# Recomender system

Maciej Szczutko Ewa Stebel

2023-03-15

## Spis treści

**1 Wprowadzenie.** **1**
1.1 Opis danych . . . 1
1.2 Cel . . . 1
1.3 Użyte algorytmy . . . 2
1.4 Testowane metody wypełniania . . . 2

**2 Opis teoretyczny modelu** **2**
2.1 Singular value decomposition . . . 2
2.2 Truncated SVD . . . 2
2.3 SVD2 . . . 3
2.4 NMF . . . 3
2.5 SGD . . . 4

**3 Wyniki** **4**
3.1 Predykcja . . . 4
3.2 Wyniki SGD . . . 4

# 1 Wprowadzenie.

Projekt przedstawia prosty system rekomendacji filmów. Na podstawie danych historycznych, tj. poprzednich ocen użytkowników, chcemy przewidzieć jak oceniłby inne filmy. Taką informację moża wykorzystać do sugerowania kolejnych filmów dla danego użytkownika. Wykorzystamy głównie podstawowe metody aproksymacji oraz faktoryzacji macierzy. Aby móc weryfikować skuteczność naszych metod, podzielimy najpierw nasz zbiór obserwacji na dwa rozłączne zbiory. Dalej będziemy je nazywać zbiorem treningowym (na którym trenować będziemy nasz model) oraz testowym, który posłuży do weryfikacji wyników. Podziału dokonamy w taki sposób, aby zbiór treningowy zawierał około 90% ocen każdego użytkownika, a pozostałe trafiły do zbioru testowego. Jako kryterium jakości modelu użyjemy **RMSE** oraz sprawdzimy skuteczność predykcji. W projekcie nie będziemy skupiać się na szczegółach implementacji poszczególnych metod. Skorzystamy z gotowych implementacji z modułu ***sklearn***

## 1.1 Opis danych

Dane z których korzystamy pochodzą ze bazy **movielens**, która zawiera oceny 610 użytkowników, którzy ocenili łącznie 9724 filmy. Oczywiście, nie każdy użytkownik ocenił wszystkie filmy. Rating jest liczbą z zakresu 1-5 wraz z połówkami.

| userId | movieId | rating | timestamp |
|---:|---:|---:|---:|
| 1 | 1 | 4 | 964982703 |
| 1 | 3 | 4 | 964981247 |
| 1 | 6 | 4 | 964982224 |
| 1 | 47 | 5 | 964983815 |
| 1 | 50 | 5 | 964982931 |

## 1.2 Cel

Pierwszym celem jest znalezienie metody wypełniania macierzy, która dobrze estymuje prawdopodobne oceny użytkownika. Następnie zredukujemy wymiar naszych danych do minimum, tzn. tak aby by zachować możliwie wiele informacji. Przetestujemy niżej wspomniane algorytmy dla różnych metod wypełniania i postaramy się znaleźć dla nich optymalne parametry tzn. takie które minimalizują błąd średniokwadratowy.

## 1.3 Użyte algorytmy

Do predykcji ocen użyjemy metod faktoryzacji macierzy (**SVD**, **NMF**) oraz **SGD** jak zoptymalizowanej metody krokowej. Tę pierwszą przetestujemy również w zmodyfikowanym wariancie, jako metodę krokową. Jednak aby móc je zastosować, musimy najpierw uzupełnić brakujące wartości w zbiorze. Przetestujemy kilka metod wypełniania naszych danych.

## 1.4 Testowane metody wypełniania

Wyszliśmy z założenia, że metody powinny być możliwie proste i nielosowe. Wybraliśmy 6 metod:

1. Wypełnianie wszystkich pól macierzy tą samą liczbą $(0, 1, \dots, 5)$.
2. Wypełnianie pól macierzy średnią dla danego filmu.
3. Wypełnianie pól macierzy medianą dla danego filmu.
4. Wypełnianie pól macierzy średnią dla danego użytkownika.
5. Wypełnianie pól macierzy medianą dla danego użytkownika.
6. Wypełnianie pól macierzy kombinacją metod 2. i 4. tzn

$$\alpha \cdot \text{movieMean} + (1 - \alpha) \cdot \text{userMean},$$

dla pewnego parametru $\alpha \in (0, 1)$.

# 2 Opis teoretyczny modelu

Do naszego modelu nie potrzebujemy kolumny *timestamp*. Oryginalną ramke danych przekształcimy tak, aby utworzyła macierz $\boldsymbol{Z}$ o wymiarach $n \times d$ gdzie $i$-ty reprezentuje wszystkie oceny użytkownika o *userId* $i$. Na takiej macierzy będziemy mogli wykonać operacje.

## 2.1 Singular value decomposition

SVD jest techniką algebry liniowej, która służy do aproksymacji macierzy. Jest wykorzystywana głównie do zredukowania wymiaru oryginalnych danych przy jednoczesnym zachowaniu możliwie dużej ilości informacji.

**Definition 2.1** (Rozkład SVD)**.** Dekompozycja wartości osobliwych to metoda faktoryzacji macierzy, która rozkłada macierz $\boldsymbol{Z}$ na iloczyn trzech macierzy:

$$\boldsymbol{Z} = \boldsymbol{U}\boldsymbol{\Lambda}^{\frac{1}{2}}\boldsymbol{V}^T$$

gdzie:

- $\boldsymbol{U}$ i $\boldsymbol{V}$ to macierze ortogonalne (tj. $\boldsymbol{U}\boldsymbol{U}^T = \boldsymbol{U}^T\boldsymbol{U} = I$ i $\boldsymbol{V}\boldsymbol{V}^T = \boldsymbol{V}^T\boldsymbol{V} = I$), gdzie $\boldsymbol{U}$ i $\boldsymbol{V}$ to macierz wektorów własnych macierzy $\boldsymbol{Z}\boldsymbol{Z}^T$ oraz $\boldsymbol{Z}^T\boldsymbol{Z}$,
- $\boldsymbol{\Lambda}$ to macierz diagonalna, której elementy to ułożone malejąco wartości własne macierzy $\boldsymbol{Z}\boldsymbol{Z}^T$.

## 2.2 Truncated SVD

Z powyższej postaci macierzy A uzyskanej przez rozkład $SVD$ mamy $\boldsymbol{Z} = \boldsymbol{U}\Lambda^{\frac{1}{2}}\boldsymbol{V}^T$. Niech $\lambda_1 \geq \cdots \geq \lambda_n \geq 0$ będą wartościami własnymi $\boldsymbol{Z}\boldsymbol{Z}^T$. Oznaczmy przez $\boldsymbol{U}_r$ macierz złożoną z $r$ pierwszych kolumn $\boldsymbol{U}$, podobnie niech $\boldsymbol{V}_r$ oznacza $r$ kolumn $\boldsymbol{V}, r < d$. Dodatkowo, niech $\boldsymbol{\Lambda}_r$ będzie kwadratową podmacierzą wymiaru $r \times r$ macierzy $\boldsymbol{\Lambda}$. Zdefiniujmy $\widetilde{\boldsymbol{Z}}_r = \boldsymbol{U}_r\Lambda_r^{\frac{1}{2}}\boldsymbol{V}_r^T$. Wtedy

$$\left\|\boldsymbol{Z} - \tilde{Z}_r\right\|_F^2 = \sum_{i=r+1}^{d} \lambda_i.$$

Zatem o ile dla pewnego $r << d$ błąd $\sum_{i=r+1}^{d} \lambda_i$ jest niewielki, wtedy $\widetilde{\boldsymbol{Z}}_r$ jest dobrym przybliżeniem oryginalnej macierzy $\boldsymbol{Z}$.

Tę metodę będziemy nazywać po $SVD$. Teraz opiszemy pewną jej modyfikację.

## 2.3 SVD2

Ta metoda jest prostą modyfikacją powyższej. Sprowadza się do 3 kroków:

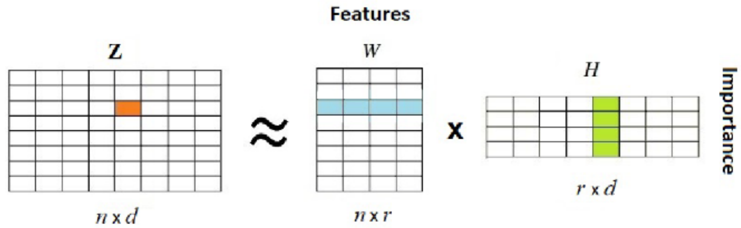
1. Użycie SVD na macierzy wypełnionej jedną z metod.
2. Nadpisanie uzyskanych wartości wartościami z macierzy treningowej.
3. Powatarzanie kroków 1. i 2. dopóki nie zostanie osiągnięta odpowiednia liczba iteracji.

Liczba iteracji, będzie kolejnym parametrem, który postaramy się dobrać aby uzyskać optymalne wyniki.

## 2.4 NMF

Non-negative matrix factorization jest metodą redukcji wymiaru, która rozkłada nieujemną macierz, na dwie macierze niższego rzędu, tak aby ich iloczyn przybliżał wartość wyjściowej macierzy. Obrazowo widać to na figurze 1. Możemy interpretować $\boldsymbol{W}$ jako zbiór cech, a $\boldsymbol{H}$ jako zbiór współczynników.

Niech $\boldsymbol{W}$ będzie nieujemną macierzą o rozmiarze $n$ x $r$, a $\boldsymbol{H}$ będzie nieujemną macierzą $r$ x $d$. Wtedy NMF podobnie jak SVD dąży do aproksymacji nieujemnej macierzy Z o rozmiarze $n$ x $d$ jako iloczyn macierzy $\boldsymbol{W}$ oraz $\boldsymbol{H}$. Za pomocą algorytmu NMF minimalizujemy odległości między macierzą Z, a iloczynem macierzy WH, a więc celem tej metody jest znalezienie




Rysunek 1: Graficzna interpretacja NMF.

$$arg\ min_{WH}\{dist(Z, WH)\},$$

gdzie $\boldsymbol{W}, \boldsymbol{H} \geq 0$.

Miarą najczęściej używaną do obliczenia odległości między macierzami jest metryka indukowana przez normę Frobeniusa:

$$\|Z\|_{Fro}^2 = \sum_{i,j} Z_{ij}^2.$$

## 2.5 SGD

W przypadku użycia poprzednich metod, pierwszym krokiem było uzupełnienie brakujących wartości macierzy. W tym przypadku będziemy badać następujący problem:

Dla macierzy $\boldsymbol{Z}$ o rozmiarze $n$ x $d$, chcemy znaleźć macierze $\boldsymbol{W}$ o rozmiarze $n$ x $r$ oraz $\boldsymbol{H}$ o rozmiarze $r$ x $d$, takie, że:

$$arg\ min_{WH} \sum_{i,j} (z_{ij} - w_i^T h_j)^2 + \lambda(\|w_i\|^2 + \|h_j\|^2),$$

gdzie $\lambda > 0$ jest parametrem, $h_j$ jest $j$-tą kolumną macierzy H, a $w_i^T$ jest $i$-tym wierszem W. Do wyznaczenia minimum tej funkcji użyjemy gradientu. W metodzie SGD dla każdej aktualizacji ocena $(i, j)$ jest losowo wybierana z macierzy Z, a odpowiednie zmienne $w_i$ i $h_j$ są zastępowane kolejno, przez

$$w_i \leftarrow w_i \eta(\frac{\lambda}{|Z_i|} w_i - R_{ij} h_j)$$

$$h_j \leftarrow h_j \eta(\frac{\lambda}{|Z_j|} h_j - R_{ij} w_i),$$

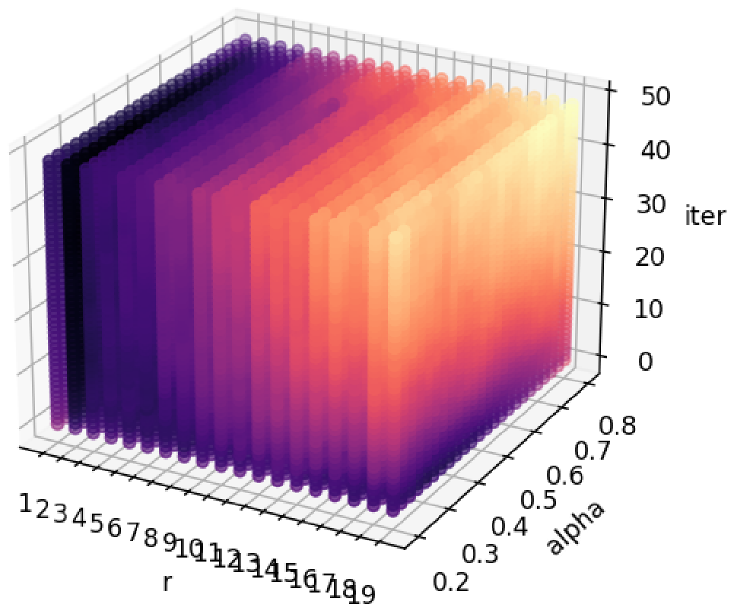
gdzie $R_{ij} = A_{ij} - w_i^T h_j$, a $\eta$ to learning rate.

# 3 Wyniki

Większość z zaproponowanych przez nasz metod wypełniania sprawdzało się dość słabo. Najgorzej wypadało wypełnianie ustaloną liczbą, a szczególnie wartościami brzegowymi (0 i 5). Dla poszczególnych metod RMSE często było większe niż 1. Metody z medianą względem kolumn/wierszy gwarantowały RMSE na poziomie 0.9 bądź nieco wyższym, ale juz mniejszym od 1. Najlepsze okazała się metoda numer 6, czyli ta która uwzględniała średnią globalną danego filmu, ale też informację o tym jak dany użytkownik oceniał inne filmy. Na niej skupiliśmy sie później podczas doboru parametrów dla algorytmów.

Prezentowane wyniki zostały uzyskane jako średnie dla kilku różnych par zbiorów testowych i treningowych. Z pierwszych trzech najlepiej sprawdził się algorytm SVD2. Z pomocą wykresu 2 szybko ustaliśmy parametry $\alpha = 0.4$ oraz r=2, max_iter = 40.

Na wykresach 4, 5 możemy zobaczyć jak wyglądały wyniki dla innych metod wypełniania. Jak widzimy tendencja zostawała zachowana, ale RMSE różniło się znacząco w zależności od metody poczatkowego wypełniania.




Rysunek 2: Wykres RMSE SVD2 w zależności od 3 parametrów.

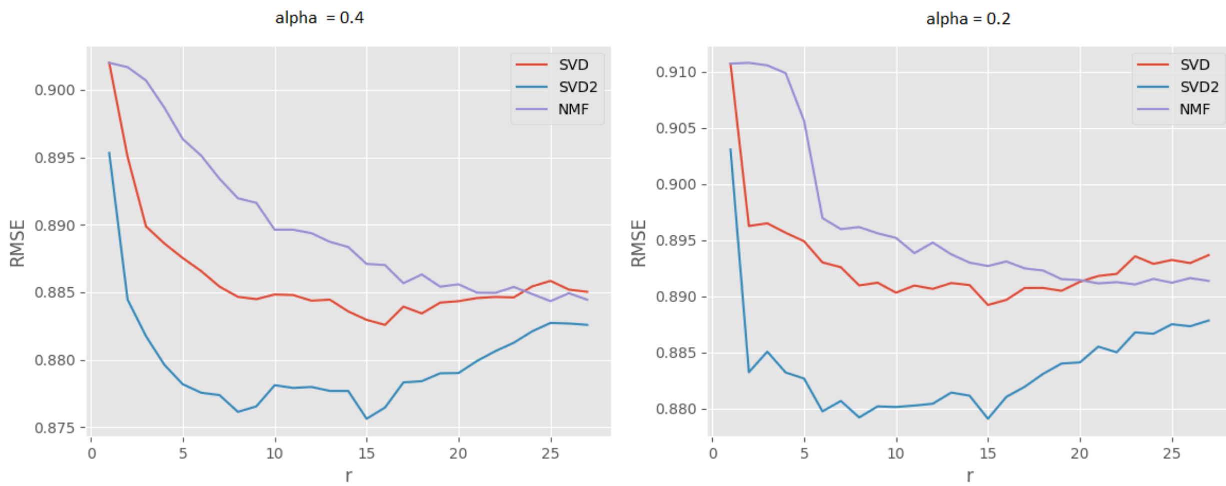
## 3.1 Predykcja

Sprawdziliśmy jak nasz algorytm przewiduje oceny użytkowników. Przypomnijmy, że nasze rating mógł przyjąć 10 możliwych wartości. Tak więc każdy wynik wyższy niż 10% należy uznać za dobry. Predykcja dla większości wariantów oscylowała w okolicy 25%. Rezultaty dla predykcji były analogiczne jak dla RMSE, tzn. jeśli dany algorytm był najlepszy pod względem RMSE, to również radził sobie najlepiej z predykcją. Poglądowe wyniki przedstawia wykres 6.

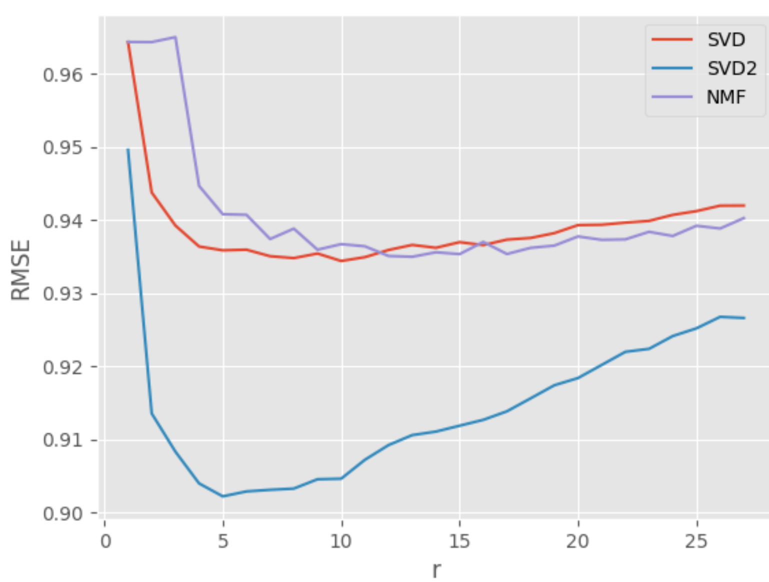
## 3.2 Wyniki SGD

W przypadku metody SGD parametry takie jak maksymalna liczba iteracji, oraz liczba iteracji bez zmian nie miały istotnego wpływu na końcową macierz, a RMSE bez względu na powyższe parametry wynosiło 1.4014, natomiast accuracy było równe 0.176.

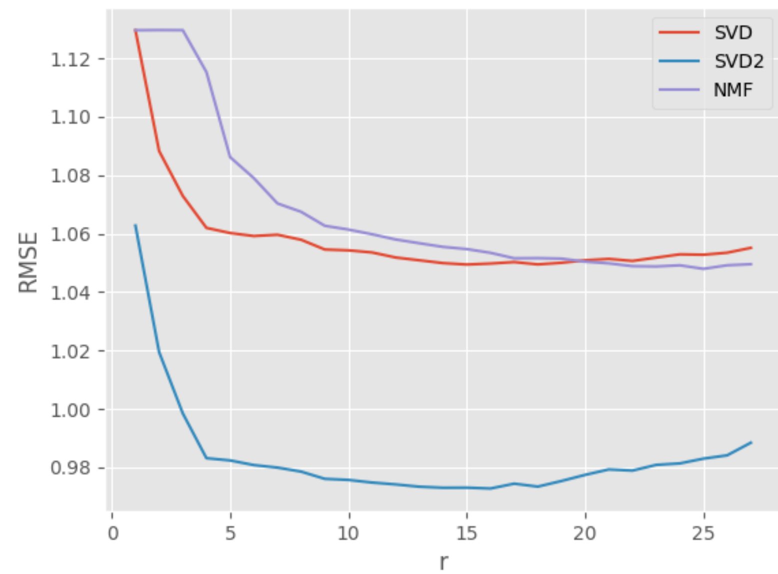
Wartość błędu RMSE jest zdeterminowana jednak przez wartość parametru $\lambda$, który jest odpowiedzialny za siłę regularyzacji. Za optymalną wartość parametru wybraliśmy $\lambda = 10$.Kolejnym istotnym parametrem jest learning rate - opisuje on szybkość aktualizowania wag podczas uczenia sie modelu. Przyjęta wartość



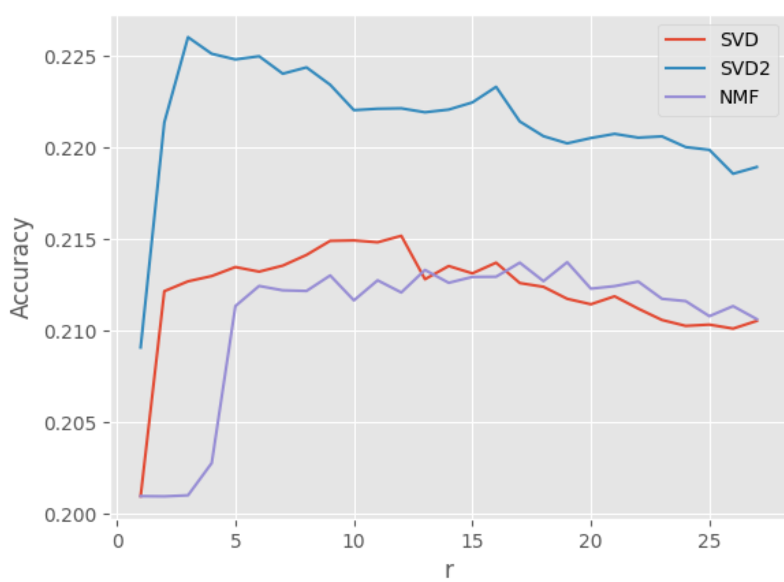

Rysunek 3: Wpływ parametru $\alpha$.




Rysunek 4: Wypełnianie medianą kolumn.




Rysunek 5: Wypelnienie trójkami.




Rysunek 6: Predykcja. Użyte $\alpha = 0.4$.

learning rate zmienia się wraz każdą kolejną iteracją i wynosi eta0 / pow(t, power_t), gdzie eta0=0.7, t- jest numerem kolejnej iteracji. a power_t=0.9.