Politechnika Łódzka

Wydział Fizyki Technicznej, Informatyki
i Matematyki Stosowanej

Kierunek: Matematyka Stosowana

Specjalność: Analiza Danych w Biznesie i Logistyce

Matematyczne modele wykorzystywane w systemach rekomendacji.

Anita Kudaj
Nr albumu: 220020

Praca magisterska napisana w Instytucie Matematyki
Politechniki Łódzkiej

Promotor: dr, mgr inż. Piotr Kowalski

Łódź, 07.2019

Spis treści

Rozdział 1

Wstęp

Znaczny wzrost ilości informacji dostępnych w internecie wywołał zapotrzebowanie na systemy, które mogą pomóc użytkownikom znaleźć cenne dla nich dane. W tym kontekście filtrowanie informacji zapewnia konkretne narzędzia - reguły rekomendujące, których celem jest uszeregowanie danych według częściowo ujawnionych preferencji, otrzymanych z uprzednio przeprowadzonych badaniach, aby w efekcie zasugerować odbiorcy odpowiedni dla niego produkt.

Celem tej pracy jest przedstawienie matematycznej strony metod systemów rekomendujących poprzez analizę algorytmów ukrytych pod ich nazwami.

Niniejsza praca składa się z czterech części. Pierwsza część zawiera wykaz oznaczeń, przypomnienie podstawowych definicji z zakresu algebry liniowej oraz rachunku prawdopodobieństwa i statystyki wykorzystanych w dalszych częściach pracy. W trzecim rozdziale przedstawione zostały elementy eksploracji danych stanowiące nieodzowną część systemów rekomendujących. Zaczynając od wstępnego przetwarzania danych oraz definicji i twierdzeń związanych z miarami podobieństwa i technikami redukcji wymiaru, tutaj szczególna uwaga została poświęcona rozkładowi według wartości osobliwych (ang. Singular Value Decomposition), przechodzimy do opisu dwóch metod eksploracji danych - algorytmu k-najbliższych sąsiadów i algorytmu k-średnich aby zakończyć oceną dokładności i jakości modeli. Rozdział czwarty opisuje modele tworzenia rekomendacji. Zostały tu zawarte sformułowania definicji związanych z systemami rekomendującymi, informacje dotyczące systemów opartych na treści, filtrowania kolaboratywnego oraz systemów kontekstowych. Rozdział został wzbogacony o algorytmy i przykłady, które pokazują zastosowanie poszczególnych metod. Ponadto znajdziemy tu przedstawienie dekompozycji macierzy ocen metodą SVD. Ostatnia część to praktyczne zastosowanie filtrowania kolaboratywnego. W rozdziale tym podstawę stanowią informacje na temat Apache Spark oraz algorytmu ALS (ang. Alternating Least Square) z biblioteki MLlib. Dodatkowo w ostatniej sekcji rozdziału została umieszczona

przykładowa implementacja algorytmu ALS wraz z komentarzami wyjaśniającymi poszczególne kroki.

Rozdział 2

Preliminaria

2.1 Oznaczenia używane w pracy

W niniejszej pracy zostały użyte następujące oznaczenia:
$\mathbb{N}$ - zbiór liczb naturalnych,
$\mathbb{R}$ - zbiór liczb rzeczywistych,
$\mathbb{K}$ - ciało liczb rzeczywistych lub zespolonych,
X - (duże, pochylone litery) jako oznaczenia zbiorów,
$\mathbf{x}$ - (małe, pogrubione litery) jako oznaczenia wektorów,
X - (duże litery) jako oznaczenia zmiennych losowych,
$\mathbb{X}$ - (duże litery z wyłączeniem $\mathbb{N}$, $\mathbb{R}$, $\mathbb{K}$) jako oznaczenia macierzy,
$[a_{ij}]_{j=1,\ldots,n}^{i=1,\ldots,m}$ - macierz o m wierszach i n kolumnach,
$[a_{ij}]$ - macierz kwadratowa,
$\mathbb{M}_{m\times n}(\mathbb{K})$ - zbiór wszystkich macierzy o wymiarach $m \times n$ i elementach z ciała $\mathbb{K}$,
$\mathcal{V}$ - przestrzeń liniowa,
$\operatorname{span}(X)$ - przestrzeń generowana przez zbiór X,
$(\Omega, \mathcal{F}, P)$ - przestrzeń probabilistyczna,
Ω - zbiór zdarzeń elementarnych,
$\mathcal{F}$ - rodzina podzbiorów zbioru Ω,
P - funkcja prawdopodobieństwa,
$\mathrm{B}(\mathbb{R}^n)$ - σ-ciało zbiorów borelowskich w $\mathbb{R}^n$,
$\mathrm{E}(\mathrm{X})$ - wartość oczekiwana,
$\operatorname{Cov}(X;Y)$ - kowariancja zmiennych losowych X, Y,
$\operatorname{Var}(\mathrm{X})$ - wariancją zmiennej losowej X,
$\sigma(\mathrm{X})$ - odchylenie standardowe zmiennej losowej X,
$\rho(\mathrm{X},\mathrm{Y})$ - współczynnik korelacji zmiennych losowych X, Y,
$\mathrm{d}(x,y)$ - odległość punktów x i y,

$\mathrm{d_e}(x, y)$ - odległość euklidesowa punktów x i y,
$\mathrm{d_r}(x, y)$ - odległość Minkowskiego punktów x i y,
$\mathrm{sim}(X, Y)$ - współczynnik podobieństwa wektorów X i Y,
$\rho^{\mathrm{P}}(\mathrm{X}, \mathrm{Y})$ - współczynnik korelacji Pearsona,
$\mathrm{J}(A, B)$ - indeks Jaccarda,
$\|\cdot\|_F$ - norma Frobeniusa.

2.2 Elementy algebry liniowej

W definicjach poniżej korzystamy z pojęcia ciała, którego wyjaśnienie odnajdziemy w książce Tadeusza Poredy i Jacka Jędrzejewskiego *Algebra liniowa z elementami geometrii analitycznej* [13, Sec 4.4].

Definicja 2.1 (Macierz [13, Sec 8.1 Def. 8.1])**.** *Niech $n, m \in \mathbb{N}$. Macierzą o m wierszach, n kolumnach (o wymiarach $m \times n$) i wyrazach w ciele $\mathbb{K}$ nazywamy funkcję*

$$\mathbb{A} : \{1, 2, \ldots, m\} \times \{1, 2, \ldots, n\} \to \mathbb{K}.$$

Wartością funkcji dla argumentu (i, j) jest element a_{ij} należący do ciała $\mathbb{K}$. Macierz zapisujemy w postaci tabeli

$$\mathbb{A} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}.$$

Przez $\mathbb{M}_{m\times n}(\mathbb{K})$ oznaczamy zbiór wszystkich macierzy o wymiarach $m \times n$ i elementach z ciała $\mathbb{K}$.

Definicja 2.2 (Wyznacznik macierzy [13, Sec 10.1, Def. 10.1])**.** *Niech $\mathcal{M}(\mathbb{K}) = \bigcup_{n\in\mathbb{N}} \mathbb{M}_{n\times n}(\mathbb{K})$ oznacza zbiór wszystkich macierzy kwadratowych o wyrazach z $\mathbb{K}$.*

Funkcję:

$$\det : \mathcal{M}(\mathbb{K}) \to \mathbb{K}$$

określamy następująco:

- *jeżeli $\mathbb{A} = [a_{11}]$, to $\det(\mathbb{A}) = a_{11}$,*
- *jeżeli $\mathbb{A} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix}$, gdy $n > 1$, to*

$$\det(\mathbb{A}) = \sum_{i=1}^{n} (-1)^{1+i} \cdot a_{i1} \cdot \det(\mathbb{A}_{i1}),$$

gdzie $\mathbb{A}_{ij}$ jest macierzą powstałą z macierzy $\mathbb{A}$ przez skreślenie i-tego wiersza i j-tej kolumny.

Funkcję det *nazywamy wyznacznikiem, natomiast wartość tej funkcji dla macierzy $\mathbb{A}$ wyznacznikiem macierzy $\mathbb{A}$.*

Uwaga 2.3 ([13, Sec 8.1])**.** *Przykładowe sposoby zapisu macierzy:*

$$[a_{ij}]_{j=1,\dots,n}^{i=1,\dots,m},\ (a_{ij})_{j=1,\dots,n}^{i=1,\dots,m},\ [a_{ij}]_{j\leqslant n}^{i=1\leqslant m},\ (a_{ij})_{j\leqslant n}^{i=1\leqslant m},\ [a_{ij}],\ (a_{ij}).$$

Sposobów $[a_{ij}]$, (a_{ij}) używamy, gdy liczba kolumn i wierszy danej macierzy jest ustalona.

W tej pracy używać będziemy zapisu $[a_{ij}]_{j=1,\dots,n}^{i=1,\dots,m}$ oraz zapisu $[a_{ij}]$ w przypadku macierzy kwadratowych.

Definicja 2.4 (Macierz transponowana [13, Sec 8.1])**.** *Niech $\mathbb{A} = [a_{ij}]_{j=1,\dots,n}^{i=1,\dots,m}$, będzie macierzą ze zbioru $\mathbb{M}_{m\times n}(\mathbb{K})$. Macierz $\mathbb{B} = [b_{ij}]_{j=1,\dots,m}^{i=1,\dots,n}$ nazywamy macierzą transponowaną macierzy $\mathbb{A}$, jeśli*

$$b_{ji} = a_{ij}$$

dla każdego $i \in \{1,\dots,n\}$ oraz $j \in \{1,\dots,m\}$. Piszemy wtedy $\mathbb{B} = \mathbb{A}^T$.

Uwaga 2.5 (Rodzaje macierzy [13, Sec 8.1, Sec 10.4])**.** *Poniżej zostały zdefiniowane niektóre rodzaje macierzy.*

- *Macierzą kwadratową nazywamy macierz, w której liczb wierszy i liczba kolumn są równe. Liczbę tę nazywamy stopniem macierzy kwadratowej.*
- *Macierzą diagonalną nazywamy macierz kwadratową $[a_{ij}]$, gdzie wszystkie elementy poza główną przekątną są równe 0. Macierz diagonalną oznaczamy $diag(a_{11}, a_{22}, \dots, a_{nn})$.*
- *Macierzą jednostkową stopnia n nazywamy macierz diagonalną, w której na głównej przekątnej wszystkie elementy są równe 1. Macierz jednostkową będziemy oznaczać $\mathbb{I}$.*
- *Macierz kwadratową $\mathbb{C}$, gdzie $\mathbb{C} = [c_{ij}]$, nazywamy macierzą ortogonalną, jeżeli spełniony jest warunek*

$$\mathbb{C}^T \cdot \mathbb{C} = \mathbb{C} \cdot \mathbb{C}^T = \mathbb{I}.$$

- *Macierzą nieosobliwą nazywamy macierz kwadratową, której wyznacznik jest różny od 0.*

- *Macierzą osobliwą nazywamy macierz kwadratową, której wyznacznik jest równy* 0.

Definicja 2.6 (Mnożenie macierzy [13, Sec 9.3 Def 9.13])**.** *Niech* $\mathbb{A} \in \mathbb{M}_{m\times n}(\mathbb{K})$ *i* $\mathbb{B} \in \mathbb{M}_{k\times m}(\mathbb{K})$*. Przyjmując następujące notacje:*

$$\mathbb{A} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}, \qquad \mathbb{B} = \begin{bmatrix} b_{11} & b_{12} & \cdots & b_{1m} \\ b_{21} & b_{22} & \cdots & b_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ b_{k1} & b_{k2} & \cdots & b_{km} \end{bmatrix}$$

iloczynem macierzy $\mathbb{B}$ *i* $\mathbb{A}$ *nazywamy taką macierz* $\mathbb{C} = [c_{lj}]_{j=1,\ldots,n}^{l=1,\ldots,k}$*, że*

$$\forall_{l\in 1,\ldots,k,\, j\in 1,\ldots,n}\, c_{lj} = \sum_{i=1}^{m} b_{li} \cdot a_{ij}.$$

Piszemy wtedy $\mathbb{C} = \mathbb{B} \cdot \mathbb{A}$*.*

Definicja 2.7 (Suma macierzy [13, Sec 8.1])**.** *Niech* $\mathbb{A}, \mathbb{B} \in \mathbb{M}_{m\times n}(\mathbb{K})$*. Sumą macierzy* $(\mathbb{B} + \mathbb{A})$ *nazywamy macierz* $\mathbb{C} \in \mathbb{M}_{m\times n}(\mathbb{K})$ *taką, że*

$$\forall_{i\in 1,\ldots,m,\, j\in 1,\ldots,n}\, c_{ij} = b_{ij} + a_{ij}.$$

Definicja 2.8 (Iloczyn macierzy przez element ciała [13, Sec 8.1])**.** *Niech* $\mathbb{A} \in \mathbb{M}_{m\times n}(\mathbb{K})$ *oraz* $\lambda \in \mathbb{K}$*. Iloczynem macierzy przez element z ciała* $(\lambda \cdot \mathbb{A})$ *nazywamy macierz* $\mathbb{C} \in \mathbb{M}_{m\times n}(\mathbb{K})$ *taką, że*

$$\forall_{i\in 1,\ldots,m,\, j\in 1,\ldots,n}\, c_{ij} = \lambda \cdot a_{ij}.$$

Twierdzenie 2.9 (Własności transpozycji macierzy [1, Sec 5.1 Tw. 5.1])**.** *Niech* $\mathbb{A} \in \mathbb{M}_{n\times m}(\mathbb{K})$*,* $\mathbb{B} \in \mathbb{M}_{n\times m}(\mathbb{K})$*,* $\mathbb{C} \in \mathbb{M}_{m\times n}(\mathbb{K})$ *oraz* $\lambda \in \mathbb{K}$*. Zachodzą następujące równości:*

- $(\mathbb{A}^T)^T = \mathbb{A}$,
- $(\mathbb{A} + \mathbb{B})^T = \mathbb{A}^T + \mathbb{B}^T$,
- $(\lambda\mathbb{A})^T = \lambda\mathbb{A}^T$,
- $(\mathbb{A}\mathbb{C})^T = \mathbb{A}^T\mathbb{C}^T$.

Definicja 2.10 (Ślad macierzy)**.** *[1, Sec 6.4] Śladem macierzy* $\mathbb{A} = [a_{ij}]$ *nazywamy wielkość:*

$$\operatorname{tr}(\mathbb{A}) = \sum_{i=1}^{n} a_{ii} = a_{11} + a_{22} + \cdots + a_{nn}.$$

Twierdzenie 2.11 (Własności śladu macierzy [1, Sec 6.4])**.** *Niech* $\mathbb{A}, \mathbb{B}, \mathbb{C} \in \mathbb{M}_{n\times n}(\mathbb{K})$ *oraz* $\lambda \in \mathbb{K}$*. Zachodzą następujące równości:*

- $\operatorname{tr}(\mathbb{A}+\mathbb{B}) = \operatorname{tr}(\mathbb{A}) + \operatorname{tr}(\mathbb{B})$,
- $\operatorname{tr}(\lambda\mathbb{A}) = \lambda \operatorname{tr}(\mathbb{A})$,
- $\operatorname{tr}(\mathbb{A}) = \operatorname{tr}\left(\mathbb{A}^T\right)$,
- $\operatorname{tr}(\mathbb{A}\mathbb{B}) = \operatorname{tr}(\mathbb{B}\mathbb{A})$,
- $\operatorname{tr}(\mathbb{A}\mathbb{B}\mathbb{C}) = \operatorname{tr}(\mathbb{C}\mathbb{A}\mathbb{B}) = \operatorname{tr}(\mathbb{B}\mathbb{C}\mathbb{A})$.

W kolejnych definicjach korzystamy z pojęć przestrzeni liniowej, wymiaru oraz przekształcenia liniowego, których wyjaśnienia możemy odnaleźć w książce *Algebra liniowa z elementami geometrii analitycznej* odpowiednio w [13, Sec 7.1], [13, Sec 7.5] oraz [13, Sec 9.1].

Uwaga 2.12 ([13, Sec 8.1])**.** *Wiersze macierzy o wymiarach $m \times n$ traktować możemy jako wektor z przestrzeni $\mathbb{K}^n$, natomiast kolumny jako wektor przestrzeni $\mathbb{K}^m$.*

Definicja 2.13 (Rząd kolumnowy i wierszowy macierzy [13, Sec 8.1])**.** *Niech $\mathbb{A} \in \mathbb{M}_{m\times n}(\mathbb{K})$. Rzędem kolumnowym macierzy $\mathbb{A}$ nazywamy wymiar przestrzeni $\mathbb{K}^n$ generowanej przez kolumny macierzy $\mathbb{A}$. Rząd ten oznaczamy symbolem $\mathrm{r_k}(\mathbb{A})$. Rzędem wierszowym macierzy $\mathbb{A}$ nazywamy wymiar podprzestrzeni generowany przez wiersze macierzy $\mathbb{A}$ i oznaczamy go $\mathrm{r_w}(\mathbb{A})$.*

Rząd wierszowy i kolumnowy danej macierzy są sobie równe.

Definicja 2.14 (Rząd macierzy [13, Sec 8.1])**.** *Rzędem macierzy $\mathbb{A}$ nazywamy wspólną wartość rzędu kolumnowego i wierszowego macierzy $\mathbb{A}$. Rząd macierzy oznaczamy symbolem $\operatorname{rz}(\mathbb{A})$.*

Niech $\mathcal{V}$ i $\mathcal{W}$ będą przekształceniami liniowymi nad ciałek $\mathbb{K}$.

Definicja 2.15 (Jądro przekształcenia liniowego [13, Sec 8.1])**.** *Jądrem przekształcenia liniowego $A : \mathcal{V} \to \mathcal{W}$ nazywamy zbór:*

$$\operatorname{Ker} A = \{\mathbf{x} \in \mathcal{V} : A(\mathbf{x}) = 0\}.$$

Definicja 2.16 (Obraz przekształcenia liniowego [13, Sec 8.1])**.** *Obrazem przekształcenia liniowego $A : \mathcal{V} \to \mathcal{W}$ nazywamy zbór:*

$$\operatorname{Im} A = \{\mathbf{y} \in \mathcal{W} : \exists_{\mathbf{x}\in\mathcal{V}}\, y = A(\mathbf{x})\}.$$

Definicja 2.17 (Przestrzeń generowana przez zbiór [13, Sec 7.1 Def 7.13])**.** *Niech X będzie dowolnym i niepustym podzbiorem przestrzeni liniowej $\mathcal{V}$. Podprzestrzenią generowaną przez zbiór X nazywamy zbiór wszystkich skończonych kombinacji liniowych wektorów ze zbioru X. Zbiór ten oznaczamy symbolem $\operatorname{span}(X)$.*

Symbolicznie zapisujemy zbiór $\operatorname{span}(X)$ *jako:*

$$\left\{x \in \mathcal{V} : \exists_{n\in\mathbb{N}}\, \exists_{(\alpha_1,\ldots,\alpha_n)\in\mathbb{K}^n}\, \exists_{(x_1,\ldots,x_n)\in X^n}\, (x = \alpha_1 \cdot x_1 + \ldots + \alpha_n \cdot x_n)\right\},$$

gdzie $\mathcal{V}$ *jest przestrzenią liniową nad ciałem liczb rzeczywistych lub ciałem liczb zespolonych* $\mathbb{K}$*.*

Definicja 2.18 (Wartość własna macierzy kwadratowej [13, Sec 12.2])**.** *Liczbę* $\lambda \in \mathbb{R}$ *nazywamy wartością własną macierzy kwadratowej* $\mathbb{A}$*, jeżeli istnieje niezerowy wektor* $\mathbf{x}$ *taki, że*

$$\mathbb{A}\mathbf{x} = \lambda\mathbf{x}.$$

Każdy niezerowy wektor $\mathbf{x}$ *spełniający powyższe równania nazywamy wektorem własnym macierzy* $\mathbb{A}$ *odpowiadającym wartości własnej* λ*.*

2.3 Elementy rachunku prawdopodobieństwa i statystyki

Definicja 2.19 (Przestrzeń probabilistyczna [6, Sec 1.2, Sec 1.4])**.** *Przestrzenią probabilistyczną nazywamy uporządkowaną trójkę* $(\Omega, \mathcal{F}, P)$*, gdzie:*

- Ω *to zbiór wszystkich zdarzeń elementarnych i* $\Omega \neq \varnothing$*,*
- $\mathcal{F}$ *rodzina podzbiorów zbioru* Ω *taka, że:*
  - $\varnothing \in \mathcal{F}$*,*
  - *jeżeli* $A \in \mathcal{F}$*, to* $\overline{A} = \Omega \backslash A \in \mathcal{F}$*,*
  - *jeżeli* $A_n \in \mathcal{F}$ *dla* $n = 1, 2, \ldots$*, to* $\bigcup_{n=1}^{\infty} A_n \in \mathcal{F}$*,*
- $P : \mathcal{F} \to [0, 1]$ *taka, że:*
  - $\forall_{A\in\mathcal{F}} P(A) \geqslant 0$*,*
  - $P(\Omega) = 1$*,*
  - *jeżeli* $A_n \in \mathcal{F}$*,* $n = 1, 2, \ldots$ *są takie, że* $A_i \cap A_j \neq \varnothing$ *dla* $i \neq j$ *to*

$$P(\bigcup_{n=1}^{\infty} A_n) = \sum_{n=1}^{\infty} P(A_n).$$

W poniższej definicji zostało użyte pojęcie σ-algebry, którego wyjaśnienie można odnaleźć w [6, Sec 1.2 Def. 1.2].

Definicja 2.20 ($\mathrm{B}(\mathbb{R}^n)$ [6, Sec 1.12]). *Niech $\mathcal{I}$ oznacza klasę wszystkich zbiorów, składających się ze skończonych sum rozłącznych zbiorów $I = I_1 \times I_2 \times \ldots \times I$, gdzie $I_k = [a_k, b_k)$. Najmniejszą σ-algebrę $\sigma(\mathcal{I})$ generowaną przez klasę zbiorów $\mathcal{I}$ nazywa się σ-algebrą borelowską zbiorów w $\mathbb{R}^n$ i oznacza się* $\mathrm{B}(\mathbb{R}^n)$.

Definicja 2.21 (Zmienna losowa [4, Sec 5.1 Def. 1]). *Odwzorowanie $\mathrm{X} : \Omega \to \mathbb{R}^n$ nazywamy zmienną losową o wartościach w $\mathbb{R}^n$, jeśli dla każdego $A \in \mathrm{B}(\mathbb{R}^n)$ zbiór $\mathrm{X}^{-1}(A) \in \mathcal{F}$.*

Definicja 2.22 (Wartość oczekiwana [4, Sec 5.6 Def. 2]). *Wartością oczekiwaną zmiennej losowej* X *o wartościach w* $\mathbb{R}$ *nazywamy liczbę:*

$$\mathrm{E}(\mathrm{X}) = \int_\Omega \mathrm{X} dP,$$

jeżeli X *jest P-całkowalna, tzn. jeżeli zachodzi:*

$$\mathrm{E}(\mathrm{X}) = \int_\Omega |\mathrm{X}| dP < \infty.$$

Definicja 2.23 (Kowariancja [6, Sec 2.8 Def.2.32]). *Kowariancją zmiennych losowych* X, Y *nazywamy liczbę:*

$$\mathrm{Cov}(\mathrm{X}; \mathrm{Y}) = \mathrm{E}\left((\mathrm{X} - \mathrm{E}(\mathrm{X}))(\mathrm{Y} - \mathrm{E}(\mathrm{Y}))\right).$$

Definicja 2.24 (Wariancja zmiennej losowej [6, Sec 2.8 Def.2.28]). *Wariancją zmiennej losowej* X *nazywamy liczbę:*

$$\mathrm{Var}(\mathrm{X}) = \mathrm{E}\left((\mathrm{X} - \mathrm{E}(\mathrm{X}))^2\right),$$

jeżeli wyznaczona wartość oczekiwana istnieje.

Definicja 2.25 (Odchylenie standardowe [6, Sec 2.8 Def.2.28]). *Odchyleniem standardowym zmiennej losowej* X *nazywamy liczbę:*

$$\sigma(\mathrm{X}) = \sqrt{\mathrm{Var}(\mathrm{X})}.$$

Definicja 2.26 (Współczynnik korelacji [6]). *Współczynnikiem korelacji nazywamy charakterystykę ilościową stopnia zależności dwóch zmiennych losowych* X *i* Y *zdefiniowaną następująco:*

$$\rho(\mathrm{X}, \mathrm{Y}) = \frac{\mathrm{Cov}(\mathrm{X}; \mathrm{Y})}{\sigma(\mathrm{X})\,\sigma(\mathrm{Y})}.$$

Rozdział 3

Elementy eksploracji danych wykorzystywane w systemach rekomendujących

Większość systemów rekomendujących opiera się na algorytmach, które możemy rozumieć jako różne techniki eksploracji danych. Zazwyczaj proces eksploracji danych składa się z trzech kroków:

1. wstępnego przetwarzanie danych,
2. analizy danych,
3. interpretacji wyników.

W tym rozdziale zostaną przeanalizowane najważniejsze i najczęściej używane w regułach rekomendujących metody. Zaczniemy od miar podobieństw i technik redukcji wymiaru. W kolejnym etapie spojrzymy na metody klasyfikacji, grupowania i regresji, aby zakończyć interpretacją wyników i oceną błędów obliczeń.

3.1 Wstępne przetwarzanie danych

Przed przystąpieniem do kroku analizy dane wymagają przygotowania. W tej sekcji zostaną zawarte informacje na temat wstępnego przetwarzania danych, które spotykamy przy regułach rekomendujących.

3.1.1 Miary podobieństwa

W systemach rekomendujących bardzo częstym podejściem jest używanie metod klasyfikacji i grupowania. Metody te opierają się na obliczaniu podobieństw i odległości.

Najprostszym i jednocześnie najczęściej używanym kryterium jest odległość euklidesowa.

Definicja 3.1 (Odległość euklidesowa [14]). *Niech* $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$, $n \in \mathbb{N}$. *Odległością euklidesową* $\mathbf{x}$ *i* $\mathbf{y}$ *nazywamy:*

$$\mathrm{d_e}(x, y) = \sqrt{\sum_{k=1}^{n} (\mathbf{x}_k - \mathbf{y}_k)^2}.$$

Warto również wspomnieć o uogólnionej wersji odległości euklidesowej - odległości Minkowskiego.

Uwaga 3.2. *Odległość Minkowskiego wyrażamy wzorem:*

$$\mathrm{d_r}(x, y) = (\sum_{k=1}^{n} |x_k - y_k|^r)^{\frac{1}{r}}.$$

W zależności od wartości stopnia odległości r *odległość Minkowskiego przyjmuje konkretne nazwy:*

- $r = 1$ - *odległość manhatan,*
- $r = 2$ - *wspomniana wcześniej odległość euklidesowa,*
- $r \to \infty$ - *supremum.*

Kolejnym podejściem, gdzie poszczególne elementy są postrzegane jako n - wymiarowe wektory, a podobieństwo między nimi jest obliczane na podstawie kąta, który tworzą jest odległość kosinusowa.

Definicja 3.3 (Odległość kosinusowa [14]). *Niech* $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$, $n \in \mathbb{N}$. *Odległością kosinusową nazywamy funkcję* $\mathrm{d} : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ *opisaną wzorem:*

$$\mathrm{d}(\mathbf{x}, \mathbf{y}) = 1 - \mathrm{sim}(\mathbf{x}, \mathbf{y}),$$

gdzie $\mathrm{sim} : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ *wyznacza współczynnik podobieństwa wektorów* $\mathbf{x}$ *i* $\mathbf{y}$ *według formuły:*

$$\mathrm{sim}(\mathbf{x}, \mathbf{y}) = \frac{\sum_{k=1}^{n} x_k y_k}{\sqrt{\sum_{k=1}^{n} x_k} \sqrt{\sum_{k=1}^{n} y_k}}.$$

Innym podejściem, które pozwala na modelowanie podobieństwa między zmiennymi losowymi jest korelacja Pearsona, którą definiujemy następująco:

Definicja 3.4 (Współczynnik korelacji Pearsona [14]). *Niech* X, Y *będą zmiennymi losowymi o rozkładach ciągłych oraz niech* $(x_1, \ldots, x_n)$, $(y_1, \ldots, y_n)$ *oznaczają losową próbę prostą. Przez* $\overline{x}$ *i* $\overline{y}$ *oznaczmy:*

$$\overline{x} = \frac{1}{n} \sum_{k=1}^{n} x_k, \ \overline{y} = \frac{1}{n} \sum_{k=1}^{n} y_k.$$

Wówczas współczynnikiem korelacji Pearsona nazywamy:

$$\rho^{\mathrm{p}}\left(\mathrm{X},\mathrm{Y}\right)=\frac{\sum_{k=1}^{n}(x_k-\overline{x})(y_k-\overline{y})}{\sqrt{\sum_{k=1}^{n}(x_k-\overline{x})^2}\sqrt{\sum_{k=1}^{n}(y_k-\overline{y})^2}}.$$

Przy innych rodzajach danych do opisu podobieństwa można użyć Indeku Jaccarda (współczynnik podobieństwa Jaccarda). Współczynnik ten mierzy podobieństwo między dwoma zbiorami, a definiujemy go niestepujący:

Definicja 3.5 (Indeks Jaccarda [3])**.** *Niech A i B oznaczają zbiory. Indeksem Jaccarda (podobieństwem Jaccrda) nazywamy funkcję:*

$$\mathrm{J}\left(A,B\right)=\frac{|A\cap B|}{|A\cup B|}.$$

3.1.2 Redukcja wymiaru

Zbyt duża ilość zmiennych, które opisują obserwacje, powoduje wzrost prawdopodobności, że zmienne te są ze sobą skorelowane, a informacje wnoszone przez część zmiennych są redundantne. W poniższym rozdziale przyjrzymy się najczęściej wybieranemu algorytmom redukcji wymiarów w kontekście reguł rekomendujących. Jest to rozkład według wartości osobliwych (ang. Singular Value Decomposition (SVD)).

Definicja 3.6 (Rozkład według wartości osobliwych [12])**.** *Rozkładem według wartości osobliwych $m\times n$ - wymiarowej macierzy $\mathbb{X}$, gdzie $m\geqslant n$ nazywamy odszukanie takich macierzy $\mathbb{U},\Sigma,\mathbb{V}$, że:*

$$\mathbb{X}=\mathbb{U}\Sigma\mathbb{V}^T,$$

gdzie:

- $\mathbb{U}^T\mathbb{U}=\mathbb{I}$, $\mathbb{V}^T\mathbb{V}=\mathbb{I}$, $\mathbb{U}$ *jest wymiaru* $m\times m$ *oraz* $\mathbb{V}$ *wymiaru* $n\times n$,
- Σ *jest macierzą wymiaru* $m\times n$, *gdzie* $\sigma_{ij}=0$, *gdy* $i\neq j$.

Uwaga 3.7. *[12] Niezerowe wyrazy macierzy Σ nazywamy wartościami osobliwymi macierzy $\mathbb{X}$. Kolumny macierzy $\mathbb{U}$ i $\mathbb{V}$ nazywamy odpowiednio lewymi i prawymi wektorami szczególnymi macierzy $\mathbb{X}$.*

Twierdzenie 3.8. *Zawsze jest możliwe dokonać dekompozycji macierzy $\mathbb{X}$ do postaci $\mathbb{U}\Sigma\mathbb{V}^T$.*

Definicja 3.9 (Wartość osobliwa macierzy)**.** *Wartością osobliwą σ_k macierzy $\mathbb{X}$ nazywamy*

$$\sigma_k=\sqrt{\lambda_k},$$

gdzie λ_k, $k\in\mathbb{N}$ jest nieujemną wartością własną macierzy $\mathbb{X}^T\mathbb{X}$.

Uwaga 3.10. *Należy zauważyć, że macierz $\mathbb{X}^T\mathbb{X}$ jest macierzą symetryczną, a jej wartości własne należą do zbiory liczb $\mathbb{R}_+ \cup \{0\}$.*

W przypadku gdy wyrazy macierzy $\mathbb{X}$ są liczbami ze zbioru liczb zespolonych szukamy wartości własnych macierzy $\mathbb{X}^*\mathbb{X}$, gdzie $\mathbb{X}^*$ jest sprzężeniem hermitowskim macierzy, tzn. złożeniem operacji transpozycji i sprzężenia zespolonego ($\mathbb{X}^* = \overline{\mathbb{X}^T}$).

Definicja 3.11 (Norma Frobeniusa[12])**.** *Niech $\mathbb{A} \in \mathbb{M}_{m\times n}(\mathbb{R})$. Normą Frobeniusa nazywamy:*

$$\|\mathbb{A}\|_F = \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n}|a_{ij}|^2} = \sqrt{\mathrm{tr}\,(\mathbb{A}^T\mathbb{A})}.$$

Twierdzenie 3.12 (Warunki równoważne SVD [12])**.** *Niech rozkład według wartości osobliwych macierzy $\mathbb{X}$ będzie dany wzorem*

$$\mathbb{X} = \mathbb{U}\Sigma\mathbb{V}^T$$

gdzie

$$\mathbb{U} = [\mathbf{u}_1, \mathbf{u}_2, ..., \mathbf{u}_m],\ \mathbb{V} = [\mathbf{v}_1, \mathbf{v}_2, ..., \mathbf{v}_n],\ \Sigma = \begin{bmatrix} \sigma_1 & 0 & \dots & 0 \\ 0 & \sigma_2 & \dots & 0 \\ \dots & \dots & \ddots & \dots \\ 0 & 0 & \dots & \sigma_n \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 0 \end{bmatrix},$$

oraz $\sigma_1 \geqslant \sigma_2 \geqslant ... \geqslant \sigma_r > \sigma_{r+1} = ... = \sigma_n = 0$. Przez $\mathbf{u}_1, \dots, \mathbf{u}_m$ i $\mathbf{v}_1, \dots, \mathbf{v}_n$ oznaczamy kolumny macierzy $\mathbb{U}$ i $\mathbb{V}$. $Im(\mathbb{X})$ i $Ker(\mathbb{X})$ oznaczają zakres i jądro macierzy. Wtedy:

1. $\mathrm{rz}\,(\mathbb{X}) = r$, $\mathrm{Ker}\,\mathbb{X} = \mathrm{span}\,(\mathbf{v}_{r+1}, ..., \mathbf{v}_n)$, $\mathrm{Im}\,\mathbb{X} = \mathrm{span}\,(\mathbf{u}_1, \mathbf{u}_2, ..., \mathbf{u}_r)$,
2. $\mathbb{X} = \sum_{i=1}^{r} \mathbf{u}_i \cdot \sigma_i \cdot \mathbf{v}_i^T$,
3. $||\mathbb{X}||_F^2 = \sigma_1^2 + ... + \sigma_r^2$.

Uwaga 3.13. *Niech $\mathbb{X}$ będzie jak w twierdzeniu 3.12.*

Wtedy:

$$\mathbb{X} = \sum_{i=1}^{r} \mathbf{u}_i \cdot \sigma_i \cdot \mathbf{v}_i^T.$$

Dowód. Niech $\mathbb{X} \in \mathbb{M}_{3\times 2}$ oraz $\mathbb{X} = \mathbb{U}\Sigma\mathbb{V}^T$, gdzie macierze $\mathbb{U}, \Sigma, \mathbb{V}$ mają postać:

$$\mathbb{U} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix},\ \mathbb{V}^T = \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix},\ \Sigma = \begin{bmatrix} \sigma_{11} & 0 \\ 0 & \sigma_{22} \\ 0 & 0 \end{bmatrix}.$$

Iloczyn $\mathbb{U}\Sigma$ jest równy

$$\mathbb{U}\Sigma = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \cdot \begin{bmatrix} \sigma_{11} & 0 \\ 0 & \sigma_{22} \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} a_{11}\sigma_{11} & a_{12}\sigma_{22} \\ a_{21}\sigma_{11} & a_{22}\sigma_{22} \\ a_{31}\sigma_{11} & a_{32}\sigma_{22} \end{bmatrix}.$$

Zatem :

$$\mathbb{U}\Sigma\mathbb{V}^T = \begin{bmatrix} a_{11}\sigma_{11} & a_{12}\sigma_{22} \\ a_{21}\sigma_{11} & a_{22}\sigma_{22} \\ a_{31}\sigma_{11} & a_{32}\sigma_{22} \end{bmatrix} \cdot \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} =$$

$$\begin{bmatrix} a_{11}\sigma_{11}b_{11} + a_{12}\sigma_{22}b_{12} & a_{11}\sigma_{11}b_{21} + a_{12}\sigma_{22}b_{22} \\ a_{21}\sigma_{11}b_{11} + a_{22}\sigma_{22}b_{12} & a_{21}\sigma_{11}b_{21} + a_{22}\sigma_{22}b_{22} \\ a_{31}\sigma_{11}b_{11} + a_{32}\sigma_{22}b_{12} & a_{31}\sigma_{11}b_{21} + a_{32}\sigma_{22}b_{22} \end{bmatrix}.$$

Z drugiej strony pokażemy że $\mathbb{X} = \sum_{i=1}^{r} \mathbf{u}_i \cdot \sigma_i \cdot \mathbf{v}_i^T$.

Niech $r = 2$.

Mamy, że:

1. $$\begin{bmatrix} a_{11} \\ a_{21} \\ a_{31} \end{bmatrix} \sigma_{11} \begin{bmatrix} b_{11} \\ b_{21} \end{bmatrix}^T = \begin{bmatrix} a_{11}\sigma_{11} \\ a_{21}\sigma_{11} \\ a_{31}\sigma_{11} \end{bmatrix} [b_{11}\, b_{21}] = \begin{bmatrix} a_{11}\sigma_{11}b_{11} & a_{12}\sigma_{22}b_{21} \\ a_{21}\sigma_{11}b_{11} & a_{22}\sigma_{22}b_{21} \\ a_{31}\sigma_{11}b_{11} & a_{32}\sigma_{22}b_{21} \end{bmatrix},$$

2. $$\begin{bmatrix} a_{12} \\ a_{22} \\ a_{32} \end{bmatrix} \sigma_{22} \begin{bmatrix} b_{12} \\ b_{22} \end{bmatrix}^T = \begin{bmatrix} a_{12}\sigma_{22} \\ a_{22}\sigma_{22} \\ a_{32}\sigma_{22} \end{bmatrix} [b_{12}\, b_{22}] = \begin{bmatrix} a_{12}\sigma_{22}b_{12} & a_{12}\sigma_{22}b_{22} \\ a_{22}\sigma_{22}b_{12} & a_{22}\sigma_{22}b_{22} \\ a_{32}\sigma_{22}b_{12} & a_{32}\sigma_{22}b_{22} \end{bmatrix}.$$

Sumując macierze powstałe w 1 i 2 otrzymujemy:

$$\mathbb{X} = \begin{bmatrix} a_{11}\sigma_{11}b_{11} + a_{12}\sigma_{22}b_{12} & a_{11}\sigma_{11}b_{21} + a_{12}\sigma_{22}b_{22} \\ a_{21}\sigma_{11}b_{11} + a_{22}\sigma_{22}b_{12} & a_{21}\sigma_{11}b_{21} + a_{22}\sigma_{22}b_{22} \\ a_{31}\sigma_{11}b_{11} + a_{32}\sigma_{22}b_{12} & a_{31}\sigma_{11}b_{21} + a_{32}\sigma_{22}b_{22} \end{bmatrix}.$$

Zatem, jeżeli $\mathbb{X} = \mathbb{U}\Sigma\mathbb{V}^T$, to $\mathbb{X} = \sum_{i=1}^{r} \mathbf{u}_i \cdot \sigma_i \cdot \mathbf{v}_i^T$. □

Uwaga 3.14. *Niech $\mathbb{X}$ będzie jak w twierdzeniu 3.12.*

Wtedy:

$$||\mathbb{X}||_F^2 = \sigma_1^2 + ... + \sigma_r^2.$$

Dowód. Niech $\mathbb{X} = \mathbb{U}\Sigma\mathbb{V}^T$. Mamy zatem

$$\|\mathbb{X}\|_F^2 = \left\|\mathbb{U}\Sigma\mathbb{V}^T\right\|_F^2.$$

Z definicji metryki Frobeniusa

$$\left\|\mathbb{U}\Sigma\mathbb{V}^T\right\|_F^2 = \operatorname{tr}\left((\mathbb{U}\Sigma\mathbb{V}^T)^T(\mathbb{U}\Sigma\mathbb{V}^T)\right).$$

Opierając się na własności $(\mathbb{AC})^T = \mathbb{A}^T\mathbb{C}^T$ transpozycji macierzy otrzymamy:

$$\left\|\mathbb{U}\Sigma\mathbb{V}^T\right\|_F^2 = \mathrm{tr}\left((\mathbb{U}\Sigma\mathbb{V}^T)^T(\mathbb{U}\Sigma\mathbb{V}^T)\right) = \mathrm{tr}\left(\mathbb{V}\Sigma^T\mathbb{U}^T\mathbb{U}\Sigma\mathbb{V}^T\right).$$

Skoro $\mathbb{U}^T\mathbb{U} = \mathbb{I}$, to

$$\mathrm{tr}\left(\mathbb{V}\Sigma^T\mathbb{U}^T\mathbb{U}\Sigma\mathbb{V}^T\right) = \mathrm{tr}\left(\mathbb{V}\Sigma^T\Sigma\mathbb{V}^T\right).$$

Następnie z własności śladu macierzy $\mathrm{tr}\,(\mathbb{ABC}) = \mathrm{tr}\,(\mathbb{CAB}) = \mathrm{tr}\,(\mathbb{BCA})$ wynika, że:

$$\mathrm{tr}\left(\mathbb{V}\Sigma^T\Sigma\mathbb{V}^T\right) = \mathrm{tr}\left(\mathbb{V}^T\mathbb{V}\Sigma^T\Sigma\right).$$

Skoro $\mathbb{V}^T\mathbb{V} = \mathbb{I}$, to

$$\mathrm{tr}\left(\mathbb{V}^T\mathbb{V}\Sigma^T\Sigma\right) = \mathrm{tr}\left(\Sigma^T\Sigma\right).$$

Bezpośrednio możemy więc sformułować, że

$$\|\mathbb{X}\|_F^2 = \sigma_1^2 + \sigma_2^2 + \ldots + \sigma_r^2.$$

□

Twierdzenie 3.15 (Twierdzenie Eckart - Younga [12])**.** *Niech* $\mathbb{X} \in \mathbb{M}_{m\times n}(\mathbb{R})$, $m \geqslant n$, $m, n \in \mathbb{N}$. $\mathbb{U}, \Sigma, \mathbb{V}$ *niech będą takie, że* $\mathbb{U} \in \mathbb{M}_{m\times m}(\mathbb{R})$, $\Sigma \in \mathbb{M}_{m\times n}(\mathbb{R})$, $\mathbb{V} \in \mathbb{M}_{n\times n}(\mathbb{R})$, $\mathbb{X} = \mathbb{U}\Sigma\mathbb{V}^T$, $\mathbb{U}^T\mathbb{U} = \mathbb{I}$, $\mathbb{V}^T\mathbb{V} = \mathbb{I}$ *oraz:*

$$\mathbb{U} = [\mathbf{u}_1, \ldots, \mathbf{u}_m],\ \Sigma = \begin{bmatrix} \sigma_1 & 0 & \ldots & 0 \\ 0 & \sigma_2 & \ldots & 0 \\ \ldots & \ldots & \ddots & \ldots \\ 0 & 0 & \ldots & \sigma_n \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & 0 \end{bmatrix},\ \mathbb{V} = [\mathbf{v}_1, \ldots, \mathbf{v}_n],$$

gdzie przez $\mathbf{u}_1, \ldots, \mathbf{u}_m$ *i* $\mathbf{v}_1, \ldots, \mathbf{v}_n$ *oznaczamy kolumny macierzy* $\mathbb{U}$ *i* $\mathbb{V}$*. (Istnienie powyższych macierzy wynika z twierdzenia 3.8 .)*

Zdefiniujmy dla $k \in \{1, \ldots, n\}$*:*

$$\mathbb{X}_k = \sum_{i=1}^{k} \mathbf{u}_i \cdot \sigma_i \cdot \mathbf{v}_i^T,$$

wtedy

$$||\mathbb{X} - \mathbb{X}_k||_F^2 = \min_{\mathrm{rz}(\mathbb{B})\leqslant k} ||\mathbb{X} - \mathbb{B}||_F^2 = \sigma_{k+1}^2 + ... + \sigma_n^2.$$

Aby udowodnić twierdzenie Eckart - Younga wprowadźmy twierdzenie Weylsa:

Twierdzenie 3.16 (Twierdzenie Weylsa [9, Tw. 4.17])**.** *Niech* $\mathbb{X}, \mathbb{Y} \in \mathbb{M}_{m\times n}(\mathbb{K})$*. Dodatkowo niech odpowiednio* $\sigma_1(\mathbb{X}) \geqslant \sigma_2(\mathbb{X}) \geqslant \ldots \geqslant \sigma_r(\mathbb{X}) \geqslant 0$*,* $\sigma_1(\mathbb{Y}) \geqslant \sigma_2(\mathbb{Y}) \geqslant \ldots \geqslant \sigma_r(\mathbb{Y}) \geqslant 0$ *i* $\sigma_1(\mathbb{Z}) \geqslant \sigma_2(\mathbb{Z}) \geqslant \ldots \geqslant \sigma_r(\mathbb{Z}) \geqslant 0$ *będą kolejnymi wartościami osobliwymi macierzy* $\mathbb{X}, \mathbb{Y}$*,* $\mathbb{Z} = \mathbb{X} + \mathbb{Y}$*. Wtedy:*

$$\sigma_{i+j-1}(\mathbb{Z}) \leqslant \sigma_i(\mathbb{X}) + \sigma_j(\mathbb{Y}),$$

gdzie $1 \leqslant i, j \leqslant r$*,* $i + j \leqslant r + 1$*.*

Uwaga 3.17. *Niech* $\mathbb{X}$ *będzie jak w twierdzeniu 3.12 oraz* $\mathbb{X}_k$ *będą jak w twierdzeniu 3.15.*

Wtedy:

$$\|\mathbb{X} - \mathbb{X}_k\|_F^2 = \sigma_{k+1}^2 + \sigma_{k+2}^2 + \ldots + \sigma_r^2.$$

Dowód. Niech $\mathbb{X} = \mathbb{U}\Sigma\mathbb{V}^T$ oraz $\mathbb{X}_k = \mathbb{U}\Sigma_k\mathbb{V}^T$. Mamy zatem

$$\|\mathbb{X} - \mathbb{X}_k\|_F^2 = \left\|\mathbb{U}\Sigma\mathbb{V}^T - \mathbb{U}\Sigma_k\mathbb{V}^T\right\|_F^2.$$

Stąd

$$\left\|\mathbb{U}\Sigma\mathbb{V}^T - \mathbb{U}\Sigma_k\mathbb{V}^T\right\|_F^2 = \left\|\mathbb{U}(\Sigma\mathbb{V}^T - \Sigma_k\mathbb{V}^T)\right\|_F^2 = \left\|\mathbb{U}(\Sigma - \Sigma_k)\mathbb{V}^T\right\|_F^2.$$

Z definicji metryki Frobeniusa

$$\left\|\mathbb{U}(\Sigma - \Sigma_k)\mathbb{V}^T\right\|_F^2 = \operatorname{tr}\left((\mathbb{U}(\Sigma - \Sigma_k)\mathbb{V}^T)^T(\mathbb{U}(\Sigma - \Sigma_k)\mathbb{V}^T)\right).$$

Opierając się na własności $(\mathbb{A}\mathbb{C})^T = \mathbb{A}^T\mathbb{C}^T$ transpozycji macierzy otrzymamy:

$$\operatorname{tr}\left((\mathbb{U}(\Sigma - \Sigma_k)\mathbb{V}^T)^T(\mathbb{U}(\Sigma - \Sigma_k)\mathbb{V}^T)\right) = \operatorname{tr}\left(\mathbb{V}(\Sigma - \Sigma_k)^T\mathbb{U}^T\mathbb{U}(\Sigma - \Sigma_k)\mathbb{V}^T\right).$$

Skoro $\mathbb{U}^T\mathbb{U} = \mathbb{I}$, to

$$\operatorname{tr}\left(\mathbb{V}(\Sigma - \Sigma_k)^T\mathbb{U}^T\mathbb{U}(\Sigma - \Sigma_k)\mathbb{V}^T\right) = \operatorname{tr}\left(\mathbb{V}(\Sigma - \Sigma_k)^T(\Sigma - \Sigma_k)\mathbb{V}^T\right).$$

Następnie z własności śladu macierzy $\operatorname{tr}(\mathbb{A}\mathbb{B}\mathbb{C}) = \operatorname{tr}(\mathbb{C}\mathbb{A}\mathbb{B}) = \operatorname{tr}(\mathbb{B}\mathbb{C}\mathbb{A})$ wynika, że:

$$\operatorname{tr}\left(\mathbb{V}(\Sigma - \Sigma_k)^T(\Sigma - \Sigma_k)\mathbb{V}^T\right) = \operatorname{tr}\left(\mathbb{V}^T\mathbb{V}(\Sigma - \Sigma_k)^T(\Sigma - \Sigma_k)\right).$$

Skoro $\mathbb{V}^T\mathbb{V} = \mathbb{I}$, to

$$\operatorname{tr}\left(\mathbb{V}^T\mathbb{V}(\Sigma - \Sigma_k)^T(\Sigma - \Sigma_k)\right) = \operatorname{tr}\left((\Sigma - \Sigma_k)^T(\Sigma - \Sigma_k)\right).$$

Bezpośrednio możemy więc sformułować, że

$$\|\mathbb{X} - \mathbb{X}_k\|_F^2 = \sigma_{k+1}^2 + \sigma_{k+2}^2 + \ldots + \sigma_r^2.$$

□

Dowód. [Twierdzenia 3.15 [9, Tw. 4.21]]

Niech $\mathbb{X} \in \mathbb{M}_{m\times n}(\mathbb{R})$ będzie macierzą o wartościach rzeczywistych, gdzie $m \geqslant n$. Załóżmy, że

$$\mathbb{X} = \mathbb{U}\Sigma\mathbb{V}^T$$

jest rozkładem według wartości osobliwych macierzy $\mathbb{X}$. Chcemy pokazać, że najlepszym przybliżeniem macierzy $\mathbb{X}$ w normie Frobeniusa (oznaczamy $||\cdot||_F$) jest

$$\mathbb{X}_k = \sum_{i=1}^{k} \mathbf{u}_i \cdot \sigma_i \cdot \mathbf{v}_i^T,$$

gdzie $\mathbf{u}_i$ i $\mathbf{v}_i$ oznaczają odpowiednio i-te kolumny macierzy $\mathbb{U}$ i $\mathbb{V}$.

Zauważmy, że z własności 2. twierdzenia 3.12 mamy

$$||\mathbb{X} - \mathbb{X}_k||_F^2 = ||\sum_{i=k+1}^{n} \mathbf{u}_i \cdot \sigma_i \cdot \mathbf{v}_i^T||_F^2.$$

Po obliczeniach wykonanych w uwadze 3.17 mamy, że

$$||\sum_{i=k+1}^{n} \mathbf{u}_i \cdot \sigma_i \cdot \mathbf{v}_i^T||_F^2 = \sum_{i=k+1}^{n} \sigma_i^2.$$

Niech $\mathbb{B}_k \in \mathbb{M}_{m\times n}(\mathbb{R})$, $\mathrm{rz}\,(\mathbb{B}_k) = k$. Stąd należy udowodnić, że

$$||\mathbb{X} - \mathbb{X}_k||_F^2 = \sum_{i=k+1}^{n} \sigma_i^2 \leqslant ||\mathbb{X} - \mathbb{B}_k||_F^2.$$

Niech $\mathbb{X}^{\text{‘}} = \mathbb{X} - \mathbb{B}_k, \mathbb{X}^{\text{“}} = \mathbb{B}_k$. Wtedy $\mathbb{X} = \mathbb{X}^{\text{‘}} + \mathbb{X}^{\text{“}}$.

Korzystając z Twierdzenia Weylsa otrzymujemy

$$\sigma_{i+j-1}(\mathbb{X}) \leqslant \sigma_i(\mathbb{X}^{\text{‘}}) + \sigma_j(\mathbb{X}^{\text{“}}).$$

Stąd

$$\sigma_{i+j-1}(\mathbb{X}) \leqslant \sigma_i(\mathbb{X} - \mathbb{B}_k) + \sigma_j(\mathbb{B}_k).$$

Niech $i \in \{1, \ldots, r-k\}$ oraz $j = k+1$. Wtedy

$$\forall_{i\in\{1,\ldots,r-k\}}\, \sigma_{i+k}(\mathbb{X}) \leqslant \sigma_i(\mathbb{X} - \mathbb{B}_k) + \sigma_{k+1}(\mathbb{B}_k).$$

Zauważmy, że

$$\sigma_{k+1}(\mathbb{B}_k) = 0,$$

wtedy

$$\forall_{i\in\{1,\ldots,r-k\}}\, \sigma_i(\mathbb{X} - \mathbb{B}_k) \geqslant \sigma_{k+1}(\mathbb{X}).$$

Stąd

$$||\mathbb{X}-\mathbb{B}_k||_F^2 = \sum_{i=1}^{r} \sigma_i(\mathbb{X}-\mathbb{B}_k)^2 \geqslant \sum_{i=1}^{r-k} \sigma_i(\mathbb{X}-\mathbb{B}_k)^2 \geqslant \sum_{i=1}^{r-k} \sigma_{k+1}(\mathbb{X})^2 \geqslant \sum_{i=k+1}^{r} \sigma_i(\mathbb{X})^2 = ||\mathbb{X}-\mathbb{X}_k||_F^2.$$

Ostatecznie

$$||\mathbb{X} - \mathbb{B}_k||_F^2 \geqslant ||\mathbb{X} - \mathbb{X}_k||_F^2.$$

□

3.2 Metody eksploracji danych

Termin eksploracja danych jest często używany jako określenie procesu odkrywania wiedzy z danych. Często jednak terminem "proces odkrywania wiedzy" określamy cały proces pracy z danymi, natomiast termin "eksploracja danych" odnosi się do etapu odkrywania pewnego rodzaju reguł.

Jak podaje Tadeusz Morzy [10] w metodach eksploracji można wyróżnić:

- odkrywanie asocjacji,
- klastrowanie,
- odkrywanie wzorców sekwencji,
- odkrywanie klasyfikacji,
- odkrywanie podobieństw w przebiegach czasowych,
- wykrywanie zmian i odchyleń.

W tej sekcji zostaną przedstawione te metody, które stosowane są najczęściej w regułach rekomendujących.

3.2.1 Algorytm k - najbliższych sąsiadów

Opis poniższego algorytmu oparty został na [14, Sec 2.3.1].

Algorytm k-najbliższych sąsiadów (k-NN) jest powszechnie używanym algorytmem klasyfikacji.

Przyporządkowanie nowych elementów zostaje przeprowadzone na podstawie porównania obserwacji z k najbardziej podobnymi jej obiektami ze zbioru treningowego. Podstawowa założenie algorytmu mówi, że jeżeli nowy rekord znajduje się w pewnym otoczeniu, to na podstawie k - najbliższych mu obserwacji zostanie przyporządkowana do niego klasa, której pojawienie się w rozważanym zbiorze jest najliczniejsze.

Niech q będzie punktem dla którego chcemy odnaleźć jego klasę l.
$X = \{\{x_1, l_1\}, \ldots, \{x_n, l_n\}\}$ niech będzie zbiorem treningowym, gdzie x_j jest j-tym elementem zbioru obserwacji, natomiast l_j etykietką klasy do której obserwacja należy, $j \in \{1, \ldots, n\}$.

Przeprowadzając algorytm k-NN zaczynamy od wyboru podzbioru

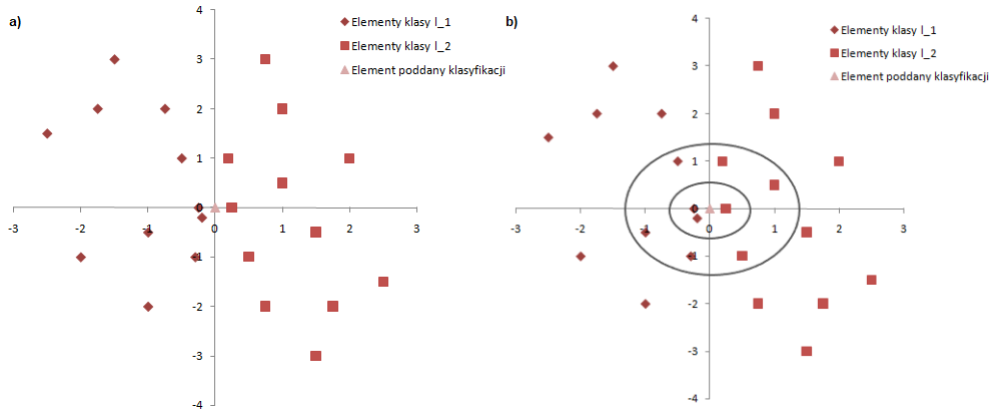
$$Y = \{\{y_1, l_1\}, \ldots, \{y_k, l_k\}\},$$

$k \in \{1, \ldots, n\}$ takiego, że $Y \in X$ oraz

$$\sum_{1}^{k} \mathrm{d}\left(q, y_k\right)$$

jest minimalna. Y zawiera więc k punktów z X, które leżą najbliżej rozważanego punktu q. Następnie do punktu q zostaje przyporządkowana klasa taka, że

$$l = \mathrm{f}\left(\{l_1, \ldots, l_k\}\right).$$



Rysunek 3.1: Metoda k - najbliższych sąsiadów. (źródło własne)

Na powyższych wykresach widzimy przykładowe zastosowanie algorytmu k-NN. Na pierwszym rysunku przedstawiony został zbiór treningowy z podziałem na dwie klasy (ręby, koła) oraz punkt (trójkąt), który będziemy chcieli przyporządkować do jednej z nich. Na drugim przedstawiono natomiast dwa okręgi, jedno prezentujące najbliższe sąsiedztwo dla $k = 3$, drugie dla $k = 9$. W obu przypadkach nowy punkt zostanie przyporządkowany do klasy l_1. Warto jednak zauważyć, że znajduje się on na granicy dwóch klastrów, przez co przy innym wyborze k może zostać przyporządkowany do klasy l_2.

Opisana metoda przyporządkowuje wybranemu rekordowi najbardziej mu podobne. Wykorzystuje do tego pewne miary odległości.

Najtrudniejszym zadaniem przy przeprowadzaniu algorytmu k-NN jest często wybór k. Jeżeli k będzie zbyt małe - klasyfikator stanie się bardzo wrażliwy, jeżeli jednak k będzie zbyt duże sąsiedztwo może zawierać zbyt dużo punktów z innych klas. Rozważając przypadek z rysunku 3.1 łatwo zauważyć, że nawet mała zmiana w obserwacjach zbioru treningowego może doprowadzić do zmiany wyniku.

3.2.2 Algorytm k - średnich

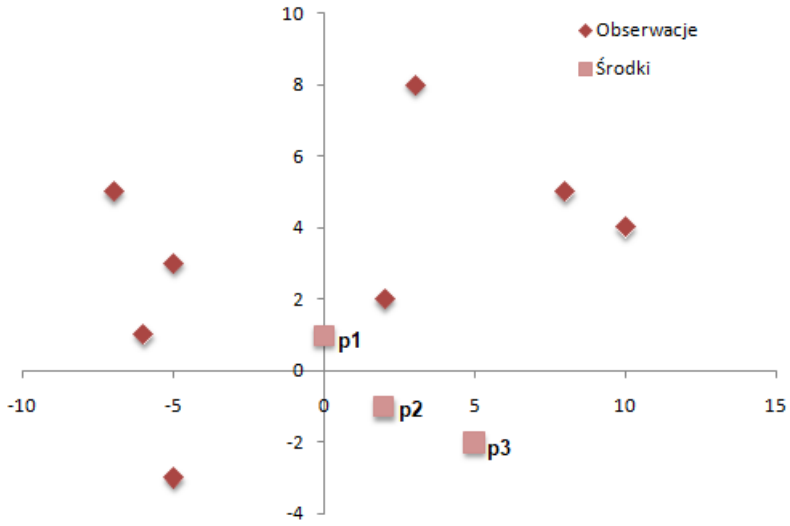
Opis poniższego algorytmu oparty został na [11, Sec 2.3.1].

Algorytm k-średnich jest prostym i zarazem efektywnym algorytmem grupowania. Głównym celem algorytmu jest podział pewnego zbioru X:

$$X = \{\mathbf{x}_i = (\mathbf{x}_{i1}, \ldots, \mathbf{x}_{id}) : i \ \in \{1, \ldots, N\}\},$$

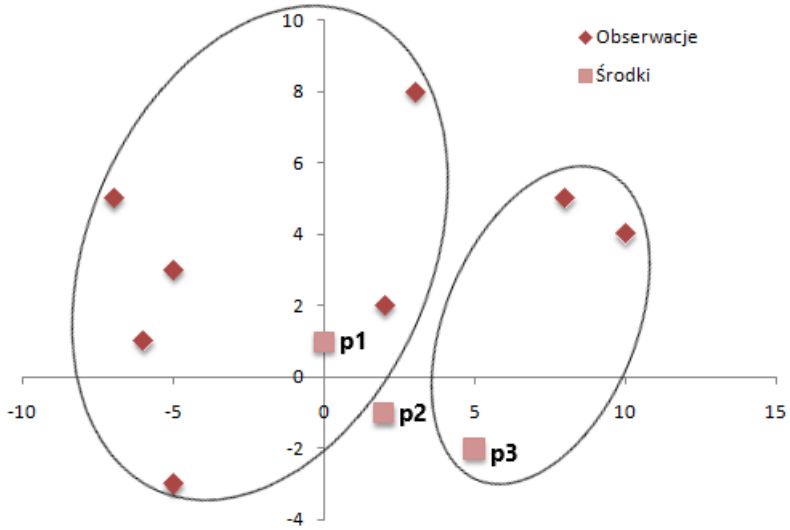
gdzie $\mathbf{x}_i$ jest d - wymiarowym wektorem cech opisującym obiekt na podzbiory.

W wyniku grupowania n - elementowego zbioru X na k podgrup otrzymujemy macierz podziału $\mathbb{A}$ o wymiarach $k \times n$. Każdy z elementów tej macierzy a_{ik} oznacza stopień w jakim wektor $\mathbf{x}_k$ przynależy do grupy. Na wstępie algorytmu ustalamy wartość parametru k jako liczbę grup, które chcemy wyodrębnione. Wybieramy k reprezentantów, które stanowią prototypy grup.



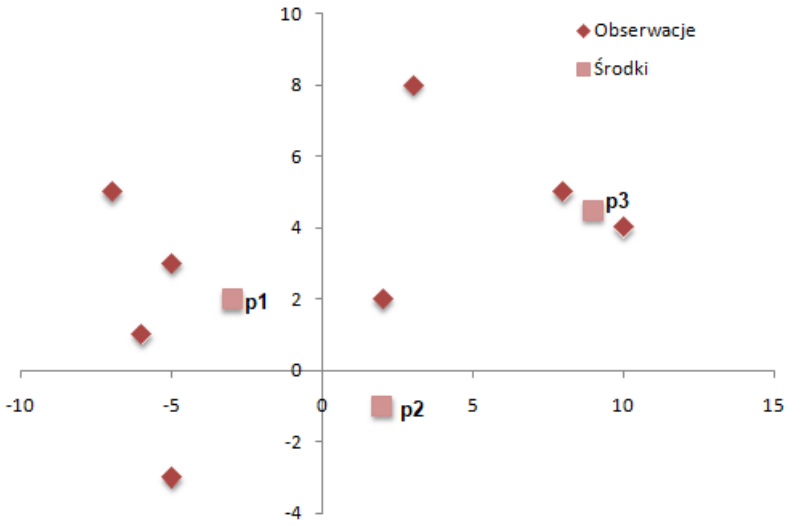
Rysunek 3.2: Metoda k-średnich. Wybór początkowych środków. (źródło własne)

W powyższym przykładzie (rysunek 3.2) wybranymi środkami są punkty $p1$, $p2$, $p3$. Kolejnym krokiem jest przypisanie każdego z elementów do najbliższej mu grupy.



Rysunek 3.3: Metoda k-średnich. Przypisanie elementów do grup.(źródło własne)

Dla każdej z tak ustalonych grup obliczamy średnią arytmetyczną współrzędnych, które staną się kolejnymi środkami.



Rysunek 3.4: Metoda k-średnich. Wybór nowych środków.(źródło własne)

Kroki te są wykonywane do momentu występowania migracji między obiektami. W algorytmie k-średnich liczba grup pozostaje więc niezmienną, zmienna jest tylko przynależność do grup. W metodzie tej poszukiwanie optymalnego podziału odpowiada wyznaczaniu takich grup, które minimalizują następującą funkcje kryterialną:

$$\mathrm{J}\left(\mathbf{p}, \mathbb{A}\right) = \sum_{i=1}^{k} \sum_{k=1}^{N} a_{ki}\, \mathrm{d}\left(\mathbf{p}_i, \mathbf{x}_k\right)^2,$$

gdzie

- $\mathrm{d}\left(\mathbf{p}_i, \mathbf{x}_k\right)$ oznacza odległość elementu reprezentowanego przez wektor $\mathbf{x}$ od grupy wyznaczonej przez środek $\mathbf{p}$,
- N to liczebność zbioru X,
- $\mathbb{A}$ oznacza macierz podziału.

3.3 Szacowanie błędów obliczeń

3.3.1 Ocena dokładności metody

Najczęściej używanymi miarami dokładności modelu są:

- błąd średni (Mean Error),
- średni błąd bezwzględny (Mean Absolute Error),
- średni błąd kwadratowy (Mean Squared Error).

Niech dla elementu i ze zbioru $P = \{x_1, \ldots, x_n\}$ będzie dostarczona predykcja $\widehat{r}_i$. Aby ocenić jakość jej wyniku należy porównać ją ze znaną wartością r_i.

Definicja 3.18 (Błąd średni [14, Sec 4.1.1])**.** *Średnim błędem nazywamy wartość wyrażenia:*

$$ME = \frac{1}{|P|} \sum_{x_i \in P} (\widehat{r}_i - r_i).$$

Definicja 3.19 (Średni błąd bezwzględny [14, Sec 4.1.1])**.** *Średnim błędem bezwzględnym nazywamy wartość wyrażenia:*

$$MAE = \frac{1}{|P|} \sum_{x_i \in P} |\widehat{r}_i - r_i|.$$

Definicja 3.20 (Średni błąd kwadratowy [14, Sec 4.1.1])**.** *Średnim błędem kwadratowym nazywamy wartość wyrażenia:*

$$MSE = \frac{1}{|P|} \sum_{x_i \in P} (\widehat{r}_i - r_i)^2.$$

Uwaga 3.21. *[14, Sec 4.1.1] Funkcja kwadratowa jest funkcją wypukłą co pozwala na dość częste zastępowanie średniego błędu kwadratowego przez średnią kwadratową błędów (Root Mean Squared Error (RMSE)):*

$$RMSE = \sqrt{MSE}$$

Normalized RMSE (NRMSE) oraz Normalized MAE (NMAE) są znormalizownymi, przez użycie zakresu wartości $r_{max} - r_{min}$, wersjami błędów RMSE i MAE.

Kolejnym rodzajem powszechnie używanego błędu, który pozwala na użycie sum ważonych, jest średni błąd RMSE (Average RMSE).

Definicja 3.22 (Średni błąd RMSE [14, Sec 4.1.1])**.** *Niech $w_i > 0$ będzie wagę dla elementu i oraz niech $\sum w_i = 1$.*

Średnim błędem RMSE nazywamy wartość wyrażenia:

$$ARMSE = \sqrt{\sum_{x_i \in P} w_i (\widehat{r}_i - r_i)^2}.$$

3.3.2 Ocena jakości modelu

Ocenę jakości modelu przeprowadza się na zbiorze testowym. Dla każdego z rekordów jest znana jego etykieta. Rekordy te są poddawane działaniu modelu, a następnie etykiety przypisane rekordom przez model są porównywalne z rzeczywistymi wartościami etykiet.

W następnym kroku zliczana jest liczba rekordów poprawnie i niepoprawnie zaklasyfikowanych przez model, a wynik testu zostaje przestawiony w postaci macierzy pomyłek.

Definicja 3.23 (Macierz pomyłek [10, Sec 4.8.1])**.** *Macierzą pomyłek nazywamy macierz kwadratową $m \times m$ (m oznacza liczbę etykiet), gdzie wiersze opisują etykiety praktyczne, natomiast kolumny etykiety przyporządkowane rekordom przez model. Element macierzy - $f_{i,j}$ oznacza liczbę rekordów z etykietą E_i, którym została przypisana etykieta E_j.*

$\mathbb{F}$	E_1	E_2
E_1	f_{11}	f_{12}
E_2	f_{21}	f_{22}

Tabela 3.1: Macierz pomyłek.

Uwaga 3.24. *[10, Sec 4.8.1] Często elementy macierzy pomyłek dla problemów klasyfikacji binarnej oznacza się symbolami : TP, TN, FN, FP. Oznaczenia te symbolizują cztery możliwe przypadki występujące w klasyfikacji binarnej. Załóżmy, że wyróżniamy klasę pozytywną (+) i negatywną (-). Wtedy :*

- *TP (ang. true - positive) - liczba pozytywnych rekordów testowych zaklasyfikowanych do klasy pozytywnej,*
- *FN (ang. false - negative) - liczba pozytywnych rekordów testowych zaklasyfikowanych do klasy negatywnej,*

- *FP (ang. false - positive) - liczba negatywnych rekordów testowych zaklasyfikowanych do klasy pozytywnej,*
- *TN (ang. true - negative) - liczba negatywnych rekordów testowych zaklasyfikowanych do klasy negatywnej.*

Macierz pomyłek przyjmuje wtedy postać:

$\mathbb{F}$	$+$	$-$
$+$	TP	FN
$-$	FP	TN

Tabela 3.2: Macierz pomyłek - przypadek klasyfikacji binarnej.

Poprzez analizę macierzy pomyłek bez problemu obliczymy łączną liczbę rekordów zaklasyfikowanych poprawnie oraz rekordów przypisanych błędnie przez klasyfikator.

Analizę zawartości macierzy można rozszerzyć o dodatkową informację - koszt błędnej klasyfikacji (ang. misclassification cost).

Definicja 3.25 (Koszt błędnej klasyfikacji [10, Sec 4.8.1])**.** *Oznaczmy przez e_{ij} koszt błędnego zaklasyfikowania do klasy E_j rekordu, który w rzeczywistości należy do klasy E_i. Koszt poprawnej klasyfikacji oznaczmy przez e_{ii} oraz załóżmy, że $\forall_i\ e_{ii} = 0$. Dodatkowo niech f_t oznacza liczbę wszystkich przykładów testowych, f_p liczbę poprawnie zaklasyfikowanych rekordów testowych oraz $f_p = \sum_{i=1}^{m} f_{ii}$, f_b niech natomiast oznacza liczbę błędnych klasyfikacji i $f_b = f_t - f_p$.*

Kosztem błędnej klasyfikacji $E(f_b)$ nazywamy sumę:

$$E(f_b) = \sum_{i=1}^{m}\sum_{j=1}^{m} f_{ij} \cdot e_{ij}.$$

W przypadkach, gdy błędne zaklasyfikowania rekordów nie różnią się kosztami, do oceny jakości klasyfikatora można wykorzystać miary takie jak trafność klasyfikacji (ang. accuracy) oraz błąd klasyfikacji (ang. error rat).

Definicja 3.26 (Trafność klasyfikacji [10, Sec 4.8.1])**.** *Trafnością klasyfikacji nazywamy stosunek liczby popranie zaklasyfikowanych rekordów testowych do łącznej liczby rekordów testowych:*

$$TR = \frac{f_p}{f_t} = \frac{\sum_{i=1}^{m} f_{ii}}{f_t}.$$

Definicja 3.27 (Błąd klasyfikacji [10, Sec 4.8.1])**.** *Błędem klasyfikacji nazywamy stosunek liczby błędnie zaklasyfikowanych rekordów testowych do łącznej liczby rekordów testowych:*

$$BK = \frac{f_b}{f_t} = \frac{\sum_{i=1}^{m}\sum_{j=1}^{m} f_{ij}}{f_t} = 1 - TR.$$

Uwaga 3.28. *[10, Sec 4.8.1] Innymi miarami, które można wywnioskować bezpośrednio z macierzy pomyłek dla klasyfikacji binarnej (tabla 3.2) są:*

- *współczynnik TP (czułość):*

$$WTP = \frac{TP}{TP + FN},$$

- *współczynnik FP:*

$$WFP = \frac{FP}{FP + TN},$$

- *współczynnik TN (specyficzność):*

$$WTN = \frac{TN}{FP + TN},$$

- *precyzja:*

$$precyzja = \frac{TP}{TP + FP},$$

- *zwrot:*

$$zwrot = \frac{TP}{TP + FN}.$$

Rozdział 4

Modele tworzenia rekomendacji

W niniejszym rozdziale zajmiemy się formalnym zdefiniowanie zadania, które ukrywa się pod nazwą tworzenia rekomendacji. Do jego poprawnego określenia będą przydatne następujące pojęcia.

Definicja 4.1 (Przedmiot [5, Sec 1.3])**.** *Przedmiotem nazwiemy klasę obiektów tego samego typu, nierozróżnialnych dla obserwatora i reprezentowanych przez co najmniej jeden element. W dalszej części pracy zbiór przedmiotów będziemy oznaczać przez P.*

Przedmioty stanowią podstawową grupę elementów w rozważaniach systemach rekomendujących.

Definicja 4.2 (Użytkownik [5, Sec 1.3])**.** *Użytkownikiem nazywamy osobę zdolną do przedstawienia własnej oceny wybranego przedmiotu. W dalszej części pracy zbiór użytkowników będziemy oznaczać przez U.*

W pracy [5] użyty jest zawsze ten sam zbiór ocen, jednak łatwo możemy pokusić się o jego uogólnioną definicję.

Definicja 4.3 (Zbiór ocen [5, Sec 1.3])**.** *Podzbiór skończony $\{0, 1, \ldots, n\}$, $n \in \mathbb{N}$ nazywamy zbiorem ocen dla przedmiotów. W dalszej części pracy zbiór ocen będziemy oznaczać przez O.*

Definicja 4.4 (Macierz preferencji [5, Sec 1.3])**.** *Rozważmy zbiór przedmiotów o liczności n oraz grupę użytkowników o liczności m. Macierzą preferencji nazywamy macierz $\mathbb{O} \in \mathbb{M}_{n\times m}(O)$.*

Macierz preferencji $\mathbb{O}$ możemy utożsamiać z tabelą, która jako nazwy kolumn przyjmuje poszczególnych użytkowników, natomiast jako nazwy wierszy przyjmuje przedmioty. Wypełnienie tabeli stanowią oceny, które użytkownicy wystawili przedmiotom. Ideę pojęcia macierzy preferencji przedstawia poniższa tabela:

	U ż y t k o w n i c y
P	Oceny
r	Oceny
z	Oceny
e	Oceny
d	Oceny
m	Oceny
i	Oceny
o	Oceny
t	Oceny
y	Oceny

,

zastosowanie tego pojęcia możemy odnaleźć w przykładzie 4.1.

Z uwagi na to, że przedmioty jako wytwory świata rzeczywistego są niemożliwe do opisania za pomocą skończonej liczby cech rozważa się ich skończoną reprezentację nazywaną wektorem własności.

Definicja 4.5 (Własność [5, Sec 1.3])**.** *Własnością nazwiemy cechę wyrażoną za pomocą wartości liczbowej lub pewnej zmiennej kategorycznej, która reprezentuje cechę przedmiotu istotną dla użytkownika w procesie tworzenia oceny. Zbiór wszystkich własności w rozważanym modelu oznaczamy W. Dla każdej $w \in W$ poprzez V_w rozumiemy zbiór wszystkich dopuszczalnych wartości własności w.*

Definicja 4.6 (Kontekst [3, Sec 3])**.** *Kontekstem nazywamy wektor współistniejących własności, które odzwierciedlają chwilowy stan użytkownika oraz wpływają na wartości jego preferencji.*

Pod pojęciem kontekstu możemy uwzględnić czas, dzień tygodnia, pogodę, wiek użytkownika, stanowisko pracy i wiele innych.

Definicja 4.7 (Funkcja anotująca [5, Sec 1.3])**.** *Funkcją anotującą własność $w \in W$ nazwiemy funkcję*

$$a_w \colon P \to V_w.$$

Mając na uwadze, że zbiór W jest skończony (jak również zbiór P) można utożsamiać funkcję anotującą z wektorem o długości $|W|$ nazywanym wektorem własności.

Problem 4.1 (Problem tworzenia rekomendacji [5, Sec 1.3])**.** *Rozważmy pewien zbiór przedmiotów P, pewien zbiór użytkowników U oraz pewien zbiór ocen O. Niech ponadto R będzie funkcją taką, że:*

$$R : P \times U \to O.$$

Załóżmy, że dla funkcji R znane są wartości dla pewnych par przedmiotów i użytkowników. Naszym zadaniem jest zaproponowanie sposobu predykcji brakujących wartości funkcji R w sposób minimalizujący wybrany funkcjonał błędu zdefiniowanego przez kwadrat normy Frobeniusa.

Problem tworzenia rekomendacji możemy utożsamić z wypełnieniem macierzy preferencji $\mathbb{O}$ co zostało dokładniej opisane w sekcji 4.4.

Przyjrzyjmy się następującemu przykładowi, który ilustruje istotę problemu 4.1.

Przykład 4.1. *Niech zbiór przedmiotów będzie w tym przypadku zbiorem sześciu książek, $P = \{p_1, p_2, p_3, p_4, p_5, p_6\}$, gdzie p_i dla $i \in \{1, 2, 3, 4, 5, 6\}$ oznacza i - tą książkę. Zbiór użytkowników niech będzie zbiorem czytelników, $U = \{u_1, u_2, u_3, u_4, u_5, u_6\}$, gdzie u_i dla $i \in \{1, 2, 3, 4, 5, 6\}$ oznacza i - tego czytelnika.*
Poniższa tabela to macierz preferencji dla ustalonego zbioru P i ustalonego zbioru U. Znak '?' oznacza brakujące wartości funkcji R, zatem czytelnik danej książki nie czytał lub czytał lecz jego ocena jest nieznana.

Czytelnicy		$\mathbf{u_1}$	$\mathbf{u_2}$	$\mathbf{u_3}$	$\mathbf{u_4}$	$\mathbf{u_5}$	$\mathbf{u_6}$
Książki	$\mathbf{p_1}$	*6*	*3*	***?***	*6*	*4*	***?***
	$\mathbf{p_2}$	***?***	*6*	*6*	*5*	*6*	***?***
	$\mathbf{p_3}$	*7*	*7*	*8*	*7*	*8*	*9*
	$\mathbf{p_4}$	*8*	*10*	*10*	*7*	*6*	*8*
	$\mathbf{p_5}$	*9*	*6*	*6*	*6*	*6*	***?***
	$\mathbf{p_6}$	*5*	*7*	*7*	*5*	*4*	*2*

Zadaniem jest przewidzieć brakujące wartości funkcji R, czyli oceny, które mogą zostać nadane przez użytkowników w sposób minimalizujący błąd popełniany przez model.

Uwaga 4.8 (Podział systemów rekomendujących)**.** *Podziału systemów rekomendujących dokonujemy ze względu na zakres wykorzystywanych informacji. Wyróżniamy:*

- *systemy rekomendujące oparte na treści,*
- *filtrowanie kolaboratywne,*
- *systemy rekomendujące kontekstowe.*

W przypadku systemów rekomendujących opartych na treści predykcja jest dokonywana na podstawie ocen wystawionych przedmiotom przez użytkowników oraz wektorów własności rozważanych przedmiotów.

W filtrowaniu kolaboratywnym wektory cech zostają pominięte, a predykcja dokonywana jest na podstawie ocen. Wyróżniamy dwa typy filtrowania kolaboratywnego:

- *filtrowanie oparte na użytkownikach*

  Niech $\mathbf{u}_1 = [o_{1,1}, \ldots, o_{1,n}]$ *oraz* $\mathbf{u}_2 = [o_{2,1}, \ldots, o_{2,n}]$ *będą wektorami opisującymi odpowiednio użytkownika* u_1 *i* u_2*. Elementami wektorów są wartości funkcji* R *w przypadkach, gdzie zarówno użytkownik* u_1*, jak i* u_2 *ocenili ten sam przedmiot. Główne założenie filtrowania kolaboratywnego opartego na użytkownikach mówi, że jeżeli odległość między wektorami* $\mathbf{u}_1$ *i* $\mathbf{u}_2$ *jest mała oraz użytkownik* u_1 *ocenił pewien przedmiot, dla którego użytkownik* u_2 *jeszcze nie wystawił oceny, to prawdopodobnie ocena użytkownika* u_2 *będzie podobna do oceny użytkownika* u_1*.*

- *filtrowanie oparte na elementach*

  Niech $\mathbf{p}_1 = [o_{1,1}, \ldots, o_{1,n}]$ *oraz* $\mathbf{p}_2 = [o_{2,1}, \ldots, o_{2,n}]$ *będą wektorami opisującymi odpowiednio przedmiot* p_1 *i* p_2*. Elementami wektorów są wartości funkcji* R *w przypadkach, gdzie przedmiot* p_1*, jak i* p_2 *został oceniony przez tego samego użytkownika. Główne założenie filtrowania kolaboratywnego opartego na elementach mówi, że jeżeli odległość między wektorami* $\mathbf{p}_1$ *i* $\mathbf{p}_2$ *jest mała oraz użytkownik ocenił w pewien sposób przedmiot* p_1 *w przeszłości to będzie skłonny w podobny sposób ocenić przedmiot* p_2*.*

Systemy rekomendujące kontekstowe są natomiast systemami rekomendującymi opartymi na treści w których zostaje uwzględniony dodatkowy wymiar - kontekst.

4.1 Systemy rekomendujące oparte na treści - Content-based recommender systems

Rozważania zawarte w tej sekcji stanowią formalizację treści zawartych w książce Gorakala S. K.: *Building Recommendation Engines* [3, Sec 3] oraz opierają się na treści kursu autorstwa Andrew Ng: *Recommender Systems* [7], którego treść została również załączona do pracy na płycie CD.

Systemy oparte na treści wyróżnia ukierunkowanie na spersonalizowany poziom użytkownika oraz treść produktu. Metoda ta opiera się na obliczaniu podobieństw oraz wykorzystuje techniki uczenia maszynowego, takie jak klasyfikacja.

Algorytm 4.9. *W metodzie tej celem jest stworzenie rekomendacji i wygenerowania listy przedmiotów, które mogę być odpowiednie użytkownikowi. Opieramy się na treści rozważanych elementów. Algorytm tego rodzaju rekomendacji możemy przedstawić w następujących krokach:*

1. *stworzenie wektora własności* $\mathbf{w} = [w_1, \ldots, w_n]$*,* $n \in \mathbb{N}$*, gdzie* $\forall_{i \in \{1,\ldots,n\}}\, w_i \in W$*,*

2. *wygenerowanie profilów produktów - przedstawienie wektorów wartości* $\mathbf{w}_{p_i}$ *gdzie elementy wektora określają wartości funkcji anotującej poszczególnych własności* $w_1, \ldots, w_n$ *dla przedmiotu* p_i *dla* $i \in \mathbb{N}$*,*

3. *wygenerowanie profilów użytkowników - stworzenie wektorów własności* $\mathbf{w}_{u_j}$ *przypisanych użytkownikom, gdzie poszczególne elementy wektora określają preferencje użytkownika* u_j *dla* $j \in \mathbb{N}$ *w stosunku do elementów wektora własności* $\mathbf{w}$ *określonego w kroku 1.,*

4. *obliczmy ocenę* $\widehat{o}_{j,i}$ *jaką użytkownik* j *wystawiłby dla przedmiotu* i*, którego wcześniej nie oceniał, za pomocą funkcji*

$$\widehat{o}_{j,i} = \mathbf{w}_{u_j}^T \mathbf{w}_{p_i},$$

5. *porównując otrzymane w kroku 3. oceny, dokonujemy rekomendacji nowego przedmiotu.*

Algorytm w ogólny sposób wyjaśnia istotę problemu generowania profili przedmiotów i użytkowników. Istnieje wiele metod, które pozwalają na precyzyjne wyznaczanie wektorów $\mathbf{w}_{p_i}$, $\mathbf{w}_{u_j}$. Szczegółowe opisy wybranych metod zostaną zawarte w kolejnych sekcjach rozdziału.

4.1.1 Wygenerowanie profilu przedmiotu - algorytm TFIDF

Rozważania na temat algorytmu TFIDF zawarte w tym rozdziale zostały przeprowadzone na podstawie książki *Recommender Systems Handbook* [14, Sec 3.3.1.1].

W większość systemów rekomendacji opartych na treści używamy gotowych modeli wyszukujących. W przypadku rozważań przeprowadzanych na dokumentach tekstowych jednym z najbardziej popularnych jest model przestrzeni wektorowej (*ang. Vector Space Model*) z algorytmem TFIDF.

Warto na początku zaznaczyć, że przedmiotem p_i jest tu opisany przez dokument tekstowy (artykuł, książka), natomiast własnościami, które go charakteryzują są słowa. Mając na uwadze te założenia możemy zdefiniować kolejne elementy algorytmu $TFIDF$.

Niech $n \in \mathbb{N}$, $P = \{p_1, p_2, \ldots, p_n\}$ będzie zestawem analizowanych przedmiotów. $W = \{w_1, w_2, \ldots, w_n\}$, $n \in \mathbb{N}$ niech będzie zbiorem rozważanych własności.

Definicja 4.10 (Model przestrzeni wektorowej [14, Sec 3.3.1.1])**.** *Modelem przestrzeni wektorowej nazywamy formę reprezentacji przedmiotów, w której przedmiot* p_i *jest*

reprezentowany przez wektor z przestrzeni n-wymiarowej, a każdy z n wymiarów reprezentuje jedną z rozważanych własności przedmiotu.

Definicja 4.11 (Liczność [14, Sec 3.3.1.1])**.** *Licznością $f_{k,j}$ nazywamy liczbę wystąpień własności w_k w przedmiocie p_j.*

Definicja 4.12 (TF [14, Sec 3.3.1.1])**.** *TF (ang. term frequency) nazywamy macierz przedstawiającą zależność własności w_k od przedmiotu p_j:*

$$TF(w_k, p_j) = \frac{f_{k,j}}{\max_z f_{z,j}},$$

gdzie:

- $\max_z f_{z,i}$ *- maksymalna w odniesieniu do wszystkich wartości $w_z \in W$, $z \in \{1, \ldots, n\}$, które pojawiły się w przedmiocie p_i, liczność wystąpień własności.*

Definicja 4.13 (IDF [14, Sec 3.3.1.1])**.** *IDF (ang. inverse dokument frequency) nazywamy funkcję:*

$$IDF(w_k) = \log \frac{N}{n_k},$$

gdzie:

- *N - całkowita liczba przedmiotów w zbiorze P,*
- *n_k - liczba przedmiotów, w których własność w_k, $k \in \{1, \ldots, n\}$ wystąpiła przynajmniej raz.*

Definicja 4.14 (TFIDF [14, Sec 3.3.1.1])**.** *TFIDF (ang. TF – term frequency, IDF - inverse document frequency) nazywamy funkcję:*

$$TFIDF(w_k, p_i) = TF(w_k, p_i) \cdot IDF(w_k).$$

Definicja 4.15 (Waga własności w przedmiocie)**.** *Wagą własności w_k w przedmiocie p_i nazywamy wartość:*

$$s_{k,i} = \frac{TFIDF(w_k, p_i)}{\sqrt{\sum_{j=1}^{|W|} TFIDF(w_j, p_i)^2}}.$$

Uwaga 4.16. *Każdy z dokumentów p_i, $i \in \{1, \ldots, n\}$ przedstawiamy jako wektor wag własności (słowa) w_k w przedmiocie p_i. Zatem $p_i = [s_{1i}, s_{2i}, ..., s_{ni}]$.*

4.1.2 Wygenerowanie profilu użytkownika

Rozważania zawarte w tej sekcji zostały oparte na: *Building Recommendation Engines* [3, Sec 3].

W tym kroku tworzymy profil użytkownika pasujący do profilu produktu rozważając własności wspólne z treścią produktu jako, że stwarza to możliwość porównywanie profili użytkowników i przedmiotów.

Algorytm 4.17 ([3, Sec 3])**.** *Generowanie profilu użytkownika odbywa się w dwu krokach:*

1. *Stworzenie macierzy* $\mathbb{A} \in \mathbb{M}_{|P|\times|U|}$*, gdzie wyrazy macierzy przyjmują wartości ze zbioru* $\{0, 1\}$*:*
   - $a_{ij} = 0$*, gdy użytkownik* j *nie ocenił przedmiotu* i*,*
   - $a_{ij} = 1$*, gdy użytkownik* j *ocenił przedmiotu* i*.*
2. *Wygenerowanie macierzy profili użytkowników* $\mathbb{B}$

$$\mathbb{B} = TFIDF(w_k, p_i)^T \cdot \mathbb{A}$$

Przykład 4.2. *Niech wektor własności będzie określony następująco* $\mathbf{w} = [w_1, w_2]$*, a każdy z elementów wektora* $\mathbf{w}$ *niech reprezentuje inny gatunek. Dodatkowo zakładamy, że istnieje gatunek* w_0*, którego cechy reprezentują wszystkie książki oraz dla każdej z książek* $w_0 = 1$*. Poniższa tabela określa wartości funkcji anotującej dla poszczególnych książek:*

Gatunki		$\mathbf{w_1}$	$\mathbf{w_1}$	$\mathbf{w_2}$
Książki	$\mathbf{p_1}$	*1*	*0.8*	*0.01*
	$\mathbf{p_2}$	*1*	*1*	*0*
	$\mathbf{p_3}$	*1*	*0.88*	*0.02*
	$\mathbf{p_4}$	*1*	*0.2*	*0.9*
	$\mathbf{p_5}$	*1*	*0*	*1*
	$\mathbf{p_6}$	*1*	*0.7*	*0.2*

W celu wygenerowanie powyższych wartości macierzy możemy się posłużyć algorytmem generującym profil produktu z sekcji 4.1.1. Zakładamy, że macierz TF będzie zawierać informacje o odniesieniu poszczególnych gatunków w_0, w_1, w_2 *do książek. Przy obliczaniu wartości funkcji IDF liczba dokumentów* N *to liczba wszystkich książek w zbiorze* P*, natomiast* n_k *to liczba książek, które reprezentują cechy gatunku* w_k*.*

Zatem wektory wartości odpowiadające poszczególnym książkom mają postać

$$\mathbf{w}_{p_1} = [1, 0.8, 0.01],\ \mathbf{w}_{p_2} = [1, 1, 0],\ \mathbf{w}_{p_3} = [1, 0.88, 0.02],$$

$$\mathbf{w}_{p_4} = [1, 0.2, 0.9],\ \mathbf{w}_{p_5} = [1, 0, 1],\ \mathbf{w}_{p_6} = [1, 0.7, 0.2].$$

Dla każdego użytkownika j wyznaczamy wektor parametrów $\mathbf{w}_{u_j} \in \mathbb{R}^3$, który przedstawia przynależność opinii użytkownika do elementów wektora własności. Preferencje czytelników zostaną więc opisane za pomocą wektorów:

$$\mathbf{w}_{u_1},\ \mathbf{w}_{u_2},\ \mathbf{w}_{u_3},\ \mathbf{w}_{u_4},\ \mathbf{w}_{u_5},\ \mathbf{w}_{u_6}.$$

Aby wygenerować profile użytkowników posłużmy się algorytmem z sekcji 4.1.2. Załóżmy, że macierz $\mathbb{A}$ przyjmuje postać:

Czytelnicy		$\mathbf{u_1}$	$\mathbf{u_2}$	$\mathbf{u_3}$	$\mathbf{u_4}$	$\mathbf{u_5}$	$\mathbf{u_6}$
Książki	$\mathbf{p_1}$	*1*	*1*	*1*	*1*	*0*	*1*
	$\mathbf{p_2}$	*1*	*1*	*1*	*1*	*0*	*1*
	$\mathbf{p_3}$	*0*	*1*	*0*	*1*	*1*	*1*
	$\mathbf{p_4}$	*1*	*0*	*0*	*1*	*1*	*0*
	$\mathbf{p_5}$	*1*	*0*	*0*	*1*	*1*	*0*
	$\mathbf{p_6}$	*1*	*1*	*1*	*1*	*1*	*1*

oraz, że $TFIDF(w_k, p_i)$ jest jak w początkowych założeniach:

Gatunki		$\mathbf{w_1}$	$\mathbf{w_1}$	$\mathbf{w_2}$
Książki	$\mathbf{p_1}$	*1*	*0.8*	*0.01*
	$\mathbf{p_2}$	*1*	*1*	*0*
	$\mathbf{p_3}$	*1*	*0.88*	*0.02*
	$\mathbf{p_4}$	*1*	*0.2*	*0.9*
	$\mathbf{p_5}$	*1*	*0*	*1*
	$\mathbf{p_6}$	*1*	*0.7*	*0.2*

Zatem:

$$\mathbb{B} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 0.8 & 1 & 0.88 & 0.2 & 0 & 0.7 \\ 0.01 & 0 & 0.02 & 0.9 & 1 & 0.2 \end{bmatrix} \cdot \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

$$\mathbb{B} = \begin{bmatrix} 5 & 4 & 3 & 6 & 4 & 4 \\ 1.1 & 3.4 & 2.5 & 3.6 & 1.8 & 3.4 \\ 2.1 & 0.2 & 0.2 & 2.1 & 2.1 & 0.2 \end{bmatrix}$$

Obliczmy następnie ocenę jaką książce p_3 wystawiłby użytkownik u_1 wiedząc, że wektor preferencji użytkownika u_1 jest postaci $\mathbf{w}_{u_1} = [5, 1.1, 2.1]^T$. Użytkownik ten preferuje więc książki gatunku w_0, gdy książki gatunków w_1 i w_2 są dla niego mniej atrakcyjne. Zatem:

$$\widehat{o}_{1,3} = \mathbf{w}_{u_1}^T \mathbf{w}_{p_3} = [5, 1.1, 2.1] \cdot [1, 0.88, 0.02]^T = 5 \cdot 1 + 1.1 \cdot 0.88 + 2.1 \cdot 0.02 = 6.01.$$

Przewidywaną oceną jest zatem 6.01.

Po przeprowadzeniu podobnych obliczeń dla wszystkich wcześniej nieznanych ocen możemy zarekomendować naszemu użytkownikowi nową lekturę.

4.2 Filtrowanie kolaboratywne - Collaborative filtering

Rozważania na temat filtrowania kolaboratywnego zostały przeprowadzone na podstawie książki Gorakala S. K.: *Building Recommendation Engines* [3, Sec 3].

Podejście kolaboratywne omija niektóre ograniczenia występujące w metodach opartych na treści. Dzięki temu systemowi możemy dokonywać rekomendacji z pominięciem wektorów preferencji.

4.2.1 Filtrowanie kolaboratywne oparte na użytkowniku

Algorytm 4.18. *Stworzenie rekomendacji filtrowania kolaboratywnego opartej na użytkownikach wykonamy w następujących krokach:*

1. *wybór użytkowników $u_j, u_k \in U$ dla $j, k \in \mathbb{N}$, między którymi chcemy obliczyć podobieństwo,*

2. *wybór przedmiotów $p_i \in P$ dla $i \in \mathbb{N}$, dla których znane wartości funkcji $R(p_i, u_j)$ i $R(p_i, u_k)$,*

3. *stworzenie wektorów ocen $o_{j,k}^{(j)}$ i $o_{j,k}^{(k)}$ dla użytkowników u_j i u_k wybranych w kroku 1., których elementy stanowią wartości $R(p_i, u_j)$ oraz $R(p_i, u_k)$, gdzie p_i to przedmioty wybrane w kroku 2.,*

4. *wyznaczenie odległości między czytelnikami u_j i u_k - najczęstszymi stosowanymi podejściami do obliczania odległości są metryka euklidesowa i współczynnik korelacji Pearsona,*

5. *wyznaczenie macierzy odległości $\mathbb{U}_1$ między wszystkimi czytelnikami ze zbioru U,*

6. *wyznaczenie macierzy odległości* $\mathbb{U}_2$ *między czytelnikami poprzez normalizację danych w celu uzyskania wartości z przedziału* $[0,1]$*, wyrazy macierzy przyjmują wartości:*

$$u_{ij}^{(2)} = \frac{u_{ij}^{(1)}}{\max_{o_i}\{o_i : o_i \in O,\ i \in \mathbb{N}\} - \min_{o_i}\{o_i : o_i \in O,\ i\ \in \mathbb{N}\}},$$

gdzie $u_{ij}^{(1)}$ *i* $u_{ij}^{(2)}$ *są odpowiednio elementami macierzy* $\mathbb{U}_1$ *i* $\mathbb{U}_2$ *dla* $i, j \in \mathbb{N}$*,*

7. *wyznaczenie macierzy podobieństwa* $\mathbb{U}_3$ *między użytkownikami - zakładając, że największa wartość prawdopodobieństwa to* 1*, macierz podobieństwa przyjmuje wartości:*

$$u_{ij}^{(3)} = 1 - u_{ij}^{(2)},$$

gdzie $u_{ij}^{(2)}$ *i* $u_{ij}^{(3)}$ *są odpowiednio elementami macierzy* $\mathbb{U}_2$ *i* $\mathbb{U}_3$*,*

8. *wyestymowanie nieznanych wartości funkcji* R *dla* $u_j \in U$*,* $j \in \mathbb{N}$ *oraz* $p_i \in P$*,* $i \in \mathbb{N}$ *- niech* u_j *będzie konkretnie ustalonym użytkownikiem, w celu obliczenia brakujących wartości funkcji* R *dla użytkownika* u_j *obliczmy średnią ważoną wykorzystując oceny i przyjmując wartości podobieństwa między* u_j *i innymi użytkownikami jako wagi.*

W celu dokładniejszego zrozumienia rozważmy ponownie przykład 4.1.

Przykład 4.3. *Chcąc obliczyć podobieństwo między użytkownikiem* u_2 *i* u_3 *wybierzmy książki, które zostały przeczytane przez obu użytkowników. W tym przypadku są to:* p_2*,* p_3*,* p_4*,* p_5*,* p_6*. Wektorami ocen uwzględniającymi książki ocenione przez obu użytkowników są więc odpowiednio: dla użytkownika* u_2 *wektor* $o_{2,3}^{(2)} = [6,7,10,6,7]^T$ *oraz dla użytkownika* u_3 *wektor* $o_{2,3}^{(3)} = [6,8,10,6,7]^T$*.*

Obliczamy odległość euklidesową między użytkownikami u_2 *i* u_3*:*

$$d_e(o_{2,3}^{(2)}, o_{2,3}^{(3)}) = \sqrt{(6-6)^2 + (7-8)^2 + (10-10)^2 + (6-6)^2 + (7-7)^2} = \sqrt{1} = 1.$$

Postępując w podobny sposób dla każdej z par użytkowników otrzymamy następującą macierz odległości $\mathbb{U}_1$*:*

$\mathbb{U}_1$	$\mathbf{u_1}$	$\mathbf{u_2}$	$\mathbf{u_3}$	$\mathbf{u_4}$	$\mathbf{u_5}$	$\mathbf{u_6}$
$\mathbf{u_1}$	*0*	*5,099*	*4,243*	*3*	*4,359*	*3,606*
$\mathbf{u_2}$	*5,099*	*0*	*1*	*4,796*	*5,196*	*5,745*
$\mathbf{u_3}$	*4,243*	*1*	*0*	*3,873*	*5*	*5,477*
$\mathbf{u_4}$	*3*	*4,796*	*3,873*	*0*	*2,828*	*3,742*
$\mathbf{u_5}$	*4,359*	*5,196*	*5*	*2,828*	*0*	*3*
$\mathbf{u_6}$	*3,606*	*5,745*	*5,477*	*3,742*	*3*	*0*

.

W procesie normalizacji danych dzielimy elementy macierzy przez

$$(\max\{0,1,2,3,4,5,6,7,8,9,10\} - \min\{0,1,2,3,4,5,6,7,8,9,10\}) = 10$$

i otrzymujemy macierz $\mathbb{U}_2$ *postaci:*

$\mathbb{U}_2$	$\mathbf{u_1}$	$\mathbf{u_2}$	$\mathbf{u_3}$	$\mathbf{u_4}$	$\mathbf{u_5}$	$\mathbf{u_6}$
$\mathbf{u_1}$	*0*	*0,5099*	*0,4243*	*0,3*	*0,4359*	*0,3606*
$\mathbf{u_2}$	*0,5099*	*0*	*0,1*	*0,4796*	*0,5196*	*0,5745*
$\mathbf{u_3}$	*0,4243*	*0,1*	*0*	*0,3873*	*0,5*	*0,5477*
$\mathbf{u_4}$	*0,3*	*0,4796*	*0,3873*	*0*	*0,2828*	*0,3742*
$\mathbf{u_5}$	*0,4359*	*0,5196*	*0,5*	*0,2828*	*0*	*0,3*
$\mathbf{u_6}$	*0,3606*	*0,5745*	*0,5477*	*0,3742*	*0,3*	*0*

Macierz podobieństwa $\mathbb{U}_3$ *przyjmuje więc wartości:*

$\mathbb{U}_3$	$\mathbf{u_1}$	$\mathbf{u_2}$	$\mathbf{u_3}$	$\mathbf{u_4}$	$\mathbf{u_5}$	$\mathbf{u_6}$
$\mathbf{u_1}$	*1*	*0,4901*	*0,5757*	*0,7*	*0,5641*	*0,6394*
$\mathbf{u_2}$	*0,4901*	*1*	*0,9*	*0,5204*	*0,4804*	*0,4255*
$\mathbf{u_3}$	*0,5757*	*0,9*	*1*	*0.6127*	*0,5*	*0,4523*
$\mathbf{u_4}$	*0,7*	*0,5204*	*0.6127*	*1*	*0,7172*	*0,6258*
$\mathbf{u_5}$	*0,5641*	*0,4804*	*0,5*	*0,7172*	*1*	*0,7*
$\mathbf{u_6}$	*0,6394*	*0,4255*	*0,4523*	*0,6258*	*0,7*	*1*

Obliczamy ocenę jaką użytkownik u_1 *zaproponuje dla książki* p_2*:*

$$\frac{0,4901 \cdot 6 + 0,5757 \cdot 6 + 0,7 \cdot 5 + 0,5641 \cdot 6}{0,4901 + 0,5757 + 0,7 + 0,5641} = 5,7$$

Na podstawie metody filtrowania kolaboratywnego opartej na użytkownikach wnioskujemy, że użytkownik u_1 *wystawiłby książce* p_2 *ocenę* $5,7$*.*

Postępując w analogiczny sposób przewidzimy wszystkie brakujące oceny :

Czytelnicy		$\mathbf{u_1}$	$\mathbf{u_2}$	$\mathbf{u_3}$	$\mathbf{u_4}$	$\mathbf{u_5}$	$\mathbf{u_6}$
Książki	$\mathbf{p_1}$	*6*	*3*	***4,57***	*6*	*4*	***4,88***
	$\mathbf{p_2}$	***5,7***	*6*	*6*	*5*	*6*	***5,72***
	$\mathbf{p_3}$	*7*	*7*	*8*	*7*	*8*	*9*
	$\mathbf{p_4}$	*8*	*10*	*10*	*7*	*6*	*8*
	$\mathbf{p_5}$	*9*	*6*	*6*	*6*	*6*	***6,67***
	$\mathbf{p_6}$	*5*	*7*	*7*	*5*	*4*	*2*

Możemy więc wnioskować, że w tym przypadku dla użytkownika u_3 *książka* p_1 *prawdopodobnie nie będzie zbyt atrakcyjna. Użytkownik* u_6*, natomiast, z chęcią przeczyta książkę* p_5*.*

4.2.2 Filtrowanie kolaboratywne oparte na przedmiotach

W przypadku filtrowania kolaboratywnego opartego na elementach wartości podobieństwa między użytkownikami zostaje zastąpiona przez wartości podobieństwa między elementami.

Algorytm 4.19. *W tym rodzaju rekomendacji należy wykonać następujące kroki:*

1. *wybór przedmiotów $p_i, p_k \in P$, $i, k \in \mathbb{N}$, dla których znamy wartość funkcji $R(p_i, u_j)$ oraz wartość funkcji $R(p_k, u_j)$, gdzie u_j jest użytkownikiem, który wystawia ocenę w obu przypadkach,*

2. *stworzenie wektorów ocen $\overline{o^{(p_i)}}$ i $\overline{o^{(p_k)}}$ dla przedmiotów p_i i p_k wybranych w kroku 2., których elementy stanowią stanowią wartości funkcji R dla użytkownika u_j,*

3. *wyznaczenie odległości między przedmiotami p_i i p_k - najczęstszymi stosowanymi podejściami do obliczania odległości jest podobieństwo kosinusów,*

4. *wyznaczenie macierzy podobieństwa $\mathbb{P}$ między wszystkimi przedmiotami P,*

5. *wyestymowanie nieznanych wartości funkcji R dla $u_j \in U$ dla $j \in \mathbb{N}$ oraz $p_i \in P$ dla $i \in \mathbb{N}$ - niech p_i będzie konkretnie ustalonym przedmiotem oraz u_j będzie konkretnie ustalonym użytkownikiem. W celu obliczenia brakujących wartości funkcji R dla u_j i p_i obliczmy średnią ważoną wykorzystując oceny oraz przyjmując wartości podobieństwa między p_i i innymi przedmiotami ocenionymi przez użytkownika jako wagi.*

Przykład 4.4. *Aby obliczyć podobieństwo między książkami p_1 i p_2 wyznaczmy wektory ocen w których uwzględnimy przypadki, gdzie jeden użytkownik ocenił obie pozycje.*

Zatem: $\overline{o^{(p_1)}} = [3, 6, 4]^T$, $\overline{o^{(p_2)}} = [6, 5, 6]^T$.

Następnie używając wzoru na podobieństwo kosinusowe obliczamy podobieństwo między wybranymi książkami

$$sim(p_1, p_2) = \frac{\overline{o^{(p_1)}} \cdot \overline{o^{(p_2)}}}{|o^{(p_1)}||o^{(p_2)}|} = \frac{3 \cdot 6 + 6 \cdot 5 + 4 \cdot 6}{\sqrt{6^2 + 3^2 + 6^2 + 4^2}\sqrt{6^2 + 6^2 + 5^2 + 6^2}} = 0,6339.$$

Postępując w analogiczny sposób otrzymamy macierz podobieństwa:

$\mathbb{P}$	$\mathbf{p_1}$	$\mathbf{p_2}$	$\mathbf{p_3}$	$\mathbf{p_4}$	$\mathbf{p_5}$	$\mathbf{p_6}$
$\mathbf{p_1}$	*1*	*0,6339*	*0,7372*	*0,7195*	*0,8935*	*0,7599*
$\mathbf{p_2}$	*0,6339*	*1*	*0,7951*	*0,8150*	*0,7977*	*0,8898*
$\mathbf{p_3}$	*0,7372*	*0,7951*	*1*	*0,9780*	*0,8586*	*0,9200*
$\mathbf{p_4}$	*0,7195*	*0,8150*	*0,9780*	*1*	*0,8860*	*0,9681*
$\mathbf{p_5}$	*0,8935*	*0,7977*	*0,8586*	*0,8860*	*1*	*0,9413*
$\mathbf{p_6}$	*0,7599*	*0,8898*	*0,9200*	*0,9681*	*0,9413*	*1*

.

Wyestymujmy teraz ocenę jaką użytkownik u_6 zaproponuje dla książki p_2. Ponownie obliczymy średnią ważoną ocen, tym razem, wykorzystując wartość podobieństwa między książką p_1, a książkami ocenionymi wcześniej przez użytkownika oraz oceny jakie nadał on tym pozycjom:

$$\frac{(0,7951 \cdot 9 + 0,8150 \cdot 8 + 0,8898 \cdot 2)}{(0,7951 + 0,8150 + 0,8898)} = 6,16.$$

Na podstawie przeprowadzonych obliczeń zakładamy, że ocena jaką wystawiłby po przeczytaniu użytkownik u_6 książce p_2 to $6,16$.

Powtarzając powyższe obliczenia dla każdej z pozycji wcześniej nieocenionej przez wybranego użytkownika otrzymamy wszystkie brakujące opinie. Następnie bazując na zdobytych danych z łatwością odnajdziemy pozycję najbardziej odpowiednią do zarekomendowania użytkownikowi.

4.3 Systemy rekomendujące kontekstowe - Context–aware recommender systems

Rozważania w tej sekcji zostały również przeprowadzone na podstawie książki Gorakala S. K.: *Building Recommendation Engines* [3, Sec 3].

Uprzednio opisane metody opierały się głownie na rozważaniu problemów dwuwymiarowych. W tym podejściu, przez dodanie nowego wymiaru, jakim jest kontekst (K), zaczynamy rozważać problemy trójwymiarowe:

$$R : U \times P \times K \to O$$

Algorytm 4.20. *W modelu kontekstowym rekomendacje są generowane w następujący sposób:*

1. *za pomocą algorytmu systemów rekomendujących opartych na treści zostają wygenerowana lista rekomendacji bazująca na preferencjach użytkownika,*
2. *odfiltrowanie rekomendacji, które odpowiadają przyjętemu kontekstowi - wyróżniamy tutaj dwa podejścia:*
   - *filtrowanie jako etap wstępny (ang. Pre-Filtering) - informacje kontekstowe są tu używane do odfiltrowania najbardziej istotnych informacji i skonstruowania dwuwymiarowego zbioru danych,*
   - *filtrowanie jako etap końcowy (ang. Post-Filtering) - informacje o kontekście są ignorowane w wejściowych danych, rekomendacja dokonywana jest na całym zbiorze, a w następnym kroku lista rekomendacji stworzona dla użytkownika jest zawężana przez uwzględnienie kontekstu.*

4.4 Dekompozycja macierzy ocen - SVD

Poniższe rozważania zostały przeprowadzone na podstawie publikacji Desrosiers K. i Karypis G. *A comprehensive survey of neighborhood-based recommendation methods* [2, Sec 4.1.1].

Ideą aproksymacji macierzy preferencji $\mathbb{O}$ w normie Frobeniusa jest aproksymacja macierzy preferencji $\mathbb{O}$ o wymiarach $|P| \times |U|$ i $\operatorname{rz}(\mathbb{O}) = n$ przez macierz $\widehat{\mathbb{O}}$ taką, że $\operatorname{rz}\left(\widehat{\mathbb{O}}\right) = k$, $k < n$.

Zatem

$$\widehat{\mathbb{O}} = \mathbb{P}\mathbb{Q}^T,$$

gdzie:

- $\mathbb{P}$ jest macierzą zawierającą koordynaty użytkowników,
- $\mathbb{Q}$ jest macierzą zawierającą koordynaty przedmiotów.

Intuicyjnie, u-ty rząd macierzy $\mathbb{P}$, $\mathbf{p}_u \in \mathbb{R}^k$, reprezentuje współrzędne użytkownika u rzutowane w k-wymiarowej przestrzeni. Podobnie i-ty wiersz macierzy $\mathbb{Q}$, $\mathbf{q}_i \in \mathbb{R}^k$, reprezentuje współrzędne przedmiotu i w tej przestrzeni.

Macierze $\mathbb{P}$ i $\mathbb{Q}$ są odnajdowane przez minimalizowanie błędu przybliżenia zdefiniowanego przez kwadrat normy Frobeniusa:

$$\mathrm{E}(\mathbb{P}, \mathbb{Q}) = ||\mathbb{O} - \mathbb{P}\mathbb{Q}^T||_F^2 = \sum_{u,i}(o_{u,i} - \mathbf{p}_u\mathbf{q}_i^T)^2.$$

Podejście to jest równoważne z wyprowadzeniu SVD macierzy $\mathbb{O}$:

$$\mathbb{T} = \mathbb{U}\Sigma\mathbb{V}^T,$$

gdzie:

- $\mathbb{U}$ jest lewą macierzą wektorów szczególnych,
- $\mathbb{V}$ jest prawą macierzą wektorów szczególnych,
- Σ jest $(m \times n)$-wymiarową macierzą wartości osobliwych.

Przez $\Sigma_k, \mathbb{U}_k, \mathbb{V}_k$ oznaczmy macierze uzyskane w wyniku wyboru k największych wartości osobliwych oraz ich odpowiednich wektorów.

Macierze $\mathbb{P}$ i $\mathbb{Q}$ odpowiadają więc postaciom:

$$\mathbb{P} = \mathbb{U}_k\sqrt{\Sigma_k},$$

oraz

$$\mathbb{Q} = \mathbb{V}_k\sqrt{\Sigma_k}.$$

Predykcja oceny zostaje dokonana na podstawie równania:

$$o_{u,i} = \mathbf{p}_u \mathbf{q}_i^T.$$

Główny problem jaki pojawia się przy implementacji metody SVD jest brak dużej liczby wyrazów macierzy $\mathbb{O}$. Rozwiązaniem tego problemy jest odnalezienie brakujących elementów macierzy $\mathbb{P}$ i $\mathbb{Q}$ używając znanych ocen oraz równania:

$$\mathrm{E}\left(\mathbb{P}, \mathbb{Q}\right) = \sum_{o_{u,i} \in \mathbb{O}} (o_{u,i} - \mathbf{p}_u \mathbf{q}_i^T)^2 + \lambda(\|\mathbf{p}_u\|^2 + \|\mathbf{q}_i\|^2),$$

gdzie λ jest parametrem kontroli poziomu regularyzacji.

To samo podejście może zostać zastosowane w przypadku obliczania podobieństwa między użytkownikami lub przedmiotami w metodzie filtrowania opartego na treści.

Rozwiązujemy tutaj następujący problem:

$$\mathrm{E}\left(\mathbb{P}, \mathbb{Q}\right) = \sum_{z_{u,i} \in \mathbb{O}} (o_{u,i} - \mathbf{p}_u \mathbf{q}_i^T)^2,$$

gdzie:

- $\forall_{u \in U} \ \|\mathbf{p}_u\| = 1$,
- $\forall_{i \in P} \ \|\mathbf{q}_i\| = 1$,
- z_{ui} jest średnią ocen o_{ui} znormalizowaną do zakresu $[-1, 1]$.

Rozdział 5

Eksperymenty / cześć praktyczne

Rozważany problem rekomendacji może być sformułowany jako problem uczenia maszynowego, w którym znane są oceny jakie użytkownicy wystawili pewnym przedmiotom, i którego zadaniem jest predykcja ocen użytkowników dla elementów przez nich nieocenionych.

Załóżmy, że mamy n użytkowników i m przedmiotów. Otrzymujemy $n \times m$ - wymiarową macierz $\mathbb{O}$, w której wyrazy $o_{i,j}$ są wartościami funkcji $R(u_i, p_j)$ wystawioną przez użytkownika u_i elementowi p_j, $j \in \{1, \ldots, m\}$, $i \in \{1, \ldots, n\}$. Naszym celem jest wypełnić macierz $\mathbb{O}$ brakującymi ocenami.

5.1 ALS z Apache Spark i MLlib

5.1.1 Apache Spark

Apache Spark to ciesząca się ostatnio dużą popularnością platforma obliczeniowa stworzona w celu przetwarzania dużych zbiorów danych (BigData). Powstała ona w odpowiedzi na MapReduce wykorzystywaną przez Apache Hadoop. Wspomniany MapReduce przetwarza dane w trybie wsadowym co oznacza, że podczas każdej operacji są one wczytywane i zapisywane na dysku (HDFS), przez co spada znacznie jego wydajność przy algorytmach iteracyjnych. W przypadku Apache Spark i głównej jego idei jaką jest Resilient Distributed Dataset zbiory danych są wczytywane do pamięci i dzięki temu są wykorzystywane przez kolejne kroki algorytmu bez konieczności ponownego wczytywania ich na dysk. Zwiększa to znacznie wydajność i szybkość wykonywania operacji.

Jedną z głównych bibliotek Apache Spark jest biblioteka MLlib. Jest to biblioteka uczenia maszynowego, której celem jest uczynić je łatwym i skalowalnym. MLLib zapewnia narzędzia do obsługi algorytmów klasyfikacji, regresji, klastrowania, redukcji

wymiaru, narzędzia algebry liniowej, statystyki i wiele innych. Biblioteka ta wspiera również narzędzia do obsługi reguł rekomendujących, a w szczególności filtrowania kolaboratywnego.

5.1.2 ALS i MLlib

Alternating Least Square (ALS) jest algorytmem faktoryzacji macierzy, który został zaimplementowany bibliotece uczenia maszynowego MLlib należącej do Apache Spark. Algorytm ten został opracowany z myślą o rozwiązywaniu problemów filtrowania na dużą skalę. Jest prosty, a zarazem dobrze skalowalny w stosunku do dużych zbiorów danych.

Niech macierz $\mathbb{O} = [o_{i,j}]_{m \times n}$, gdzie $o_{i,j}$ oznacza ocenę nadaną przedmiotowi p_j przez użytkownika u_i, m to liczba użytkowników, natomiast n jest liczbą przedmiotów.

Zauważmy, że niektóre wartości macierzy $\mathbb{O}$ są nieznane. W procesie faktoryzacji macierzy $\mathbb{O}$ z liczbą n_f faktorów każdy użytkownik u_i jest utożsamiany z wektorem $\mathbf{u}_i \in \mathbb{R}^{n_f}$ oraz każdy z przedmiotów p_j z wektorem $\mathbf{p}_j \in \mathbb{R}^{n_f}$. Elementy wektora $\mathbf{p}_j$ mierzą stopień w jakim przedmiot p_j posiada czynnik lub cechę, elementy wektora $\mathbf{u}_i$ natomiast powinowactwo użytkownika u_i do każdej z cech lub czynników.

Iloczyn skalarny $\mathbf{u}_i^T \mathbf{p}_j$ przedstawia interakcję między użytkownikiem u_i i przedmiotem p_j przybliżając ocenę użytkownika u_i dla przedmiotu p_j: $o_{i,j} \approx \mathbf{u}_i^T \mathbf{p}_j$.

Oznaczając $\mathbb{U} = [\mathbf{u}_i] \in \mathbb{M}_{n_f,m}(\mathbb{R})$ jako macierz cech użytkowników oraz $\mathbb{P} = [\mathbf{p}_j] \in \mathbb{M}_{n_f,n}(\mathbb{R})$ jako macierz cech przedmiotów staramy się zminimalizować błąd najmniejszych kwadratów postaci [8]:

$$\min_{\mathbb{U},\mathbb{P}} \sum_{o_{i,j}} (o_{i,j} - \mathbf{u}_i^T \mathbf{p}_j)^2 + \lambda (\sum_{u_i} \|\mathbf{u}_i\|^2 + \sum_{p_j} \|\mathbf{p}_j\|^2),$$

gdzie

- $o_{i,j}$ są znanymi elementami macierzy $\mathbb{O}$ wystawionymi przez użytkowników,
- $\lambda(\sum_{u_i} \|\mathbf{u}_i\|^2 + \sum_{p_j} \|\mathbf{p}_j\|^2)$ jest czynnikiem powszechnie stosowanym w funkcji straty zapobiegającym przeuczeniu modelu .

Powyższa funkcja nie jest funkcją wypukłą (ze względu na obiekt $u_i^T p_j$). Ustalając jednak jedną z macierzy $\mathbb{U}$ lub $\mathbb{P}$, otrzymujemy postać kwadratową, którą można rozwiązać. Rozwiązanie zmodyfikowanego problemu gwarantuje monotoniczne obniżenie ogólnej funkcji kosztów. Stosując ten krok naprzemiennie do macierzy $\mathbb{U}$ i $\mathbb{P}$, możemy iteracyjnie poprawiać dopasowanie modelu. Podejście to określamy jako algorytm ALS (Alternating Least Squares).

Algorytm 5.1 (ALS [8]). *Zainicjowanie* $\mathbb{P}$ *z losowymi wartościami.*

Powtarzamy:

- *Wykonujemy:*

$$\mathbf{u}_i = (\sum_{o_{i,j} \in o_{i,*}} \mathbf{p}_j \mathbf{p}_j^T + \lambda \mathbb{I}_k)^{-1} \sum_{o_{i,j} \in o_{i,*}} o_{i,j} \mathbf{p}_j$$

  dla $i = 1, \ldots, n$,

- *wykonujemy:*

$$\mathbf{p}_j = (\sum_{o_{i,j} \in o_{*,j}} \mathbf{u}_i \mathbf{u}_i^T + \lambda \mathbb{I}_k)^{-1} \sum_{o_{i,j} \in o_{*,j}} o_{i,j} \mathbf{u}_i$$

  dla $j = 1, \ldots, m$.

do momentu spełnienia kryteriów końcowych.

Otrzymujemy macierze $\mathbb{U}$ *i* $\mathbb{P}$.

5.1.3 Implementacja algorytmu

Poniższa implementacja została stworzona na podstawie dokumentacji Apache Spark [16]. Dane użyte w implementacji zostały zaczerpnięte ze strony internetowej grouplens.org [17]. Wszystkie element graficzne umieszczone w rozdziale pochodzą ze źródła własnego i stanowią części kodu, którego całość została załączona do pracy w plikach *rekomendacja_podejscie_1.py* i *rekomendacja_podejscie_2.py* (płyta CD). **1.**

Biblioteki:

Załadowanie bibliotek pyspark i MLlib.

- pyspark - język do przeprowadzania eksploracyjnej analizy danych, budowania algorytmów uczenia maszynowego i tworzenia narzędzi ETL,
- MLlib - jest częścią frameworku Apache Spark. Pozwala aplikować uczenie maszynowe na dużych zbiorach danych bez problemów ze skalowalnością. Dysponuje dużą liczbą algorytmów uczenia maszynowego, które można zastosować w zależności od przypadku biznesowego.

```
%python
from pyspark import *
from pyspark.mllib.recommendation import ALS, Rating
```

2. Dane:

Zbiór danych składa się z następujących plików:

- *ratings.dat* w formacie: UserID::MovieID::Rating::Timestamp,

- *users.dat* w formacie: UserID::Gender::Age::Occupation::Zip-code,
- *movies.dat* w formacie: MovieID::Title::Genres.

Podczas tworzenia algorytmu zostaną wykorzystane dwa z trzech plików *ratings.dat* i *movies.dat*. Ze względu na wygodę przetwarzania i czytelność separator w plikach został zamieniony z "::" na ",". Plik *ratings.dat* składa się kolejno z id_użytkownika w zakresie od 1 do 6040, id_filmu w zakresie od 1 do 3952, oceny wystawionej przez użytkownika dla filmu w pięciostopniowej skali i znacznika czasu. Istotne jest, że plik nie zawiera oceny każdego użytkownika dla każdego filmu, ale każdy z użytkowników wystawił co najmniej dwadzieścia ocen. Plik *movies.dat* zawiera id_filmu w zakresie od 1 do 3952, tytuł filmu oraz gatunek.

3. Przygotowanie danych:

Wczytanie pliku *ratings.dat* oraz sprawdzenie ilości obserwacji.

```
%python
oceny = sc.textFile("/FileStore/tables/ratings.dat")

print ('Plik liczy ' + str(oceny.count()) + ' obserwacji.')
```

```
▸ (1) Spark Jobs
Plik liczy 1000209 obserwacji.
```

Po wczytaniu łańcuch danych zostaje podzielony na wieloelementową listę. Proces ten zostaje przeprowadzony przy użyciu dwóch funkcji:

- map() - pobiera jako parametry funkcję oraz listę, a zwraca nową listę, która powstaje w wyniku wywołania funkcji przekazanej w pierwszym parametrze dla każdego elementu listy przekazanej w drugim parametrze,
- split() - dzieli łańcuch znaków na wieloelementową listę, jako argument funkcji podaje się separator. W tym przypadku jest to ','. Domyślnym parametrem funkcja split() jest biały znak.

Z pliku *ratings.dat* wybieramy tylko te dane, które są interesujące pod względem tworzenia modelu. Pominięto tutaj ostatnie pole, które jest znacznikiem czasu. W procesie rekomendacji nie wnosi ono bowiem żadnej wartości, a jest jedynie zbędną informacją, która w efekcie wydłuża czas przetwarzania. Dodatkowo, w celu wypisania elementów nowej listy użyta zostaje funkcja collect(), która zwraca tablicę wszystkich elementów w RDD (ang. Resilient Distributed Dataset). RDD to rdzeń Sparka, zbiór elementów podzielonych na węzły klastra, które mogą być obsługiwane równolegle.

```
%python
oceny2 = oceny.map(lambda l: l.split(',')).map(lambda l: (int(l[0]), int(l[1]), float(l[2])))
for wiersz in oceny2.collect():
   print(wiersz)
```

▸ (1) Spark Jobs

```
(1, 1193, 5.0)
(1, 661, 3.0)
(1, 914, 3.0)
(1, 3408, 4.0)
(1, 2355, 5.0)
(1, 1197, 3.0)
(1, 1287, 5.0)
(1, 2804, 5.0)
(1, 594, 4.0)
(1, 919, 4.0)
(1, 595, 5.0)
(1, 938, 4.0)
(1, 2398, 4.0)
```

Dane zostają przekształcone w obiekty typu Rating, gdzie jako argumenty podajemy id_użytkownika, id_produktu oraz ocenę: Rating(int user, int product, double rating). Obiekt pochodzi z biblioteki MLlib, a jego składowe zawierają dane postaci użytkownik-produkt-wartość.

```
%python
oceny3 = oceny.map(lambda l: l.split(',')).map(lambda l: Rating(int(l[0]), int(l[1]), float(l[2])))
for wiersz in oceny3.collect():
  print(wiersz)
```

▸ (1) Spark Jobs

```
Rating(user=1, product=1193, rating=5.0)
Rating(user=1, product=661, rating=3.0)
Rating(user=1, product=914, rating=3.0)
Rating(user=1, product=3408, rating=4.0)
Rating(user=1, product=2355, rating=5.0)
Rating(user=1, product=1197, rating=3.0)
Rating(user=1, product=1287, rating=5.0)
Rating(user=1, product=2804, rating=5.0)
Rating(user=1, product=594, rating=4.0)
Rating(user=1, product=919, rating=4.0)
Rating(user=1, product=595, rating=5.0)
Rating(user=1, product=938, rating=4.0)
```

Wczytanie pliku *movies.dat*. Zostają tu wykorzystane informacje z pierwszej i drugiej kolumny. Pomijamy parametry dotyczące gatunków filmów, gdyż tak jak dane o czasie są one bezwartościowe w modelu filtrowania kolaboratywnego.

```
%python
filmy = sc.textFile("/FileStore/tables/movies.dat")
filmy2 = filmy.map(lambda l: l.split(','))
filmy3 = filmy2.map(lambda l: (int(l[0]), l[1]))
for wiersz in filmy3.collect():
  print(wiersz)
```

▸ (1) Spark Jobs

```
(1, 'Toy Story (1995)')
(2, 'Jumanji (1995)')
(3, 'Grumpier Old Men (1995)')
(4, 'Waiting to Exhale (1995)')
(5, 'Father of the Bride Part II (1995)')
(6, 'Heat (1995)')
(7, 'Sabrina (1995)')
(8, 'Tom and Huck (1995)')
(9, 'Sudden Death (1995)')
(10, 'GoldenEye (1995)')
(11, 'American President')
(12, 'Dracula: Dead and Loving It (1995)')
(13, 'Balto (1995)')
(14, 'Nixon (1995)')
(15, 'Cutthroat Island (1995)')
(16, 'Casino (1995)')
(17, 'Sense and Sensibility (1995)')
```

Po przygotowaniu danych przechodzimy do budowy algorytmu.

4. Algorytm ALS:

Podzielmy zbiór danych *oceny3* na dwa zbiory : *testowy* i *treningowy*. Zbiór *testowy* będzie stanowił 20% całego zbioru, natomiast zbiór *treningowy* 80%.

```
%python
(treningowy, testowy) = oceny3.randomSplit([0.8, 0.2])
```

Sprawdźmy liczbę rekordów w każdym ze zbiorów.

```
%python

print("Cały zbiór oceny3 liczy " + str(oceny3.count()) + " rekordów, zbiór treningowy to: " +
str(treningowy.count()) +" rekordów, a zbiór testowy: " + str(testowy.count())+" rekordów.")
```

▸ (3) Spark Jobs

```
Cały zbiór oceny3 liczy 1000209 rekordów, zbiór treningowy to: 799876 rekordów, a zbiór testowy: 200333 r
ekordów.
```

Matematyczny opis algorytmu został przedstawiony we wcześniejszej sekcji pracy. Aby zaimplementować algorytm ALS przyjrzyjmy się metodzie

train(ratings, rank, iterations=5, lambda=0.01, blocks=-1, nonnegative=False, seed=None)

z klasy pyspark.mllib.recommendation.ALS. Według dokumentacji metoda przyjmuje następujące parametry:

- ratings – RDD ocen lub (id_użytkownika, id_produktu, ocena) krotka,

- rank - liczba wyszukiwanych cech,
- iterations – liczba iteracji algorytmu ALS - domyślnie przyjmuje wartość 5,
- lambda – parametr regulujący algorytmu ALS - domyślnie przyjmuje wartość 0.01,
- blocks – liczba bloków używanych do przeprowadzania równoległych obliczeń - domyślnie przyjmuje wartość -1, co powoduje użycie automatycznie skonfigurowanej liczby bloków,
- nonnegative – określa, czy należy stosować nieujemne ograniczenia - domyślnie wartość false,
- seed – losowy parametr dla zainicjalizowania modelu faktoryzacji macierzy - domyślnie przyjmuje wartość None, co powoduje użycie czasu systemowego jako wartości parametru.

Przeprowadzimy uczenie modelu faktoryzującego macierzy na zbiorze RDD ocen nadanych przez użytkowników określonemu podzbiorowi produktów. Macierz ocen jest aproksymowana jako iloczyn dwóch macierzy niższego rzędu. Aby rozwiązać rozważany problem algorytm ALS jest uruchamiany iteracyjnie z konfigurowalnym poziomem równoległości.

Algorytm został przeprowadzony z wyborem różnych parametrów aby następnie porównać wyniki i wybrać najlepszy z nich. Poniżej został przedstawiony model dla którego otrzymano najmniejszy błąd średni kwadratowy. Pozostałe modele wykazujące przebieg prób zostały załączone w postaci załącznika do pracy *rekomendacja_podejscie_1.py*.

W próbie tej przyjęto następujące parametry rank = 5 i iterations = 20. Załóżmy również, że seed = 2019. Zbiorem danych (ratings) jest przygotowany wcześniej zbiór treningowy.

```
%python
rank = 5
iterations = 20
model_2 = ALS.train(treningowy, rank, iterations, seed = 2019)
```

Aby ocenić jakoś zaproponowanego model_2 przygotowano zbiór dane_testowe. Zbiór ten stanowić będą wszystkie rekordy ze zbioru *testowego* z pominięciem trzeciej wartości jaką jest ocena.

```
%python
dane_testowe = testowy.map(lambda l: (l[0], l[1]))
for rekord in dane_testowe.collect():
  print(rekord)
```

▸ (1) Spark Jobs

```
(1, 1193)
(1, 661)
(1, 595)
(1, 1035)
(1, 2687)
(1, 720)
```

Używając modelu_2 dokonano predykcji wcześniej usuniętej wartości. Predykcja zostaje przeprowadzona za pomocą metody predictAll(user,product). Metoda ta zwraca listę przewidywanych ocen dla par użytkowników i produktów podanych jako parametr.

```
%python
predykcja_2 = model_2.predictAll(dane_testowe).map(lambda l: ((l[0], l[1]), l[2]))
```

Dokonajmy teraz zestawienia danych pokazując dla każdej z par (id_użytkownika, id_przedmiotu) ocenę pierwotną i ocenę wygenerowaną przez model_2.

```
%python
oceny3_vs_predykcja_2 = testowy.map(lambda l: ((l[0], l[1]), l[2])).join(predykcja_2)
```

Miarą oceny jakości modelu niech będzie średni błąd kwadratowy (MSE).

```
%python
MSE2 = oceny3_vs_predykcja_2.map(lambda l: (l[1][0] - l[1][1])**2).mean()
print("Średni błąd kwadratowy model_2 (MSE) = " + str(MSE2))
```

▸ (1) Spark Jobs

```
Średni błąd kwadratowy model_2 (MSE) = 0.7681980311146712
```

W jednej z prób (wszystkie są dostępne w postaci załącznika *rekomendacja_podejście_2.py*) przyjęto większe parametry: rank = 50 oraz iterations = 100. Warto zauważyć, że większa liczba iteracji jednocześnie zapewnia dokładniejszy wynik. Parametry seed oraz ratings zostają niezmienione. Schemat postępowania zostaje zachowany. Zbiorem treningowym jest jednak teraz cały zbiór *oceny3*.

```
%python
rank = 50
iterations = 100
model_4 = ALS.train(oceny3, rank, iterations, seed = 2019)
```

Dane testowe stanowi teraz cały zbiór *oceny3* z usunięciem trzeciej wartości jaką jest ocena.

```
%python
dane_testowe2 = oceny3.map(lambda l: (l[0], l[1]))
for rekord in dane_testowe.collect():
  print(rekord)
```

▸ (1) Spark Jobs

```
(1, 1193)
(1, 661)
(1, 595)
(1, 1035)
(1, 2687)
(1, 720)
(1, 2340)
(1, 1566)
(1, 1836)
(1, 1028)
(2, 648)
```

5. Wybór modelu:

Z wszystkich modeli wybieramy model_4, gdzie średni błąd kwadratowy model_4 (MSE)=0.27165466938970756 . Mimo, że jest to nietypowe podejście oceny algorytmu użyjmy go do stworzenia rekomendacji. Wybierzmy zatem użytkownika, któremu chcemy zaproponować nowy film. Niech w tym przypadku będzie to użytkownik o id_użytkownika = 3.

```
%python
id_uzytkownika = 3
```

Poniższa funkcja wypisuje wszystkie oceny wystawione przez wybranego użytkownika dla poszczególnych filmów (na zdjęciu można zobaczyć tylko kilka przykładowych).

```
%python
oceny_uzytkownika = oceny3.filter(lambda l: l[0] == id_uzytkownika)

print("Użytkownik: " + str(id_uzytkownika) + " zaproponował dla książki:")
for i in oceny_uzytkownika.collect():
  id_filmu = i[1]
  tytul = filmy3.filter(lambda l: l[0] == id_filmu)
  for j in tytul.collect():
    print("'" + j[1] + "', ocenę: " + str(i[2]) +".")
```

▸ (52) Spark Jobs

```
Użytkownik: 3 zaproponował dla książki:
'Animal House (1978)', ocenę: 4.0.
'Full Monty', ocenę: 2.0.
'Mission: Impossible (1996)', ocenę: 3.0.
'Raising Arizona (1987)', ocenę: 4.0.
'28 Days (2000)', ocenę: 3.0.
```

6. Rekomendacja:

Rekomendacja zostanie dokonana za pomocą metody recommendProducts(int user,int num), gdzie:

- user - id_użytkownika dla którego będzie dokonywana rekomendacja,
- num - liczba rekomendacji, którą chcemy otrzymać. Warto zauważyć, że w niektórych przypadkach liczba elementów może być mniejsza.

Funkcja zwraca obiekty, z których każdy zawiera id_użytkownika, id_produktu i wynik w polu oceny. Każdy z obiektów reprezentuje jeden zalecany produkt, są one sortowane malejąco według pola z przewidzianą oceną. Pierwszy ze zwróconych obiektów jest tym, który najprawdopodobniej będzie najbardziej odpowiadał użytkownikowi. Warto zauważyć, że wynik w polu oceny nie jest z zakresu 1-5. Jest to bowiem nie ocena, a wskaźnik mówiący jak bardzo produkt jest zalecany.

```
%python
print("Najlepsze rekomendacje dla wybranego użytkownika:")
rekomendacje = model_4.recommendProducts(id_uzytkownika,3)
print(rekomendacje)
```

```
▸ (1) Spark Jobs
Najlepsze rekomendacje dla wybranego użytkownika:
[Rating(user=3, product=1627, rating=7.523104223641156), Rating(user=3, product=1649, ratin
g=6.752193042078868), Rating(user=3, product=1046, rating=6.6399033990142176)]
```

Mając oceny przewidziane dla filmów zastąpiono ich id tytułami:

```
%python

print("Najlepszymi rekomendacjami dla użytkownika: " + str(id_uzytkownika) + " są filmy:")
for i in rekomendacje:
  id_filmu = i[1]
  tytul = filmy3.filter(lambda l: l[0] == id_filmu)
  for j in tytul.collect():
    print("'" + j[1] + "', przewidywana ocena: " + str(round(i[2],1)))
```

```
▸ (3) Spark Jobs
Najlepszymi rekomendacjami dla użytkownika: 3 są filmy:
'U Turn (1997)', przewidywana ocena: 7.5
'Fast', przewidywana ocena: 6.8
'Beautiful Thing (1996)', przewidywana ocena: 6.6
```

Pozostało tylko zaproponować użytkownikowi nowy film.

Rozdział 6

Podsumowanie

Celem niniejszej pracy było przedstawienie modeli reguł rekomendujących za pomocą teorii matematycznej.

W rozdziale 3. zaprezentowano wybrane elementy eksploracji danych spotykane w systemach rekomendujących, ze szczególnym naciekiem na rozkład według wartości osobliwych, warunki równoważne temu rozkładowi oraz twierdzenie Eckart - Younga.

W rozdziale 4. zaprezentowano trzy podstawowe metody systemów rekomendujących. Pierwszym były systemy rekomendujące oparte na treści, ukierunkowane na spersonalizowane potrzeby użytkownika oraz treść produktu. Reguły te znajdują zastosowanie w analizie tekstu, czego przykładem był przedstawiony w pracy algorytm TFIDF. Kolejnym rozważanym modelem w rozdziale było filtrowanie kolaboratywne, które pomijało ograniczenia wektorów preferencji, a a sam model pozwalał na zwięzłe i intuicyjne wyjaśnianie obliczeń prognostycznych. Kierowaliśmy się tu ocenami użytkowników wyszukując rekomendacji za pomocą dwóch podejść - filtrowania opartego na użytkownikach i opartego na elementach. Trzecim wspomnianym modelem były systemy rekomendujące kontekstowe, gdzie rozważany problem był trójwymiarowy, z dodatkowym elementem jakim był kontekst. Dla każdego z modeli został zaproponowany algorytm systematyzujący metodę rozwiązania problemu oraz przykład przedstawiający jego zastosowanie.

W rozdziale 5. zaprezentowano algorytm filtrowania kolaboratywnego w kontekście uczenia maszynowego. Problem, którego rozwiązanie chcieliśmy uzyskać - na podstawie zbiorów użytkowników, filmów i ocen zaproponować użytkownikowi kolejną pozycję ze zbioru, która będzie dla niego interesująca. Próbą rozwiązania tego problemu było przygotowanie implementacji algorytmu filtrowania kolaboratywnego przy zastosowaniu języka Python i technologii Big Data - Spark. Schemat wyprowadzony w rozdziale 4. znacznie ułatwił zrozumienie kolejnych kroków i metod implementowanych w kodzie. Ostatecznie, efekty uzyskane przy użyciu utworzonego programu zostały uznane

za zadowalające. Program pozwala na znalezienie ustalonej liczby pozycji, które są odpowiednie do zaproponowania wybranemu użytkownikowi.

Na podstawie analizy rożnych metod nasunął się wniosek, iż ważnym aspektem wykorzystywanym w rekomendacjach jest faktoryzacja macierzy, która pozwala zmniejszyć wymiarowość problemu do najbardziej znaczących cech. Istotnym wkładem własnym, który został zaprezentowany w pracy było przeanalizowanie modelu SVD, wyprowadzenie warunków równoważnych temu rozkładowi oraz dowód twierdzenia Eckard-Younga. Dodatkowo wartościowym dopełnieniem tematu są zaproponowane algorytmy i przygotowanie skryptu zawierającego implementację jednej z rozważanych metod.

Bibliografia

[1] Banaszak G., Gajda W.: *Elementy algebry liniowej część I.* Wydawnictwo Naukowe - Techniczne, 2002.

[2] Desrosiers K., Karypis G.: A comprehensive survey of neighborhood-based recommendation methods. Springer, 2011.

[3] Gorakala S. K.: *Building Recommendation Engines.* Packt, 2016.

[4] Jakubowski J., Sztencel R.: *Wstęp do teorii prawdopodobieństwa.* SCRIPT, 2001.

[5] Kidziński Ł.: Statistical foundation of recommender systems. Master's thesis, University of Warsaw, 2011.

[6] Krzyśko M.: *Wykład z teorii prawdopodobieństwa.* Wydawnictwo Naukowe - Techniczne, 2000.

[7] *Kurs Recommender Systems, Andrew Ng :* `www.youtube.com/watch?v=giIXNoiqO_U&list=PL-6SiIrhTAi6x4Oq28s7yy94ubLzVXabj` (dostęp 19.05.2019).

[8] Lublin M. Pere Y. Haoming L., Bangzheng H.: Matrix completion via alternating least square (ALS). Stanford University, 2015.

[9] Mazeika M.: *The singular value decomposition and low rank approximation.* University of Chicago, 2016.

[10] Morzy T.: *Eksploracja danych. Metody i algorytmy.* Wydawnictwo Naukowe PWN, 2013.

[11] Nowak Brzezińska A.: Analiza skupień. konspekt do zajęć: Statystyczne metody analizy danych. Uniwersytet Śląski, 2012.

[12] O'Brien G.W. Berry M.W., Dumais S.T.: Using linear algebra for intelligent information retrival. SIAM, 1995.

[13] Poreda T., Jędrzejewski J.: *Algebra liniowa z elementami geometrii analitycznej.* Politechnika Łódzka, 2011.

[14] Ricci F. and Rokach L. and Shapira B. and Kantor P.: *Recommender Systems Handbook.* Springer, 2011.

[15] Walesiak M., Gantar E.: *Statystyczna analiza danych z wykorzystaniem programu R.* Wydawnictwo Naukowe PWN, 2009.

[16] *Dokumentacja Apache Spark*
Przewodnik biblioteki MLlib:
`https://spark.apache.org/docs/latest/ml-guide.html`
Informacje o filtrowaniu kolaboratywnym:
`https://spark.apache.org/docs/2.1.0/mllib-collaborative-filtering.html`
`http://spark.apache.org/docs/latest/ml-collaborative-filtering.html?fbclid=IwAR05mYhbkFUTfBE0NScVhNaK1GiM4o8R4nqW1iiDePpp85_OBvKM5QiJ58Y`
Informacje o metodzie predictAll(user_product):
`https://spark.apache.org/docs/2.2.0/api/python/pyspark.mllib.html#pyspark.mllib.recommendation.MatrixFactorizationModel.predictAll`
Informacje o metodzie train(ratings, rank, iterations=5, lambda=0.01, blocks=-1, nonnegative=False, seed=None) :
`https://spark.apache.org/docs/2.1.0/api/python/pyspark.mllib.html#pyspark.mllib.recommendation.ALS`
`https://spark.apache.org/docs/2.1.0/api/python/_modules/pyspark/mllib/recommendation.html#ALS.train`
Informacje o metodzie recommendProducts(int uzytkownik,int numer):
`https://spark.apache.org/docs/2.2.0/api/java/org/apache/spark/mllib/recommendation/MatrixFactorizationModel.html#recommendProducts-int-int-`
(dostęp 24.05.2019).

[17] *Źródło danych:*
`https://grouplens.org/datasets/movielens/` (dostęp 25.05.2019).