

מערכת המלצות Recommendation System

- הבעיה • יש מסד נתונים מאוד גדול
- צריך להמליץ לכל משתמש על פריט, ככה שהמשתמשים יהיו כמה שיותר מרוצים (מה זה מרוצים?)
- מה אנחנו יודעים על המשתמשים?

דרך מקובלת היא לא להגיד כן/לא על פריט, אלא לתת דירוג. אם המשתמש יכול לדרג פריטים, נרצה להשתמש בזה בשביל ללמוד עליו ולהעריך איזה דירוג יתן לפריטים.
יש 3 גישות, ובדרך כלל נרצה לשלב כי לכל גישה יש חסרונות:

1. Collaborative - בדרך כלל, חוץ מהאדם הספציפי שאליו נרצה להמליץ, יש למערכת את הדירוגים שהוזנו מעוד הרבה אנשים - ואפשר להשתמש בזה. כאן חשוב לדעת איזה משתמשים אחרים לבחור בשביל לקחת את הדירוגים שלהם.

- יתרונות: • לא צריך שום ידע על הdomain.
- חסרונות: • צריך feedback מהמשתמשים
- אי אפשר לשרת בצורה טובה משתמשים חדשים (cold start) - הם צריכים לדרג דברים בשביל שנוכל ללמוד עליהם
- אי אפשר לשרת בצורה טובה פריטים חדשים - משתמשים צריכים לדרג אותם (וזה אומר שצריך להמליץ עליהם עוד לפני שלמדנו אותם)

2. Content Based - בנוסף לאנשים, יש מידע גם על הפריטים. למשל, אם זה סרט - מתי הוא יצא? מה הז'אנר? מי השחקנים?

- יתרונות: • אין צורך בקהילה
- אפשר להשוות בין פריטים
- חסרונות: • צריך לבנות metadata של הפריטים
- cold start - משתמשים חדשים צריכים להתחיל למלא טפסים כדי שנדע מה הם רוצים

3. Knowledge Based - שימוש במודלים וחוקים שמישהו כתב (אם המשתמש הוא X אז תמליץ לו Y)

- יתרונות: • אין בעייה של cold start - עבור מערכת חדשה בלי נתונים לא על המשתמשים ולא על הפריטים זה מאוד מוצלח
- דטרמיניסטי
- חסרונות: • מאוד קשה והרבה עבודה לכתוב את החוקים

Collaborative Filtering(CF)

ההנחה המרכזית - המשתמשים נותנים feedback על הפריטים. או בצורה אקטיבית(דירוג) או בצורה implicit(it קונים/לא קונים). צריך לשים לב שלפעמים המשתמשים משנים את הטעם שלהם, אז צריך להיזהר עם הסתכלות על מידע ישן.

יש Collaborative Filtering שתי שיטות מרכזיות:

1. User-based nearest-neighbor - הולכים לפי המשתמש

2. Item-based - הולכים לפי הפריטים

תמיד יש מטריצה(בדרך כלל מאוד דלילה) של הדירוג שכל משתמש נתן לכל פריט, ונרצה להשלים את הדירוג של משתמש מסויים עבור פריט מסויים שחסר בטבלה.

שיטת user-based

שלב ראשון - בחירת משתמשים דומים

נרצה למצוא משתמשים שדומים למשתמש שלנו, ולראות איך הם דירגו את הפריט המדובר. בשביל זה צריך user similarity function. פונקציה פופולרית היא Pearson Correlation:

$$\text{sim}(a, b) = \frac{\sum_{p \in P} (r_{a,p} - \bar{r}_a)(r_{b,p} - \bar{r}_b)}{\sqrt{\sum_{p \in P} (r_{a,p} - \bar{r}_a)^2} \cdot \sqrt{\sum_{p \in P} (r_{b,p} - \bar{r}_b)^2}}$$

כאשר: a, b המשתמשים

P קבוצת כל הפריטים שדורגו גם ע"י a וגם ע"י b

$r_{a,p}$ הדירוג ש a נתן ל p

\bar{r}_a, \bar{r}_b הדירוג הממוצע ש a ו b דירגו את הפריטים(רק על P ? על כל הפריטים? בדרך כלל לוקחים את כל הפריטים, אבל יש ווריאציות)

נשים: ♥ • מנרמלים כדי לקבל ערכים בטווח $[-1, 1]$

• בגלל שלאנשים שונים יש סקאלות שונות, לא בודקים את הדירוג ישירות, אלא משווים לממוצע של המשתמש

השיטה הזו מאוד נפוצה - אבל בעיקרון אפשר להשתמש בכל פונקציית מרחק בין ווקטורים(מרחק אוקלידי, מרחק מנהטן, ...)

שלב שני - חיזוי הדירוג

לאחר שבחרנו קבוצה N (בגודל קבוע) משתמשים הכי דומים, נרצה לחזות את הדירוג שהמשתמש שלנו יתן לפי הדירוג שהמשתמשים הדומים נתנו. נרצה לנרמל ולשקלל לפי הדומות:

$$\text{pred}(a, p) = \bar{r}_a + \frac{\sum_{b \in N} \text{sim}(a, b) \cdot (r_{b,p} - \bar{r}_b)}{\sum_{b \in N} \text{sim}(a, b)}$$

בעיות user based

- זה לוקח הרבה זמן
- אי אפשר לחשב דברים מראש

שיטת item-based

הרעיון - לבדוק איך המשתמש שלנו דירג פריטים דומים לפריט שאותו נרצה לחזות.

שלב ראשון - בחירת פריטים דומים

שוב - אפשר כל פונקציית מרחק, אבל גם כאן מקובל להשתמש ב-Pearson Correlation:

$$\text{sim}(i, j) = \frac{\sum_{u \in U} (r_{u,i} - \bar{r}_u)(r_{u,j} - \bar{r}_u)}{\sqrt{\sum_{u \in U} (r_{u,i} - \bar{r}_u)^2} \cdot \sqrt{\sum_{u \in U} (r_{u,j} - \bar{r}_u)^2}}$$

- נשים ♡: מסתכלים רק על פריטים שהמשתמש שלנו דירג(אחרת אנחנו לא יכולים להשתמש בהם בשביל לחזות את הדירוג)
- משתמשים בממוצע של המשתמש, לא בממוצע של הפריט. החיסור הוא כדי לנרמל את הסקאלה של המשתמש - לפריט אין סקאלה כזו.

שלב שני - חיזוי הדירוג

לאחר שיש לנו J פריטים הכי דומים(לוקחים רק את ה- k הכי דומים כדי להימנע מרעש):

$$\text{pred}(u, i) = \frac{\sum_{j \in J} r_{u,j} \cdot \text{sim}(i, j)}{\sum_{j \in J} \text{sim}(i, j)}$$

יתרונות

- אפשר לחשב מראש את הדומות בין הפריטים