמיים, כלומר אם המסלול מ $a$ ל $b$ מינימלי, אזי המסלול מ $a$ לכל קודקוד $v$ בדרך, הוא מינימלי גם כן. מדוע? אחרת ניתן היה לשפר על ידי בחירת מסלול אחר מ $a$ ל $v$.

לכן:

$$W(a,b) = \min_{v:v\to b} \left(W(a,v) + C(v,b)\right)$$

$$M(a,b) = \left(M\left(a, argmin_{v:v\to b}\left(W(a,v) + C(v,b)\right)\right), bx\right)$$

האלגוריתם למציאת המסלול בעל המשקל המינימלי (אלגוריתם ויטרבי) כרוך שוב במעבר על הקודקודים מ $s$ ואילך ומציאת המסלול המינימלי עד אליהם.

## 2.4 פיענוח כבעיה בהסתברות

נתון לנו קוד ליניארי, נשלחה מילת קוד $t^*$ בערוץ רועש, והתקבלה $y$. אנחנו יודעים את $y$ ורוצים למצוא את $t^*$.
בעבר עשינו שימוש במרחק המינג כדי למדוד את הטעויות שהתרחשו בערוץ, אולם במקרים מסויימים זהו מדד גס מדי. נניח שאנחנו שולחים ביט מעל ערוץ, כשההסתברות ש 0 יהפוך ל 1 קטנה מההסתברות ש 1 יהפוך ל 0, אזי כביכול מרחק המינג מחמיץ את הרובד הזה של המידע על הטעויות.

### 2.4.1 מה מילת הקוד $t$ שממקסמת את $Pr(t|y)$?

$$argmax_t \left(P(t|y)\right)$$

בעיה זו נקראת "בעיית $MAP$"

**הערה 2.6** לפי חוק $Baise$

$$P(t|y) = \frac{P(y|t) \cdot P(t)}{P(y)}$$

**הערה 2.7** $P(y|t)$ קל לחישוב. בערוץ תקשורת חסר זכרון,

$$P(y|t) = \prod_{i=1}^{n} Pr(y_i|t_i)$$

וההתפלגות על ה"אותיות" $y_i|t_k$ נתונה לנו בהגדרת הערוץ.

**הערה 2.8** מניחים הסתברות אחידה על מרחב ההודעות, ואז $Pr(t)$ הוא מספר קבוע. ואז מנוסחת בייס, $t$ הממקסם את $Pr(t|y)$ ממקסם גם את $Pr(y|t)$.

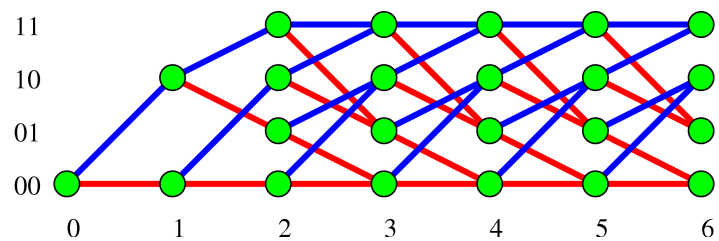
**הערה 2.9** מתקבל "אלגוריתם" למציאת $t$: נעבור על כל ההודעות $t$, ונבחר את $t$ שממקסם את $Pr(y|t)$ (שאותו אנחנו יודעים לחשב).

**בעיה**

ככלל - מספר מילות הקוד יהיה אקספוננציאלי באורך הקוד.

## 2.5 גרפי טרליס $(trellis)$ וקודים

**הגדרה 2.10** גרף טרליס הוא גרף $G = (V, E)$ שקודקודיו מסודרים בחתכים אנכיים שנקראים "זמנים". כל צלע מחברת בין 2 קודקודים בזמנים עוקבים, לכל צלע מתאימה "אות" (תיוג) שהיא 0 או 1. בנוסף - בזמן 0 ובזמן האחרון יש קודקוד יחיד.




לכל מסלול בגרף מ $s$ ל $t$ מתאימים מילת קוד לפי התיוגים של הצלעות במסלול. כל גרף מגדיר קוד שהוא אוסף מילות הקוד שמתקבלות כנ"ל.

**הערה 2.11** לכל טרליס מתאים קוד יחיד, אבל לקוד מסויים יכולים להיות גרפי־טרליס רבים.

**הגדרה 2.12** גרף טרליס יקרא טרליס ליניארי אם הקוד שהוא מגדיר הוא קוד לינארי.

**הגדרה 2.13** טרליס מינימלי של קוד הוא טרליס עם מספר קודקודים מינימלי, שמתאים לקוד.

### 2.5.1 תכונות של טרליס לינארי מינימלי, מעל שני ביטים (ללא הוכחה)

1. בטרליס מינימלי, יש לכל היותר 2 צלעות שיוצאות מכל קודקוד, וכנ"ל שתי צלעות נכנסות.
2. ב"זמן" נתון, לכל הקודקודים יש את אותה דרגת כניסה ודרגת יציאה.
3. מס' הקודקודים בזמן נתון הוא תמיד חזקה של 2.

## 2.6 שימוש באלגוריתם ויטרבי $(Viterbi)$ לפיענוח קודי טרליס לינארים

נניח שנתון לנו קוד לינארי בתור גרף טרליס מינימלי. נרצה למצוא את מילת הקוד $t$ שממקסמת את

$$Pr\left(y|t\right) = \prod_{i=1}^{n} Pr\left(y_i|t_i\right)$$

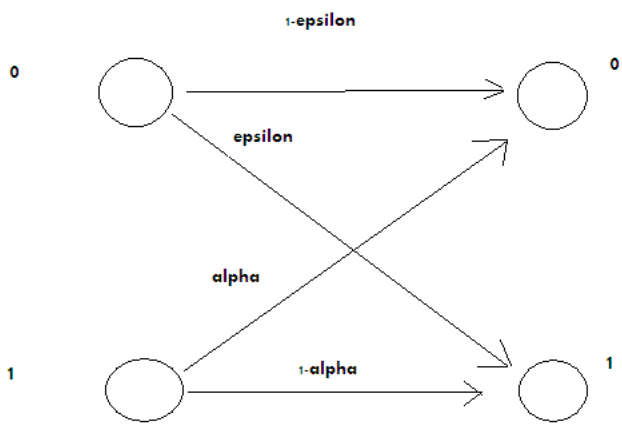
נמשקל את הצלעות כך שמסלול בעל מחיר מינימלי מתאים למילת קוד $t$ שממקסמת את $Pr\left(y|t\right)$.
רוצים למקסם את $\prod_{i=1}^{n} Pr\left(y_i|t_i\right)$, זה כמו למצוא מינימום ל $-\log\left(\prod_i Pr\left(y_i|t_i\right)\right)$ וזה בעצם:

$$-\sum_i \log\left(Pr\left(y_i|t_i\right)\right)$$

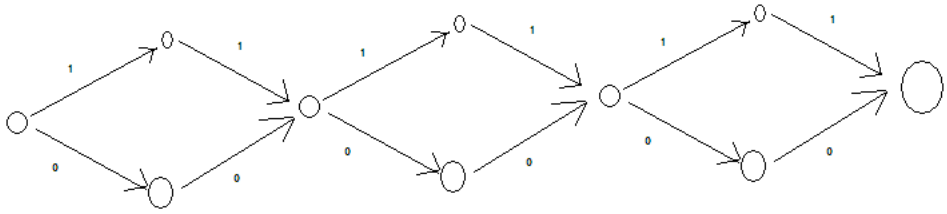
כעת ניתן לצלע בגרף טרליס את המחיר $-log\left(Pr\left(y_i|t_i\right)\right)$ כאשר $y_i$ הוא הביט ה $i$ ב $y$.
$t_i$ - התיוג של הצלע, הביט ה $i$ במילת הקוד שעוברת בצלע.
כעת אלגוריתם ויטרבי יתן לנו מסלול בעל מחיר מינימלי: מילת הקוד המתאימה תמקסם את $Pr\left(y|t\right)$.

### 2.6.1 דוגמא:

נניח שיש לנו ערוץ בינארי לא סימטרי:




עבור $\varepsilon = \frac{1}{8}$ ו $\alpha = \frac{1}{4}$ עם גרף טרליס בן 3 "עינים" כך שהמילים בו הן חזרות של פעמיים על אות, ובכל מילה יש 3 חזרות כאלו:



נחשב מה יהיו משקלי הצלעות לפי מינוס לוג של ההסתברויות, למשל:

$$\gamma_{00} = -\log Pr\left(y_1 = 0 | t_1 = 0\right) = -\log\left(\frac{7}{8}\right)$$

כך נחשב לכל הצלעות ואז נמצא מסלול מינימלי לפי ויטרבי וסיימנו.

# 3 קודי $LDPC$

## 3.1 הגדרת קודי $LDPC$

**הגדרה 3.1** קודי $LDPC$($Low\,Density\,Parity\,Check$): נניח שנתון לנו קוד ליניארי בתור מטריצה $n \times m$ שהיא $H$, כאשר:

$$C = \{x \in \mathbb{Z}_2^m | Hx = 0\}$$

(נזכיר שקוד ליניארי הוא תת מרחב וקטורי, וניתן להגדיר תת מרחב וקטורי כגרעין של העתקה ליניארית)
נסמן $Wr$ מספר ה 1ות בשורה $r$.
נסמן $Wc$ מספר ה 1ות בעמודה ה $c$.
נאמר ש $H$ דלילה אם $Wr << m$ ו $Wc << n$ לכל $r, c$.

**דוגמה**

$$(X_1, ..., X_m) \in \mathbb{Z}_2^m$$

1. $X_3 + X_7 = 0$
2. $X_1 + X_3 + X_{12} = 0$
3. ...

וכן הלאה עד המשוואה ה $n$.

**מסקנה 3.2** המימד של $C$ כמרחב וקטורי (בהנחה ששורות $H$ בת"ל) הוא $m - n$.

### 3.1.1 יצוג קודי $LDPC$ בתור גרף

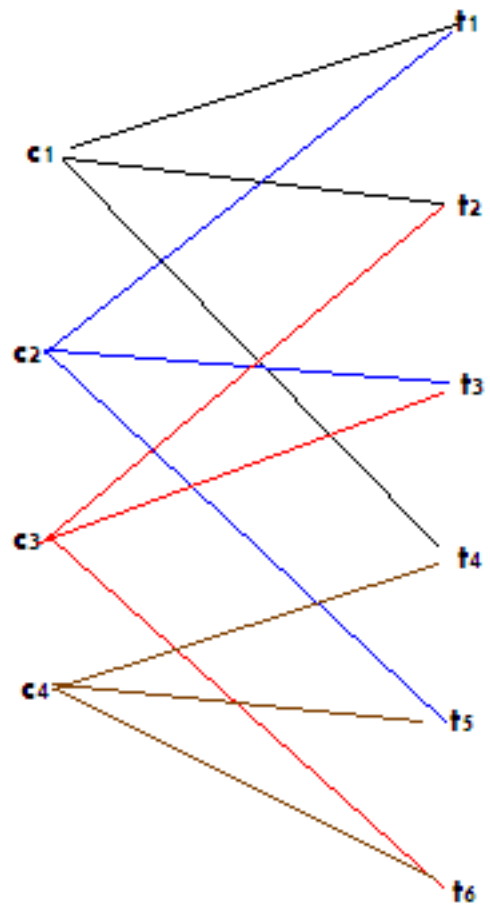
בהנתן קוד $LDPC$ עם מטריצה $H_{n\times m}$ נגדיר גרף ע"י:

$$V = \{c_1, ..., c_n, t_1, ..., t_m\}$$

$$E = \{\{t_i, c_j\} | H(j, i) = 1\}$$

**דוגמה**

$$H = \begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 \end{pmatrix}$$




**הערה 3.3** זהו גרף דו צדדי.

- יש קודקוד עבור כל אילוץ לינארי $\leftrightarrow$שורה של $H$.
- יש קודקוד עבור כל קואורדינטה $X_i$ כך ש $1 \leq i \leq m$.
- יש צלע בין $t_i$ ל $c_j$ כשהמשתנה $X_i$ מופיע באילוץ ה $j$.
- מס' הצלעות בגרף = מספר ה1ות ב $H$.
- גרף זה יקרא הגרף טנר $(Tanner)$של הקוד.

**הגדרה 3.4** קוד $LDPC$ יקרא רגולרי אם למטריצה $H$ יש את אותו המספר של 1ות בכל עמודה ואותו מספר של 1ות בכל שורה.
(בדוגמה שניתנה לעיל - זהו קוד רגולרי)

### 3.1.2 בניית קודי $LDPC$ ותכונותיהם

1. יש קודי $LDPC$ שהם קודים "טובים" במובן של משפט קיבולת הערוץ (כלומר ניתן להתקרב כרצוננו לקיבולת ע"י קודי $LDPC$).

2. קודי $LDPC$ שנדגמים באקראיות הם טובים בהסתברות גבוהה.

תזכורת: משפט קיבולת הערוץ אומר - נניח שיש ערוץ עם קיבולת $C$, אז יש סדרה של קודים $(M,n)$ שהקצב שלהם שואף ל $C$ וההסתברות המקסימלית לטעות שואפת לאפס כאשר $n \to \infty$.

## 3.2 פיענוח קודי $LDPC$

היינו רוצים למצוא את מילת הקוד $t$ שממקסמת את $Pr(t|y)$ כאשר $y$ הוא פלט של ערוץ רועש.
אם מניחים התפלגות אחידה על מילות הקוד אז שקול לחשב את $t$ שממקסמת $Pr(y|t)$. באופן כללי בעיית הפיענוח היא $NP$-קשה. (בשיעור שעבר פתרנו בעיה זו עבור קודים עם ייצוג כגרף טרליס קטן), מה היתה ה"רמאות"? שהאלגוריתם היה פולינומי בגודל הייצוג ע"י גרף טרליס, אבל זה עשוי להיות מעריכי בגודל הקוד. נשתמש הפעם באלגוריתם קירוב, שהוא למעשה יוריסטיקה, והוא דומה לאלגוריתם $Viterbi$.

### 3.2.1 פיענוח "קשה"

נסביר בעזרת דוגמא:
נניח שעבור הקוד בדוגמא הקודמת, נשלחה מילת הקוד $t = 110011$.
נניח שהמילה נשלחה בערוץ בינארי סימטרי, והביט השני התהפך כך שהתקבלה המילה $y = 100011$.

**האלגוריתם**

1. בשלב הראשון קודקודי $t_i$ מאתחלים את הניחוש שלהם להיות $y_i$. כל $t_i$ שולח את הניחוש שלו לשכניו (ה $c_j$ים).
2. כל קודקוד $c_j$ מחשב הודעה עבור כל שכן שלו מסוג $t_i$. ההודעה מ $c_j$ ל $t_i$ תהיה הערך ש $t_i$ צריך לקבל בהנחה ששאר השכנים של $c_j$ שלחו ניחוש נכון כדי שהמשוואה ה $j$ תסתפק. כלומר, אם קודקוד $c$ קיבל משניים משכניו 1 ו 1 ומקודקוד $t$ את 1, ההודעה מ $c$ ל $t$ תהיה 0. במילים אחרות - ההודעה מ $c_j$ ל $t_i$ היא $XOR$ של השכנים האחרים של $c_j$.
3. קודקודי $t$ מקבלים את ההודעות מכל ה $c_j$ים שלהם ומעדכנים את הניחוש שלהם לפי שיטת הרוב (קול אחד לעצמו, וקול אחד לכל אחד מה $c_j$ים שהם שכניו), שולחים את ניחושיהם המעודכנים ל $c_j$ים.
4. חוזרים לשלב 2.

בכל איטרציה (חיזרור) של האלגוריתם, בודקים אם כל המשוואות מסתפקות, אם כן - עוצרים.

### 3.2.2 פיענוח "רך"

**סימונים:** $p_i = Pr(t_i = 1|y_i)$ (נוכל לחשב זאת בקלות אם נניח התפלגות אחידה על מילות הקוד).
$q_{i,j}$ - ההודעה שקודקוד $t_i$ שולח לקודקוד $c_j$.
$q_{i,j}$ הוא וקטור (בניגוד להיותו ביט יחיד באלגוריתם הקודם) כך ש:
$q_{i,j}(0)$= ההערכה להסתברות ש $t_i$ הוא 0
$q_{i,j}(1)$= ההערכה להסתברות ש $t_i$ הוא 1
$r_{j,i}$ - ההודעה של $c_j$ ל $t_i$, וזה גם כן וקטור כך ש:
$r_{j,i}(0)$= ההערכה של $c_j$ להסתברות ש $t_i$ הוא 0.
$r_{j,i}(1)$= ההערכה של $c_j$ להסתברות ש $t_i$ הוא 1.
$C_i$= קבוצת השכנים של $t_i$.
$T_j$ = קבוצת השכנים של $c_j$.

**האלגוריתם ($Beliefs\ Propegation$)**

1. מאתחלים לכל $t_i$ את $q_{i,j}(0) = 1 - p_i$ ו $q_{i,j}(1) = p_i$, לכל $j$ כך ש $c_j$ שכן של $t_i$.
2. הקודקודים $c_j$ מחשבים הודעות כאשר:
$r_{j,i}(0)$ = ($t_i$ הוא 0 בהנתן שהערכות ההסתברות של שאר השכנים של $c_j$ נכונות)$Pr$
= (מספר האחדות בשאר השכנים של $c_j$ הוא זוגי)$Pr$
$r_{j,i}(1) = 1 - r_{j,i}(0)$

3. הקודקודים מסוג $t_i$ מחשבים את ההודעות שלהם:

$$\hat{Q}_i(0) = (1-p_i)\prod_{j\in C_i} r_{j,i}(0)$$
$$\hat{Q}_i(1) = p_i\prod_{j\in C_i} r_{j,i}(1)$$

ומנרמלים כך ש $\hat{Q}_i(0)+\hat{Q}_i(1)=1$ כדי שזו תהיה הסתברות. נגדיר:

$$\hat{t_i} = \begin{cases} 1 & \hat{Q}_i(1) > \hat{Q}_i(0) \\ 0 & otherwise \end{cases}$$

נבדוק האם המשוואות מסופקות. אם כן - נעצור. אם לא:

$$q_{i,j}(0) = (1-p_i)\prod_{j'\in C_i\setminus j} r_{j,i}(0)$$
$$q_{i,j}(1) = p_i\prod_{j'\in C_i\setminus j} r_{j,i}(1)$$

ומנרמלים כך ש $q_{i,j}(0)+q_{i,j}(1)=1$.

4. חוזרים לשלב 2.

גם כאן - אחרי מספר כלשהו של איטרציות אפשר לוותר אם זה לא מצליח. למרות שאין כל הבטחה שזה יתכנס, בפועל - במרבית הפעמים זה מתכנס ולכן הקוד הזה שימושי מאד.

# 4 כיווץ תמונות

## 4.1 המשימה

כיווץ של תמונות מהעולם האמיתי (מצלמה, לא תמונות של טקסט או ציורי קוים)

### 4.1.1 מאפיינים:

1. גרדיאנטים חלקים ככלל (כלומר מעברי הצבע הם בדרך כלל לא חדים, אלא חלקים יחסית)
2. יש דמיון בתוך איזורים בתמונה
3. העין האנושית רגישה יותר לשינויים מסויימים בתמונה (ולכן נוכל לוותר על דברים מסויימים בתמונה)
4. אין לנו מודל הסתברותי שמתאר במדוייק את התופעה.

## 4.2 שיטות $lossless$ לכיווץ תמונות

### 4.2.1 דוגמאות

1. $gif$ - מסדרים את הפיקסלים בתמונה בסדרה, ומכווצים את הסדרה עם למפל זיו.
2. $png$ - קודם כל עוברים על כל הפיקסלים ולכל פיקסל מנחשים מה יהיה צבעו (למשל - על ידי ממוצע השכנים), ושומרים את ההפרש בין הניחוש לבין הערך האמיתי.

   (א) ככלל, $png$ משפר את הכיווץ של $gif$ ב $\sim 30\%$

   (ב) יש תמונות שעליהן $gif$ מבצע יותר טוב

## 4.3 $JPEG$

שיטת כיווץ $lossy$.

באופן טיפוסי, מכווצים תמונות פי 20־10 (לעומת פי 2 או 3 של $PNG$) בלי שינוי נראה לעין באיכות התמונה.

זוהי שיטה יוריסטית שמשתמשת במספר טכניקות שונות.

### 4.3.1 אלגוריתם $JPEG$

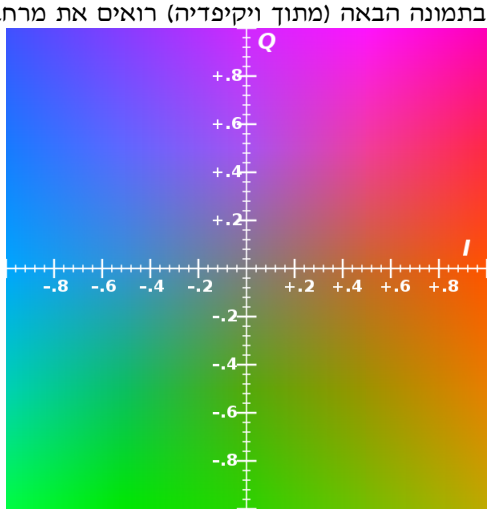
מעבירים את התמונות למרחב $(Y, I, Q)$:

$Y$ - הבהירות

$I, Q$ - ערוצי הצבעים

מניסויים רואים שהעין האנושית רגישה יותר לשינויי בהירות מאשר לשינויי צבע, ולכן ניתן לכווץ יותר (וכך לפגוע יותר) בפרמטרים של צבעוניות.

בתמונה הבאה (מתוך ויקיפדיה) רואים את מרחב הצבעים של $I$ ו $Q$.




מעתה והלאה, נדבר על ערוץ יחיד

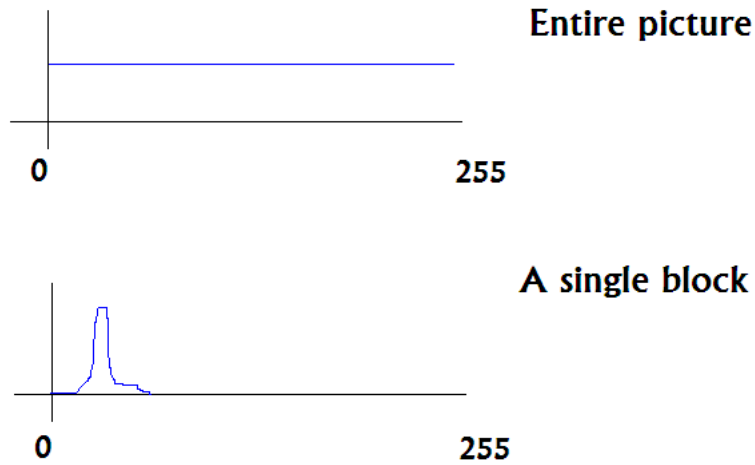
1. מחלקים את התמונה לבלוקים בגודל $8 \times 8$. מעתה והלאה מתמקדים בכיווץ בלוק יחיד.
2. על כל בלוק מפעילים טרנספורם פורייה $Discrete\,Cosine\,Transform$
3. קוונטיזציה - "מעגלים" את המקדמים (זה השלב היחיד שבו מאבדים מידע)
4. מקודדים את המטריצה משלב 3 בעזרת שיטות כיווץ $lossless$ (כמו הופמן, קידוד אריתמטי)

## 4.4 פירוט האלגוריתם

### 4.4.1 בלוקים - עבודה לוקאלית

למה לעבוד על בלוקים קטנים?

דוגמא: נניח ששיטת הכיווץ שלנו תהיה לראות את התמונה כסדרה של גווני אפור בלתי תלויים, ולכווץ אותה בעזרת קידוד הופמן. יש הגיון בחלוקת התמונה לבלוקים קטנים וטיפול בכל בלוק בנפרד כי בעולם של תמונות מצולמות סביר להניח שברוב הבלוקים ההתפלגות של גווני הפיקסלים תהיה מרוכזת. למשל:




**הערה/סכנה: עלולים להיווצר ארטיפקטים (תוצרי לוואי של דחיסה שלא היו בתמונה המקורית) בגבולות בין הבלוקים.**

#### 4.4.2 טרנספורם פורייה

ה $DCT$ בכמה מילים

ניתן לראות תמונה $8 \times 8$ כאיבר במרחב הוקטורי של מטריצות $8 \times 8$ מעל $\mathbb{R}$.

תמונה היא למעשה מטריצת מקדמים לייצוג של וקטור כצירוף לינארי של איברי הבסיס, $e_{ij}$=מטריצה של אפסים שבה בתא $i, j$ יש 1.

אם $T$ היא התמונה:

$$T = \begin{bmatrix} t_{11} & & t_{18} \\ & & \\ t_{81} & & t_{88} \end{bmatrix}$$

$$T = t_{11}e_{11} + t_{12}e_{12} + ... + t_{88}e_{88}$$

נגדיר בסיס אחר, ע"י $B_{u,v}$=מטריצה $8 \times 8$ ואוסף האיברים $\{B_{u,v}\}_{u,v=1}^{8}$ כך ש:

$$B_{u,v}(i,j) = \cos\left(\frac{\pi}{8}\left(i + \frac{1}{2}\right)u\right) + \cos\left(\frac{\pi}{8}\left(j + \frac{1}{2}\right)v\right)$$

עובדה שלא נוכיח - לכל $u, v \neq 0, 0$, סכום איברי $B_{u,v}$ שווה לאפס.

תכונות:

1. $B_{0,0} = \vec{1}$. אם $a_{0,0}$ הוא המקדם של $B_{0,0}$ אחרי הטרנספורם, אז $a_{0,0}$=הגוון הממוצע של פיקסל בתמונה.

הוכחה: המטריצה לפני הטרנספורם ואחרי הטרנספורם היא אותו וקטור (במרחב הוקטורי של המטריצות שבנינו) כמקודם ולכן:

$$\sum_{u,v} t_{u,v}e_{uv} = \sum_{u,v} a_{u,v}B_{u,v}$$

נסכום את האיברים:

$$\underbrace{\sum_{i,j}\sum_{u,v}\left(t_{u,v}e_{u,v}\right)(i,j)}_{=\sum_{u,v}t_{u,v}=1} = \underbrace{\sum_{i,j}\sum_{u,v}a_{u,v}B_{u,v}(i,j)}_{\underbrace{=a_{0,0}\sum_{i,j}B_{0,0}(i,j)+\sum_{u,v\neq 0,0}a_{u,v}\underbrace{\sum B_{u,v}(i,j)}_{=0}}_{=64a_{00}+0}}$$

ולכן

$$a_{00}=\frac{\sum_{u,v}t_{u,v}}{64}$$

2. עבור $u,v$ גדולים יש במטריצה תנודות מהירות. עבור $u,v$ קטנים יש מעברים חלקים.

**מדוע טוב לעשות $DCT$?**

1. מסתבר שהעין רגישה יותר לשינויים בתדרים הנמוכים מאשר לשינויים בתדרים הגבוהים.
2. בתמונה אופיינית, המקדמים של התדרים הגבוהים יהיו קטנים, לכן לאחר שלב הקוונטיזציה יהיו רובם יעברו לאפס, וזה ישפר את הכיווץ בשלב הבא.
3. כשמטפלים בכל בלוק בנפרד, יכולים להיווצר ארטיפקטים בגבולות. כשעושים $DCT$, התופעות הללו הן פחות נפוצות.

### 4.4.3 שלב הקוונטיזציה

מחזיקים מטריצת קוונטיזציה $Q_{8\times 8}$, דוגמא:

$$Q=\begin{bmatrix} 10 & 12 & & 57 \\ 13 & & & \\ & & & \\ 50 & & & 75 \end{bmatrix}$$

ככלל - $Q_{i,j}$ גדול יותר כש $i+j$ גדול יותר.
יש לנו מטריצת מקדמים $A$, נחלק כל איבר ב $A$ באיבר המתאים לו ב $Q$, ונעגל למספר השלם הקרוב ביותר.
למשל עבור:

$$A=\begin{bmatrix} 800 & 120 & 31 & & 1 \\ & & & & \\ & & & & \\ & & & 12 & 15 \end{bmatrix} \rightarrow \begin{bmatrix} 80 & 10 & 2 & & \\ & & & & \\ & & & & \\ & & & 0 & 0 \end{bmatrix}$$

כל התדרים שעברו לאפס למעשה הלכו לאיבוד, כי בתהליך הפיענוח נכפול כל איבר במטריצה באיבר המתאים לו ב $Q$ - ואם הוא 0 - נקבל 0 בכל מקרה.

**הערה 4.1** רוב המקדמים של התדרים הגבוהים - יעברו לאפס.

### 4.4.4 קידוד

בשלב זה מסתכלים במטריצה $B$ כוקטור, כשמסדרים את איברי המטריצה במין זיגזג (כלומר $a_{1,1},a_{1,2},a_{2,1},a_{3,1},a_{2,2},a_{1,3}...$ וכך הלאה). זורקים את סדרת האפסים שבסוף הוקטור, ואת שאר הוקטור מכווצים בעזרת קידוד $lossless$ (השיטות הנפוצות הן הופמן ואריתמטי).