

Mátrix faktorizáció (vázlatos bevezetés)

Buza Krisztián

Számítástudományi és Információelméleti Tanszék
Budapesti Műszaki és Gazdaságtudományi Egyetem

Mátrix faktorizáció (áttekintés)



5	?	4	?	...
?	4	?	?	...
?	5	4	?	...
4	?	4	5	...
...

- Feladat: egy mátrix hiányzó elemeit keressük.
- Tudjuk, hogy van valamilyen kapcsolat a mátrix elemei között (egyébként a feladatnak nem lenne értelme).
- Példák: ajánló rendszerek online kereskedelemben, gyógyszerek-gének-betegségek közti kölcsönhatások vizsgálata

Mátrix faktorizáció (áttekintés)



5	?	4	?	...
?	4	?	?	...
?	5	4	?	...
4	?	4	5	...
...

- „Egyszerű” megoldás:
 - Legközelebbi szomszédok keresése (pl. „user-based collaborative filtering)
 - Melyik filmet ajánljuk az első (piros) ügyfélnek?
- „Jobb megoldás”
 - Az ismert elemek alapján próbáljuk meg felírni (közelíteni) a mátrixot két kisebb mátrix szorzataként.

Mátrix faktorizáció algoritmus (vázlat)

- Perceptron algoritmushoz hasonló iteratív algoritmus

$$\begin{array}{|c|c|} \hline 2 & 1 \\ \hline 2 & 2 \\ \hline 3 & 2 \\ \hline 1 & 1 \\ \hline \dots & \dots \\ \hline \end{array} \times \begin{array}{|c|c|c|c|c|} \hline 2 & 2 & 1 & 3 & \dots \\ \hline 1 & 0 & 3 & 3 & \dots \\ \hline \end{array} \approx \begin{array}{|c|c|c|c|c|} \hline 5 & ? & 4 & ? & \dots \\ \hline ? & 4 & ? & ? & \dots \\ \hline ? & 5 & 4 & ? & \dots \\ \hline 4 & ? & 4 & 5 & \dots \\ \hline \dots & \dots & \dots & \dots & \dots \\ \hline \end{array}$$

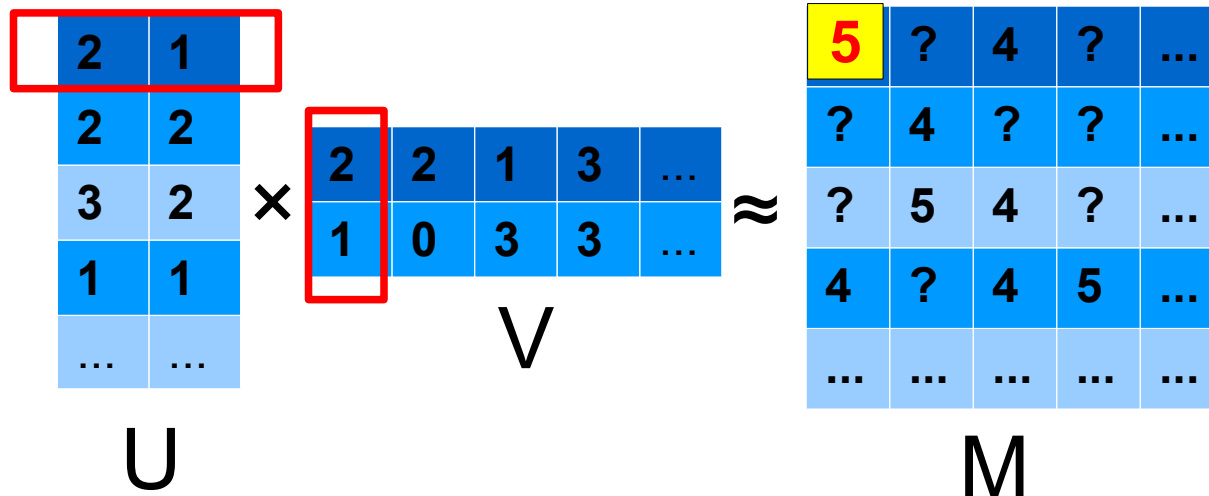
$U \quad \quad \quad V \quad \quad \quad M$

- Véletlenszerűen inicializáljuk U és V elemeit

- Amíg $U \times V$ nem közelíti elég jól M ismert elemeit:
 - Válasszuk M egy ismert elemét, ezt jelöljük x -szel
 - Ha x eltér U és V megfelelő sorának illetve oszlopának szorzatától, akkor – az eltérés irányától függően – csökkentjük vagy növeljük „kicsit” U illetve V adott sorban illetve oszlopban lévő elemeit

Mátrix faktorizáció algoritmus (vázlat)

- Példa:

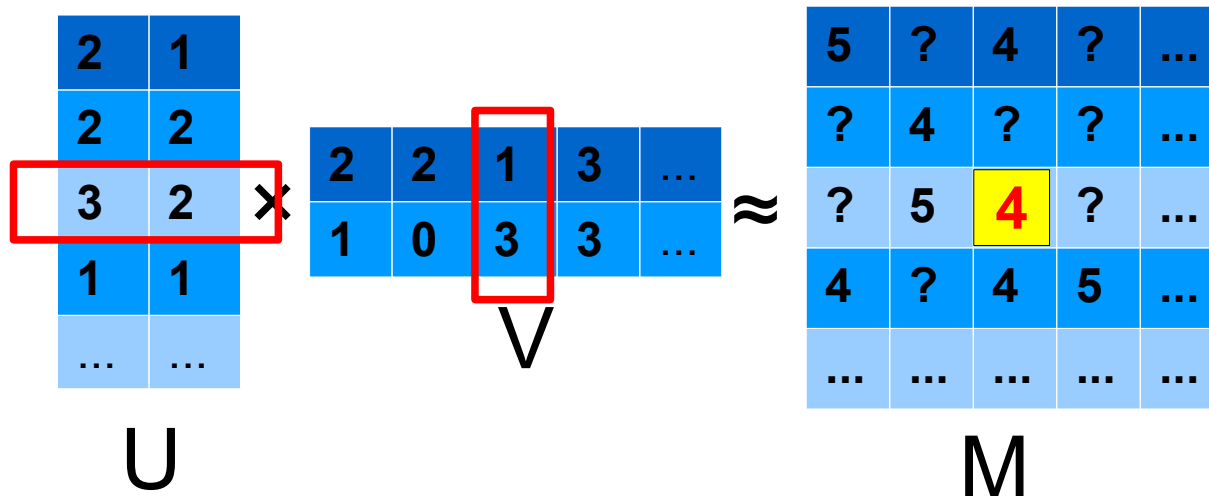


$$(2 \times 2) + (1 \times 1) = 5$$

meggyezik a választott elemmel → nem csinálunk semmit

Mátrix faktorizáció algoritmus (vázlat)

- Példa:

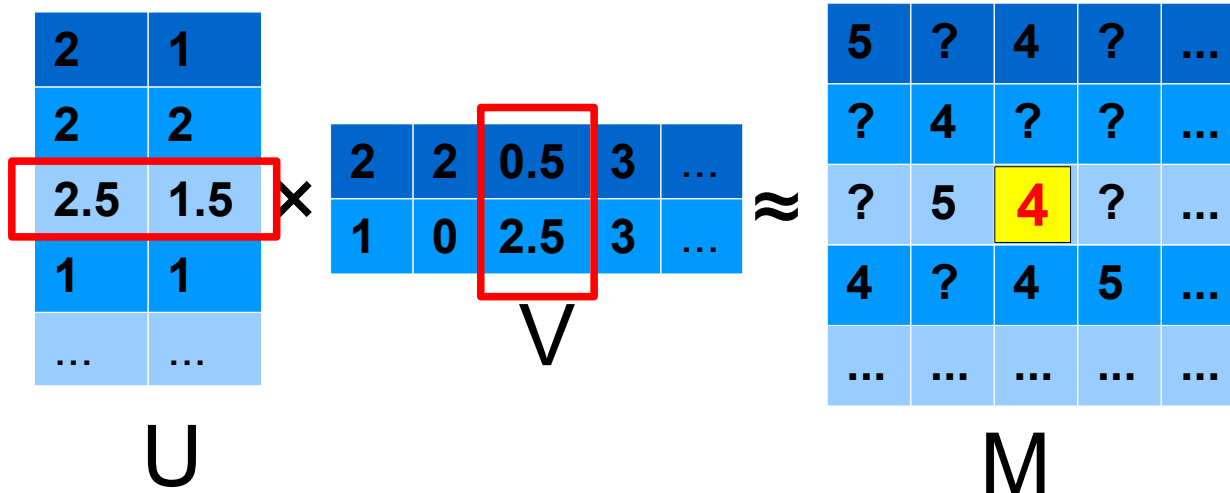


$$(3 \times 1) + (2 \times 3) = 9$$

nagyobb, mint a választott elem \rightarrow csökkentjük a U és V aktuális sorában ill. oszlopában lévő számokat

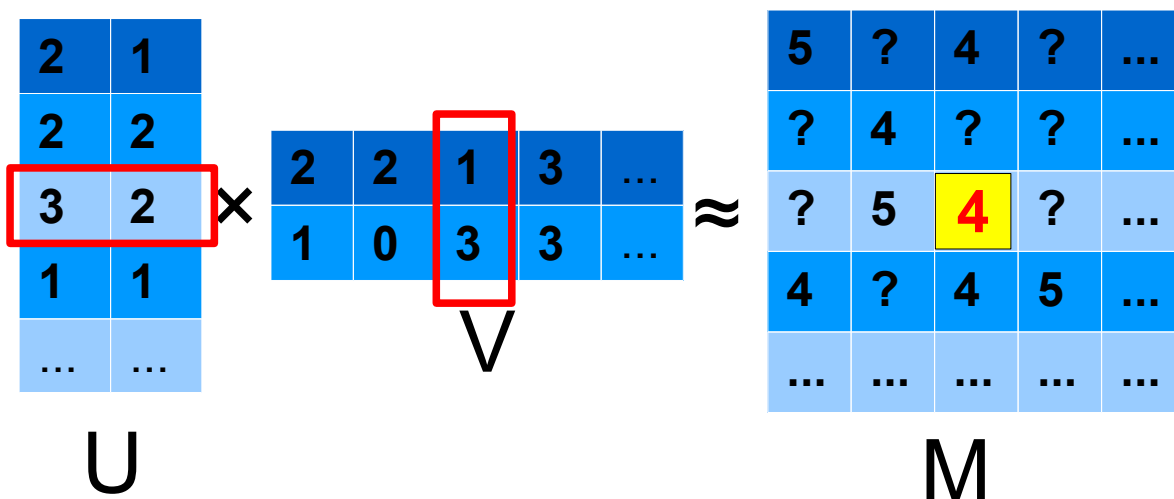
Mátrix faktorizáció algoritmus (vázlat)

- Példa:



Mátrix faktorizáció algoritmus (vázlat)

- Mennyivel csökkentjük/növeljük az aktuális sorok/oszlopok elemeit? (Mit jelent, hogy „kicsit”?)
 - Az, hogy mennyivel csökkentünk/növelünk függjön attól, hogy a két vektor szorzata mennyire tér el a választott elemtől.
 - Csökkentünk/növeljük például az eltérés ε -szorosával
 - **Megjegyzés:** valós algoritmusok ennél valamivel összetettebb frissítési lépést használnak, lásd pl. Yehuda Koren, Robert Bell, Chris Volinsky: *Matrix factorization techniques for Recommender Systems*, IEEE Computer, Aug. 2009



$\varepsilon = 0.1$ esetén:

$$(3 \times 1) + (2 \times 3) = 9$$

$$\text{Eltérés: } 9 - 4 = 5$$

Csökkentünk
 $0.1 \times 5 = 0.5$ -tel

Mátrix faktorizáció – további megjegyzések

- Mekkora legyen ε ?
 - Ha ε túl kicsi \rightarrow túl keveset csökkentünk/növelünk \rightarrow lassan találjuk meg a legjobb közelítést
 - Ha ε túl nagy \rightarrow túl sokat csökkentünk/növelünk \rightarrow túl nagy léptekben haladunk, lehet, hogy nem jutunk az optimumhoz
- Az előbbi algoritmus lokális optimumot keres \rightarrow indítsuk többször egymás után különböző kezdőpontokból, végül válasszuk a legjobbat
- Hány oszlopa/sora legyen U-nak ill. V-nek?
 - Külső paraméter: „rejtett faktor” (latent factor)

Mátrix faktorizáció – további megjegyzések

- Relatívén kevés memóriában implementáció:
 - Nem kell a nagy mátrix (M) hiányzó elemeinek memóriahelyet foglalnunk, mert az iteratív keresés közben csak a meglévő elemekhez férünk hozzá
 - Amikor megtaláltuk az „elég jó” U és V mátrixokat: U és V alapján külön-külön becsülhetjük a hiányzó értékeket, ezeket közvetlenül fájlba írhatjuk (anélkül, hogy a nagy mátrixot valaha is teljes egészében tárolnunk kellene a memóriában)
 - Ajánló rendszerek esetében akár az elemek 99%-a is lehet hiányzó elem (nagyon sok termék!)
- Matematikai háttér: gradiens módszer (gradient descent), parciális deriváltak... → sok különböző változat