מאמר זה פורסם בכתב העת "חידושים בניהול", גיליון מס' 3, דצמבר 2018,
היוצא לאור על ידי הפקולטה לניהול על שם קולר באוניברסיטת תל אביב.

מערכת המלצות המבוססת על סינון שיתופי

יעקב זהבי

יעקב זהבי, פרופסור אמריטוס בפקולטה לניהול על שם קולר באוניברסיטת תל אביב. הוא אחד מפורצי הדרך בתחום כריית המידע (Data Mining) בעולם נתוני העתק, תחום שהוא מעורב בו במספר חזיתות – מחקר, הוראה, פיתוח תוכנה ויישומים לקבלת החלטות. המערכת לקבלת החלטות בתחום השיווק שפרופ' זהבי פיתח בסוף שנות השמונים תוך שימוש בכלים אנליטיים מתקדמים, שלימים נקראו "כריית מידע", נחשבת לאחת מהמערכות הראשונות שפותחו מעולם בתחום זה, אם לא הראשונה שבהן. בשנים הבאות המשיך פרופ' זהבי לעסוק במחקר ובפיתוח של אלגוריתמים מתקדמים בתחום כריית מידע לשם שימושים מגוונים בקבלת החלטות, נושאים שהוא ממשיך לעסוק בהם עד היום. פרופ' זהבי זכה פעמיים רצופות במדליית הזהב בתחרות השנתית לגילוי ידע (Knowledge Discovery) שמאורגנת על ידי ACM – American Computation Machinery. מספר מאמרים שלו בתחום זה זכו בפרסים על מצוינות אקדמית.

תקציר

המעמד המוביל של חברת אמזון בתחום המוצרים והשירותים ושל חברת נטפליקס בתחום הסרטים הביאו את הנושא של מערכות המלצה לקדמת הבמה והפכו אותו לאחד היישומים הבולטים והמשפיעים ביותר של כריית מידע. מאמר זה עוסק בשימוש בגישות "סינון שיתופי" (CF – Collaborative Filtering) לצורך התאמה אישית (personalization) של מוצרים ושירותים לצרכנים וכן לתמיכה בבעיות של מכירה צולבת. מבין שלל השיטות של CF אנו דנים ב-2 שיטות עיקריות – שיטת השכן הקרוב ביותר (KNN) ושיטת K-ממוצעים (KM). שיטה נוספת היא באמצעות חוקים אסוציאטיביים, הישימה למשתמשים שקנו רק מוצר אחד או שניים בעבר. בנוסף, אנו סוקרים בקצרה גם את גישת ה-Matrix-Factorization, שמשמשת לצורך מתן המלצות המסתמכת על דירוגים של משתמשים. המאמר מציג גם תוצאות מחקר שהתבסס על נתונים אמיתיים והראה שמערכת המלצות מבוססת CF מניבה המלצות קנייה עם שיעורי התאמה גבוהים משמעותית מהמלצות אקראיות. להשלמת היריעה, אנו דנים גם במספר יישומים פוטנציאליים של גישות CF בתחומים שונים.

מילות מפתח: מערכת המלצות, מכירה צולבת, סינון שיתופי, התאמה אישית, כריית מידע

הקדמה

סוגיית ה"המכירה הצולבת" (Cross-Selling) היא אחת הסוגיות המרכזיות בתחום השיווק: מהם הפריטים שיש להמליץ עליהם לצרכן שכבר רכש מוצר או שירות כדי למקסם את הרווח ולהגדיל ROI. סוגיה זו מתייחסת לא רק למוצרים (ספרים, מכשירי חשמל, מכוניות, מוצרי אופנה ועוד) אלא גם לשירותים (תוכניות ביטוח, מוצרים בנקאיים, מסעדות וכדומה) וגם למוצרי תוכן (חדשות, מסמכים, מאמרים, דוחות, סרטי וידאו, מוסיקה, שירים וכיוצא בהם). ניתן גם לשדרג מכירה צולבת באמצעות מוצרים משלימים (Up-Selling) - היינו להציע לצרכן מוצרים משלימים (או נלווים), למשל חפיצים (gadgets) לבית או למחשב, מוצרים היקפיים (peripheral) למחשב, כיסויים נוספים לפוליסות ביטוח, ועוד. השאלה המתבקשת: בעולם שבו הצרכנים שונים כל כך זה מזה, איזה מוצר, שירות או דבר תוכן להציע לכל צרכן כך שיתאים להעדפותיו ולרצונותיו. בעיה זו נעשית סבוכה עוד יותר עקב המגוון הגדול של מוצרים, שירותים ומוצרי תוכן שקיים כיום בשוק ולנוכח התחרות הגוברת בשוק.

ההתקדמות הטכנולוגית בתחום מערכות המידע והמחשבים ובתחומים של כריית מידע (Data Mining), בינה מלאכותית (AI – Artificial Intelligence) ולמידה חישובית (Machine Learning) מאפשרים היום להתייחס לבעיית המכירה הצולבת באמצעות מנגנונים של שיתוף ידע הלקוחים מתחום ה"סינון השיתופי" (CF – Collaborative Filtering). במקור, גישות CF, מושג שמקורו מתחום הסוציולוגיה, הן גישות של שיתוף מידע בין אנשים. לדוגמה, אם אנחנו מחפשים מידע על דרכי פעולה - איזה ספר לקרוא? לאיזה סרט קולנוע כדאי ללכת? מהי המסעדה המומלצת? אנחנו בדרך כלל מתייעצים עם קבוצת אנשים שיש לנו עניין משותף איתם (משפחה, חברים, קולגות לעבודה) ובוחרים את המוצר/שירות על פי ההמלצות שאנחנו מקבלים. אם הנשאלים אינם יודעים את התשובה, הם יכולים להפנות אותנו לאנשים אחרים שיוכלו לעזור לנו. כלומר, אנשים מסתמכים על קבוצת המכרים הקרובים אליהם כדי לסנן מידע ולבחור בדרך הפעולה המועדפת, ומכאן השם CF - "סינון שיתופי". למעשה, מדובר כאן בשיטה של "פה לאוזן", המשמשת כבר אלפי שנים דרך אפקטיבית לתקשורת בין אנשים.

בהקשר של בעיית המכירה הצולבת, גישות CF מאפשרות להמליץ לצרכנים על מוצרים ושירותים בהתבסס על שיתוף מידע ביניהם. אבל לנוכח הגודל והמגוון של קהל הצרכנים וריבוי המוצרים והשירותים בשוק, השיטות המסורתיות של החלפת מידע באמצעות אינטראקציה בין-אישית אינן תקפות. אין זה מעשי לפנות למיליוני אנשים כדי לשאול אותם על מוצרים ושירותים שקנו בעבר ועל ההעדפות שלהם, לא כל שכן לנתח את הנתונים. ככל שאנשים הם אינטליגנטים, יכולתם לאחסן ולעבד מידע היא מוגבלת.

גישות CF מודרניות עוקפות את המגבלות האנושיות האלה בכך שהן מאפשרות לאחסן את המידע על קניות של מוצרים ושירותים מחוץ למוח האנושי ולנתח את הכמות העצומה של הנתונים באמצעות מחשבים עוצמתיים וכלים אנליטיים מתקדמים. ההתפתחות המהירה של האינטרנט מאפשרת היום ל"היפנש", באופן וירטואלי ולהחליף מידע בין מיליוני לקוחות בעלי עניין משותף. יתרה מזו, בניגוד לגישות CF מסורתיות שמתבססות על תשאול של אנשים, גישות CF מודרניות מתבססות על ההתנהגות האמיתית של

הצרכנים כפי שהיא משתקפת באמצעות העסקאות שהם מבצעים בפועל (קניות, הפניות, הורדה של מוצרי תוכן), מה שמקנה להן יותר אמינות. זאת, הן בשל העובדה שאנשים לא בהכרח אומרים את האמת כשהם מתבקשים לענות על שאלות או להשתתף בסקרים, והן משום שלעיתים קשה להעריך מוצרים ושירותים בעלי תכונות ומאפיינים רבים מאוד.

כיום, גישות CF הן השיטות המובילות בקבלת החלטות בנושא של מכירות צולבות והן הבסיס של מרבית מנועי ההמלצה. שתי הגישות המובילות בבניית מערכות המלצה הן:

- מבוססות-פריטים (item-based) – המלצות הרכישה מבוססות על התנהגות המשתמשים כפי שהיא באה לידי ביטוי ברכישותיהם בפועל (implicit).

- מבוססות-משתמש (user-based) – המלצות הרכישה מבוססות על דירוגים שהמשתמשים נותנים לפריטים השונים (explicit).

שתי החברות ה"חלוצות" שפיתחו מערכות מסחריות להמלצות הן אמזון ונטפליקס. חברת אמזון, שהייתה הראשונה בתחום, השתמשה באלגוריתם מבוסס-פריטים כדי להציע לאנשים שמתעניינים בקניית מוצר מסוים לרכוש מוצרים נוספים, וזאת על סמך קניות של לקוחות שקנו בעבר את המוצר דנן. נניח שמדובר באדם שמתעניין בקנייה של מוצר A (למשל, ספר מסדרת הארי פוטר); האלגוריתם של אמזון יציע לצרכן זה ספרים מתוך רשימת הספרים שקנו כל הלקוחות בבסיס הנתונים שגם הם קנו בעבר את מוצר A, לדוגמה ספרים אחרים מתוך הסדרה של הארי פוטר או ספרים על נושאים דומים. הכלל פשוט למדי: "אנשים שקנו מוצר A רכשו בעבר גם את המוצרים B,C, D,..." ולכן רשימת ההמלצות ללקוח שמתעניין בקניית מוצר A נבחרת מתוך קבוצת המוצרים B,C,D,... בראשית הדרך יישמה אמזון את האלגוריתם על ספרים ופריטי מוזיקה, אך מאז החילה אותו גם על מגוון גדול של מוצרים אחרים. בד בבד היא שכללה גם את מנגנון ההמלצות.

חברת נטפליקס היא החברה המובילה בתחום של יישום גישות מבוססות-משתמש להמלצות רכישה. כאן עסקינן במערכת המלצות לסרטי וידאו שמתבססת לא רק על התנהגות הצרכנים כפי שהיא משתקפת מהסרטים שבהם צפו בפועל, אלא גם על הדירוגים שהצרכנים נתנו לסרטים אלה. מערכת ההמלצות של נטפליקס עלתה לכותרות בשנים האחרונות בגין תחרות עולמית שיזמה החברה לפתח אלגוריתם שישפר את יעילות מערכת ההמלצות שלה ויעלה את ה ROI של החברה.

במאמר זה נתמקד בעיקר במערכת המלצות מבוססת-פריטים מהסוג שבו משתמשת חברת אמזון (שנקראת גם pure CF), המלצות שמשקפות את העדפות הצרכנים באמצעות הפעולות האמיתיות שהם נוקטים בשטח (למשל, קניות של מוצרים). להשלמת התמונה נסקור בקצרה גם את המערכת מבוססת-המשתמש של חברת נטפליקס המתבססת על דירוגים שהמשתמשים נתנו לסרטי וידאו שבהם צפו.

להלן נשתמש במושגים "מוצרים" (products) ו"משתמשים" (users) בתור מושגים גנריים. לפי סוג היישום, "מוצר" יכול להיות מוצר אמיתי, פריט מוזיקה, סרט וידאו, מוצר תוכן, דף אינטרנט ועוד. "משתמש" מייצג צרכנים בפועל, מבקרים באינטרנט, לקוחות פוטנציאליים ("prospects") ואחרים. בהינתן המושגים האלה, למערכת המלצות יש מספר ממדים:

איזה מוצר להציע לכל משתמש?
מיהם המשתמשים ה"דומים" ביותר לכל משתמש?
לאילו משתמשים נציע כל מוצר?
מהם המוצרים ה"דומים" ביותר לכל מוצר?

במאמר זה נתרכז בבעיית המכירה הצולבת, היינו איזה מוצר להציע לכל משתמש? המטרה היא לתפור הצעה מותאמת אישית לכל משתמש כדי להעלות את שיעורי התגובה. מידת ההתאמה האישית שונה מאלגוריתם לאלגוריתם ואנו נדון בהבדלים ביניהם.

נציין שגישות CF זוכות לאחרונה לעדנה הודות לאינטרנט, אבל גישות אלה אינן מוגבלות רק לתחום האינטרנט אלא ניתן ליישם אותן גם על בעיות המלצה offline. כיום, מערכות המלצות מבוססות CF אינן "מונופול" של אמזון ונטפליקס, והן נפוצות גם בקרב מגוון של חברות העוסקות בשיווק של מוצרים ושירותים.

המלצות מבוססות-פריטים
ניתוח אשכולות (Cluster Analysis)

מבחינה אנליטית, בעיית המכירה הצולבת שייכת לתחום בכריית מידע הידוע בשם ניתוח אשכולות (Cluster Analysis), שהוא אחד האלגוריתם המרכזיים של למידה בלתי מונחית (Unsupervised Learning). נניח n משתמשים שקנו יחד אוסף של J מוצרים. ניתן לייצג כל משתמש בבסיס הנתונים כווקטור במרחב ה-J ממדי, שבו כל אלמנט מקבל ערך 1 אם המשתמש קנה את המוצר המתאים, ואם לא - 0. בצורה זו אנו מייצגים את בסיס נתוני המשתמשים כדיאגרמת פיזור (Scatter Diagram) שמכילה n נקודות (משתמשים) במרחב ה-J ממדי (מוצרים). כך משתמשים שיש להם מאפייני קניות דומים מתקבצים יחד ל"עננים" וניתן להשתמש בניתוח אשכולות כדי לאפיין את ה"עננים" האלה.

דרך חלופית לייצג את קניות המשתמשים במקום להשתמש בערכים בינריים 0/1 היא באמצעות מספר הפריטים שכל משתמש קנה.

גישות של ניתוח אשכולות מקבצות עצמים (objects) לאשכולות (clusters) על סמך מידת הקרבה (proximity) או הדמיון (similarity) של המאפיינים (attributes) שלהם. מידת הקרבה בין העצמים נמדדת באמצעות מדדי מרחק (distance). המטרה היא לחלק את העצמים המצויים בבסיס הנתונים לאשכולות "הומוגניים" כך שהעצמים בתוך כל אשכול יהיו דומים זה לזה (למשל, יש להם מאפייני קנייה דומים) ואילו העצמים הנמצאים באשכולות שונים יהיו שונים זה מזה. במקרה שלנו, העצמים הם המשתמשים בבסיס הנתונים והמאפיינים שלהם הם המוצרים שהם רכשו.

נסמן:

$i, i = 1, 2, \cdots, n$ - אינדקס המשתמש

$j, j = 1, 2, \cdots, J$ - אינדקס המוצר

C_{ij} - מספר בינרי המקבל את הערך 1 אם משתמש i קנה מוצר j בעבר, ואם לא – 0. כעת נוכל לתאר את פרופיל הקניות של משתמש i באמצעות הווקטור ה-J-ממדי:

$$C_i = (C_{i1}, C_{i2}, \cdots, C_{ij})$$

כאמור, מידת הדמיון בין המשתמשים נמדדת באמצעות מדדי מרחק. המדד הנפוץ ביותר הוא המרחק האוקלידי (euclidean distance). נניח ש- ℓ ו–m הם 2 משתמשים מתוך בסיס נתונים המונה n משתמשים; המרחק האוקלידי ביניהם מוגדר על ידי:

$$\text{distance}(C_\ell, C_m) = \sqrt{\sum_j (C_{\ell j} - C_{m j})^2}$$

ככל שהמרחק האוקלידי קטן יותר, נאמר שהמשתמשים דומים יותר זה לזה, ולהפך. כאשר 2 המשתמשים זהים (היינו, יש להם בדיוק אותו פרופיל קניות), המרחק האוקלידי ביניהם הוא 0.

כדאי לציין שבמקרה הדו-ממדי נוסחה זו מצטמצמת למשפט פיתגורס המפורסם עבור המרחק בין 2 נקודות במישור.

מדד חלופי, שמתאים יותר למאפיינים בינריים או שלמים (integers), הוא מרחק הקוסינוס (cosine distance) שמוגדר על ידי:

$$\text{distance}(C_\ell, C_m) = \frac{\sum_{j=1}^{J} C_{\ell j}\, C_{m j}}{\sqrt{\sum_{j=1}^{J} C_{\ell j}^2 \sum_{j=1}^{J} C_{m j}^2}}$$

בניגוד למרחק האוקלידי, כאן אם 2 המשתמשים זהים, מרחק הקוסינוס מקבל את הערך 1 ואם הם שונים לחלוטין ("אורתוגונליים"), המדד מקבל את הערך 0.

נציין שבמערכת המלצות, משתמשים בדרך כלל אינם מעוניינים לקבל המלצות על מוצרים פופולריים ושכיחים שהם מכירים אלא רק המלצות על מוצרים "מעניינים" (interesting) שהם אינם מודעים אליהם. לדוגמה, אם אנחנו ממליצים על מוזיקת פופ, לא נרצה לכלול המלצות על מוזיקה של החיפושיות היות שרוב המשתמשים מכירים את המוזיקה הזו והמלצה עליה הופכת אותה להמלצה "לא מעניינת". ניתן להימנע ממכשלה זאת על ידי הקטנת החשיבות של מוצרים שכיחים באלגוריתם של ניתוח

אשכולות, בכך שנצמיד לכל מוצר משקל שהוא פונקציה יורדת של הפופולריות שלו. דוגמה למשקל כזה היא (Herz et al., 1997):

$$W_j = \left(1/b_j\right)$$

כאשר b_j מבטא את רמת הפופולריות של מוצר j , למשל את מספר הקניות הכולל של מוצר j בפרק זמן מסוים בעבר. ככל שהמספר הזה גדול יותר, המוצר פופולרי יותר והמשקל שלו קטן יותר.

מספר אלגוריתמים לניתוח אשכולות מופיעים בספרות, וביניהם נציין את K-Means Algorithm (Fukunaga, 1990), Expectation Maximization (Lauritzen, 1995), Linkage-based methods (Bock, 1974), Kernel Density estimation (Silverman, 1986), רשתות עצביות (Kohonen et al., 1991) ואחרים. לאחרונה פותחו גם וריאציות שונות של האלגוריתמים לטיפול בבעיית הסלמה (scalability) עבור בסיסי נתונים גדולים, למשל BIRCH algorithm (Zhang et al., 1996), ואחרים.

במאמר זה נתמקד באלגוריתם ה-K-ממוצעים, KM (K-Means Algorithm) שהוא האלגוריתם הנפוץ ביותר לניתוח אשכולות, ונשווה אותו לאלגוריתם ה-KNN (K-Nearest Neighbour).

נציין ששני האלגוריתמים האלה תקפים עבור משתמשים שעשו בעבר יותר משתי קניות. אשר למשתמשים שעשו בעבר רק קנייה אחת או שתיים, אין מספיק מידע כדי לענות על השאלה לאיזה אשכול שייך כל אחד מהם. במקרה זה נפעיל אלגוריתם המבוסס על חוקים אסוציאטיביים (Association Rules) כדי להמליץ על המוצרים למשתמשים אלה.

להלן, נדון בקצרה בשלושת האלגוריתמים האלה.

אלגוריתם KM (K-Means Algorithm)

אלגוריתם KM מחלק את התצפיות בבסיס הנתונים (במקרה שלנו, המשתמשים) ל-K אשכולות, כאשר מספר האשכולות K נקבע מראש.

נסמן ב- S_k את נקודת המרכז (centroid) של אשכול k. S_k הוא וקטור עם J ממדים (או קואורדינטות), קואורדינטה אחת לכל מאפיין (במקרה שלנו, מוצר j). כל אחת מהקואורדינטות של מרכז האשכול מתקבלת באמצעות הממוצע של כל הקואורדינטות של המשתמשים השייכים לאשכול (ומכאן השם K-Means).

אלגוריתם KM מוצא לאיזה אשכול שייך כל משתמש בצורה איטרטיבית, כדלקמן:

צעד 1: אתחול - בחר K נקודות מרכז (centroids) התחלתיות, $S_k, k = 1,\ldots,K$ (אחד לכל אחד מ-K האשכולות). קיימות מספר שיטות לבחור את נקודות המרכז. הדרך הפשוטה ביותר היא בחירה אקראית.

צעד 2 : עבור (loop) על כל המשתמשים בבסיס הנתונים. עבור כל משתמש מצא את המרחק של פרופיל הקניות שלו מכל אחד מהצנטרואידים, על בסיס פונקציית המרחק שנבחרה, ושייך כל משתמש לאשכול הקרוב ביותר .

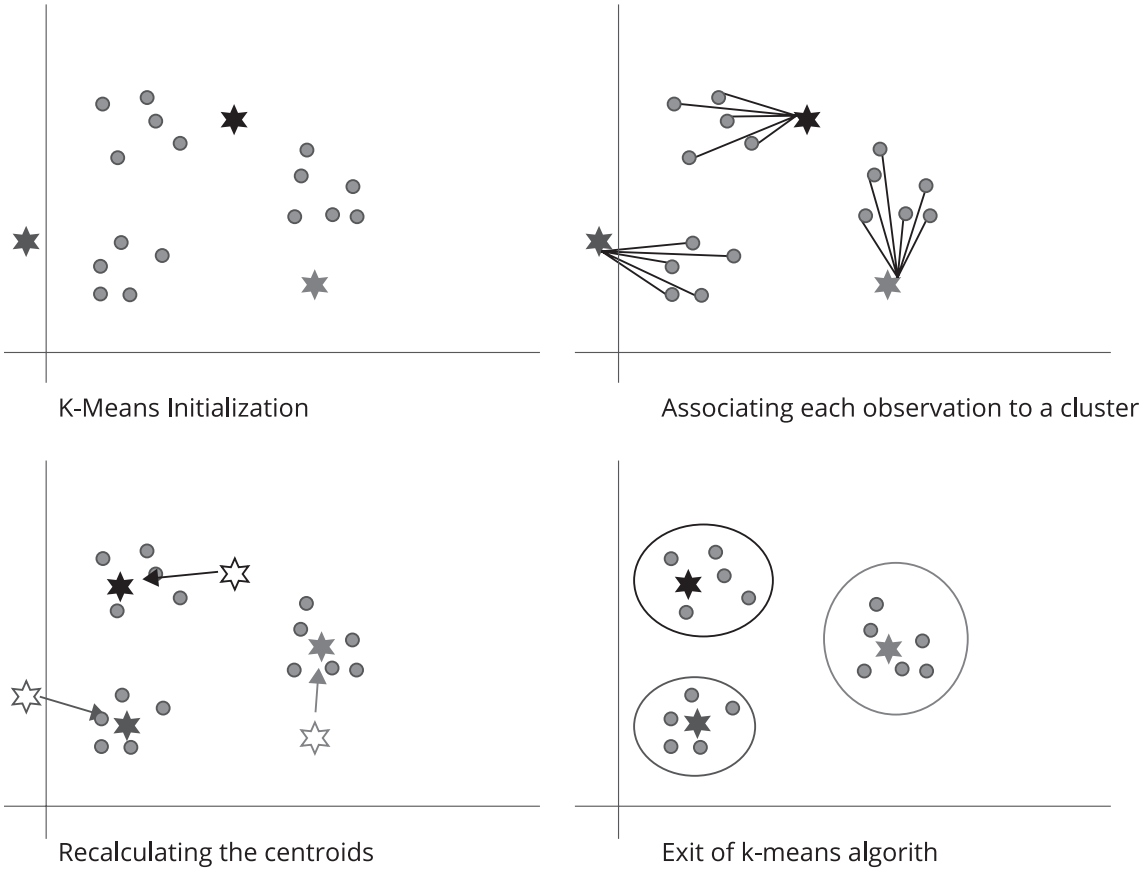
צעד 3 : עבור (loop) על כל K האשכולות. עבור כל אשכול, $k = 1,\cdots,K$, חשב מחדש את נקודת המרכז של כל אשכול כממוצע של הקוארדינטות של כל המשתמשים השייכים לאותו אשכול.

צעד 4: סיים את התהליך אם מתקיימים תנאי העצירה. אחרת, חזור לצעד 2.

קיימים מספר כללי עצירה עבור אלגוריתם KM. הנפוץ שבהם, שבו השתמשנו גם באירוע שיתואר להלן, הוא כאשר השינוי של מרכזי האשכולות בצעד 3 קטן מרמה ε שנקבעת מראש.

תרשים 1 מתאר את ארבעת הצעדים של אלגוריתם KM. ה"כוכבים" מציינים את נקודות המרכז (centroids) של האשכולות והעיגולים הקטנים את התצפיות.

תרשים 1: המחשה של אלגוריתם KM (Reena Shaw, 2017)



אלגוריתם השכן הקרוב ביותר
KNN (K-Nearest Neighbor)

אלגוריתם השכן הקרוב ביותר, KNN, מוצא את K השכנים הקרובים ביותר לכל משתמש, כאשר K, מספר השכנים, נקבע מראש. KNN איננו אלגוריתם למציאת אשכולות אבל הוא אלגוריתם המלצות ראוי, שהרי אין דבר הגיוני יותר מאשר להמליץ למשתמש על מוצרים מתוך רשימת המוצרים שהשכנים הקרובים ביותר אליו קנו בעבר. מידת הקרבה בין 2 משתמשים מתבססת על פונקציית המרחק, כמתואר לעיל. האלגוריתם הוא פשוט:

צעד 1: אתחול – בחר את מספר השכנים K.

צעד 2: עבור (loop) על כל המשתמשים וחשב לכל משתמש את המרחק שלו מכל אחד מהמשתמשים שבבסיס הנתונים.

צעד 3: עבור כל משתמש מצא את K המשתמשים הקרובים לו ביותר על פי פונקציית המרחק.

הקושי שאלגוריתם KNN מעורר הוא שהוא עתיר חישובים, במיוחד כשמדובר בבסיסי נתונים גדולים. במקרים כאלה יש צורך ב"קיצורי דרך" כדי להפחית את מספר החישובים. דרך אחת היא לחלק את המשתמשים בבסיס הנתונים לאשכולות על בסיס משתנים סוציו-אקונומיים ואז להפעיל את אלגוריתם KNN על כל אשכול בנפרד. ההנחה כאן היא שיש סבירות גבוהה שהמשתמשים הקרובים ביותר לכל משתמש שייכים לאותה קבוצה סוציו-אקונומית.

חוקים אסוציאטיביים – AR (Association Rules)

גם אלגוריתם KM וגם חברו KNN מתבססים על כך שמספר המוצרים שהמשתמש קנה בעבר יהיה לפחות שניים. אבל הרי רוב המשתמשים ביצעו רק קנייה אחת בעבר, או לכל היותר 2 קניות. עבור משתמשים אלה אנו מציעים להשתמש בחוקים אסוציאטיביים – AR, כדי למצוא את המוצרים הרלוונטיים ביותר. למעשה, חוקי AR אינם אלא הסתברויות מותנות (conditional probabilities) מתורת ההסתברות. הרעיון הוא לחשב את ההסתברות שבקרות אירוע A יקרה אירוע X, כלומר $P(X|A)$ עבור כל האפשרויות (ריאליזציות) של X. ואז, האירוע X שעבורו ההסתברות המותנית $P(X|A)$ היא המקסימלית הוא האירוע הרלוונטי ביותר לאירוע A.

בהקשר של בעיית המכירה הצולבת, האירוע A מייצג את המוצר ה"תורן" שבו מתעניין המשתמש והאירוע X הוא מוצר אחד מתוך כל המוצרים האחרים שהמשתמשים שרכשו את מוצר A קנו בעבר. האתגר שלפנינו הוא להחליט איזה מוצר X מתוך כל המוצרים להציע למשתמש שמתעניין במוצר A.

על פי תורת ההסתברות, ההסתברות המותנית שיקרה אירוע A בהינתן שקרה אירוע X, $P(X|A)$, היא:

$$P(X|A) = P(X \Omega A)/P(A)$$

כדי לחשב הסתברויות אלה יש צורך לחשב את מספר המקרים שבהם שני האירועים קורים יחד, היינו את מספר המשתמשים בבסיס הנתונים שקנו בעבר גם את מוצר A וגם את מוצר X. מספר זה, מחולק במספר המשתמשים, הוא המונה בנוסחת ההסתברות המותנית הנ"ל. המכנה הוא פרופורציית המשתמשים שקנו את מוצר A בעבר.

ניתן להרחיב את המהלך האמור גם למקרה של 2 מוצרים, A ו-B. כאן המטרה שלנו למצוא איזה מוצר X להציע למשתמש שמתעניין בשני המוצרים (למשל, קנה בעבר מוצר B ועכשיו מתעניין גם במוצר A), כלומר למצוא את המוצר X שעבורו ההסתברות המותנית $P(X|A \Omega B)$ היא הגבוהה ביותר.

הפרופורציה של המשתמשים שקנו מוצר A מתוך כלל המשתמשים בבסיס הנתונים נקראת ה"תמיכה" (support) של החוק. פרופורציית המשתמשים שקנו את מוצר X בין אלה שקנו את מוצר A (כלומר, ההסתברות המותנית שלעיל) נקראת ה"ביטחון" (confidence) של החוק.

חוקים משמעותיים הם כאלה שיש להם support גבוה (כדי למנוע רעשים) וגם confidence גבוה (כדי להבטיח נכונות (correctness) של החוק. חוקים הם "מעניינים" (interesting) אם $P(X|A)$ גדול באופן משמעותי מ-$P(X)$.

תקצר היריעה במאמר זה לדון בפרוטרוט בחוקי AR. אלגוריתמים אלה נדונים בהרחבה בספרות, למשל Agrawal and Srikant, 1994.

איך מתרגמים את תוצאות המודל להמלצות?

תהליך התרגום של תוצאות המודל להמלצות תלוי באלגוריתם CF:

גישת KNN תמליץ עבור כל משתמש את r המוצרים הפופולריים ביותר שקנו K ה"שכנים" הקרובים ביותר שלו.

גישת KM תמליץ עבור כל משתמש את r המוצרים הפופולריים ביותר שנקנו על ידי המשתמשים השייכים לאשכול שלו.

ולבסוף, גישת AR תמליץ עבור כל משתמש את r המוצרים בעלי ההסתברות המותנית הגדולה ביותר.

ראוי לציין שרמת ההתאמה האישית שונה מגישה אחת לשנייה. ההמלצות המותאמות ביותר אישית מתקבלות בגישת ה-KNN. הסיבה היא שגישת ה-KNN מופעלת מחדש על כל משתמש על מנת למצוא את K השכנים הקרובים לו ביותר, בשעה שגישת ה-KM מתעדכנת אחת לתקופה. לכן שני משתמשים שקנו מוצר מסוים, או הביעו עניין בקניית המוצר הזה, יקבלו את אותן המלצות בגישת KM אבל לא בהכרח את אותן המלצות בגישת KNN.

ניתן, כמובן, לשפר את ההתאמה האישית של ההמלצות בגישות CF על ידי התחשבות בהיסטוריית הקניות של המשתמש. למשל, להסיר מרשימת ההמלצות מוצרים שהמשתמש קנה בעבר בטווח זמן מסוים, או להתחשב בגורמים אחרים. למשל, ביישומים של מסחר אלקטרוני (e-Commerce), לחשוף את המשתמש להמלצות בזמן שבו קיימת סבירות הכי גבוהה שהמשתמש יגיב להמלצה. לדוגמה, לחשוף את ההמלצות למשתמש הידוע כ"חיית לילה" (night owl) רק בשעות הלילה, ולמשתמש שנוהג להזמין מוצרים רק בסופי שבוע - רק בימי סוף השבוע, וכו'. אבל פעולות אלה מחייבות להתחשב בשיקולים נוספים שנמצאים מעבר לטווח (scope) של מאמר זה.

תיקוף המודל

המטרה של תיקוף מודל בכריית מידע היא להבטיח שהמודל הוא בעל יכולת הכללה (generalization), היינו שניתן ליישם את תוצאות המודל שנבנה על סמך התנהגות בעבר גם על תצפיות חדשות. במקרה של מערכת המלצות, מטרתנו היא לבדוק את איכות ההמלצות כשמיישמים אותן על משתמשים חדשים. אבל בניגוד לבעיות של חיזוי אנליטי (predictive analytics) שבו מתקפים מודל על סמך מדגם אימות שלא לקח חלק בבניית המודל (יעקב זהבי, 2017), במקרה שלנו נשתמש בגישה שהוצעה על ידי Herz et al., (1997) ונוציא מסל המוצרים של כל משתמש מספר מוצרים שאותם "נחזיק בצד" ואז נשווה, בדיעבד, באיזו מידה ההמלצות של מודל ה-CF תואמות את הקניות בפועל.

לצורך זה הגדרנו שני מדדים:

שיעור ההתאמה (matching rate) - אחוז המשתמשים שעבורם המוצרים "המוחזקים בצד" לצורך תיקוף המודל מופיעים ברשימת ההמלצות.

ה"גידול" (lift) בשיעור ההתאמה של המלצות שנבחרו בגישת CF בהשוואה להמלצות שנבחרו באופן אקראי.

להלן נדגים את המדדים האלה על אירוע אמיתי.

אירוע מעשי

בסעיף זה נתאר יישום של מערכת CF על אתר e-Commerce של חברה אמריקאית גדולה המשווקת מוצרים לאספנים (collectible items). ה"משתמשים" באירוע זה הם גולשים המתעניינים ברכישות באמצעות אתר האינטרנט של החברה. ה"מוצרים" הם דפי נחיתה באתר, כשכל דף מתייחס למוצר מסוים. את הנתונים הפקנו מתוך קובצי הלוג (log files) של האתר בתקופה של חודש. לאחר עיבוד מקדים (pre-processing) של הנתונים שבו סיננו דפים לא רלוונטיים (כגון דף הבית), רעשים, כפילויות, דפי jpeg וכו', נותרנו עם 94 דפי נחיתה (URLs) ובנינו מטריצה עם n שורות, שבה שורה לכל גולש עם 94 עמודות 0/1: 1 - אם הגולש נחת על הדף המתאים, 0 - אם לא. מתוך המטריצה הזו "שלפנו" באופן אקראי מוצר אחד או שניים לכל גולש שאותם השארנו בצד לצורך תיקוף המודל.

להלן נציג את התוצאות עבור מדגם התיקוף בגישות KNN, KM עבור K=10 ומרחק הקוסינוס. לצורך בדיקת הרגישות של המודל, בדקנו מספר פרמטרים:

Min Pages - מספר מינימלי של "ביקורים" באתר על מנת לכלול את המשתמש במודל (3 - 6).

Taken Out - מספר המוצרים שמוחזקים בצד לצורך תיקוף המודל (1 או 2). למשל, אם המשתמש ביקר ב-6 דפי נחיתה במשך החודש ואנו מחזיקים בצד 2 מוצרים לצורך תיקוף המודל, מדגם האימון לצורך בניית מודל CF עבור הגולש המסוים הזה יכלול 4 מוצרים, ומדגם התיקוף יכלול 2 מוצרים.

Weight – פונקציית המשקל, כדי ליצור המלצות "מעניינות" – פונקציה הפכית (inverse function) של הפופולריות של דף הנחיתה (היינו אחד חלקי מספר הביקורים הכולל ב-URL).

טבלה מספר 1 מפרטת את הממדים של מדגם האימון כפונקציה של 2 הפרמטרים Min Pages, Taken Out. נשים לב שככל שהפרמטרMin Pages גדול יותר, מדגם האימון המשמש לבניית אלגוריתם CF קטן יותר.

טבלה 1: הממדים של מדגם האימון

Min Pages	Taken out	Number of Users	Number of Pages
3	1	65850	290733
4	1	47922	254877
4	2	47922	206995
5	1	34052	213627
5	2	34052	179215
6	2	23978	148993

סוגיה נוספת בהקשר של תיקוף מודל ההמלצות היא קבוצת הייחוס (baseline) להשוואה של המודל, או במילים אחרות, כנגד מה אנחנו משווים את ההמלצות ממודל ה-CF? דרך אפשרית אחת היא להשוות עם המוצרים הפופולריים ביותר, דרך אחרת - עם מוצרים "פעילים", למשל מוצרים שנקנו בשנה האחרונה או באזור מסוים, ועוד. אבל מה שאנו מעוניינים בו הוא התרומה "נטו" של גישות CF לשיפור איכות ההמלצות. לכן בתהליך התיקוף שלהלן אנו משווים את מערכת ההמלצות המתקבלות בגישות CF עם מוצרים לתיקוף שנבחרו בצורה אקראית.

טבלה 2 מפרטת את תוצאות התיקוף עבור המוצרים המוחזקים בצד - טבלה 2A עבור מודל KNN וטבלה 2B עבור מודל KM.

טבלה 2 מפרטת את שיעור ההתאמה (matching rate) בין ההמלצות מבוססות CF לבין ההתנהגות בפועל: עמודה First Recom כאשר מציגים למשתמש רק את ההמלצה המועדפת ביותר ממודל ה-CF, עמודהFirst 2 Recom כאשר מציגים למשתמש את 2 ההמלצות המועדפות ביותר ממודל ה- CF , וכן הלאה. מספר ההמלצות הכולל לכל משתמש נע בתחום 1-10 אבל טבלה 2 מציגה תוצאות מפורטות רק עבור 5 המוצרים המועדפים ביותר. הטור הימני בטבלה 2A מתייחס למקרה של K=10, היינו כאשר מציגים למשתמש את כל 10 ההמלצות. על מנת להגדיל את ההתאמה האישית של מערכת ההמלצות לכל משתמש, הסרנו מרשימת ההמלצות של המודל מוצרים שהמשתמש רכש בעבר כך שכל ההמלצות למשתמש, בין אם מדובר בהמלצה אחת, בשתיים או בעשר, כוללות רק מוצרים שהוא לא קנה בעבר.

להבנת הנתונים, נסתכל לדוגמה בטבלה 2A המפרטת את תוצאות התיקוף עבור מודל ה-KNN. שתי השורות הראשונות בטבלה מתייחסות למקרה שבו נדרשים לפחות 3 ביקורים באתר כדי שהמשתמש ייכלל בתהליך, כאשר מוציאים אחת מהן באופן מקרי לצורך תיקוף המודל (מה שמשאיר רק 2 ביקורים עבור בניית מודל ה-KNN). השורה הראשונה מפרטת את התוצאות ללא שימוש במדד הפופולריות, שנזכור באה לידי ביטוי באמצעות פרמטר המשקל (Weight), לצורך בניית המודל, ואילו השורה השנייה עושה שימוש במדד הפופולריות. עבור מקרים אלה, מסתבר שאפילו כשמדובר רק בהמלצה אחת, יש התאמה של כמעט 11% בין ההמלצה של מודל ה-KNN לבין התוצאות בפועל. כלומר, עבור 11% מהמשתמשים, ההמלצה של מודל ה-KNN תואמת את מוצר ה"תיקוף" (היינו, את המוצר שאנחנו מחזיקים בצד לצורך תיקוף המודל). ניתן לפרש את שיעור ההתאמה כהסתברות, ובמילים אחרות, עבור הפרמטרים Min Pages=3, Taken Out=1, ההסתברות שההמלצה של המודל תואמת את מוצר ה"תיקוף", כאשר מדובר בהמלצה יחידה, היא 11%. כאשר מדובר ב-5 המלצות, ההסתברות שלפחות אחת מהן תואמת את מוצר ה"תיקוף" עולה ל-23.4%, וכאשר מדובר בכל 10 ההמלצות - ההסתברות עולה ומתקרבת ל-25% .

שיעורים אלה גדלים ככל שהפרמטר Min Pages גדול יותר. למשל, כאשר נדרשים לפחות 6 ביקורים באתר על מנת לכלול את המשתמש במדגם האימון ו-5 המלצות, ההסתברות שלפחות אחת מההמלצות תואמת את המוצרים המשמשים לתיקוף המודל עולה ל-40% במודל ה-KNN. הסתברות זו עולה כמעט ל- 50% כשמדובר בכל 10 ההמלצות. הסיבה לכך הגיונית, שכן ככל שהמודל מתבסס על מספר דפי נחיתה (מוצרים) גדול יותר, המלצות המודל ממוקדות יותר, מה שמעלה את הסתברות ההתאמה בין תוצאות המודל להתנהגות בפועל.

השוואה מעניינת היא בין שיעורי ההתאמה של מודל KNN לאלה של מודל KM. ההשוואה בין טבלה 2A לבין טבלה 2B מעידה ששיעורי ההתאמה של מערכת ההמלצות עבור מודל KM גדולים יותר, באופן עקבי, מאשר במודל KNN. הסיבה לכך היא שבגישת KM ההמלצות לכל משתמש מתבססות על מאפייני הקנייה של מספר גדול מאוד של משתמשים המונים כמה אלפים או יותר (למעשה, כל המשתמשים השייכים לאותו אשכול). זאת לעומת גישת KNN שבה ההמלצות לכל משתמש מתבססות על מספר קטן מאוד של משתמשים, למעשה רק על K המשתמשים הקרובים לו ביותר (למשל, 10 בדוגמה שלעיל). עובדה זו מגדילה את סיכויי ההתאמה בין ההמלצות לבין מוצרי התיקוף בגישת KM לעומת גישת KNN.

כך או כך, שיעורי ההתאמה של שני המודלים הם מרשימים, אפילו כשמדובר בהמלצה אחת: 10%-15% עבור מודל KNN, 25%-35% עבור מודל KM .

דרך אחרת לבחון את התוצאות היא באמצעות מדד ה-lift שמבטא את הגידול בשיעור ההתאמה (ההסתברות) בין המלצות מבוססות CF לבין המלצות אקראיות. כשמדובר ב-94 מוצרים, כמו במקרה שלנו, ההסתברות שהמלצה שנבחרה באופן מקרי תהיה זהה למוצר ה"תיקוף" היא 1/94, כלומר קרוב ל-1%. אם אותה המלצה נבחרת על פי המודל, ההסתברות (שבאה לידי ביטוי באמצעות שיעור ההתאמה) קופצת לרמה 10%-15% עבור מודל KNN. ולגבי המקרה של 5 המלצות - אם נבחר אותן באופן מקרי, ההסתברות שאחד המוצרים יהיה זהה למוצר שהמשתמש רכש בפועל עולה ל-5/95 או כמעט ל-5%. לעומת זאת, אם נבחר את 5 המוצרים על פי מודל KNN, ההסתברות קופצת לרמה של 20%-40%. בשני המקרים מדובר בעלייה משמעותית בהחלט ב-lift. אותה תופעה קיימת גם לגבי מודל KM.

המשמעות הכלכלית של מערכת המלצות מבוססות CF ברורה: היא מעלה את הסתברות התגובה של המשתמשים בהשוואה להמלצות שנבחרות באופן אקראי. כשמדובר בחברה יצרנית או בחברה שמשווקת מוצרים, מערכת המלצות מבוססות CF מעלה את רמת המכירות ומשפרת את הרווחיות של החברה.

טבלה 2A: תוצאות התיקוף עבור מודל KNN, K=10

Min Pages	Taken out	Weight	First Recom.	First 2 Recom.	First 3 Recom.	First 5 Recom.	All 10 Recom.
3	1	No	10.8	16.8	20.1	23.4	24.9
3	1	Yes	10.9	15.2	17.4	19.3	20.5
4	2	No	9.0	14.9	18.9	23.5	25.9
4	2	Yes	9.7	15.1	18.0	20.8	22.7
4	1	No	14.6	22.5	27.1	31.8	34.1
4	1	Yes	14.2	20.2	23.3	26.1	27.8
5	2	No	12.0	20.2	26.0	32.3	36.0
5	2	Yes	13.4	21.0	25.2	29.1	31.7
5	1	No	19.2	29.4	35.5	41.8	45.4
5	1	Yes	18.7	26.5	30.5	34.4	36.8
6	2	No	15.8	27.0	34.7	43.4	49.3
6	2	Yes	17.5	27.5	33.0	38.3	42.1

טבלה 2B: תוצאות התיקוף עבור מודל KM, K=10

Min Pages	Taken out	Weight	First Recom.	First 2 Recom.	First 3 Recom.	First 5 Recom.	All 10 Recom.
3	1	No	28.6	43.9	52.9	64.5	77.6
3	1	Yes	29.0	45.9	56.6	70.8	88.2
4	2	No	23.9	40.0	50.5	63.2	77.4
4	2	Yes	25.2	41.7	53.2	68.0	85.8
4	1	No	32.4	48.7	59.1	72.5	85.5
4	1	Yes	32.2	48.6	59.7	73.4	87.1
5	2	No	21.5	35.4	46.0	60.7	77.2
5	2	Yes	24.8	40.5	50.6	64.0	81.9
5	1	No	29.4	44.9	54.8	67.3	79.5
5	1	Yes	33.2	48.0	58.3	72.2	87.9
6	2	No	22.4	39.9	51.8	66.0	81.1
6	2	Yes	25.4	41.2	51.5	64.2	81.8

יישומים של גישות CF מבוססות-פריטים

אולי בשל הדומיננטיות של חברות אמזון ונטפליקס, מרבית היישומים של CF שמתוארים בספרות הם בתחום של מסחר אלקטרוני. אבל לגישות CF יש יישומים נוספים שבחלק מהם נדון בסעיף זה.

מסחר אלקטרוני (e-Commerce): האינטרנט, מערכות התקשורת המתקדמות ועולם ה-WWW, הם זירה אידיאלית למערכות המלצה מבוססות CF. ברוב המקרים, ההמלצות באות בתגובה לעסקה שעושה לקוח (משתמש) באתר, למשל קנייה של מוצר, הורדה של שיר, הזמנה של סרט וכיו"ב. בעקבות העסקה, מערכת CF נכנסת לפעולה ומציגה ללקוח, בדרך כלל כשהוא עדיין משוטט באתר, את ההמלצות המתאימות לו ביותר, שמשקפות את העסקה הנוכחית ואת מאפייני הקניות של הלקוח ושל אלה הדומים לו ("peers"). לרוב, ההמלצות הן בנוסח: "אנשים שקנו מוצר X קנו גם את מוצר Y". כאמור, החברה המובילה בתחום זה היא חברת אמזון. העובדה שיש צורך ליצור את מערכת ההמלצות בזמן אמיתי ("on the fly") מסבכת את תהליך היישום של מערכת ה-CF אך איננה מונעת אותו.

שיווק ישיר (Direct Marketing): מערכת המלצות מבוססות CF מעשירה את ארגז הכלים העומדים לרשות המשווקים הישירים כדי לשפר את ההחלטות על מכירה צולבת. ברוב היישומים של שיווק ישיר המטרה היא לקבוע איזה מוצר יחיד (solo piece) או סט של מוצרים (קטלוג) להציע לכל לקוח בבסיס הנתונים. החלטות אלה מבוססות בדרך כלל על מודלים של חיזוי אנליטי ונועדו לאתר את קהל היעד שכדאי לפנות אליו (targeting) (יעקב זהבי, 2017). בגישות אלה, כל הלקוחות שעוברים סף תגובה מסוים נחשפים לאותו מוצר או קטלוג. שימוש בגישות CF מאפשר למקד יותר את ההמלצות ולהתאים אותן למאפיינים האישיים של הלקוח ועל ידי כך להעלות את שעורי התגובה (response).

ניהול תוכן (Content Management): ארגונים משתמשים בניהול תוכן כדי לבנות תדמית, להעביר מידע רלוונטי ללקוחות שלהם, לתמוך במבצעי שיווק ועוד. דא עקא, שכמות מוצרי התוכן כגון מאמרים, מסמכים, דוחות, ספרים, ידיעות בעיתונות הכתובה והאלקטרונית וכיו"ב היא עצומה, מה שהופך את ניהול התוכן לסוגיה מורכבת מאוד. גישות CF יכולות לעזור לניהול תוכן באמצעות המלצות ממוקדות שמציעות לכל לקוח אך ורק את מוצרי התוכן הרלוונטיים עבורו בצורה דומה להמלצות על מוצרים, בהבדל שהעמודות במטריצת הנתונים מתייחסות למוצרי

תוכן (במקום למוצרים) והשורות מתייחסות ללקוחות שרכשו מוצרי תוכן בעבר.

ניהול גלישה באתר (Web Management and Navigation): אתרי אינטרנט של חברות וארגונים עשויים להכיל אלפים, אם לא יותר, של דפים, מה שהופך את הניווט באתרים אלה למשימה בלתי אפשרית. ואמנם, הרבה גולשים שמנסים לדלות אינפורמציה מאתרי אינטרנט "מרימים ידיים" ופורשים מהאתר, מה שעשוי להביא להפסד של מכירות. אחד הפתרונות להקל על הבעיה הוא לבנות תוכנית ניווט אישית. רעיון אפשרי הוא לקבץ את כל הדפים באתר לאשכולות (clusters) על פי מאפייני הגולשים ולהפנות כל גולש שנכנס לאתר לאשכול הרלוונטי ביותר עבורו (על פי פונקציית המרחק). אזי ניתן להשתמש בכל אשכול במנגנון מבוסס CF כדי לעזור לגולש למצוא את דרכו באתר.

ניהול פרסום מקוון (On Line Advertising Management): שיעור ההקלקה (Clickthrough Rate) על באנרים באתרים מקוונים הוא נמוך ביותר, במיוחד במקרים שבהם ההקצאה של באנרים ללקוחות נעשית באופן אקראי. הדרך להעלות את שיעור ההקלקה היא באמצעות התאמה אישית של הבאנרים המוצגים ללקוח תוך שילוב של גישות CF. כל שורה במטריצת הנתונים מתייחסת לגולש שביקר באתר בעבר, כל עמודה לבאנר. כשגולש נכנס לאתר, מפעילים מנגנון של CF על מנת להציף את הגולש בבאנרים שלקוחות שדומים לו הקליקו עליהם בעבר. ואמנם, כיום קיימים בשוק מספר ad servers העושים שימוש בשיטות CF ואחרות במטרה להגדיל את שיעורי ההקלקה של גולשים.

המלצות מבוססות-משתמש המסתמכות על דירוגים של משתמשים (Ranking Based Recommendations)

אי-אפשר לסיים מאמר על מערכת המלצות בלי להזכיר את התחרות הבין-לאומית המפורסמת של חברת נטפליקס (Netflix) שיצאה לדרך בשנת 2007. מטרת התחרות הייתה לבנות מודל שיאפשר לחזות בצורה מדויקת יותר את רמת ההנאה מצפייה בסרטים על סמך דירוגים בסולם 1-5 שהמשתמשים נתנו לסרטים שבהם צפו בעבר. נטפליקס הקצתה פרס כספי בגובה של מיליון דולר לאלגוריתם שישפר את הביצועים של אלגוריתם הבית שלהם, Cinematch, לפחות ב-10%.

הקריטריון להשוואה היה RMSE (Root Mean Square Error) בין התחזית לביצועים בפועל. ה-RMSE של אלגוריתם הבית בשנת 2007 היה 0.9525 והיעד היה להורידו ל-0.8572 או פחות. התחרות נמשכה עד להשגת היעד בשנת 2009. הקבוצה הזוכה, Belkore's Pragmatic Chaos, הצליחה להוריד את ה-RMSE לרמה של 0.8567, שיפור של 10.06% לעומת ה-RMSE המקורי. מפתחי הקבוצה דיווחו שהשקיעו קרוב ל-2000 שעות עבודה כדי להגיע לפתרון שזיכה אותם בפרס שהתבסס על קומבינציה של 107 אלגוריתמים(!).

על אף המאמץ האדיר, חברת נטפליקס החליטה לא ליישם את השיטה החדשה לאחר שהגיעה למסקנה שהשיפור בתוצאות אינו מצדיק את המאמץ הרב שהיה כרוך ביישום השיטה בפועל. כמו כן, המעבר לשירות של streaming במקום שירות של סרטי וידאו, שחל באותה תקופה, חייב את חברת נטפליקס להתאים את מערכת ההמלצות כדי לטפל בשירות החדש.

הרווח שנשאר מהתחרות היה אפוא המחקר הקולקטיבי שתרם רבות לקידום הנושא של פיתוח מערכות המלצה. האלגוריתם המרכזי של הקבוצה הזוכה התבסס על גישת Matrix Factorization (להלן: MF) מאלגברה לינארית, גישה שאותה נסקור כאן בקצרה.

בבסיס גישות MF עומדת מטריצת הדירוגים A , שבה N שורות, שורה אחת לכל משתמש, ו-M עמודות, עמודה אחת לכל פריט (במקרה שלנו, סרט). מטריצת הדירוגים A היא לרוב מטריצה "מדוללת" (sparse) שיש בה יותר תאים ריקים מאשר מלאים. דוגמה פשטנית של מטריצת דירוגים עבור 4 משתמשים ו-7 פריטים מופיעה בטבלה שלהלן:

	H. Potter 1	H. Potter 2	H. Potter 3	Twilight	Star Wars 1	Star Wars 2	Star Wars 3
User 1	4			5	1		
User 2	5	5	4				
User 3				2	4	5	
User 4		3					3

במטריצה זו, 4 המשתמשים התבקשו לדרג 7 פריטים (סרטים): 3 סרטים מסדרת Harry Potter, הסרט Twilight, וכן 3 אפיזודות של הסדרה Star Wars (SW). הסיבה שמטריצת הדירוגים מדוללת היא שמרבית המשתמשים רואים רק מעט סרטים (למשל, User 1 ראה רק 3 סרטים מתוך 7), וגם משתמשים שראו סרטים בעבר לא בהכרח מדרגים אותם. המטרה שלנו היא לחזות את הדירוגים החסרים בטבלה על סמך הדירוגים שמופיעים בטבלה.

דרך אחת היא לנסות למצוא דמיון בין סרטים ו/או בין משתמשים. למשל, בהינתן הדירוגים האלה אפשר לטעון שהסרטים SW2 , SW1 הם "דומים" ולכן, מכיוון ש User -1 לא אהב את הסרט SW1 סביר להניח שהוא לא ייהנה גם מהסרט SW2 ולכן ייתן גם לו דירוג נמוך של 1.

במקום לפענח את טעמו של המשתמש או לסווג סרטים על פי מידת הדמיון ביניהם, דרך אלטרנטיבית ושיטתית יותר היא לחזות איזה דירוג ייתן כל משתמש לסרטים בלתי מדורגים בהתבסס על המשתנים ה"נסתרים" (לטנטיים) העומדים בבסיס מטריצת הדירוגים. ההנחה היא שהדירוג הסופי שכל משתמש נותן לסרט מונחה על ידי מספר משתנים נסתרים ובלתי נראים המשקפים את העדפותיו. בהקשר של סרטים, משתנים נסתרים אלה הם:

- הז'אנר של הסרט (קומדיה, דרמה, רומנטיקה)
- קהל היעד (ילדים , מבוגרים)
- האופי של הסרט (קליל, רציני)
- השחקנים המופיעים בסרט
- הנושא של הסרט
- אחרים

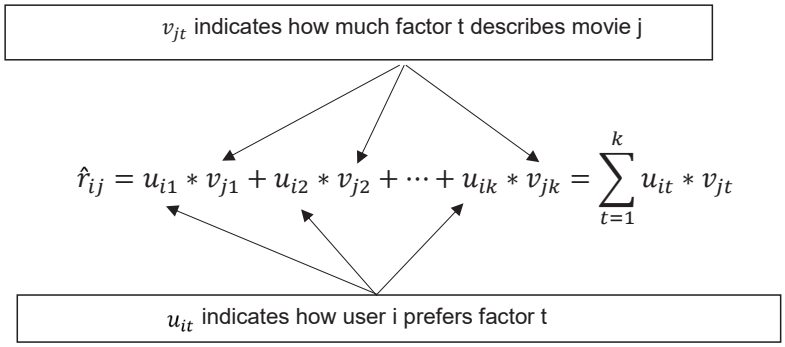
ובסה"כ k משתנים לטנטיים.

כל משתנה לטנטי כזה משפיע על הדירוג של המשתמש. למשל, משתמש שאוהב קומדיות ייתן ציון גבוה לסרטים קומיים וציון נמוך לסרטים רציניים יותר; משתמש שמעדיף שחקן מסוים ייתן דירוג גבוה יותר לסרטים שבהם מופיע השחקן הזה, וכן הלאה.

נגדיר 2 מטריצות V ו-U:

V - מטריצה של המשתנים הלטנטיים שבה כל שורה מתייחסת לסרט וידאו מסוים וכל עמודה מייצגת אחד מהמשתנים הלטנטיים. האלמנט v_{jt} מבטא באיזו מידה הסרט j מושפע מהמשתנה הלטנטי t.

U - מטריצת ההעדפות של המשתמש שבה כל שורה מייצגת משתמש מסוים וכל עמודה את אחד מהמשתנים הלטנטיים. האלמנט u_{it} מבטא באיזו מידה משתמש i מעדיף את המשתנה הלטנטי t.



האלמנטים, u_{it} ,v_{jt} נקראים בספרות בשם Loading Factors -LF ("מקדמי הטעינה") שכן הם מבטאים כיצד הסרטים "נטענים" במרחב של המשתנים הלטנטיים.

תוך שימוש בהגדרות אלה, ניתן לחזות את הדירוג שמשתמש i ייתן לסרט j, $\hat{r}_{ij}$, באמצעות המכפלה הפנימית (inner product) של ההעדפות של המשתמש (הערכיםu_{it}) וההשפעה של המשתנים הלטנטיים (הערכים v_{jt}). ראה הנוסחה בעמוד הקודם.

המכפלה הנ"ל מתייחסת לחיזוי הדירוג של משתמש ספציפי i לסרט ספציפי j. הרחבה לגבי כל המשתמשים וכל הסרטים מתקבלת באמצעות המכפלה הפנימית של המטריצות: $U * V'$ (הגרש מסמן מטריצה הפוכה - transpose).

האלמנטים של המטריצות U, V אינם ידועים ואנו נרצה לאמוד אותם מתוך המשוואה $A = U * V'$ כפונקציה של הדירוגים שהמשתמשים נתנו לסרטים השונים. כדי להקטין את ממדי הבעיה, נדרוש שהמטריצות תהיינה בעלות דרגה K (rank), כאשר K<<min(M,N):

U_K

	LF1	$LF2$	$LF3$
User 1	−0.7	−0.8	−0.4
User 2	0.2	−0.6	1.0
User 3	−0.8	−0.1	−0.8
User 4	0.4	0.3	−0.1
...		...	
User N	−0.3	1.0	0.4

V'_K

	Item1	Item 2	Item 3	...	Item M
LF 1	−0.4	0.6	0.8	...	−0.8
LF 2	−0.8	−0.5	−0.7	...	−0.4
LF 3	0.1	0.9	0.7	...	−0.7

אנו אומדים את האלמנטים של המטריצות U, V באמצעות פתרון של בעיית אופטימיזציה כשפונקציית המטרה היא למזער את ה-RMSE, היינו את סכום הריבועים בין הדירוגים בפועל, r_{ij}, לבין התחזיות של הדירוגים $\hat{r}_{ij}$, על פי הפירוט בנוסחה לעיל:

$$Min_{U,V} \sum_{r_{ij} \in A} (r_{ij} - \sum_1^k u_{it} * v_{jt})^2$$

הניסוח הזה הופך את בעיית האופטימיזציה לבעיית למידה מונחית (supervised learning) שכן פתרונה מונחה על ידי הדירוגים בפועל (r_{ij}). כמו כן, ישנן הרחבות של בעיית האופטימיזציה לטיפול בסוגיות של התאמת יתר (over fitting) וכן הרחבות לטיפול במשתמשים שלא צפו בסרטים בכלל או שצפו בסרטים אבל לא דירגו אותם .

תקצר היריעה לתאר כאן את גישת MF ואת פתרון בעיית האופטימיזציה. תיאור מפורט ניתן למצוא בספרות, לדוגמה, Koren et al., (2009).

סיכום

נציין שהאלמנטים במטריצות הנ"ל הובאו לשם הדגמה בלבד ואינם מייצגים נתונים אמיתיים. ניתן לפרש אותם בדומה לרמות קורלציה בין משתנים, ככל שהרמה שלהם גדולה יותר, הקשר בין המשתנים הרלוונטיים חזק יותר (נשים לב שהקשר הזה יכול להיות גם שלילי). בדרך כלל, מקדמים שהערך האבסולוטי שלהם גדול מ-0.30 נחשבים ל"מובהקים".

מאמר זה עוסק בשימוש בגישות CF לצורך התאמה אישית (personalization) של המלצות ותמיכה בבעיות של מכירה צולבת. בין שלל השיטות של CF אנו דנים ב-2 גישות עיקריות - גישת השכן הקרוב ביותר (KNN) וגישת K-ממוצעים (KM), שתי גישות מובילות בכריית מידע השייכות לתחום של למידה בלתי מונחית (Unsupervised Learning). גישה נוספת למתן המלצות שנסקרה במאמר היא באמצעות

חוקים אסוציאטיביים, והיא ישימה למשתמשים שקנו רק מוצר אחד או שניים בעבר. בנוסף, סקרנו בקצרה גם את גישת ה-Matrix Factorization, שמשמשת למתן המלצות המתבססות על דירוגים של משתמשים.

איכות ההמלצות המתקבלות באמצעות גישות CF נמדדת באמצעות שיעור ההתאמה בין המלצות המודל לבין ההתנהגות בפועל. לשם כך יישמנו את גישות KM ו-KNN על אירוע אמיתי שמתבסס על נתוני גלישה של חברת שיווק מובילה בארצות הברית, ומדדנו את שיעור ההתאמה בין המלצות המודל לבין מספר מוצרים שנבחרו באופן מקרי מרשימת הקניות של כל גולש והוחזקו ב"צד" לצורך תיקוף המודל. שתי הגישות הניבו המלצות עם שיעורי התאמה גבוהים באופן משמעותי מהמלצות אקראיות. פועל יוצא של מסקנות אלה הוא שהמלצות מבוססות CF תורמות להעלות את שיעורי התגובה של המשתמשים, עובדה שיש משמעות כלכלית חשובה.

להשלמת היריעה, דנו גם במספר יישומים פוטנציאליים של גישות CF מבוססות-פריטים בתחומים שונים.

בעולם שבו מספר המוצרים, השירותים, פריטי תוכן, קטעי מוסיקה, סרטים ושירים מגיע לרמות שמקשות על תהליך הבחירה של משתמשים, שיטות CF תופסות מקום חשוב לצורך הפקת המלצות ממוקדות. מצד אחד, המלצות אלה מקלות על המשתמשים לעשות את הבחירה הנכונה של מוצרים ושירותים, ומצד שני הן משפרות את הרווחיות של הארגונים שיעשו שימוש בגישות אלה. לא בכדי, חברת אמזון וחברת נטפליקס, שעשו את השימוש המושכל ביותר בגישות CF, הגיעו, כל אחת בתחומה, לשיאים מרשימים מבחינת מכירות ושווי חברה.

פרופ' יעקב זהבי | jacobz@tauex.tau.ac.il

רשימת מקורות

Agrawal, R. & Srikant R. (1994). Fast Algorithms for Mining Association Rules. Proceedings 20th International Conference on Very Large Databases.

Bock, H.H. (1974). Automatic Classification. Vandenhoeck and Ruprecht, Gottingen.

Herz, F., Ungar, L. & Labys, P. (1997). A Collaborative Filtering System for the Analysis of Consumer Data. University of Pennsylvania, Philadelphia.

Kohonen, K., Makisara, K., Simula, O. & Kangas, J. (1991). Artificial Networks. Amsterdam.

Koren, Y., Bell, R. and Volinski, C. (2009), Matrix Factorization for Recommender Systems, Computer, Vol 42, 8.

Lauritzen, S.L. (1995). The EM algorithm for Graphical Association Models with Missing Data. Computational Statistics and Data Analysis, 19, 191-201.

Reena Shaw (2017), kdnuggets, 10.

Silverman, B.W. (1986). Density Estimation for Statistics and Data Analysis. Chapman & Hall.

Zhang, R., Ramakrishnan, R. & Livny, M. (1996). An Efficient Data Clustering Method for Very Large Databases. Proceedings ACM SIGKDD International Conference on Management of Data. 103-114.

יעקב זהבי, חיזוי אנליטי (Predictive Analytics) – הלכה למעשה (2017), חידושים בניהול, הפקולטה לניהול ע"ש קולר, אוניברסיטת תל אביב, 1, 55-69.