# **ניסוי בקבלת החלטות**

כל הקוד נמצא כאן: <https://github.com/nufarlevy2/Rinat-Nufar-exp.git>

ניתן להוריד את כל הקבצים שם בZIP ולפתוח אותם על המחשב שלך.

## **אסביר על החלקים החשובים:**

* **template1.m** – בקובץ הזה כתוב הניסוי עצמו וצירפתי הערות על דברים שאני שיניתי. בעיקרון את הדברים החשובים הוצאתי לחלק של ההגדרות. החלק החשוב באמת הוא Number Of Trials Per Block וזה חלק שאפשר לשנות בחלק של ההגדרות בתחילת הניסוי.  
  בתחילת הניסוי מתרחשים הדברים הבאים:
* (stage 3) הוא טוען את כל ההוראות מהתיקייה שנמצא ב: MSGS\_FOLDER – resources/instructions
* (stage 3) טוען את כל התמונות של המועמדים מהתיקייה: PATH\_TO\_PICS – resources/pics
* randomizeConditions() – פונקציה שעושה רנדום לכל הטריילים, בחלק הראשון של המטריצת תנאים היא שמה 2 מועמדים בכל טרייל ובחלק השני רק אחד. הפונקציה דואגת שכל מועמד יהיה לפחות פעם אחת במידה ויש מספיק טריילים.
* Run\_blocks+run\_trial – הם דיי מה שאת רואה. אז כתבתי הערות בקוד ותוכלי לראות שם אם תצטרכי לשנות/להוסיף דברים.

## **מהלך הניסוי:**

* בלוק 1 - הוצגה לנבדקים נקודת פיקסציה באמצע המסך. לאחר מכן הוצגו להם שתי תמונות של מועמדים פוליטים והם היו צריכים לבחור במקש עם **החץ** **לימין** **לתמונה הימנית**, במקש עם **החץ לשמאל לתמונה השמאלית** ובמקש עם **החץ** **למטה** אם הם רוצים **להימנע**.
* בלוק 2 - בדומה לחלק הראשון, הוצגה לנבדקים נקודת פיקסציה ולאחר מכן הוצג לנבדקים תמונה של אחד המועמדים הפוליטיים. לאחר מכן **בשונה**מהחלק הראשון, הנבדקים לא צריכים לבחור, אלא לדרג. הם צריכים לדרג את התמונה בדירוג שנע בין 1 ל10.

## **אנליזה**

* **CompleteAnalyze.m** - בcomplete analyze אנחנו מנחים את התוצאות לאחר הרצת הנבדקים. התוכנה מקבלת מספר נבדק ופרמטרים למודלים Race, Diffusion ו-LCA.   
  function result = CompleteAnalyze(subjectNumber, Race, Diffusion, LCA)

התוכנה שולפת את כל הקבצים הרלווטיים מהתיקיות הנ"ל לפי מספר הנבדק:

* resources\data files\behavioral data – את המידע של EXPDATA.
* resources\matlabFiles – שם יש תיקיה לכל נבדק ובתוכה נשים:
  + את הstruct שאנחנו מוצאים לאחר שהוצאינו את האנליזה מהתכנה של עומר - **analysis\_struct.mat** – חשוב שנקרא לקובץ באותו השם שרשמתי כאן באדום.
  + **familiratyTable.mat** - אם אנחנו רוצים להכניס גם מטריצת הכירות עם כל המועמדים כך שהאנליזה תחשב בכך שהמועמד מכיר מועמדים מסויימים מתוך המאגר, נוכל לעשות ואלידציה נוספת על כל הטריילים. נצטרך לשים בתיקייה קובץ מטלב נוסף שמכיל שורות כמספר התמונות שהיו בניסוי. ניקח את כל התמונות לפי סדר אלפבתי (כמו שהם בתיקייה) ולגבי כל אחד נכתוב: אם הנבדק הכיר את המועמד נשים 1 אם לא נשים 0.

כל אחד מהפרמטרים של המודלים מכיל את הדברים הבאים:

* Threshold – סף שקובע מתי המודל מפסיק, בדכ אנחנו מגיעים ל"ניצחון" הרבה לפני הthreshold אז אני בד"כ שמה משהו ממש גדול (10 בחזקת 10).
* primaryInput – ה"משקל" שנותנים למנצח באותו הטרייל, בכל אחד מהמודלים. ברירת מחדל 0.9
* SecondaryInput - ה"משקל" שנותנים למפסיד באותו הטרייל, בכל אחד מהמודלים. ברירת מחדל 0.1
* noiseM – ממוצע הרעש שנוסיף למודל (בד"כ 0)
* noiseSTD – סטיית תקן של הרעש (משתנה בין כל אחד מהאנליזות)
* Leak – קיים רק בLCA.

## **הסבר על קבצי הפלט:**

### **פר נבדק:**

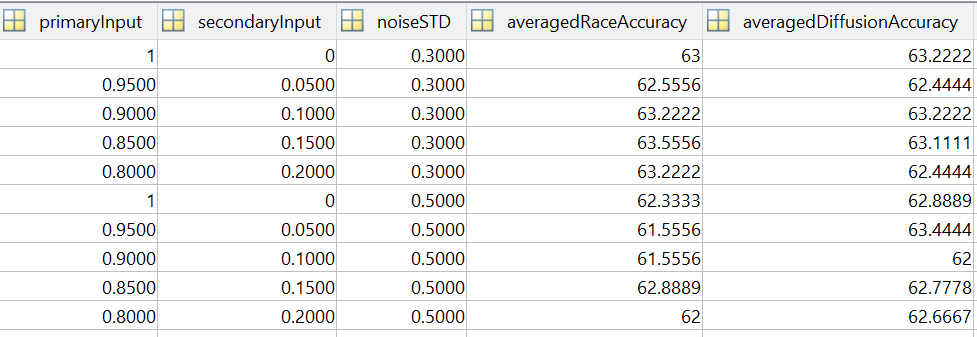
האנליזה על כל נבדק כללה מספר בדיקות ואיגדה את כולן בקובץ mat. אחד:  
**הקובץ נמצא ב:** resources\matlabFiles\s<subject number>, צריך להיכנס ל: analysis\_struct{1,1}.c2.fixations :

* + - תגובה (response), דירוג תמונה ימנית ושמאלית (left/right\_picture\_rank), מספר התמונות שהוצגו ושמות המועמדים.
    - משך זמן ההסתכלות על כל תמונה - בתצורה של זמן מכונה: זמן הסתכלות רציפה על התמונה (left/right\_pic\_duration\_per\_fixation), סכימה של כל הזמן שהסתכלו על כל אחת מן התמונות (left/right\_pic\_duration\_sum), ניתוח זה כלל גם את אותן בדיקות על איזורים שונים כמו איזור הפנים ואיזור העיניים  (eyes/face\_of\_left/right\_pic\_duration\_sum,  eyes/face\_of\_left/right\_pic\_duration\_per\_fixation).
    - מספר מעברים לימין ולשמאל (passes\_to\_rightqleft)
    - האם הפיקסציה האחרונה ניבאה את הבחירה של הנבדק (last\_fixation\_predicted\_response).
    - האם המודל של הRACE או המודל של הDIFFUSION ניבאו בצורה טובה את תגובת הנבדק (הסבר יותר רחב בהמשך - race/diffusion/lca\_model\_predicted\_response)

## **פר הרצת כלל הנבדקים -** הוצאת דוח שמפרט על כל הנבדקים ובדיקות נוספות רוחביות פר נבדק.

## C:\Users\levyn\AppData\Local\Microsoft\Windows\INetCache\Content.Word\1.png

* RankMatchesModelsOtherDirectionPresentage - קיבלנו אינדיקציה ממשה גליקמן לכך שיש נבדקים שמסתכלים יותר על התמונות שהם לא בוחרים ולכן המודלים ינבאו בקורלציה הפוכה לגביהם. אז לגבי המועמדים שהמודל לא ניבא בצורה משמעותית מעבר ל"צ'אנס" הרצנו בדיקה שבודקת על פי ממוצע הדירוגים פר תמונה והאם המודל ניבא בצורה טובה את הדירוגים הנמוכים יותר - קיבלנו בערך ממוצע של 60 שזה הרבה מעבר ל"צ'אנס".
* ReliablePrecentage - פה בדקנו את מהימנות הנבדקים (הטבלה הזאת לא מכילה 2 נבדקים שמהימנותם יצאה נמוכה מאוד) לפי: האם הם בחרו בבלוק הראשון את התמונה שאותה דירגו יותר גבוה בבלוק השני.
* Race/DiffusionAccuracyOrigin - תוצאות מודלים הrace והdiffusion ללא ההשפעה של האם יש לנו נבדק שהמודל מנבא בצורה הפוכה.
* Race/DiffusionAccuracyFinal - תוצאות מודלים הrace והdiffusion לאחר הפיכת האחוזים לנבדקים שהמודל מנבא בצורה הפוכה.

**אופטימיזציה של המודלים -** ניסיון לשחק עם הפרמטרים של המודלים (לא ראינו שינוי משמעותי בין הפרמטרים השונים)  
  
המודלים עוברים על כל מילישנייה בdata ובודקים את הנבדק הסתכל על הפנים של התמונה הימנית/השמאלית/אף אחד מהדברים.

* primaryInput - האינפוט שנתנו לתמונה שעליה הנבדק הסתכל
* secondaryInput - האינפוט שנתנו לתמונה שעליה הנבדק לא הסתכל, במקרה והסתכל על התמונה השנייה.  
  במקרה והנבדק לא הסתכל על עף אחת מן התמונות שתי התמונות יקבלו אינפוט ניטרלי של 0
* noiseSTD - סטיית התקן של הרעש שהוספנו למודלים הנדגם מהתפלגות נורמלית עם ממוצע 0.
* averagedRaceAccuracy - דיוק של הRaceModel.
* averagedDiffusionAccuracy - דיוק של הDiffusionModel.

טוב, אז התוצאות שקיבלנו בשני המודלים הגיוניות כי לא השתמשנו בדירוגים, נתנו למודל לנבא והגענו לתוצאות שמגיעים בתחום שזה סביבות ה60-65%.

### **הסבר קצר על כל אחד מן המודלים:**

* Race - מודל זה הוא מודל נאיבי כאשר כל פעם שמסתכלים על מועמד המודל מקבל input גבוה והמועמד השני, שלא מסתכלים עליו, מקבל input נמוך. עד שכל הinputים מצטברים והגבוה ביותר מנצח בניבוי ההחלטה.  
  נוסחאת המודל היא:  
  X1(t+1) = X1(t) + leftInput + normrnd(Race.noiseM, Race.noiseSTD);  
  X2(t+1) = X2(t) + rightInput + normrnd(Race.noiseM, Race.noiseSTD);
* Diffusion - מודל זה הוא מודל כמו המודל הקודם אך ההבדל כאן הוא שיש lateral inhibition ולפי זה ההסתכלות על מועמד א מדכאת את ההסתכלות על מועמד ב'.  
  נוסחאת המודל היא:  
  Delta1 = leftInput + normrnd(Diffusion.noiseM, Diffusion.noiseSTD);  
  Delta2 = rightInput + normrnd(Diffusion.noiseM, Diffusion.noiseSTD);  
  X1(t+1) = X1(t) + Delta1 - Delta2;  
  X2(t+1) = X2(t) + Delta2 - Delta1;

## **יש אפשרות לתת מספר נבדקים בבת אחת עם מספר פרמטרים:**

אם נשתמש בcompleteAnalyzeForNumOfSubject עם הפרמטרים הבאים:

function competeAnalyzeForNumOfSubject(subjectNumbers, parameters)

1. subjectNumbers – cell של string של מספרי נבדקים. לדוגמא:

>> subjectNumbers = {'100', '107', '108', '110', '113', '115', '116', '117'};

1. parameters – optional = לא חובה להכניס פרמטר, אם לא מכניסים הוא עושה את הברירת מחדל שלו שזה: [[0.95,0.05,0.3]]

אם רוצים להכניס פרמטרים אחרים: צריך להביא מטריצה של פרמטרים כשכל שורה מייצגת סדרה אחת של פרמטרים. בעמודה הראשונה נשים את הprimaryInput בעמודה השניה נשים את הsecondaryInput ובעמודה האחרונה את הnoiceSTD. לדוגמא:

>> parameters = [[1,0,0.3];[0.95,0.05,0.3];[0.9,0.1,0.3];[1,0,0.5];[0.95,0.05,0.5];[0.9,0.1,0.5]];

1. NUM\_OF\_TRIALS\_FOR\_MULTIPLE\_CHECKS – optional – גם אותו לא חובה להכניס, אם לא מכניסים את הפרמטר הוא מריץ פעם אחת כל נבדק וכל סט של פרמטרים.  
   המטרה שלו הוא להריץ מספר סטים של פרמטרים+מספרי נבדקים כדי לעשות בסוף ממוצע על הכל (למשל להריץ 100 פעמים את נבדק 108 על סט פרמטרים של [[0.95,0.05,0.3]].

**אם יש שאלות מעבר להערות ולתיעוד שלי אשמח לענות תמיד 😊**