CHAPTER 1

The Recommendation Process

1.1 INTRODUCTION

RS = Recommendation System

Netflix prize 2006

$U = \{u_1, \dots, u_M\}$ users

$I = \{i_1, \dots, i_N\}$ items

($M >> N >> 1$ לרוב)

r_i^u היא עד כמה u מעריך את i – $R \in \mathbb{R}^{m \times n}$

לכל זוג (u,i), $r_i^u \in V = \{0, \dots, V\}$

כלומר, $R \in V^{m \times n}$

0 – דירוג לא ידוע, V- דירוג מקסימלי.

נסמן $\bar{r}_R$ – הדירוג הממוצע מבין הדירוגים הידועים (=הגדולים ממש מ-0)

R לרוב דלילה.

Denote by $\langle u, i \rangle$ the enumeration of all those dyads in R s.t. $r_i^u > 0$.
Denote by $\langle u, i, r \rangle$ an enumeration of all the explicit ratings.

Denote $\mathcal{F}$ – the whole set of predicates relative to U, I or $U \times I \times V$.

$$I(u) = I_R(u) = \{i \in I \mid \langle u, i \rangle \in R\}$$ כל הפריטים שדורגו ע"י u

$$\mathcal{U}(i)=\mathcal{U}_R(i)=\{u\in\mathcal{U}\mid\langle u,i\rangle\in R\}$$

כל המשתמשים שדרגו את i

Active user - any user u s.t. $\mathcal{I}_R(u)\neq\emptyset$

(any user with a rating history)

Cold Start - הבעיה ש- $\mathcal{I}(u)$ או $\mathcal{U}(I)$ עשויים להיות ריקים $\emptyset$.

בהינתן active user u, המטרה של RS היא לספק ל-u רשימת המלצה $\mathcal{L}_u\subseteq\mathcal{I}$ של פריטים שטרם דרג - $\mathcal{L}_u\cap\mathcal{I}_R(u)=\emptyset$.

Algorithm 1 Generation of recommendation list.

Require: A number of candidate items, D (positive integer such that $D\leq N$)
Require: The size of the recommendation list, L (positive integer such that $L\leq D$)

1: Choose $\mathcal{C}\subseteq\mathcal{I}$, according to a business-specific criterion, such that $|\mathcal{C}|=D$ and $\mathcal{C}\cap\mathcal{I}_{\mathbf{R}}(u)=\emptyset$.
2: Associate each item $i\in\mathcal{C}$ with a score p_i^u, which represents the appreciation of u for i
3: Let $\mathcal{L}_u$ be the selection of the top L items in $\mathcal{C}$ with the highest values p_i^u
4: Return $\mathcal{L}_u$

1.2.1 EVALUATION

- Online evaluation:
  מבקשים ממדגם קטן מהמשתמשים לדרג אי-אלו פיצ'רים של ה-RS.
  לא מומלץ.

- Offline evaluation:
  נחלק את המדגם R ל- T (Train)
  S (Test)

Prediction Accuracy

מתייחס לבעיה כבעיית רגרס'

MAE = Mean Absolute Error (widely used)

$$MSE=\frac{1}{|S|}\sum_{\langle u,i\rangle\in S}\left(r_i^u-\hat{r}_i^u\right)^2$$

$$RMSE = \sqrt{MSE}$$

MPE = Mean Prediction Error

$$= \frac{1}{|S|} \sum_{\langle u,i \rangle \in S} \mathbb{1}\{r_i^u \neq \hat{r}_i^u\}$$

Confusion Matrix and ROC analysis

Recommendation Accuracy

L_u - רשימת ההמלצה ל-u

T_u - רשימה של מוצרים רלוונטיים ל-u.

לדוגמה, כשיש explicit feedback ניתן לבחור אותה כתת-קבוצה רלוונטית של

$\{i \in I_S(u) : r_i^u > t\}$ עבור t - threshold כלשהו.

היכולת של מערכת המלצה יוצרת רשימת המלצה בגודל L יכולה להימדד ע"י:

$$Recall(L) = \frac{1}{M} \sum_{u \in U} \frac{|L_u \cap T_u|}{|T_u|}$$

$$Precision(L) = \frac{1}{M} \sum_{u \in U} \frac{|L_u \cap T_u|}{|L_u|}$$

ממוצע הרמוני של שניהם - F-score:

$$F = \frac{2 \cdot precision \cdot recall}{precision + recall}$$

Algorithm 2 User satisfaction of user u for item i.

Require: An item $i \in \mathcal{T}_u$
Require: A number of candidate items, D (positive integer such that $D \leq |\mathcal{I}|$)
Require: The size of the recommendation list, L (positive integer such that $L \leq D$)
1: Let $\mathcal{C}$ be a random subset of $\mathcal{I}$, with size D, whose elements have not been rated by u
2: Add i in $\mathcal{C}$
3: Assign to each item $i \in \mathcal{C}$ a score p_i^u, which represents the appreciation of u for i
4: Let $\mathcal{L}_u$ be the selection of the L items in $\mathcal{C}$ with the highest values p_i^u
5: **if** $i \in \mathcal{L}_u$ **then**
6: return a *hit*
7: **else**
8: return a *miss*
9: **end if**

מריצים את האלגוריתם הזה על כל $i \in \mathcal{T}_u$

וכו

$$US\text{-}Recall(u, L) = \frac{\#hits}{|\mathcal{T}_u|}$$

$$US\text{-}Precision(u, L) = \frac{\#hits}{L \cdot |\mathcal{T}_u|} = \frac{\#hits \backslash L}{|\mathcal{T}_u|}$$

<u>הסבר:</u> נניח ש- $|\mathcal{T}_u| = 1$ אז במקרה של hit ה recall הוא 1
וה- precision הוא $\frac{1}{L}$. במקרה של miss שניהם 0.
הערכים האלה הם ממוצע על פני כל $\mathcal{T}_u$.

Rank Accuracy

פה אנו מניחים כי דירוג אמיתי של ההעדפות על של
$\mathcal{I}_s(u)$ - τ_u. $\hat{\tau}_u$ - התחזית של הדירוג

Kendall's coef.
$$\mathcal{K}(\tau_u, \hat{\tau}_u) = \frac{2\left(\sum_{i,j \in \mathcal{I}} \mathbb{1}[\![\tau_u(i) \prec \tau_u(j) \ \wedge \ \hat{\tau}_u(j) \prec \hat{\tau}_u(i)]\!]\right)}{N(N-1)} \in [0,1]$$

Spearman's coef.
$$\rho(\tau_u, \hat{\tau}_u) = \frac{\sum_{i \in \mathcal{I}} (\tau_u(i) - \bar{\tau}_u)(\hat{\tau}_u(i) - \bar{\hat{\tau}}_u)}{\sqrt{\sum_{i \in \mathcal{I}} (\tau_u(i) - \bar{\tau}_u)^2 \sum_{i \in \mathcal{I}} (\hat{\tau}_u(i) - \bar{\hat{\tau}}_u)^2}} \in [-1,1]$$ (

<u>Some other Evaluation Metrics</u>

Novelty, Serendipity, Diversity, Coverage

Main Challenges in RSs

- Sparsity and Cold Start –

בעיית הספרסיות ידועה כ- "reduced coverage".

כשנכנסים יוזרים או items חדשים - cold start.

דרך להתמודד - preference elicitation - מבקשים מהמשתמש לספר על העדפותיו כשהוא נכנס לראשונה למערכת.

דרך נוספת להתמודד - להשתמש במידע דמוגרפי ומידע על הפריטים כדי לקשר בין פריטים/משתמשים חדשים לקיימים.

- Scalability –

לרוב פריטים $M >> N$ משתמשים

לכן ישפיעו על קשרי פריט-פריט ופחות משתמש-משתמש.

- Item Churn

נוספים פריטים ומשתמשים חדשים וזה עלול לפגוע במודל להפוך לאימון - [illegible]

incremental model-maintenance techniques

- User Privacy

- Security

against malicious users

דרך להתמודד: קלסיפיקציה לגילוי attacker profiles.

1.3 RECOMMENDATION AS INFORMATION FILTERING

1.3.1 DEMOGRAPHIC FILTERING

לא צוק

1.3.2 CONTENT-BASED FILTERING

חמוד,

Minikovski Distance:

$$dist_p^m(i,j) = \left(\sum_{j=1}^{K} |w_{i,d} - w_{j,d}|^p \right)^{1/p}$$

Cosine Similarity:

$$sim^C(i,j) = \frac{w_i^T w_j}{\|w_i\|_2 \cdot \|w_j\|_2}$$

(effective in dealing with sparse feature vectors)

Jaccard Similarity:

$$sim^J(i,j) = \frac{w_i^T w_j}{w_i^T w_i + w_j^T w_j - w_i^T w_j}$$

$$sim^J(x, -x) = \frac{-\|x\|^2}{\|x\|^2 + \|x\|^2 - (-\|x\|^2)} = -\frac{1}{3} \quad ? \quad 0$$

Jaccard similarity exhibits aspects of both the Euclidean and cosine measures. In particular, it tends to behave like the former (resp. the latter) if $sim^J(i,j) \to 1$ (resp. if $sim^J(i,j) \to 0$).

1.4 COLLABORATIVE FILTERING

The assumption:

Users who adopted the same behaviour in the past will tend to agree also in the future.

Baselines:

$$itemAvg(i) = \bar{r}_i = \frac{1}{|\mathcal{U}(i)|} \sum_{u \in \mathcal{U}(i)} r_i^u$$

$$userAvg(u) = \bar{r}^u = \frac{1}{|\mathcal{I}(u)|} \sum_{i \in \mathcal{I}(u)} r_i^u$$

$$doubleCentering(u,i) = \alpha \bar{r}_i + (1-\alpha) \bar{r}_u$$

פרמטר כיול ללמוד

נסמן ב- b_i^u baseline כלשהו.

CF Approaches

memory-based

בזמן חיזוי ישירות ניגשים
ל- DB.
עצלן-

neighborhood-based methods

חיסרון: מאוד יקרה לכל הבקשה
בזמן חיזוי - דורש טכניקות indexing
מסובכות

neighborhood-based methods tend to detect strong but local relationships

model-based

יש שלב off-line שבונים
מודל personalized,
ובשלב ה- on-line
משתמשים בו כדי
לחזות

חיסרון: עשוי להיות
פחות interpretable

לרוב ה- model-based,
יש
dimensionality reduction
אז הדגש הוא יותר על
weak but global relationships

CF

item-based approach

ההנחה: משתמשים עשוי להעדיף פריטים דומים
למה שאהב בעבר,

$K - \mathcal{N}^K(i;u)$ הפריטים הדומים ביותר ל- i
מבין אלה שדורגו ע"י u.

$$\hat{r}_i^u = b_i^u + \frac{\sum_{j\in\mathcal{N}^K(i;u)} s_{i,j}\cdot(r_j^u - b_j^u)}{\sum_{j\in\mathcal{N}^K(i;u)} s_{i,j}}.$$

neighborhood-based approaches

(הרעיון: למשתמשים דומים העדפות דומות)

$\mathcal{N}^K(u)$ - K השכנים
הקרובים ביותר של u.

$s_{u,v}$ = הדמיון בין u ל- v

$$\hat{r}_i^u = b_i^u + \frac{\sum_{v\in\mathcal{N}^K(u)} s_{u,v}\cdot(r_i^v - b_i^v)}{\sum_{v\in\mathcal{N}^K(u)} s_{u,v}}$$

כיצד נמדוד דמיון?

המשתמשים שדירגו גם את i
וגם את j.

$$\mathcal{U}_R(i,j) \overset{\text{סימון}}{=} \mathcal{U}_R(i)\cap\mathcal{U}_R(j)$$

$$Pearson(i,j) \triangleq \frac{\sum_{u\in\mathcal{U}_{\mathbf{R}}(i,j)}\left(r_i^u - \overline{r}_i\right)\cdot\left(r_j^u - \overline{r}_j\right)}{\sqrt{\sum_{u\in\mathcal{U}_{\mathbf{R}}(i,j)}\left(r_i^u - \overline{r}_i\right)^2}\sqrt{\sum_{u\in\mathcal{U}_{\mathbf{R}}(i,j)}\left(r_j^u - \overline{r}_j\right)^2}};$$

$$AdjCosine(i, j) \triangleq \frac{\sum_{u \in \mathcal{U}_{\mathbf{R}}(i,j)} \left(r_i^u - \bar{r}_u\right) \cdot \left(r_j^u - \bar{r}_u\right)}{\sqrt{\sum_{u \in \mathcal{U}_{\mathbf{R}}(i,j)} \left(r_i^u - \bar{r}_u\right)^2} \sqrt{\sum_{u \in \mathcal{U}_{\mathbf{R}}(i,j)} \left(r_j^u - \bar{r}_u\right)^2}}.$$

הצגה אופטימלית:

$$\hat{r}_i^u = \sum_{j \in \mathcal{N}^K(i;u)} w_{i,j} \cdot r_j^u$$

ואת המשקלות מגדירים לפי:

$$\min \sum_{v \in \mathcal{U}_{\mathbf{R}}(i)} \left(r_i^v - \sum_{j \in \mathcal{N}^K(i;u,v)} w_{i,j} \cdot r_j^v \right)^2$$

$$s.t. \quad w_{i,j} \geq 0 \ \sum_j w_{i,j} = 1 \ \forall i, j \in \mathcal{I}.$$

K פריטים
הכי דומים
ל-i שדורגו
על ידי u ו-v.

1.4.2 LATENT FACTOR MODELS

ההנחה מאחורי latent factor models היא שיש גורמים כמה מאפיינים עיקריים סמויים.

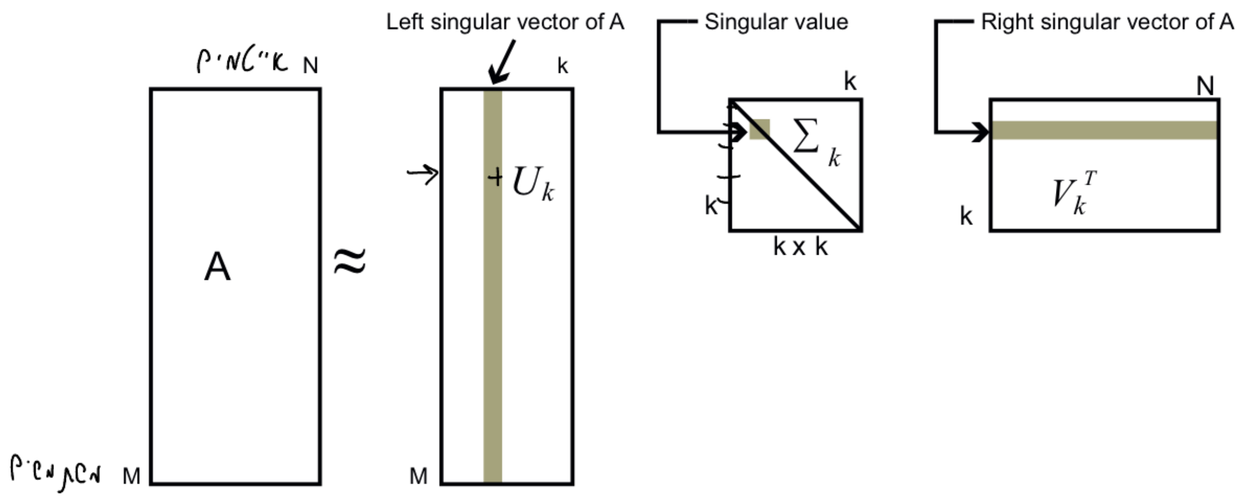
Latent Semantic Indexing = LSI

טכניקת dimensionality reduction ב- IR שמטרתה

מבנה סמוי (latent) בכינוי במילים במסמכים.

$$SVD(A^{m \times n}) = U^{m \times m} \Sigma^{m \times n} V^{T\, n \times n}$$

$$A \approx U_k \cdot \Sigma_k \cdot V_k^{\;T}$$



הפתרון של $\min_{\substack{A_k \in \mathbb{R}^{m \times n} \\ s.t.\ rank(A_k)=k}} \|A - A_k\|_F^2$ הוא

$$A_k = U_k \Sigma_k V_k^T$$

(הקירוב חושף את האופי של הבעיה ומסיר רעש.)

פרשנות לשורות U_k, V_k:

השורה $U_u \in \mathbb{R}^k$ היא הטלה של המשתמש u

למוצר ה-k-מימדי שמייצג את ה features
הסמויים משפיעים ~~על~~ החלטותיו של u

השורה $V_i \in \mathbb{R}^k$ ממפה את i לנקודה במרחב
ה-k-מימדי, שמייצג עד כמה ל-i יש כל אחד מה- features.

גרסה אחרת לכתיבת SVD:

$$R \approx \left(U_k \sqrt{\Sigma_k}^{T}\right)\left(\sqrt{\Sigma_k}\, V_k^{T}\right) = \overset{m\times k}{U} \cdot \overset{k\times n}{V}$$

אינטואיטיבית, כל דירוג r_i^u נכתב כמכפלה הפנימית של ההעדפה של u למוצר
ה-k-מימדי עם ההכלה של i למוצר הא-מימדי.

$$\hat{r}_i^u = \sum_{k=1}^{K} U_{u,k} \cdot V_{k,i}$$

בעיה: R מאוד sparse! - SVD סטנדרטי דורש לדעת
בדיוק מה כל כניסה.
לשים 0 בכניסות הללו, יוצרות עושה bias לכיוון לתת
דירוג נמוך למוצרים שמשתמש לא השתמש בהם, בעוד מה שעלול
להיות לא נכון.

"the absence of evidence is not the evidence of absence"

לכן צריך דרך אחרת לפרק את ה- feature matrices
(U ו-V)

$$(U,V) = \arg\min_{U,V} \left[\sum_{(u,i)\in \Upsilon} \left(r_i^u - \sum_{k=1}^{K} U_{u,k}\cdot V_{k,i}\right)^2 \right]$$

ואפשר לפתור עם GD.

בעיה - זה עלול לגרום overfit.

דרך פתרון regularization:

Maximum Margin Matrix factorization = MMMF

MMMF מגבילה את הנורמות של U,V
שזה שקול להגביל את ה"עוצמה": של ה-features
ולכן אנו מסננת.
אפשר יכולים להיות גם הרבה features אבל רק
קצת מהם יכולים להיות גדול החשובים.

$$U,V = \arg\min_{U,V} \left[\sum_{(i,u)\in \Upsilon} \left(r_i^u - \sum_{k=1}^{K} U_{u,k}\cdot V_{k,i}\right)^2 + \lambda_U \, tr(U^T U) + \lambda_V \, tr(V^T V) \right]$$