מערכות לומדות תשע"ז - תרגיל 1

מבוא להסתברות

בתרגיל זה תממשו בקוד ותבחנו כמה מן התופעות המתוארות בשיעור 2 – מבוא להסתברות.

נושא 1 - פעולת הקונבולוציה, פעולות קרובות ומספר שימושים

בשיעור 2 תארנו משתנה מקרי בדיד ורציף ותיאור ההתפלגויות. בתנאים מסוימיים ההתפלגות של סכום משתנים מקריים בלתי תלויים מתקרבת להתפלגות נורמלית (גאוסיאנית).

בשאלה זו תשתמשו בקונבלוציה כדי להמחיש תופעה זו, עבור המקרה הבדיד.

ההגדרת קונבולוציה למקרה הבדיד:

$$(f*g)[n] \stackrel{\text{def}}{=} \sum_{m=-\infty}^{\infty} f[m]\, g[n-m] = \sum_{m=-\infty}^{\infty} f[n-m]\, g[m].$$

שימו לב שזו הגדרה סימטרית ל ***f*** ול ***g***.

לפעמים נרצה להשתמש בפונקציות שתחום הערכים שלהן מוגבל - finite support, **ואף** אינו באותו אורך. זה נפוץ כאשר ***g*** מהווה את גרעין הקונבוליצה ותחום הערכים שלה קטן בצורה משמעותית משל ***f***.

עבור מקרה זה של תחום ערכים מוגבל עבור ***g***

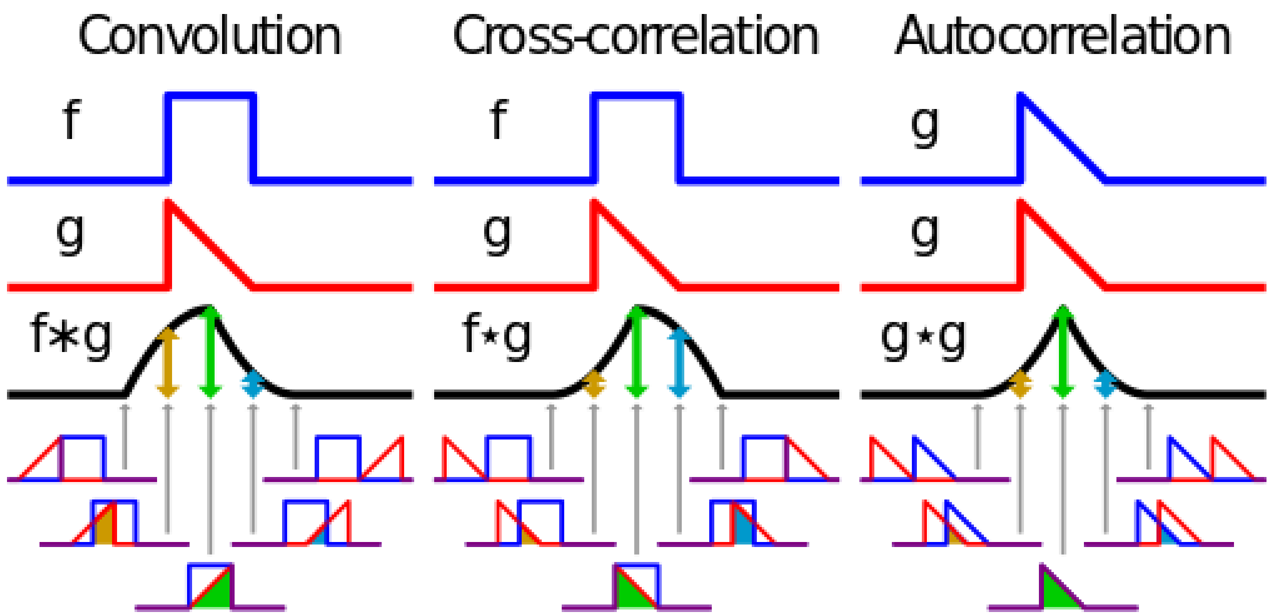
$$\{-M, -M+1, \ldots, M-1, M\}$$

ההגדרה היא:

$$(f*g)[n] = \sum_{m=-M}^{M} f[n-m]g[m]$$

פעולות הקשורות לקונבולוציה הן קרוס-קורלציה ואוטוקורלציה.

הנה המחשה לפעולתן (מתוך וויקיפדיה)



בשיעור הזכרנו משפט הטוען כי התפלגות של סכום שניים (או יותר) משתנים מקריים בלתי תלויים היא הקונבולוציה של ההתפלגויות:
https://en.wikipedia.org/wiki/Convolution_of_probability_distributions

הנה רשימה של התפלגויות ידועות וההתפלגות של הסכום שלהן:
https://en.wikipedia.org/wiki/List_of_convolutions_of_probability_distributions

בנוסף הזכרנו את משפט הגבול המרכזי בהקשר של סכום משתנים מקריים - סכום של משתנים מקריים בלתי תלויים נוטה להתפלג נורמלית ככל שמספר המשתנים המקריים גדול יותר (גם אם המשתנים המקריים עצמם אינם מתפלגים נורמלית!):
https://en.wikipedia.org/wiki/Central_limit_theorem

קונבולוציה מהווה כלי עזר חישובי ליצירת ההתפלגות של הסכומים.

כעת נשתמש בשני המשפטים כדי לייצר קרוב דיסקרטי להתפלגות גאוסיאנית.
נתחיל עם קרוב דיסקרטי של שני איברים בלבד להתפלגות גאוסיאנית: מה הוא?

התפלגות נורמלית היא סימטרית, וככל התפלגות סכומה 1, על כן הקרוב הוא [0.5, 0.5].
נכנה אותו הקרוב הראשון, (שכבר נתון לנו לא דורש עוד עבודה).

כדי להשיג את הקרובים הבאים, נשתמש בקרוב הראשון כגרעין הקונבולוציה g, כך:

```
f = g
f = conv(f,g)
```

על כן הקרוב השני יכיל שלושה איברים ויהיה

[0.25, 0.5, 0.25] = conv ([0.5, 0.5], [0.5, 0.5])

וכך, ליצירת כל קרוב נוסף, נבצע קונבולוציה של הקודם f, עם הגרעין g.
(שאלה למחשבה: האם ניתן לבצע זאת בצורה יותר יעילה?)

שאלה 1

א. כיתבו פונקציה בשם `discrete_gauss` המחזירה קרוב בעל **n איברים**.
הפונקציה צריכה לפעול עבור n שלם בלבד, בין הערכים 2 ל 1000.
שימו לב לקשר בין n למספר הקונבולוציות הנדרשות על פי השיטה שהצענו.

ב. מה צריך להיות סכום הערכים של התוצאה? הוסיפו בדיקה לקוד שלכם.

ג. כיתבו פונקציה `show_discrete_gauss` המציגה בצורה של bar graph את תוצאות `discrete_gauss`. הריצו אותה עבור הערכים 2, 5, 10, 25, 50, 100, 500, 1000. העתיקו את הfigures אל קובץ הפתרון המוגש.

ד. כעת בחנו את ההשפעה של שימוש בגרעין לא אופטמלי מן הצורה

[a, 1-a],   0 < a < 1

הוסיפו לפונקציה `discrete_gauss` קלט שני שהוא הגרעין, והשתמשו גרעין האופטימלי כערך ברירת מחדל.

השתמשו בחתימה הבאה:

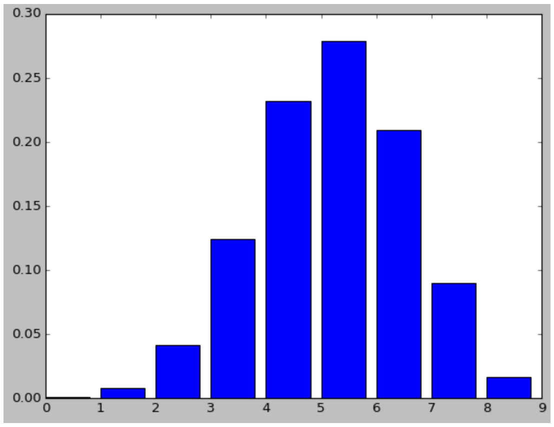
```
def discrete_gauss(n, g=[0.5, 0.5]):
    """
    discrete_gauss(n, g=[0.5, 0.5])

    Estimates the discrete Gaussian distribution (probability mass function)
    by multiple convolutions with a minimal kernel g.

    :param n: scalar.
              the number of elements of the result (n = 2..1000).
              the functions performs n-2 convolutions to create the result.

    :param g: 1-D array.
              the minimal kernel. Default value is [0.5, 0.5].
              Other kernels of the form [a, 1-a],
              where 0 > a > 1.0 are possible, but they are less effective:
                  1. a larger n should be used to be as similar to a Gaussian.
                  2. the peak of the result is not centered.

    :return: 1-D array.
             f, the discrete estimate of Gaussian distribution.
             f has n elements.
    """
```

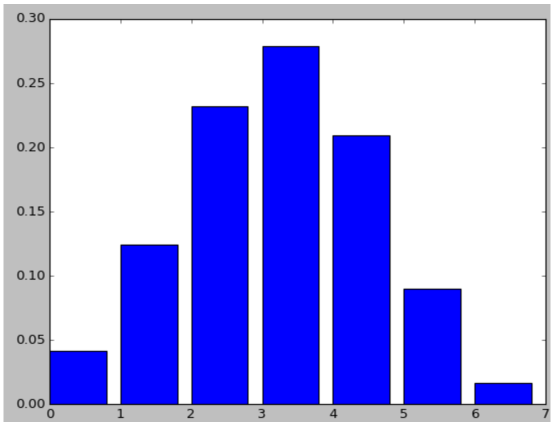
התאימו את הפונקציה `show_discrete_gauss` לחתימה החדשה של `discrete_gauss`. הריצו עם הגרעין [0.1, 0.9] עבור אותם ערכי n של סעיף ג. העתיקו את הfigures לקובץ הפתרון המוגש.

ה. השוו את איכות הקרוב של גרעין אופטימלי לזה של גרעין לא אופטימלי בצורה הבאה.

כיתבו פונקציה ממרכזת, המקבלת את התוצאה של `discrete_gauss` מסעיף ד, ומזיזה את הערכים כך שהשיא יהיה האיבר המרכזי בתשובה. למשל, עבור n=9:

```
a2 = discrete_gauss(9, [0.4, 0.6]); show_bar(a2)
a3 = move_peak_to_center(a2); show_bar(a3)
```



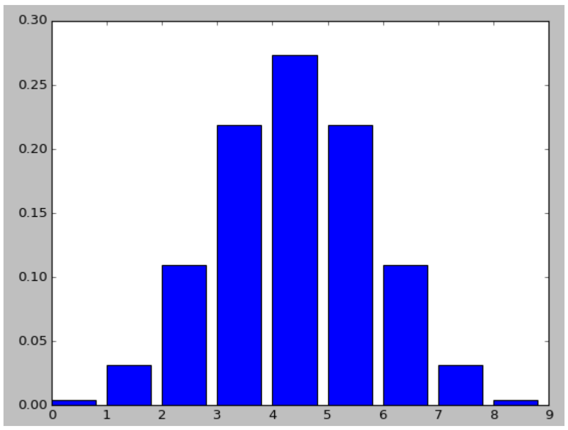


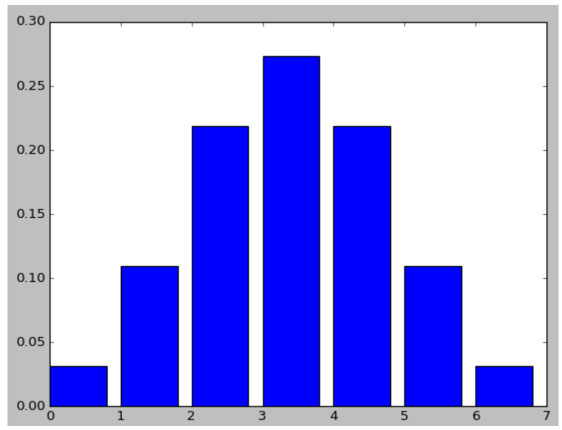
כזכור, נרצה לחשב את המרחק בין הקרוב הלא אופטימלי לאחר שמורכז, אותו נכנה non_optimal_n לבין הקרוב האופטימלי עבור אותו n מקורי, אותו נכנה optimal_n.

שימו לב שההשוואה תיתכן רק לאחר חיתוך optimal_n לאותו מספר איברים כמו ב non_optimal_n. כיתבו פונקציה crop המבצעת חיתוך זה.

למשל, עבור n=9 כמקודם:

```
a1 = discrete_gauss(9); show_bar(a1)
a4 = crop(a1,a3); show_bar(a4)
```





לכל התהליך של הזזת וקטור אחד וקיצוצו, והתאמת הוקטור השני לאותו מספר איברים נקרא **Alignment**.

מרכיב נוסף שדורש הגדרה הוא פונקצית המרחק בעזרתה מחושב המרחק בין הקרובים השונים – שהוא מרחק בין וקטורים של ערכים. פונקצית המרחק בה תשתמשו היא scipy.spatial.distance.cosine:

$$1 - \frac{u \cdot v}{||u||_2 ||v||_2}$$

ראו https://docs.scipy.org/doc/scipy/reference/generated/scipy.spatial.distance.cosine.html

1. מדוע מרחק זה נקרא cosine distance ?
2. מה המרחק עבור וקטורים זהים? הסבירו.
3. מה המרחק עבור וקטורים הפונים בדיוק בכיוונים מנוגדים? הסבירו.

4. מה טווח ערכי המרחק הצפוי עבור ההשוואות של שאלה זו? הסבירו.

כעת, כשכל המרכיבים קיימים כיתבו פונקציה המחשבת ומציגה:
עבור n=999 (מספר האיברים בוקטור התוצאה של הקרוב) את המרחק בין הקרוב הנבנה עם הגרעין האופטימלי, לקרוב הנבנה עם הגרעין $0 < a < 1$, $[a, 1-a]$, כפונקציה של ערכי a.
השתמשו בערכי a החל מ 0.02 ועד 0.98 בקפיצות של 0.02 (בעזרת numpy.arange).

הפונקציה תציג את המרחק כפונקציה של a בעזרת גרף. הוסיפו כיתוב לצירים וכותרת. העתיקו את ה figure המתקבל אל מסמך הפתרון.

ו. כמו הסעיף הקודם, הפעם עם פונקצית מרחק אחרת: סכום הערך המוחלט של ההפרשים בין הוקטורים.

1. מה המרחק עבור וקטורים כללים זהים? הסבירו.
2. מה המרחק עבור וקטורים כללים הפונים בדיוק בכיוונים מנוגדים? הסבירו.
3. מה טווח ערכי המרחק הצפוי עבור ההשוואות של שאלה זו? הסבירו.

העתיקו את ה figure אל מסמך הפתרון.

ז. (סעיף בונוס)
בסעיפים הקודמים הצגנו שיטת Alignment שכללה: מירכוז של non_optimal_n, בצורה שמקטינה את מספר האיברים בוקטור; חיתוך של optimal_n לאותו מספר איברים (בצורה השומרת כמובן את הסימטריה של הערכים); ולבסוף מציאת המרחק בין שני הוקטורים הממורכזים ובעלי אותו מספר איברים.

בצורת Alignment זו, מאבדים חלק מן המידע הקיים בכל אחד מן הוקטורים.

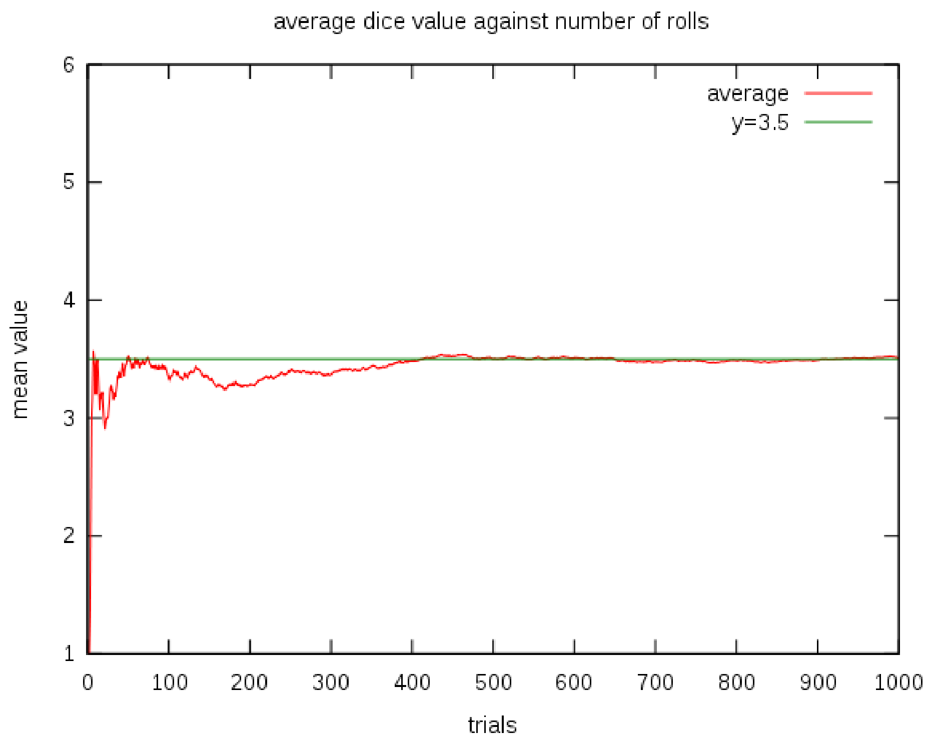
1. הציעו שיטת Alignment אחרת, המביאה את שני הוקטורים לכך שלשניהם יהיה אותו מספר איברים, וגם שהערך הגבוה ביותר של שניהם יהיה באותו אינדקס, כל זאת ללא כל איבוד מידע. הסבירו את השיטה במסמך הפתרון.
2. ממשו את השיטה שהצעתם.
3. בצעו את הסעיפים ה' ו ו' עבור אותם תנאים, אך עם שיטת ה Alignment שהצעתם. הציגו את התוצאות הגרפיות במסמך הפתרון.
4. האם יש הבדלים בין התוצאות? הסבירו.

נושא 2 – חוק המספרים הגדולים

בשיעור 2 תיארנו שממוצע המדגם שואף (מתכנס) לתוחלת כאשר גודל המדגם שואף לאינסוף.

ראו http://en.wikipedia.org/wiki/Law_of_large_numbers

וההדגמה הגרפית



שאלה 2

בשאלה זו תייצרו כמה סימולציות לחישוב ממוצע המדגם ביחס לתוחלת ותציגו את התוצאות בגרפים.

א. ממשו פונקציה המדמה הגרלה בודדת של קוביה הוגנת.
ב. כיתבו קוד המשתמש בפונקציה להגרלת 1000 הטלות, ואוסף את התוצאות – ערך הקוביה בכל הטלה.
ג. מה היא התוחלת של ערכי ההטלות? הגישו חישוב מפורש במסמך הפתרון.
ד. כתבו קוד להצגת גרף (כפי שמוצג כאן מעל) ובו ערך ההטלה כנגד מספר ההטלה, והצגת ערך התוחלת בעזרת קו אופקי. הוסיפו כיתוב לצירים וכותרת. צרפו את ה figure המתקבל למסמך הפתרון.

ה. הריצו את המערכת פעם נוספת ל 1000 הטלות. יצרו גרף וצרפו גם אותו. האם הגרפים המתקבלים זהים בערכיהם? הסבירו.

ו. הוסיפו למערכת שכבה חיצונית המריצה 100 פעמים, בכל פעם 1000 הטלות. עבור כל אינדקס הטלה, הקוד יחשב את הממוצע ואת השונות על פני 100 הפעמים.

הציגו שני גרפים:

1. **הממוצע** של כל הטלה על פני 100 הפעמים, כפונקציה של מספר ההטלה.
2. **השונות** של כל הטלה על פני 100 הפעמים, כפונקציה של מספר ההטלה.

הגשה

א. תאריך הגשה: עד יום ראשון, 4.12.16, בשעת חצות הלילה.

ב. ניתן להגיש בזוגות.

ג. יש לכתוב **שם \ שמות + תז** בראשית כל מסמך מוגש (**כולל בקבצי הקוד**).

ד. כל מגיש (ביחיד או בזוג) צריך לדעת להסביר כל מה שנעשה בפתרון המוגש. חלק מן המגישים ידרשו להסביר את הפתרון שלהם למרצה.

ה. יש להגיש מסמך Word המכיל את כל התשובות לתרגיל. שם מסמך זה יהיה **ex1.docx**.

הקפידו שמיספור סעיפי התשובות שלכם יהיה זהה למיספור סעיפי השאלות.

ו. לכל פונקציה צריך להיות תיעוד במתכונת של הדוגמה המצורפת.

ז. יש להגיש את כל הקוד לתרגיל בקובץ יחיד בשם **ex1.py**. בתחילת הקובץ יבואו הגדרות כל הפונקציות (אלה שנדרשו בתרגיל וגם כל פונקציות העזר שלכם). בהמשך הקובץ יבוא חלק ההרצה. חלק זה **יופרד על ידי הערות לכל אחד מסעיפי השאלות.**

ח. שני הקבצים ישכנו בתוך תיקייה הכוללת את שמכם.

שם התיקיה למגיש יחיד:

EX1FamilyName

שם התיקיה לשני מגישים:

EX1Family1Family2

התיקיה תארז לקובץ **zip** בעל אותו שם כשל התיקיה (לא rar).

הנה תבנית לקובץ הקוד ex1.py:

```python
# Family1 Name1  Id 000000000
# Family2 Name2  Id 111111111
# ex1.py

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


# --------- CONSTANTS declarations ---------------------------
epsilon = math.ldexp(1.0, -53)



# --------- FUNCTIONS / CLASSES declarations -----------------

def discrete_gauss(n, g=[0.5, 0.5]):
    """
    discrete_gauss(n, g=[0.5, 0.5])

    Estimates the discrete Gaussian distribution (probability mass function)
    by multiple convolutions with a minimal kernel g.

    :param n: scalar.
              the number of elements of the result (n = 2..1000).
              the functions performs n-2 convolutions to create the result.

    :param g: 1-D array.
              the minimal kernel. Default value is [0.5, 0.5].
              Other kernels of the form [a, 1-a],
              where 0 > a > 1.0 are possible, but they are less effective:
                  1. a larger n should be used to be as similar to a Gaussian.
                  2. the peak of the result is not centered.

    :return: 1-D array.
             f, the discrete estimate of Gaussian distribution.
             f has n elements.
    """



# --------- RUNING the solution to the exercise -----------------

# --------- Question 1a ----------------------------------------


# --------- Question 1b ----------------------------------------



# --------- END OF FILE ----------------------------------------
```