

רקע

עקב הצלחתכם הרבה בתרגיל הבית הראשון בעזרה לחברת הצעצועים "DollsRUs", פנתה אליכם הפעם חברת "Flexnit" וביקשה שתבנו בשבילה אלגוריתם למערכת ההמלצה. מערכת ההמלצה שלהם מורכבת ומכילה שלושה סוגי משתתפים:

1. צרכני התוכן- משתמשי המערכת (לדוגמא, אתם כשאתם צופים בסדרות ב-"Flexnit"). הצרכנים מחולקים לכמה קבוצות של משתמשים. כל קבוצה מייצגת סוג אוכלוסייה בעלת העדפות שונות.
2. יצרני התוכן- האנשים שמכניסים תוכן למערכת ההמלצה (לדוגמא, יוצרי הסדרה האהובה "יום רביעי" המשודרת ב-"Flexnit").
3. החברה המסחרית אשר מפעילה את מערכת ההמלצה ("Flexnit" עצמה).

משימה

בתרגיל זה תתנסו ביצירת אלגוריתם הממליץ על תכנים לצרכנים המשתמשים במערכת. עליכם לממש אלגוריתם אשר בוחר לכל משתמש המגיע אל המערכת של "Flexnit" את הסדרה שתומלץ לו. לאחר שהמשתמש קיבל את ההמלצה, הוא צופה בסדרה ומדווח למערכת את שביעות רצונו (הרווח שלו) מהתוכן שקיבל (לדוגמא דירוג או לייק). הרווח של משתמש מסוג i מהתוכן של יצרן j הינו משתנה מקרי המגיע מהתפלגות לא ידועה. שימו לב, הרווח של שני משתמשים מאותו יצרן תוכן עלול לנבוע מהתפלגויות שונות ובדומה לכך גם הרווח של משתמש מסוים משני יצרנים שונים. אחת ממטרות האלגוריתם שלכם היא ללמוד את ההתפלגויות הנ"ל (תוך שימוש באלגוריתם Multi-Armed Bandit לדוגמא) על מנת להמליץ למשתמשים בצורה הטובה ביותר.

בכדי להציג את הסדרה שלהם ב-"Flexnit", יצרני התוכן צריכים להשקיע כסף (קניית ציוד, תשלום לעובדים וכו'), העלויות האלו מכוסות על ידי הרווחים שהם מפיקים מהמערכת (בכל פעם שמשתמש צופה בתוכן שלהם, "Flexnit" משלמת להם דולר אחד). לכן, ככל שצופים בהם יותר משתמשים הם מרווחים יותר כסף. אם יצרן תוכן לא מקבל מספיק צפיות במערכת (כלומר התוכן שלו לא מומלץ למספיק משתמשים) הוא מבטל את החוזה שלו עם "Flexnit" ומאותו רגע הסדרה שלו אינה זמינה לצפייה. אם תמליצו על סדרה שהוסרה על ידי יצרן התוכן שלה, יתקבל מסך כחול והרווח למשתמש יהיה 0. לכל יצרן תוכן יש ערך סף (הידוע לכם) המציין את כמות המשתמשים שהיצרן דורש שיצפו בתוכן שלו במשך חלון זמן מסוים (phase, כפי שיוסבר בהמשך). המשימה שלכם, היא להמליץ למשתמשים על תוכן כך שיהיו מרוצים ככל האפשר, תחת התייחסות לאילוצים שדורשים מכם היצרנים.

הסבר כללי על הקבצים המצורפים

- `- id_123456789_987654321.py` – קובץ המממש את אלגוריתם ההמלצה שלכם. זה הקובץ היחיד בתרגיל הבית שעליכם לשנות, כמפורט בהמשך.
- `- simulation.py` – קובץ המנהל את הסימולציה ומכיל את כלל הפונקציות הנדרשות לכך. הסבר על הסימולציה מפורט בהמשך.

כאמור, הקובץ היחידי שעליכם לשנות הוא הראשון. זיכרו כי כל שינוי אחר בקבצים יתבצע **לוקלית** אצלכם, ובבדיקת התרגיל נשתמש בקבצי המקור שקיבלתם.

מהלך הסימולציה

כל סימולציה מורכבת ממספר פרמטרים, כל קומבינציה אחרת של פרמטרים תוביל לתוצאה שונה לחלוטין. כל סימולציה מתקיימת למשך מספר סיבובים שנקבע בפרמטרים של הסימולציה. בכל סיבוב מגיע משתמש אל המערכת (שנדגם מהסתברות המצויינת בפרמטרי הסימולציה), והאלגוריתם שלכם ממליץ לו על יצרן תוכן. מהמלצה זו מתקבל רווח (משתנה מקרי הנדגם על פי הפרמטרים הקשורים למשתמש וליצרן התוכן). מטרתכם היא למקסם את סכום הרווחים. בכל חלון זמן באורך τ סיבובים המערכת בודקת מי הם היצרנים שאינם מרוצים, כלומר לא הומלצו מספיק פעמים על פי הדרישות שלהם, והיא מסמנת אותם כיצרנים לא פעילים. הרווח של משתמש מיצרן לא פעיל הוא 0.

סיפקנו לכם חמישה סטים שונים של פרמטרים להרצת הסימולציה, אתם רשאים לשנות אותם ולהוסיף עוד סטים כרצונכם.

הנחות הסתברותיות

תוצאות הסימולציה נדגמות לפי הכללים הבאים:

- משתמש מסוג i מגיע אל המערכת בסיבוב מסוים בהסתברות $users_distribution[i]$. לדוגמא: אם $user_distribution = [0.2, 0.8]$ אז בסיבוב מסוים בהסתברות 0.2 יגיע אל המערכת משתמש מסוג 0, ובהסתברות 0.8 יגיע אל המערכת משתמש מסוג 1.
- הרווח למשתמש מסוג i מצריכת התוכן של יצרן j נדגם מהתפלגות יוניפורמית בקטע $[0, b_{ij}]$ כאשר b_{ij} אינו ידוע למערכת.

הסבר קוד הסימולציה

הסימולציה ממומשת בקובץ `simulation.py` בפונקציה `run_simulation`. אנא קראו את ההסבר הבא עד להבנה מלאה, וכן עברו על הקוד בקובץ המתאים.

הסבר הקוד בקצרה:

קבלת הפרמטרים הרלוונטיים לריצה הנוכחית.	<code>params = get_simulation_params(simulation_num)</code>
מייצרים אובייקט של הסימולציה בעזרת הפרמטרים הנוכחיים.	<code>mab = MABSimulation(**params)</code>
יצירת אובייקט אלגוריתם ההמלצה שלכם.	<code>planner = Planner(params['num_rounds'], params['phase_len'], params['num_arms'], params['num_users'], params['arms_thresh'], params['users_distribution'])</code>
הרצת הסימולציה בעזרת האלגוריתם וקבלת סכום הרווחים מכלל הסיבובים.	<code>reward = mab.simulation(planner)</code>

הסבר הקוד בפונקציה MABSimulation.simulation:

דגימת המשתמש שהגיע בסיבוב הנוכחי למערכת.	user_context = self.sample_user()
קריאה לאלגוריתם שלכם לבחירת יצרן התוכן שיוצג למשתמש.	chosen_arm = planner.choose_arm(user_context)
דגימת הרווח למשתמש מהיצרן שהומלץ.	reward = self.sample_reward(user_context, chosen_arm)
עדכון האלגוריתם שלכם בתוצאת ההמלצה.	planner.notify_outcome(reward)
עדכון כמות הפעמים שהומלץ אותו יצרן תוכן.	self.exposure_list[chosen_arm] += 1
בסיום כל חלון זמן בודקים אלו יצרני תוכן עוזבים את המערכת.	if (i + 1) % self.phase_len == 0 and with_deactivation: self.deactivate_arms()

שימו לב – אלגוריתם איטי מדי (שמשך סיבוב שלו הינו ארוך זמן מ- $TIME_CAP$) יקבל רווח כולל של 0.

מומלץ מאוד לקרוא ולהבין את הפונקציות MABSimulation.simulation ו-run_simulation!

יכולות אלגוריתם ההמלצה

כאמור, הקובץ היחיד שעליכם לשנות הינו הקובץ id_123456789_987654321.py. הקובץ כולל מחלקה בשם Planner בה תממשו את הקוד שלכם. על המחלקה לתמוך ב-4 פונקציות:

- **__init__** - מקבל את הפרמטרים אשר זמינים לאלגוריתם ובונה אובייקט של המחלקה. הפרמטרים הינם:
 - num_rounds – מספר שלם המייצג את כמות הסיבובים בסימולציה.
 - phase_len – מספר שלם המייצג את אורך חלון זמן בסימולציה.
 - num_arms – מספר שלם המייצג את כמות הזרועות (יצרני תוכן).
 - num_users – מספר שלם המייצג את כנות המשתמשים במערכת.
 - arms_thresh – מערך המכיל את סף ההמלצות שדורש כל יצרן בכל חלון זמן.
 - users_distribution – מערך המכיל את ההסתברות של כל משתמש להגיע אל המערכת.
- **choose_arm** - פונקציה אשר נקראת בכל סיבוב. מקבלת את המזהה של המשתמש שהגיע אל המערכת ומחזיר את המזהה של יצרן שנבחר עבורו (המקום לאלגוריתם שלכם).
 - user_context – מזהה המשתמש שהגיע אל המערכת.
- **notify_outcome** - פונקציה אשר נקראת בתום סיבוב ומקבלת:
 - reward – הרווח של המשתמש מהזרוע שהומלצה לו.
- **get_id** - פונקציה המחזירה מחרוזת עם תעודות הזהות שלכם.

המימוש שלכם

בקובץ id_123456789_987654321.py ישנן 4 הערות TODO שהן המשימות שלכם:

1. לשנות את שם הקובץ כך שיתאים לתעודות הזהות שלכם. כך למשל אם אליס ובוב מגישים עבודה ביחד ותעודת הזהות של אליס הינה "111" ושל בוב "222", עליהם להגיש קובץ בשם "id_111_222.py". אם צ'ארלי מגיש את העבודה לבד ותעודת הזהות שלו הינה "333", אז עליו להגיש את הקובץ בשם "id_333.py". שימו לב – הקפידו להשאיר את התחילית "id" בשם הקובץ.
2. מימוש choose_arm.
3. מימוש notify_outcome. אין חובה לשנות את הפונקציה הזאת. אם choose_arm שמימשתם אינה משתמשת בנתוני העבר, גם אין סיבה.
4. מימוש get_id. המחרוזת שהפונקציה מחזירה צריכה להיות זהה לשם הקובץ אותו אתם מגישים (כמובן ללא ".py").

שימו לב – אין לשנות שום דבר אחר בקובץ, כמו לדוגמא את שם ה-class.

מבנה הציון

הציון שלכם מבוסס על שתי משימות:

1. משימה בסיסית המהווה 80% מהציון. במשימה זו נריץ את האלגוריתם שלכם במספר רב של סימולציות על כל אחת **מקומבינציות הפרמטרים שקיבלתם**. במשימה זו אתם נמדדים על פי התועלת הכוללת הממוצעת אותה השגתם בתום התהליך. הציון במשימה זו יתחלק לשני חלקים:
 - סף – הגשה שתשיג תוצאה שתעבור את הסף שנפרסם בהמשך (בריצת הבדיקה שנבצע אצלנו) תזכה את המגישים בציון 70 לפחות בחלק זה.
 - תחרות – תקבלו נקודות (וגם נקודות בונוס) בהתאם לתוצאות שלכם ביחס לביצועים של האלגוריתמים האחרים שהוגשו.
2. משימה מתקדמת המהווה 20% מהציון. במשימה זו נריץ את האלגוריתם שלכם במספר רב של סימולציות על קומבינציות פרמטרים **שלא נחשף בפניכם**. במשימה זו אתם נמדדים על פי התועלת הכוללת הממוצעת אותה השגתם בתום התהליך. הציון במשימה זו יתחלק גם כן לשני חלקים:
 - סף - הגשה שתשיג תוצאה שתעבור את הסף שנקבע (בריצת הבדיקה שנבצע אצלנו) תזכה את המגישים בציון 50 לפחות בחלק זה.
 - תחרות – תקבלו נקודות (וגם נקודות בונוס) בהתאם לתוצאות שלכם ביחס לביצועים של האלגוריתמים האחרים שהוגשו.

טיפים

- שימו לב שהאלגוריתם שלכם עובר את דרישת יעילות זמן הריצה גם אם כמות המשתמשים / יצרנים גדולה מ 3 (יכולה להיות לכל היותר 6).
- אם האלגוריתם שלכם עבר באופן מובהק את משימה 1, שמתם לב למקרי קצה והקשבתם לטיפ הראשון, אז אין סיבה שלא תעברו גם את משימה 2.

- מומלץ מאוד (אך לא חובה לפתירת התרגיל) לקרוא על contextual multi-armed bandits. שימו לב שניתן להתייחס למשתמשים כ-contexts וליצרני התוכן כ-arms.
- שימו לב שייתכן שיגיע משתמש שמעדיף את יצרן i אבל עדיף להמליץ לו על יצרן j , כיוון שיצרן j לא קיבל מספיק צפיות בסיבובים האחרונים ואם הוא יעזוב זה יפגע בתוחלת סכום הרווחים הכללי.
- הקפידו לא להמליץ בטעות על זרועות שכבר עזבו את המערכת.
- ודאו שהקוד שלכם לא קורס, בפרט במקרי קצה. לדוגמא, ודאו שהקוד תקין כאשר $\text{num_arms} > \text{num_users}$ וגם כאשר $\text{num_arms} \leq \text{num_users}$. הגשה בה הקוד קורס תיפסל ותוביל לציון 0.

הוראות נוספות

- ודאו כי שיניתם את שם הקובץ `id_123456789_987654321.py` לפי ההוראות מעלה.
- אין להשתמש בספריות נוספות מלבד אלו הכתובות בקובץ וספריות בסיסיות של פייתון (`math`, `itertools` וכו'). אם אינכם בטוחים שספריה היא חוקית, תשאלו.
- רצוי לעבור על כלל הקובץ `simulation`.

הוראות הגשה

יש להגיש שני קבצים בתוך תיקיית `zip`:

- קובץ אלגוריתם ההמלצה שלכם. ודאו כי עקבתם אחר ההוראות המופיעות ב-"יכולות אלגוריתם ההמלצה". חריגה תהווה פסילה וציון 0.
- קובץ pdf המתאר מילולית את המימוש שלכם עבור האלגוריתם. אין להגיש יותר מעמוד אחד. אם המימוש שלכם פשוט מאוד (כמו זה שהקובץ הגיע איתו), גם משפט אחד יכול להספיק.

בהצלחה !