מערכות המלצה, תרגיל מספר 1

Rating Prediction

כללי: בתרגיל זה נממש שיטות שונות לחיזוי rating. במהלך התרגיל נריץ ניסויים על מאגר דירוגים של דירוגי עסקים על ידי משתמשים. רצוי מאוד לכתוב קוד מודולרי ככל האפשר, מאחר והתרגילים הבאים יתבססו על הקוד שכתבתם בתרגיל זה. הינכם רשאים לממש את התרגיל בכל דרך שתבחרו, תוך הקפדה על הממשק המוגדר במחלקה RecommenderSystem.

1. הורידו את בסיס הנתונים מאתר kaggle -

<https://www.kaggle.com/c/yelp-recsys-2013/data>?

המכיל הצבעות של משתמשים על עסקים (אנו מעונינים בעיקר ב-yelp\_training\_set). חלצו את הארכיון yelp\_training\_set. נתמקד כרגע בקובץ yelp\_training\_set\_review.json. הקובץ מכיל שורות, כאשר בכל שורה אנו מעונינים בשדות user\_id,business\_id,stars המכילים מידע על משתמש שנתן דירוג לעסק מסוים.

1. ממשו את השיטה Load הטוענת את המידע מהקובץ לבסיס נתונים, ומחלקת את ההצבעות רנדומלית לשני חלקים: train, test ע"פ החלק הנקבע בפרמטר השני לשיטה. רצוי לממש את בסיס הנתונים באופן יעיל כפי יכולתכם על מנת לסייע במימוש השלבים הנוספים. ממשו חלוקה של מסד הנתונים לשני חלקים זרים (כלומר, אף דירוג לא יופיע בשני החלקים) באופן הבא:
   1. נתון p חלקו של ה-training set מתוך כל המידע (לדוגמא, p=0.95).
   2. צרו שני מסדי נתונים ריקים: train ו-test.
   3. בחרו משתמש אקראי u שטרם בחרתם, בחרו מספר אקראי k של דירוגים. לדוגמא, אם המשתמש דירג 100 סרטים נבחר k∈[0,100].
   4. בחרו k דירוגים אקראיים של u והוסיפו אותם ל-train. את שאר הדירוגים של u הוסיפו ל-test.
   5. המשיכו באופן זה עד (חזרו על c,d) עד שיש ב-test מספיק דירוגים (1-p מתוך כל הדירוגים).
   6. את שאר המשתמשים הוסיפו ל-train.
2. ממשו את השיטה GetRating במחלקה RecommenderSystem – השיטה מחפשת בבסיס הנתונים את הדירוג (הקיים, ללא תחזיות) שנתן משתמש לפריט (פרופיל). במידה והמשתמש לא נתן לפריט דירוג התנהגות השיטה אינה מוגדרת.
3. ממשו מערכת תחזיות החוזה ציונים לסרטים על ידי ציוני המשתמשים הדומים לו:
   1. Pearson Correlation – השתמשו בנוסחת קורולצית פירסון על מנת לחשב דומות בין משתמשים. חזו את ציוני הסרטים באמצעות ממוצע משוקלל של ציוני המשתמשים הדומים. שימו לב לא לחשב דומות בין המשתמש לעצמו, כיוון שזה יכול ליצור סטיות גדולות בחישובים. יש לבצע התיחסות מיוחדת למשקלות שליליים, כפי שנלמד בכיתה.
   2. Cosine Similarity – חזרו על הסעיף הקודם, כאשר חישוב הדומות בין משתמשים נעשה על ידי מדידת הזוית בין הוקטורים.
   3. עליכם למממש את השיטה PredictRating המקבלת שיטה, משתמש, ופריט, וחוזה את הדירוג של הפריט על ידי המשתמש באמצעות השיטה המתאימה. ערכים אפשריים לשם השיטה –Pearson, Cosine.
4. Random – ממשו מערכת החוזה ציונים באופן רנדומלי, על ידי בחירת ציון מתוך התפלגות הציונים של המשתמש (זה לא מסובך כמו שזה נשמע). הוסיפו לשיטה PredictRating את השיטה תחת השם Random.
5. ממשו את מודל הבסיס שנידון בכיתה. עליכם לממש את השיטה TrainBaseModel במחלקה Recommender System המקבלת את מספר ה-latent features (גודל הווקטורים p,q) ומאמנת מודל בסיס. להזכירכם, בהינתן מספר הפיצ'רים:
   1. חלקו את ה-dataset ל-train ול-validation (ישנו חלק נוסף – test – בו נשתמש מאוחר יותר לבדיקת איכות). את כל האימון נבצע בהנתן ה-train בלבד.
   2. חשבו את μ – ממוצע הדירוגים.
   3. אתחלו את bu,bi,pu,qi – באופן רנדומלי (לערכים קטנים מאוד חיוביים ושליליים).
   4. עברו על הדירוגים ru,i שב-training set ובצעו לכל דירוג
      1. eu,i=ru,i-μ-bi-bu-pu⋅qi
      2. עדכנו את הפרמטרים:
         * 1. bu=bu+γ ⋅ (eui-λ ⋅ bu)
           2. bi=bi+γ ⋅ (eui-λ ⋅ bi)
           3. qi=qi+γ ⋅ (eui ⋅ pu - λ ⋅ qi)
           4. pu=pu+γ ⋅ (eui ⋅ qi - λ ⋅ pu)
      3. רצוי להשתמש בפרמטרים קטנים עבור γ ו-λ (באיזור ה-0.05).
   5. חשבו את הטעות (RMSE) על גבי ה-validation set. נעצור כאשר הטעות גדלה מהאיטרציה הקודמת. אם הטעות קטנה – חזרו על d.
6. ממשו השוואה בין אלגוריתמי החיזוי. כדי לקבל ציון עבור אלגוריתם חיזוי נשתמש בפרוטוקול הבא:
   1. נתונה חלוקה של בסיס הנתונים ל-train ול-test כפי שהוסבר בסעיף הקודם.
   2. עבור כל דירוג ru,i ב-test, חשבו תחזית לפריט i עבור משתמש u באמצעות כל אחד מהאלגוריתמים על גבי ה-training set בלבד.
   3. חשבו את הטעות eu,i=ru,i-r̂u,i
   4. חשבו את שורש הטעות הממוצעת
   5. ממשו את השיטה ComputeRMSE המקבלת רשימה של שמות שיטותוהחזירו מילון ובו לכל שיטה רשום ה-RMSE שלה.
7. ממשו את מבחן הסימן (sign test) להערכת הביטחון (confidence) בתוצאת ההשוואה:
   1. סיפרו עבור כל שני אלגוריתמים כמה פעמים הצליח האחד יותר מהשני (קיבל טעות קטנה יותר). במקרים של שוויון (שני האלגוריתמים הצליחו באותה מידה) נעניק חצי נקודה לכל אלגוריתם.
   2. נניח כי אלגוריתם A "ניצח" יותר פעמים מאלגוריתם B. יהי nA מספר הפעמים בהם ניצח A את B ו-nB מספר הפעמים בהם ניצח B את A. נחשב:
      1. ההסתברות ש-A טוב יותר מ-B היא *1 - p(A not better than B).*
8. הוסיפו לשיטה ComputeRMSE,, פרמטר dConfidence, המחזיר את הבטחון בתוצאה ע"פ מבחן הסימן.

הגשה: עליכם להגיש עבור תרגיל זה את קבצי הקוד. את הקבצים יש לכווץ באמצעות תוכנת zip ולהגיש לשרת ה-ftp (כפי שיפורסם בהודעה בהמשך) לא יאוחר מיום ראשון, ה-19 לאפריל ב-11:59 בלילה.

את העבודה ניתן לעשות בזוגות. אל תעתיקו!

שאלות, בעיות, טענות ומענות – [shanigu@bgu.ac.il](mailto:shanigu@bgu.ac.il) בהצלחה!