Contents

מבוא למערכות לומדות - תרגיל 1

ת.ז: 305612038

23 במרץ 2017

1 שאלה 1

1.1 סעיף א

נראה שקיימת סביבה E כך שמתקיים $\lfloor log\left(N\right)\rfloor = \lfloor log\left(|H|\right)\rfloor \geq M_A\left(A\right)$:
ניצור סביבה בגודל $\lfloor log\left(N\right)\rfloor$ כך שתתקבל שגיאה בכל דגימה בסביבה.
נגדיר סדרת קטעים שמאליים $\{a_1,...,a_n\}$ וכן סדרת קטעים ימניים $\{b_1,...,b_n\}$ באופן רקורסיבי. בהתחלה נגדיר $a_1 = 0,\ b_1 = N$. כעת בכל איטרציה נגדיר $x_i = \frac{a_i+b_i}{2}$ וכן נתבונן בהיפותיזה $\hat{h}$.
אם $\hat{y}_i = 1$ כאשר $h(x_i) = 1$ אזי נגדיר $y_i = 1$ ונעדכן $a_{i+1} = x_i$, $b_{i+1} = b_i$.
אם $\hat{y}_i = -1$ נגדיר $y_i = 1$ ונעדכן הפוך, כלומר $b_{i+1} = x_i$ וכן $a_{i+1} = a_i$.
נשים לב שע"פ הגדרה זו קיימת שגיאה לכל $instance$ ועל כן $\lfloor log\left(|H|\right)\rfloor = \lfloor log\left(N\right)\rfloor \geq M_A(E)$.
בנוסף, נשים לב שסביבה זו ברת מימוש שכן:

$$\{h_\theta|\ \theta \in [a_i, b_i]\}$$

$$\Downarrow$$

$$\{h \in V_t \mid h(x_i) = 1\} \neq \emptyset \ \wedge \ \{h \in V_t \mid h(x_i) = -1\} \neq \emptyset$$

כנדרש.

1.2 סעיף ב

נחליף את הסט X עם הסט $[0,1]$ כלומר $\Theta \in [0,1] \cup \left\{-\frac{1}{2}, \frac{3}{2}\right\}$, ונראה כי לכל אלגוריתם למידה A ולכל מספר סיבובים n מתקיים $M_A(H) = n$:
בדומה לסעיף א, נאתחל את $a_1 = 0$, $b_1 = 1$ ונעשה בדיוק כמו סעיף קודם. נשים לב כי לכל n נוכל לקבל סדרה $\{x\}_{i=1}^n$ שכן מצפיפות הממשיים תמיד קיים $x_j = \frac{a_j+b_j}{2}$ לכל $1 \leq j \leq n$, ולכן לכל n שרירותי נוכל לייצר סדרה בגודל n.
מכאן שעבור כל אלגוריתם לומד A מתקבל $M(A) = \infty$ כנדרש.

2 שאלה 2 - אלגברה לינארית

2.1 סעיף א

תהי $A \in \mathbb{R}^{m\times d}$ עם SVD $A = U\Sigma V^T$. נסמן v_i, u_i להיות העמודות ה-i של V,U בהתאמה, וכן $\sigma_i = \Sigma_{ii}$. נראה כי $Av_i = \sigma_i u_i$ וכן $A^T u_i = \sigma_i v_i$.
נזכור כי Σ היא מהצורה:

$$\Sigma = \begin{pmatrix} \sigma_1 & 0 & \cdots & 0 \\ 0 & \sigma_2 & & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \cdots & \ddots \end{pmatrix}$$

וכן כי המטריצות V,U הם אורתונורמליות, קרי עבור $i \neq j$ מתקיים $\langle u_i|u_j\rangle = \langle v_i|v_j\rangle = 0$ ועבור $i = j$ מתקיים $\langle u_i|u_j\rangle = \langle v_i|v_j\rangle = 1$.

ואז:

$$Av_i = U\Sigma V^T v_i = U\Sigma\cdot \begin{pmatrix} v_{11} & \cdots & \cdots & v_{1d} \\ \vdots & v_{22} & & \vdots \\ \vdots & & \ddots & \vdots \\ v_{d1} & \cdots & \cdots & v_{dd} \end{pmatrix} \cdot \begin{pmatrix} v_{1i} \\ v_{2i} \\ \vdots \\ v_{di} \end{pmatrix} = U\Sigma\cdot \begin{pmatrix} 0 \\ \vdots \\ 0 \\ \langle v_i | v_i \rangle \\ 0 \\ \vdots \\ 0 \end{pmatrix}_{d\times 1} = U\Sigma_{m\times d}\cdot \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}_{d\times 1} = U_{m\times m}\cdot \begin{pmatrix} 0 \\ \vdots \\ 0 \\ \sigma_i \\ 0 \\ \vdots \\ 0 \end{pmatrix}_{m\times 1} =$$

$$= U_{1i}\cdot \sigma_i + U_{2i}\cdot\sigma_i + ... + U_{mi}\cdot\sigma_i = \sigma_i \cdot u_i \blacksquare$$

נזכור כי עבור מטריצות A, B כלשהם מתקיים $(AB)^T = B^T A^T$, ואז באופן דומה:

$$A^T u_i = \left(U\Sigma V^T\right)^T u_i = \left(\left(V^T\right)^T (U\Sigma)^T\right) u_i = V\left(\Sigma^T U^T\right) u_i = V\Sigma^T \begin{pmatrix} u_{11} & \cdots & \cdots & u_{1m} \\ \vdots & u_{22} & & \vdots \\ \vdots & & \ddots & \vdots \\ u_{m1} & \cdots & \cdots & u_{mm} \end{pmatrix} \cdot \begin{pmatrix} u_{1i} \\ u_{2i} \\ \vdots \\ u_{mi} \end{pmatrix} =$$

$$= V\Sigma^T \begin{pmatrix} 0 \\ \vdots \\ 0 \\ \langle u_i | u_i \rangle \\ 0 \\ \vdots \\ 0 \end{pmatrix} = V\left(\Sigma^T\right)_{m\times d} \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} = V\Sigma_{d\times m} \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}_{m\times 1} = V \begin{pmatrix} 0 \\ \vdots \\ 0 \\ \sigma_i \\ 0 \\ \vdots \\ 0 \end{pmatrix}_{d\times 1} =$$

$$= V_{1i}\sigma_i + V_{2i}\sigma_i + ... + V_{di}\sigma_i = \sigma_i \cdot v_i \blacksquare$$

2.2 סעיף ב

הסבר: מאחר והמטריצות U, V הם אורתונורמליות אז מתקיים עבור $i \neq j$ $\langle u_i | u_j \rangle = \langle v_i | v_j \rangle = 0$ ועבור $i = j$ מתקיים $\langle u_i | u_j \rangle = \langle v_i | v_j \rangle = 1$.

אבל נשים לב כי למעשה מלינאריות במשתנים מתקיימת השקילות $\langle -v_i | -v_j \rangle = -1\cdot -1 \langle v_i | v_j \rangle = \langle v_i | v_j \rangle = 1$ לכל $i = j$.

לכן, ניתן להניח כי $\sigma_i \geq 0$ שכן אחרת ניתן לבצע את ההחלפה שלו לאי-שלילי ולקבל אותו ביטוי.

2.3 סעיף ג

תהי $A \in \mathbb{R}^{m\times d}$.

2.3.1 סעיף ב

נראה כי אם $A = U\Sigma V^T$ SVD של A אזי $AA^T = U\Sigma\Sigma^T U^T$ הינה EVD של AA^T, וכן $A^T A = V\Sigma^T\Sigma V^T$ הינה EVD של $A^T A$.

נזכור כי V אורתונורמלית ולכן $VV^T = I$ ואז:

$$AA^T = \left(U\Sigma V^T\right)\left(U\Sigma V^T\right)^T = \left(U\Sigma V^T\right)\left(\left(\Sigma V^T\right)^T U^T\right) = \left(U\Sigma V^T\right)\left(V\Sigma^T U^T\right) = U\Sigma I\Sigma^T U^T = U\Sigma\Sigma^T U^T$$

Σ היא מטריצה אלכסונית מסדר $m\times d$, לכן Σ^T היא גם אלכסונית עם אותם ערכים באלכסון רק מסדר $d\times m$, לכן $\Sigma\Sigma^T$ היא מטריצה מסדר $m\times m$ אלכסונית וסימטרית כאשר ערכי האלכסון הם $\lambda_{ii} = \langle\sigma_i|\sigma_i\rangle$.

בנוסף, U אורתונורמלית ולכן בסה"כ הפירוק הנ"ל הוא EVD של AA^T.

באופן דומה U אורתונורמלית ולכן $UU^T = I$ ואז:

$$A^TA = \left(V\Sigma^TU^T\right)\left(U\Sigma V^T\right) = V\Sigma^TI\Sigma V^T = V\Sigma^T\Sigma V^T$$

כמו מקודם, Σ היא מטריצה אלכסונית מסדר $m \times d$, לכן Σ^T היא גם אלכסונית עם אותם ערכים באלכסון רק מסדר $d \times m$, לכן $\Sigma^T\Sigma$ היא מטריצה מסדר $d \times d$ אלכסונית וסימטרית כאשר ערכי האלכסון הם $\lambda_{ii} = \langle\sigma_i|\sigma_i\rangle$.
בנוסף, V אורתונורמלית ולכן בסה"כ הפירוק הנ"ל הוא EVD של A^TA.
כנדרש. ■

2.3.2 סעיף ג

נסמן $k = rank(A) \leq min(m, d)$, ונסמן $\bar{v}_i$ להיות העמודה ה-i ב-V.
נתון כי $A^TA = VDV^T$ היא EVD של A^TA לכן, לכל V אורתוגונולי קרי $\langle v_i|v_j\rangle = \delta_{i,j}$. ואז נגדיר וקטורים $\bar{w}_i = A\bar{v}_i$ לכל $1 \leq i \leq d$. ונתבונן בסדרת הוקטורים $\{\bar{w}_i\}_{i=1}^d$:
אם $k = d$ אזי ישנם d וקטורים אותוגונליים, ואז נוכל להרחיבם לבסיס על ידי כך שניקח $m - d$ וקטורים בת"ל לd הוקטורים האורתוגונליים. כעת ישנם m וקטורים בת"ל, לכן נותר רק לנרמל כל אחד מהוקטורים הנ"ל וקיבלנו בסיס אורתונורמלי
$\{\tilde{w}_1, ..., \tilde{w}_m\}$ שהם ההרחבה המנורמלת של $\{\bar{w}_i\}_{i=1}^d$. אם $k = m$ ההוכחה היא דומה.

3 שאלה 3

תהי Vתת-מרחב ממימד k של $\mathbb{R}^d$, ויהיו $v_1, ..., v_k \in \mathbb{R}^d$ בסיס אורתונורמלי ל-V. נגדיר $P = \sum_{i=1}^{k} v_iv_i^T$ מטריצת ההטלה על V.

3.1 סעיף א

נראה כי P סימטרית:
$v_1, ..., v_k$ בסיס אורתונורמלי לכן:

$$P = \sum_{i=1}^{k} v_iv_i^T = \sum_{i=1}^{k} \left(v_i^T\right)^T v_i^T = \left(\sum_{i=1}^{k} \left(v_iv_i^T\right)\right)^T = P^T$$

3.2 סעיף ב

נראה כי $v_1, ..., v_k$ וקטורים עצמיים של P וכולם עם הערך העצמי 1, נבחר v_j וקטור בסיס אורתונורמלי כלשהו ואז:

$$Pv_j = \left(\sum_{i=1}^{k} v_iv_i^T\right) v_j = \sum_{i=1}^{k} v_iv_i^Tv_j = v_jv_j^Tv_j = v_j\left(v_j^Tv_j\right) = v_j \cdot (1) = v_j$$

והראינו כי לכל v_i וקטור בסיס אורתונורמלי עבור P הערך העצמי λ היחיד שמקיים $Pv_i = \lambda v_i$ הוא 1 כנדרש. ■

3.3 סעיף ג

נמצא EVD עבור P:
ראשית ראינו כי הערכים העצמיים של P הם רק 1, וכן P סימטרית.

3.4 סעיף ד

נראה כי מתקיים $P^2 = P^TP = PP^T = P$:
ראינו סעיף קודם כי $P = UDU^T$ הינה פירוק EVD עבור P, ובנוסף ראינו כי $P^T = P$ ולכן $P^2 = P^TP = PP^T$ ואז נפתח ונקבל:

$$P^2 = UDU^T\left(UDU^T\right)^T = UDU^T\left(U\left(UD\right)^T\right) = UDU^TUD^TU^T = UDD^TU^T$$

נזכור כי D היא מטריצה אלכסונית ולכן $D = D^T$, בנוסף הערכים עליה הם 0 או 1ולכן $D = D^2$ מכאן שנמשיך ונקבל:

$$UDD^TU^T = UD^2U^T = UDU^T = P$$

כנדרש. ■

3.5 סעיף ה

נראה כי P $(1-P)$ היא העתקת האפס, קרי לכל $v \in V$ $(1-P)\,Pv = 0$. נזכור כי $P^2 = P$ לכן:

$$(1-P)\,Pv = (I-P)\,Pv = Pv - PPv = Pv - P^2v = Pv - Pv = 0$$

כנדרש. ■

3.6 סעיף ו

נראה כי לכל וקטור $x \in V$ מתקיים $Px = x$, או במילים אחרות כל וקטור במרחב הוא וקטור עצמי בעל ערך עצמי 1:
יהי $x \in V$ וקטור, נוכל לייצג את x כצירוף לינארי יחיד של הבסיס האורתונורמלי שלנו $\{v_1, ..., v_k\}$, כלומר קיימים סקלרים בשדה $\alpha_1, ..., \alpha_k$ כך ש-

$$x = \sum_{i=1}^{k} \alpha_i v_i$$

ואז נזכור כי $\left\langle v_i^T | v_j \right\rangle = 1$ עבור $i = j$ ואחרת 0ונקבל:

$$Px = \left(\sum_{i=1}^{k} v_i v_i^T\right) x = \left(\sum_{i=1}^{k} v_i v_i^T\right) \sum_{j=1}^{k} \alpha_j v_j = \sum_{j=1}^{k}\sum_{i=1}^{k} v_i v_i^T \alpha_j v_j = \sum_{j=1}^{k}\sum_{i=1}^{k} \alpha_j v_i \left(v_i^T v_j\right) =$$

$$= \sum_{j=1}^{k}\sum_{i=1}^{k} \alpha_j v_i \left(\delta_{ij}\right) = \sum_{i=1}^{k} \alpha_i v_i$$

כנדרש. ■

4 שאלה 5

4.1 סעיף א

נגדיר משתנה מקרי Z_i שהוא המציין של ההטלה ה-i, קרי $Z_i = \begin{cases} 1 & H \\ 0 & T \end{cases}$, לכן $Z_i \sim Ber(p)$.
נזכור כי התוחלת והשונות של מ"מ המתפלג ברנולי הם p ו- $p\,(1-p)$ בהתאמה. לכן עבור m הטלות נקבל:

$$Var[\frac{1}{m} \cdot \sum_{i=1}^{m} Z_i] = \frac{1}{m^2} \cdot Var[\sum_{i=1}^{m} Z_i] + 2\sum_{i<j} Cov(z_i, z_j) \underset{z_i\ is\, ind'}{=} \frac{1}{m^2} \cdot Var[\sum_{i=1}^{m} Z_i] + 0 = \frac{1}{m^2} \cdot m \cdot p(1-p) = \frac{1}{m} \cdot p \cdot (1-p)$$

כעת, מאי-שיוויון צ'ביצב מתקבל:

$$P\left(\left|\frac{1}{m}\sum_{i=1}^{m} Z_i - p\right| > \epsilon\right) \le \frac{Var\left[\frac{1}{m}\sum_{i=1}^{m} Z_i\right]}{\epsilon^2} = \frac{p \cdot (1-p)}{m \cdot \epsilon^2}$$

כעת נזכור כי השונות של מ"מ המתפלג ברנולי חסומה מלעיל ע"י $\frac{1}{4}$ ולכן הביטוי שקיבנו מקיים:

$$\frac{p(1-p)}{m \cdot \epsilon^2} \leq \frac{\frac{1}{4}}{m \cdot \epsilon^2} = \frac{1}{4 \cdot m \cdot \epsilon^2}$$

נרצה לקבל $m > 0$ כך ש- $\frac{1}{4 \cdot m \cdot \epsilon^2} < \delta$, לכן נעביר אגפים ונקבל:

$$\frac{1}{4 \cdot \delta \cdot \epsilon^2} < m$$

מכאן שדי לבחור m טבעי שמקיים את התנאי הנ"ל.
כעת, מכיון שע"פ הגדרת Z_i, מתקיים $Z_i \in [0,1]$ לכל $1 \leq i \leq m$, ובנוסף מ"מ אלה הם ב"ת אזי נוכל להפעיל את אי-שיוויון הופדינג ולקבל:

$$P\left(\left|\frac{1}{m}\sum_{i=1}^{m} Z_i - p\right| > \epsilon\right) \leq 2e^{-2m\epsilon^2}$$

ואז באופן דומה נרצה m כך שהביטוי שקיבלנו קטן מ-δ, קרי:

$$2e^{-2m\epsilon^2} < \delta$$

ומכיוון ששתי האגפים חיוביים נוכל להפעיל log על שני האגפים ולבודד את m ולקבל:

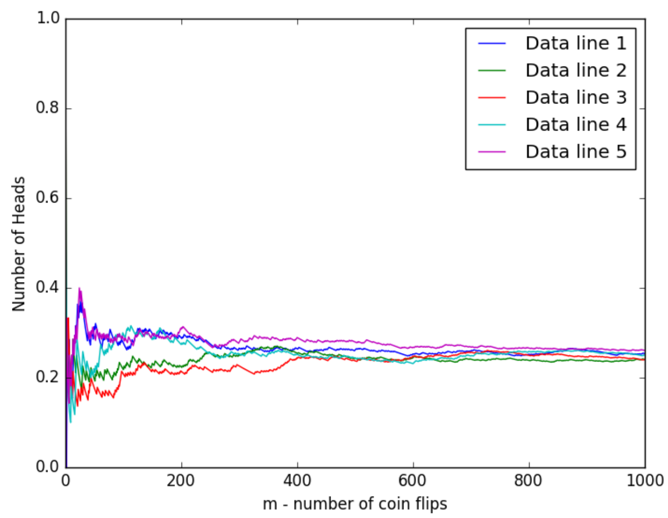
$$\frac{1}{2\epsilon^2} log(\frac{2}{\delta}) < m$$

מכאן שדי לבחור m טבעי שמקיים את התנאי הנ"ל.
כנדרש.■

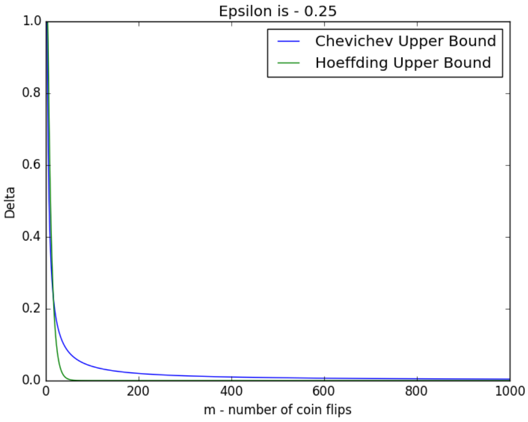
4.2 סעיף ב

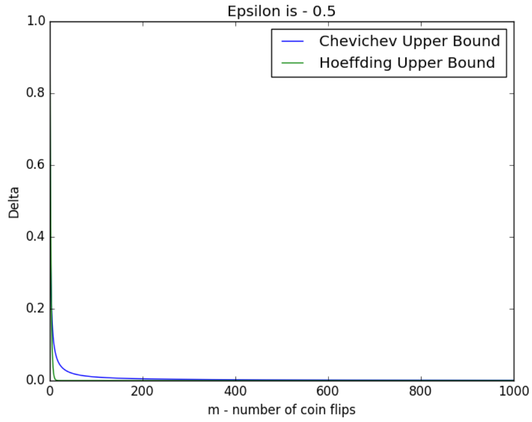
4.2.1 סעיף ב

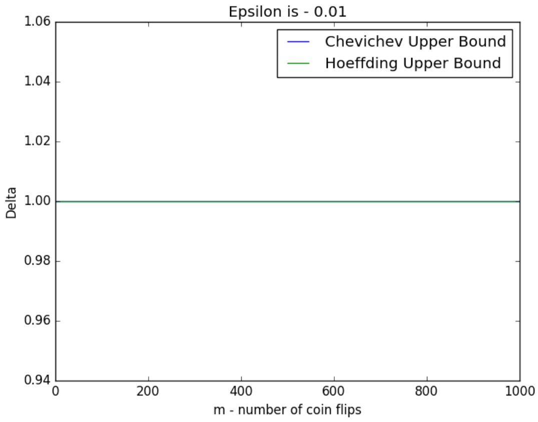
אני מצפה (וכך גם יצא) שככל ש-m גדל (ככל שבוצעו יותר הטלות) כך בסדרות ההטלות השונות (שורות 1-5) ההפרש בין הערכים המתקבלים עבורם יהיו קטנים יותר, כלומר הגרף המייצג שלהם ישאף להתלכדות על גרף התוחלת שלהם $y = 0.25$.
באופן דומה ככל ש-m קטן כך הגרפים עשויים לקבל ערכים רחוקים יותר כפונקציה של m.

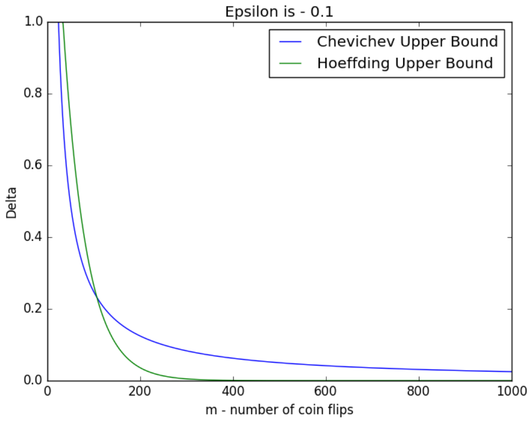


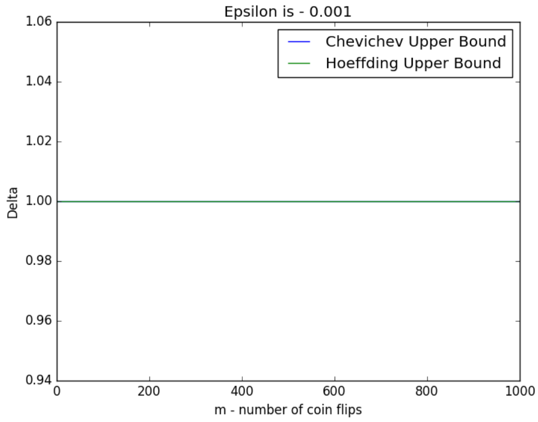
4.2.2 סעיף ג







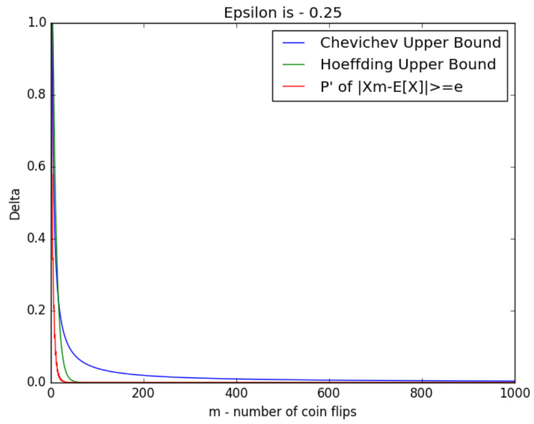


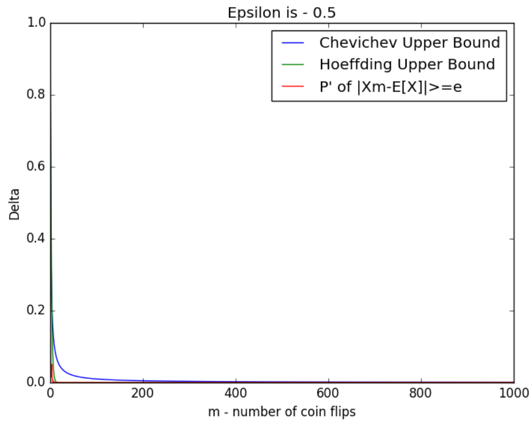


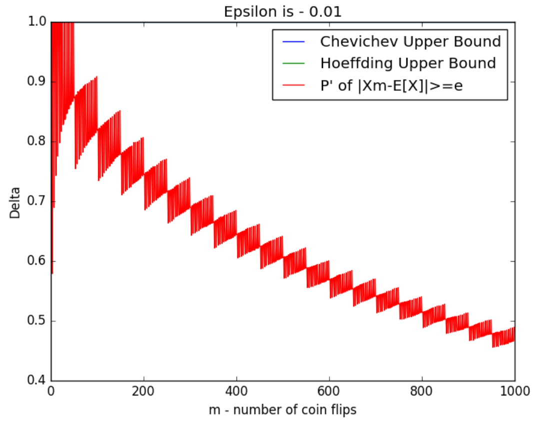
4.2.3 סעיף ד

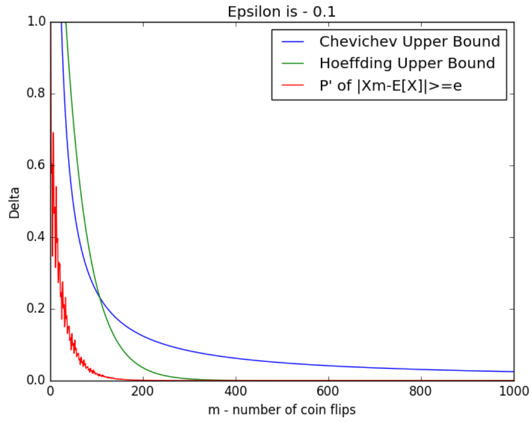
אני מצפה שלכל אפסילון ככל שנבצע יותר הטלות כך X_m יתקרב לתוחלת שלו שהיא p ועל כן עבור m'ים קטנים יש יותר סיכוי לקבל שגיאה (לחרוג מאפסילון) וככל ש-m גדל כך הסיכוי לחרוג מאפסילון קטן.

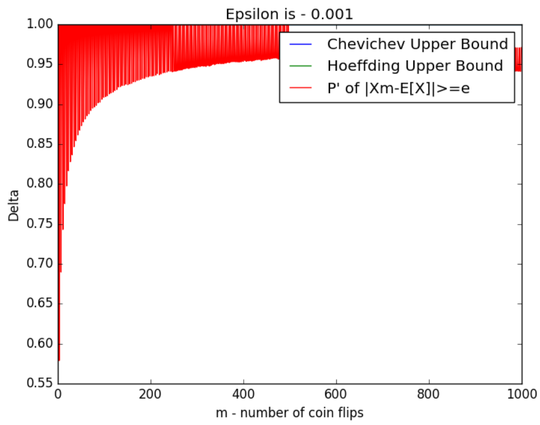
ההבדל בין האפסילונים הוא שערך האפסילון ישפיע על מספר ההטלות שדרוש עבור כל סדרת הטלות כדי שנחרוג פחות מאפסילון, כלומר ככל שאפסילון קטן כך נצטרך יותר הטלות (להגדיל את m) כדי להקטין את השגיאה.


Epsilon is - 0.25
Chevichev Upper Bound
Hoeffding Upper Bound
P' of |Xm-E[X]|>=e
Delta
m - number of coin flips


Epsilon is - 0.5
Chevichev Upper Bound
Hoeffding Upper Bound
P' of |Xm-E[X]|>=e
Delta
m - number of coin flips


Epsilon is - 0.01
Chevichev Upper Bound
Hoeffding Upper Bound
P' of |Xm-E[X]|>=e
Delta
m - number of coin flips


Epsilon is - 0.1
Chevichev Upper Bound
Hoeffding Upper Bound
P' of |Xm-E[X]|>=e
Delta
m - number of coin flips


Epsilon is - 0.001
Chevichev Upper Bound
Hoeffding Upper Bound
P' of |Xm-E[X]|>=e
Delta
m - number of coin flips

2 ex1code.py

```
import numpy
import matplotlib.pyplot as plt
import math


def generator():
    data = numpy.random.binomial(1, 0.25, (100000, 1000))
    epsilon_arr = [0.5, 0.25, 0.1, 0.01, 0.001]
    plot_five(data)
    chev_data, hoef_data = calc_chevichev_hoeffding(epsilon_arr)
    perceneteges = calc_deviators(data, epsilon_arr)
    for epsilon in range(5):
        plt.title("Epsilon is - " + str(epsilon_arr[epsilon]))
        plt.ylabel("Delta")
        plt.xlabel("m - number of coin flips")
        plt.plot(chev_data[epsilon], label='Chevichev Upper Bound')
        plt.plot(hoef_data[epsilon], label='Hoeffding Upper Bound')
        plt.plot(perceneteges[epsilon], label="P' of |Xm-E[X]|>=e")
        plt.legend()
        plt.show()


def plot_five(data):
    rows_sum = [[0]*1000, [0]*1000, [0]*1000, [0]*1000, [0]*1000]
    plt.ylabel("Number of Heads")
    plt.xlabel("m - number of coin flips")

    for m in range(1, 1001):
        for i in range(m):
            rows_sum[0][m-1] += data[0][i]
            rows_sum[1][m-1] += data[1][i]
            rows_sum[2][m-1] += data[2][i]
            rows_sum[3][m-1] += data[3][i]
            rows_sum[4][m-1] += data[4][i]
        rows_sum[0][m-1] /= float(m)
        rows_sum[1][m-1] /= float(m)
        rows_sum[2][m-1] /= float(m)
        rows_sum[3][m-1] /= float(m)
        rows_sum[4][m-1] /= float(m)
    plt.plot(rows_sum[0], label='Data line 1')
    plt.plot(rows_sum[1], label='Data line 2')
    plt.plot(rows_sum[2], label='Data line 3')
    plt.plot(rows_sum[3], label='Data line 4')
    plt.plot(rows_sum[4], label='Data line 5')
    plt.legend()
    plt.show()


def calc_chevichev_hoeffding(epsilon_arr):
    upper_bounds_chev = [[0]*1000, [0]*1000, [0]*1000, [0]*1000, [0]*1000]
    upper_bounds_hoef = [[0]*1000, [0]*1000, [0]*1000, [0]*1000, [0]*1000]
    for m in range(1, 1001):
        for i in range(5):
            chev_res = float(1 / float(4 * m * epsilon_arr[i] * epsilon_arr[i]))
            hoef_res = 2 * float(math.exp(-2 * m * epsilon_arr[i] * epsilon_arr[i]))
            if chev_res > 1:
                chev_res = 1
            if hoef_res > 1:
                hoef_res = 1
```

```python
                upper_bounds_chev[i][m - 1] = chev_res
                upper_bounds_hoef[i][m - 1] = hoef_res
        return upper_bounds_chev, upper_bounds_hoef


def calc_deviators(data, epsilon_arr):
    p = 0.25
    cum_data = numpy.cumsum(data, axis=1)
    total_data = [[0]*1000, [0]*1000, [0]*1000, [0]*1000, [0]*1000]
    for curr_epsilon in range(len(epsilon_arr)):
        for m in range(1, 1001):
            bad_lines_counter = 0
            for i in range(len(data)):
                if math.fabs((cum_data[i][m - 1]/float(m)) - p) >= epsilon_arr[curr_epsilon]:
                    bad_lines_counter += 1
            total_data[curr_epsilon][m - 1] = float(bad_lines_counter/float(len(data)))
        print("Finished epsilon number " + str(curr_epsilon))
    return total_data

if __name__ == '__main__':
    generator()
```